

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“SISTEMA DE INFORMACIÓN WEB PARA LA ADMINISTRACIÓN Y PUBLICACIÓN DIGITAL DE REVISTAS CIENTÍFICAS”

CASO: (Dirección de Posgrado U.P.E.A)

Para optar al Título de Licenciatura en Ingeniería de Sistemas

MENCIÓN: INFORMATICA Y COMUNICACIONES

Postulante : Gabriel Limachi Misme
Tutor Metodológico : M.Sc. Ing. Enrique Flores Baltazar
Tutor Especialista : Ing. Walter Emilio Paco Siles
Tutor revisor : Lic. Mario Torrez Cupiticono

EL ALTO – BOLIVIA

2022

DECLARACIÓN JURADA DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, **Gabriel Limachi Misme**, con cedula de identidad **9104987 LP** y registro universitario **14000996**. Estudiante de la carrera de Ingeniería de Sistemas, mediante la presente declaro de manera pública que la propuesta del **PROYECTO DE GRADO** titulado: “**SISTEMA DE INFORMACIÓN WEB PARA LA ADMINISTRACIÓN Y PUBLICACIÓN DIGITAL DE REVISTAS CIENTÍFICAS CASO: DIRECCION DE POSGRADO**”, es original, siendo resultado de mi trabajo personal y no constituye una copia o replica de trabajos similares elaborados.

Autorizo la publicación del resumen de mi propuesta en internet y me comprometo a responder a todos los cuestionamientos que se desprenden de su lectura

Asimismo, me hago responsable ante la universidad o terceros, de cualquiera irregularidad o daño que pudiera ocasionar, por el incumplimiento de lo declarado.

De identificarse falsificación, plagio, fraude, o que el **PROYECTO DE GRADO** haya sido publicado anteriormente, asumo las consecuencias y sanciones que de mi acción se deriven, responsabilizándome por todas las cargas legales que se deriven de ello sometiéndome a las normas establecidas y vigentes de la Carrera de Ingeniería de Sistemas de la Universidad Pública de el Alto.

El Alto, Diciembre de 2022

.....
Gabriel Limachi Misme
C.I. 9104987 LP
e-mail: gabrielmisme@gmail.com

DEDICATORIA

A mis queridos padres, por ser muy comprensivos y brindarme su apoyo incondicional en esta etapa de mi vida, gracias a ellos ha sido posible este trabajo.

AGRADECIMIENTOS

A mi Tutor Metodológico, M. Sc. Ing. Enrique Flores Baltazar por su apoyo y motivación con este proyecto.

A mi Tutor Especialista, Ing. Walter Emilio Paco Siles por la paciencia que me tuvo y compartir sus conocimientos durante el desarrollo de este proyecto.

A mi Tutor Revisor, Lic. Mario Torrez Cupiticoná por su disponibilidad de tiempo, sus observaciones y recomendaciones del presente proyecto.

A la Universidad Pública de El Alto y la carrera de Ingeniería de Sistemas por todos los conocimientos adquiridos y a mis amigos por los momentos más felices.

A la Dirección de Posgrado, por brindarme la oportunidad y la confianza para el desarrollo de este proyecto de grado.

ÍNDICE GENERAL

	PAG (s)
1.1. INTRODUCCIÓN.....	1
1.2. ANTECEDENTES	1
1.2.1. Antecedentes de la Institución	1
1.2.1.1. Universidad Pública de El Alto	1
1.2.1.2. Dirección de Posgrado	2
1.2.2. Antecedentes afines al proyecto	3
1.2.2.1. Internacional	3
1.2.2.2. Nacional.....	4
1.2.2.3. Local	5
1.3. PLANTEAMIENTO DEL PROBLEMA.....	6
1.3.1. Problema principal.....	6
1.3.2. Problemas secundarios	6
1.3.3. Formulación del problema.....	7
1.4. OBJETIVOS	7
1.4.1. Objetivo general	7
1.4.2. Objetivos específicos.....	7
1.5. JUSTIFICACIÓN	8
1.5.1. Justificación técnica.....	8
1.5.2. Justificación económica	8
1.5.3. Justificación social.....	8
1.6. METODOLOGÍA.....	8
1.6.1. Metodología de desarrollo.....	8
1.6.1.1. UWE (UML-Based Web Engineering).....	8
1.6.2. Métricas de calidad del software	9
1.6.2.1. ISO/IEC 25000	9
1.6.3. Estimación de costos.....	10
1.6.3.1. Cocomo II	10
1.6.4. Seguridad del software.....	10
1.6.4.1. ISO 27000	10
1.6.5. Pruebas de software.....	10
1.6.5.1. Pruebas de caja negra	10
1.6.5.2. Pruebas de caja blanca	10
1.6.5.3. Pruebas de rendimiento	11
1.6.6. Métodos de recolección de datos.....	11

1.6.6.1.	La entrevista	11
1.6.6.2.	Cuestionarios y encuestas	11
1.6.6.3.	La observación	11
1.7.	HERRAMIENTAS	11
1.7.1.	PHP Hypertext Pre-Processor.....	11
1.7.2.	Codeigniter	12
1.7.3.	JavaScript.....	12
1.7.4.	Node js	12
1.7.5.	Vue js.....	13
1.7.6.	Boostrap	13
1.7.7.	XAMPP	13
1.7.8.	PostgreSQL	13
1.7.9.	Apache	14
1.7.10.	MagicDraw.....	14
1.8.	LÍMITES Y ALCANCES.....	14
1.8.1.	Límites	14
1.8.2.	Alcances	14
1.9.	APORTES	16
2.1.	INTRODUCCIÓN.....	17
2.2.	SISTEMA.....	17
2.2.1.	Sistemas informáticos	18
2.2.1.1.	Estructura de los sistemas informáticos.....	18
2.3.	APLICACIÓN WEB.....	18
2.3.1.	El cliente	19
2.3.2.	El servidor.....	19
2.3.3.	Importancia del modelo cliente servidor.....	20
2.3.4.	Componentes	21
2.3.5.	Diferencia entre cliente y servidor	22
2.3.6.	Tipos de arquitecturas cliente servidor.....	22
2.3.7.	Ventajas y desventajas	23
2.3.8.	Api Rest.....	24
2.3.8.1.	Rest	25
2.3.8.2.	Principales características de una Api Rest.....	25
2.3.8.3.	Uso de la especificación HTTP	26
2.3.8.4.	Ventajas de una Api Rest.....	27
2.4.	INFORMACIÓN	27
2.5.	PUBLICACIÓN	28

2.5.1.	Publicación digital.....	28
2.5.2.	Revistas digitales.....	29
2.5.2.1.	Ventajas.....	29
2.5.3.	Revistas científicas.....	30
2.6.	METODOLOGIA UWE (UML- Based Web Engineering).....	30
2.6.1.	Características de UWE.....	30
2.6.2.	Fases de la Metodología UWE.....	30
2.6.3.	Modelos de la Metodología UWE.....	32
2.6.3.1.	Modelo de Caso de Uso.....	32
2.6.3.2.	Modelo Conceptual.....	33
2.6.3.3.	Modelo de Navegación.....	34
2.6.3.4.	Modelo de Presentación.....	34
2.6.3.5.	Modelo de Procesos.....	35
2.6.4.	Ventajas y desventajas.....	37
2.7.	MÉTODOS DE PRUEBA DEL SOFTWARE.....	37
2.7.1.	Métodos de prueba de caja blanca.....	37
2.7.2.	Método de prueba de caja negra.....	38
2.7.3.	Prueba de rendimiento.....	39
2.8.	MÉTRICAS DE CALIDAD DEL SOFTWARE.....	39
2.8.1.	Métrica de calidad.....	39
2.8.2.	Medida de la calidad.....	40
2.8.3.	ISO/IEC 25000.....	40
2.8.3.1.	ISO/IEC 2500n – División de Gestión de Calidad.....	41
2.8.3.2.	ISO/IEC 2501n – División de Modelo de Calidad.....	41
2.8.3.3.	ISO/IEC 2502n – División de Medición de Calidad.....	42
2.8.3.4.	ISO/IEC 2503n – División de Requisitos de Calidad.....	42
2.8.3.5.	ISO/IEC 2504n – División de Evaluación de Calidad.....	43
2.8.4.	Ciclo de vida de la calidad del Producto Software.....	44
2.8.5.	ISO/IEC 25010.....	45
2.8.5.1.	Modelo de Calidad del Producto Software (Calidad Interna y Externa).....	45
A)	Adecuación funcional.....	46
B)	Fiabilidad.....	46
C)	Eficiencia en el desempeño.....	47
D)	Facilidad de uso.....	47
E)	Seguridad.....	48
F)	Compatibilidad.....	48
G)	Mantenibilidad.....	49

H)	Portabilidad.....	49
2.8.5.2.	Modelo para la Calidad en Uso	50
A)	Efectividad	50
B)	Eficiencia	50
C)	Satisfacción	51
D)	Libertad de riesgo.....	51
E)	Cobertura de contexto	51
2.8.5.3.	Métricas para la calidad interna, externa y en uso - ISO/IEC 25022 y 25023	52
2.8.5.4.	Definición de características de calidad	87
2.8.5.5.	Definición de subcaracterísticas y atributos de calidad interna/externa más relevantes.....	89
2.8.5.6.	Modelo de indicadores y métricas	92
A)	Niveles de puntuación final para la calidad interna, externa y en uso	93
2.9.	SEGURIDAD DEL SOFTWARE.....	94
2.9.1.	Estándar ISO/IEC 27000	94
2.9.1.1.	ISO/IEC 27001	94
2.9.1.2.	ISO/IEC 27002	94
2.9.2.	Niveles de acceso al sistema	94
2.10.	MÉTODOS DE ESTIMACIÓN DE COSTOS	95
2.10.1.	Modelo Constructivo de Costos (COCOMO II)	95
2.11.	HERRAMIENTAS DE DESARROLLO	96
2.11.1.	Lenguaje de Programación PHP (HYPERTEXT PRE-PROCESSOR).....	96
2.11.1.1.	Ventajas y desventajas.....	97
2.11.2.	Framework Codeigniter	98
2.11.2.1.	MVC.....	98
2.11.2.2.	Ventajas del framework CodeIgniter	99
2.11.3.	Lenguaje de programación JavaScript.....	99
2.11.3.1.	Características de JavaScript.....	99
2.11.4.	Node js	100
2.11.4.1.	Características de Node js	100
2.11.5.	Framework Vue js.....	101
2.11.6.	Framework Bootstrap	101
2.11.7.	Servidor Web Apache	102
2.11.7.1.	Características.....	102
2.11.7.2.	Ventajas y desventajas.....	103
2.11.8.	Gestor de Base de Datos PostgreSQL	103
2.11.8.1.	Características.....	104
2.11.8.2.	Ventajas y desventajas.....	104

3.1.	INTRODUCCIÓN.....	106
3.2.	ANÁLISIS DE REQUERIMIENTO.....	106
3.2.1.	Requerimientos funcionales	106
3.2.2.	Requisitos no funcionales.....	108
3.2.3.	Definición de actores	109
3.2.4.	Descripción de funciones	110
3.3.	DISEÑO DEL SISTEMA.....	111
3.3.1.	Modelo de casos de uso	111
3.3.2.	Modelo conceptual	132
3.3.3.	Modelo de navegación	133
3.3.4.	Modelo de presentación	142
3.4.	DESARROLLO DEL SISTEMA.....	152
3.4.1.	Diseño de la base de datos	153
3.4.2.	Web de administración	154
3.4.2.1.	Inicio de sesión.....	154
3.4.2.2.	Página de inicio	155
3.4.2.3.	Administrar usuario.....	156
3.4.2.4.	Administrar rol y permisos.....	157
3.4.2.5.	Administrar categoría	158
3.4.2.6.	Administrar Revistas y publicación.....	160
3.4.2.7.	Administrar proveedor	161
3.4.2.8.	Administrar compras.....	162
3.4.2.9.	Administrar cliente	163
3.4.2.10.	Administrar ventas	164
3.4.3.	Web pública.....	165
3.4.3.1.	Iniciar sesión.....	165
3.4.3.2.	Registro de usuario	166
3.4.3.3.	Revistas	167
3.4.3.4.	Detalle de revistas	168
3.4.3.5.	Perfil de usuario.....	169
3.4.3.6.	Mis favoritos	170
3.4.3.7.	Mis compras	171
3.5.	IMPLEMENTACIÓN DEL SISTEMA.....	172
3.5.1.	Requerimientos de hardware	172
3.5.2.	Requerimientos de software.....	172
3.5.3.	IMPLANTACIÓN DEL SISTEMA.....	172
3.5.3.1.	Instalación del gestor de base de datos Postgresql.....	173

3.5.3.2.	Instalación del servidor apache	174
3.5.3.3.	Instalación de PHP	175
3.5.3.4.	Configurar la zona horaria del servidor	176
3.5.3.5.	Despliegue del aplicativo	177
3.6.	EL MANTENIMIENTO	178
4.1.	INTRODUCCIÓN.....	179
4.2.	PRUEBAS DE SOFTWARE	179
4.2.1.	Pruebas de caja blanca	179
4.2.2.	Pruebas de caja negra	181
4.2.3.	Pruebas de estrés	183
4.3.	SEGURIDAD SEGUN ISO 27000 (ISO/IEC 27001, ISO/IEC 27002).....	184
1.1.1	Seguridad de la Base de Datos.....	185
1.1.2	Seguridad de Autenticación y de usuario	185
1.1.3	Seguridad de la Aplicación	187
1.1.4	Seguimiento de acciones	188
4.4.	METRICA DE CALIDAD DEL SOFTWARE	189
4.4.1.	Selección de las características y subcaracterísticas de calidad interna más relevantes	189
4.4.2.	Métricas de calidad interna seleccionadas	192
4.4.3.	Ponderación en porcentaje de las características de calidad más relevantes para la calidad interna	193
4.5.	ESTIMACION DE COSTO	196
4.5.1.	Técnica de Punto Función (ISO/IEC 20926).....	196
4.5.2.	COCOMO II	206
4.5.2.1.	Costo del Desarrollo de Software	208
4.5.2.2.	Costo de Implementación del software	210
4.5.2.3.	Costo de Elaboración del software.....	211
4.5.2.4.	Costo Total del software	211
4.6.	ANTES Y DESPUES DEL SISTEMA	212
5.1.	INTRODUCCIÓN.....	213
5.1.1.	CONCLUSIONES.....	213
5.1.2.	RECOMENDACIONES	213
	BIBLIOGRAFÍA.....	215

ÍNDICE DE FIGURAS

Figura 1.1 Organigrama de la dirección de posgrado	3
Figura 2.1 Esquema básico de una aplicación web	19
Figura 2.2 Tecnologías empleadas en el cliente y en el servidor web	20
Figura 2.3 Fases de la Metodología UWE	32
Figura 2.4 Diagrama de Casos de Uso - directorio	33
Figura 2.5 Modelo conceptual - directorio	33
Figura 2.6 Modelo de Navegación - directorio	34
Figura 2.7 Modelo Presentación - directorio	35
Figura 2.8 Modelo Proceso - procesamiento de contacto	36
Figura 2.9 Modelo de flujo del proceso – Crear Contacto	36
Figura 2.10 Métodos de prueba de caja blanca	38
Figura 2.11 Método de prueba de caja negra	39
Figura 2.12 División de norma Iso/iec 25000	41
Figura 2.13 Ciclo de Vida de la Calidad del Producto Software	44
Figura 2.14 Modelo de Calidad del Producto Software	46
Figura 2.15 Modelo de Calidad para Calidad en Uso	50
Figura 3.1 Caso de Uso Web de Administración	111
Figura 3.2 Inicio de sesión	112
Figura 3.3 Administrar usuario	113
Figura 3.4 Administrar roles y permisos.....	114
Figura 3.5 Administrar categoría	115
Figura 3.6 Administrar autor.....	116
Figura 3.7 Administrar revistas y publicación.....	117
Figura 3.8 Administrar proveedores.....	118
Figura 3.9 Administrar compra	119
Figura 3.10 Administrar clientes.....	120
Figura 3.11 Administrar ventas	121
Figura 3.12 Administrar ventas en línea	122
Figura 3.13 Caso de uso web pública	123
Figura 3.14 Inicio de sesión	124
Figura 3.15 Registro de usuario	125

Figura 3.16 Revistas	126
Figura 3.17 Detalle de revista	127
Figura 3.18 Perfil del usuario	128
Figura 3.19 Mis compras.....	129
Figura 3.20 Mis favoritos	130
Figura 3.21 Carrito de compras.....	131
Figura 3.22 Modelo conceptual de la base de datos	132
Figura 3.23 Inicio de sesión	133
Figura 3.24 Administrar de usuario (administrador).....	133
Figura 3.25 Administrar roles y permisos (administrador)	134
Figura 3.26 Administrar categoría (administrador, editor)	134
Figura 3.27 Administrar autor (administrador, editor)	135
Figura 3.28 Administrar revistas y publicación (administrador, editor)	135
Figura 3.29 Administrar proveedores (administrador, vendedor)	136
Figura 3.40 Administrar compras (administrador, vendedor).....	136
Figura 3.41 Administrar clientes (administrador, vendedor)	137
Figura 3.42 Administrar ventas (administrador, vendedor).....	137
Figura 3.43 Administrar ventas en línea (administrador, vendedor)	138
Figura 3.44 Inicio de sesión (lector).....	138
Figura 3.45 Registro de usuario (lector, usuario común).....	139
Figura 3.46 Revistas (lector, usuario común)	139
Figura 3.47 Detalle de revistas (lector, usuario común)	140
Figura 3.48 Perfil del usuario (lector)	140
Figura 3.49 Mis compras (lector).....	141
Figura 3.50 Mis favoritos (lector).....	141
Figura 3.51 Carrito de compras (lector)	142
Figura 3.52 Inicio de sesión	142
Figura 3.53 Administrar de usuario (administrador).....	143
Figura 3.54 Administrar roles y permisos (administrador)	143
Figura 3.55 Administrar categoría (administrador, editor)	144
Figura 3.56 Administrar autor (administrador, editor)	144
Figura 3.57 Administrar revistas y publicación (administrador, editor)	145
Figura 3.58 Administrar proveedores (administrador, vendedor)	145
Figura 3.59 Administrar compra (administrador, vendedor)	146

Figura 3.60 Administrar clientes (administrador, vendedor)	146
Figura 3.61 Administrar ventas (administrador, vendedor).....	147
Figura 3.62 Administrar ventas en línea (administrador, vendedor)	147
Figura 3.63 Inicio de sesión (lector)	148
Figura 3.64 Registro de usuario (usuario común).....	148
Figura 3.65 Revistas (lector)	149
Figura 3.66 Detalle de revistas (lector)	149
Figura 3.67 Perfil de usuario (lector).....	150
Figura 3.68 Mis compras (lector).....	150
Figura 3.69 Mis favoritos (lector).....	151
Figura 3.70 Carrito de compras (lector)	151
Figura 3.71 Modelo físico de la base de datos	153
Figura 3.72 Muestra de código sql de la base de datos (tabla revista)	154
Figura 3.73 Inicio de sesión	154
Figura 3.74 Muestra de código (iniciar sesión)	155
Figura 3.75 Página de inicio (rol de administrador)	155
Figura 3.76 Muestra de código (página de inicio).....	156
Figura 3.77 Administrar usuario (rol de administrador).....	156
Figura 3.78 Muestra de código (Administrar usuario).....	157
Figura 3.79 Administrar rol y permisos (rol de administrador)	157
Figura 3.80 Muestra de código (Administrar rol y permisos)	158
Figura 3.81 Administrar categoría (rol de administrador)	158
Figura 3.82 Muestra de código (Administrar categoría).....	159
Figura 3.83 Administrar autores (rol de administrador).....	159
Figura 3.84 Muestra de código (Administrar autores).....	160
Figura 3.85 Revistas y publicación (rol de administrador)	160
Figura 3.86 Muestra de código (Revistas y publicación)	161
Figura 3.87 Administrar proveedor (rol de administrador)	161
Figura 3.88 Muestra de código (Administrar proveedor)	162
Figura 3.89 Administrar compras (rol de administrador).....	162
Figura 3.90 Muestra de código (Administrar compras).....	163
Figura 3.91 Administrar cliente (rol de administrador)	163
Figura 3.92 Muestra de código (Administrar cliente)	164
Figura 3.93 Administrar ventas (rol de administrador).....	164

Figura 3.94 Muestra de código (Administrar ventas)	165
Figura 3.95 Iniciar sesión (rol lector)	165
Figura 3.96 Muestra de código (iniciar sesión)	166
Figura 3.97 Registro de nuevo usuario lector	166
Figura 3.98 Muestra de código (registro de usuario)	167
Figura 3.99 Revistas disponibles (rol lector)	167
Figura 3.100 Muestra de código (revistas).....	168
Figura 3.101 Detalle de revistas (rol lector)	168
Figura 3.102 Muestra de código (detalle de revistas).....	169
Figura 3.103 Perfil de usuario (rol lector).....	169
Figura 3.104 Muestra de código (perfil de usuario)	170
Figura 3.105 Mis favoritos (rol lector).....	170
Figura 3.106 Muestra de código (mis favoritos).....	171
Figura 3.107 Mis compras (rol lector).....	171
Figura 3.108 Muestra de código (mis compras).....	172
Figura 3.109 Uso de los comandos para la instalación de PostgreSQL	174
Figura 3.110 Uso de los comandos para la instalación del servidor web apache	175
Figura 3.111 Uso de los comandos para la instalación de PHP	176
Figura 3.112 Uso de los comandos para la configuración de la zona horaria	176
Figura 3.113 Transferencia de archivos al servidor	177
Figura 3.114 Página de inicio como administrador	177
Figura 4.1 Diagrama de estado Registro de revistas.....	179
Figura 4.2 Grafo de flujo Registro de revistas.....	180
Figura 4.3 Prueba de estrés de la página de inicio de los usuarios lectores.....	183
Figura 4.4 Parámetros de peticiones 100 usuarios virtuales	184
Figura 4.5 Resultados obtenidos de la prueba con 100 peticiones	184
Figura 4.6 Autenticación en el lado del backend	186
Figura 4.7 Uso de la librería de jwt.....	187
Figura 4.8 Seguridad en rutas http.....	188
Figura 4.9 Prueba de seguridad http.....	188
Figura 4.10 Calidad del producto software categorías y subcategorías	189
Figura 4.11 Aplicación de la matriz de calidad para evaluar la calidad interna	194

ÍNDICE DE TABLAS

Tabla 2.1 Métricas de calidad interna/externa para Adecuación funcional	53
Tabla 2.2 Métricas de calidad interna/externa para Fiabilidad	55
Tabla 2.3 Métricas de calidad interna/externa para Eficiencia en el desempeño.....	58
Tabla 2.4 Métricas de calidad interna/externa para Facilidad de uso.....	61
Tabla 2.5 Métricas de calidad interna/externa para Seguridad	65
Tabla 2.6 Métricas de calidad interna/externa para Compatibilidad	67
Tabla 2.7 Métricas de calidad interna/externa para Mantenibilidad.....	69
Tabla 2.8 Métricas de calidad interna/externa para Portabilidad	73
Tabla 2.9 Métricas de calidad en uso de Efectividad.....	77
Tabla 2.10 Métricas de calidad en uso de Eficiencia	78
Tabla 2.11 Métricas de calidad en uso de Satisfacción	81
Tabla 2.12 Métricas de calidad en uso de Libertad de riesgo	82
Tabla 2.13 Métricas de calidad en uso de Cobertura de contexto.....	86
Tabla 2.14 Definición del nivel de importancia.....	87
Tabla 2.15 Ejemplo del nivel de importancia para las características de calidad interna más relevantes	88
Tabla 2.16 Ejemplo del nivel de importancia para las características de calidad externa más relevantes	88
Tabla 2.17 Ejemplo del nivel de importancia para las características de calidad en uso más relevantes	88
Tabla 2.18 Ejemplo del nivel de importancia de subcaracterísticas y atributos de calidad interna más relevantes	89
Tabla 2.19 Ejemplo del nivel de importancia de subcaracterísticas y atributos de calidad externa más relevantes	90
Tabla 2.20 Ejemplo del nivel de importancia de subcaracterísticas y atributos de calidad en uso más relevantes	91
Tabla 2.21 Ejemplo de ponderación en porcentajes para la calidad interna	93
Tabla 2.22 Niveles de puntuación final para la calidad interna, externa y en uso	93
Tabla 2.23 Coeficiente de coste modelo básico	96
Tabla 2.24 Requisitos Funcionales Web de Administración	106

Tabla 2.25 Requisitos Funcionales Web Público.....	107
Tabla 2.26 Requisitos no Funcionales	108
Tabla 2.27 Lista de Actores.....	109
Tabla 2.28 Descripción de funciones	110
Tabla 2.29 Caso de Uso general Web de Administración	112
Tabla 2.30 Inicio de sesión.....	113
Tabla 2.31 Administrar usuario	114
Tabla 2.32 Administrar roles y permisos	115
Tabla 2.33 Administrar categoría	116
Tabla 2.34 Administrar autor	117
Tabla 2.35 Administrar revistas y publicación	118
Tabla 2.36 Administrar proveedores	119
Tabla 2.37 Administrar compra	120
Tabla 2.38 Administrar clientes	121
Tabla 2.39 Administrar ventas.....	122
Tabla 2.40 Administrar ventas en línea.....	123
Tabla 2.41 Caso de uso web pública	124
Tabla 2.42 Inicio de sesión.....	125
Tabla 2.43 Registro de usuario	126
Tabla 2.44 Revistas.....	127
Tabla 2.45 Detalle de revista.....	127
Tabla 2.46 Perfil del usuario.....	128
Tabla 2.47 Mis compras	129
Tabla 2.48 Mis favoritos	130
Tabla 2.49 Carrito de compras.....	131
Tabla 3.1 Habilitación del repositorio PostgreSQL Apt.....	173
Tabla 3.2 Instalación de PostgreSQL	173
Tabla 3.3 Conexión a PostgreSQL.....	173

Tabla 3.4 Instalación de apache como usuario root	174
Tabla 3.5 Instalación y configuración de PHP 7	175
Tabla 3.6 Configurar la zona horaria del servidor	176
Tabla 4.1 Prueba de caja negra Registro de revistas	182
Tabla 4.2 Prueba de caja negra Registro de autores	182
Tabla 4.3 Características de calidad interna seleccionadas	189
Tabla 4.4 Subcaracterísticas y atributos de calidad interna seleccionadas	190
Tabla 4.5 Subcaracterísticas y atributos de calidad interna seleccionadas	192
Tabla 4.6 Ponderación en porcentajes para la calidad interna	193
Tabla 4.7 Resultado final de calidad interna	195
Tabla 4.7 Niveles de puntuación final para la calidad interna, externa y en uso	196
Tabla 4.8 Funciones según su tipo y su complejidad	197
Tabla 4.9 Módulo inicio de sesión	198
Tabla 4.10 Módulo tablero de información	198
Tabla 4.11 Módulo de usuarios	198
Tabla 4.12 Módulo de roles y permisos.	198
Tabla 4.13 Módulo de categoría.....	199
Tabla 4.14 Módulo de autores.....	199
Tabla 4.15 Módulo de revistas y publicación.	199
Tabla 4.16 Módulo de proveedor.	200
Tabla 4.17 Módulo de compra.....	200
Tabla 4.18 Módulo de clientes.	201
Tabla 4.19 Módulo de ventas.	201
Tabla 4.20 Módulo de ventas en línea.	201
Tabla 4.21 Módulo inicio.	202
Tabla 4.22 Módulo revistas	202
Tabla 4.23 Módulo detalle de las revistas.	202
Tabla 4.24 Módulo inicio de sesión.	202

Tabla 4.25 Módulo registro de usuario.....	202
Tabla 4.26 Módulo perfil de usuario.....	203
Tabla 4.27 Módulo cambiar contraseña.....	203
Tabla 4.28 Módulo favorito del usuario.....	203
Tabla 4.29 Módulo compras del usuario.....	203
Tabla 4.30 Módulo carrito de compras del usuario.....	203
Tabla 4.31 Factor de ponderación	204
Tabla 4.32 Tabla de Valores de ajuste de la complejidad	205
Tabla 4.33 Coeficiente a y c y los exponentes b y d.....	207
Tabla 4.34 Conversión de puntos de Función a KDLC.....	208
Tabla 4.35 Resumen de Cocomo II.....	210
Tabla 4.36 Costos de Implementación.....	211
Tabla 4.37 Costos de Elaboración	211
Tabla 4.38 Costos de Elaboración	212
Tabla 4.39 Antes y después del sistema	212

LISTA DE ABREVIATURAS

API. Application Programming Interface
COCOMO. Constructive Cost Model
EI. Número de Entradas externas
EIF. Numero de interfaces externas
EO. Número de Salidas externas
EQ. Número de Consultas externas
HTML. Lenguaje de Marcas de Hipertexto
HTTP. Hypertext Transfer Protocol
ILF. Número de archivos lógico interno
JSON. JavaScript Object Notation
KLDC. Kilos de Líneas de Código
LDC. Líneas de código
MVC. Modelo Vista Controlador
PFA. Punto de función ajustado
PFSA. puntos de función sin ajustar
PHP. Hypertext Pre-Processor
Rest. Representational State Transfer
SQuaRE. System and Software Quality Requirements and Evaluation
UML. Lenguaje Unificado de Modelado
URI. Uniform Resource Identifier
UWE. UML-based Web Engineering
XML. Extensible Markup Language

RESUMEN

El presente proyecto fue desarrollado para la administración de revistas físicas, la compra y ventas de los mismos, y la publicación de revistas científicas en archivo digital que estén disponible para estudiantes, docentes y público en general las revistas gratuitas y las de compra.

En el presente proyecto tiene como objetivo desarrollar e implementar el sistema de información web para la administración y publicación digital de revistas científicas, para la Dirección de Posgrado de la Universidad Pública de EL Alto, que permita administrar la información de manera más organizada y eficiente.

Para el desarrollo de sistema web e aplicó la metodología UWE, para la fase de diseño se utilizó la herramienta Magicdraw. Para evaluar la calidad del producto software se utilizó la ISO/IEC 25000, para la seguridad del software se utilizó la ISO 27000, la estimación de costo del software se lo hizo con COCOMO II basándonos en puntos de función.

Las herramientas de desarrollo que se usaron fueron: PostgreSQL como gestor de base de datos, PHP 7.4 como lenguaje de programación, Apache 2 como servidor HTTP, Bootstrap 5, Vuejs como framework de frontend y CodeIgniter 4 como framework de backend.

PALABRAS CLAVE

Publicación digital, Revistas, Metodología UWE, ISO/IEC 25000, ISO 27000, COCOMO II.

ABSTRACT

This project was developed for the administration of physical magazines, their purchase and sale, also for the publication of scientific magazines in digital files that are available to students, teachers and the general public, free and purchased magazines.

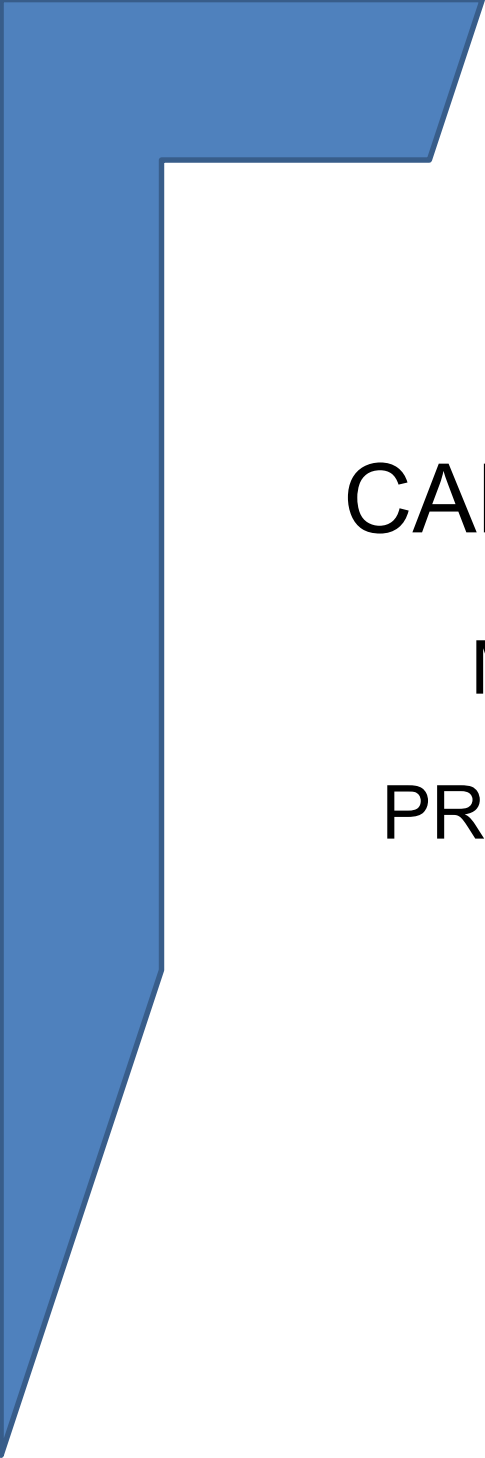
The objective of this project is to develop and implement the web information system for the administration and digital publication of scientific journals, for the Postgraduate Directorate of the Public University of El Alto, which allows managing the information in a more organized and efficient way.

For the development of the web system, the UWE methodology was applied, for the design phase the Magicdraw tool was used. ISO/IEC 25000 was used to evaluate the quality of the software product, ISO 27000 was used for software security, and the software cost estimate was made with COCOMO II based on function points.

The development tools used were: PostgreSQL as database manager, PHP 7.4 as programming language, Apache 2 as HTTP server, Bootstrap 5, Vuejs as frontend framework and CodeIgniter 4 as backend framework

KEYWORDS

Digital publication, Magazines, UWE Methodology, ISO/IEC 25000, ISO 27000, COCOMO II.



CAPÍTULO I
MARCO
PRELIMINAR

1.1. INTRODUCCIÓN

La revista moderna tiene sus raíces en almanaques, panfletos, chapbooks y posters. Entre los años 1663 a 1668, fue publicada la revista alemana Erbauliche monaths-unterredunge que se considera la primera revista del mundo, a pesar de que se pareciese más a un libro con artículos de teología que a una revista en sí, fue la precursora para que revistas de esta misma especie empezaran a aparecer en Francia, Inglaterra e Italia. (Ruiz, 2021)

Actualmente, la Dirección de Posgrado se encarga de gestionar distintos programas de diplomados, maestrías, doctorados y además cuenta con el desarrollo de material intelectual de revistas científicas, tecnológicas, educación y otros.

El presente proyecto tiene por objeto desarrollar un sistema Web de administración y publicación de revistas digitales, usando la metodología de desarrollo UWE (UML-Basado en Ingeniería Web), además del lenguaje de programación de PHP, gestor de base de datos Postgresql, framework Codeigniter, Bootstrap y librerías de JavaScript, para propiciar un acceso completo a una colección de revistas gratuitas con la creación de un cuenta gratis a la comunidad universitaria y público en general, así como al contenido completo de los artículos en archivo digital de las revistas con costo, además de su publicación y adquisición de las revistas impresas.

1.2. ANTECEDENTES

1.2.1. Antecedentes de la Institución

1.2.1.1. Universidad Pública de El Alto

La Universidad Pública de El Alto (UPEA) es una universidad de Bolivia, con sede en la ciudad de El Alto del Departamento de La Paz. La institución de educación superior cuenta con 8 áreas y 37 carreras en las áreas de medicina, ingeniería, socio-político-económica y tecnología. Dentro de su funcionamiento se enmarca en el estatuto orgánico aprobado en 2007. (Upea, 2021)

- **Creación**

El 5 de septiembre de 2000 tras movilizaciones sociales se promulgó la Ley 2115 que determinó la creación de la Universidad Pública de El Alto, la misma determina

que la UPEA tendría autonomía en 5 años, tiempo durante el cual estaría a cargo de un consejo formado por el Ministerio de Educación de Bolivia y otros organismos gubernamentales. De acuerdo a la Ley de su creación, el ente de mayor decisión en la universidad sería el Consejo de Desarrollo Institucional (CDI), mismo en el que estaban insertos miembros de organizaciones sociales de la ciudad de El Alto que tenían poca relación con el quehacer académico. La UPEA inició un proceso de institucionalización, se reinstauró el Consejo Universitario.

- **Ubicación**

La Universidad Pública de El Alto, UPEA, está ubicada en la avenida Sucre, esquina de la calle Pascoe, en la zona Villa Esperanza, a dos cuadras de la avenida Juan Pablo II.

1.2.1.2. Dirección de Posgrado

- **La Visión de Posgrado:**

“El centro de formación de Posgrado de la UPEA, es una institución reconocida de prestigio nacional e internacional, con programas orientados a la innovación institucional, la generación de investigación científica con la comunidad local, regional y nacional, a partir de una gestión y acreditación coherente al desarrollo socioeconómico, tecnológico que responda a los intereses y reclamos de la sociedad”.

- **La Misión de Posgrado:**

“Formar profesionales idóneos y con calidad en diversos niveles de Posgrado recreando la actividad académica, investigación y transferencia e integración tecnológica y el conocimiento, en los procesos de integración social, confrontando y asimilando la incidencia de los valores socioeducativos, para el desarrollo económico, productivo y social a nivel local, regional e internacional”.

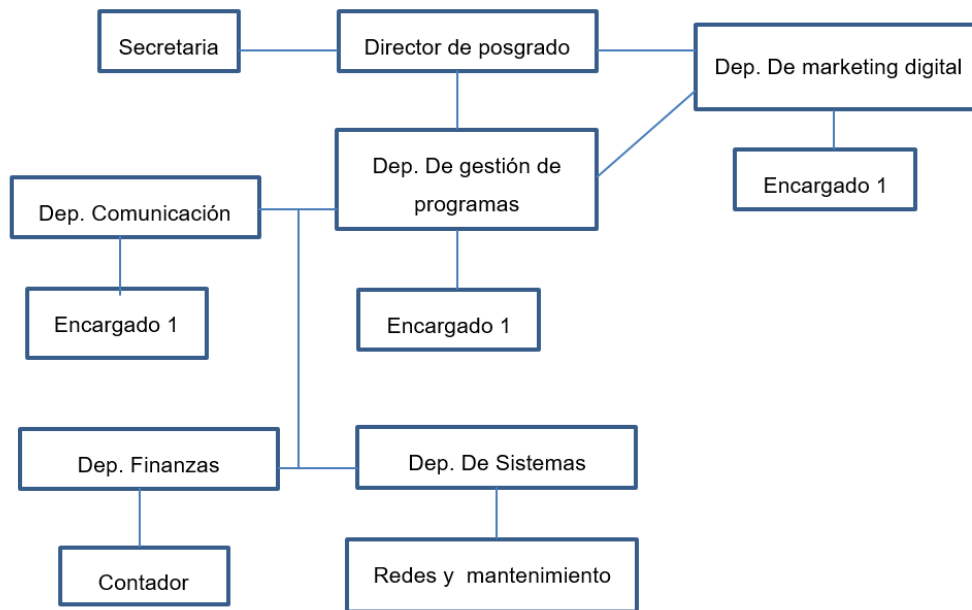
- **Naturaleza**

Se entiende por estudios de Posgrado, toda actividad que tenga por objeto elevar el nivel académico y de desempeño profesional luego de haber obtenido el grado de técnico Superior Universitario o Licenciatura. Tiene como finalidad la formación de profesionales universitarios que profundicen la investigación y el conocimiento, respondan a la demanda social en campos específicos y del ejercicio profesional, además de formar investigadores que contribuyan a los latos fines del desarrollo de la ciencia y la tecnología de país. (Upea, 2021)

- **Organigrama de la dirección de posgrado**

Figura 1.1

Organigrama de la dirección de posgrado



Nota: Se muestra la organización de la dirección de posgrado.

1.2.2. Antecedentes afines al proyecto

1.2.2.1. Internacional

- María Paola Reibàn Morales, 2012, autora de la tesis del trabajo titulado **“Estudio, diseño e implementación de un sistema de comercio electrónico para la empresa Atenas Tours operadora de turismo”**. Plantea una implementación mediante el uso de Sistemas de Gestión

de Contenidos (CMS) para el comercio electrónico. Uno de sus objetivos es el uso de gestor de contenido CMC JadaSite y el uso de una base de datos MySQL. La conclusión que llegó el autor. El sitio web fue diseñado e implementado de acuerdo a las necesidades de la empresa Atenas Tours.

Fue desarrollada en la Universidad Politécnica Salesiana Sede Cuenca Facultad de Ingenierías.

- Andrés Arturo Vergara Salas, 2017, autor del proyecto de grado titulado **“Desarrollo de página web accesible con contenido informativo del programa de ingeniería de sistemas de la universidad libre seccional Bogotá”**. Plantea el uso de la metodología SCRUM a partir de la recolección de la información y herramientas de modelado UML. los objetivos que plantea el autor son, diseñar el sitio web a partir de la información obtenida de los actores, desarrollar el sitio web utilizando la metodología de desarrollo SCRUM y bajo las normas internacionales de la W3C, para cumplir con los estándares de desarrollo WEB. La conclusión que llegó el autor. El sitio web cuenta con un asistente de voz con el cual los usuarios con algún tipo de discapacidad visual congénita o adquirida van a poder acceder al contenido del sitio web.

Fue desarrollada en la Universidad Libre de Colombia Seccional Bogotá.

1.2.2.2. Nacional

- Erick Osvaldo Cadima Cárdenas, 2013, autor de la tesis de trabajo titulado **“Desarrollo de una tienda virtual mediante el estudio comparativo de una tienda física de productos de computación”**. Plantea estudiar la dinámica de la venta virtual, desarrollando una aplicación Web en base a una comparación de una Tienda Física, para facilitar la venta de material de computación, así como ayudar a su gestión, de una forma sencilla y clara para los usuarios y el administrador de la aplicación. Los objetivos que plantea el autor son, realizar un análisis comparativo entre una tienda física y una virtual, diseñar un sitio

web e implementar usando la arquitectura de software (MVC) Modelo Vista Controlador usando el lenguaje de programación PHP.

Fue desarrollada en la Universidad Mayor de San Andres Facultad de Ciencias Puras y Naturales Carrera de Informática en la ciudad de La Paz, Bolivia.

- Juan Antonio Apaza Guachalla, 208, autor proyecto de grado titulado **“Biblioteca Virtual basada en Web Semántica para la Unidad del postgrado en Informática”**. utilizando la metodología de desarrollo de software la OOHDM y el lenguaje de programación Java. Los objetivos que plantea el autor son, desarrollar e implementar módulos de búsqueda, módulo de gestor de contenido, módulo de registros de usuarios y otros.

Fue desarrollada en la Universidad Mayor de San Andres Facultad de Ciencias Puras y Naturales Carrera de Informática en la ciudad de La Paz, Bolivia.

1.2.2.3. Local

- Yhosselyn Yessica Camargo Mita, 2020, autora del proyecto de grado titulado **“Sistema de información web para el control de compras ventas e inventarios”**. Plantea el uso la metodología de desarrollo Uwe, herramientas como: MariaDB, framework Bootstrap, lenguaje de programación PHP, para su desarrollo. Los objetivos que plantea el autor son, brindar información exacta de los productos para realizar el pedido a los proveedores, facilitar el registro de los productos en el proceso de compra, mostrar la información sobre el stock de los productos.

Fue desarrollada en la Universidad Pública de El Alto en la carrera de Ingeniería de Sistemas en la ciudad el Alto, Bolivia.

- Jose Alvaro Mamani Saucedo, 2020, autor del proyecto de grado titulado **“Aplicación web para la administración y catálogo de productos”**.

Plantea el uso de metodologías Uwe, herramientas como: MariaDB, framework Bootstrap, lenguaje de programación Php, para su desarrollo. Los objetivos que plantea el autor son, realizar el análisis de la situación actual de la institución, implementar módulos de adicionar, actualizar, eliminar los registros de los registros, módulo de catálogo para que visualicen los clientes.

Fue desarrollada en la Universidad Pública de El Alto en la carrera de Ingeniería de Sistemas en la ciudad el Alto, Bolivia.

1.3. PLANTEAMIENTO DEL PROBLEMA

Actualmente, la Dirección de Posgrado se encarga de gestionar distintos programas de diplomados, maestrías, doctorados y además cuenta con el desarrollo de material intelectual de revistas científicas, tecnológicas, educación y otros. En la dirección de posgrado se observó que no cuenta con un control adecuado de los registros de las revistas físicas compradas por el proveedor, utilizan la herramienta de Excel para esos registros, además la revistas no están difundidas en internet para el conocimiento del público y las revistas están almacenadas sin darle uso, lo que presenta un problema en gastos de impresión.

1.3.1. Problema principal

- La Dirección de Posgrado no cuenta con herramientas tecnológicas que apoyen en la administración de sus revistas o un sitio web en línea para las publicaciones de las revistas para el conocimiento del público, porque las revistas son almacenadas en hojas de cálculo como Excel, lo que lleva a pérdidas económicas de los ejemplares impresos.

1.3.2. Problemas secundarios

- El manejo de registros en Excel conlleva a la pérdida de información.
- El manejo de la existencia exacta de las revistas no se realiza de forma adecuada, por lo que la información que se tiene no es confiable respecto a las existencias físicas.

- La actualización de registros de las revistas es manual, generando problemas en los niveles de existencias actuales.
- No se cuenta con la promoción o publicidad en internet del material intelectual elaborado por la dirección de posgrado la cual conlleva a pérdidas económicas de los ejemplares físicos que se vende.
- Se dificulta las consultas, reportes y estadísticas en archivo Excel ya que requieren bastante tiempo y esfuerzo de parte del personal, lo que no coadyuva a una buena toma de decisiones en la administración.
- Demoras en la verificación de las compras de las revistas en los registros de Excel.

1.3.3. Formulación del problema

¿Cómo el sistema de información web para la administración y publicación digital de revistas científicas puede ayudar a gestionar el material intelectual generado dentro de la dirección de Posgrado. ?

1.4. OBJETIVOS

1.4.1. Objetivo general

Desarrollar un sistema de información web de administración y publicación digital de revistas científicas para la dirección de Posgrado de la Universidad Pública de El Alto, que ayude a gestionar la publicación de las revistas como la compra, venta de las mismas y además de proporcionar una colección de revistas para la comunidad universitaria y público en general.

1.4.2. Objetivos específicos

- Realizar el análisis de la situación actual de la dirección de Posgrado.
- Diseñar un sistema web de acuerdo a las necesidades y requerimientos de la dirección de Posgrado.
- Realizar la programación del sistema Web mediante los requerimientos funcionales y no funcionales.

- Realizar las pruebas de funcionamiento del sistema Web mediante las técnicas de caja negra y caja blanca.
- Realizar la evaluación de la calidad del software usando la ISO/IEC 25000.
- Realizar la evaluación del costo del sistema web utilizando la técnica de estimación de costo Cocomo II.

1.5. JUSTIFICACIÓN

1.5.1. Justificación técnica

Se justifica técnicamente, la dirección de Posgrado cuenta con los equipos necesarios como: servidor apache, conexión a internet, sistema operativo Linux y otros, tanto en software y hardware. Para la implementación del sistema web.

1.5.2. Justificación económica

El proyecto de justifica económicamente porque permitirá reducir el tiempo de la organización y publicación de las revistas científicas, y se utilizará herramientas de desarrollos libres como software en base de datos PostgreSQL, lenguaje de programación PHP, JavaScript, framework codeigniter y framework Vue, para su diseño y programación.

1.5.3. Justificación social

El proyecto se justifica socialmente porque se beneficiara al personal de Posgrado, en especial el administrador del sistema, donde tendrá acceso a toda la información, gestión y administración de los usuarios encargados de las revistas.

Beneficiará a la comunidad universitaria, público en general y profesionales dentro del país, con la creación de una cuenta gratis en el sistema web público para ver el contenido de las revistas.

1.6. METODOLOGÍA

1.6.1. Metodología de desarrollo

1.6.1.1. UWE (UML-Based Web Engineering)

UWE es una metodología de ingeniería del software para el desarrollo de aplicaciones web basado en UML prestando especial atención en la sistematización

y personalización. Los modelos más usados de esta metodología son el modelo lógico-conceptual, navegacional, presentación, visualización de escenarios web, interacción temporal y diagrama de estado, secuencia, colaboración, actividad, entre otros. (Narvaez, 2016)

Las fases que contempla la metodología son:

- Análisis de Requerimientos
- Diseño del Sistema.
- Codificación del Software
- Pruebas
- La Instalación o Fase de Implementación
- El Mantenimiento

1.6.2. Métricas de calidad del software

La ingeniería del software se diferencia de otras áreas, al no estar basada en leyes cuantitativas básicas, en su lugar se realiza un conjunto de medidas conocidas como métricas, las cuales proporcionan una referencia de la calidad algún producto de software. (Pressman, 2010)

1.6.2.1. ISO/IEC 25000

ISO/IEC 25000, conocida como SQuaRE (System and Software Quality Requirements and Evaluation), es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software.

La familia ISO/IEC 25000 es el resultado de la evolución de otras normas anteriores, especialmente de las normas ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto software, e ISO/IEC 14598, que abordaba el proceso de evaluación de productos software. Esta familia de normas ISO/IEC 25000 se encuentra compuesta por cinco divisiones. (ISO25000, 2022)

1.6.3. Estimación de costos

1.6.3.1. Cocomo II

Las siglas COCOMO significan Constructive Cost Model, o “Modelo constructivo de costos” en español (COCOMO II). El método COCOMO fue desarrollado originalmente por el Dr. Barry Boehm en 1981. Luego en los años 90 se realizó una actualización y a partir de 1995 se conoce como COCOMO II.

El Modelo Constructivo de Costes es un modelo matemático de base empírica, utilizando para la estimación de costes de software. Incluye tres sub modelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado. (Pressman, 2010)

1.6.4. Seguridad del software

1.6.4.1. ISO 27000

La ISO 27001 es una norma internacional emitida por la Organización Internacional de Normalización (ISO) y describe cómo gestionar la seguridad de la información en una empresa. La revisión más reciente de esta norma fue publicada en 2013 y ahora su nombre completo es ISO/IEC 27001:2013. La primera revisión se publicó en 2005 y fue desarrollada en base a la norma británica BS 7799-2. (Alvarez, 2017)

1.6.5. Pruebas de software

1.6.5.1. Pruebas de caja negra

La técnica de pruebas de caja negra, consiste en ver el programa que queremos probar como una caja negra despreocupándonos del comportamiento interno y concentrando el esfuerzo en encontrar el comportamiento incorrecto, de acuerdo a las especificaciones de dicho programa, teniendo sólo en cuenta las entradas y salidas de dicho programa. (Sánchez, 2015)

1.6.5.2. Pruebas de caja blanca

La técnica de pruebas de caja blanca, al contrario de las pruebas de caja negra, consiste en verificar la estructura interna de un programa. (Sánchez, 2015)

1.6.5.3. Pruebas de rendimiento

En estas pruebas se medirán la velocidad de procesamiento y el tiempo de respuesta del sistema. (Sánchez, 2015)

1.6.6. Métodos de recolección de datos

1.6.6.1. La entrevista

La entrevista es el instrumento más importante de la investigación, junto con la construcción del cuestionario. En una entrevista además de obtener los resultados subjetivos del encuestado acerca de las preguntas del cuestionario, se puede observar la realidad circundante, anotando el encuestador además de las respuestas tal cual salen.

1.6.6.2. Cuestionarios y encuestas

Un cuestionario es un conjunto impreso de preguntas, abiertas o cerradas, que los encuestados deben responder en función de sus conocimientos y experiencia con el tema.

1.6.6.3. La observación

La observación es una técnica que consiste precisamente en observar el desarrollo del fenómeno que se desea analizar. Este método puede usarse para obtener información cualitativa o cuantitativa de acuerdo con el modo en que se realiza. (Torres, 1996)

1.7. HERRAMIENTAS

Las herramientas que se utilizan en las etapas de análisis, diseño y desarrollo del sistema se detallan a continuación:

1.7.1. PHP Hypertext Pre-Processor

PHP (acrónimo recursivo de PHP: Hypertext Pre-processor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web, para crear contenido dinámico y para trabajar con bases de datos y HTML.

Soporta la mayoría de bases de datos, MySQL, PostgreSQL, SQL Server, MongoDB... para casi todas existen drivers, y si no es así podemos utilizar el driver ODBC, que se conecta a cualquier base de datos. (Group, 2021)

1.7.2. Codeigniter

Codeigniter es un entorno de desarrollo de aplicaciones para gente que construye sitios web usando PHP. El objetivo es habilitar el desarrollo de proyectos de forma mucho más rápida de lo que podría si escribiese código desde cero, a través de proveer un rico conjunto de librerías para tareas comúnmente necesarias, tanto como una simple interface y estructura lógica para acceder a estas librerías. Codeigniter le permite concentrarse creativamente en su proyecto, minimizando el volumen de código necesario para una tarea determinada. (Andalucía, 2021)

1.7.3. JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. (Pérez, 2019)

1.7.4. Node js

Node.js es un entorno de ejecución para JavaScript construido con V8, motor de JavaScript de Chrome, además de ser multiplataforma y utilizado en la capa del servidor.

Node.js es un entorno controlado por eventos diseñado para crear aplicaciones escalables, permitiéndote establecer y gestionar múltiples conexiones al mismo tiempo. Gracias a esta característica, no tienes que preocuparte con el bloqueo de procesos, pues no hay bloqueos. (NodeJs, 2022)

1.7.5. Vue js

Creado por Evan You, Vue.js es un marco JavaScript progresivo de código abierto para crear interfaces de usuario (UI) y aplicaciones de una sola página; se conoce comúnmente como Vue. Este marco utiliza "alto desacoplamiento", lo que permite a los desarrolladores crear progresivamente interfaces de usuario (UI).

Vue.js tiene una línea de comando especial (CLI) creada en Node JS. Esta herramienta permite iniciar un proyecto utilizando un modelo estándar (o plantilla base). Además, el equipo de desarrollo de Vue.js mantiene una extensión de Chrome que permite ver cómo se representa el árbol de componentes, cómo se inician y registran los eventos, cómo se guarda el estado interno de cada componente y cómo se muestra el estado global del componente. (Mamani, 2019)

1.7.6. Bootstrap

Bootstrap es el popular HTML, CSS y JavaScript Marco para el desarrollo de un responsivo y móvil sitio web amigable Bootstrap es el HTML más popular, Framework CSS y JavaScript para desarrollar un sitio web receptivo y compatible con dispositivos móviles. Es absolutamente para descargar y usar gratis. Es un marco front-end utilizado para un desarrollo web más fácil y rápido. Incluye HTML y plantillas de diseño basadas en CSS para tipografía, formularios, botones, tablas, navegación, modales, carruseles de imágenes y muchos otros. También puede usar complementos de JavaScript. Te facilita crear diseños responsivos. (Gaikwad, 2019)

1.7.7. XAMPP

Es el entorno más popular de desarrollo con PHP. XAMPP es una distribución de Apache gratuita y que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser fácil de instalar y usar. (Friends, 2022)

1.7.8. PostgreSQL

PostgreSQL es un avanzado sistema de bases de datos relacionales basado en Open Source. Esto quiere decir que el código fuente del programa está disponible a cualquier persona libre de cargos directos, permitiendo a cualquiera colaborar con el desarrollo del proyecto o modificar el sistema para ajustarlo a sus necesidades.

PostgreSQL está bajo licencia BSD (distribución de software Berkeley). Un sistema de base de datos relacionales es un sistema que permite la manipulación de acuerdo con las reglas del álgebra relacional. Los datos se almacenan en tablas de columnas y renglones. Con el uso de llaves, esas tablas se pueden relacionar unas con otras. (Denzer, 2002)

1.7.9. Apache

Es un servidor web HTTP de código abierto para plataformas Unixlike (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. (Gdr, 2021)

1.7.10. MagicDraw

MagicDraw, herramienta CASE desarrollada por No Magic. Es compatible con el estándar UML 2.3, desarrollo de código para diversos lenguajes de programación (Java, C++ y C#, entre otros) así como para modelar datos. Cuenta con capacidad para trabajar en equipo y es compatible con varios entornos de desarrollo (IDEs). (EduRed, 2022)

1.8. LÍMITES Y ALCANCES

1.8.1. Límites

El sistema de información web para la administración y publicación digital de revistas científicas estará enfocada a la publicación del material digital para que sea consultado por los usuarios externos de internet y la administración del material impreso en el entorno local para la adquisición y venta de ejemplares físicos por el encargado de la unidad.

El sistema web no tendrá integrado lo siguiente:

- No contemplará el diseño de las portadas de las revistas.
- No contemplará el diseño y manejo de los empleados, pagos.

1.8.2. Alcances

El Sistema de información web para la administración y publicación digital de revistas científicas dispondrá de un contenido de la siguiente manera:

- Centralizar la información en un módulo de registro de las revistas para su fácil acceso y publicación.
- Centralizar la información de las compras de las revistas por los proveedores y venta de las revistas físicas de forma detallada además categorizarlas.
- Generar reportes importantes sobre las revistas.
- Facilitar al público información de las revistas gratuitas y puedan visualizar el contenido creando una cuenta gratis.

Web de administración:

- **Módulo inicio de sesión:** Los usuarios ingresar con el rol que tengan, ejemplo administrador, editor, vendedor.
- **Módulo tablero de información:** Contiene información de la cantidad de usuarios, revistas publicadas, etc.
- **Módulo de usuarios:** Se administra a los nuevos usuarios y asignar su rol como usuario.
- **Módulo de roles y permisos:** Se asigna los permisos de contenido, creación, eliminación, lista, edición, reportes, a los usuarios con el rol de editor, vendedor.
- **Módulo de categoría:** Se administra las distintas categorías de las revistas,
- **Módulo de autores:** Se administra a los autores de las revistas.
- **Módulo de revistas y publicación:** Se administra el registro, actualización, eliminación de las revistas y su publicación.
- **Módulo de proveedor:** Se administra a las empresas que provee las revistas físicas para posgrado.
- **Módulo de compra:** Se administra las compras de las revistas de los proveedores y los detalles, cantidad, precio, y otros.
- **Módulo de clientes:** Se administra a los clientes que deseen adquirir una copia física de una revista.
- **Módulo de ventas:** Se administra las ventas de las revistas y sus detalles.
- **Módulo de ventas en línea:** Se administra las ventas que se realizaron a los usuarios que adquieran una copia digital.

- **Módulo de reportes importantes:** Son los distintos informes que se realicen de los distintos módulos.

Web público:

- **Módulo inicio:** Es el inicio principal de la página con todas las revistas.
- **Módulo revistas:** Se visualizan las revistas publicadas.
- **Módulo detalle de las revistas:** Se visualizan los detalles de cada revista como, título, fecha de publicación, autor, etc.
- **Módulo inicio de sesión:** Los usuarios ingresar con el rol lector.
- **Módulo registro de usuario:** Se registran los usuarios interesados en ver revista y guardar en favoritos para ver luego.
- **Módulo perfil de usuario:** Contiene la información personal de los usuarios con una cuenta gratis.
- **Módulo cambiar contraseña:** Los usuarios con una cuenta pueden modificar su contraseña.
- **Módulo favorito del usuario:** Almacena las revistas de interés para ver luego.
- **Módulo compras del usuario:** Se almacena las compras que realizo los usuarios que adquirieron un copia de una revista de pago.
- **Módulo carrito de compras del usuario:** Se almacena las revistas que desea adquirir el usuario.

1.9. APORTES

Con la implementación del Sistema de información web para la administración y publicación digital de revistas científicas como herramienta de apoyo dentro de la dirección de posgrado para los administradores y que estará al alcance del público las revistas publicadas a si para la comunidad universitaria, docentes, que requieran de información de un material educativo, científico, etc., y además de estar disponible en todo momento.



CAPÍTULO II

MARCO

TEORICO

2.1. INTRODUCCIÓN

En esta sección del contenido se mencionará los conceptos que serán parte del proyecto, tanto como las herramientas, metodologías que serán implementadas para el diseño y desarrollo del sistema.

2.2. SISTEMA

Se entiende por un sistema a un conjunto ordenado de componentes relacionados entre sí, ya se trate de elementos materiales o conceptuales, dotado de una estructura, una composición y un entorno particulares. Se trata de un término que aplica a diversas áreas del saber, como la física, la biología y la informática o computación.

En informática, se entiende por un sistema a un conjunto de datos ordenados conforme a una serie de instrucciones o algoritmos, que permiten su ubicación y recuperación rápida y simple. (Etecé, 2021)

Por sistema se entiende un conjunto organizado de elementos que interactúan entre sí y con elementos externos. Pueden ser de tipo material (concretos) o conceptual (teóricos). Las entidades que constituyen un sistema obtienen siempre un resultado mayor en su conjunto al que tendrían cada una por separado.

Según ciertas teorías al respecto, todo objeto existente forma parte de algún sistema o está conformado por alguno, desde los átomos en una sustancia hasta las instituciones sociales en una democracia.

El estudio de los sistemas se rige por la Teoría General de Sistemas, producto de las labores del biólogo alemán Ludwig von Bertalanffy entre 1950 y 1968. (Uriarte, Características, 2020)

Los sistemas son parte de la relación entre uno a varios elementos que forman como un todo.

2.2.1. Sistemas informáticos

Por sistema informático (SI) se entiende un sistema automatizado de almacenamiento, procesamiento y recuperación de datos, que aprovecha las herramientas de la computación y la electrónica para llevar a cabo su serie compleja de procesos y operaciones. En otras palabras, un sistema informático es un computador de alguna índole. (Uriarte, 2020)

2.2.1.1. Estructura de los sistemas informáticos

Todo sistema informático se compone de tres instancias integradas, también denominadas “recursos”:

- **Hardware.** La parte “dura” del sistema, que abarca sus componentes físicos, materiales, como las placas, los circuitos, conectores, pantallas, teclas, tornillos, etc. Sin ella el sistema no podría simplemente encender.
- **Software.** La parte “blanda” del sistema, se refiere a la información y los sistemas programados, digitales, virtuales, que son intangibles pero esenciales para que el sistema tenga un propósito más allá de encender.
- **Humanware.** Este nombre se emplea a menudo para referirse al factor humano en la ecuación: los usuarios de diversa índole que intervienen en una red informática, que usan un computador o que lo programan.

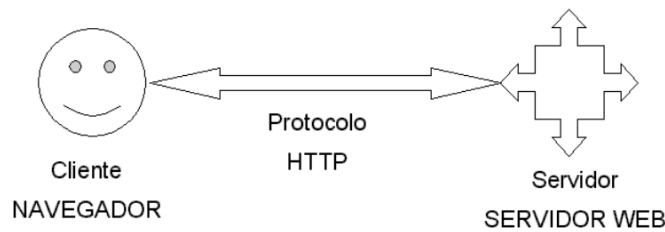
2.3. APLICACIÓN WEB

Una aplicación web (web based application) es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (HTTP) están estandarizados y no han de ser creados por el programador de aplicaciones (Figura 2.1).

El protocolo HTTP forma parte de la familia de protocolos de comunicaciones TCP/IP, que son los empleados en Internet. Estos protocolos permiten la conexión de sistemas heterogéneos, lo que facilita el intercambio de información entre distintos ordenadores. HTTP se sitúa en el nivel 7 (aplicación) del modelo OSI. (Mora, 2002)

Figura 2.1

Esquema básico de una aplicación web



Nota: La figura muestra la arquitectura de cliente servidor. En base a (Mora, 2002)

2.3.1. El cliente

El cliente web es un programa con el que interacciona el usuario para solicitar a un servidor web el envío de los recursos que desea obtener mediante HTTP. La parte cliente de las aplicaciones web suele estar formada por el código HTML que forma la página web más algo de código ejecutable realizado en lenguaje de script del navegador (JavaScript o VBScript) o mediante pequeños programas (applets) realizados en Java. Por tanto, la misión del cliente web es interpretar las páginas HTML y los diferentes recursos que contienen (imágenes, sonidos, etc.). Las tecnologías que se suelen emplear para programar el cliente web son:

- HTML.
- CSS.
- Lenguajes de script: JavaScript, VBScript, etc.
- ActiveX. Applets programados en Java

2.3.2. El servidor

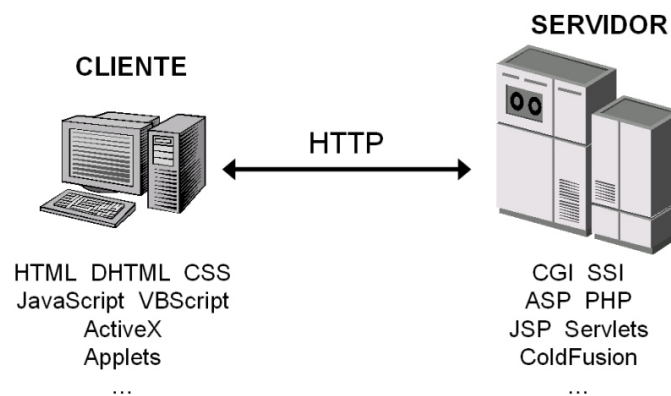
El servidor web es un programa que está esperando permanentemente las solicitudes de conexión mediante el protocolo HTTP por parte de los clientes web. En los sistemas Unix suele ser un “demonio” y en los sistemas Microsoft Windows un servicio. La parte servidor de las aplicaciones web está formada por: Páginas estáticas (documentos HTML) que siempre muestran el mismo contenido. Recursos adicionales (multimedia, documentos adicionales, etc.) que se pueden emplear dentro de las páginas o estar disponibles para ser descargados y ejecutados

(visualizados) en el cliente. Programas o scripts que son ejecutados por el servidor web cuando el navegador del cliente solicita algunas páginas. La salida de este script suele ser una página HTML estándar que se envía al navegador del cliente. Tradicionalmente este programa o script que es ejecutado por el servidor web se basa en la tecnología CGI. En algunos casos pueden acceder a bases de datos.

En la Figura 2.2 se han resumido las tecnologías que se emplean en la actualidad para programar el cliente y el servidor de las aplicaciones web.

Figura 2.2

Tecnologías empleadas en el cliente y en el servidor web



Nota: La figura muestra la arquitectura de cliente servidor y algunas herramientas para su programación de una aplicación web. En base a (Mora, 2002)

2.3.3. Importancia del modelo cliente servidor

La arquitectura cliente servidor tiene dos partes claramente diferenciadas, por un lado la parte del servidor y por otro la parte de cliente o grupo de clientes donde lo habitual es que un servidor sea una máquina bastante potente con un hardware y software específico que actúa de depósito de datos y funciona como un sistema gestor de base de datos o aplicaciones.

El más claro ejemplo de uso de una arquitectura cliente servidor es la red de Internet donde existen ordenadores de diferentes personas conectados alrededor del mundo, las cuales se conectan a través de los servidores de su proveedor de Internet por ISP donde son redirigidos a los servidores de las páginas que desean visualizar

y de esta manera la información de los servicios requeridos viajan a través de Internet dando respuesta a la solicitud demandada.

La principal importancia de este modelo es que permite conectar a varios clientes a los servicios que provee un servidor y como sabemos hoy en día, la mayoría de las aplicaciones y servicios tienen como gran necesidad que puedan ser consumidos por varios usuarios de forma simultánea. (Schiaffarino, 2019)

2.3.4. Componentes

Para entender este modelo vamos a nombrar y definir a continuación algunos conceptos básicos que lo conforman.

- **Red:** Una red es un conjunto de clientes, servidores y base de datos unidos de una manera física o no física en el que existen protocolos de transmisión de información establecidos.
- **Cliente:** El concepto de cliente hace referencia a un demandante de servicios, este cliente puede ser un ordenador como también una aplicación de informática, la cual requiere información proveniente de la red para funcionar.
- **Servidor:** Un servidor hace referencia a un proveedor de servicios, este servidor a su vez puede ser un ordenador o una aplicación informática la cual envía información a los demás agentes de la red.
- **Protocolo:** Un protocolo es un conjunto de normas o reglas y pasos establecidos de manera clara y concreta sobre el flujo de información en una red estructurada.
- **Servicios:** Un servicio es un conjunto de información que busca responder las necesidades de un cliente, donde esta información pueden ser mail, música, mensajes simples entre software, videos, etc.
- **Base de datos:** Son bancos de información ordenada, categorizada y clasificada que forman parte de la red, que son sitios de almacenaje para la utilización de los servidores y también directamente de los clientes.

2.3.5. Diferencia entre cliente y servidor

Como hemos mencionado anteriormente una máquina cliente como servidor se refieren a computadoras que son usadas para diferentes propósitos.

El cliente es un computador pequeño con una estructura al igual a la que tenemos en nuestras oficinas u hogares la cual accede a un servidor o a los servicios del mismo a través de Internet o una red interna. Un claro ejemplo a este caso es la forma en que trabaja una empresa modelo con diferentes computadores donde cada uno de ellos se conectan a un servidor para poder obtener archivos de una base de datos o servicios ya sea correos electrónicos o aplicaciones.

El servidor al igual que el cliente, es una computadora pero con diferencia de que tiene una gran capacidad que le permite almacenar gran cantidad de diversos de archivos, o correr varias aplicaciones en simultaneo para así nosotros los clientes poder acceder los servicios.

En la actualidad existen varios tipos de servidores como hablamos anteriormente. Los mismos pueden contener y ejecutar aplicaciones, sitios web, almacenaje de archivos, diversas bases de datos, entre muchos más.

Es importante mencionar que un cliente también puede tener una función de servidor ya que el mismo puede almacenar datos en su disco duro para luego ser usados en vez de estar conectándose al servidor continuamente por una acción que quizás sea muy sencilla. (Mora, 2002)

2.3.6. Tipos de arquitecturas cliente servidor

Dentro de la arquitectura cliente servidor existen tres tipos en donde hablaremos brevemente de cómo funciona cada uno de ellos. (Schiaffarino, 2019)

- **Arquitectura de dos capas:** Esta se utiliza para describir los sistemas cliente servidor en donde el cliente solicita recursos y el servidor responde directamente a la solicitud con sus propios recursos. Eso significa que el servidor no requiere de una aplicación extra para proporcionar parte del servicio.

- **Arquitectura de tres capas:** En la arquitectura de tres capas existe un nivel intermediario, eso significa que la arquitectura generalmente está compartida por un cliente que como hablamos más arriba es el que solicita los recursos equipado con una interfaz de usuario o mediante un navegador web. La capa del medio es denominada software intermedio cuya tarea es proporcionar los recursos solicitados pero que requiere de otro servidor para hacerlo. La última capa es el servidor de datos que proporciona al servidor de aplicaciones los datos necesarios para poder procesar y generar el servicio que solicitó el cliente en un principio.
- **Arquitectura N capas:** En la arquitectura de tres capas, los servidores dos y tres realizaron una tarea específica por lo tanto un servidor web puede usar los servicios de otros servidores para poder proporcionar su propio servicio. Por consiguiente la arquitectura en tres niveles es potencialmente una arquitectura en N capas ya que así como está contemplado en tres niveles como el caso anterior puede estar compuesto por N servidores donde cada uno de ellos brindan su servicio específico.

2.3.7. Ventajas y desventajas

Este modelo cliente servidor tiene varias ventajas y desventajas las cuales son importantes mencionar y conocer a la hora de establecer si es lo que estamos necesitando o si se acomoda a lo que estamos buscando.

Ventajas

- Facilita la integración entre diferentes sistemas y comparte información permitiendo por ejemplo que las máquinas ya existentes puedan ser utilizadas mediante una interfaz más amigable para el usuario. De esta manera podemos integrar varias PCs con sistemas medianos y grandes sin necesidad de que todos tengan que utilizar el mismo sistema operativo.
- Al favorecer el uso de la interfaz de gráficas interactivas, los sistemas contruidos bajo este esquema tienen una mayor interacción con el usuario.

- La estructura modular facilita de más la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional favoreciendo así la estabilidad de las soluciones.
- El modelo cliente servidor permite además proporcionar a las diferentes áreas de una empresa generar un orden de trabajo en donde cada sector puede trabajar en su área pero accediendo al mismo servidor e información que los demás sin generar conflictos. Esto es de gran utilidad ya que si se pone como ejemplo una empresa con varios empleados al momento de trabajar es importante que todos puedan hacerlo en simultáneo.

Desventajas

- Requiere habilidad para que un servidor sea reparado. Por ejemplo si un problema ocurre en la red, se requiere de alguien con un amplio de esta para poder repararla en su totalidad para así dejar que la información y el correcto funcionamiento siga su flujo.
- Otro problema es la seguridad, el hecho que se comparte canales de información entre servidores y clientes requieren que estas pasen por procesos de validación, es decir protocolos de seguridad que pueden tener algún tipo de puerta abierta permitiendo que se generen daños físicos, amenazas o ataques de malware.
- Este modelo representa una limitación importante en cuanto a los costos económicos debido a que estos servidores son computadoras de alto nivel con un hardware y software específicos para poder dar un correcto funcionamiento a nuestras aplicaciones. Algo importante a destacar es que no solo es caro a la hora de solucionar problemas como mencionamos antes, sino que también tiene un costo elevado para reemplazar componentes que estén averiados.

2.3.8. Api Rest

Una API de REST, o API de RESTful, es una interfaz de programación de aplicaciones (API o API web) que se ajusta a los límites de la arquitectura REST y permite la interacción con los servicios web de RESTful. (Civantos, 2021)

2.3.8.1. Rest

Rest: Representational State Transfer, es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP. Fue definida en el 2000 por Roy Fielding, uno de los padres de la especificación HTTP y un referente en la arquitectura de redes.

La mayoría de las aplicaciones que se desarrollan para servicios profesionales disponen de una API REST para el intercambio de información entre el frontend y el backend. Lo que la hace tan potente es precisamente el aislamiento que proporciona entre la lógica del back-end y cualquier cliente consumidor de éste. Esto le permite ser usada por cualquier tipo de cliente: web, móvil, etc. Así, cualquier dispositivo/cliente que entienda de HTTP puede hacer uso de su propia API REST de manera muy simple. Esto ha hecho que en los últimos años este tipo de arquitectura haya ganado peso frente a otras más complejas como SOAP, para el intercambio y manipulación de datos.

2.3.8.2. Principales características de una Api Rest

Los objetos REST son manipulados a través de una URI (Uniform Resource Identifier)

Esta URI (endpoint) hace de identificador único de cada recurso del sistema REST, por lo que no puede ser compartida por más de un recurso. La estructura básica de una URI es la siguiente:

- **Ejemplo:**

```
{protocolo}://{hostname}:{puerto}/{ruta del recurso}?{parámetros de filtrado (opcional)}
```

El nombre de la URI no debe contener palabras que impliquen acciones, por lo que deben evitarse los verbos en su construcción. Además, las URI siguen una jerarquía lógica de capas que permite ordenar los recursos y englobar las distintas funcionalidades entre sí. Por ejemplo: <http://tech.tribalyte.eu/category/apps>

O bien agregando un cuerpo a la llamada REST en cualquier tipo de formato, siendo los más usados JSON y XML

2.3.8.3. Uso de la especificación HTTP

Para el desarrollo de una API REST es necesario un conocimiento profundo de la especificación HTTP, sobre todo en lo referente a métodos permitidos, códigos de estado y aceptación de tipos de contenido.

Los métodos son usados para manipular los diferentes recursos que conforman la API. Los principales métodos soportados por HTTP y por ello usados por una API REST son:

- **POST**: crear un recurso nuevo.
- **PUT**: modificar un recurso existente.
- **GET**: consultar información de un recurso.
- **DELETE**: eliminar un recurso determinado.
- **PATCH**: modificar solamente un atributo de un recurso.

Estos métodos junto con la URI, nos proporciona una interfaz uniforme que nos permite la transferencia de datos en el sistema REST aplicando operaciones concretas sobre un recurso determinado. Aunque la mayoría de las operaciones que componen una API REST podrían llevarse a cabo mediante métodos GET y POST, el abuso de ellos para operaciones que nada tienen que ver con el propósito con el que se concibieron, puede provocar un mal uso del protocolo alejado del estándar o la construcción de URIs con nomenclatura errónea mediante el uso de verbos.

Cuando se realiza una petición determinada, es de vital importancia conocer si dicha operación se ha llevado a cabo de manera satisfactoria o por el contrario se ha producido algún tipo de error. Para ello, HTTP dispone de un amplio número de códigos de error/éxito que cubren todas las posibles respuestas que el usuario puede recibir cuando trata de manipular un recurso mediante el uso de una API REST.

Las más comunes son:

- **200 OK**. Respuesta estándar para peticiones correctas.
- **201 Created**. La petición ha sido completada y ha resultado en la creación de un nuevo recurso.

- **202 Accepted.** La petición ha sido aceptada para procesamiento, pero este no ha sido completado.
- **400 Bad Request.** La solicitud contiene sintaxis errónea.
- **403 Forbidden.** La solicitud fue legal, pero el servidor rehúsa responder dado que el cliente no tiene los privilegios para hacerla.
- **404 Not Found.** Recurso no encontrado. Se utiliza cuando el servidor web no encuentra la página o recurso solicitado.
- **500 Internal Server Error.** Es un código comúnmente emitido por aplicaciones empotradas en servidores web, cuando se encuentran con situaciones de error ajenas a la naturaleza del servidor web.

2.3.8.4. Ventajas de una Api Rest

La principal ventaja del uso de una API REST reside en la independencia que proporciona frente a cualquier consumidor, sin importar el lenguaje o plataforma con el que se acceda a ella. Esto permite que una misma API REST sea consumida por infinidad de clientes sea cual sea la naturaleza de estos y que el cambio a cualquier otro tipo de consumidor no provoque impacto alguno en ella. Esta característica proporciona fiabilidad, escalabilidad y una fácil portabilidad a cualquier otra plataforma, ya que aísla por completo al cliente del servidor. Sólo se requiere que el intercambio de información de las respuestas se haga en un formato soportado, por lo general JSON o XML. Dicha separación entre el cliente y el servidor hace que se pueda migrar a otros servidores o bases de datos de manera transparente, siempre y cuando los datos se sigan enviando de manera correcta. Esto convierte a las APIs REST en una de las arquitecturas web más utilizadas por la flexibilidad que aportan a cualquier entorno de trabajo sea cual sea su naturaleza.

2.4. INFORMACIÓN

Según Idalberto Chiavenato, información "es un conjunto de datos con un significado, o sea, que reduce la incertidumbre o que aumenta el conocimiento de algo. En verdad, la información es un mensaje con significado en un determinado contexto, disponible para uso inmediato y que proporciona orientación a las acciones por el

hecho de reducir el margen de incertidumbre con respecto a nuestras decisiones" (Thompson, 2008)

La información (general) es cualquier dato obtenido por una persona, independientemente de su forma de presentación.

Si consideramos el concepto en informática, hay que señalar que en esta ciencia la palabra "información" tiene un carácter más bien abstracto. La informática es una doctrina que se ocupa de los datos, es decir, de las formas de almacenarlos, transferirlos, procesarlos y representarlos de forma conveniente para el usuario.

(C.E.S.U.M.A, 2022)

La información es parte esencial en la actualidad que puede ser obtenida por medios digitales en sitios webs o procesados como datos para su uso.

2.5. PUBLICACIÓN

El término publicación se utiliza para hacer referencia al acto mediante el cual se hace público o se da a conocer determinada información, acto, dato, etc. La publicación puede ser en diferentes tipos de soportes que han variado en disponibilidad y en importancia a lo largo de la historia, siendo los más comunes los escritos, impresos o, actualmente, los digitales.

Por otro lado, la publicación puede conocer varios tipos dependiendo del público, las temáticas tratadas, el material, el soporte, etc. Así, la publicación en editoriales de materiales escritos deviene en libros, revistas y otro tipo de material que es leído por una gran cantidad de gente. Las publicaciones científicas y académicas son, por el contrario, más exclusivas y se suelen reservar para quienes se desempeñan en esos ámbitos ya que utilizan información, terminología y conceptos muy específicos y complejos. (Bembibre, 2013)

2.5.1. Publicación digital

Tipo de publicación concebida para ser almacenada o distribuida mediante sistemas informáticos y por ámbitos digitales, que pueden ser consultadas y preservadas en múltiples soportes. (ACyT, 2010)

Se llama publicación al acto y la consecuencia de publicar: difundir algo o darlo a conocer a través de la impresión u otro mecanismo. El término suele emplearse para aludir al texto impreso. (Gardey, 2019)

Publicación digital es toda información generada a difundir en sitios webs.

2.5.2. Revistas digitales

Las Revistas Digitales, a diferencia de la versión impresa, aprovechan la tecnología de la Publicación Digital que te permite agregar animaciones, multimedia y enlaces dentro de la revista para que sea más interactiva y mejor estéticamente.

Las revistas se pueden distribuir de forma gratuita como plan de Marketing o monetizarla por medio de suscripciones y espacios publicitarios.

Pueden ser una réplica exacta de la versión impresa, una réplica de la versión impresa más interactividad y contenido extra, o también pueden ser diseñadas directamente para su distribución digital. (Vázquez, 2022)

2.5.2.1. Ventajas

- **Descarga instantánea:** Ya sea que se compre un solo ejemplar o una suscripción anual, la revista digital se descargará instantáneamente mientras se tenga conexión a internet.

Para el caso de las suscripciones, el último ejemplar está disponible tan pronto como sea publicado por la editorial, sin tener que esperar a trasladarse a un kiosco de revistas.

- **Lee sin conexión a internet:** Una vez que se haya descargado el ejemplar, se puede leer la Revista Digital, sin conexión a internet, ideal para cuando se viaja en avión o se está de vacaciones.
- **Fácil ser verde:** Las Revistas Digitales son, por su naturaleza propia, ecológicas. Ahorrar en la huella de carbono que se genera de la impresión, el transporte, y si no se vende, el reciclaje; es la opción ideal para los guerreros ecológicos.

Con una suscripción a una Revista Digital, se puede leer sabiendo que no se está contribuyendo a la deforestación

2.5.3. Revistas científicas

Una revista científica es una “publicación periódica” en la que se intenta recoger el progreso de la ciencia, entre otras cosas incluyendo informes sobre las nuevas investigaciones. Muchas de las revistas son altamente especializadas, aunque algunas de las más antiguas como Nature y Science, por ejemplo publican artículos en una amplia gama de campos científicos, por lo que comúnmente reciben el nombre de revistas multidisciplinarias. (Aguado, 2022)

2.6. METODOLOGIA UWE (UML- Based Web Engineering)

2.6.1. Características de UWE

UWE es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web. UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML.

Es una herramienta que nos permitirá modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos). UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito. (Galiano, V-INF-3T, 2012)

2.6.2. Fases de la Metodología UWE

UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrandose además su atención en aplicaciones personalizadas o adaptativas. (Galiano, V-INF-3T, 2012)

Las fases o etapas a utilizar son:

- **Captura, análisis y especificación de requisitos:** En simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web.

Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales. Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipado de la interfaz de usuario.

- **Diseño del sistema:** Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web.
- **Codificación del software:** Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.
- **Pruebas:** Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.
- **La Instalación o Fase de Implementación:** es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final.
Esto incluye la implementación de la arquitectura, de la estructura del hiperespacio, del modelo de usuario, de la interfaz de usuario, de los mecanismos adaptativos y las tareas referentes a la integración de todas estas implementaciones.
- **El Mantenimiento:** es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control.

Figura 2.3

Fases de la Metodología UWE



Nota: La figura muestra la estructura de la metodología en cascada. (Galiano, V-INF-3T, 2012)

2.6.3. Modelos de la Metodología UWE

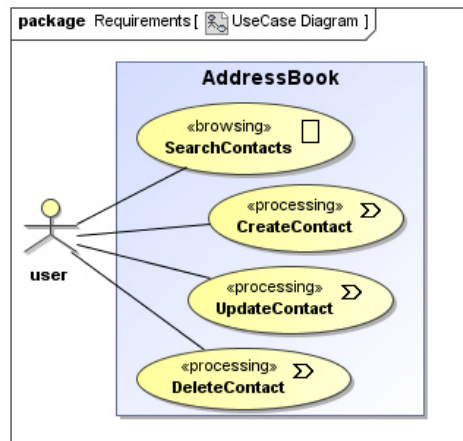
Las actividades de modelado de UWE son el análisis de requerimientos, el modelo conceptual, el modelo navegación y el modelo de presentación. A estos modelos se pueden sumar otros modelos como lo son el modelo de interacción y la visualización de escenarios Web. (Nora Koch, 2002)

2.6.3.1. Modelo de Caso de Uso

Un diagrama de casos de uso muestra la relación entre los actores y los casos del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

Figura 2.4

Diagrama de Casos de Uso - directorio



Nota: La figura muestra la interacción entre actores. (Maximilians, 2016)

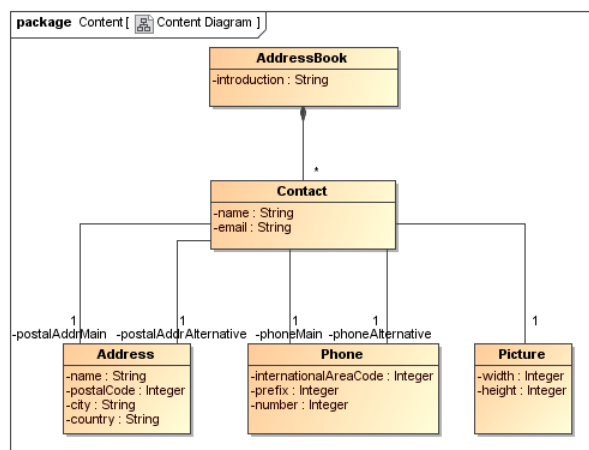
2.6.3.2. Modelo Conceptual

El modelo conceptual visualiza el dominio de información relevante para el sistema Web que incluye principalmente el contenido de la aplicación web.

El modelo conceptual es donde se presentan los modelos lógicos de cómo van a interactuar estas variables dentro del sistema que se está realizando.

Figura 2.5

Modelo conceptual - directorio



Nota: Basado en (Maximilians, 2016)

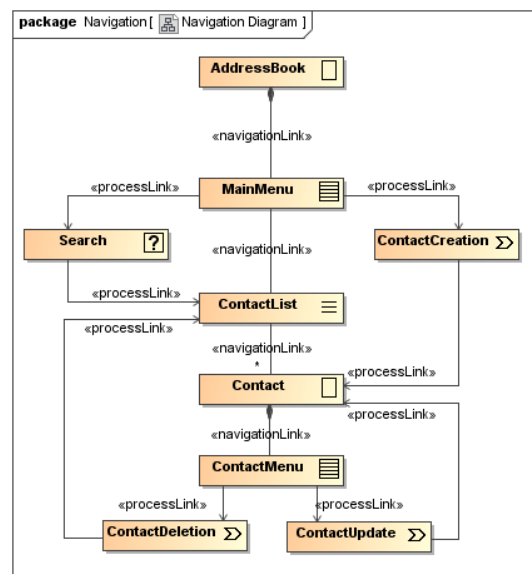
2.6.3.3. Modelo de Navegación

Describe dónde y cómo los objetos de navegación y accesos primitivos serán presentados al usuario, es decir, una presentación esquemática de los objetos visibles al usuario.

El Modelo de Navegación detalla cuáles son las clases de navegación y de proceso que pertenecen a una página Web.

Figura 2.6

Modelo de Navegación - directorio



Nota: Basado en (Maximilians, 2016)

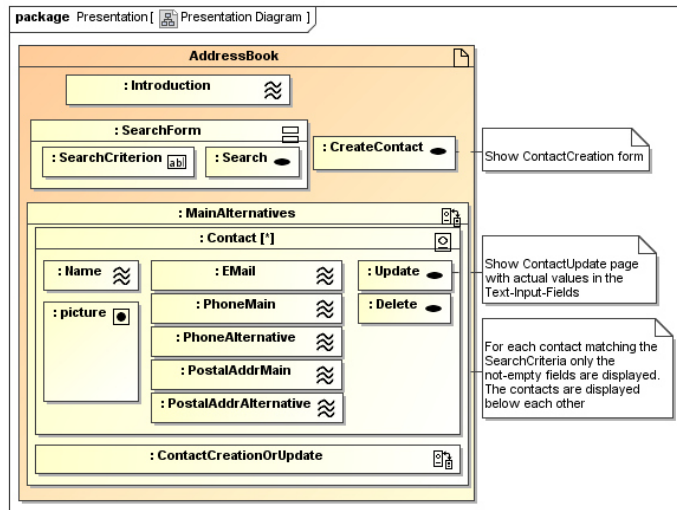
2.6.3.4. Modelo de Presentación

Describe dónde y cómo los objetos de navegación y accesos primitivos serán presentados al usuario, es decir, una presentación esquemática de los objetos visibles al usuario.

El Modelo de Navegación detalla cuáles son las clases de navegación y de proceso que pertenecen a una página Web.

Figura 2.7

Modelo Presentación - directorio



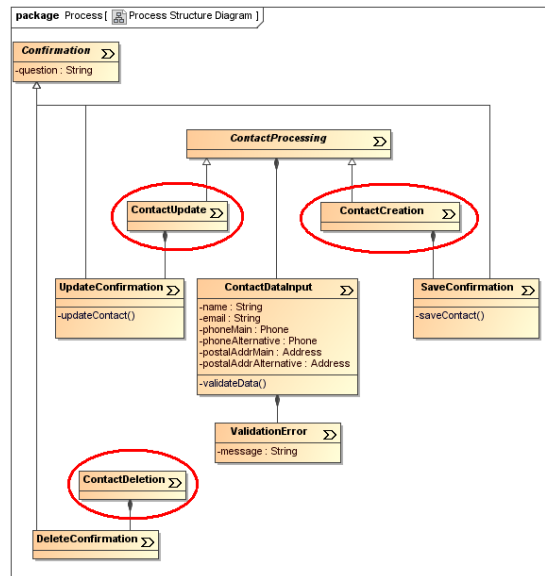
Nota: Basado en (Maximilians, 2016)

2.6.3.5. Modelo de Procesos

El modelo de tareas o procesos integra los procesos de negocios al modelo de UWE, especificando los comportamientos de cada proceso y de las interfaces que permiten manejar a cada uno de ellos. Este modelo representa la parte dinámica de la aplicación Web, especificando la funcionalidad de las transacciones y de los flujos de trabajo complejos de las actividades.

Figura 2.8

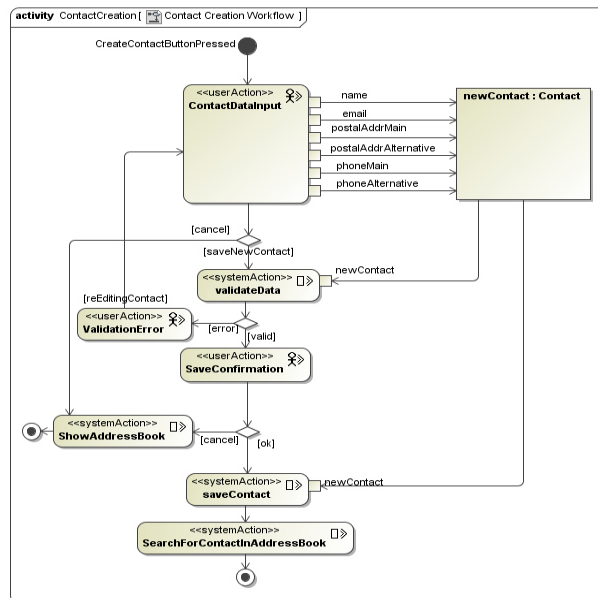
Modelo Proceso - procesamiento de contacto



Nota: Basado en (Maximilians, 2016)

Figura 2.9

Modelo de flujo del proceso – Crear Contacto



Nota: Basado en (Maximilians, 2016)

2.6.4. Ventajas y desventajas

Las principales razones para el uso de los mecanismos de extensión de UML en lugar de una técnica de modelado de propiedad es la aceptación del UML en el desarrollo de sistemas de software, la flexibilidad para la definición de un lenguaje de modelado específico de dominio Web: el llamado perfil UML, y amplio apoyo de modelado visual por herramientas CASE UML existentes.

UWE utiliza "puro" notación UML y tipos de diagramas UML siempre que sea posible para el análisis y diseño de aplicaciones Web, es decir, sin las extensiones de cualquier tipo. Por las características Web, como nodos y enlaces de la estructura de hipertexto, el perfil UWE incluye estereotipos, valores etiquetados y restricciones definidas para los elementos de modelado. La extensión UWE cubre la navegación, la presentación, los procesos de negocio y los aspectos de adaptación. (UTN, 2015)

2.7. METODOS DE PRUEBA DEL SOFTWARE

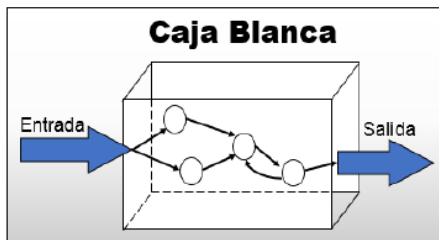
2.7.1. Métodos de prueba de caja blanca

La prueba de caja blanca, en ocasiones llamada prueba de caja de vidrio, es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba. Al usar los métodos de prueba de caja blanca, puede derivar casos de prueba que:

- Garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez.
- Revisen todas las decisiones lógicas en sus lados verdadero y falso.
- Ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas.
- Revisen estructuras de datos internas para garantizar su validez.

Figura 2.10

Métodos de prueba de caja blanca



Nota: (Pressman R. , 2010)

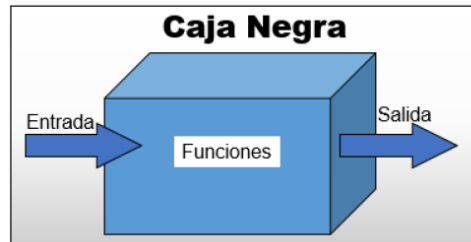
2.7.2. Método de prueba de caja negra

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software, las pruebas de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra no son una alternativa para las técnicas de caja blanca. En vez de ello, es un enfoque complementario que es probable que descubra una clase de errores diferente que los métodos de caja blanca. Las pruebas de caja negra intentan encontrar errores en las categorías siguientes.

- Funciones incorrectas o faltantes.
- Errores de interfaz.
- Errores en las estructuras de datos o en el acceso a bases de datos externa.
- Errores de comportamiento o rendimiento.
- Errores de inicialización y terminación.

Figura 2.11

Método de prueba de caja negra



Nota: (Pressman R. , 2010)

2.7.3. Prueba de rendimiento

La prueba de rendimiento es una técnica de prueba de software no funcional que determina cómo la estabilidad, la velocidad, la escalabilidad y la capacidad de respuesta de una aplicación se mantiene bajo una determinada carga de trabajo. Es un paso clave para asegurar la calidad del software, pero desafortunadamente, a menudo se ve como una reflexión posterior, en aislamiento, y para comenzar una vez que se completan las pruebas funcionales, y en la mayoría de los casos, después de que el código está listo para ser distribuido. (Micro Focus, 2022)

2.8. MÉTRICAS DE CALIDAD DEL SOFTWARE

2.8.1. Métrica de calidad

Las métricas de software se utilizan para propósitos estratégicos y son utilizadas en el proyecto para minimizar la planificación de desarrollo haciendo los ajustes necesarios que eviten retrasos y reduzcan problemas y riesgos potenciales, son utilizadas también para evaluar la calidad de los productos en el momento actual y cuando sea necesario, modificando el enfoque técnico que mejore la calidad. Para establecer objetivos de mejora durante el proceso de desarrollo de software, se debe comprender el estado actual del desarrollo del software. Si no se mide, no hay una forma real de determinar si se está mejorando y si no se está mejorando, se está perdido. (Pressman R. , 2010)

2.8.2. Medida de la calidad

La corrección, facilidad de mantenimiento, integridad, y facilidad de uso son medidas de calidad que proporcionan indicadores útiles para el equipo del proyecto.

- **Corrección:** La corrección es el grado en el que el software lleva a cabo su función requerida.
- **Facilidad de mantenimiento:** Es la facilidad con la que se puede corregir un programa si se encuentra un error, se puede adaptar si su entorno cambia, o mejorar si el cliente desea un cambio de requisitos. Esta actividad cuenta con más esfuerzo que cualquier otra actividad de ingeniería del software.
- **Integridad:** Mide la capacidad de un sistema para resistir ataques (tanto accidentales como intencionados) contra su seguridad. El ataque se puede realizar en cualquiera de los tres componentes del software: programas, datos y documentos.
- **Facilidad de uso:** La facilidad de uso es un intento de cuantificar lo amigable que puede ser el programa con el usuario. Se puede medir en función de cuatro características: Habilidad intelectual y/o física requerida para aprender el sistema. El tiempo requerido para llegar a ser moderadamente eficiente en el uso del sistema. Aumento neto en productividad, medida cuando alguien utiliza el sistema moderadamente y eficientemente. Valoración subjetiva de la disposición de los usuarios hacia el sistema, a veces obtenida mediante un cuestionario.

2.8.3. ISO/IEC 25000

ISO/IEC 25000, conocida como SQuaRE (System and Software Quality Requirements and Evaluation), es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software. (ISO25000, 2022)

La familia ISO/IEC 25000 es el resultado de la evolución de otras normas anteriores, especialmente de las normas ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto software, e ISO/IEC 14598, que abordaba el proceso

de evaluación de productos software. Esta familia de normas ISO/IEC 25000 se encuentra compuesta por cinco divisiones.

División de norma *Iso/iec 25000*

Figura 2.12

División de norma Iso/iec 25000



Nota: Basado en (ISO25000, 2022)

2.8.3.1. ISO/IEC 2500n – División de Gestión de Calidad

Las normas que forman este apartado definen todos los modelos, términos y definiciones comunes referenciados por todas las otras normas de la familia 25000. Actualmente esta división se encuentra formada por:

- ISO/IEC 25000 - *Guide to SQuaRE*: contiene el modelo de la arquitectura de SQuaRE, la terminología de la familia, un resumen de las partes, los usuarios previstos y las partes asociadas, así como los modelos de referencia.
- ISO/IEC 25001 - *Planning and Management*: establece los requisitos y orientaciones para gestionar la evaluación y especificación de los requisitos del producto software.

2.8.3.2. ISO/IEC 2501n – División de Modelo de Calidad

Las normas de este apartado presentan modelos de calidad detallados incluyendo características para calidad interna, externa y en uso del producto software. Actualmente esta división se encuentra formada por:

- ISO/IEC 25010 - *System and software quality models*: describe el modelo de calidad para el producto software y para la calidad en uso. Esta Norma presenta las características y subcaracterísticas de calidad frente a las cuales evaluar el producto software.
- ISO/IEC 25012 - *Data Quality model*: define un modelo general para la calidad de los datos, aplicable a aquellos datos que se encuentran almacenados de manera estructurada y forman parte de un Sistema de Información.

2.8.3.3. ISO/IEC 2502n – División de Medición de Calidad

Estas normas incluyen un modelo de referencia de la medición de la calidad del producto, definiciones de medidas de calidad (interna, externa y en uso) y guías prácticas para su aplicación. Actualmente esta división se encuentra formada por:

- ISO/IEC 25020 - *Measurement reference model and guide*: presenta una explicación introductoria y un modelo de referencia común a los elementos de medición de la calidad. También proporciona una guía para que los usuarios seleccionen o desarrollen y apliquen medidas propuestas por normas ISO.
- ISO/IEC 25021 - *Quality measure elements*: define y especifica un conjunto recomendado de métricas base y derivadas que puedan ser usadas a lo largo de todo el ciclo de vida del desarrollo software.
- ISO/IEC 25022 - *Measurement of quality in use*: define específicamente las métricas para realizar la medición de la calidad en uso del producto.
- ISO/IEC 25023 - *Measurement of system and software product quality*: define específicamente las métricas para realizar la medición de la calidad de productos y sistemas software.
- ISO/IEC 25024 - *Measurement of data quality*: define específicamente las métricas para realizar la medición de la calidad de datos.

2.8.3.4. ISO/IEC 2503n – División de Requisitos de Calidad

Las normas que forman este apartado ayudan a especificar requisitos de calidad que pueden ser utilizados en el proceso de e licitación de requisitos de calidad del

producto software a desarrollar o como entrada del proceso de evaluación. Para ello, este apartado se compone de:

- ISO/IEC 25030 - *Quality requirements*: provee de un conjunto de recomendaciones para realizar la especificación de los requisitos de calidad del producto software.

2.8.3.5. ISO/IEC 2504n – División de Evaluación de Calidad

Este apartado incluye normas que proporcionan requisitos, recomendaciones y guías para llevar a cabo el proceso de evaluación del producto software. Esta división se encuentra formada por:

- ISO/IEC 25040 - *Evaluation reference model and guide*: propone un modelo de referencia general para la evaluación, que considera las entradas al proceso de evaluación, las restricciones y los recursos necesarios para obtener las correspondientes salidas.
- ISO/IEC 25041 - *Evaluation guide for developers, acquirers and independent evaluators*: describe los requisitos y recomendaciones para la implementación práctica de la evaluación del producto software desde el punto de vista de los desarrolladores, de los adquirentes y de los evaluadores independientes.
- ISO/IEC 25042 - *Evaluation modules*: define lo que la Norma considera un módulo de evaluación y la documentación, estructura y contenido que se debe utilizar a la hora de definir uno de estos módulos.
- ISO/IEC 25045 - *Evaluation module for recoverability*: define un módulo para la evaluación de la subcaracterística Recuperabilidad (Recoverability).

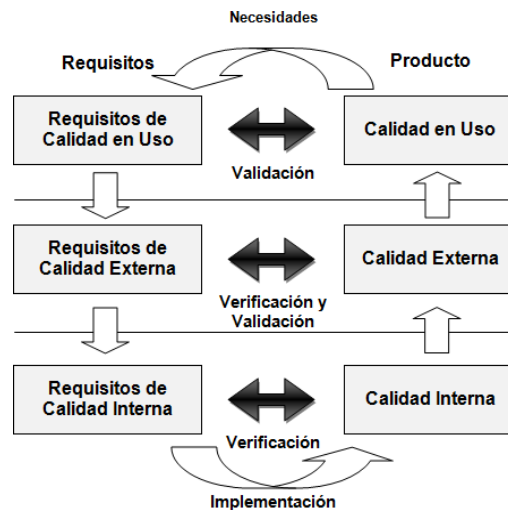
La división de extensión de SQuaRE (ISO/IEC 25050 a ISO/IEC 25099) se reserva para normas o informes técnicos que aborden dominios de aplicación específicos o que puedan ser utilizados para complementar otras normas de la familia SQuaRE.

2.8.4. Ciclo de vida de la calidad del Producto Software

El ciclo de vida especifica que la calidad de un producto software requiere un proceso similar al proceso de desarrollo para cada uno de los tipos de calidad como son: Especificación de Requisitos, Implementación y Validación de los resultados, tal como se muestra en la Figura 2.13.

Figura 2.13

Ciclo de Vida de la Calidad del Producto Software



Nota: Basado en (ISO25000, 2022)

El ciclo de vida en SQuaRE maneja la calidad del producto software en tres principales fases:

- **Calidad Interna:** cuando el producto software se encuentra en desarrollo.
- **Calidad Externa:** cuando el producto software se encuentra en funcionamiento.
- **Calidad en Uso:** cuando el producto software se encuentra en uso.

Para cada una de estas fases existen requisitos que responden a una necesidad del producto que deberán ser implementados y validados, a continuación se describe los tipos de requisitos de acuerdo a la Figura 2.13.

- Los requisitos de calidad en uso, especifican el nivel de calidad requerido desde el punto de vista del usuario. Estos requisitos son los que determinan la validación del software por parte del usuario. Como indica el modelo de ciclo de vida, la especificación de requisitos de calidad en uso ayuda a determinar los requisitos de calidad externa.
- Los requisitos de calidad externa se utilizan para la verificación y validación técnica del producto. Estos requisitos ayudan a determinar los requisitos de calidad interna pero además, pueden servir para predecir si se alcanzará la calidad en uso deseada.
- Los requisitos de calidad interna se utilizan para verificar el producto a lo largo de las distintas etapas del desarrollo y pueden utilizarse también para definir estrategias y criterios de evaluación y verificación.

2.8.5. ISO/IEC 25010

2.8.5.1. Modelo de Calidad del Producto Software (Calidad Interna y Externa)

El modelo define 8 características para la calidad interna y externa de un producto software: Adecuación Funcionalidad, Fiabilidad, Eficiencia en el Desempeño, Facilidad de Uso, Seguridad, Compatibilidad, Mantenibilidad y Portabilidad, las cuales a su vez son subdivididos en subcaracterísticas descritas en la Figura 15. Estas subcaracterísticas pueden ser medidas con métricas internas o externas.

Figura 2.14

Modelo de Calidad del Producto Software



Nota: Basado en (ISO25000, 2022)

A) Adecuación funcional

Representa la capacidad del producto o sistema software para proporcionar las funciones necesarias para satisfacer al usuario. Esta característica se divide en las siguientes subcaracterísticas:

- **Compleitud funcional:** capacidad del sistema software para proporcionar un conjunto de funcionalidades apropiadas para cubrir todas las tareas y objetivos determinados por el usuario.
- **Exactitud funcional:** capacidad del sistema software para proporcionar los resultados correctos con el grado necesario de precisión.

B) Fiabilidad

Capacidad del producto o sistema software para realizar las funciones específicas cuando se utiliza bajo ciertas condiciones y periodos de tiempo determinadas. Esta característica se divide en las siguientes subcaracterísticas:

- **Madurez:** capacidad del sistema software para satisfacer las necesidades de fiabilidad durante el funcionamiento normal.
- **Disponibilidad:** capacidad de un sistema software de estar operativo y accesible para su uso cuando se necesite.

- **Tolerancia a Fallos:** capacidad de un sistema software para operar cuando se presenten fallos.
- **Recuperabilidad:** capacidad de un sistema software para reestablecer el estado del sistema y recuperar datos que se hayan afectado, en caso de interrupción o fallo.

C) Eficiencia en el desempeño

Capacidad de un producto o sistema software de proporcionar un rendimiento apropiado, respecto a la cantidad recursos utilizados bajo determinadas condiciones. Esta característica se divide en las siguientes subcaracterísticas:

- **Comportamiento Temporal:** capacidad de un sistema software para proporcionar los tiempos de respuesta y procesamiento apropiados.
- **Utilización de Recursos:** capacidad en que un sistema software utiliza las cantidades y tipos de recursos adecuados.
- **Capacidad:** capacidad de un sistema software de cumplir con los requisitos determinados.

D) Facilidad de uso

Capacidad del producto o sistema software para que sea entendido, aprendido, agrado y usado por el usuario. Esta característica se divide en las siguientes subcaracterísticas:

- **Capacidad de reconocer su adecuación:** capacidad del sistema software que permite al usuario entender si el software es adecuado para sus necesidades.
- **Capacidad para ser entendido:** capacidad del sistema, que permite al usuario entender si el software es adecuado para alcanzar sus objetivos determinados.
- **Operatividad:** capacidad de un sistema software que permite al usuario operarlo y controlarlo con facilidad.
- **Protección contra errores del usuario:** capacidad en que el sistema brinda la protección necesaria contra errores que realizan los usuarios.

- **Estética de la Interfaz del usuario:** capacidad en que la interfaz de usuario llega a satisfacer y agradar al usuario.
- **Accesibilidad técnica:** capacidad del sistema software para que se permita ser utilizado por usuarios con determinadas discapacidades.

E) Seguridad

Capacidad de proteger la información y los datos, de manera que personas o sistemas no autorizados puedan tener acceso para consultas o actualizaciones. Esta característica se divide en las siguientes subcaracterísticas:

- **Confidencialidad:** capacidad de proteger la información y el acceso a datos no autorizados, ya sea de manera accidental o intencional.
- **Integridad:** capacidad de un producto, sistema o componente software para evitar accesos no autorizados a datos o programas de computación.
- **No – repudio:** capacidad para demostrar que los eventos han ocurrido, de manera que dichos eventos no puedan ser refutados posteriormente.
- **Responsabilidad:** capacidad de dar seguimiento a las acciones que fueron realizadas por una entidad.
- **Autenticidad:** capacidad de demostrar la identidad de un sujeto o un recurso.

F) Compatibilidad

Capacidad de dos o más sistemas software, para llevar a cabo sus funciones intercambiando información mientras comparten el mismo entorno. Esta característica se divide en las siguientes subcaracterísticas:

- **Co-Existencia:** capacidad de un sistema software para coexistir en un entorno en el cual comparten recursos comunes con otro software independiente.
- **Interoperatividad:** capacidad de dos o más sistemas software para intercambiar información y utilizar dicha información.

G) **Mantenibilidad**

Capacidad del sistema software para ser modificado o actualizado debido a necesidades evolutivas y correctivas. Esta característica se divide en las siguientes subcaracterísticas:

- **Modularidad:** capacidad de un sistema software que cuando sea modificado no afecte a otras funcionalidades del sistema
- **Reusabilidad:** capacidad de un activo (Información, Software, Hardware, Usuarios) para ser utilizado en más de un sistema o en la construcción de otros activos.
- **Capacidad de ser Analizado:** facilidad con la que se puede llevar a cabo un análisis del impacto de una determinada modificación en el sistema.
- **Capacidad de ser Modificado:** capacidad del sistema para permitir que sea modificado sin causar daños o reducir la calidad del producto existente.
- **Capacidad de ser Probado:** facilidad de realizar pruebas a un sistema o componente software, para determinar si se han cumplido con los requerimientos establecidos.

H) **Portabilidad**

Capacidad de un sistema o componente software de ser trasladado de un entorno a otro sin que esto afecte la funcionalidad de cada sistema. Esta característica se subdivide en las siguientes subcaracterísticas:

- **Adaptabilidad:** capacidad de un sistema software de ser adaptado a distintos entornos.
- **Capacidad de ser Instalado:** capacidad de un sistema para que pueda ser fácilmente instalado y/o desinstalado.
- **Capacidad de ser Reemplazado:** capacidad del sistema software para ser utilizado en lugar de otro sistema en el mismo entorno y cumpliendo con el mismo objetivo.

2.8.5.2. Modelo para la Calidad en Uso

El modelo de calidad en uso define 5 características: Efectividad, Eficiencia, Satisfacción, Libertad de Riesgo y Cobertura de Contexto, las cuales a su vez son subdivididos en subcaracterísticas descritas en la Figura 2.15. Estas subcaracterísticas pueden ser medidas con métricas de calidad en uso.

Figura 2.15

Modelo de Calidad para Calidad en Uso



Nota: Basado en (ISO25000, 2022)

El resultado de la calidad en uso depende necesariamente del logro de la calidad externa, que a su vez depende necesariamente del logro de la calidad interna.

A) Efectividad

Capacidad del sistema software para alcanzar los objetivos o necesidades del usuario, al momento de utilizar el sistema.

B) Eficiencia

Capacidad del sistema software para alcanzar los objetivos del usuario, utilizando los recursos mínimos.

C) Satisfacción

Capacidad del sistema software para satisfacer las diferentes necesidades mínimas de los usuarios al utilizarlo. Esta característica se divide en las siguientes subcaracterísticas:

- **Utilidad:** grado en que un usuario es satisfecho cuando logra alcanzar sus objetivos planteados.

D) Libertad de riesgo

Capacidad que tiene un producto o sistema software en reducir el riesgo potencial relacionado con la situación económica, vida humana, salud o medio ambiente.

Esto incluye la salud y seguridad, tanto del usuario y aquellos afectados por el uso, así como las consecuencias materiales o económicas no deseadas.

En este caso, el riesgo es la probabilidad de ocurrencia y las posibles consecuencias negativas cuando se presenta una amenaza determinada.

Esta característica se subdivide en las siguientes subcaracterísticas las que permiten establecer el grado en el cual los objetivos podrían estar en riesgo.

- Libertad del riesgo económico.
- Libertad del riesgo de salud y seguridad.
- Libertad del riesgo ambiental.

E) Cobertura de contexto

Capacidad de un producto o sistema software para ser utilizado con efectividad, eficiencia, libertad de riesgo y satisfacción en ámbitos de uso que fueron definidos.

Esta característica se subdivide en las siguientes subcaracterísticas:

- **Integridad de Contexto:** capacidad de un sistema software para ser utilizado en los ámbitos de uso definidos.
- **Flexibilidad:** capacidad de un sistema software para ser utilizado fuera de los ámbitos de uso que fueron definidos inicialmente.

2.8.5.3. Métricas para la calidad interna, externa y en uso - ISO/IEC 25022 y 25023

Las normas ISO/IEC 25023 e ISO/IEC 25022, proveen un conjunto de métricas de calidad, tanto para la calidad interna, externa y en uso, que son usadas con el modelo de calidad ISO/IEC 25010.

Tabla 2.1

Métricas de calidad interna/externa para Adecuación funcional

Métricas para la característica de calidad Adecuación funcional								
Subcaracterística	Métrica	Fase del ciclo de vida de calidad del producto	Propósito de la métrica de calidad	Método de aplicación	Fórmula	Valor deseado	Tipo de medida	Recursos utilizados
Compleitud funcional	Compleitud de la implementación funcional	Interna/Externa	¿Cuán completa es la implementación de acuerdo a la especificación de requerimientos?	Contar el número de las funciones indicadas en la especificación de requerimientos y el número de funciones que faltan o están incorrectas	$X = A / B$ A = Número de funciones que están incorrectas o que no fueron implementadas B = Número de las funciones establecidas en la especificación de requisitos Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0 es el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester
Exactitud funcional	Exactitud	Interna/Externa	¿Cuánto del estándar requerido de exactitud se cumple?	Contar el número de elementos de datos implementados con el estándar específico de exactitud y el número total de elementos de datos implementados	$X = A/B$ A = Número de elementos de datos implementados con el estándar específico de exactitud B = Número total de elementos de datos implementados Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester

	Precisión computacional	Interna/Externa	¿Con qué frecuencia ocurren los resultados inexactos?	Contar el número de cálculos inexactos encontrados y tomar el tiempo de operación	$X = A/T$ A = Numero de cálculos inexactos encontrados T = Tiempo de operación Dónde: $T > 0$	$X = A/T$ El más cercano a $0/t$ es el mejor. Donde el peor caso es $\geq 10/t$.	X= Contable/ Tiempo A= Contable B= Tiempo	Código fuente, Desarrollador, Tester
	Precisión computacional	Interna/Externa	¿Con qué frecuencia ocurren los resultados inexactos?	Contar el número de cálculos inexactos encontrados y tomar el tiempo de operación	$X = A/T$ A = Numero de de cálculos inexactos encontrados T = Tiempo de operación Dónde: $T > 0$	$X = A/T$ El más cercano a $0/t$ es el mejor. Donde el peor caso es $\geq 10/t$.	X= Contable/ Tiempo A= Contable B= Tiempo	Código fuente, Desarrollador, Tester

Nota: Basado en ISO/IEC 25023

Tabla 2.2

Métricas de calidad interna/externa para Fiabilidad

Métricas para la característica de calidad Fiabilidad								
Subcaracterística	Métrica	Fase del ciclo de vida de calidad del producto	Propósito de la métrica de calidad	Método de aplicación	Fórmula	Valor deseado	Tipo de medida	Recursos utilizados
Madurez	Eliminación de errores	Interna/Externa	¿Cuántos Errores detectados han sido corregidos?	Contar el número de fallas corregidas en la fase de diseño/codificación/pruebas y el número de fallas detectadas en las pruebas	$X = A/B$ A = Número de fallas corregidas en la fase de diseño/codificación/pruebas B = Número de fallas detectadas en las pruebas Dónde: $B > 0$	$0 \leq X \leq 1$ Cuanto más se acerque a 1 es lo mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos, Documento de casos de pruebas, Código fuente, Desarrollador, Tester
	Cobertura de pruebas	Interna/Externa	¿Cuántos casos de prueba requeridos han sido ejecutados durante la etapa de pruebas?	Contar el número de casos de pruebas realizados en un escenario de operación durante la prueba y el número de casos de prueba a ser realizados para cubrir los requerimientos	$X = A/B$ A = Número de casos de pruebas realizados en un escenario de operación durante la prueba B = Número de casos de prueba a ser realizados para cubrir los requerimientos Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos, Documento de casos de pruebas, Código fuente, Desarrollador, Tester

	Tiempo medio entre fallos	Externa	¿Cuál es la frecuencia en que el sistema falla en la operación?	Tomar el tiempo de operación y contar el número total de fallas detectadas actualmente	$X = A/T$ A = Número total de fallas detectadas actualmente T = Tiempo de operación Donde $T > 0$	$X = A/T$ El más cercano a 0/t es el mejor	X= Contable/ Tiempo A= Tiempo B= Contable	Especificación de requerimientos, Desarrollador, Tester
Disponibilidad	Tiempo de servicio	Externa	¿Cuál es el tiempo de servicio del sistema que proporciona realmente?	Tomar el tiempo de servicio del sistema que se proporciona actualmente y tomar el tiempo de servicio del sistema regulado en el cronograma operacional	$X = A/B$ A = Tiempo de servicio del sistema que se proporciona actualmente B = Tiempo de servicio del sistema regulado en el cronograma operacional Dónde: $B > 0$	$0 \leq X \leq 1$ Cuanto más se acerque a 1 es lo mejor	X= Tiempo/ Tiempo A= Tiempo B= Tiempo	Especificación de requerimientos, Desarrollador, Tester
	Tiempo medio de inactividad	Externa	¿Cuál es el tiempo promedio que el sistema está inactivo después de que ocurre un fallo?	Tomar el tiempo total de inactividad y contar el número de fallos observados	$X = A/T$ A = Número de fallos observados T = Tiempo total de inactividad Dónde: $T > 0$	$X = A/T$ El más cercano a 0/t es el mejor	X= Contable/ Tiempo A= Tiempo B= Contable	Especificación de requerimientos, Desarrollador, Tester

Tolerancia a fallos	Prevención de fallas	Externa	¿Cuántas fallas iniciales estuvieron bajo control para evitar fallas serias y críticas?	Contar el número de ocurrencia de fallas serias y críticas evitadas contra los casos de pruebas de fallas iniciales y el número de casos de pruebas de fallas iniciales ejecutados durante las pruebas	$X = A/B$ A = Número de ocurrencia de fallas evitadas contra los casos de pruebas de fallas iniciales B = Número de casos de pruebas de fallas iniciales ejecutados durante las pruebas Dónde: $B > 0$	$0 \leq X \leq 1$ Cuanto más se acerque a 1 es lo mejor	X= Contable / Contable A= Contable B= Contable	Especificación de requerimientos, Desarrollador, Tester
	Redundancia	Interna/Externa	¿Cuántos tipos de componentes/sistemas del son instalados de forma redundante para evitar un fallo en el sistema?	Contar el número total de tipos de componentes y el número de tipos de componentes instalados de forma redundante	$X = A / B$ A= Número componentes/sistemas instalados de forma redundante B = Número total de componentes/sistemas instalados Dónde: $B > 0$	$0 \leq X \leq 1$ Cuanto más se acerque a 1 es lo mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester
	Anulación de operación incorrecta	Interna	¿Cuántas funciones son implementadas con capacidad de anular operaciones incorrectas?	Contar el número de funciones implementadas que evitan fallas críticas y serias causadas por operaciones incorrectas y contar el número de operaciones incorrectas presentadas	$X = A/B$ A = Número de operaciones incorrectas presentadas B = Número total de funciones implementadas para anular operaciones incorrectas Dónde: $B > 0$	$0 \leq X \leq 1$ Cuanto más se acerque a 0 es lo mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester

Recuperabilidad	Tiempo medio de recuperación	Interna/Externa	¿Cuál es el tiempo promedio que toma el sistema en recuperarse completamente después un fallo?	Tomar el tiempo que le tomó al sistema en recuperarse y contar el número de casos en los cuales se ha observado que el sistema entró en recuperación	$X = A / T$ A = Número de casos en los cuales se ha observado que el sistema entró en recuperación T = Tiempo que le tomó al sistema en recuperarse Dónde: $T > 0$	$X = A/T$ El más cercano a 0/t es el mejor. Donde el peor caso es $\geq 10/t$.	X= Contable /Tiempo A= Contable B= Tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Tester
------------------------	------------------------------	-----------------	--	--	---	---	---	--

Nota: Basado en ISO/IEC 25023

Tabla 2.3

Métricas de calidad interna/externa para Eficiencia en el desempeño

Métricas para la característica de calidad Eficiencia en el desempeño								
Subcaracterística	Métrica	Fase del ciclo de vida de calidad del producto	Propósito de la métrica de calidad	Método de aplicación	Fórmula	Valor deseado	Tipo de medida	Recursos utilizados
Comportamiento del tiempo	Tiempo de respuesta	Interna/Externa	¿Cuál es el tiempo estimado para completar una tarea?	Tomar el tiempo desde que se envía la petición hasta obtener la respuesta	$X = B - A$ A= Tiempo de envío de petición B = Tiempo en recibir la primera respuesta	$0 \leq X \leq 1$ El más cercano a 0 es el mejor. Donde el peor caso es $\geq 15t$.	X=Tiempo/ Tiempo A=Tiempo B=Tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Tester

	Tiempo de espera	Interna/Externa	¿Cuál es el tiempo desde que se envía una instrucción, para que inicie un trabajo, hasta que lo completa?	Tomar el tiempo cuando se inicia un trabajo y el tiempo en completar el trabajo	$X = B - A$ A= Tiempo cuando se inicia un trabajo B = Tiempo en completar el trabajo	$0 \leq X \leq 1$ El más cercano a 0 es el mejor. Donde el peor caso es $\geq 15t$.	$X = \text{Tiempo} / \text{Tiempo}$ A=Tiempo B=Tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Tester
	Rendimiento	Interna/Externa	¿Cuántas tareas pueden ser procesadas por unidad de tiempo?	Contar el número de tareas completadas en un intervalo de tiempo	$X = A/T$ A= Número de tareas completadas T = Intervalo de tiempo Dónde: $T > 0$	$X = A/T$ El más lejano a 0/t es el mejor. Donde el mejor caso es $\geq 10/t$	$X = \text{Contable} / \text{Tiempo}$ A= Contable T= Tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Tester
	Líneas de código	Interna	¿Cuántas líneas de código existen por cada función implementada?	Contar el número de líneas de código (sin tomar en cuenta espacios ni comentarios) que existen en una determinada función	$X = A$ A = Número de líneas de código	$1 \leq X \leq 50$ El más cercano a 1 es el mejor. Donde el peor caso es ≥ 50 líneas de código	$X = \text{Contable} / \text{Contable}$ A= Contable	Código fuente
	Utilización de CPU	Interna/Externa	¿Cuánto tiempo de CPU es usado para realizar una tarea dada?	Tomar el tiempo de operación y la cantidad de tiempo de CPU que se usa para realizar una tarea	$X = B - A$ A= La cantidad de tiempo de CPU que realmente es usado para realizar una tarea B = Tiempo de operación Dónde: $B > 0$	$0 \leq X \leq 1$ Cuanto más se acerque a 0 es lo mejor. Donde el peor caso es $\geq 15t$.	$X = \text{Tiempo} / \text{Tiempo}$ A= Tiempo B= Tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Tester

Utilización de recursos	Utilización de la memoria	Interna/Externa	¿Cuánto espacio de memoria es usado para realizar una tarea dada?	Medir la cantidad total de espacios de memoria y la cantidad de espacios de memoria que realmente es usado para realizar una tarea	$X = B - A$ $A =$ Cantidad de espacios de memoria que realmente es usado para realizar una tarea $B =$ Cantidad total de espacios de memoria Dónde: $B > 0$	$0 \leq X \leq 15$ El más cercano a 0 es el mejor	$X =$ Tamaño/ Tamaño $A =$ Tamaño $B =$ Tamaño	Especificación de requerimientos, Código fuente, Desarrollador, Tester
	Utilización de los dispositivos de E/S	Interna/Externa	¿Cuánto tiempo los dispositivos de E/S utilizan para realizar una tarea?	Tomar el tiempo de operación y el tiempo que los dispositivos de E/S pasan ocupados para realizar la tarea	$X = B - A$ $A =$ Tiempo que los dispositivos de E/S pasan ocupados para realizar la tarea $B =$ Tiempo de operación Dónde: $B > 0$	$0 \leq X \leq 15$ El más cercano a 0 es el mejor	$X =$ Tiempo/ Tiempo $A =$ Tiempo $B =$ Tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Tester
Capacidad	Número de peticiones online	Interna/Externa	¿Cuántas peticiones online pueden ser procesadas por unidad de tiempo?	Contar el número máximo de peticiones online procesadas y tomar el tiempo de operación	$X = A / T$ $A =$ Número máximo de peticiones online Procesada $T =$ Tiempo de operación Dónde: $T > 0$	$X = A / T$ El más lejano a 0/t es el mejor. Donde el mejor caso es $\geq 10/t$.	$X =$ Contable/ Tiempo $A =$ Contable $T =$ Tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Tester
	Número de accesos simultáneos	Interna/Externa	¿Cuántos usuarios pueden acceder al sistema simultáneamente en un cierto tiempo?	Contar el número máximo de accesos simultáneos y tomar el tiempo de operación	$X = A / T$ $A =$ Número máximo de accesos simultáneos $T =$ Tiempo de operación Dónde: $T > 0$	$X = A / T$ El más lejano a 0/t es el mejor. Donde el mejor caso es $\geq 10/t$	$X =$ Contable/ Tiempo $A =$ Contable $B =$ Tiempo	Especificación de requerimientos, Código fuente, Desarrollador, Tester

	Sistema de transmisión de ancho de banda	Externa	¿Cuánto es el valor límite absoluto de transmisión necesaria para cumplir con las funciones?	Contar la cantidad máxima de transmisión de datos y tomar el tiempo de operación	$X = A/T$ A= Cantidad máxima de transmisión de datos B = Tiempo de operación Dónde: $T > 0$	$X = A/T$ El más lejano a 0/t es el mejor. Donde el mejor caso es $\geq 10/t$	$X = \text{Tamaño} / \text{Tiempo}$ A=Tamaño B=Tiempo	
--	--	---------	--	--	--	--	---	--

Nota: Basado en ISO/IEC 25023

Tabla 2.4

Métricas de calidad interna/externa para Facilidad de uso

Métricas para la característica de calidad Facilidad de uso								
Subcaracterística	Métrica	Fase del ciclo de vida de calidad del producto	Propósito de la métrica de calidad	Método de aplicación	Fórmula	Valor deseado	Tipo de medida	Recursos utilizados
Capacidad de reconocer su adecuación	Integridad de descripción	Interna/Externa	¿Qué cantidad de funciones (o tipos de funciones) son descritas como entendibles en la descripción del producto?	Contar el número de funciones (o tipos de funciones) descritas como entendibles en la descripción del producto y contar el número total de funciones (o tipos de funciones)	$X = A/B$ A = Número de funciones (o tipos de funciones) descritas como entendibles en la descripción del producto B = Número total de funciones (o tipos de funciones) Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	$X = \text{Contable} / \text{Contable}$ A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester

Capacidad para ser entendido	Capacidad de demostración	Interna/Externa	¿Qué cantidad de funciones tienen la capacidad de demostración?	Contar el número de funciones implementadas con capacidad de demostración y contar el número total de funciones que requieren capacidad de demostración	$X = A/B$ A = Número de funciones implementadas con capacidad de demostración B = Número total de funciones que requieren capacidad de demostración Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos , Código fuente, Desarrollador, Tester
	Funciones evidentes	Interna	¿Qué cantidad de funciones del producto son evidentes al usuario?	Contar el número de funciones que son evidentes al usuario y comparar con el número total de funciones.	$X = A / B$ A= Número de funciones (o tipo de funciones) evidentes al usuario B = Número total de funciones (o tipo de funciones) Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable/ Contable A= Contable B= Contable	Código fuente
	Efectividad de la documentación del usuario o ayuda del sistema	Interna/Externa	¿Qué cantidad de funciones están descritas correctamente en la documentación del usuario o ayuda en línea?	Contar el número de funciones descritas correctamente y contar el número total de funciones implementadas	$X = A / B$ A= Número de funciones descritas correctamente B = Número total de funciones implementadas Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos , Código fuente, Desarrollador, Tester
Operatividad	Recuperabilidad de error operacional	Interna	¿Qué cantidad de funciones pueden tolerar errores de usuario?	Contar el número de funciones implementadas con tolerancia de error de usuarios y el número total de funciones requeridas con capacidad de tolerancia	$X = A / B$ A= Número de funciones implementadas con tolerancia de error de usuarios B = Número total de funciones requeridas con capacidad de tolerancia. Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable/ Contable A= Contable B= Contable	Código fuente

	Claridad del mensaje	Interna/Externa	¿Qué cantidad de mensajes son auto explicativo?	Contar el número de mensajes implementados con explicaciones claras y el número total de mensajes implementados	$X = A / B$ A= Número de mensajes implementados con explicaciones claras B = Número total de mensajes implementados Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos , Código fuente, Desarrollador, Tester
	Consistencia operacional	Interna/Externa	¿Cuántas operaciones similares pueden llevarse a cabo consecuentemente?	Contar el número de operaciones que se comportan de manera incoherente y el número total de operaciones que se comportan de forma normal	$X = A / B$ A= Número de operaciones que se comportan de manera incoherente B = Número total de operaciones que se comportan de forma normal Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0 es el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos , Código fuente, Desarrollador, Tester
	Posibilidad de personalización	Interna/Externa	¿Cuántas funciones y procedimientos operacionales puede un usuario modificar para su conveniencia?	Contar el número de funciones implementadas que pueden ser personalizadas durante la operación y el número de funciones que requieran la capacidad de personalización	$X = A / B$ A = Número de funciones implementadas que pueden ser personalizadas durante la operación B = Número de funciones que requieran la capacidad de personalización Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos , Código fuente, Desarrollador, Tester
Protección contra errores del usuario	Verificación de entradas válidas.	Interna/Externa	¿Qué cantidad de ítems de entrada son validados?	Contar el número de ítems de entrada que son validados y el número de ítems que necesitan ser validados	$X = A/B$ A= Número de ítems de entrada que son validados B = Número de ítems que necesitan ser validados Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos , Código fuente, Desarrollador, Tester

	Prevención del uso incorrecto	Interna/Externa	¿Cuántas funciones tienen la capacidad de evitar operaciones incorrectas?	Contar el número de funciones implementadas para evitar fallos de funcionamiento provocados por un uso incorrecto y el número total de operaciones iniciales incorrectas	$X = A/B$ A = Número de operaciones iniciales incorrectas B = Número de funciones implementadas para evitar fallos de funcionamiento provocados por un uso incorrecto Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester
Estética de la Interfaz del usuario	Personalización de la apariencia de la interfaz del usuario	Interna/Externa	¿Qué cantidad de los elementos de la interfaz de usuario pueden ser personalizados en apariencia?	Contar el número de tipos de elementos de interfaz que pueden ser personalizados y contar el número total de tipos de elementos de interfaz	$X = A/B$ A= Número de elementos de interfaz que pueden ser personalizados B = Número total de elementos de interfaz Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester
Accesibilidad técnica	Accesibilidad física	Interna/Externa	¿A qué cantidad de funciones puede acceder un usuario con discapacidades físicas?	Contar el número de funciones a las que pueden acceder personas con discapacidad y contar el número total de funciones implementadas	$X = A/B$ A= Número de funciones a las que pueden acceder personas con discapacidad B = Número total de elementos de interfaz Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester

Nota: Basado en ISO/IEC 25023

Tabla 2.5

Métricas de calidad interna/externa para Seguridad

Métricas para la característica de calidad Seguridad								
Subcaracterística	Métrica	Fase del ciclo de vida de calidad del producto	Propósito de la métrica de calidad	Método de aplicación	Fórmula	Valor deseado	Tipo de medida	Recursos utilizados
Confidencialidad	Capacidad de control de acceso	Interna/Externa	¿Qué tan controlable son los accesos al sistema?	Contar el número de diferentes tipos de operaciones ilegales detectados y el número de tipos de operaciones ilegales en la especificación	$X = A / B$ A = Número de diferentes tipos de operaciones ilegales detectados B = Número de tipos de operaciones ilegales en la especificación Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester
	Encriptación de datos	Interna/Externa	¿Qué tan correctamente es la implementación de encriptación / desenscriptación de datos de acuerdo a la especificación de requerimientos?	Contar el número de elementos de datos encriptados/ desenscriptados correctamente y el número de elementos de datos que requiere el encriptación/ desenscriptación	$X = A / B$ A = Número de elementos de datos encriptados/ desenscriptados correctamente B = Número de elementos de datos que requiere la encriptación/ desenscriptación Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester

Integridad	Prevención de corrupción de datos	Interna/Externa	¿Hasta qué punto se puede prevenir la corrupción de datos?	Contar el número de casos de corrupción de datos ocurridos en la actualidad y el número de accesos donde se espera que ocurran daños de datos	$X = A / B$ A = Número de casos de corrupción de datos ocurridos en la actualidad B = Número de accesos donde se espera que ocurran daños de datos Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0, es el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester
No repudio	Utilización de firma digital	Interna/Externa	¿Qué proporción de eventos que requieran no - repudio se procesan utilizando la firma digital?	Contar el número de eventos procesados usando firma digital y el número de eventos que requieran la propiedad de no - repudio	$X = A / B$ A = Número de eventos procesados usando firma digital B = Número de eventos que requieran la propiedad de no - repudio Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester
Responsabilidad	Capacidad de auditoría de acceso	Interna/Externa	¿Qué tan completa es la pista de auditoría en relación al acceso de los usuarios al sistema y a los datos?	Contar el número de accesos al sistema y los datos registrados en el log del sistema y el número de accesos ocurridos en la realidad	$X = A / B$ A = Número de accesos ocurridos en la realidad B = Número de accesos al sistema y los datos registrados en el log del sistema Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester

Autenticidad	Métodos de autenticación	Interna/Externa	¿Qué tan bien el sistema autentica la identidad de un sujeto o recurso?	Contar el número de métodos de autenticación previstos	$X = A$ A = Número de métodos de autenticación previstos	$X \geq 0$ Donde X es mayor a 0, siendo X el mejor igual o mayor a 2	X= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester
---------------------	--------------------------	-----------------	---	--	---	---	-------------	--

Nota: Basado en ISO/IEC 25023

Tabla 2.6

Métricas de calidad interna/externa para Compatibilidad

Métricas para la característica de calidad Compatibilidad								
Subcaracterística	Métrica	Fase del ciclo de vida de calidad del producto	Propósito de la métrica de calidad	Método de aplicación	Fórmula	Valor deseado	Tipo de medida	Recursos utilizados
Co – existencia	Co – Existencia disponible	Interna/Externa	¿Qué tan adaptable es el sistema en compartir su entorno con otros sistemas sin causar efectos adversos?	Contar el número de entidades con las que el producto puede coexistir y el número de entidades en el entorno de operación que requieren de coexistencia	$X = A/B$ A = Número de entidades con las que el producto puede coexistir B = Número de entidades en el entorno de operación que requieren de coexistencia Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester

<p>Interoperatividad</p>	<p>Conectividad con sistemas externos</p>	<p>Interna/Externa</p>	<p>¿Qué tan correctamente se ha implementado los protocolos de interfaz externa?</p>	<p>Contar el número de interfaces implementadas con otros sistemas y el número total de interfaces externas</p>	<p>$X = A/B$ A= Número de interfaces implementadas con otros sistemas B = Número total de interfaces externas Dónde: $B > 0$</p>	<p>$0 \leq X \leq 1$ El más cercano a 1 es el mejor</p>	<p>X= Contable/ Contable A= Contable B= Contable</p>	<p>Especificación de requerimientos, Código fuente, Desarrollador, Tester</p>
	<p>Capacidad de intercambiar de datos</p>	<p>Interna/Externa</p>	<p>¿Qué tan exacto es el intercambio de datos entre el sistema otros sistemas de enlace?</p>	<p>Contar el número de datos que se han intercambiado sin problemas con otro sistema y el número total de datos que se intercambiarán</p>	<p>$X = A/B$ A= Número de datos que se han intercambiado sin problemas con otro sistema B = Número total de datos que se intercambiarán Dónde: $B > 0$</p>	<p>$0 \leq X \leq 1$ El más cercano a 1 es el mejor</p>	<p>X= Contable/ Contable A = Contable B = Contable</p>	<p>Especificación de requerimientos, Código fuente, Desarrollador, Tester</p>

Nota: Basado en ISO/IEC 25023

Tabla 2.7

Métricas de calidad interna/externa para Mantenibilidad

Métricas para la característica de calidad Mantenibilidad								
Subcaracterística	Métrica	Fase del ciclo de vida de calidad del producto	Propósito de la métrica de calidad	Método de aplicación	Fórmula	Valor deseado	Tipo de medida	Recursos utilizados
Modularidad	Capacidad de condensación	Interna	¿Qué tan fuerte es la relación entre los componentes del sistema?	Contar el número de componentes que no son afectados por cambios de otros componentes y el número total de componentes específicos	$X = A / B$ A = Número de componentes que no son afectados por cambios de otros componentes B = Número total de componentes específicos Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0, es el mejor	X= Contable/ Contable A= Contable B= Contable	Código fuente
	Acoplamiento de clases	Interna	¿Qué tan fuerte es la relación entre una función del sistema con otras clases implementadas?	Contar el número de relaciones que tiene una función con respecto a otras clases	$X = A$ A = Número de relaciones que tiene una función con respecto a otras clases	$1 \leq X \leq 4$ El más cercano a 1, es el mejor	X= Contable A= Contable	Código fuente
Reusabilidad	Ejecución de reusabilidad	Interna	¿Cuántos elementos pueden ser reutilizados?	Contar el número de elementos reutilizados y el número total de elementos de la biblioteca reutilizable	$X = A / B$ A = Número de elementos reutilizados B = Número total de elementos de la biblioteca reutilizable Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor	X= Contable/ Contable A= Contable B= Contable	Código fuente

Capacidad de ser analizado	Capacidad de pistas de auditoría	Interna/Externa	¿Los usuarios pueden identificar fácilmente la operación específica que causó el fallo?	Contar el número de datos realmente grabados durante la operación y el número de datos previstos a grabarse para controlar el estado del sistema durante la operación	$X = A / B$ A = Número de datos realmente grabados durante la operación B = Número de datos previstos a grabarse para controlar el estado del sistema durante la operación Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor	X = Contable/ Contable A = Contable B = Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester
	Diagnóstico de funciones suficientes	Interna/Externa	¿Hasta qué punto las funciones de diagnóstico están preparadas o hasta qué punto funcionan para el análisis causal?	Contar el número de funciones de diagnóstico implementadas y contar el número de funciones de diagnóstico requeridas en la especificación de requerimientos	$X = A/B$ A = Número de funciones de diagnóstico implementadas B = Número de funciones de diagnóstico requeridas en la especificación de requerimientos Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor	X = Contable/ Contable A = Contable B = Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester
Capacidad de ser modificado	Complejidad ciclomática	Interna	¿Cuál es la complejidad estructural de un código fuente?	Contar las instrucciones condicionales, bucles, salidas de métodos y cláusulas AND y OR dentro de los condicionales.	$X = A+1$ A = Número de instrucciones condicionales que tiene una función	$1 \leq X < 15$ El más cercano a 1, es el mejor	X = Contable A = Contable	Código fuente

	Profundidad de herencia	Interna	¿Qué tan profunda es la jerarquía de las clases involucradas en una determinada función?	Contar las jerarquías empleadas en una determinada función o método.	$X = A$ A = Número de jerarquías empleadas para una determinada función.	$0 \leq X \leq 4$ El más cercano a 0 es el mejor	X= Contable A= Contable	Código fuente
	Grado de localización de corrección de impacto	Interna/Externa	¿Hasta qué punto los problemas causados pueden tener como consecuencia un mantenimiento?	Contar el número de fallas aparecidas después que se ha resuelto un fallo y contar el número de fallas resultas	$X = A/B$ A = Número de fallas aparecidas después que se ha resuelto un fallo B = Número de fallas resueltas Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0, es el mejor	X= Contable A= Contable B = Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester
	Complejidad de modificación	Externa	¿Con qué facilidad el desarrollador puede modificar el software para resolver problemas?	Tomar el tiempo de trabajo que le toma al desarrollador modificar y contar el número de modificaciones	$X = A/T$ A = Número de modificaciones B = a Tiempo de trabajo que le toma al desarrollador modificar Dónde: $T > 0$	$X = A/T$ El más lejano a 0/t es el mejor	X= Contable/ Tiempo A= Contable B= Tiempo	Desarrollador
	Índice de éxito de modificación	Externa	¿Hasta qué punto puede el sistema ser operado sin fallas después del mantenimiento?	Contar el número de problemas dentro de un determinado período antes de mantenimiento y contar el número de problemas en el mismo período después del mantenimiento	$X = A/B$ A = Número de problemas dentro de un determinado período antes de mantenimiento B = Número de problemas en el mismo período después del mantenimiento Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0, es el mejor	X= Contable/ Contable A= Contable B= Contable	Desarrollador

Capacidad de ser probado	Complejidad funcional de funciones de pruebas	Interna	¿Son las funciones de pruebas completas y fáciles de implementar?	Contar el número de funciones de prueba implementadas y contar el número de funciones de prueba requeridas	$X = A/B$ A = Número de funciones de prueba implementadas B = Número de funciones de prueba requeridas Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor	X= Contable/ Contable A= Contable B= Contable	Código fuente, Tester
	Capacidad de prueba autónoma	Interna	¿Qué tan independiente es el software al ser probado?	Contar el número de pruebas que están dependiendo de otros sistemas y contar el número total de pruebas dependientes con otros sistemas	$X = A/B$ A = Número de pruebas que están dependiendo de otros sistemas B = Número total de pruebas dependientes con otros sistemas Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0, es el mejor	X= Contable/ Contable A= Contable B= Contable	Código fuente, Tester
	Capacidad de reinicio de pruebas	Externa	¿Con qué facilidad se puede llevar a cabo las pruebas nuevamente después del mantenimiento?	Contar el número de casos en los cuales el mantenedor puede pausar y restaurar las pruebas y contar el número de casos de pausa en la ejecución de pruebas	$X = A/B$ A = Número de casos en los cuales el mantenedor puede pausar y restaurar las pruebas B = Número de casos de pausa en la ejecución de pruebas Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor	X= Contable/ Contable A= Contable B= Contable	Desarrollador, Tester

Nota: Basado en ISO/IEC 25023

Tabla 2.8

Métricas de calidad interna/externa para Portabilidad

Métricas para la característica de calidad Portabilidad								
Subcaracterística	Métrica	Fase del ciclo de vida de calidad del producto	Propósito de la métrica de calidad	Método de aplicación	Fórmula	Valor deseado	Tipo de medida	Recursos utilizados
Adaptabilidad	Adaptabilidad en entorno hardware	Interna/Externa	¿Es el sistema lo suficientemente capaz de adaptarse al entorno hardware?	Contar el número de funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con el entorno hardware y contar el número total de funciones las cuales han sido probadas	$X = A/B$ A = Número de funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con el entorno hardware B = Número total de funciones que han sido probadas Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0, es el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester
	Adaptabilidad en entorno de software	Interna/Externa	¿Es el sistema lo suficientemente capaz de adaptarse al entorno del sistema software?	Contar el número de funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con el sistema y contar el número total de funciones las cuales han sido probadas	$X = A/B$ A = Número de funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con el sistema B = Número total de funciones que han sido probadas Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0, es el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester

	Adaptabilidad en entorno empresarial	Interna/Externa	¿Es el sistema lo suficientemente capaz de adaptarse al entorno operacional?	Contar el número de funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con usuarios del entorno empresarial y contar el número total de funciones las cuales han sido probadas	$X = A/B$ $A =$ Número de funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con usuarios del entorno empresarial $B =$ Número total de funciones que han sido probadas Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0, es el mejor	$X =$ Contable/ Contable $A =$ Contable $B =$ Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester
Capacidad de ser Instalado	Eficiencia en el tiempo de instalación	Externa	¿Cuánto tiempo es requerido para realizar una instalación?	Contar el tiempo total transcurrido al instalar el sistema y contar el número de reintentos al instalar el sistema	$X = A/T$ $A =$ Número de reintentos al instalar el sistema $T =$ Tiempo total transcurrido al instalar el sistema Dónde: $T > 0$	$X = A/T$ El más lejano a 0/t es el mejor	$X =$ Contable/ Contable $A =$ Contable $B =$ Contable	Desarrollador, Tester
	Facilidad de instalación	Externa	¿Puede fácilmente el usuario o el desarrollador instalar el software en un entorno operacional?	Contar el número de casos en que los usuarios tuvieron éxito al instalar el sistema cambiando proceso de instalación para su conveniencia y contar el número total de casos en que los usuarios han intentado cambiar el proceso de instalación para su conveniencia	$X = A/B$ $A =$ Número de casos en que los usuarios tuvieron éxito al instalar el sistema cambiando proceso de instalación para su conveniencia $B =$ Número total de casos en que los usuarios han intentado cambiar el proceso de instalación para su conveniencia Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor	$X =$ Contable/ Contable $A =$ Contable $B =$ Contable	Desarrollador, Tester

Capacidad de ser Reemplazado	Consistencia en la función de soporte al usuario	Interna/Externa	¿Cuán consistente es el nuevo componente con la interfaz de usuario existente?	Contar el número de nuevas funciones que son consideradas como no consistentes por el usuario y contar el número de nuevas funciones	$X = A/B$ A = Número de nuevas funciones que son consideradas como no consistentes por el usuario B = Número de nuevas funciones Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0, es el mejor	X= Contable/ Contable A= Contable B= Contable	Especificación de requerimientos, Código fuente, Desarrollador, Tester
	Inclusividad funcional	Externa	¿Pueden fácilmente las funciones ser utilizadas después de ser cambiadas a por otras similares?	Contar el número de funciones que producen resultados similares con anterioridad y que no se han exigido cambios y contar el número de funciones probadas que son similares a las funciones proporcionadas por otro software para ser reemplazado	$X = A/B$ A = Número de funciones que producen resultados similares con anterioridad y que no se han exigido cambios B = Número de funciones probadas que son similares a las funciones proporcionadas por otro software para ser reemplazado Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor	X= Contable/ Contable A= Contable B= Contable	Desarrollador, Tester

	Uso continuo de datos	Externa	¿Pueden los datos fácilmente ser utilizados después de reemplazar el software por otro similar?	Contar el número de datos que son continuamente utilizables por el software a ser reemplazado y contar el número de datos que son continuamente reutilizables por el software a ser reemplazado	$X = A/B$ A = número de datos que son continuamente solo utilizables por el software a ser reemplazado B = Número de datos que son reutilizables por el software a ser reemplazado Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor	X= Contable/ Contable A= Contable B= Contable	Desarrollador, Tester
--	-----------------------	---------	---	---	--	--	--	--------------------------

Nota: Basado en ISO/IEC 25023

Tabla 2.9

Métricas de calidad en uso de Efectividad

Métricas para la característica de calidad Efectividad								
Subcaracterística	Métrica	Fase del ciclo de vida de calidad del producto	Propósito de la métrica de calidad	Método de aplicación	Fórmula	Valor deseado	Tipo de medida	Recursos utilizados
Efectividad	Compleitud de la tarea	Uso	¿Qué cantidad de tareas son completadas correctamente?	Contar el número de tareas completadas y el número total de tareas intentadas	$X = A/B$ A= Número de tareas completadas B = Número total de tareas intentadas Dónde: B > 0	$0 \leq X \leq 1$ El más cercano a 1, el mejor	X= Contable /Contable A= Contable B= Contable	Usuario
	Efectividad de la tarea	Uso	¿Qué cantidad de los objetivos de la tarea se realiza completamente?	Tomar el valor proporcional de cada componente faltante o incorrecto en la salida de la tarea	$X = A/B$ A=Cantidad de objetivos completados por la tarea. B=Cantidad de objetivos planeados que realice la tarea	$0 \leq X \leq 1$ El más cercano a 1, el mejor	X= Contable A= Contable	Usuario
	Frecuencia de error	Uso	¿Cuál es la frecuencia de los errores cometidos por el usuario en comparación con lo planeado?	Contar el número de errores cometidos por los usuarios y contar el número de tareas	$X = A/B$ A = Número de errores cometidos por los usuarios B = Número de tareas Dónde: B > 0	$0 \leq X \leq 1$ El más cercano a 0 es el mejor	X= Contable /Contable A= Contable B= Contable	Usuario

Nota: Basado en ISO/IEC 25023

Tabla 2.10

Métricas de calidad en uso de Eficiencia

Métricas para la característica de calidad Eficiencia								
Subcaracterística	Métrica	Fase del ciclo de vida de calidad del producto	Propósito de la métrica de calidad	Método de aplicación	Fórmula	Valor deseado	Tipo de medida	Recursos utilizados
Eficiencia	Tiempo de la tarea	Uso	¿Cuánto tiempo se tarda en completar una tarea en comparación con lo planeado?	Tomar el tiempo planeado y el tiempo actual	$X = A/B$ A= Tiempo actual B = Tiempo planeado Dónde: $A > 0$	$0 \leq X \leq 1$ Si $A \leq B$ el más cercano a 0 es lo mejor. Si $A > B$ será considerado como el peor caso	X= Tiempo/Tiempo A= Tiempo B= Tiempo	Usuario
	Tiempo relativo de la tarea	Uso	¿Cuánto tiempo necesita un usuario normal en completar una tarea en comparación con un experto?	Tomar el tiempo que completa una tarea un usuario normal y el tiempo que completa una tarea un usuario experto	$X = A/B$ A = Tiempo que completa una tarea un usuario experto B = Tiempo que completa una tarea un usuario normal Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, el mejor	X= Tiempo/Tiempo A= Tiempo B= Tiempo	Usuario

	Eficiencia de la tarea	Uso	¿Qué tan eficientes son los usuarios?	Contar el número de tareas efectivas y tomar el tiempo de la tarea	$X = A/T$ $A = \text{Número de tareas efectivas}$ $T = \text{Tiempo de la tarea}$ Dónde: $T > 0$	$X = A/T$ El más lejano a 0/t es el mejor	$X = \text{Contable/ Tiempo}$ $A = \text{Contable}$ $B = \text{Tiempo}$	Usuario
	Eficiencia relativa de la tarea	Uso	¿Qué tan eficiente es un usuario comparado con lo planeado?	Contar el número de tareas eficientes realizadas por un usuario ordinario y contar el número de tareas eficientes planeadas	$X = A/B$ $A = \text{Número de tareas eficientes realizadas por un usuario ordinario}$ $B = \text{Número de tareas eficientes planeadas}$ Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es lo mejor	$X = \text{Contable/ Contable}$ $A = \text{Contable}$ $B = \text{Contable}$	Usuario, Especificación de requerimientos
	Productividad económica	Uso	¿Qué tan rentable es el usuario?	Contar el número de tareas efectivas y tomar el costo total de las tareas	$X = A/B$ $A = \text{Número de tareas efectivas}$ $B = \text{Número de tareas totales}$ Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es lo mejor	$X = \text{Contable/ Contable}$ $A = \text{Contable}$ $B = \text{Contable}$	Usuario, Especificación de requerimientos

	Porcentaje productivo	Uso	¿Cuál es el porcentaje de tiempo que el usuario realiza acciones de productividad?	Tomar el tiempo de productividad y el tiempo de la tarea	$X = A/B$ A = Tiempo de la tarea B = Tiempo de productividad. Dónde: $B > 0$	$0 \leq X \leq 1$ Si $A \leq B$ el más cercano a 0 es lo mejor. Si $A > B$ será considerado como el peor caso	$X = \text{Tiempo} / \text{Tiempo}$ A= Tiempo B= Tiempo	Usuario
	Numero relativo de acciones del usuario	Uso	¿Cuál es el número de acciones mínimas necesarias que realizan los usuarios?	Contar el número de acciones realizadas por los usuarios y contar el número de acciones necesarias actualmente	$X = A/B$ A = Número de acciones realizadas por los usuarios B = Número de acciones necesarias actualmente Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es lo mejor	$X = \text{Contable} / \text{Contable}$ A= Contable B= Contable	Usuario

Nota: Basado en ISO/IEC 25023

Tabla 2.11

Métricas de calidad en uso de Satisfacción

Métricas para la característica de calidad Satisfacción								
Subcaracterística	Métrica	Fase del ciclo de vida de calidad del producto	Propósito de la métrica de calidad	Método de aplicación	Fórmula	Valor deseado	Tipo de medida	Recursos utilizados
Utilidad	Nivel de satisfacción	Uso	¿Qué tan satisfecho está el usuario?	Realizar un cuestionario sobre el nivel de satisfacción sobre el sistema.	$X = A/B$ A= Numero de preguntas con respuesta satisfactorias B = Número total de preguntas realizadas en el cuestionario. Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es lo mejor	X = Contable / Contable A = Contable B = Contable	Usuario
	Uso discrecional de las funciones	Uso	¿Qué porcentaje de los usuarios optan por utilizar las funciones sistema?	Observación de uso	$X = A/B$ A= Número de funciones específicas del software que se utilizan B= Número total de funciones que están destinados a ser usados Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, mejor	X= Contable / Contable A= Contable B= Contable	Usuario

	Porcentaje de quejas de los clientes	Uso	¿Cuál es el porcentaje de quejas realizadas por los clientes?	Contar el número de clientes que se quejan y contar el número total de clientes	$X = A/B$ A = Número de clientes que se quejan B = Número total de clientes Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0, mejor	X = Contable/ Contable A = Contable B = Contable	Cientes
--	--------------------------------------	-----	---	---	---	--	---	---------

Nota: Basado en ISO/IEC 25023

Tabla 2.12

Métricas de calidad en uso de Libertad de riesgo

Métricas para la característica de calidad Libertad de riesgo								
Subcaracterística	Métrica	Fase del ciclo de vida de calidad del producto	Propósito de la métrica de calidad	Método de aplicación	Fórmula	Valor deseado	Tipo de medida	Recursos utilizados
	Retorno de la Inversión (ROI)	Uso	¿Cuál es el retorno de la inversión?	Consultar los beneficios obtenidos y el capital invertido	$X = A / B$ A = Beneficios obtenidos B = Beneficios esperados. Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es lo mejor	X = Contable / Contable A = Contable B = Contable	Cliente

Libertad del riesgo económico	Tiempo para lograr el retorno de la inversión	Uso	¿El retorno de la inversión es logrado en un tiempo aceptable?	Tomar el tiempo para lograr el ROI y tomar el tiempo aceptable para lograr el ROI	$X = A/B$ A = Tiempo real para lograr el ROI B = Tiempo aceptable para lograr el ROI Dónde: $B > 0$	$0 \leq X \leq 1$ Si $A \leq B$ el más cercano a 0 es lo mejor. Si $A > B$ será considerado como el peor caso	$X =$ Tiempo / Tiempo A = Tiempo B = Tiempo	Usuario
	Rendimiento relativo de negocios	Uso	¿Qué tan comparable es el rendimiento del negocio con otras empresas de primera clase en la industria o en la misma empresa	Consultar el monto de la inversión de TI o de las ventas de la empresa y el monto de inversión de TI o las ventas planeadas de la empresa para la comparación	$X = B/A$ A = Monto de inversión de TI o las ventas planeadas de la empresa para la comparación B = Monto real de la inversión de TI o de las ventas de la empresa Dónde: $B > 0$	$0 \leq X \leq 1$ Si $B \leq A$ el más cercano a 1 es lo mejor. Si $B > A$ será considerado como el mejor caso		Cliente
	Balanced Score Card	Uso	Los beneficios de la inversión en IT evaluados utilizando los Balanced Score Card para cumplir los objetivos	Consultar el resultado del BSC y el BSC planeado	$X = A/B$ A = Resultado del BSC B = BSC planeado Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, mejor.		Cliente
	Tiempo de entrega	Uso	¿Cuál es el tiempo de entrega para cumplir los con objetivos?	Consultar el tiempo de entrega actual o retrasos en las entregas y el tiempo de entrega planeado o retrasos en las entregas	$X = A/B$ A = Tiempo de entrega planeado o retrasos en las entregas B = Tiempo de entrega actual o retrasos en las entregas Dónde: $B > 0$	$0 \leq X \leq 1$ Si $A \leq B$ el más cercano a 0 es lo mejor. Si $A > B$ será considerado como el peor caso	$X =$ Tiempo/ Tiempo A= Tiempo B= Tiempo	Cliente

	Ganancias para cada cliente	Uso	Las ganancias de cada cliente al cumplir con sus objetivos	Consultar los ingresos reales de un cliente y los ingresos planeados de un cliente	$X = A/B$ A = Ingresos reales de un cliente B = Ingresos planeados de un cliente Dónde: $B > 0$	$0 \leq X \leq 1$ Si $A \leq B$ el más cercano a 1 es lo mejor. Si $A > B$ será considerado como el mejor caso	X= Contable/ Contable A= Contable B= Contable	Cliente
	Errores con consecuencias económicas	Uso	La frecuencia de errores humanos o del sistema con consecuencias económicas	Contar el número de errores con consecuencias económicas y contar número total de situaciones de uso	$X = A/B$ A = Número de errores con consecuencias económicas B = Número total de situaciones de uso Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0 es el mejor	X= Contable/ Contable A= Contable B= Contable	Cliente
	Corrupción del software	Uso	La frecuencia de corrupción del software resultado de errores humanos o del sistema	Contar el número de ocurrencias de corrupción del software y contar número total de situaciones de uso.	$X = A/B$ A = Número de ocurrencias de corrupción del software B = Número total de situaciones de uso Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0 es el mejor	X= Contable/ Contable A= Contable B= Contable	Cliente

Libertad del riesgo de salud y seguridad	Frecuencia de problemas en la salud y seguridad del usuario	Uso	La frecuencia de problemas de salud entre los usuarios del producto	Contar el número de usuarios que notificaron problemas de salud y contar el número total de usuarios	$X = A/B$ A = Número de usuarios que notificaron problemas de salud B = Número total de usuarios Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0 es el mejor	X= Contable/ Contable A= Contable B= Contable	Usuario
	Impacto en la salud y seguridad del usuario	Uso	El impacto en la salud y la seguridad en los usuarios del producto	Contar el número de personas afectadas, tomar el tiempo y el grado de importancia	$X = A/T$ A = Número de personas afectadas T = Tiempo	$0 \leq X \leq 5$ El más cercano a 0 es el mejor		
	Seguridad de las personas afectadas por el uso del sistema	Uso	La incidencia de riesgo para las personas afectadas por el uso del sistema	Contar el número de personas puestas en peligro y contar el número total de personas potencialmente afectadas por el sistema	$X = A/B$ A = Número de personas puestas en peligro B = Número total de personas potencialmente afectadas por el sistema Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0 es el mejor	X= Contable/ Contable A= Contable B= Contable	Usuario
Libertad del riesgo ambiental	Impacto Ambiental	Uso	El impacto ambiental de la elaboración y el uso del sistema	Estimar el impacto ambiental y el impacto ambiental aceptable	$X = A/B$ A = Impacto ambiental aceptable B = Impacto ambiental real Dónde: $B > 0$	$0 \leq X \leq 1$ Si $A \leq B$ el más cercano a 0 es lo mejor. Si $A > B$ será considerado como el peor caso	X= Contable/ Contable A= Contable B= Contable	Usuario, cliente, desarrollador

Nota: Basado en ISO/IEC 25023

Tabla 2.13

Métricas de calidad en uso de Cobertura de contexto

Métricas para la característica de calidad Cobertura de contexto								
Subcaracterística	Métrica	Fase del ciclo de vida de calidad del producto	Propósito de la métrica de calidad	Método de aplicación	Fórmula	Valor deseado	Tipo de medida	Recursos utilizados
Compleitud de Contexto	Compleitud de Contexto	Uso	Porcentaje en que el producto puede utilizarse con facilidad en contextos de uso	Contar el número de contextos con la facilidad de uso inaceptable y el número total de distintos contextos de uso	$X = A/B$ A= Número de distintos contextos de uso inaceptables B = Número total de distintos contextos de uso Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0 es el mejor	X = Contable /Contable A = Contable B = Contable	Cliente
Flexibilidad	Función flexible del diseño.	Uso	Grado en que el producto puede adaptarse para satisfacer las diferentes necesidades de los usuarios	Contar el número de características diseñadas con completa flexibilidad y contar el número total de características de diseño	$X = A/B$ A= Número de características diseñadas con completa flexibilidad B = Número total de características de diseño Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor	X = Contable /Contable A = Contable B = Contable	Usuario

Nota: Basado en ISO/IEC 25023

2.8.5.4. Definición de características de calidad

Las características de calidad que se presenta en la norma están aplicadas para todos los sistemas de software, pero dependiendo del tipo de sistema a evaluar las características tendrán un grado de importancia mayor o menor que otros, por lo tanto las características de calidad que se aplicarán a un producto software se encontrarán definidas por el tipo de producto.

Tabla 2.14

Definición del nivel de importancia

Nivel de importancia	Simbología	Porcentaje referencial del nivel de importancia	Significado
Alto	A	70% - 100%	El grado de importancia de la característica y subcaracterística es alto por ende se realizará las mediciones
Medio	M	25% - 69%	La característica y subcaracterística no es tan relevante pero puede o no ser medida dependiendo del criterio del evaluador
Bajo	B	1% - 24%	La característica y subcaracterística no tiene relevancia y no será medida.
No Aplica	NA	0%	Este valor se dará a la característica y subcaracterística que no se pueden medir dependiendo de diferentes factores

Nota: Basado en ISO/IEC 25023

Por ejemplo, en la Tabla 2.15, Tabla 2.16 y Tabla 2.17 se especifica el nivel de importancia de las características de calidad interna, externa y en uso de un producto software de tipo página web.

Tabla 2.15

Ejemplo del nivel de importancia para las características de calidad interna más relevantes

CARACTERÍSTICAS DE CALIDAD INTERNA	
Características	Nivel de importancia
Adecuación funcional	M
Fiabilidad	B
Eficiencia en el desempeño	M
Facilidad de uso	M
Seguridad	M
Compatibilidad	B
Mantenibilidad	A
Portabilidad	NA

Nota: Basado en ISO/IEC 25023

Tabla 2.16

Ejemplo del nivel de importancia para las características de calidad externa más relevantes

CARACTERÍSTICAS DE CALIDAD EXTERNA	
Características	Nivel de importancia
Adecuación funcional	A
Fiabilidad	M
Eficiencia en el desempeño	M
Facilidad de uso	M
Seguridad	B
Compatibilidad	A
Mantenibilidad	M
Portabilidad	NA

Nota: Basado en ISO/IEC 25023

Tabla 2.17

Ejemplo del nivel de importancia para las características de calidad en uso más relevantes

CARACTERÍSTICAS DE CALIDAD EN USO	
Características	Nivel de importancia

Efectividad	A
Eficiencia	A
Satisfacción	A
Libertad de Riesgo	B
Cobertura de Contexto	B

Nota: Basado en ISO/IEC 25023

2.8.5.5. Definición de subcaracterísticas y atributos de calidad interna/externa más relevantes

En la Tabla 2.18 y Tabla 2.19 se especifica como ejemplo, el nivel de importancia de las subcaracterísticas y atributos de calidad interna y externa, respectivamente, de un producto software de tipo página web.

Tabla 2.18

Ejemplo del nivel de importancia de subcaracterísticas y atributos de calidad interna más relevantes

SUBCARACTERÍSTICAS Y ATRIBUTOS DE CALIDAD INTERNA		
Características	Subcaracterísticas	Nivel de importancia
Adecuación funcional	Complejidad funcional	A
	Exactitud funcional	B
Fiabilidad	Madurez	B
	Disponibilidad	B
	Tolerancia a fallos	B
	Recuperabilidad	B
Eficiencia en el desempeño	Comportamiento Temporal	B
	Utilización de recursos	A
	Capacidad	B
Facilidad de uso	Capacidad de reconocer su adecuación	B
	Capacidad de ser entendido	A
	Operatividad	A
	Protección frente a errores de	M

	usuarios	
	Estética de la interfaz de usuario	B
	Accesibilidad técnica	B
Seguridad	Confidencialidad	B
	Integridad	B
	No repudio	A
	Responsabilidad	B
	Autenticidad	A
Compatibilidad	Co - existencia	B
	Interoperabilidad	B
Mantenibilidad	Modularidad	A
	Reusabilidad	M
	Capacidad de ser analizado	M
	Capacidad de modificación	A
	Capacidad de ser probado	B
Portabilidad	Adaptabilidad	NA
	Facilidad de instalación	NA
	Capacidad de ser reemplazado	NA

Nota: Basado en ISO/IEC 25023

Tabla 2.19

Ejemplo del nivel de importancia de subcaracterísticas y atributos de calidad externa más relevantes

SUBCARACTERÍSTICAS Y ATRIBUTOS DE CALIDAD EXTERNA		
Características	Subcaracterísticas	Nivel de importancia
Adecuación funcional	Complejidad funcional	A
	Exactitud funcional	A
Fiabilidad	Madurez	A
	Disponibilidad	A
	Tolerancia a fallos	M
	Recuperabilidad	M

Eficiencia en el desempeño	Comportamiento Temporal	A
	Utilización de recursos	A
	Capacidad	B
Facilidad de uso	Capacidad de reconocer su adecuación	B
	Capacidad de ser entendido	A
	Operatividad	A
	Protección frente a errores de usuarios	B
	Estética de la interfaz de usuario	B
	Accesibilidad técnica	B
Seguridad	Confidencialidad	B
	Integridad	B
	No repudio	B
	Responsabilidad	B
	Autenticidad	B
Compatibilidad	Co - existencia	A
	Interoperabilidad	A
Mantenibilidad	Modularidad	NA
	Reusabilidad	NA
	Capacidad de ser analizado	B
	Capacidad de modificación	A
	Capacidad de ser probado	B
Portabilidad	Adaptabilidad	NA
	Facilidad de instalación	NA
	Capacidad de ser reemplazado	NA

Nota: Basado en ISO/IEC 25023

Tabla 2.20

Ejemplo del nivel de importancia de subcaracterísticas y atributos de calidad en uso más relevantes

SUBCARACTERÍSTICAS Y ATRIBUTOS DE CALIDAD EN USO		
Características	Subcaracterísticas	Nivel de importancia

Efectividad	Efectividad	A
Eficiencia	Eficiencia	A
Satisfacción	Utilidad	A
Libertad de riesgo	Libertad del riesgo económico	B
	Libertad del riesgo de salud y seguridad	B
	Mitigación del riesgo ambiental	B
Cobertura de contexto	Integridad de contexto	B
	Flexibilidad	B

Nota: Basado en ISO/IEC 25023

2.8.5.6. Modelo de indicadores y métricas

Ponderación en porcentaje de las características de calidad interna, externa y en uso más relevantes para el producto software

La ponderación que se otorgara a las características de calidad interna, externa y en uso dependerá del nivel de importancia que fue asignado al producto software en la sección 2.8.5.4. *Definición de características de calidad*. Tomando en cuenta que las ponderaciones que serán asignadas dependerán del criterio del evaluador y del tipo de producto a evaluarse, además estas ponderaciones deben ser divididas entre las características que fueron aplicadas, y la sumatoria no debe pasar del 100%.

A continuación en la Tabla 2.21 se presenta un ejemplo de la ponderación en porcentaje que se asigna a las características más relevantes para la calidad interna de un producto software de tipo página web.

Tabla 2.21*Ejemplo de ponderación en porcentajes para la calidad interna*

CARACTERÍSTICAS DE CALIDAD INTERNA		
Características	Nivel de importancia	Ponderación
Adecuación funcional	M	25%
Fiabilidad	B	0%
Eficiencia en el desempeño	M	15%
Facilidad de uso	M	15%
Seguridad	M	15%
Compatibilidad	B	0%
Mantenibilidad	A	30%
Portabilidad	NA	0%

Nota: Basado en ISO/IEC 25023**A) Niveles de puntuación final para la calidad interna, externa y en uso**

La escala de medición que se presenta en la Tabla 2.22 se utilizará para analizar el resultado final de las características de calidad interna, externa y en uso, los cuales nos dará el nivel de puntuación final que se le asignará al producto software después de su análisis.

Tabla 2.22*Niveles de puntuación final para la calidad interna, externa y en uso*

Escala de medición	Niveles de puntuación	Grado de satisfacción
8.75 - 10	Cumple con los requisitos	Muy satisfactorio
5 – 8.74	Aceptable	Satisfactorio
2.75 – 4.9	Mínimamente aceptable	Insatisfactorio
0 – 2.74	Inaceptable	

Nota: Basado en ISO/IEC 25023

2.9. SEGURIDAD DEL SOFTWARE

2.9.1. Estándar ISO/IEC 27000

La serie de normas ISO/IEC 27000 son estándares de seguridad publicados por la Organización Internacional para la Estandarización (ISO) y la Comisión Electrotécnica Internacional (IEC).

La serie contiene las mejores prácticas recomendadas en Seguridad de la información para desarrollar, implementar y mantener Especificaciones para los Sistemas de Gestión de la Seguridad de la Información (SGSI). (ISO27000, 2021)

2.9.1.1. ISO/IEC 27001

Es la certificación que deben obtener las organizaciones. Norma que especifica los requisitos para la implantación del SGSI. Es la norma más importante de la familia. Adopta un enfoque de gestión de riesgos y promueve la mejora continua de los procesos. Fue publicada como estándar internacional en octubre de 2005.

2.9.1.2. ISO/IEC 27002

La ISO 27002 proporciona las mejores prácticas de la gestión de seguridad de información a todos los interesados en iniciar, implantar o mantener sistemas de gestión de la seguridad de la información. La seguridad de la información se define en el estándar como la prevención de la confidencialidad que solo quienes estén autorizados puedan acceder a la información, integridad que la información y sus métodos de proceso son exactos y completos y disponibilidad que los usuarios autorizados tienen acceso a la información del sistema cuando lo requieran.

2.9.2. Niveles de acceso al sistema

- **Webmaster (Súper Usuario):** El "Súper Usuario" es el primer Usuario (con ID = 1) que se crea cuando se instala el sistema y tiene acceso irrestricto a todas las secciones y contenidos del sistema, y no se ve afectado por los privilegios.

- **Admin:** Puede tener acceso a todo excepto a la sección de Configuración. Tenga presente que los Admin no pueden gestionar a Usuarios Webmaster o a otros Admin.
- **Autor:** Sólo puede tener acceso a las secciones Canales, Clips, Galerías, Noticias & Eventos, Programador; adicionalmente, puede tener acceso a la pantalla de edición de su perfil de Usuario.
- **Colaborador:** Como Autor, pero el contenido creado por Colaboradores requiere de moderación para que esté públicamente disponible.
- **Suscriptor:** Sólo puede tener acceso a la pantalla de edición de su perfil de Usuario.

2.10. METODOS DE ESTIMACIÓN DE COSTOS

2.10.1. Modelo Constructivo de Costos (COCOMO II)

El Modelo Constructivo de Costo COCOMO (Constructive Cost Model) es un modelo matemático de base empírica utilizada para estimación de costos de software incluye tres sub módulos, cada uno ofrece un nivel de detalle y aproximación del software: básico, intermedio y detallado. (Gómez, 2021)

El modelo se utiliza para obtener una primera aproximación rápida del esfuerzo, y hace uso de la siguiente tabla de constantes para calcular distintos aspectos de costes:

$$E = a * (Kl)^b + m(X), \text{ en persona mes}$$

$$Tdev = c * (E)^d, \text{ en meses}$$

$$P = \frac{E}{Tdev}, \text{ en personas}$$

Donde:

- **E.** es el esfuerzo requerido por el proyecto, en persona-mes
- **Tdev.** es el tiempo requerido por el proyecto, en meses
- **P.** Es el número de personas requerido para el proyecto.
- **a, b, c, d.** Son constantes con valores definidos en una tabla, según cada submodulo.

- **KI.** Son el número de miles de líneas de código fuente que tiene el software que estamos intentando estimar.
- **m(X): Multiplicador que depende de 15 atributos.**
- **Modo orgánico:** es un pequeño grupo de programadores experimentados desarrollando software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas docenas de miles (medio).
- **Modo semi – libre o semi – acoplado:** Corresponde a un esquema intermedio entre el orgánico y el rígido, el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.
- **Modo rígido o empotrado:** El proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único, siendo difícil basarse en la experiencia puesto que puede no haberla.

La tabla 2.23 muestra los coeficientes del proyecto de software.

Tabla 2.23

Coeficiente de coste modelo básico

PROYECTO SOFTWARE	A	B	C	D
Orgánico	2.4	1.05	2.5	0.38
Semi-Libre	3.0	1.12	2.5	0.35
Rígido	3.6	1.20	2.5	0.32

Nota: (Adriana Gómez, 2021)

2.11. HERRAMIENTAS DE DESARROLLO

2.11.1. Lenguaje de Programación PHP (HYPERTEXT PRE-PROCESSOR)

Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían

incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos.

El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

Fue creado originalmente por Rasmus Lerdorf en 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. Este lenguaje forma parte del software libre publicado bajo la licencia PHP, que es incompatible con la Licencia Pública General de GNU debido a las restricciones del uso del término PHP. (Sites Google, 2021)

2.11.1.1. Ventajas y desventajas

Ventajas

- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- PHP es que puede funcionar en un servidor Windows y en LINUX. Su gran comunidad de PHP hace que el soporte, guías, libros y soluciones de dudas sea mucho más fácil en foros o redes sociales
- PHP no requiere ningún tipo de licencia
- Permite las técnicas de programación orientada a objetos.

Desventajas

- El lugar más seguro para ejecutar una aplicación es en un servidor propio, por lo cual, si un cliente o usuario requiere su código en su pc, tendríamos que dejar su código, sin manera de ocultarlo, aunque hay muchas aplicaciones que nos ayuda a encriptar el código fuente.
- Debes saber cuándo menos HTML para poder hacer un trabajo medianamente funcional.
- Si no lo configuras correctamente dejas abiertas muchas brechas de seguridad.
- Se necesita instalar un servidor web.

2.11.2. Framework Codeigniter

Codeigniter es un marco de desarrollo de aplicaciones, un conjunto de herramientas, para personas que crean sitios web con PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido de lo que podría si estuviera escribiendo código desde cero, al proporcionar un amplio conjunto de bibliotecas para tareas comúnmente necesarias, así como una interfaz simple y una estructura lógica para acceder a estas bibliotecas. Codeigniter le permite concentrarse creativamente en su proyecto al minimizar la cantidad de código necesario para una tarea determinada. (Codeigniter, 2022)

Codeigniter es un framework para el desarrollo de aplicaciones en php, que utiliza el MVC. Esto permite a los programadores o desarrolladores Web mejorar su forma de trabajar, además de dar una mayor velocidad a la hora de crear páginas Webs. (Pineda, 2016)

2.11.2.1. MVC

El MVC o Modelo Vista Controlador es un patrón de arquitectura de software que separa la lógica de control, la interfaz del usuario y los datos del sistema. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir por un lado define los componentes para la representación de la información y por otro lado la interacción del usuario.

- **Modelo:** se trata de la capa que trabaja con los datos, por lo que tiene los mecanismos para acceder a la información y también actualizar su estado. Estos datos estarán habitualmente en una base de datos.
- **Vista:** esta capa contiene el código de la aplicación que va a producir la visualización de las interfaces de usuario, es decir, el código que permitirá renderizar los estados de nuestra aplicación en HTML.
- **Controlador:** contiene el código necesario para responder a las acciones que solicita la aplicación, como por ejemplo realizar una compra o visualizar un elemento. Esta capa sirve de enlace entre la vista y el modelo.

2.11.2.2. Ventajas del framework CodeIgniter

- Las páginas se procesan más rápido, el núcleo de CodeIgniter es bastante ligero.
- Es sencillo de instalar, basta con subir los archivos al ftp y tocar un archivo de configuración para definir el acceso a la base de datos.
- Existe abundante documentación en la red.
- Facilidad de edición del código ya creado.
- Facilidad para crear nuevos módulos, páginas o funcionalidades.
- Estandarización del código
- Separación de la lógica y arquitectura de la web, el MVC.
- Cualquier servidor que soporte PHP+MySQL sirve para CodeIgniter.
- CodeIgniter se encuentra bajo una licencia open source, es código libre.

2.11.3. Lenguaje de programación JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. (EcuRed, 2022)

2.11.3.1. Características de JavaScript

- Es simple, no hace falta tener conocimientos avanzados de programación para aprender a manejar JavaScript y es recomendado por muchos expertos a la hora de encontrar un lenguaje para comenzar a programar.

- Maneja objetos dentro de nuestra página Web y sobre ese objeto podemos definir diferentes eventos. Dichos objetos facilitan la programación de páginas interactivas, a la vez que se evita la posibilidad de ejecutar comandos que puedan ser peligrosos para la máquina del usuario, tales como formateo de unidades, modificar archivos etc.
- Es dinámico, responde a eventos en tiempo real. Eventos como presionar un botón, pasar el puntero del mouse sobre un determinado texto o el simple hecho de cargar la página o caducar un tiempo. Con esto podemos cambiar totalmente el aspecto de nuestra página al gusto del usuario, evitándonos tener en el servidor un página para cada gusto, hacer cálculos en base a variables cuyo valor es determinado por el usuario, etc.
- Existen un montón de tecnologías utilizadas en varios campos basadas en JavaScript, algunos ejemplos son Node, Vue o React, este último creado por Facebook.

2.11.4. Node js

Node.js es un entorno de tiempo de ejecución de JavaScript, de ahí su terminación “.js”. Este entorno de tiempo es open source, es decir, de código abierto, multiplataforma y que se ejecuta del lado del servidor. (Acibeiro, 2022)

2.11.4.1. Características de Node js

- Velocidad. Node.js está construido sobre el motor de JavaScript V8 de Google Chrome, por eso su biblioteca es muy rápida en la ejecución de código.
- Sin búfer. Las aplicaciones de Node.js generan los datos en trozos (chunks), nunca los almacenan en búfer.
- Asíncrono y controlado por eventos. las APIs de la biblioteca de Node.js son asíncronas, sin bloqueo. Un servidor basado en Node.js no espera que una API devuelva datos. El servidor pasa a la siguiente API después de llamarla, y un mecanismo de notificación de eventos ayuda al servidor a obtener una respuesta de la llamada a la API anterior.

2.11.5. Framework Vue js

Creado por Evan You, Vue.js es un marco JavaScript progresivo de código abierto para crear interfaces de usuario (UI) y aplicaciones de una sola página; se conoce comúnmente como Vue. Este marco utiliza "alto desacoplamiento", lo que permite a los desarrolladores crear progresivamente interfaces de usuario (UI). (Mamani, 2019)

Vue (pronunciado vju, como view) es un framework progresivo para construir interfaces de usuario. A diferencia de otros frameworks monolíticos, Vue está diseñado desde cero para ser utilizado incrementalmente. La librería central está enfocada solo en la capa de visualización, y es fácil de utilizar e integrar con otras librerías o proyectos existentes. Por otro lado, Vue también es perfectamente capaz de impulsar sofisticadas Single-Page Applications cuando se utiliza en combinación con herramientas modernas y librerías de apoyo. (Vuejs, 2022)

2.11.6. Framework Bootstrap

Bootstrap es un framework CSS desarrollado por Twitter en 2010, para estandarizar las herramientas de la compañía.

Inicialmente, se llamó Twitter Blueprint y, un poco más tarde, en 2011, se transformó en código abierto y su nombre cambió para Bootstrap. Desde entonces fue actualizado varias veces y ya se encuentra en la versión 5.

El framework combina CSS y JavaScript para estilizar los elementos de una página HTML. Permite mucho más que, simplemente, cambiar el color de los botones y los enlaces.

Esta es una herramienta que proporciona interactividad en la página, por lo que ofrece una serie de componentes que facilitan la comunicación con el usuario, como menús de navegación, controles de página, barras de progreso y más.

Además de todas las características que ofrece el framework, su principal objetivo es permitir la construcción de sitios web responsive para dispositivos móviles.

Esto significa que las páginas están diseñadas para funcionar en desktop, tablets y smartphones, de una manera muy simple y organizada. (Rockcontent, 2020)

2.11.7. Servidor Web Apache

Apache es un servidor web de código abierto, multiplataforma y gratuito.

Este web server es uno de los más utilizados en el mundo, actualmente el 43% de los sitios webs funcionan con él.

Este servidor web desarrollado por Apache Software Foundation lleva en funcionamiento desde 1995.

El nombre Apache se refiere a la tribu de los nativos americanos, conocidos por su gran resistencia en el combate y por sus estrategias de guerra.

Se ha vuelto muy popular entre los programadores debido a su modularidad y actualización constante por parte de la comunidad.

Los servidores Apache podemos encontrarlos en la mayoría de hosting a nivel mundial, funcionando sin problema con paneles como Panel, Plesk, VestaCP, etc.

Una de las principales características de Apache es el uso del archivo .htaccess, muy utilizado entre todos los usuarios web. (Webempresa, 2021)

2.11.7.1. Características

La función esencial del servidor Apache es servir las webs alojadas en el servidor a los diversos navegadores como Chrome, Firefox, Safari.

Apache consigue que la comunicación entre el servidor web y el cliente web (usuario que solicita la información) sea fluida y constante.

Haciendo que cuando un usuario haga una petición HTTP a través de navegador para entrar a una web o URL específica, Apache devuelva la información solicitada a través del protocolo HTTP.

En Apache podemos aplicar una alta personalización a través de su sistema modular, de forma que podemos activar o desactivar diversas funcionalidades a través de los módulos de Apache.

Estos módulos de Apache hay que usarlos con cautela ya que pueden afectar a la seguridad y funcionalidades del servidor web.

2.11.7.2. Ventajas y desventajas

Ventajas

- Cuenta con una comunidad grande de desarrolladores en todo el mundo, que contribuyen a mejorar el software, ya que el código fuente original está disponible de forma gratuita para su visualización y colaboración.
- Compatible con webs que usen WordPress y la mayor parte de los CMS más populares del mercado.
- Estructura constituida por módulos.
- Es multiplataforma. Puede ser usado en servidores Windows y Linux lo que amplía sus posibilidades de uso.
- Es de código abierto y gratis.
- Alto nivel de seguridad debido a sus actualizaciones constantes.
- Podrás usar .htaccess lo que permite trabajar de forma más sencilla con los principales CMS.

Desventajas

- El rendimiento puede verse afectado en páginas con alto tráfico, sobre todo a partir de la 10 k conexiones.
- Si no se hace un buen uso de los módulos se pueden generar brechas de seguridad.

2.11.8. Gestor de Base de Datos PostgreSQL

PostgreSQL es uno de los sistemas de gestión de bases de datos relacionales más usados en la actualidad. Está orientado a objetos, es multisistema, por tanto, puedes usarlo en cualquier sistema operativo y bajo licencia libre.

Está desarrollado desde 1996 por una comunidad de desarrolladores (no una empresa ni fundación) a partir del SGBD POSTGRES, que surgió (como muchas otras tecnologías) a partir de un proyecto de investigación militar estadounidense (DARPA, ARO) con participación civil. (González, OpenWebinars, 2018)

2.11.8.1. Características

- El lenguaje SQL que usa es muy próximo al estándar ISO/IEC, gracias a lo que resulta relativamente sencillo portar consultas y scripts de otros sistemas de bases de datos, y así aprender fácilmente las variantes de este lenguaje.
- Cumple con ACID, es decir provee atomicidad, consistencia, aislamiento y durabilidad para sus operaciones.
- Permite crear esquemas, tablas heredadas y triggers orientados a eventos que no poseen otros motores.
- Permite definir procedimientos, no solo en PostgreSQL, sino también en otros muchos lenguajes como Pearl, TCL o Python. Incluso si lenguaje que queramos usar no está soportado, podemos definirlo con nuevas extensiones.
- Si necesitamos algún tipo de dato que no esté soportado de serie, también podemos definirlos.
- Podemos extender la funcionalidad con extensiones, provistas por la propia PostgreSQL, por terceros o incluso programando por nuestra cuenta.
- Tiene un soporte nativo de replicación maestro-esclavo, pero también es posible añadir otros tipos a través de productos de terceros, libres o de pago.
- También provee una excelente escalabilidad vertical.

2.11.8.2. Ventajas y desventajas

Ventajas

- Open source.
- Altamente ampliable.
- En gran parte conforme con el estándar SQL.
- Permite procesar tipos complejos de datos (p. ej., datos geográficos).
- Búsqueda de texto completo flexible.
- Pueden crearse funciones propias, triggers, tipos de datos, etc.
- Amplia compatibilidad con varios lenguajes (Python, Java, Perl, PHP, C, C++, etc.).
- Soporta JSON.
- Multiplataforma.

Desventajas

- No está disponible por defecto en todos los hostings.
- La documentación es mejorable y está solo disponible en inglés.
- La velocidad de lectura es menor que en otros gestores.

A blue decorative frame surrounds the text. It consists of a top horizontal bar, a left vertical bar, a right vertical bar, and a bottom horizontal bar. The top and bottom bars are slightly slanted towards the center. The left and right bars are straight vertical lines.

CAPÍTULO III

MARCO

APLICATIVO

3.1. INTRODUCCIÓN

En este capítulo se verá el desarrollo, análisis y diseño del sistema web, bajo la metodología UWE.

Para la implementación es necesario estar en constante comunicación con los usuarios que participaran en el uso y administración del sistema, con el fin de identificar las tareas o escenarios para elaborar los distintos diagramas de la metodología UWE así también el diseño de las interfaces que se mostraran con los datos obtenidos de la base de datos, y las herramientas de programación e implementación del sistema.

3.2. ANÁLISIS DE REQUERIMIENTO

La correcta obtención de requerimientos de la organización puede llegar a describir claramente en forma consistente para el diseño del sistema web, por esta razón se toman en cuenta, para diferenciar los conceptos de requisitos:

3.2.1. Requerimientos funcionales

Los requerimientos funcionales son todas las características que necesita el sistema.

Tabla 2.24

Requisitos Funcionales Web de Administración

N.	TIPO	OBSERVACIONES
R1-1	Ingreso al sistema	Acceder al sistema por tipos de usuarios tales como: Administrador, Vendedor, editor.
R1-2	Administración de usuarios	Crear, listar, actualizar y eliminar usuarios.
R1-3	Administrar roles y permisos	Permite la asignación de los roles a usuarios.
R1-4	Administración de las categorías	Crear, listar, actualizar y eliminar registros de categorías. Las revistas deben tener asignada una categoría.
R1-5	Administración de autores	Crear, listar, actualizar y eliminar registros de autores. Los autores deben estar asignados a una o varias

		revistas.
R1-6	Administración de las revistas y publicación	Crear, listar, actualizar y eliminar registros de revistas. Las revistas deben estar asignados a un autor o varios y categoría.
R1-7	Administración de proveedores	Crear, listar, actualizar y eliminar registros de los proveedores.
R1-8	Registro de compras de las revistas físicas	Crear, listar y ver registros de las compras. Registrar el Tipo de comprobante de compra.
R1-9	Registro de clientes	Crear, listar, actualizar y eliminar registros de los clientes.
R1-10	Registro de ventas de revistas físicas	Crear, listar y ver registros de las ventas con su detalle de ventas.
R1-11	Registro de ventas en línea de revistas	Crear, listar y ver registros de las ventas en línea con su detalle de ventas.
R1-12	Reportes	Generar reportes a diferentes niveles, reportes instantáneos, reportes entre rangos de fechas, reportes semanales y mensuales, según el usuario requiera.

Nota: Se muestra los requerimientos funcionales el sistema de administración

Tabla 2.25

Requisitos Funcionales Web Público

N.	TIPO	OBSERVACIONES
R2-1	Inicio de sesión del usuario	Acceder al sistema con el rol de lector
R2-2	registro de usuarios	Se registran nuevos usuarios que tendrán una cuenta con la que tener acceso a revistas gratuitas y comprar una copia digital.
R2-3	Perfil de usuario	Permite modificar sus datos de su cuenta.

R2-4	Cambiar o actualizar contraseña	Permite modificar su contraseña de su cuenta
R2-5	Favoritos del usuario	Lista los registros de revistas que sean de su interés.
R2-6	Compras del usuario	Lista los registros de revistas que compro el usuario, además de visualizar el archivo comprado.
R2-7	Carrito de compras del usuario	Agrega las revistas que comprara el usuario.
R2-8	inicio de sitio web	Se muestra todas las revistas disponibles al público.
R2-9	revistas	Se muestra todas las revistas disponibles al público y además de los usuarios que cuentan con una cuenta.
R2-10	Detalle de las revistas	Se muestra toda la información de las revistas.

Nota: Se muestra los requerimientos funcionales de la web pública

3.2.2. Requisitos no funcionales

Los requisitos no funcionales son aquellas que no forman parte de la funcionalidad en si del sistema, pero son requisitos que forman parte general del sistema.

Tabla 2.26

Requisitos no Funcionales

ROL	FUNCIÓN
R3-1	El sistema web debe funcionar correctamente en cualquier navegador web, como ser: Internet Explorer, Mozilla Firefox, Chrome, etc.
R3-2	El sistema web debe estar implementada en un servidor web y base de datos, para estar disponible las 24 horas del día.

R3-3	Dar soporte y mantenimiento periódico del sistema web, para resguardar el buen funcionamiento del mismo.
R3-4	Almacenar copias de seguridad de manera sencilla de la base de datos.
R3-5	El sistema web debe contar con conexión a internet.

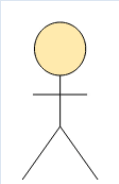
Nota: Se muestra los requerimientos no funcionales del sistema.

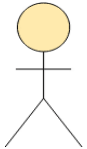
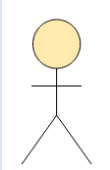
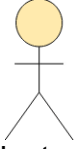
3.2.3. Definición de actores

En primera se identificara los actores que interactuaran con el sistema y posteriormente los roles y permisos de los mismos. En la siguiente tabla se muestran la lista de actores y a su lado una descripción de sus actividades relacionadas con el sistema web.

Tabla 2.27

Lista de Actores

ACTOR	FUNCIONES
 <p>Administrador</p>	<ul style="list-style-type: none"> ➤ Acceso total en el sistema Web. ➤ Registra a los nuevos usuarios. ➤ Asigna los roles y permisos de los usuarios. ➤ Registra las categorías de las revistas. ➤ Registra los autores de las revistas. ➤ Registra las revistas y su publicación. ➤ Registra a los proveedores. ➤ Registra las compras de las revistas. ➤ Registra a los clientes. ➤ Registra cada una de las ventas de las revistas físicas. ➤ Da de alta cada una de las ventas en línea. ➤ Elabora informes a diferentes niveles.

 Editor	<ul style="list-style-type: none"> ➤ Realiza actualización de información de las revistas. ➤ Registra los autores de las revistas. ➤ Registra las revistas y su publicación.
 vendedor	<ul style="list-style-type: none"> ➤ Registra nuevos clientes. ➤ Registra las ventas que realice de las revistas físicas. ➤ Da de alta cada una de las ventas en línea.
 lector	<ul style="list-style-type: none"> ➤ Visualiza la información de las revistas publicadas. ➤ Visualiza su información personal. ➤ Adiciona las revistas de interés en sus favoritos. ➤ Adiciona las revistas de interés que va a comprar en el carrito de compras. ➤ Visualiza el listado de sus compras.

Nota: Se muestra los principales actores que usaran el sistema.

3.2.4. Descripción de funciones

A continuación, detallamos un resumen de las funciones que tiene cada actor o cada usuario en el sistema web.

Tabla 2.28

Descripción de funciones

ACTOR	FUNCIONES
Administrador	La función del administrador del sistema, es estar ligado directamente con el encargado de área, esta tiene todas las funciones del director de unidad.
editor	Su función es registrar las revistas y las publicaciones de las además de la edición de información de las mismas.
vendedor	Las funciones del encargado de ventas son: registrar a sus

	clientes, registras las ventas de las revistas físicas.
lector	Su función, una vez registrado en el sistema, puede visualizar las revistas gratuitas disponibles, ver detalles y si desea compras una copia digital, agrega en el carrito de compras y posteriormente visualiza sus compras en su cuenta de usuario. Adiciona a sus favoritos las revistas de su interés.

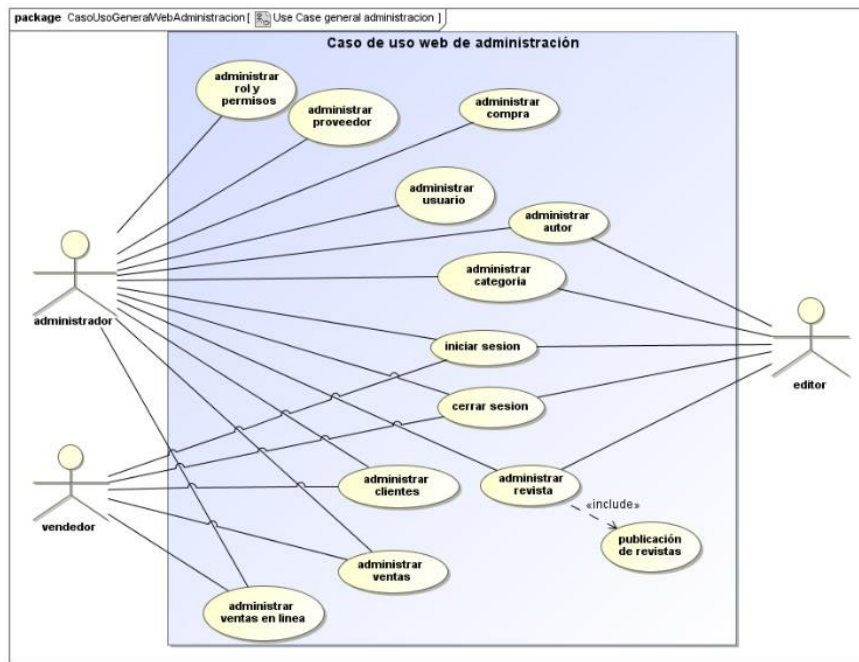
Nota: Se muestra las funciones de actores que usaran el sistema.

3.3. DISEÑO DEL SISTEMA

3.3.1. Modelo de casos de uso

Figura 3.1

Caso de Uso Web de Administración



Nota: Se muestra lo actores que usaran el sistema.

Tabla 2.29

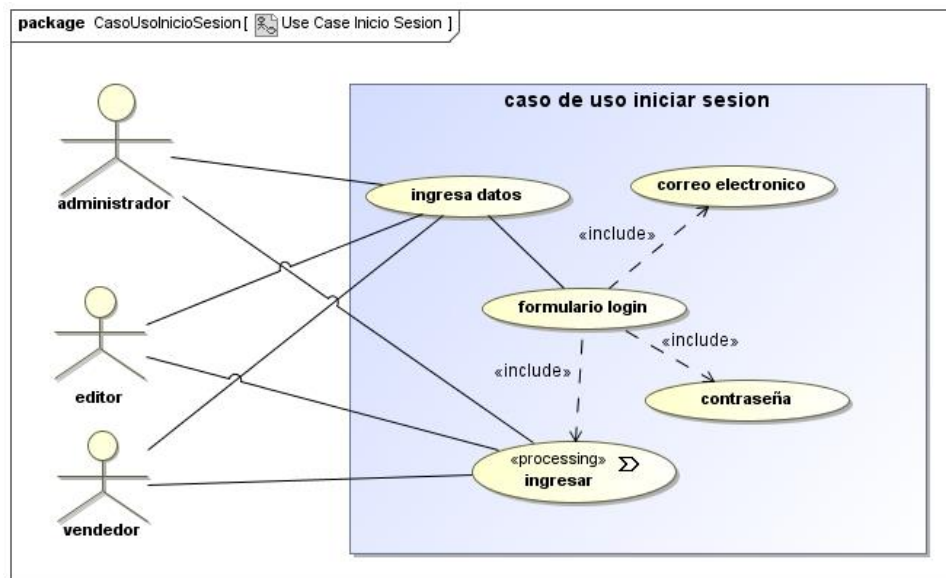
Caso de Uso general Web de Administración

Caso de uso	Caso de uso general web de administración
actores	Administrador, vendedor, editor
tipo	Primario esencial
descripción	<p>El administrador cuenta con el privilegio de registrar, editar, eliminar, cambiar estados y otros dentro del sistema de los diferentes módulos.</p> <p>El vendedor puede registrar, editar, eliminar, cambiar estados y otros dentro del sistema asignado con los privilegios que le proporciona el administrador.</p> <p>El editor puede registrar, editar, eliminar, cambiar estados y otros dentro del sistema asignado con los privilegios que le proporciona el administrador.</p>

Nota: Se muestra la descripción de los actores.

Figura 3.2

Inicio de sesión



Nota: Se muestra el caso de uso de los actores que ingresan en el sistema.

Tabla 2.30

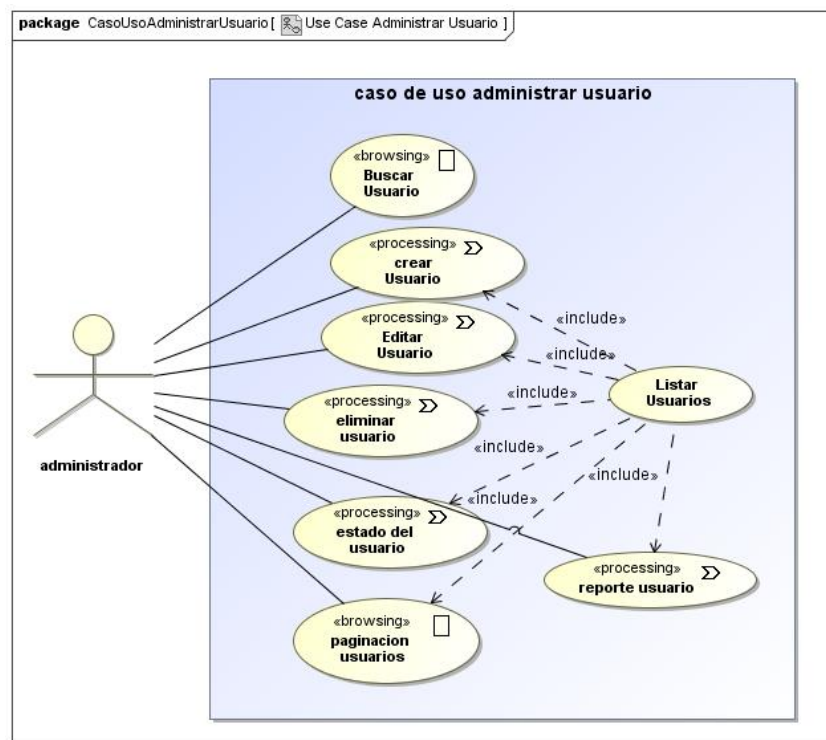
Inicio de sesión

Caso de uso	Inicio de sesión
actores	Administrador, vendedor, editor
tipo	Primario esencial
descripción	El administrador, vendedor y editor deben ingresar con su correo y contraseña al sistema.

Nota: Se muestra el caso de uso de los actores que ingresan en el sistema.

Figura 3.3

Administrar usuario



Nota: Se muestra el caso de uso del administrador que gestiona a los usuarios.

Tabla 2.31

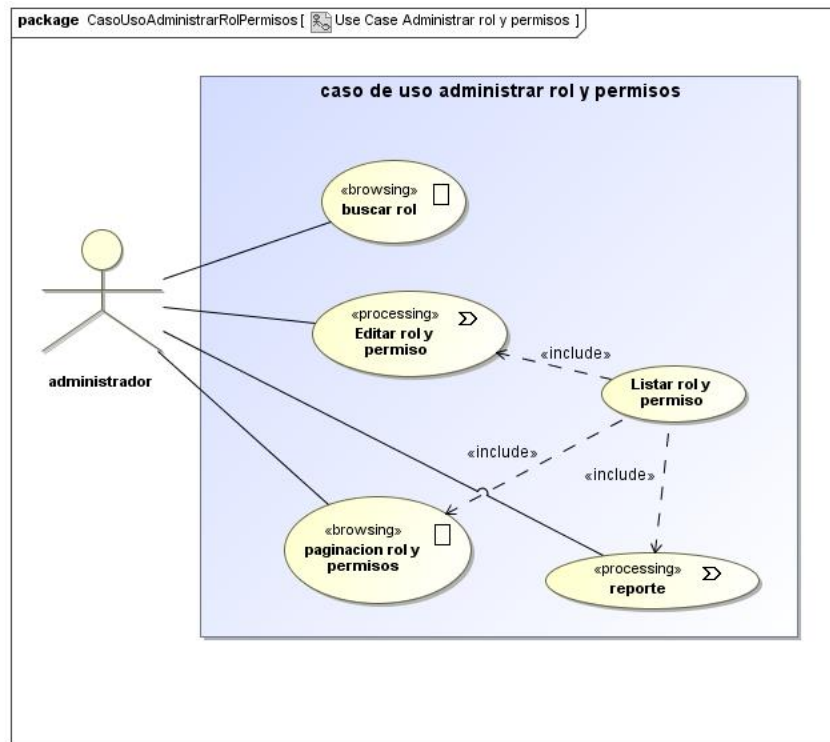
Administrar usuario

Caso de uso	Administrar usuario
actores	Administrador
tipo	Primario esencial
descripción	El administrador, registrar, buscar, eliminar, editar, cambiar el estado del usuario.

Nota: Se muestra el caso de uso del administrador que gestiona a los usuarios, crear, eliminar, editar, cambiar estado.

Figura 3.4

Administrar roles y permisos



Nota: Se muestra el caso de uso del administrador que gestiona los permisos de los usuarios.

Tabla 2.32

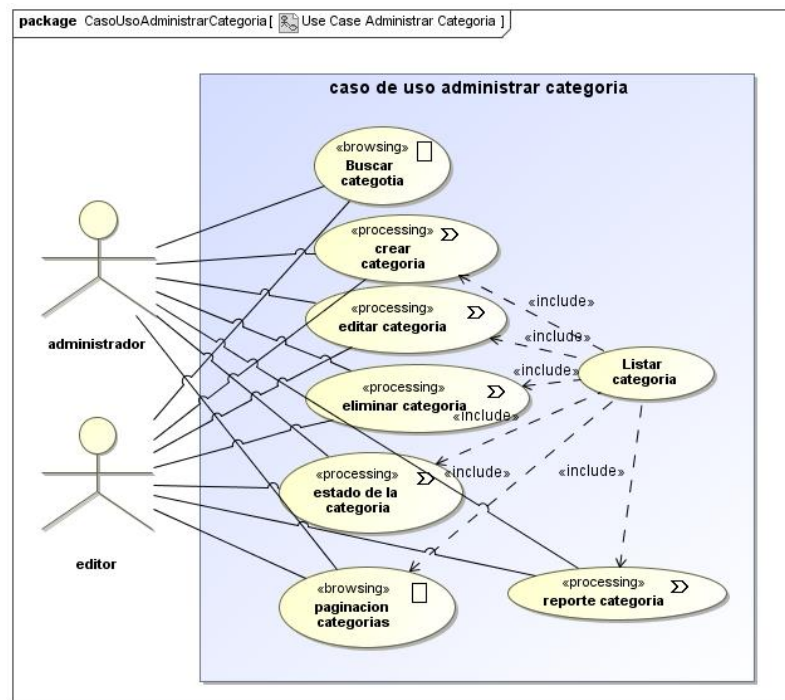
Administrar roles y permisos

Caso de uso	Administrar roles y permisos
actores	Administrador
tipo	Primario esencial
descripción	El administrador, editar el rol y permiso del usuario.

Nota: Se muestra el caso de uso del administrador que gestiona los permisos de los usuarios, editar el rol y permiso del usuario.

Figura 3.5

Administrar categoría



Nota: Se muestra el caso de uso del administrador que gestiona a las categorías, crear, eliminar, editar, cambiar estado.

Tabla 2.33

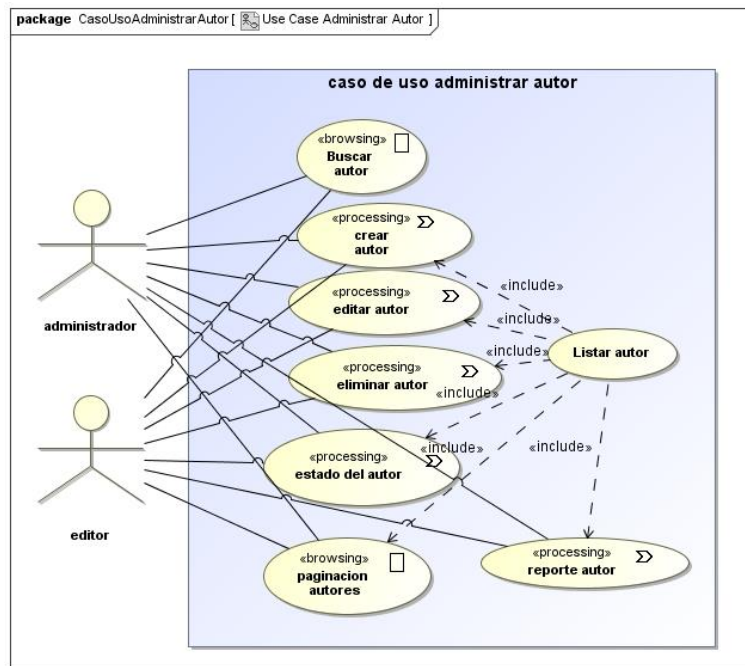
Administrar categoría

Caso de uso	Administrar categoría
actores	Administrador, editor
tipo	Primario esencial
descripción	<p>El administrador tiene los privilegios de crear, eliminar, editar, cambiar estado, buscar, emitir el reporte.</p> <p>El editor tiene los privilegios asignados por el administrador como: crear, eliminar, editar, cambiar estado, buscar, emitir el reporte.</p>

Nota: Se muestra el caso de uso del administrador y editor que gestiona a las categorías.

Figura 3.6

Administrar autor



Nota: Se muestra el caso de uso del administrador y editor que gestiona a los autores.

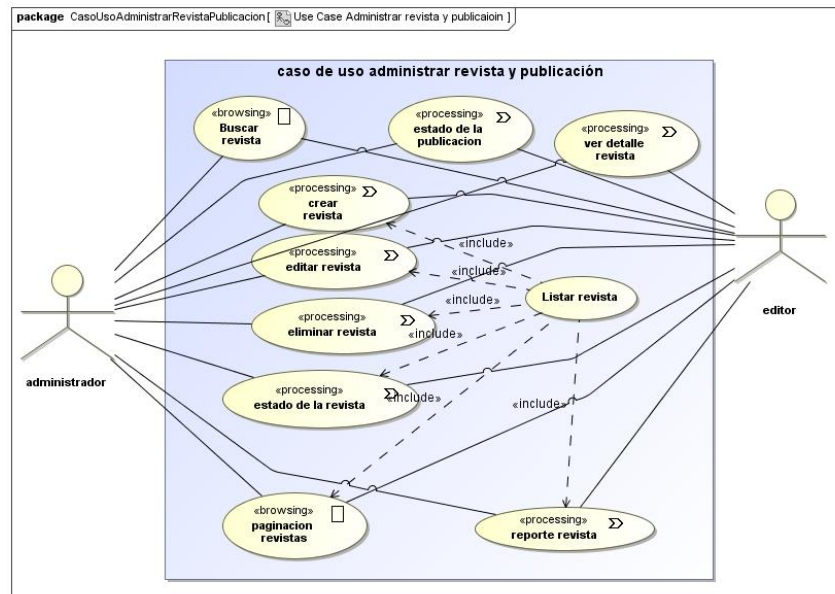
Tabla 2.34
Administrar autor

Caso de uso	Administrar autor
actores	Administrador, editor
tipo	Primario esencial
descripción	<p>El administrador tiene los privilegios de crear, eliminar, editar, cambiar estado, buscar, emitir el reporte.</p> <p>El editor tiene los privilegios asignados por el administrador como: crear, eliminar, editar, cambiar estado, buscar, emitir el reporte.</p>

Nota: Se muestra el caso de uso del administrador y editor que gestiona a los autores.

Figura 3.7

Administrar revistas y publicación



Nota: Se muestra el caso de uso del administrador y editor que gestiona a las revistas.

Tabla 2.35

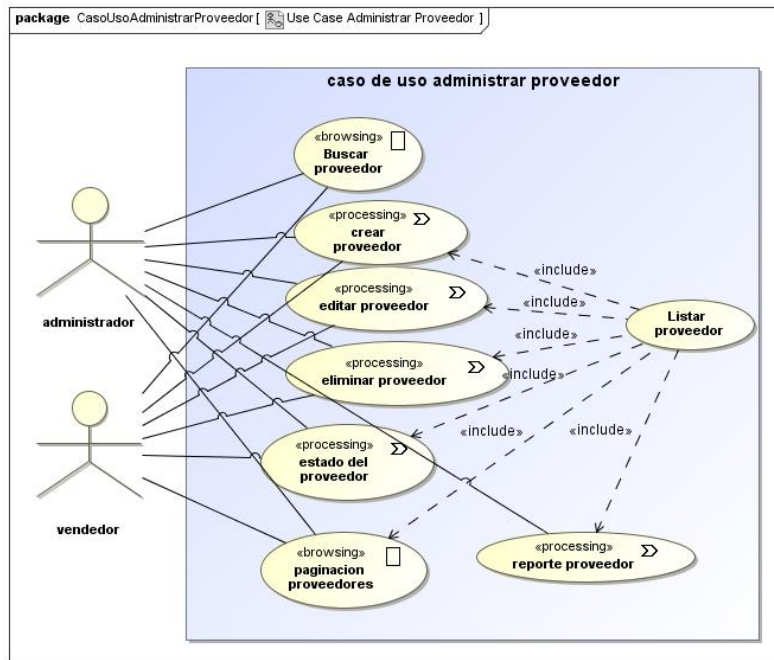
Administrar revistas y publicación

Caso de uso	Administrar revistas y publicación
actores	Administrador, editor
tipo	Primario esencial
descripción	<p>El administrador tiene los privilegios de crear, eliminar, editar, cambiar estado, buscar, emitir el reporte y publicar las revistas.</p> <p>El editor tiene los privilegios asignados por el administrador como: crear, eliminar, editar, cambiar estado, buscar, emitir el reporte.</p>

Nota: Se muestra el caso de uso del administrador y editor que gestiona a las revistas y las publicaciones.

Figura 3.8

Administrar proveedores



Nota: Se muestra el caso de uso del administrador y vendedor que gestiona a los proveedores de las revistas.

Tabla 2.36

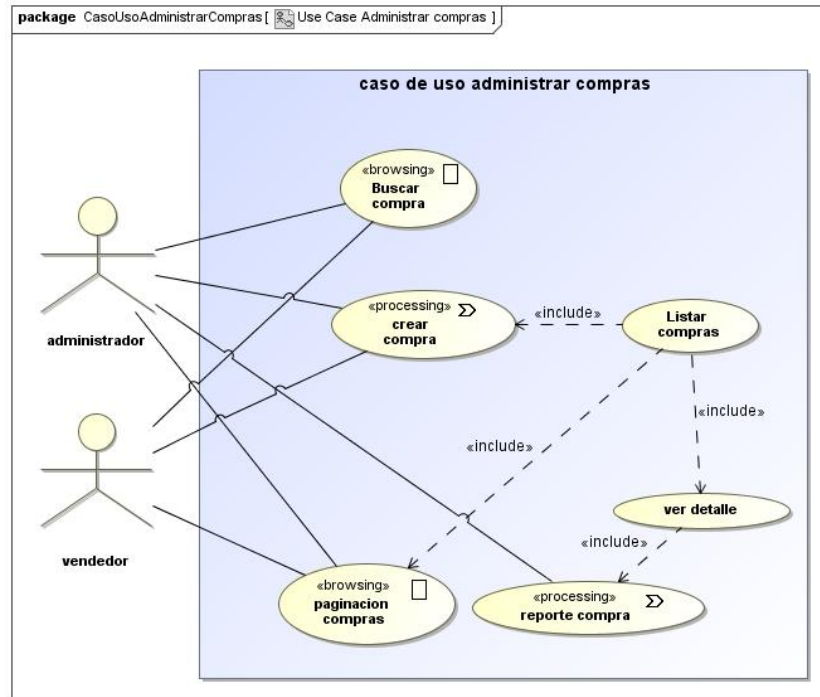
Administrar proveedores

Caso de uso	Administrar proveedores
actores	Administrador, vendedor
tipo	Primario esencial
descripción	<p>El administrador tiene los privilegios de crear, eliminar, editar, cambiar estado, buscar, emitir el reporte.</p> <p>El vendedor tiene los privilegios asignados por el administrador como: crear, eliminar, editar, cambiar estado, buscar, emitir el reporte.</p>

Nota: Se muestra el caso de uso del administrador y vendedor que gestiona a los proveedores de las revistas.

Figura 3.9

Administrar compra



Nota: Se muestra el caso de uso del administrador y vendedor que gestiona las compra de revistas.

Tabla 2.37

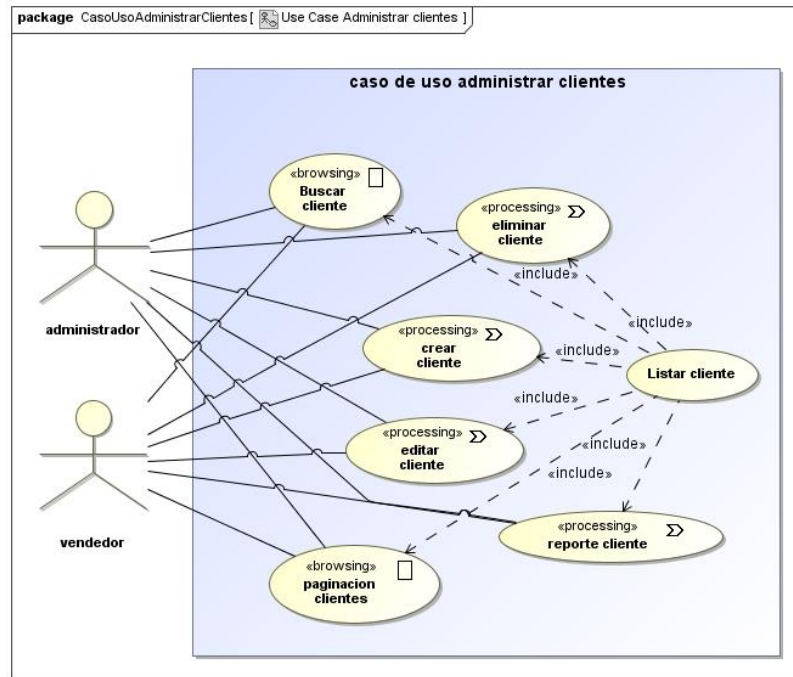
Administrar compra

Caso de uso	Administrar compra
actores	Administrador, vendedor
tipo	Primario esencial
descripción	El administrador tiene los privilegios de crear, editar, buscar, emitir el reporte. El vendedor tiene los privilegios asignados por el administrador como: crear, buscar, emitir el reporte.

Nota: Se muestra el caso de uso del administrador y vendedor que gestiona las compra de revistas.

Figura 3.10

Administrar clientes



Nota: Se muestra el caso de uso del administrador y vendedor que gestiona a los clientes.

Tabla 2.38

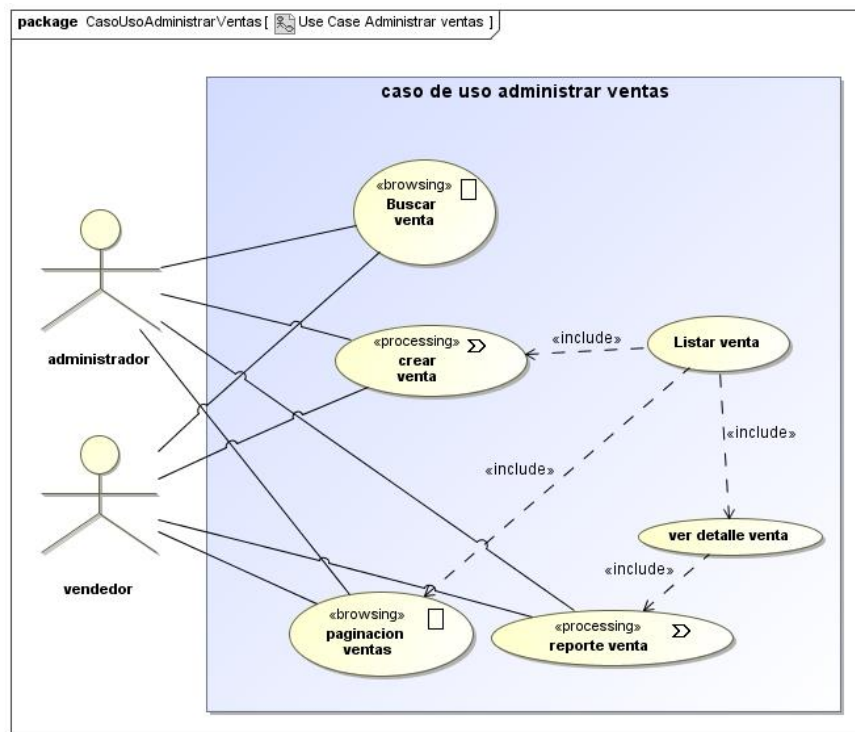
Administrar clientes

Caso de uso	Administrar clientes
actores	Administrador, vendedor
tipo	Primario esencial
descripción	El administrador tiene los privilegios de crear, eliminar, editar, buscar, emitir el reporte. El vendedor tiene los privilegios asignados por el administrador como: crear, eliminar, editar, buscar, emitir el reporte.

Nota: Se muestra el caso de uso del administrador y vendedor que gestiona a los clientes.

Figura 3.11

Administrar ventas



Nota: Se muestra el caso de uso del administrador y vendedor que gestiona las ventas de las revistas.

Tabla 2.39

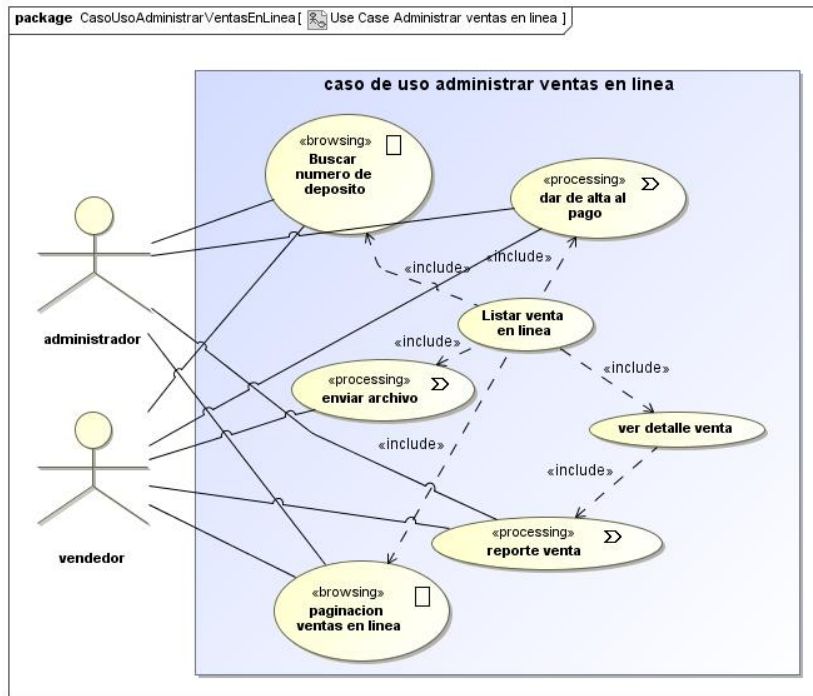
Administrar ventas

Caso de uso	Administrar ventas
actores	Administrador, vendedor
tipo	Primario esencial
descripción	<p>El administrador tiene los privilegios de crear, buscar, emitir el reporte, ver detalle de la venta.</p> <p>El vendedor tiene los privilegios asignados por el administrador como: crear, buscar, emitir el reporte, ver detalle de la venta.</p>

Nota: Se muestra el caso de uso del administrador y vendedor que gestiona las ventas de las revistas.

Figura 3.12

Administrar ventas en línea



Nota: Se muestra el caso de uso del administrador y vendedor que gestiona a las ventas en línea.

Tabla 2.40

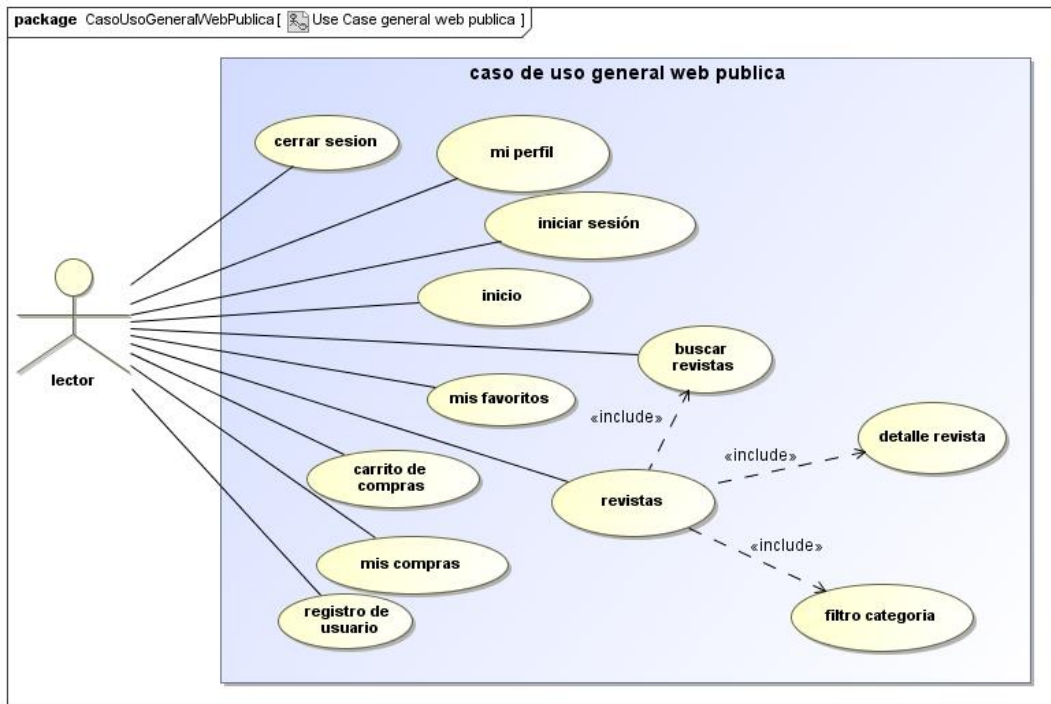
Administrar ventas en línea

Caso de uso	Administrar ventas en línea
actores	Administrador, vendedor
tipo	Primario esencial
descripción	<p>El administrador tiene los privilegios de buscar número depósito, emitir el reporte, ver detalle de la venta, enviar archivos de la compra del usuario lector, dar de alta al pago que se realizó por la compra.</p> <p>El vendedor tiene los privilegios asignados por el administrador como: buscar, emitir el reporte, ver detalle de la venta, enviar archivos de la compra del usuario lector, dar de alta al pago que se realizó por la compra.</p>

Nota: Se muestra el caso de uso del administrador y vendedor que gestiona a las ventas en línea.

Figura 3.13

Caso de uso web pública



Nota: Se muestra el caso de uso del usuario lector.

Tabla 2.41

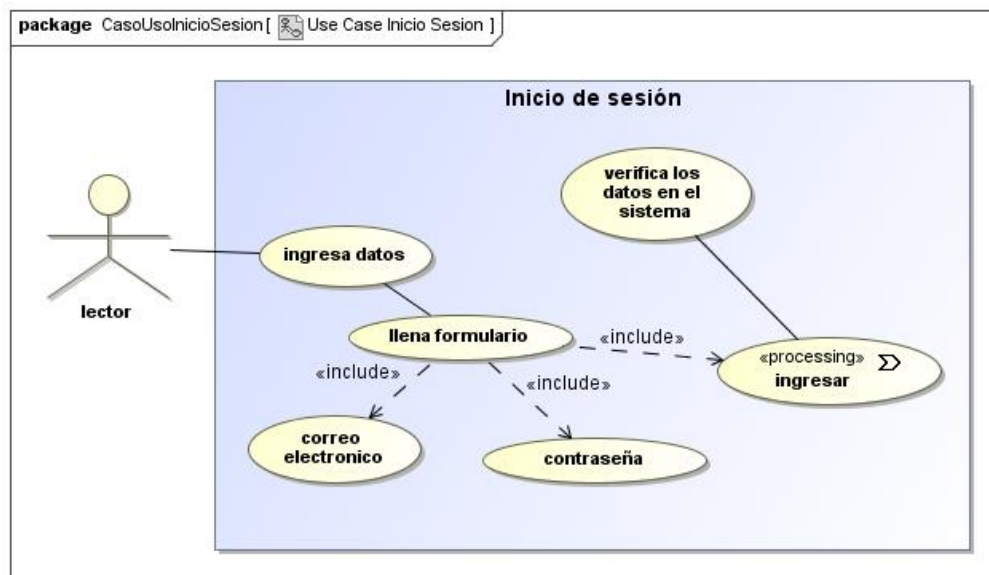
Caso de uso web pública

Caso de uso	Web pública
actores	Usuario Lector
tipo	Primario esencial
descripción	El usuario lector puede ver revistas disponibles gratuitas, filtrar por categorías, agregar en su cuenta a mis favoritos si son de su interés además de comprar una copia digital de la revista.

Nota: Se muestra el caso de uso del usuario lector y su descripción.

Figura 3.14

Inicio de sesión



Nota: Se muestra el caso de uso del usuario lector, usando la contraseña y correo electrónico para iniciar sesión.

Tabla 2.42

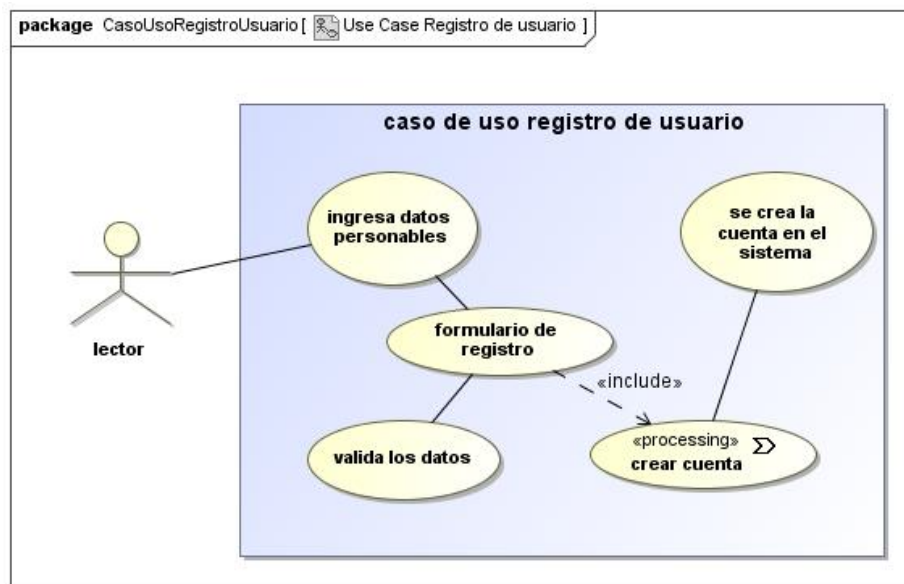
Inicio de sesión

Caso de uso	Inicio de sesión
actores	Usuario Lector
tipo	Primario esencial
descripción	El usuario lector antes de ingresar al sistema debe estar registrado, posteriormente ingresar con su correo electrónico y contraseña.

Nota: Se muestra el caso de uso del usuario lector y su descripción.

Figura 3.15

Registro de usuario



Nota: Se muestra el caso de uso del usuario lector creación de cuenta.

Tabla 2.43

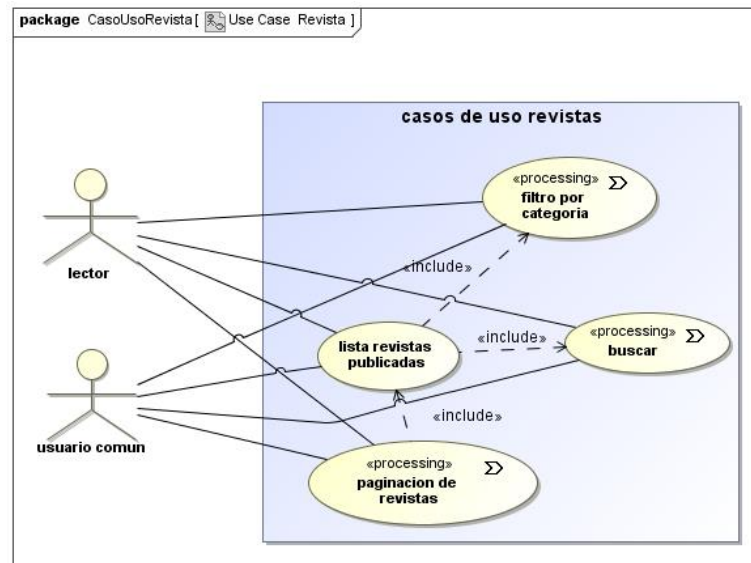
Registro de usuario

Caso de uso	Registro de usuario
actores	Usuario común
tipo	Primario esencial
descripción	El usuario común debe ingresar sus datos como: nombre apellidos, correo electrónico, y contraseña para crear su cuenta, el sistema lo registrara con el rol de lector y posteriormente ingresar con su correo electrónico y contraseña para iniciar sesión.

Nota: Se muestra el caso de uso del usuario lector creación de cuenta y la descripción.

Figura 3.16

Revistas



Nota: Se muestra el caso de uso del usuario lector y usuario común visualización de revistas.

Tabla 2.44

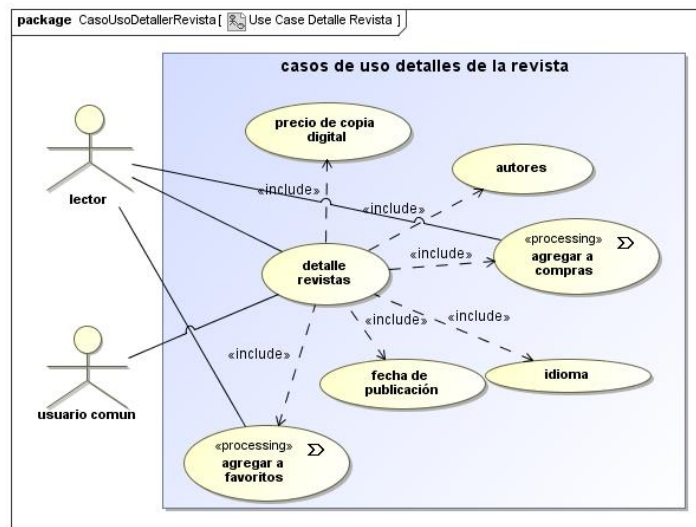
Revistas

Caso de uso	Revistas
actores	Usuario común, usuario lector
tipo	Primario esencial
descripción	El usuario común y el usuario lector pueden ver las revistas publicadas, filtrar por categorías, buscar y ver los detalles.

Nota: Se muestra el caso de uso del usuario lector y usuario común visualización de revistas.

Figura 3.17

Detalle de revista



Nota: Se muestra el caso de uso del usuario lector y usuario común visualización del detalle de las revistas.

Tabla 2.45

Detalle de revista

Caso de uso	Detalle de revista
actores	Usuario común, usuario lector
tipo	Primario esencial

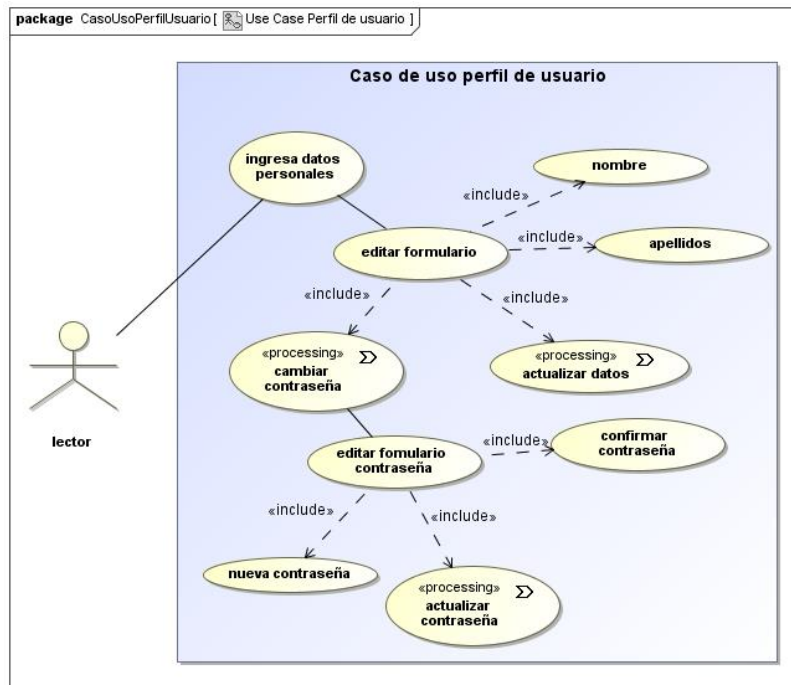
descripción

El usuario común puede acceder a los detalles de las revistas sin una cuenta de usuario y el usuario lector pueden agregar a sus favoritos las revistas de su interés y comprar.

Nota: Se muestra la descripción del usuario lector y usuario común la visualización del detalle de las revistas.

Figura 3.18

Perfil del usuario



Nota: Se muestra el caso de uso del usuario lector su perfil con sus datos a editar y cambiar contraseña.

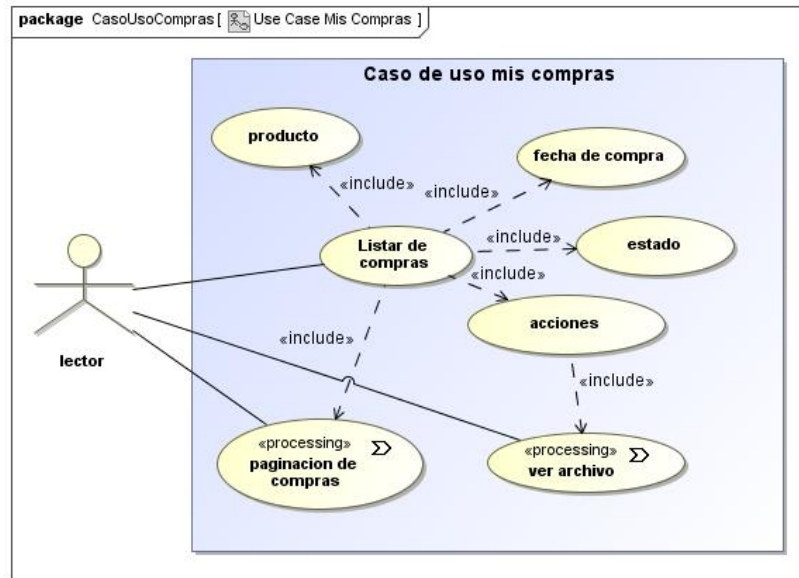
Tabla 2.46

Perfil del usuario

Caso de uso	Perfil del usuario
actores	usuario lector
tipo	Primario esencial
descripción	El usuario lector ingresando a su cuenta puede editar sus datos personales y cambiar su contraseña por seguridad.

Nota: Se muestra la descripción del usuario lector y las acciones que puede realizar en su perfil.

Figura 3.19
Mis compras



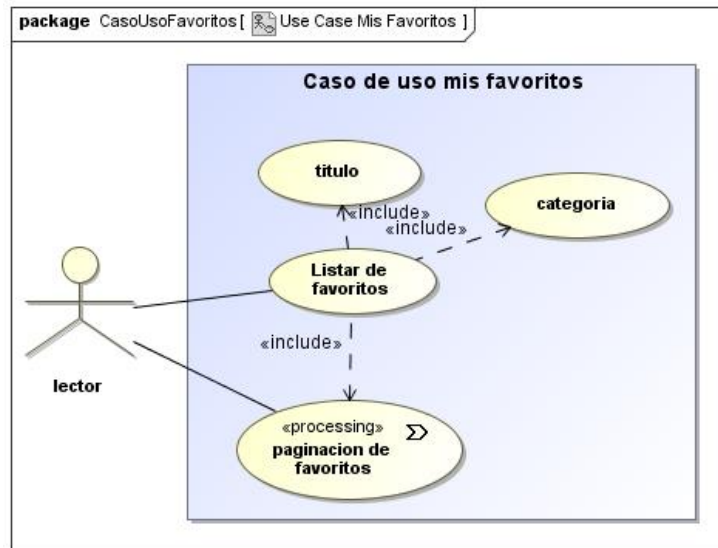
Nota: Se muestra el caso de uso del usuario lector y sus compras de revista.

Tabla 2.47
Mis compras

Caso de uso	Mis compras
actores	usuario lector
tipo	Primario esencial
descripción	El usuario lector ingresando a su cuenta ver la lista de sus compras, ver los archivos digitales.

Nota: Se muestra la descripción del usuario lector y las acciones que puede realizar en el módulo mis compras.

Figura 3.20
Mis favoritos



Nota: Se muestra el caso de uso del usuario lector y sus favoritos, las revistas que agrego en ese modulo.

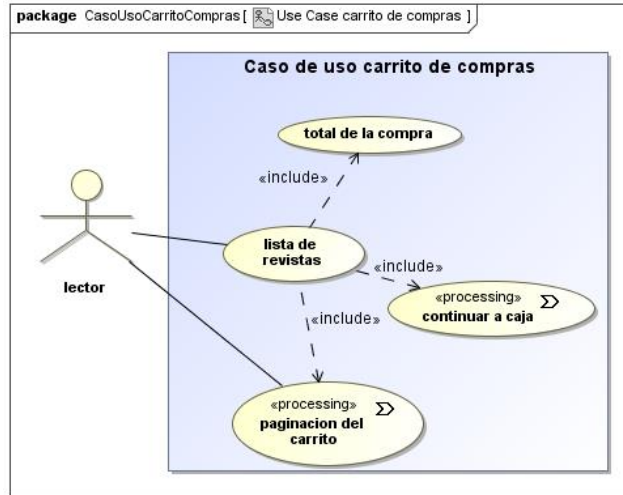
Tabla 2.48
Mis favoritos

Caso de uso	Mis favoritos
actores	usuario lector
tipo	Primario esencial
descripción	El usuario lector ingresando a su cuenta podrá ver la opción favorito con una lista de favoritos

Nota: Se muestra la descripción del usuario lector y las acciones que puede realizar en el módulo mis favoritos.

Figura 3.21

Carrito de compras



Nota: Se muestra el caso de uso del usuario lector y carrito de compras, las revistas que agrego en ese modulo.

Tabla 2.49

Carrito de compras

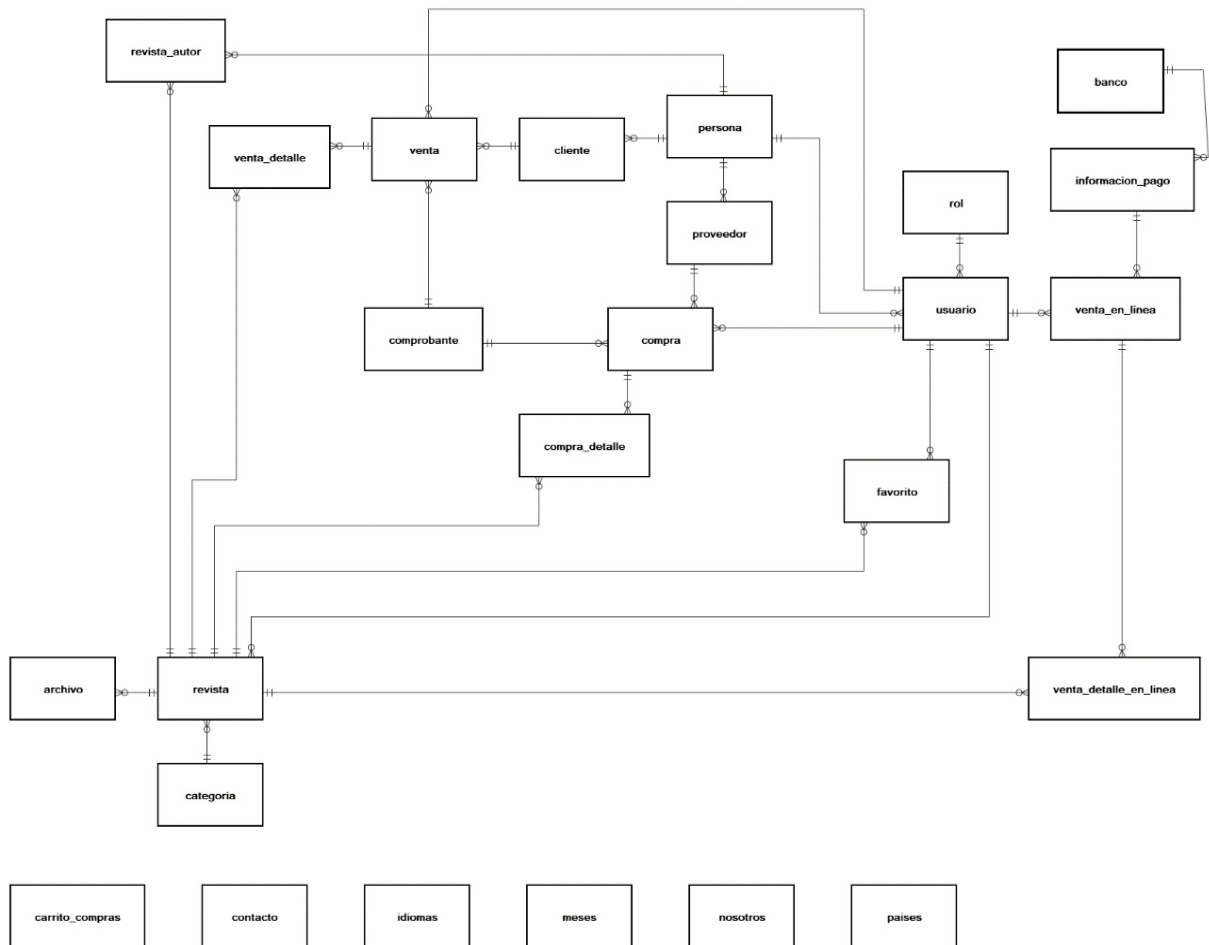
Caso de uso	Carrito de compras
actores	usuario lector
tipo	Primario esencial
descripción	El usuario lector ingresando a su cuenta podrá ver la opción carrito de compras en la parte superior de la página web y ver una lista de sus compras.

Nota: Se muestra la descripción del usuario lector y las acciones que puede realizar en el módulo mis favoritos.

3.3.2. Modelo conceptual

Figura 3.22

Modelo conceptual de la base de datos

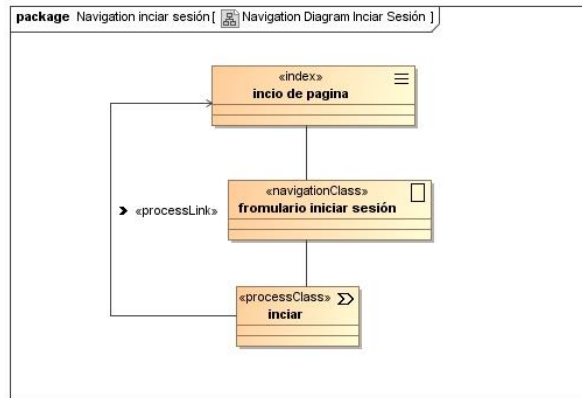


Nota: Se muestra el modelo conceptual de la base de datos.

3.3.3. Modelo de navegación

Figura 3.23

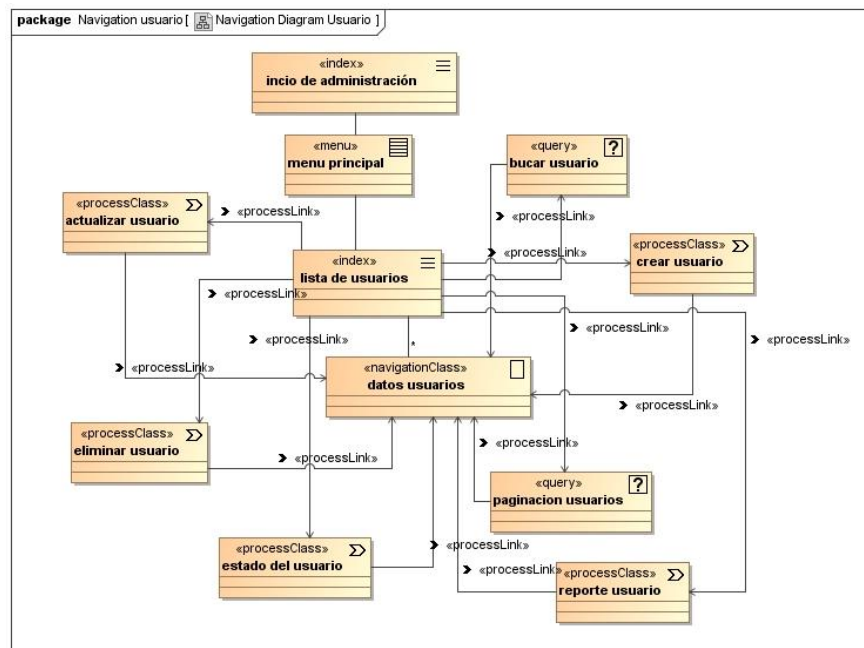
Inicio de sesión



Nota: Se muestra el modelo de navegación del inicio de sesión de los usuarios.

Figura 3.24

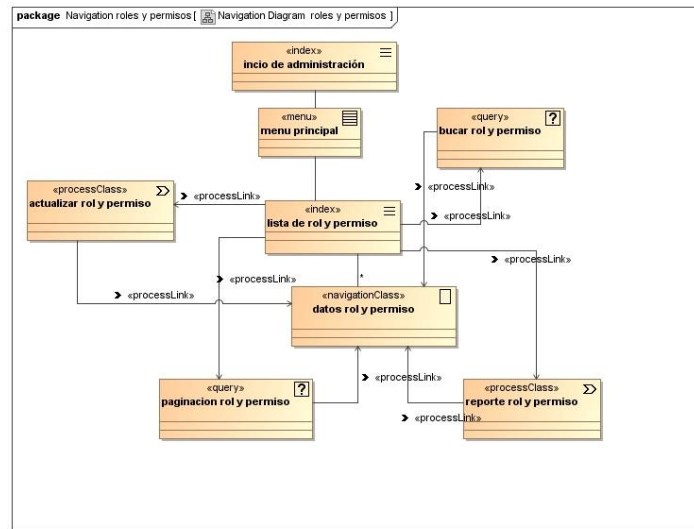
Administrar de usuario (administrador)



Nota: Se muestra el modelo de navegación de la administración de usuarios.

Figura 3.25

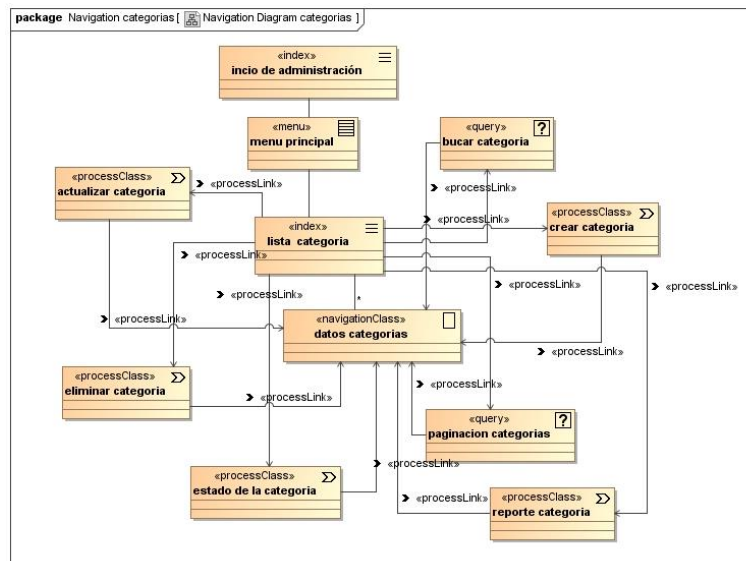
Administrar roles y permisos (administrador)



Nota: Se muestra el modelo de navegación de la administración de roles y permisos de los usuarios.

Figura 3.26

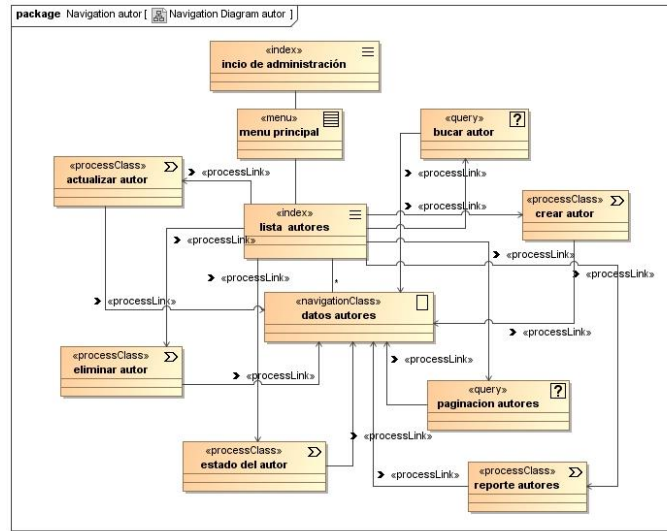
Administrar categoría (administrador, editor)



Nota: Se muestra el modelo de navegación de la administración de categorías de las revistas.

Figura 3.27

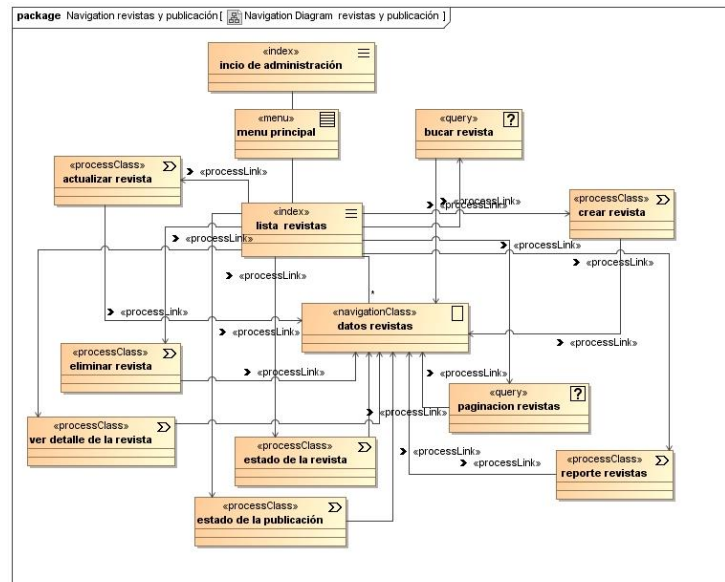
Administrar autor (administrador, editor)



Nota: Se muestra el modelo de navegación de la administración de autores de las revistas.

Figura 3.28

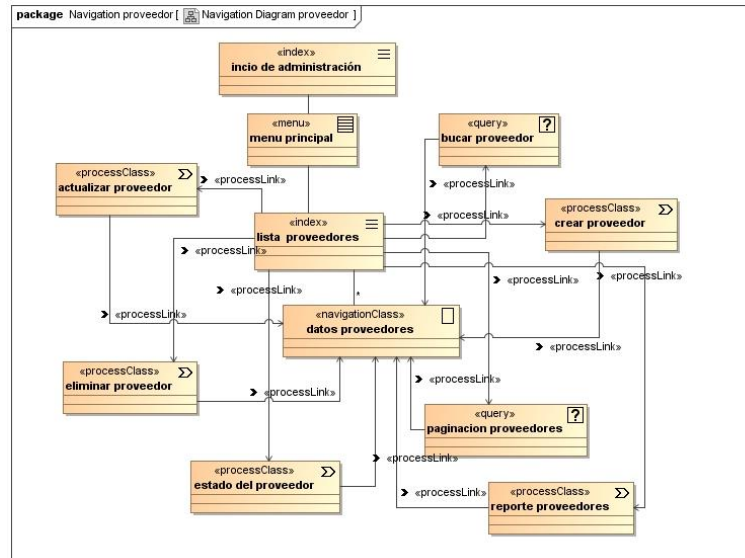
Administrar revistas y publicación (administrador, editor)



Nota: Se muestra el modelo de navegación de la administración de revistas y publicaciones.

Figura 3.29

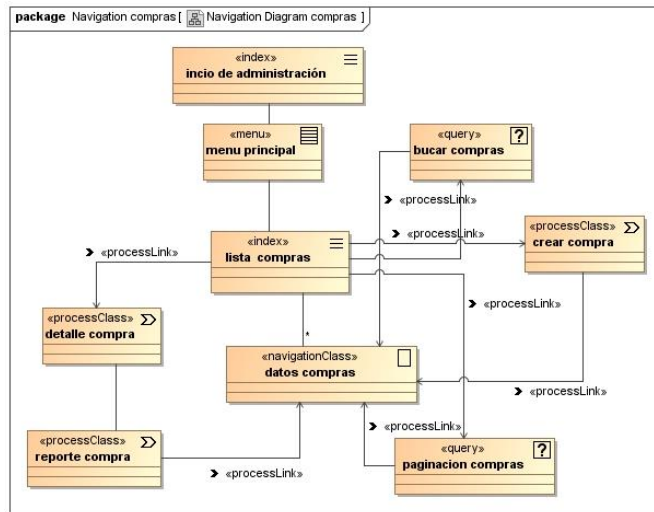
Administrar proveedores (administrador, vendedor)



Nota: Se muestra el modelo de navegación de la administración de revistas y publicaciones.

Figura 3.40

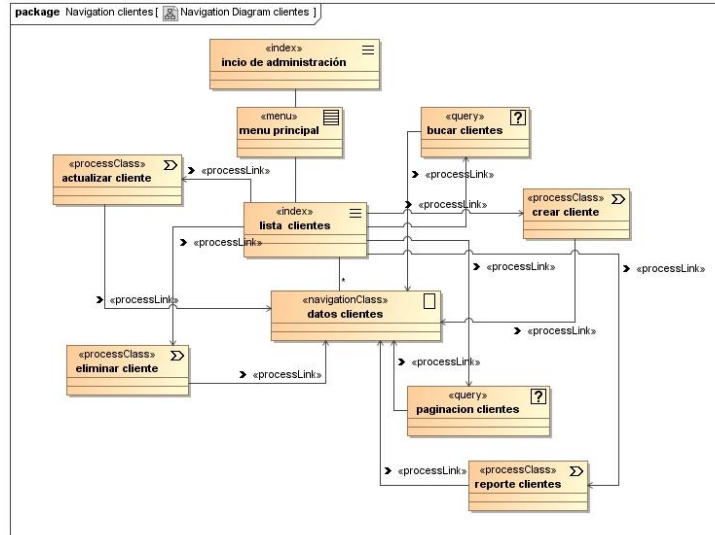
Administrar compras (administrador, vendedor)



Nota: Se muestra el modelo de navegación de la administración de compras.

Figura 3.41

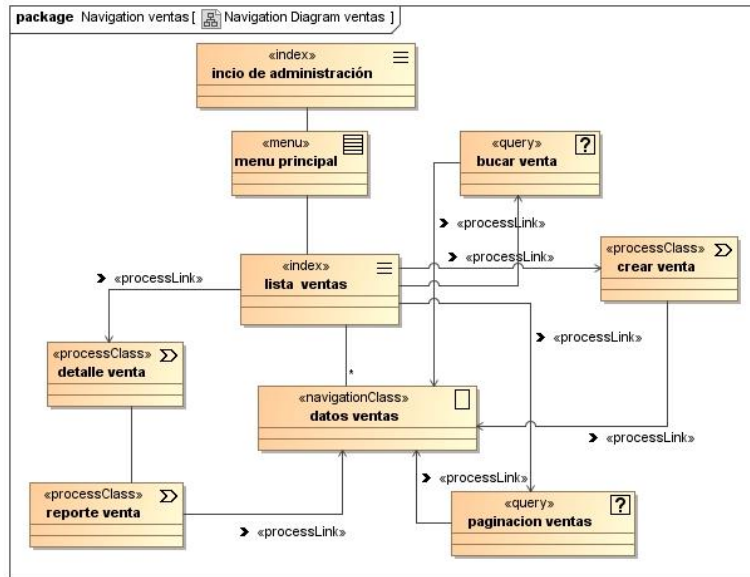
Administrar clientes (administrador, vendedor)



Nota: Se muestra el modelo de navegación de la administración de clientes.

Figura 3.42

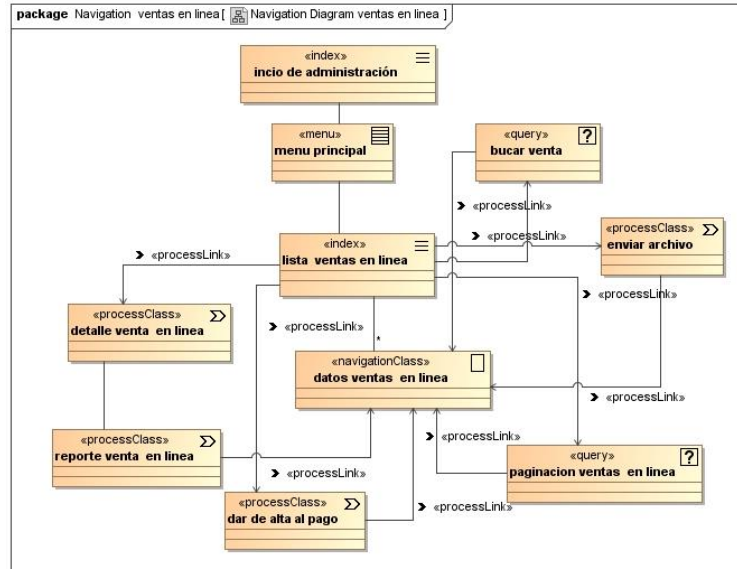
Administrar ventas (administrador, vendedor)



Nota: Se muestra el modelo de navegación de la administración de ventas.

Figura 3.43

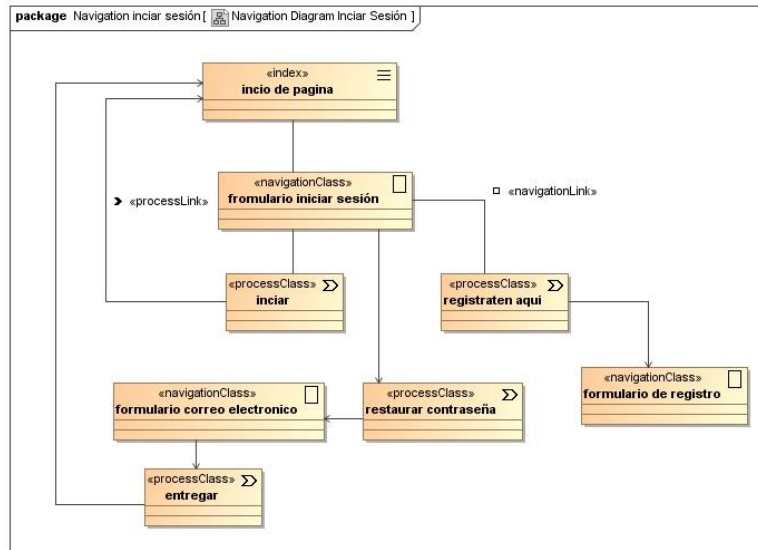
Administrar ventas en línea (administrador, vendedor)



Nota: Se muestra el modelo de navegación de la administración de ventas en línea.

Figura 3.44

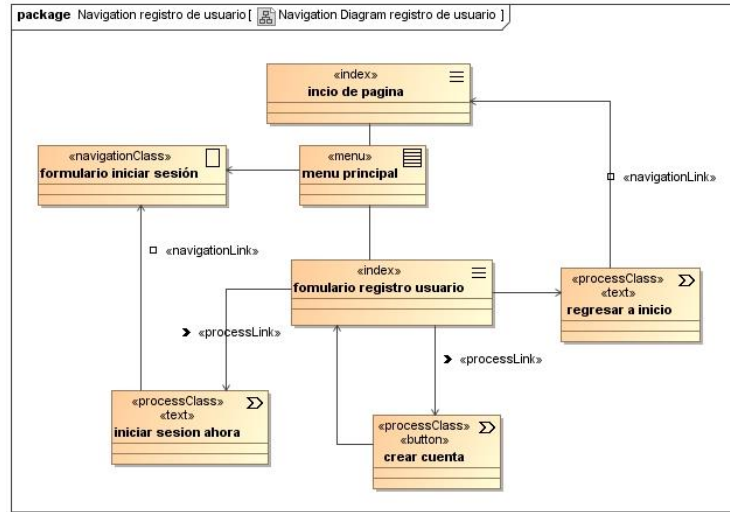
Inicio de sesión (lector)



Nota: Se muestra el modelo de navegación del usuario lector el inicio de sesión.

Figura 3.45

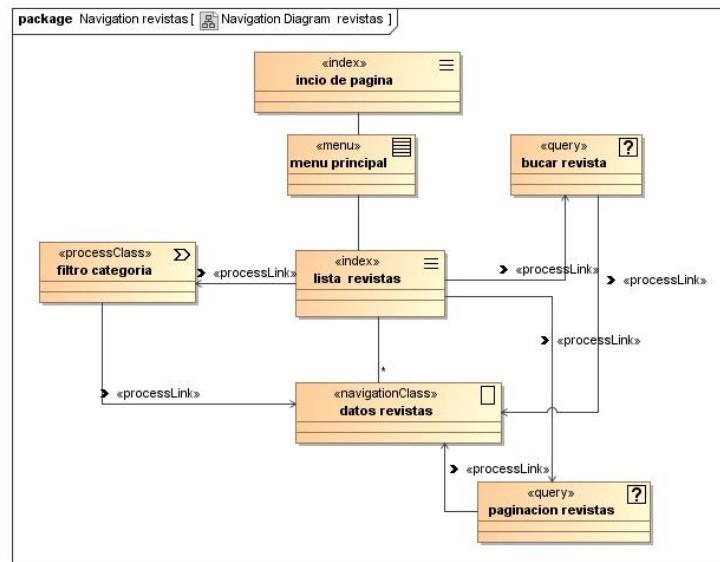
Registro de usuario (lector, usuario común)



Nota: Se muestra el modelo de navegación del usuario lector creación de cuenta.

Figura 3.46

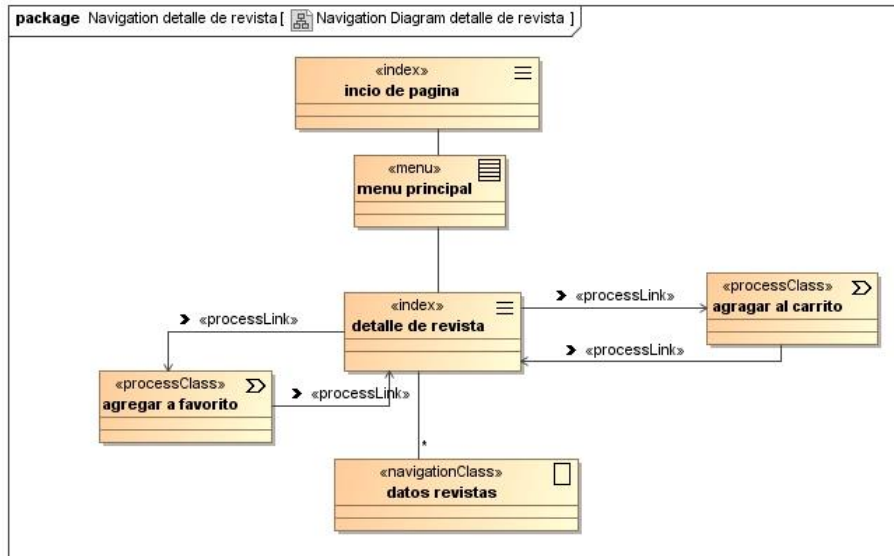
Revistas (lector, usuario común)



Nota: Se muestra el modelo de navegación del usuario lector ver revistas.

Figura 3.47

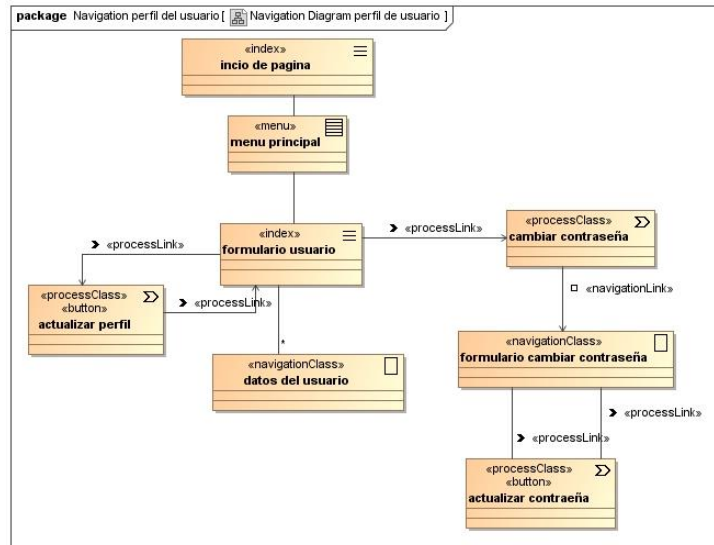
Detalle de revistas (lector, usuario común)



Nota: Se muestra el modelo de navegación del usuario lector ver detalle de la revista.

Figura 3.48

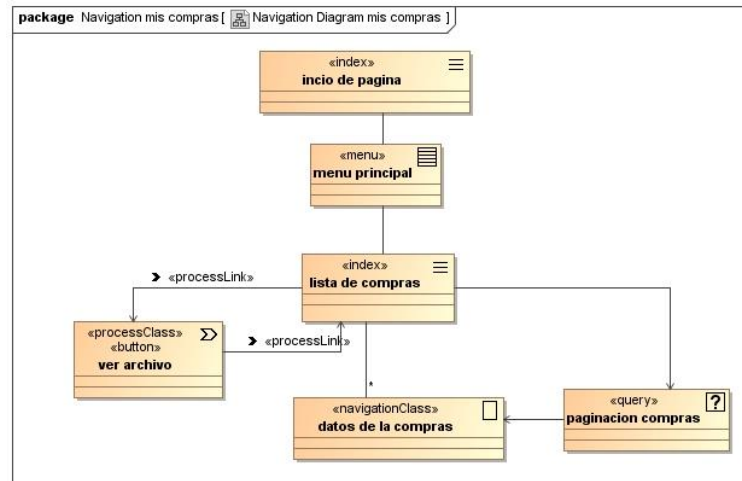
Perfil del usuario (lector)



Nota: Se muestra el modelo de navegación del usuario lector su perfil de usuario.

Figura 3.49

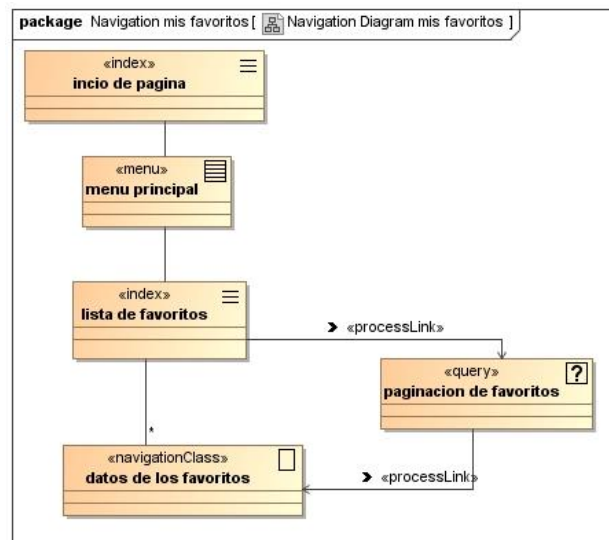
Mis compras (lector)



Nota: Se muestra el modelo de navegación del usuario lector ver sus compras, ver archivo, descargar archivo.

Figura 3.50

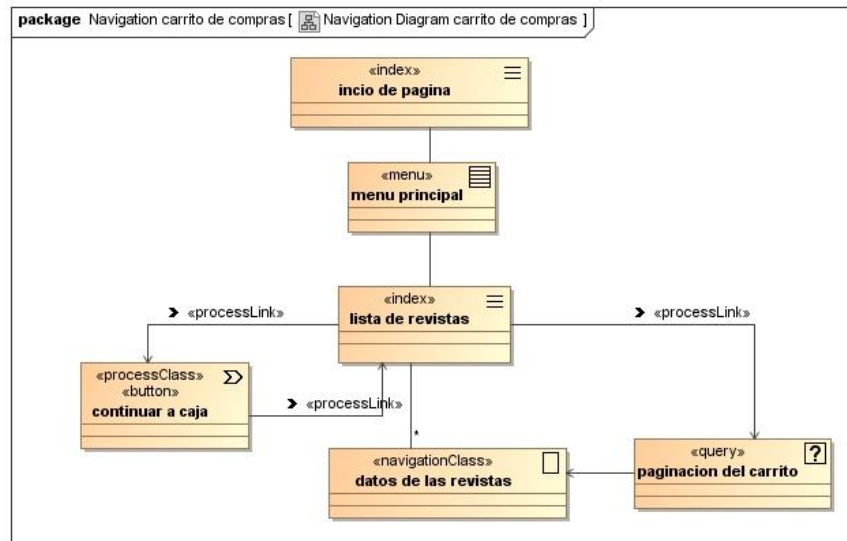
Mis favoritos (lector)



Nota: Se muestra el modelo de navegación del usuario lector ver sus favoritos.

Figura 3.51

Carrito de compras (lector)

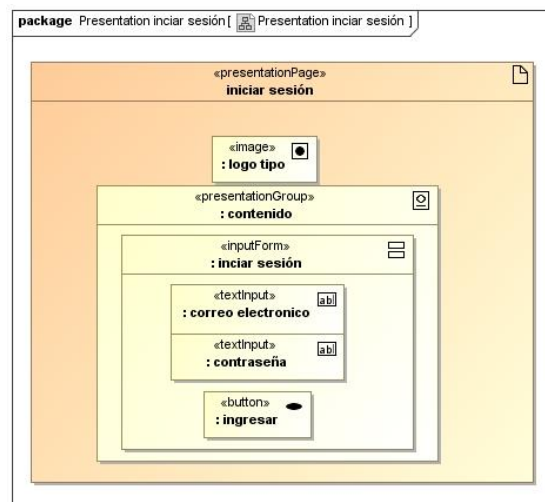


Nota: Se muestra el modelo de navegación del usuario lector ver sus compras el carrito de compras.

3.3.4. Modelo de presentación

Figura 3.52

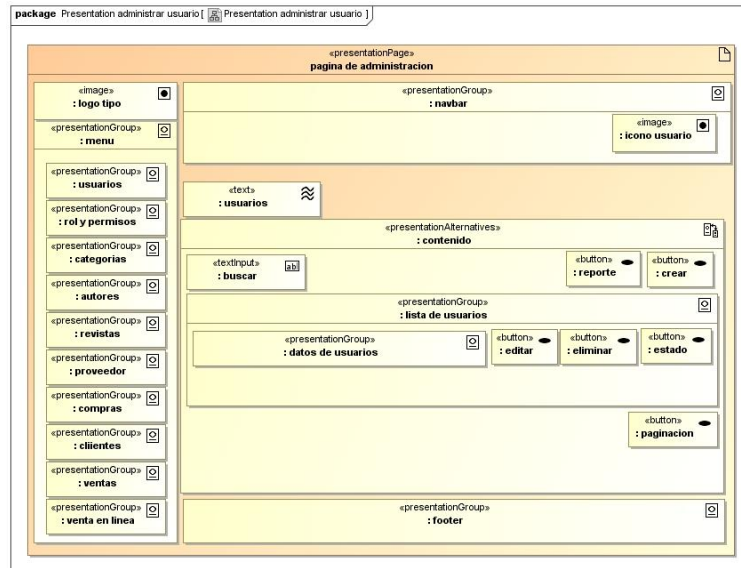
Inicio de sesión



Nota: Se muestra el modelo de presentación inicio de sesión de los usuarios.

Figura 3.53

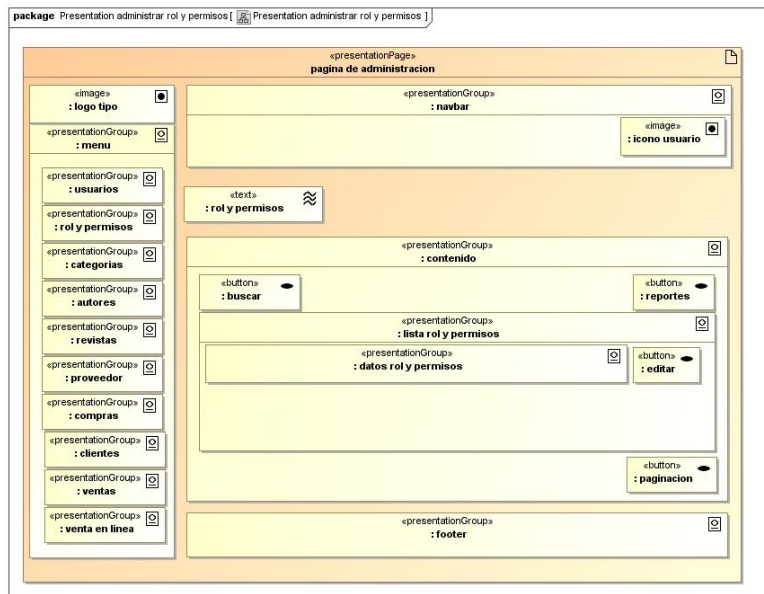
Administrar de usuario (administrador)



Nota: Se muestra el modelo de presentación administrar usuarios.

Figura 3.54

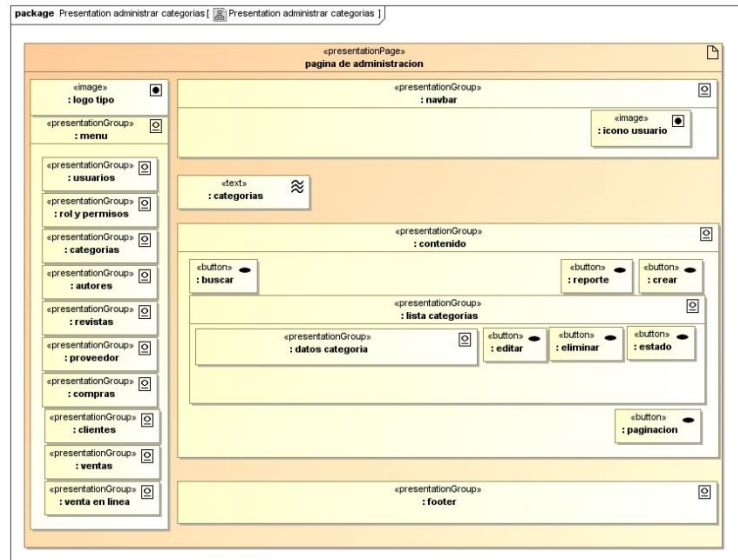
Administrar roles y permisos (administrador)



Nota: Se muestra el modelo de presentación administrar roles y permisos.

Figura 3.55

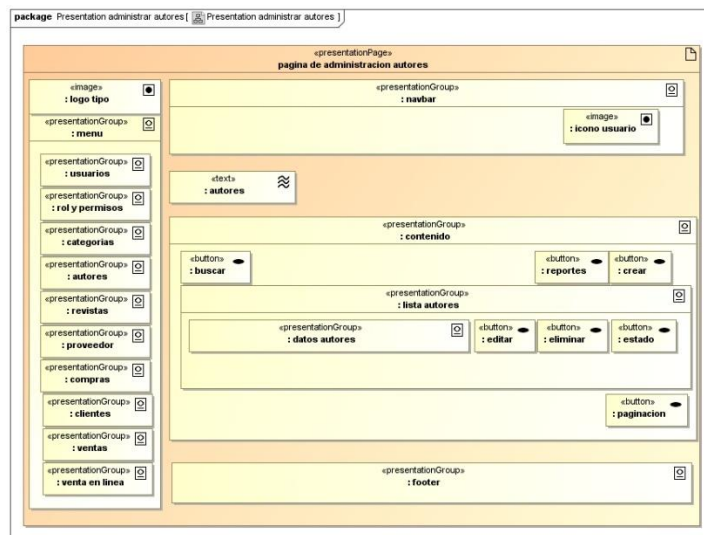
Administrar categoría (administrador, editor)



Nota: Se muestra el modelo de presentación administrar categoría.

Figura 3.56

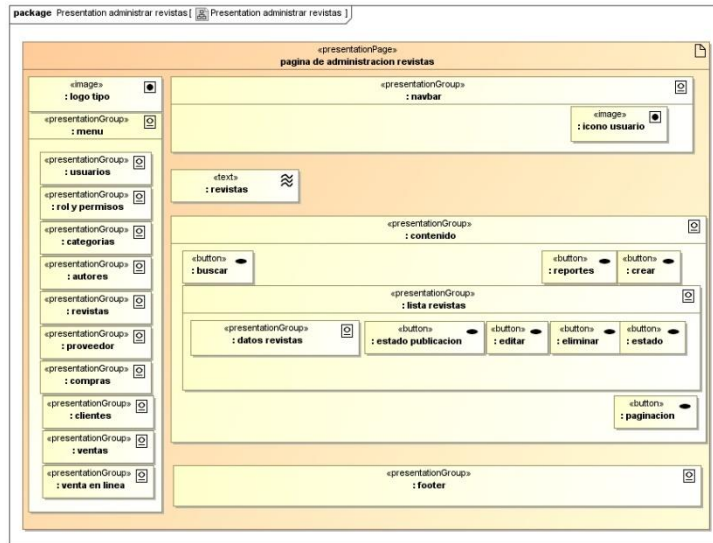
Administrar autor (administrador, editor)



Nota: Se muestra el modelo de presentación administrar autor.

Figura 3.57

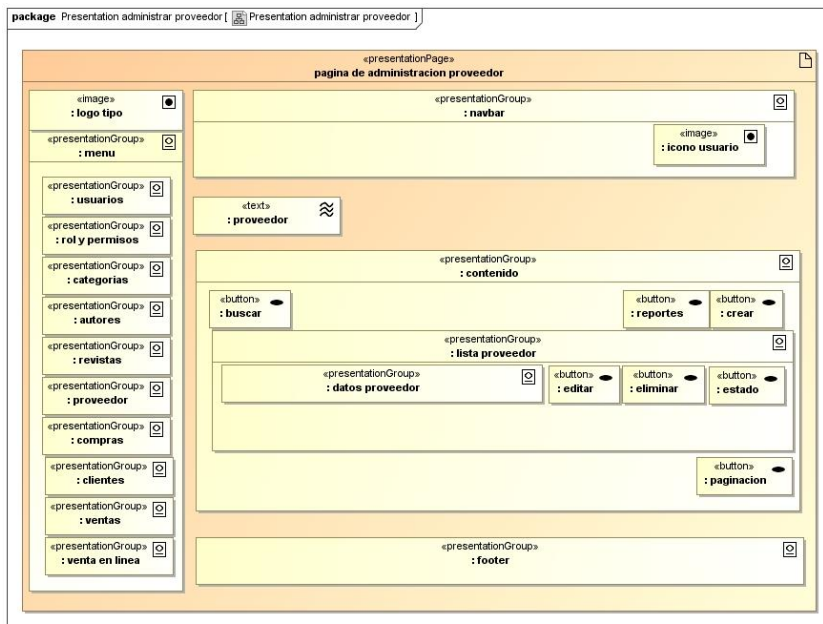
Administrar revistas y publicación (administrador, editor)



Nota: Se muestra el modelo de presentación administrar revistas y publicaciones.

Figura 3.58

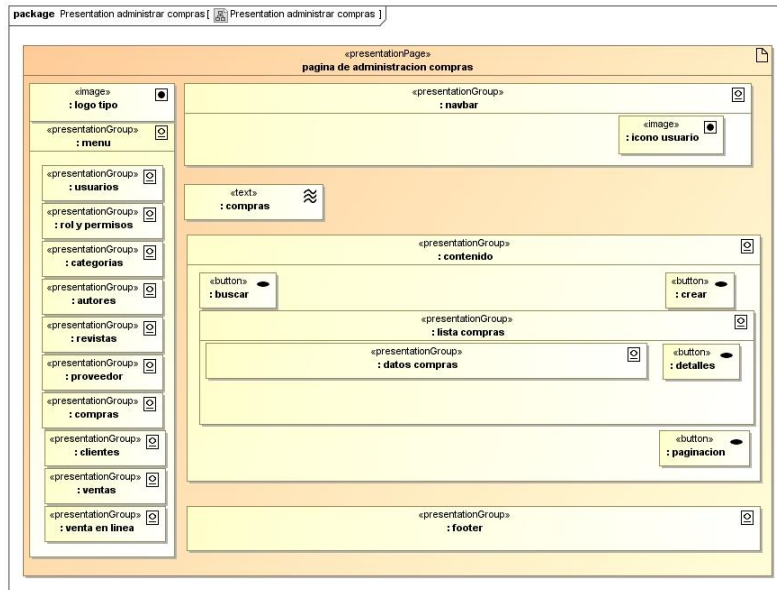
Administrar proveedores (administrador, vendedor)



Nota: Se muestra el modelo de presentación administrar proveedores.

Figura 3.59

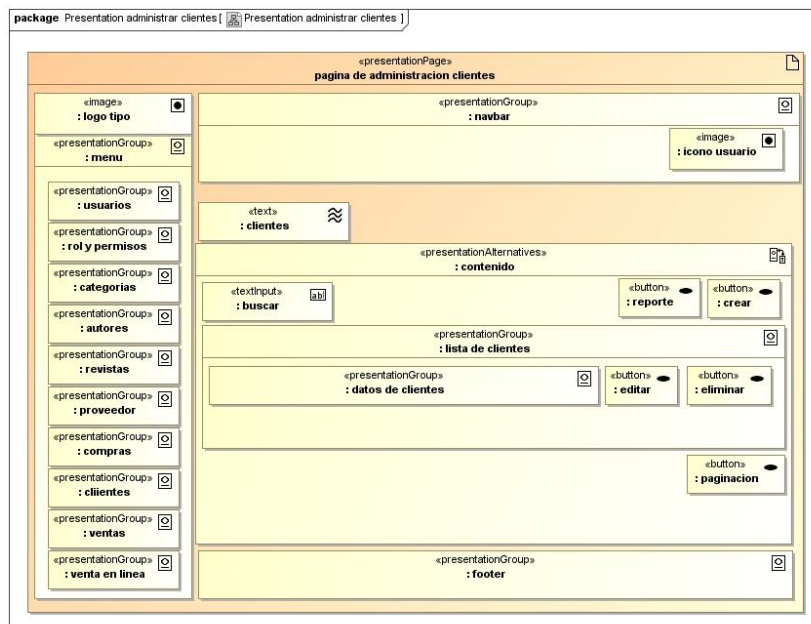
Administrar compra (administrador, vendedor)



Nota: Se muestra el modelo de presentación administrar compras.

Figura 3.60

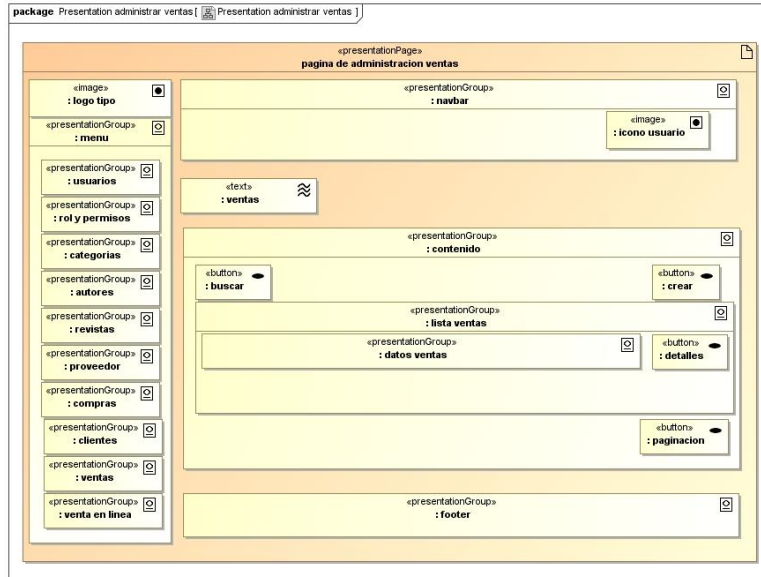
Administrar clientes (administrador, vendedor)



Nota: Se muestra el modelo de presentación administrar clientes.

Figura 3.61

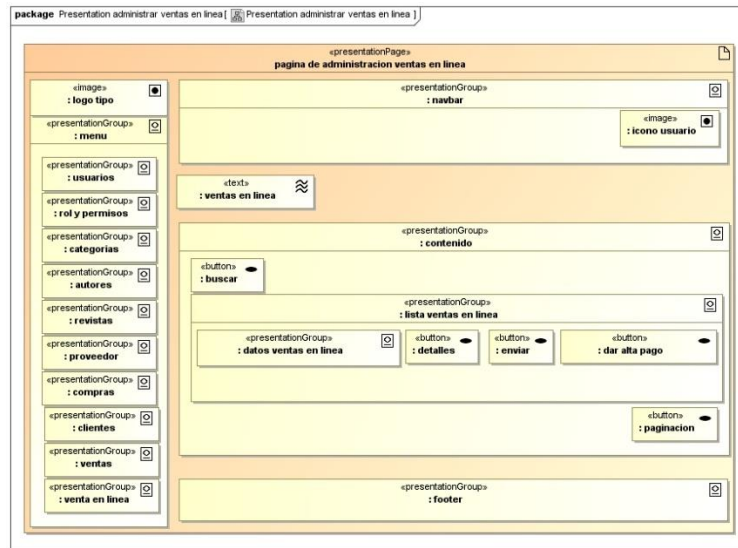
Administrar ventas (administrador, vendedor)



Nota: Se muestra el modelo de presentación administrar ventas.

Figura 3.62

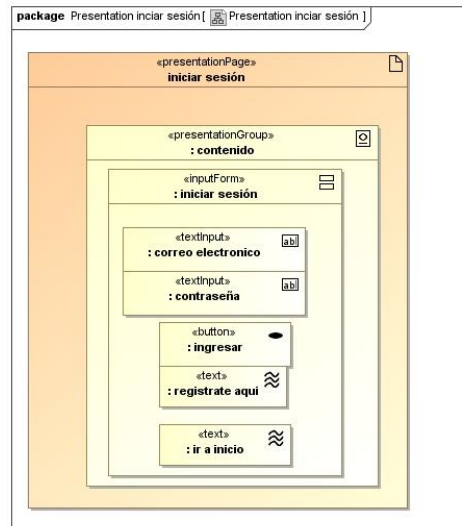
Administrar ventas en línea (administrador, vendedor)



Nota: Se muestra el modelo de presentación administrar ventas en línea.

Figura 3.63

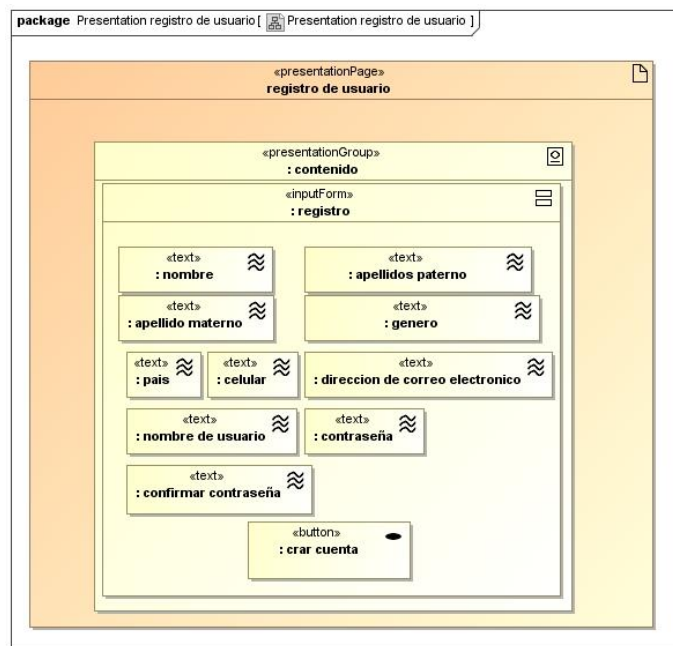
Inicio de sesión (lector)



Nota: Se muestra el modelo de presentación inicio de sesión del lector.

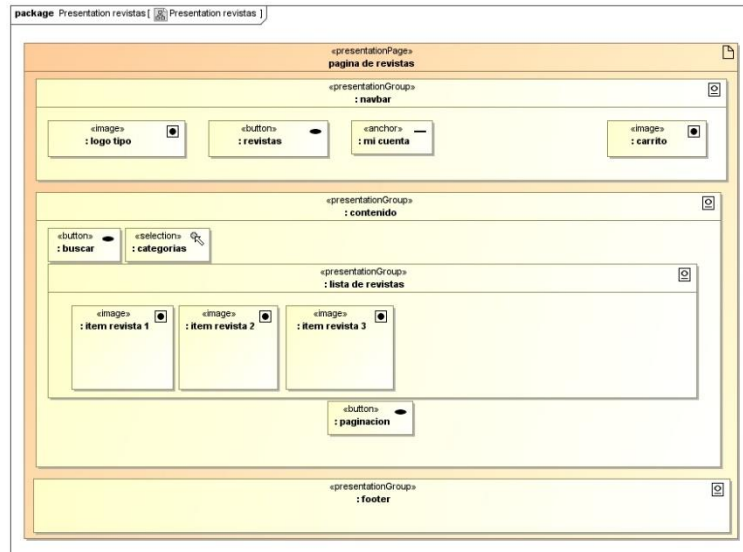
Figura 3.64

Registro de usuario (usuario común)



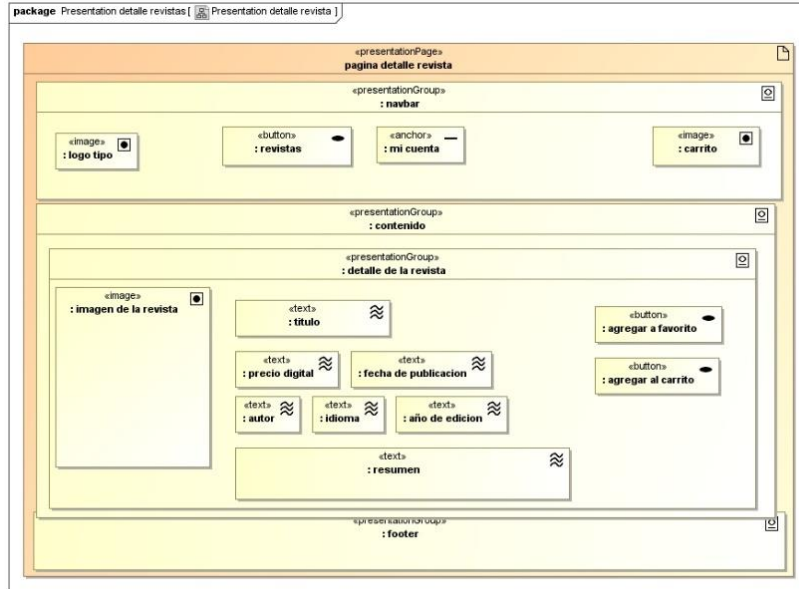
Nota: Se muestra el modelo de presentación registro de un nuevo usuario.

Figura 3.65
Revistas (lector)



Nota: Se muestra el modelo de presentación ver revista, buscar y filtrar.

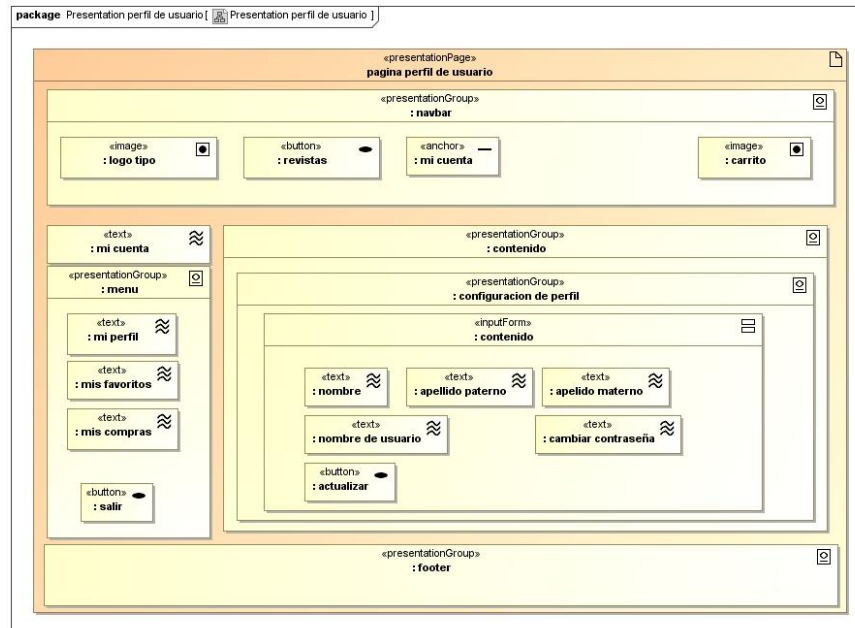
Figura 3.66
Detalle de revistas (lector)



Nota: Se muestra el modelo de presentación detalle de las revistas.

Figura 3.67

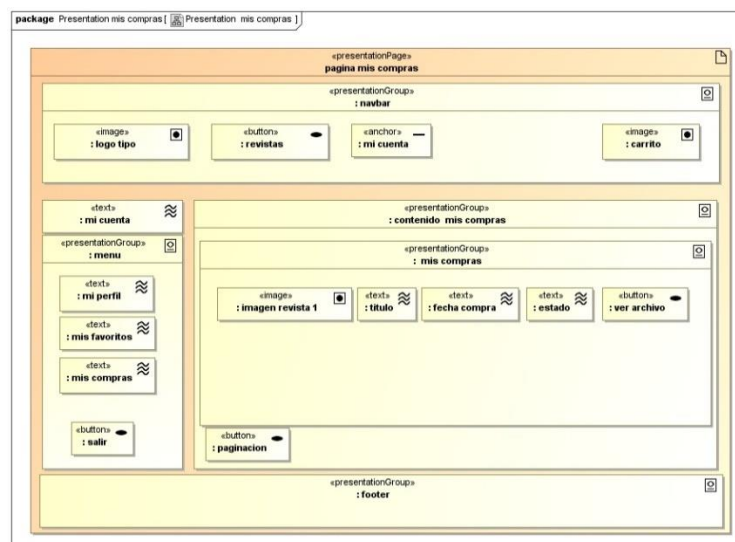
Perfil de usuario (lector)



Nota: Se muestra el modelo de presentación perfil de usuario lector.

Figura 3.68

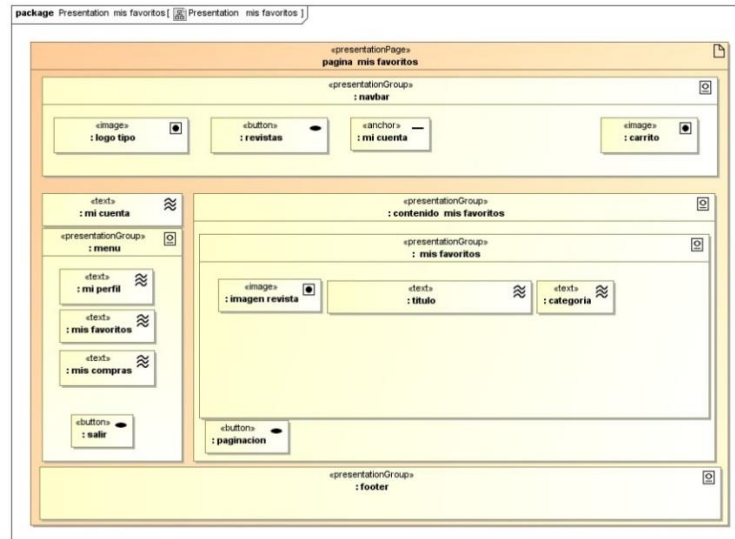
Mis compras (lector)



Nota: Se muestra el modelo de presentación mis compras del usuario lector.

Figura 3.69

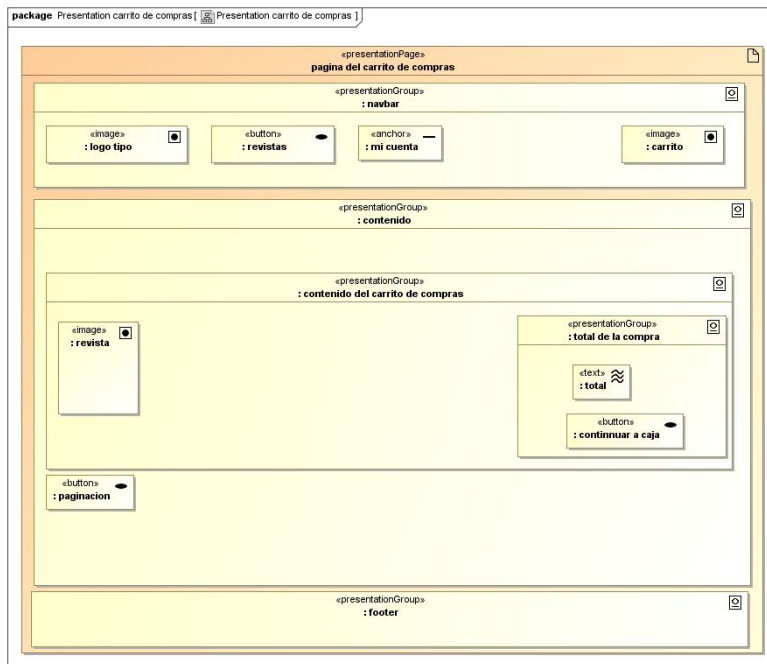
Mis favoritos (lector)



Nota: Se muestra el modelo de presentación mis favoritos del usuario lector.

Figura 3.70

Carrito de compras (lector)



Nota: Se muestra el modelo de presentación mis compras del usuario lector.

3.4. DESARROLLO DEL SISTEMA

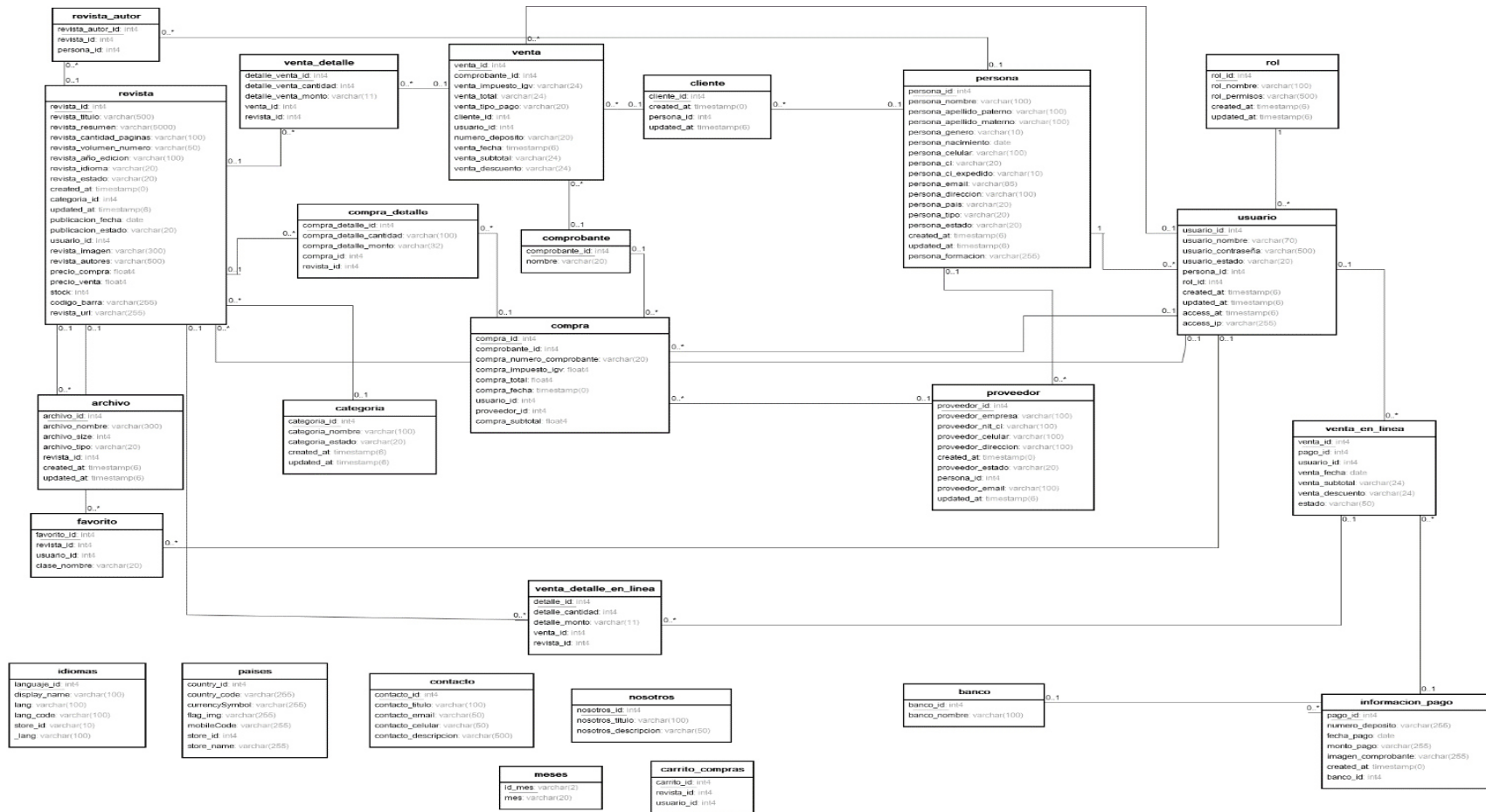
En esta etapa se realiza la programación del sistema web de acuerdo con la fase de diseño realizada anteriormente, además teniendo en cuenta todos los requerimientos presentados en la fase de análisis de requerimientos.

A continuación, se describe cada una de las interfaces presentadas en la fase de diseño, además de un fragmento de código del sistema web desarrollado.

3.4.1. Diseño de la base de datos

Figura 3.71

Modelo físico de la base de datos



Nota: Se muestra el modelo físico de la base de datos.

Figura 3.72

Muestra de código sql de la base de datos (tabla revista)

```
CREATE TABLE "revista" (  
  "revista_id" int4 NOT NULL DEFAULT nextval('revista_revista_id_seq'::regclass),  
  "revista_titulo" varchar(500) COLLATE "pg_catalog"."default",  
  "revista_resumen" varchar(5000) COLLATE "pg_catalog"."default",  
  "revista_cantidad_paginas" varchar(100) COLLATE "pg_catalog"."default",  
  "revista_volumen_numero" varchar(50) COLLATE "pg_catalog"."default",  
  "revista_año_edicion" varchar(100) COLLATE "pg_catalog"."default",  
  "revista_idioma" varchar(20) COLLATE "pg_catalog"."default",  
  "revista_estado" varchar(20) COLLATE "pg_catalog"."default",  
  "created_at" timestamp(0),  
  "categoria_id" int4,  
  "updated_at" timestamp(6),  
  "publicacion_fecha" date,  
  "publicacion_estado" varchar(20) COLLATE "pg_catalog"."default",  
  "usuario_id" int4,  
  "revista_imagen" varchar(300) COLLATE "pg_catalog"."default",  
  "revista_autores" varchar(500) COLLATE "pg_catalog"."default",  
  "precio_compra" float4,  
  "precio_venta" float4,  
  "stock" int4,  
  "codigo_barra" varchar(255) COLLATE "pg_catalog"."default",  
  "revista_url" varchar(255) COLLATE "pg_catalog"."default",  
  CONSTRAINT "revista_pkey" PRIMARY KEY ("revista_id")  
);  
ALTER TABLE "revista" OWNER TO "postgres";
```

Nota: Se muestra un fragmento de código de la base de datos.

3.4.2. Web de administración

3.4.2.1. Inicio de sesión

Figura 3.73

Inicio de sesión



Posgrado
Universidad Pública de El Alto

Iniciar sesión

Correo Electronico

Contraseña

 Ingresar

Nota: vemos la página de Inicio de Sesión del sistema web en el cual los usuarios ingresar con el rol que tengan, ejemplo administrador, editor, vendedor.

Figura 3.74

Muestra de código (iniciar sesión)

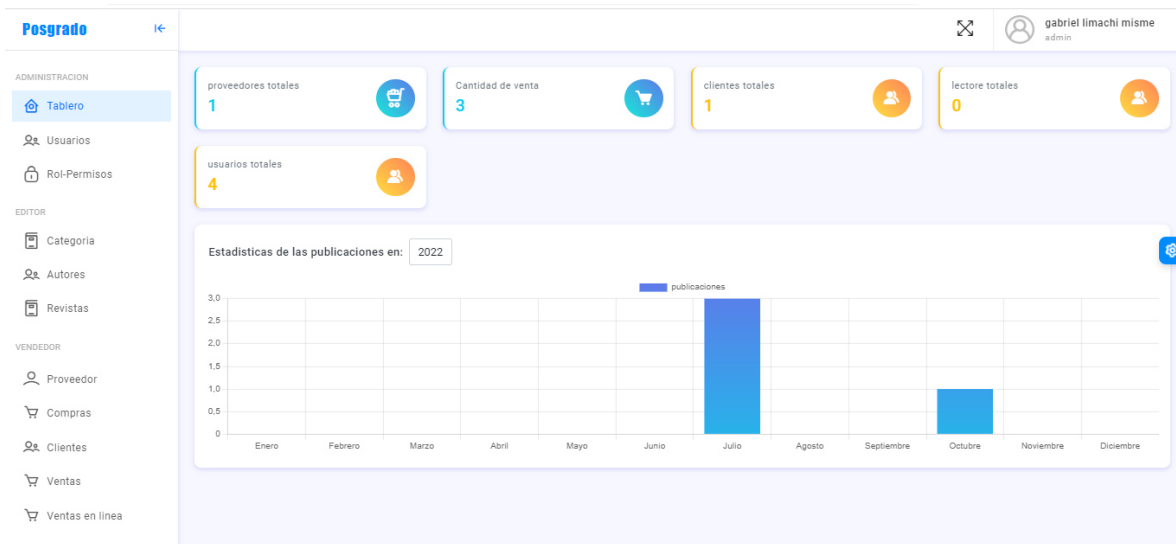
```
1 <template>
2 <div class="wrapper">
3 <div class="section-authentication-signin d-flex align-items-center justify-content-center my-5 my-lg-0">
4 <div class="container-fluid">
5 <div class="row row-cols-1 row-cols-lg-2 row-cols-xl-3">
6 <div class="col mx-auto">
7 <div class="mb-4 text-center">
8 <div class="animate animated animate__jackInTheBox animate_infinite animate_slower">
9 
10 </div>
11 </div>
12 <div class="card">
13 <div class="card-body">
14 <div class="border p-4 rounded">
15 <div class="text-center">
16 <h3 class="">Iniciar sesión</h3>
17 </div>
18 <div class="form-body">
19 <form class="row g-3">
20 <div class="col-12">
21 <label for="inputEmailAddress" class="form-label">Correo Electronico</label>
22 <input type="text" class="form-control" placeholder="nombre de usuario" v-model="usuario.username">
23 </div>
24 <div class="col-12">
25 <label for="inputChoosePassword" class="form-label">Contraseña</label>
26 <div class="input-group" id="show_hide_password">
27 <input type="password" class="form-control border-end-0" id="inputChoosePassword" value="" placeholder="contraseña" v-model="usuario.password">
28 </div>
29 </div>
30 <div>
31 <Alerta v-if="msg" :msg="msg" :classAlert="classAlert"></Alerta>
32 <div class="col-12">
33 <div class="d-grid">
34 <button type="button" class="btn btn-primary" @click="login()">Ingresar</button>
35 </div>
36 </div>
37 </form>
</div>
</div>
</div>
</div>
</div>
</div>
```

Nota: Se muestra un fragmento de código inicio de sesión.

3.4.2.2. Página de inicio

Figura 3.75

Página de inicio (rol de administrador)



Nota: vemos la página de Inicio del administrador con algunos datos y menú de opciones.

Figura 3.76

Muestra de código (página de inicio)

```
1 |<template>
2 |<div class="animate__animated animate__fadeInDown">
3 |<!-- start page wrapper -->
4 |<div class="page-wrapper">
5 |<div class="page-content">
6 |<div class="row row-cols-1 row-cols-md-2 row-cols-xl-4">
7 |<div class="col">
8 |<div class="card radius-18 border-start border-0 border-3 border-info">
9 |<div class="card-body">
10 |<div class="d-flex align-items-center">
11 |<div>
12 |<p class="mb-0 text-secondary">proveedores totales</p>
13 |<h4 class="my-1 text-info">{{proveedorCantidad}}</h4>
14 |</div>
15 |<div class="widgets-icons-2 rounded-circle bg-gradient-scooter text-white ms-auto"><i class="lni lni-cart-full"></i>
16 |</div>
17 |</div>
18 |</div>
19 |</div>
20 |<div class="col">
21 |<div class="card radius-18 border-start border-0 border-3 border-info">
22 |<div class="card-body">
23 |<div class="d-flex align-items-center">
24 |<div>
25 |<p class="mb-0 text-secondary">Cantidad de venta</p>
26 |<h4 class="my-1 text-info">{{ventasCantidad}}</h4>
27 |</div>
28 |<div class="widgets-icons-2 rounded-circle bg-gradient-scooter text-white ms-auto"><i class="bx bx-cart"></i>
29 |</div>
30 |</div>
31 |</div>
32 |</div>
33 |</div>
34 |</div>
```

Nota: Se muestra un fragmento de código la página de inicio del administrador.

3.4.2.3. Administrar usuario

Figura 3.77

Administrar usuario (rol de administrador)

The screenshot shows the 'Administrar usuario' page in the Posgrado system. The page has a sidebar on the left with navigation options: Posgrado, ADMINISTRACION (Tablero, Usuarios, Rol-Permisos), EDITOR (Categoría, Autores, Revistas), and VENDEDOR (Proveedor, Compras, Clientes, Ventas, Ventas en línea). The main content area is titled 'USUARIOS' and features a search bar 'Buscar por nombre', a 'Imprimir reporte pdf' button, and a 'Nuevo' button. Below these is a table with the following data:

#	nombre usuario	estado	rol	creado en	actualizado en	Acciones
1	editor	ACTIVO	editor	2022-07-08 02:18:58	2022-07-08 02:18:58	[editar] [eliminar] [estado]
2	gabriel	ACTIVO	lector	2022-06-19 23:38:17	2022-10-25 23:30:29	[editar] [eliminar] [estado]
3	vendedor	ACTIVO	vendedor	2022-06-17 13:30:01	2022-06-29 02:58:33	[editar] [eliminar] [estado]
4	admin	ACTIVO	admin			[editar]

Nota: vemos la página para administrar usuarios con algunos datos y las opciones de crear nuevo usuario, editar, eliminar, cambiar estado y reporte.

Figura 3.78

Muestra de código (Administrar usuario)

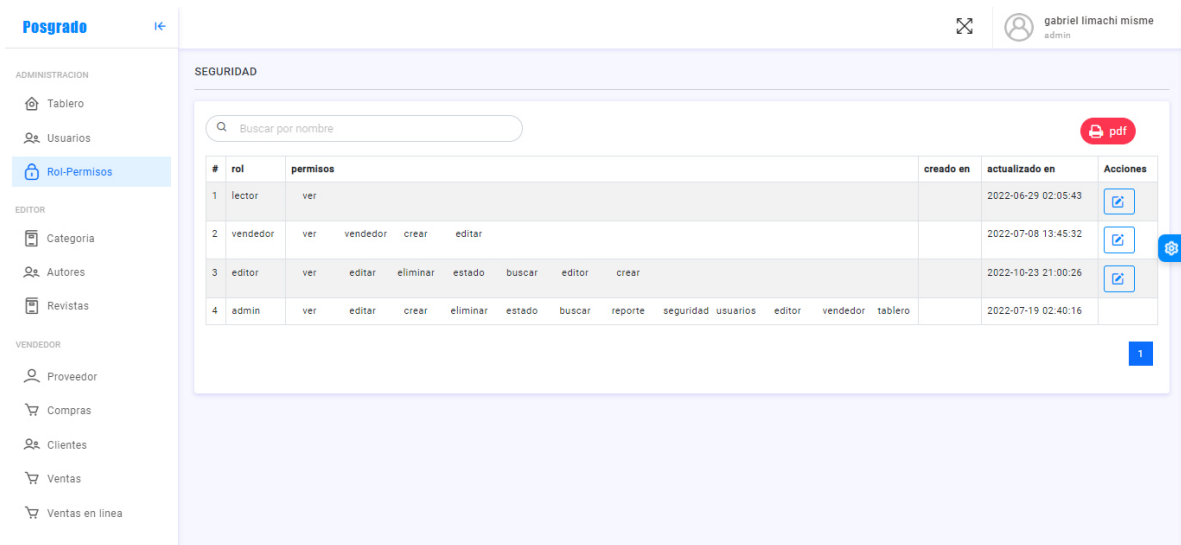
```
1 <template>
2 <div>
3
4 <!--start page wrapper -->
5 <div class="page-wrapper">
6 <div class="page-content">
7 <h2 class="mb-0 text-uppercase">Usuarios</h2>
8 </div>
9 <div class="card">
10 <div class="card-body">
11 <div class="col">
12 <div class="d-jg-flex align-items-center mb-3">
13 <div class="col-12 col-md-4 col-lg-4">
14 <div class="position-relative">
15 <input type="text" v-model="buscarTexto" v-on:keyup.enter="buscar" class="form-control ps-5 radius-30" placeholder="Buscar por nombre">
16 <span class="position-absolute top-50 translate-middle-y"><i class="bx bx-search" @click="buscar()">/></span>
17 </div>
18 </div>
19 <div class="ms-auto">
20 <div class="position-relative bottom-0 end-0 px-4">
21 <button type="button" class="btn btn-danger mt-2 radius-30 me-2" @click="pdf()"><i class="fa fa-print">/> Imprimir reporte pdf</button>
22 <button type="button" class="btn btn-primary radius-30 mt-2" @click="Botoncrear()"><i class="bx bx-plus-square">/> Nuevo</button>
23 </div>
24 </div>
25 </div>
26 <Spinner v-if="loader=false">/Spinner>
27 <ListTabla v-if="loader=true && datosPaginados.length> :datosPaginados="datosPaginados" @BotonDesactivar="BotonModalEstado" @BotonEditar="BotonEditar" @BotonEliminar="BotonModalElimina
28 </ListTabla>
29 <Paginacion v-if="loader=true && !datosPaginados.length=0" :paginaActual="paginaActual" :totalPaginas="totalPaginas" :listadoDatos="listadoDatos" @getNextPage="getNextPage" @getPr
30 <MensajesSinDatos v-if="loader=true && datosPaginados.length=0" @listarTodo="getUsuarios">/MensajesSinDatos>
31 </div>
32 </div>
33 </div>
34 </div>
35 </div>
36 </div>
37 <!--end page wrapper -->
38
39 <div>
40 <div class="col">
41 <div class="modal fade" id="eliminarModal" tabindex="-1" aria-hidden="true">
42 <div class="modal-dialog modal-sm">
```

Nota: Se muestra un fragmento de código la página administrar usuario.

3.4.2.4. Administrar rol y permisos

Figura 3.79

Administrar rol y permisos (rol de administrador)



Nota: vemos la página para administrar rol y permisos con algunos datos y la opción de editar.

Figura 3.82

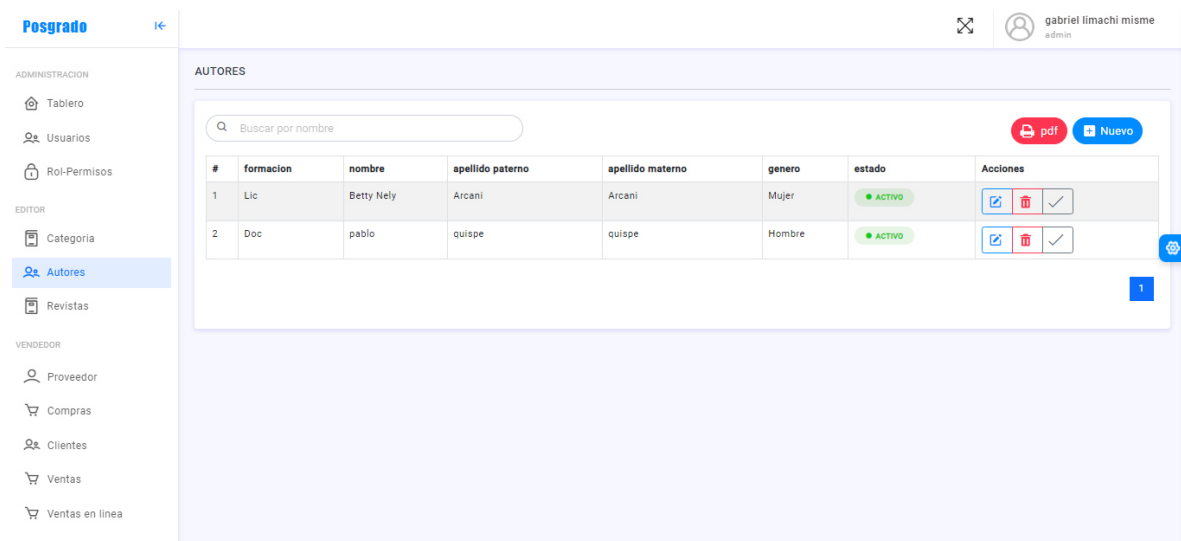
Muestra de código (Administrar categoría)

```
1 <template>
2 <div>
3   <div class="wrapper">
4     <!--start page wrapper-->
5     <div class="page-wrapper">
6       <div class="page-content">
7         <h2 class="mb-0 text-uppercase">Categoría / {{ CategoríaContenido.vue }}
8         </h2>
9         <div class="card">
10          <div class="card-body">
11            <div class="col">
12              <div class="d-lg-flex align-items-center mb-3">
13                <div class="col-12 col-md-4 col-lg-4">
14                  <div class="position-relative">
15                    <input type="text" v-model="buscarTexto" v-on:keyup.enter="buscar" class="form-control ps-5 radius-30" placeholder="Buscar por nombre">
16                    <span class="position-absolute top-0 product-show translate-middle-y"><i class="bx bx-search" @click="buscar()"></i></span>
17                  </div>
18                </div>
19                <div class="ms-auto">
20                  <div class="position-relative bottom-0 end-0 px-4">
21                    <button type="button" class="btn btn-danger mt-2 radius-30 me-2" @click="pdf()"><i class="fa fa-print"></i> Imprimir reporte pdf.</button>
22                    <button type="button" class="btn btn-primary radius-30 mt-2" @click="BotonCrear()" v-show="Pcrear.length > 0" @click="Pcrear.length > 0" class="bx bx-plus-square"></i> Nuevo.</button>
23                  </div>
24                </div>
25              </div>
26              <div v-if="!loader && !datosPaginados.length"></div>
27              <div class="table-responsive" v-if="loader=true && datosPaginados.length">
28                <table id="example2" class="table table-striped table-bordered">
29                  <thead>
30                    <tr>
31                      <th>#</th>
32                      <th>nombre categoría </th>
33                      <th>estado </th>
34                      <th>creado en </th>
35                      <th>actualizado en </th>
36                      <th>Acciones </th>
37                    </tr>
38                  </thead>
39                  <tbody v-if="datosPaginados.length > 0">
40                    <tr v-for="(lista,index) in datosPaginados" :key="lista.categoria_id" :index="index">
41                      <td>{{index}}</td>
42                      <td>{{lista.categoria_nombre}}</td>
43                      <td>
44                        <div v-if="lista.categoria_estado=='activo'" class="badge rounded-pill text-success bg-light-success p-2 text-uppercase px-3">
```

Nota: Se muestra un fragmento de código la página administrar categoría.

Figura 3.83

Administrar autores (rol de administrador)



Nota: vemos la página para administrar autores con algunos datos y las opciones crear nuevo, editar, eliminar, cambiar estado, imprimir reporte.

Figura 3.84

Muestra de código (Administrar autores)

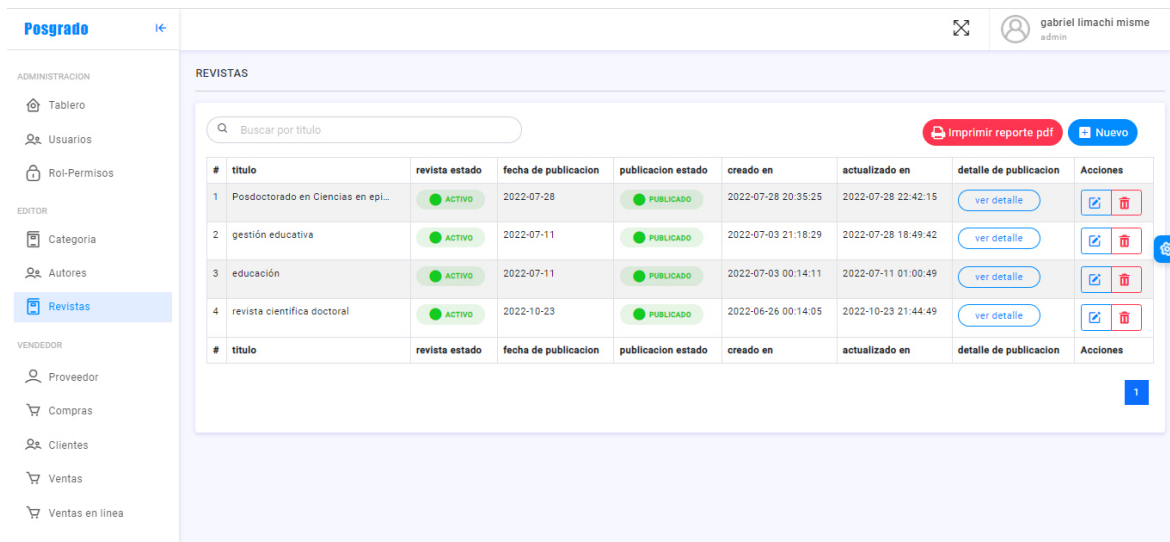
```
1 <template>
2 <div>
3   <div class="wrapper">
4     <!-- start page wrapper -->
5     <div class="page-wrapper">
6       <div class="page-content">
7         <h2 class="mb-0 text-uppercase">Autores</h2>
8         <hr />
9         <div class="card">
10          <div class="card-body">
11            <div class="col">
12              <div class="d-lg-flex align-items-center mb-3">
13                <div class="col-12 col-md-4 col-lg-4">
14                  <div class="position-relative">
15                    <input type="text" v-model="buscarTexto" v-on:keyup.enter="buscar" class="form-control ps-5 radius-30" placeholder="Buscar por nombre">
16                    <span class="position-absolute top-50 product-show translate-middle-y"><i class="bx bx-search" @click="buscar()"></i></span>
17                  </div>
18                </div>
19                <div class="ms-auto">
20                  <div class="position-relative bottom-0 end-0 px-4">
21                    <button type="button" class="btn btn-danger mt-2 radius-30 me-2" @click="pdf()"><i class="fa fa-print"></i> pdf</button>
22                    <button type="button" class="btn btn-primary radius-30 mt-2" @click="botonCrear()" v-show="Pcrear.length>0 && Pcrear"><i class="bx bx-plus-square"></i> Nuevo</button>
23                  </div>
24                </div>
25              </div>
26              <Spinner v-if="loader==false && !datosPaginados.length==0"></Spinner>
27              <ListTable v-if="loader==true && datosPaginados.length>0" :datosPaginados="datosPaginados" @BotonDesactivar="BotonModalEstado" @BotonEditar="BotonModalEditar" @BotonEliminar="BotonModalEliminar"
28                :Paginacion v-if="loader==true && !datosPaginados.length==0" :paginaActual="paginaActual" :totalPaginas="totalPaginas" :listadoDatos="listadoDatos" @getNextPage="getPaginaSiguiente" @getPreviousPage="getPaginaAnterior"
29                :MensajeSinDatos v-if="loader==true && datosPaginados.length==0" @listarTodos="getAutor" /></ListTable>
30            </div>
31          </div>
32        </div>
33      </div>
34    </div>
35  </div>
36  <!-- end page wrapper -->
37 </div>
38 <div class="col">
39   <div class="modal fade" id="eliminarModal" tabindex="-1" aria-hidden="true">
40     <div class="modal-dialog modal-sm">
41       <div class="modal-content">
42         <div class="modal-header">
43           <h3 class="modal-title">{{tituloEliminar()}}</h3>
44           <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
45         </div>
46       </div>
47     </div>
48   </div>
49 </div>
50 </div>
```

Nota: Se muestra un fragmento de código la página administrar autores.

3.4.2.6. Administrar Revistas y publicación

Figura 3.85

Revistas y publicación (rol de administrador)



Nota: vemos la página para administrar revistas y publicación con algunos datos y las opciones crear nuevo, editar, eliminar, cambiar estado, imprimir reporte y ver detalle.

Figura 3.90

Muestra de código (Administrar compras)

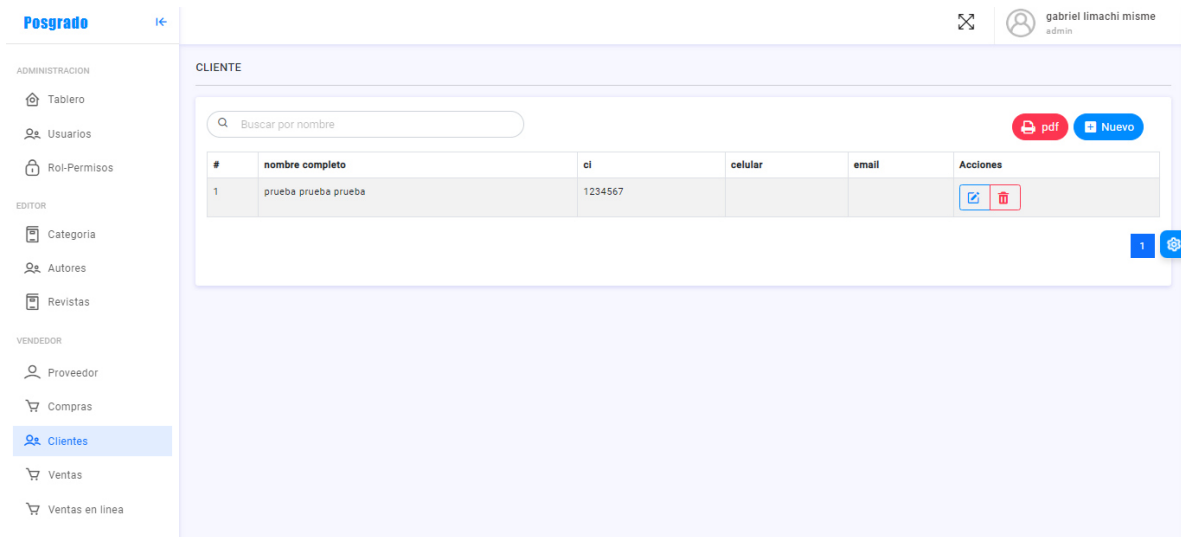
```
13 <div class="lg:file align-items-center mb-3">
14 <div class="col" col-md="4" rol="lg-4">
15 <div class="position-relative">
16 <input type="text" v-model="buscarTexto" v-on:keyup.enter="buscar" class="form-control ps-5 radius-30" placeholder="Buscar por nombre">
17 <span class="position-absolute top-50 product-show translate-middle-y"><i class="bx bx-search" @click="buscar()">/i></span>
18 </div>
19 </div>
20 <div class="position-absolute bottom-0 end-0 ps-4">
21 <button type="button" class="btn btn-primary radius-30 mt-2" @click="botonCrear()" v-show="!Pcrear.length0 && Pcrear"><i class="bx bx-plus-square">/i> Nuevo /button>
22 </div>
23 </div>
24 </div>
25 <div class="text-align-right">
26 <div class="table-responsive" v-if="!datosPaginados.length0">
27 <table id="example2" class="table table-striped table-bordered">
28 <thead>
29 <tr>
30 <th>#</th>
31 <th>proveedor</th>
32 <th>fecha</th>
33 <th>total</th>
34 <th>tipo de documento</th>
35 <th>Acciones</th>
36 </tr>
37 </thead>
38 </div>
39 <tbody v-if="datosPaginados.length > 0">
40 <tr v-for="(lista,index) in datosPaginados" :key="lista.compra_id" :index="index">
41 <td>{{index+1}}</td>
42 <td>{{lista.proveedor_empresa}}</td>
43 <td>{{lista.compra_fecha}}</td>
44 <td>{{lista.compra_total}}</td>
45 <td>{{lista.nombre}}</td>
46 <td>
47 <div class="row row-cols-auto g-3">
48 <div class="col">
49 <div class="btn-group" role="group">
50 <button type="button" class="btn btn-outline-primary ps-2" @click="botonDetalleCompra(lista.compra_id)">
51 <i class="bx bx-show">/i> detalles
52 </button>
53 </div>
54 </div>
55 </td>
56 </tr>
57 </tbody>
58 </table>
59 </div>
60 </div>
```

Nota: Se muestra un fragmento de código la página administrar compras.

3.4.2.9. Administrar cliente

Figura 3.91

Administrar cliente (rol de administrador)



Nota: vemos la página para administrar cliente con algunos datos y las opciones crear nuevo, editar, eliminar, imprimir reporte.

Figura 3.92

Muestra de código (Administrar cliente)

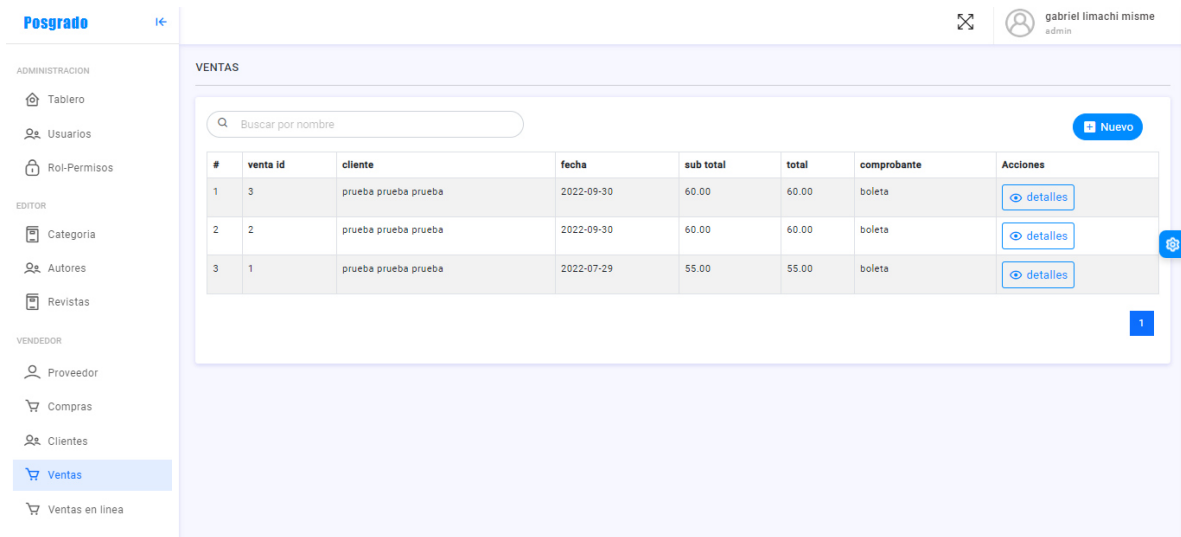
```
1 |<template>
2 | <div>
3 |   <div class="wrapper">
4 |     <!--start page wrapper -->
5 |     <div class="page-wrapper">
6 |       <div class="page-content">
7 |         <h6 class="mb-0 text-uppercase">cliente</h6>
8 |         <hr />
9 |         <div class="card">
10 |          <div class="card-body">
11 |            <div class="col">
12 |              <div class="d-lg-flex align-items-center mb-3">
13 |                <div class="col-12 col-md-4">
14 |                  <div class="position-relative">
15 |                    <input type="text" v-model="buscarTexto" v-on:keyup.enter="buscar" class="form-control ps-5 radius-30" placeholder="Buscar por nombre">
16 |                    <span class="position-absolute top-50 percent left-100 percent translate-100 percent"><i class="bx bx-search" @click="buscar()">/></span>
17 |                  </div>
18 |                </div>
19 |              </div>
20 |              <div class="text-align-right">
21 |                <div class="position-relative bottom-0 end-0 px-4">
22 |                  <button type="button" class="btn btn-danger mt-2 radius-30 me-2" @click="pdf()"><i class="fa fa-print">/i> pdf</button>
23 |                  <button type="button" class="btn btn-primary radius-30 mt-2" @click="BotonCrear()" v-show="Pcrear.length > 0" ><i class="bx bx-plus-square">/i> Nuevo</button>
24 |                </div>
25 |              </div>
26 |            </div>
27 |            <div v-if="!loader && !datosPaginados.length"></div>
28 |            <div class="table-responsive" v-if="!loader && datosPaginados.length">
29 |              <table id="example2" class="table table-striped table-bordered">
30 |                <thead>
31 |                  <tr>
32 |                    <th>#</th>
33 |                    <th>nombre completo</th>
34 |                    <th>ci</th>
35 |                    <th>telefono</th>
36 |                    <th>email</th>
37 |                    <th>Acciones</th>
38 |                  </tr>
39 |                </thead>
40 |                <tbody v-if="datosPaginados.length > 0">
41 |                  <tr v-for="(lista,index) in datosPaginados" :key="lista.cliente_id" :index="index">
42 |                    <td>{{index}}</td>
43 |                    <td>{{lista.nombre_completo}}</td>
44 |                    <td>{{lista.persona_ci}}</td>
45 |
```

Nota: Se muestra un fragmento de código la página administrar cliente.

3.4.2.10. Administrar ventas

Figura 3.93

Administrar ventas (rol de administrador)



Nota: vemos la página para administrar ventas con algunos datos y las opciones crear nuevo, ver detalle e imprimir reporte.

Figura 3.94

Muestra de código (Administrar ventas)

```
1 <template>
2 <div>
3   <div class="wrapper">
4     <!--start page wrapper -->
5     <div class="page-wrapper">
6       <div class="page-content">
7         <h2 class="mb-8 text-uppercase">ventas</h2>
8         <div class="card">
9           <div class="card-body">
10            <div class="col">
11              <div class="d-lg-flex align-items-center mb-3">
12                <div class="col-12 col-md-4 col-lg-4">
13                  <div class="position-relative">
14                    <input type="text" v-model="buscarTexto" v-on:keyup.enter="buscar" class="form-control ps-5 radius-30 placeholder="Buscar por nombre">
15                    <span class="position-absolute top-50 translate-middle-y" class="bx bx-search" @click="buscar()" /></span>
16                  </div>
17                </div>
18              </div>
19              <div class="text-auto">
20                <div class="position-relative bottom-0 end-0 px-4">
21                  <button type="button" class="btn btn-primary radius-30 mt-2" @click="BotonCrear()" v-show="Pcrear.length > 0" class="bx bx-plus-square" /><i>Nuevo</i> /button
22                </div>
23              </div>
24            </div>
25            <div class="table-responsive" v-if="!loader && datosPaginados.length > 0">
26              <table id="example" class="table table-striped table-bordered">
27                <thead>
28                  <tr>
29                    <th>#</th>
30                    <th>venta id</th>
31                    <th>cliente</th>
32                    <th>fecha</th>
33                    <th>sub total</th>
34                    <th>total</th>
35                    <th>comprobante</th>
36                    <th>Acciones</th>
37                  </tr>
38                </thead>
39                <tbody v-if="datosPaginados.length > 0">
40                  <tr v-for="(lista,index) in datosPaginados" :key="lista.compra_id" :index="index">
41                    <td>{{index+1}}</td>
42                    <td>{{lista.venta_id}}</td>
43                    <td>{{lista.nombre_completo}}</td>
44                    <td>{{lista.fecha}}</td>
45                    <td>{{lista.subtotal}}</td>
46                    <td>{{lista.total}}</td>
47                    <td>{{lista.comprobante}}</td>
48                    <td>{{lista.acciones}}</td>
49                  </tr>
50                </tbody>
51              </table>
52            </div>
53          </div>
54        </div>
55      </div>
56    </div>
57  </div>
58 </template>
```

Nota: Se muestra un fragmento de código la página administrar ventas.

3.4.3. Web pública

3.4.3.1. Iniciar sesión

Figura 3.95

Iniciar sesión (rol lector)

Iniciar sesión

Correo Electronico

Contraseña

[Ingresar](#)

[¿No tienes una cuenta? Registrarte aquí](#)

[¿Olvidaste la contraseña? Haz clic aquí](#)

[regresar a inicio](#)

Nota: vemos la página de inicio de sesión del lector.

Figura 3.96

Muestra de código (iniciar sesión)

```
1 |<template>
2 |<div>
3 |  <section class="login section" v-if="!loggedIn">
4 |    <div class="container">
5 |      <div class="row">
6 |        <div class="col-lg-4 offset-lg-4 col-md-8 offset-md-2 col-12">
7 |          <div class="form-head">
8 |            <h4 class="title">iniciar sesión:</h4>
9 |            <form>
10 |              <div class="form-group">
11 |                <label for="email">Correo electrónico:</label>
12 |                <input type="text" class="form-control" name="email" id="email" v-model="usuario.username" placeholder="ingresa su correo electrónico ejemplo@gmail.com" />
13 |              </div>
14 |              <div class="form-group">
15 |                <label for="password">Contraseña:</label>
16 |                <input type="password" class="form-control" name="password" id="password" v-model="usuario.password" />
17 |              </div>
18 |              <alert v-if="msg" :msg="msg" :classAlert="classAlert"></alert>
19 |              <div class="button">
20 |                <button type="button" class="btn" @click="login()">Ingresar</button>
21 |              </div>
22 |              <p class="outer-link"><a href="#">No tienes una cuenta?</a>
23 |                <router-link to="/registrarse">
24 |                  Registrar aquí
25 |                </router-link>
26 |              </p>
27 |              <p class="outer-link"><a href="#">Olvidaste la contraseña?</a>
28 |                <router-link to="/forgot_password">
29 |                  Haz clic aquí
30 |                </router-link>
31 |              </p>
32 |              <p class="outer-link"><a href="#">regresar a inicio</a>
33 |            </form>
34 |          </div>
35 |        </div>
36 |      </div>
37 |    </div>
38 |  </section>
39 | </div>
40 | </template>
```

Nota: Se muestra un fragmento de código la página iniciar sesión usuario lector.

3.4.3.2. Registro de usuario

Figura 3.97

Registro de nuevo usuario lector

The image shows a registration form titled "Registro" with the following fields and layout:

- Nombre (text input)
- Apellido paterno (text input)
- Apellido materno (text input)
- género (radio buttons for "Hombre")
- pais (dropdown menu showing "bolivia")
- celular (text input)
- Dirección de correo electrónico (text input)
- Nombre de usuario (text input)
- Contraseña (text input)
- Confirmar Contraseña (text input)
- Crear Cuenta (blue button)

Nota: vemos la página de registro del nuevo usuario lector.

Figura 3.98

Muestra de código (registro de usuario)

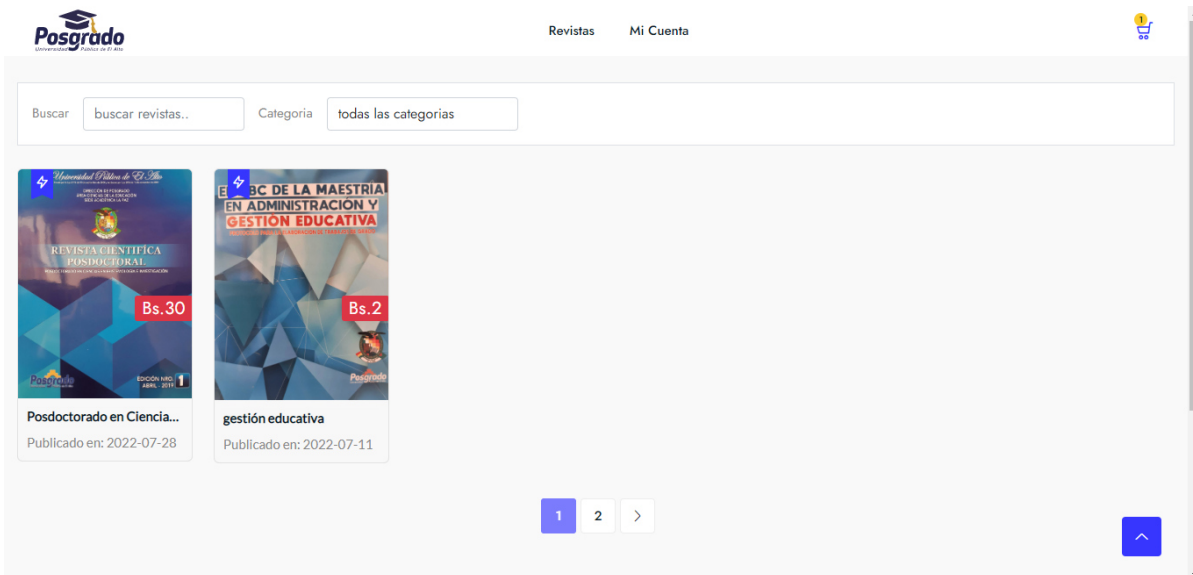
```
1 <template>
2 <div>
3   <section class="login registration section">
4     <div class="container">
5       <div class="row">
6         <div class="col-12 col-sm-8 offset-sm-2 col-md-6 offset-md-3 mt-2 pt-2">
7           <div class="form-head">
8             <h4 class="title">Registro</h4>
9             <form @submit.prevent="register()" class="action" /> </register" method="post">
10            <div class="row">
11              <div class="col-12 col-sm-6">
12                <div class="form-group">
13                  <label for="nombre">Nombre</label>
14                  <input type="text" class="form-control" name="nombre" id="nombre" v-model="nombre" />
15                </div>
16              </div>
17              <div class="col-12 col-sm-6">
18                <div class="form-group">
19                  <label for="paterno">Apellido paterno</label>
20                  <input type="text" class="form-control" v-model="apellido_paterno" />
21                </div>
22              </div>
23              <div class="col-12 col-sm-6">
24                <div class="form-group">
25                  <label for="materno">Apellido materno</label>
26                  <input type="text" class="form-control" v-model="apellido_materno" />
27                </div>
28              </div>
29              <div class="col-12 col-sm-6">
30                <div class="form-group">
31                  <label for="as">Género</label>
32                  <select class="form-control" name="genero" id="genero">
33                    <option>Hombre</option>
34                    <option>Mujer</option>
35                  </select>
36                </div>
37              </div>
38              <div class="col-12 col-sm-3">
39                <div class="form-group">
40                  <label for="as">País</label>
41                  <select class="form-control" name="pais" id="pais">
42                    <option>Bolivia</option>
43                  </select>
44                </div>
45              </div>
46              <div class="col-12 col-sm-6">
47                <div class="form-group">
48                  <label for="as">Celular</label>
49                  <input type="text" class="form-control" />
50                </div>
51              </div>
52            </div>
53          </div>
54        </div>
55      </div>
56    </div>
57  </div>
58 </div>
59 </div>
60 </div>
61 </div>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>
```

Nota: Se muestra un fragmento de código la página de registro.

3.4.3.3. Revistas

Figura 3.99

Revistas disponibles (rol lector)



Nota: vemos la página de revistas con detalles de con filtro de categoría y búsqueda y paginado.

Figura 3.100

Muestra de código (revistas)

```
1 <template>
2 <div>
3 </div>
4 <section class="section blog-single">
5 <div class="container">
6 <fieldset class="border p-3 mb-3 buscarContenido">
7 <div class="row g-3 align-items-center">
8 <div class="col-auto">
9 <label class="col-form-label">Buscar</label>
10 </div>
11 <div class="col-auto">
12 <input type="text" class="form-control" placeholder="buscar revistas..." v-model="filtros.buscarTexto" @input="buscar()">
13 </div>
14 <div class="col-auto">
15 <label for="" class="col-form-label">Categoría</label>
16 </div>
17 <div class="col-auto" v-if="categorias.length > 0">
18 <select class="form-control" name="" id="" v-model="categoriaID">
19 <option value="0">Todas las categorías</option>
20 <option v-for="(data, index) in categorias" :key="data.categoria_id" :index="index" :value="data.categoria_id">
21 {{ data.categoria_nombre }}
22 </option>
23 </select>
24 </div>
25 </div>
26 </fieldset>
27
28 <!-- resultados -->
29 <div class="lista-revista">
30 <div class="row">
31 <div class="col-lg-12 col-md-12 col-12">
32 <loaden v-if="loader===false">/loaden
33 <listar v-if="loader===true && datosPaginados.length>0" :datos="datosPaginados" :/:Listar>
34 <MensajeSinDatos v-if="loader===true && datosPaginados.length=0" @listar:todo="cargarPublicacion" :/:MensajeSinDatos>
35 <Paginacion v-if="loader===true" :paginaActual="paginaActual" :totalPaginas="totalPaginas" :listadoDatos="listadoDatos" @getNextPage="getPaginaSiguiente" @getPreviousPage="getPaginaAnterior" @getDataP
36 </div>
37 </div>
38 </div>
39
40 </div>
41 </section>
42 <footer></footer>
43
44 </div>
45 </template>
```

Nota: Se muestra un fragmento de código la página ver revistas del usuario lector.

3.4.3.4. Detalle de revistas

Figura 3.101

Detalle de revistas (rol lector)



Nota: vemos la página detalle de las revistas, agregar favoritos, agregar al carrito, información básica.

Figura 3.102

Muestra de código (detalle de revistas)

```
1 <template>
2 <div>
3 <!--Navbar/Navbar-->
4 <section class="section blog-single">
5 <div class="container">
6 <div class="loader-detalle" v-if="loader==false">
7 <!--Loader-->
8 <!--Loader v-if="loader=true"-->
9 </div>
10 <div class="detalle-revista">
11 <div class="row v-if="ListaDatos && loader==true">
12 <div class="col-lg-3 col-md-12 col-12 text-center">
13 <div class="single-inner">
14 <div class="post-thumbails">
15 
16 
17 <div class="row">
18 <div class="redesSociales">
19 </div>
20 </div>
21 </div>
22 </div>
23 </div>
24 </div>
25 <div class="col-lg-9 col-md-12 col-12">
26 <div class="post-detalle">
27 <div class="detail-inner">
28 <h2 class="post-title">
29 <{{ ListaDatos.revista_titulo }}
30 </h2>
31 <div class="row">
32 <ul class="info" v-show="loggedin">
33 <li class="price">$880.00/111
34 <span class="classiker" @click="like()">
35 <span href="javascript:void(0)"><i class="lni lni-heart"></i></span>
36 </span>

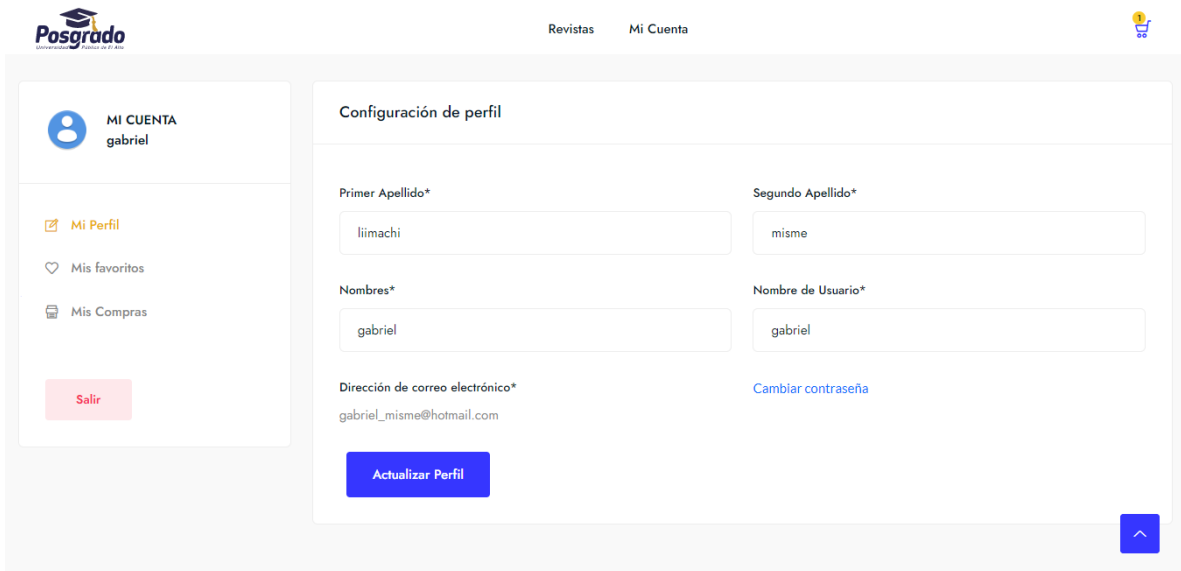
```

Nota: Se muestra un fragmento de código la página detalle de las revistas usuario lector.

3.4.3.5. Perfil de usuario

Figura 3.103

Perfil de usuario (rol lector)



Nota: vemos la página de cuenta del usuario menú perfil del usuario con sus datos personales que podrá actualizar y cambiar contraseña.

Figura 3.106

Muestra de código (mis favoritos)

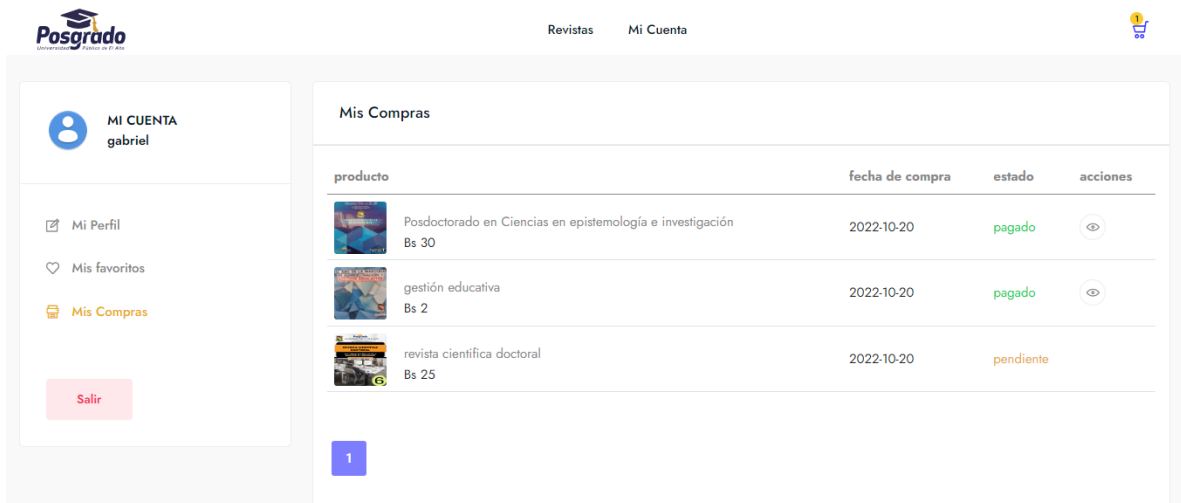
```
1 <template>
2 <div>
3   <div class="main-content">
4     <div class="dashboard-block mt-0">
5       <h3 class="block-title">Mis Favoritos</h3>
6       <Loader v-if="loader=false"></Loader>
7
8       <div class="bd-example" v-show="loader=true && datosPaginados.length >0">
9         <div class="table-responsive">
10          <table class="table align-middle">
11            <thead>
12              <tr class="thead-titulo">
13                <th scope="col">titulo</th>
14                <th scope="col">categoria</th>
15              </tr>
16            </thead>
17            <tbody>
18              <tr v-for="(item,index) in datosPaginados" :key="index">
19                <td>
20                  <div class="item-image">
21                    
22                    
23                    <div class="caption">
24                      <h3 class="title">
25                        <router-link to="" />{{item.revista_titulo}}
26                      </h3>
27                      </router-link>
28                    </div>
29                    <span class="price">Bs {{item.precio_venta}}</span>
30                  </div>
31                </td>
32                <td>{{item.categoria_nombre}}</td>
33              </tr>
34            </tbody>
35          </table>
36        </div>
37      </div>
38    </div>
39  </div>
40 </template>
```

Nota: Se muestra un fragmento de código la página mis favoritos del usuario lector.

3.4.3.7. Mis compras

Figura 3.107

Mis compras (rol lector)



Nota: vemos la página de cuenta del usuario menú mis compras con algunas revistas y estado.

Figura 3.108

Muestra de código (mis compras)

```
1 <template>
2 <div>
3   <div class="main-content">
4     <div class="dashboard-block et-0">
5       <h3 class="block-title">Mis Compras</h3>
6       <loader v-if="loader===false"></loader>
7
8       <div class="bd-example" v-show="loader===true && datosPaginados.length >0">
9         <div class="table-responsive">
10          <table class="table align-middle">
11            <thead>
12              <tr class="thead-titulo">
13                <th scope="col">producto</th>
14                <th scope="col">fecha de compra</th>
15                <th scope="col">estado</th>
16                <th scope="col">acciones</th>
17              </tr>
18            </thead>
19            <tbody>
20              <tr v-for="(item,index) in datosPaginados" :key="index">
21                <td>
22                  <div class="item-image">
23                    
24                    
25                  <div class="content">
26                    <h3 class="title">
27                      <router-link :to="'/detalle/' + item.revista_url">
28                        {{item.revista_titulo}}
29                      </router-link>
30                    </h3>
31                    <span class="price">Bs {{item.precio_venta}}</span>
32                  </div>
33                </td>
34                <td {{item_venta_fecha}}</td>
35                <td class="pending" v-show="item.estado==='pendiente'"{{item.estado}}</td>
36                <td class="paid" v-show="item.estado==='pagado'"{{item.estado}}</td>
37              </tr>
38            </tbody>
39          </table>
40        </div>
41      </div>
42    </div>
43  </div>
44 </template>
```

Nota: Se muestra un fragmento de código la página mis compras del usuario lector.

3.5. IMPLEMENTACIÓN DEL SISTEMA

Las características del equipo que cuenta la institución para la implementación del presente sistema:

3.5.1. Requerimientos de hardware

- Microprocesador Intel(R) Core(TM) i3-4005U CPU 1.70 GHz o superior
- Memoria Ram de 6 Gb o superior
- Disco duro de 100 Gb
- Sistema operativo Ubuntu

3.5.2. Requerimientos de software

- Servidor Apache/2.4
- PHP versión 7.4 o superior
- PostgreSQL 11

3.5.3. IMPLANTACIÓN DEL SISTEMA

El servidor que cuenta la dirección de posgrado ya tiene todo lo necesario desde el sistema operativo y la base de datos las herramientas de comunicación ssh

Para las pruebas se realizó en un sistema virtual.

3.5.3.1. Instalación del gestor de base de datos Postgresql

Tabla 3.1

Habilitación del repositorio PostgreSQL Apt

descripción	comandos
Instalación de certificados	# sudo apt install wget ca-certificates
Actualiza los paquetes de instalación de claves.	# wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc sudo apt-key add -
Descarga los paquetes del repositorio de postgresql.	# sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ `lsb_release -cs`-pgdg main" >> /etc/apt/sources.list.d/pgdg.list'

Nota: Se muestra los comandos para habilitar el repositorio de PostgreSQL.

Tabla 3.2

Instalación de PostgreSQL

descripción	comandos
Actualiza los paquetes de instalación	# sudo apt update
Instalación de Postgresql	# sudo apt-get install postgresql postgresql-contrib

Nota: Se muestra los comandos para instalar PostgreSQL.

Tabla 3.3

Conexión a PostgreSQL

descripción	comandos
Asigna contraseña y usuario postgres	# sudo su - postgres
Ingresa como usuario postgres	# psql
Información de la conexión	postgres-# \conninfo
Salir de la conexión	postgres-# \q

Nota: Se muestra los comandos para una conexión a PostgreSQL.

Figura 3.109

Uso de los comandos para la instalación de PostgreSQL

```
gab@gab-virtual-machine: ~  
gab@gab-virtual-machine:~$ lsb_release -a  
No LSB modules are available.  
Distributor ID: Ubuntu  
Description:    Ubuntu 20.04.1 LTS  
Release:        20.04  
Codename:       focal  
gab@gab-virtual-machine:~$ sudo apt install wget ca-certificates  
[sudo] password for gab:  
gab@gab-virtual-machine:~$ sudo apt install wget ca-certificates  
[sudo] password for gab:  
gab@gab-virtual-machine:~$ sudo sh -c 'echo "deb http://apt.postgresql.org/pub/  
repos/apt/ `lsb_release -cs`-pgdg main" >> /etc/apt/sources.list.d/pgdg.list'  
gab@gab-virtual-machine:~$ sudo apt update  
  
postgres@gab-virtual-machine:~$ psql  
psql (15.0 (Ubuntu 15.0-1.pgdg20.04+1))  
Type "help" for help.  
  
postgres=# \conninfo  
You are connected to database "postgres" as user "postgres" via socket in "/var/run/postgresq  
l" at port "5432".  
postgres=#
```

Nota: Se muestra los comandos utilizados para la instalación de PostgreSQL.

3.5.3.2. Instalación del servidor apache

Tabla 3.4

Instalación de apache como usuario root

Descripción	comandos
Instalación	apt install apache2
inicio	systemctl start apache2
habilitar	systemctl enable apache2
estado	systemctl status apache2
reiniciar apache	systemctl restart apache2

Nota: Se muestra los comandos para la instalación de Apache.

Figura 3.110

Uso de los comandos para la instalación del servidor web apache

```
root@gab-virtual-machine:/var/lib/postgresql# apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libfprint-2-tod1 libllvm9
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following packages will be upgraded:
  apache2 apache2-bin apache2-data apache2-utils
4 upgraded, 0 newly installed, 0 to remove and 566 not upgraded.
Need to get 1.520 kB of archives.
After this operation, 14,3 kB of additional disk space will be used.
Do you want to continue? [Y/n]

root@gab-virtual-machine:/var/lib/postgresql# systemctl start apache2
root@gab-virtual-machine:/var/lib/postgresql#
root@gab-virtual-machine:/var/lib/postgresql# systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/
systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
root@gab-virtual-machine:/var/lib/postgresql#
root@gab-virtual-machine:/var/lib/postgresql# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
   Active: active (running) since Tue 2022-11-01 19:29:56 -04; 16min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 14146 (apache2)
    Tasks: 6 (limit: 2285)
   Memory: 14.9M
   CGroup: /system.slice/apache2.service
           └─14146 /usr/sbin/apache2 -k start
           └─14177 /usr/sbin/apache2 -k start
```

Nota: Se muestra los comandos de la instalación de Apache.

3.5.3.3. Instalación de PHP

Tabla 3.5

Instalación y configuración de PHP 7

Descripción	Comandos
Ingresa como súper usuario (root)	# sudo su
Agregue el repositorio para que pueda realizar la descarga de los paquetes PHP en la versión correcta.	# add-apt-repository ppa:ondrej/php #apt-get update
Instalar PHP y aceptar la instalación de dependencias.	# apt-get install php7.0 php7.0-dev php7.0-cli
Verifique la versión de PHP	# php -v
Activando La Extensión De La Base	#apt-get install php7.0-pgsql

De Datos.

Nota: Se muestra los comandos para la instalación de Php.

Figura 3.111

Uso de los comandos para la instalación de PHP

```
root@gab-virtual-machine:/var/lib/postgresql# php -v
PHP 7.4.3 (cli) (built: Oct  6 2020 15:47:56) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
root@gab-virtual-machine:/var/lib/postgresql#
root@gab-virtual-machine:/var/lib/postgresql# apt-get install php7.0-pgsql
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Nota: Se muestra los comandos de la instalación de Php.

3.5.3.4. Configurar la zona horaria del servidor

Tabla 3.6

Configurar la zona horaria del servidor

Descripción	Comandos
listar la hora y fecha actual	timedatectl
lista de zonas activas	timedatectl list-timezones
Modo grafico zonas	dpkg-reconfigure tzdata
Actualizar la zona horaria	timedatectl set-timezone America/La_Paz

Nota: Se muestra los comandos de la configuración la zona horaria.

Figura 3.112

Uso de los comandos para la configuración de la zona horaria

```
root@gab-virtual-machine: /
root@gab-virtual-machine:/# timedatectl
    Local time: mar 2022-11-01 20:12:38 -04
    Universal time: mié 2022-11-02 00:12:38 UTC
    RTC time: mié 2022-11-02 00:12:38
    Time zone: America/La_Paz (-04, -0400)
System clock synchronized: yes
    NTP service: active
    RTC in local TZ: no
root@gab-virtual-machine:/#
```

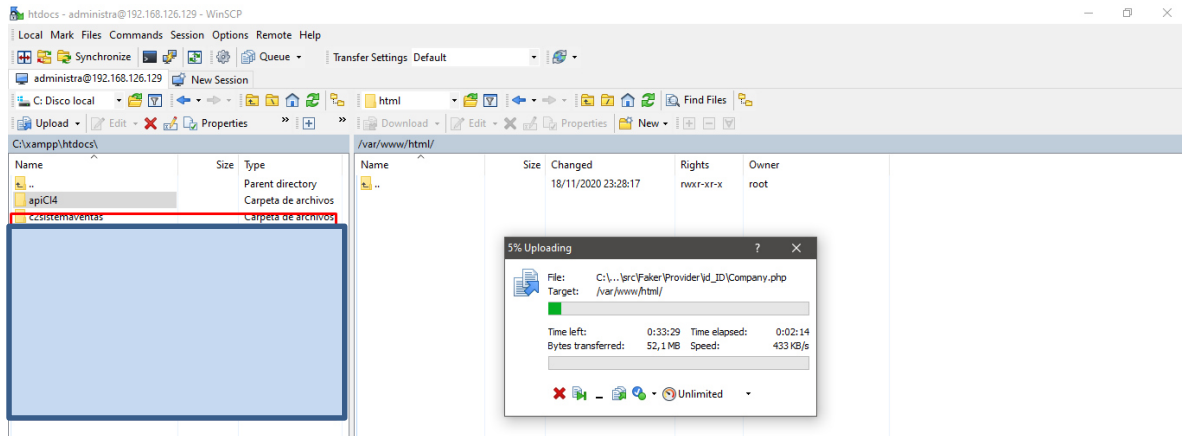
Nota: Se muestra los comandos de la configuración de la zona horaria.

3.5.3.5. Despliegue del aplicativo

Copia del “sistema” en la dirección /var/www/html/ usando la herramienta de winscp.

Figura 3.113

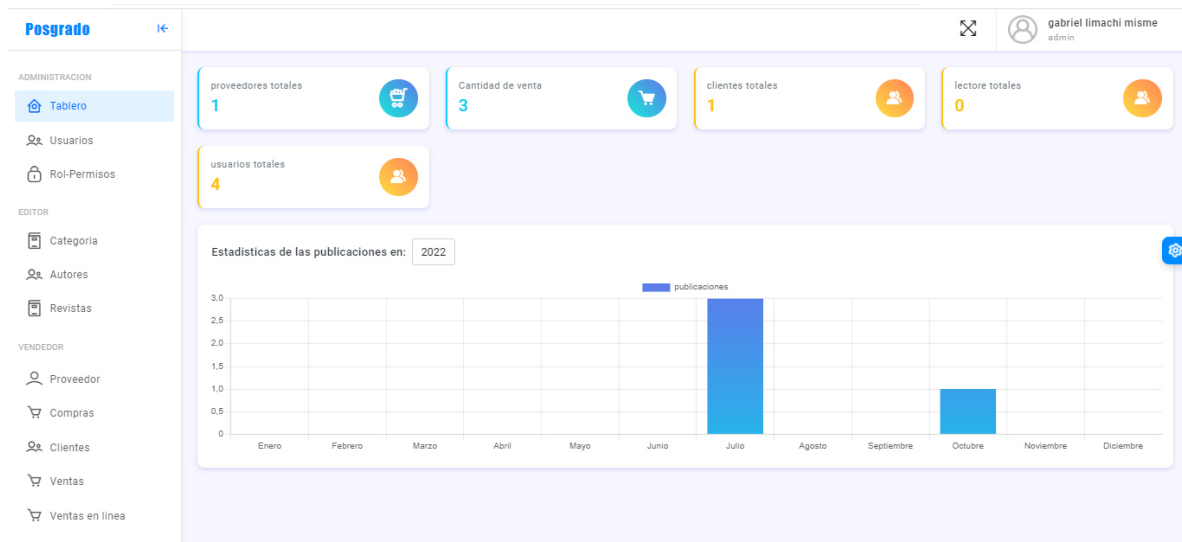
Transferencia de archivos al servidor



Nota: Se muestra la transferencia de archivos del proyecto al servidor.

Figura 3.114

Página de inicio como administrador



Nota: vemos la página de Inicio del administrador con algunos datos y menú de opciones.

3.6. EL MANTENIMIENTO

En esta etapa al estar el sistema web implementada requiere de mantenimiento a largo plazo o requiere actualización, por otra parte, la base de datos si requiere de mantenimiento se lo hace directamente desde el pgAdmin4 para PostgreSQL, pero por cualquier situación el desarrollador podrá ingresar al código fuente del sistema web mediante SSH herramientas como: Navicat, heidisql para bases de datos o winscp, Putty gestiona el servidor directamente desde la línea de comandos.

A blue decorative frame surrounds the text, consisting of a vertical bar on the left, a vertical bar on the right, and horizontal bars at the top and bottom, all with a slight trapezoidal shape.

CAPÍTULO IV

PRUEBAS,
SEGURIDAD,
CALIDAD
Y COSTO

4.1. INTRODUCCIÓN

En este capítulo se desarrollara las pruebas de software, calidad del software usando la Iso/iec 25000, seguridad del software según Iso 27000 y por último la estimación del software.

4.2. PRUEBAS DE SOFTWARE

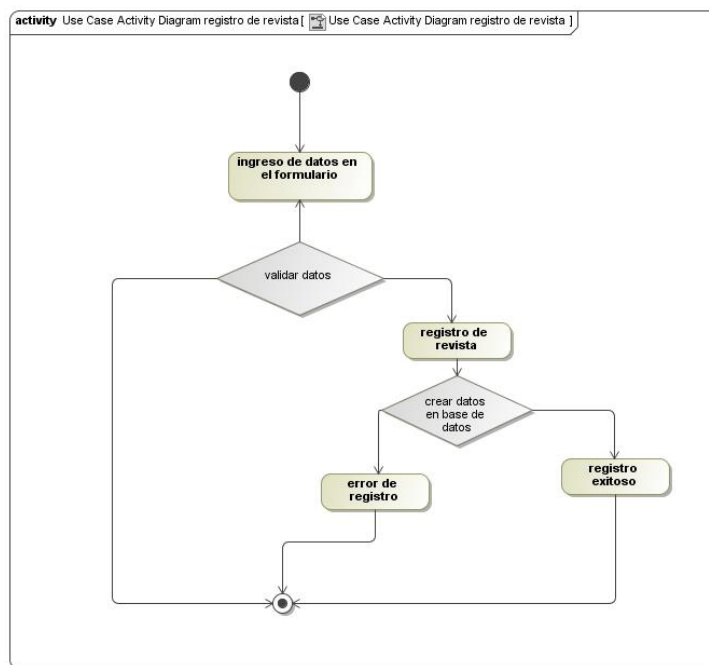
4.2.1. Pruebas de caja blanca

En esta prueba se realizara el cálculo de regiones de un diagrama de flujo o estado en nodos de un gráfico de flujo que deben ser consideradas como partes independientes del sistema, las cuales son entradas que se ejecutan al menos una vez.

Para la prueba de caja blanca se tomará en cuenta el registro de revista.

Figura 4.1

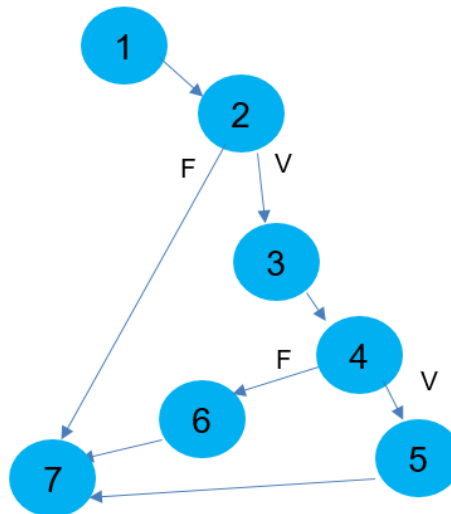
Diagrama de estado Registro de revistas



Nota: Se muestra el diagrama de flujo o de estado.

Figura 4.2

Grafo de flujo Registro de revistas



Número de nodos = 7

Número de aristas = 8

Analizado el diagrama de estado registro de revista y generando el grafo, ahora se procede a determinar la complejidad ciclomática mediante la siguiente formula:

$$(G) = A - N + 2$$

Dónde:

(G): complejidad ciclomática del grafo

A:(Aristas)

N:(Nodos)

Por tanto:

$$(G) = 8 - 7 + 2$$

$$(G) = 3$$

Determinar el conjunto básico de caminos linealmente independientes. Los caminos que deben ser probados dadas ciertas variables son 3. Estos caminos

son los siguientes:

Camino 1: 1 – 2 – 3 – 4 – 5 – 7

Camino 2: 1 – 2 – 3 – 4 – 6 – 7

Camino 3: 1 – 2 – 7

Preparar los casos de prueba para forzar la ejecución de cada camino. Esta última condición establece que, para la ejecución de ciertos caminos, se deben establecer las condiciones en las que al menos se ejecuta los nodos establecidos en el camino.

Camino 1: Inicia con el registro de revistas, se valida los campos requeridos del formulario, si todos están correctamente llenados, se crean los datos de la revista y se envía por el método post, posteriormente se crea en la base de datos, y finaliza con mensaje de respuesta exitosa.

Camino 2: Inicia con el registro de revistas, se valida los campos requeridos del formulario, si todos están correctamente llenados, se pasa a crear los datos de la revista se envía los datos por método post y posteriormente ocurre un error en la base de datos y finaliza con mensaje de error de respuesta.

Camino 3: Inicia con el registro de revistas, los campos ingresados no son válidos, y finaliza el proceso hasta pasar la validación.

4.2.2. Pruebas de caja negra

Las pruebas de caja negra o también conocidas como pruebas de comportamiento se centran en los requisitos funcionales del software.

Para las pruebas de caja negra se tomará en cuenta el registro de productos y el registro de una venta simple.

Tabla 4.1*Prueba de caja negra Registro de revistas*

Caso de prueba: Registro de revistas	
Descripción	El registro se realiza primeramente seleccionando en el menú revista y darnos clic en el botón nuevo, posteriormente vemos el formulario de registro de la revista.
Condiciones de ejecución	El usuario deberá estar autenticado y tener el rol de Administrador o editor.
Entradas	<ul style="list-style-type: none"> ➤ Elegir nuevo ➤ Llenar el formulario de registro de la revista con los datos requeridos.
Resultado esperado	Luego de haber realizado el proceso de registro, la revista debería estar listado.

Tabla 4.2*Prueba de caja negra Registro de autores*

Caso de prueba: Registro de autores	
Descripción	El registro se realiza primeramente seleccionando en el menú autores y darnos clic en el botón nuevo, posteriormente vemos el formulario de registro del autor de la revista.
Condiciones de ejecución	El usuario deberá estar autenticado y tener el rol de Administrador o editor.
Entradas	<ul style="list-style-type: none"> ➤ Elegir nuevo. ➤ Seleccionar el tipo de formación. ➤ Seleccionar el género.

	➤ Llenar el formulario de registro con los datos del autor.
Resultado esperado	Luego de haber realizado el proceso de registro, el autor debería estar listado.

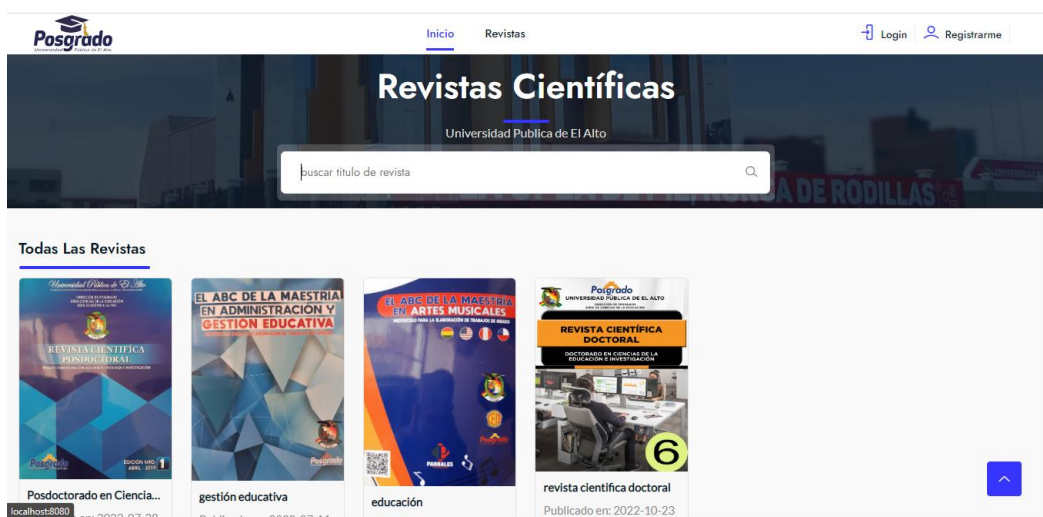
4.2.3. Pruebas de estrés

El objetivo de las pruebas de estrés es saturar el sistema web hasta un punto de quiebre, para estas pruebas se utilizó la herramienta Apache JMeter 5.5. Antes de poner en marcha el sistema web.

A continuación se tienen los resultados.

Figura 4.3

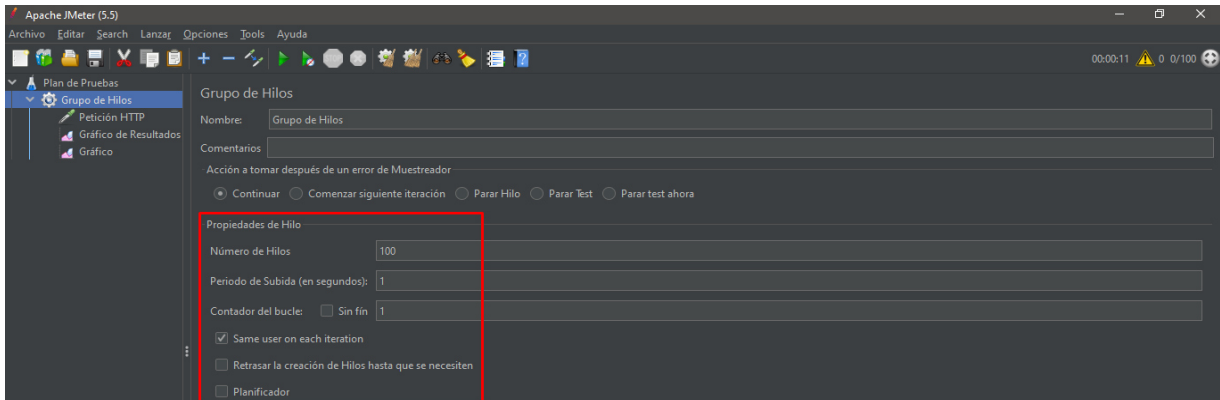
Prueba de estrés de la página de inicio de los usuarios lectores



Nota: Se muestra la página de inicio que se sometió a pruebas de estrés.

Figura 4.4

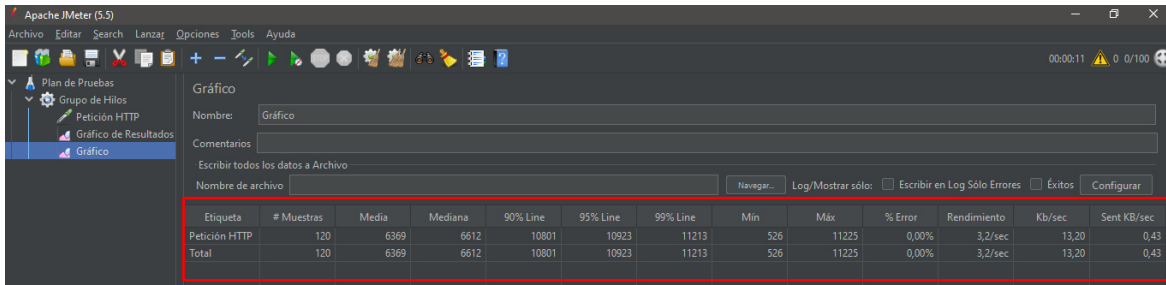
Parámetros de peticiones 100 usuarios virtuales



Nota: Se muestra los parámetros de configuración de 100 usuarios virtuales.

Figura 4.5

Resultados obtenidos de la prueba con 100 peticiones



Nota: Se muestra los resultados de la prueba de estrés a la página de inicio.

Usuarios virtuales: 100

Tiempo de respuesta o rendimiento: 3.2/sec

Solicitudes realizadas Max: 11225

Fallas HTTP: 0

4.3. SEGURIDAD SEGUN ISO 27000 (ISO/IEC 27001, ISO/IEC 27002)

El objetivo de la norma ISO 27001 es asegurar la inclusión de controles de seguridad y validación de datos en la adquisición y el desarrollo de los sistemas de información. Definir y documentar las normas y procedimientos que se aplicarán durante el ciclo de vida del software como: Definición y análisis, Desarrollo, Pruebas, Despliegue y Mantenimiento.

El objetivo de la norma ISO 27002 es que la empresa tenga conocimiento preciso sobre todos los activos que posee como parte importante de la administración de riesgos. Ejemplo:

- **Seguridad en los recursos de la información:** bases de datos y archivos, documentación de sistemas, manuales de usuario, material de capacitación, procedimientos operativos o de soporte, planes de continuidad y contingencia, información archivada, etc.
- **Seguridad de recursos de software:** software de aplicaciones, sistemas operativos, herramientas de desarrollo y publicación de contenidos, utilitarios, etc.
- **Activos físicos:** equipamiento informático, equipos de comunicaciones, medios, otros equipos técnicos, mobiliario, lugares de emplazamiento, etc.
- **Servicios:** servicios informáticos y de comunicaciones, utilitarios generales.

A continuación se desarrolla los controles propuestos según las normas propuestas.

1.1.1 Seguridad de la Base de Datos

Se usó como Base de datos Postgresql, para su resguardo se realizó los siguientes puntos:

- Cuando una acción del usuario en el sistema requiere o solicita algunos registros de la base datos, existe una conexión segura para esta acción.
- Para la seguridad de datos del sistema los usuarios ingresan con correo electrónico y contraseña de acceso, según su rol de acceso.

La información en una empresa o institución es muy valiosa, por lo tanto, su resguardo es fundamental, la conexión a la base de datos y el cierre de la conexión es de forma automática.

1.1.2 Seguridad de Autenticación y de usuario

Se refiere al control de sesión o verificación de la autenticación de un usuario registrado con anterioridad.

Cuando el usuario ingresa la contraseña, esta no se muestra en pantalla, además la contraseña de cada usuario esta encriptada mediante el uso de funciones propias del framework codeigniter. Adicionalmente se utilizó la librería firebase jwt (json web tokens) que genera una clave única (Token) para los usuarios que ingresar al sistemas.

Figura 4.6

Autenticación en el lado del backend

```
1 </php
2
3 namespace App\Controllers;
4
5 use App\Models\UsuarioModel;
6 use CodeIgniter\API\ResponseTrait;
7 use Config\Services;
8 use Firebase\JWT\JWT;
9
10 class Auth extends BaseController
11 {
12     use ResponseTrait;
13
14     public function __construct()
15     {
16         helper('secure_password');
17     }
18
19     public function login()
20     {
21         try {
22             $DatosFormulario = $this->request->getJSON();
23             $usuarioModel = new UsuarioModel();
24             $username = base64_decode($DatosFormulario->username);
25             $password = base64_decode($DatosFormulario->password);
26             $validateUsuario = $usuarioModel->loginEmail($username);
27             if ($validateUsuario == null)
28                 return $this->failNotFound('Usuario no encontrado');
29             if (verifyPassword($password, $validateUsuario['usuario_contrasena'])) :
30                 $datos = $usuarioModel->UsuarioRolesId($validateUsuario['rol_id']);
31                 $cantidad = $usuarioModel->UsuarioCartId($validateUsuario['usuario_id']);
32                 $jwt = $this->generateJWT($datos);
33                 $DataRespond = [
34                     'token' => $jwt,
35                     'status' => 'success',
36                     'rol_nombre' => $datos->rol_nombre,
37                     'userdetails' => [
38                         'uid' => $datos->usuario_id,
39                         'username' => $datos->usuario_nombre,
40                     ],
41                     'cart' => $cantidad,
42                 ];
43                 return $this->respond($DataRespond, 201);
44             else :
45                 return $this->failValidationErrors('Contraseña invalida');
46             endif;
47         } catch (\Exception $e) {
48             return $this->failServerError('Ha ocurrido un error en el servidor');
49         }
50     }
51 }
```

Nota: Se muestra la función de inicio de sesión de los usuarios

Figura 4.7

Uso de la librería de jwt

```
51
52     protected function generateJWT($usuario)
53     {
54         $key = Services::getSecretKey();
55         $time = time();
56         $payload = [
57             'aud' => base_url(),
58             'iat' => $time, //como entero el tiempo,
59             'exp' => $time + 48 * 3600, //como entero el tiempo cuando expira el token
60             'data' => [
61                 'id' => $usuario->usuario_id,
62                 'username' => $usuario->usuario_nombre,
63                 'estado' => $usuario->usuario_estado,
64                 'rol_id' => $usuario->rol_id,
65                 'rol_nombre' => $usuario->rol_nombre
66             ]
67         ];
68
69         $jwt = JWT::encode($payload, $key, 'HS256');
70         return $jwt;
71     }
72
```

Nota: Se muestra la función generar el Token jwt.

1.1.3 Seguridad de la Aplicación

Para el acceso a la sistema Web se los hace mediante roles y permisos que son asignados al momento de la creación de un usuario.

Para más seguridad en este aspecto el framework Codeigniter nos ofrece el uso de CSRF o Middlewares que son una serie de funciones que se ejecuta antes o después de cada petición HTTP. Dentro del Sistema Web se utilizó en las rutas para filtrar a los usuarios permitidos, las peticiones HTTP permitidas y endpoint (punto final) es una url de consulta.

Figura 4.8

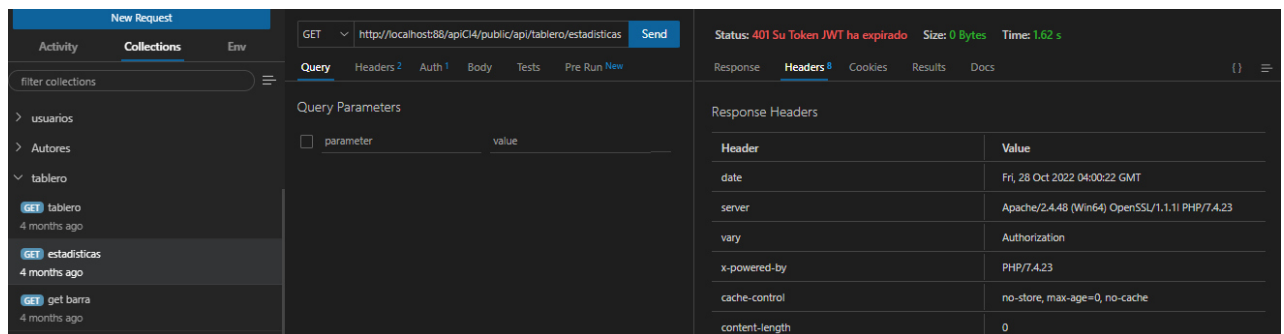
Seguridad en rutas http

```
apiC4 > app > Config > Routes.php > ...
38 $routes->get('/', 'Home::index');
39 // publico inicio de sesion
40 $routes->post('auth/login', 'Auth::login', ['filter' => 'cors']);
41 $routes->post('auth/restablecer', 'Auth::restablecer', ['filter' => 'cors']);
42 $routes->post('auth/resetPassword', 'Auth::resetPassword', ['filter' => 'cors']);
43 // publico inicio
44 $routes->get('inicio', 'InicioPublico::index', ['filter' => 'cors']);
45 $routes->post('inicio/buscar', 'InicioPublico::buscar', ['filter' => 'cors']);
46 // publico revistas
47 $routes->get('revistas', 'RevistasPublico::index', ['filter' => 'cors']);
48 $routes->post('revistas/buscar', 'RevistasPublico::buscar', ['filter' => 'cors']);
49 // publico detalle de revista
50 $routes->get('revista-detalle/id/{:segment}', 'RevistasPublico::publicacionId/$1', ['filter' => 'cors']);
51 // publico registro de usuario
52 $routes->post('registro', 'NuevoUsuario::nuevoUsuario', ['filter' => 'cors']);
53 // *****
54
55 $routes->group('api', ['namespace' => 'App\Controllers\API', 'filter' => 'authFilter', 'filter' => 'cors'], function ($routes) {
56     // LOS ADMINISTRADORES TIENEN ACCESO
57     $routes->get('tablero/estadisticas', 'Tablero::index');
58     $routes->get('tablero/getBarras/{:num}', 'Tablero::barras/$1');
59     // administrar usuario
60     $routes->get('usuarios', 'Usuarios::index');
61     $routes->get('usuarios/id/{:num}', 'Usuarios::id/$1');
62     $routes->post('usuarios/create', 'Usuarios::create');
63     $routes->get('usuarios/show/{:num}', 'Usuarios::show/$1');
64     $routes->post('usuarios/update/{:num}', 'Usuarios::update/$1');
65     $routes->post('usuarios/delete/{:num}', 'Usuarios::delete/$1');
66     $routes->post('usuarios/desactivar/{:num}', 'Usuarios::desactivarEstado/$1');
67     $routes->post('usuarios/buscar', 'Usuarios::buscar');
68 }
69 // administrar categoria
70 $routes->get('categoria', 'Categoria::index');
71 $routes->post('categoria/create', 'Categoria::create');
72 $routes->post('categoria/show/{:num}', 'Categoria::show/$1');
```

Nota: Se muestra las rutas protegidas.

Figura 4.9

Prueba de seguridad http



Nota: El error 401 indica que la solicitud no se ha procesado debido a la falta de credenciales de autenticación "Token" válidas para el recurso de destino.

1.1.4 Seguimiento de acciones

Además de limitar las acciones del usuario dentro del Sistema Web según roles y permisos, también se hace seguimiento de las acciones del usuario y estas se almacenan en la base de datos con el propósito de estar preparados para una auditoria.

4.4. METRICA DE CALIDAD DEL SOFTWARE

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las ocho características de calidad que se muestran en la siguiente figura.

Figura 4.10

Calidad del producto software categorías y subcategorías



Nota: Extraída de la norma ISO/IEC 25010.

4.4.1. Selección de las características y subcaracterísticas de calidad interna más relevantes

Tabla 4.3

Características de calidad interna seleccionadas

CARACTERÍSTICAS DE CALIDAD INTERNA		
Características	Nivel de importancia	Motivo de selección
Adecuación funcional	M	Se califica con valor de importancia M porque es necesario evaluar que las funciones codificadas estén de acuerdo a los requerimientos especificados para el sistema.
Fiabilidad	M	Se califica con valor de importancia M porque es necesario evaluar que las funciones codificadas estén de acuerdo a los requerimientos especificados para el sistema cuando éstas son sometidas a ciertas condiciones y periodos de tiempo determinados.
Eficiencia en el desempeño	M	Se califica con valor de importancia M porque a nivel de código es necesario evaluar el rendimiento del sistema tomando en cuenta los recursos que serán utilizados.
Facilidad de uso	M	Se califica con valor de importancia M porque es necesario evaluar que el código sea entendido, aprendido y usado por cualquier programador.

Seguridad	A	Se califica con valor de importancia A porque es necesario que en el código existan funciones que se encargue de la protección de los datos e información.
Compatibilidad	M	Se califica con valor de importancia M porque es muy necesario evaluar que el sistema lleve a cabo sus funciones normales mientras intercambia información y comparte el mismo entorno con otro producto software.
Mantenibilidad	A	Se califica con valor de importancia A porque es muy necesario que el código sea modificado o actualizado por cualquier programador de acuerdo a las necesidades correctivas.
Portabilidad	NA	Se califica con valor de importancia NA porque no aplica realizar la evaluación a un producto de tipo página web.

Nota: Asignando nivel de importancia a las características de la calidad interna de la norma ISO/IEC 25010.

Tabla 4.4

Subcaracterísticas y atributos de calidad interna seleccionadas

SUBCARACTERÍSTICAS Y ATRIBUTOS DE CALIDAD INTERNA			
Características	Subcaracterísticas	Nivel de importancia	Motivo de selección
Adecuación funcional	Complejidad funcional	A	Se califica con valor de importancia A porque es muy necesario evaluar que las funciones codificadas cubran todas las tareas determinadas por el usuario.
Fiabilidad	Tolerancia a fallos	M	Se califica con valor de importancia M porque es necesario evaluar que ciertas funciones codificadas permitan que al sistema se opere cuando se presenten fallos.
Eficiencia en el desempeño	Utilización de recursos	M	Se califica con valor de importancia M porque es necesario evaluar que se utilice adecuadamente la cantidad necesaria de código.
Facilidad de uso	Capacidad de reconocer su adecuación	M	Se califica con valor de importancia M porque es necesario evaluar si las funciones codificadas son entendibles para cualquier programador.

	Capacidad de ser entendido	M	Se califica con valor de importancia M porque es necesario evaluar si en el código existen funciones evidentes para cualquier programador.
	Operatividad	M	Se califica con valor de importancia M porque es necesario evaluar si existen funciones para que al sistema se lo pueda operar con facilidad.
	Protección frente a errores de usuarios	M	Se califica con valor de importancia M porque es necesario evaluar si existen elementos de entrada que son validados en el código fuente.
Seguridad	Confidencialidad	A	Se califica con valor de importancia A porque es necesario evaluar, ya que no existe un sistema externo encargado de la protección de los datos e información.
	Autenticidad	A	Se califica con valor de importancia A porque es muy necesario evaluar si existen métodos de autenticación en el código fuente.
Compatibilidad	Co - existencia	M	Se califica con valor de importancia M porque es necesario evaluarlo.
	Interoperabilidad	M	Se califica con valor de importancia M porque es necesario evaluarlo.
Mantenibilidad	Modularidad	A	Se califica con valor de importancia A porque es muy necesario evaluar si existe afectación de otras funciones en caso de modificar en código.
	Reusabilidad	A	Se califica con valor de importancia A porque es muy necesario evaluar si el código puede ser reutilizado.

Portabilidad	Adaptabilidad	NA	Se califica con valor de importancia NA porque no aplica realizar la evaluación a un producto de tipo página web.
---------------------	---------------	----	---

Nota: Asignando nivel de importancia a las subcaracterísticas de la calidad interna de la norma ISO/IEC 25010.

4.4.2. Métricas de calidad interna seleccionadas

Tabla 4.5

Subcaracterísticas y atributos de calidad interna seleccionadas

MÉTRICAS PARA LA CALIDAD INTERNA			
Características	Subcaracterísticas	Métricas	Significado
Adecuación funcional	Compleitud funcional	Compleitud de la implementación funcional	Qué tan completa es la implementación de acuerdo a la especificación de requerimientos
Fiabilidad	Tolerancia a fallos	Anulación de la operación incorrecta	Cantidad de funciones implementadas con capacidad de anular operaciones incorrectas
Eficiencia en el desempeño	Utilización de recursos	Líneas de código	Cantidad de líneas de código existe por cada función implementada
Facilidad de uso	Capacidad de reconocer su adecuación	Integridad de descripción	Cantidad de funciones que son descritas como entendibles en la descripción del producto
	Capacidad de ser entendido	Funciones evidentes	Cantidad de funciones del producto que son evidentes al usuario
	Operatividad	Claridad de mensajes	Cantidad de mensajes que son auto explicativo para el usuario
	Protección contra errores del usuario	Verificación de entradas válidas	Cantidad de elementos de entrada que son validados
Seguridad	Confidencialidad	Encriptación de datos	Cantidad de elementos encriptado/ descriptados
	Autenticidad	Métodos de autenticación	Qué tan bien el sistema autentica la identidad de un sujeto o recurso

Mantenibilidad	Modularidad	Capacidad de condensación	Qué tan fuerte es la relación entre los componentes del sistema
		Acoplamiento de clases	Qué tan fuerte es la relación entre una función del sistema con otras clases implementadas
	Reusabilidad	Ejecución de reusabilidad	Cuántos elementos pueden ser reutilizados

Nota: Descripción de las métricas de la calidad interna de la norma ISO/IEC 25010.

4.4.3. Ponderación en porcentaje de las características de calidad más relevantes para la calidad interna

Tabla 4.6

Ponderación en porcentajes para la calidad interna

CARACTERÍSTICAS DE CALIDAD INTERNA			
Características	Nivel de importancia	Ponderación	Motivo de ponderación
Adecuación funcional	M	25%	Se pondera con valor de 25% porque es necesario evaluar que las funciones codificadas estén de acuerdo a los requerimientos especificados para el sistema.
Fiabilidad	M	10%	Se pondera con valor de 10% porque es necesario evaluar que las funciones codificadas estén de acuerdo a los requerimientos especificados para el sistema cuando éstas son sometidas a ciertas condiciones y periodos de tiempo determinados.
Eficiencia en el desempeño	M	15%	Se pondera con valor de 15% porque a nivel de código es necesario evaluar el rendimiento del sistema tomando en cuenta los recursos que serán utilizados.
Facilidad de uso	M	15%	Se pondera con valor de 15% porque es necesario evaluar que el código sea entendido, aprendido y usado por cualquier programador.
Seguridad	A	25%	Se pondera con valor de 25% porque es necesario que en el código existan funciones que encripten la contraseña a la hora de crear un nuevo usuario
Compatibilidad	B	10%	Se pondera con valor de 10% porque no es necesario evaluar la compatibilidad a nivel de código, por lo que será evaluada desde punto de vista externo.

Mantenibilidad	A	25%	Se pondera con valor de 25% porque es muy necesario que el código sea modificado o actualizado por cualquier programador de acuerdo a las necesidades correctivas.
Portabilidad	NA	0%	Se pondera con valor de 0% porque no aplica realizar la evaluación a un producto de tipo página web.

Nota: Ponderación de las características de la calidad interna de la norma ISO/IEC 25010.

Figura 4.11

Aplicación de la matriz de calidad para evaluar la calidad interna

Característica	Subcaracterística	Métrica	Formula	Valor deseado (umbral,etc)	Aplica	Valor obtenido(x)		Ponderacion (/10)	valor parcial total (/10)	nivel de importancia	porcentaje de importancia (%)	valor final	calidad del sistema(/10)
						A	B						
Adecuación funcional	Complettud funcional	Complettud de la implementación funcional	$X = A / B$ A = Número de funciones que están incorrectas o que no fueron implementadas B = Número de las funciones establecidas en la especificación de requisitos Dónde: B > 0	1	si	A =	0	7,00	7,00	M	25	1,75	
						B =	22						
						X =	0,00						
Fiabilidad	Tolerancia a fallos	Anulación de la operación incorrecta	$X = A/B$ A = Número de operaciones incorrectas presentadas B = Número total de funciones implementadas para anular operaciones incorrectas Dónde: B > 0	1	si	A =	0	3,00	3,00	M	10	0,30	
						B =	1						
						X =	0,00						
Eficiencia en el desempeño	Utilización de recursos	Lineas de código	$X = A$ A = Número de líneas de código	deseado:1 peor caso: >=50	si	A =	30	3,00	3,00	M	15	0,45	
						X =	30,00						
Facilidad de uso	Capacidad de reconocer su adecuación	Integridad de descripción	$X = A/B$ A = Número de funciones (o tipos de funciones) descritas como entendibles en la descripción del producto B = Número total de funciones (o tipos de funciones) Dónde: B > 0	1	si	A =	22	10,00	9,01	M	15	1,35	
	Capacidad para ser entendido	Funciones evidentes	$X = A / B$ A= Número de funciones (o tipo de funciones) evidentes al usuario B = Número total de funciones (o tipo de funciones) Dónde: B > 0	1	si	A =	20						
						B =	22						
						X =	0,91						
Operatividad	Claridad del mensaje	$X = A / B$ A= Número de mensajes implementados con explicaciones claras B = Número total de mensajes implementados Dónde: B > 0	1	si	A =	36	10,00	6,94					
					B =	36							
					X =	1,00							
Protección contra errores del usuario	Verificación de entradas válidas.	$X = A/B$ A= Número de ítems de entrada que son validados B = Número de ítems que necesitan ser validados Dónde: B > 0	1	si	A =	25	10,00	10,00	A	25	2,50	8,1	
					B =	36							
					X =	0,69							
Seguridad	Confidencialidad	Encriptación de datos	$X = A / B$ A = Número de elementos de datos encriptados/ desencriptados correctamente B = Número de elementos de datos que requiere la encriptación/ desencriptación Dónde: B > 0	1	si	A =	1	10,00	10,00	A	25	2,50	8,1
	Autenticidad	Metodos de autenticación	$X = A$ A = Número de métodos de autenticación previstos	esperado:3 peor caso:0	si	A =	1						
						X =	1,00						

Compatibilidad	Co – existencia	Co – Existencia disponible	X = A/B A = Número de entidades con las que el producto puede coexistir B = Número de entidades en el entorno de operación que requieren de coexistencia Dónde: B > 0	1	si	A = 2 B = 2 X = 1,00	10,00	7,50	B	10	0,75
	Interoperatividad	Conectividad con sistemas externos	X = A/B A = Número de interfaces implementadas con otros sistemas B = Número total de interfaces externas Dónde: B > 0	1	si	A = 1 B = 2 X = 0,50	5,00				
Mantenibilidad	Modularidad	Capacidad de condensación	X = A / B A = Número de componentes que no son afectados por cambios de otros componentes B = Número total de componentes específicos Dónde: B > 0	0	si	A = 1 B = 2 X = 0,50	5,00	4,17	A	25	1,04
		Acoplamiento de clases	X = A A = Número de relaciones que tiene una función con respecto a otras clases	deseado:1 peor caso: >=4	si	A = 1 X = 1,00	5,00				
	reusabilidad	Ejecución de reusabilidad	X = A / B A = Número de elementos reutilizados B = Número total de elementos de la biblioteca reutilizable Dónde: B > 0	1	si	A = 1 B = 4 X = 0,25	2,50				
Portabilidad	Adaptabilidad	Adaptabilidad en entorno de software	X = A/B A = Número de funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con el sistema B = Número total de funciones que han sido probadas Dónde: B > 0	0	si	A = 0 B = 22 X = 0,00	0,00	0,00	NA	0	0

Tabla 4.7

Resultado final de calidad interna

CARACTERÍSTICAS DE CALIDAD INTERNA						
Características	Valor Parcial total(/10)	Nivel de importancia	Porcentaje de importancia	Valor final	Calidad total del sistema (/10)	Calidad final en (%)
Adecuación funcional	7.00	M	25%	1.75	8.1	81 %
Fiabilidad	3.00	M	10%	0.30		
Eficiencia en el desempeño	3.00	M	15%	0.45		
Facilidad de uso	9.01	M	15%	1.35		
Seguridad	10.00	A	25%	2.50		
Compatibilidad	7.50	M	10%	0.75		
Mantenibilidad	4.17	A	25%	1.04		
Portabilidad	0	NA	0	0		

Nota: La calidad total encontrada es de **8.1**

Según la iso/iec 250010 el nivel de puntuación final es aceptable de acuerdo a la siguiente tabla.

Tabla 4.7

Niveles de puntuación final para la calidad interna, externa y en uso

Escala de medición	Niveles de puntuación	Grado de satisfacción
8.75 - 10	Cumple con los requisitos	Muy satisfactorio
5 – 8.74	Aceptable	Satisfactorio
2.75 – 4.9	Mínimamente aceptable	Insatisfactorio
0 – 2.74	Inaceptable	

4.5. ESTIMACION DE COSTO

4.5.1. Técnica de Punto Función (ISO/IEC 20926)

Esta técnica permite cuantificar el tamaño de la Aplicación Web en unidades independientes del lenguaje de programación o la metodología utilizada.

Para el cálculo de Punto Función se toma en cuenta las siguientes características:

- **Número de entradas de usuario** (EI del inglés External Inputs): Se refiere a cada entrada que proporciona datos al sistema.
- **Número de salidas de usuario** (EO, del inglés External Outputs): Se refiere a cada salida que proporciona el sistema al usuario, entre estos están: pantallas, informes, listado de datos.
- **Número de peticiones de usuario** (EQ, del inglés External Inquiries): Una petición se define como una entrada interactiva que produce la generación de alguna respuesta de software en forma de salidas interactivas.
- **Número de Archivos** (ILF, del inglés Internal Logical Files): Se cuenta archivos maestros lógico, estos pueden ser: grupo lógico de datos o un archivo independiente, tablas en la base de datos.
- **Numero de interfaces externas** (EIF, del inglés External Interface Files): son datos referenciados a otros sistemas, pero usados por el actual sistema.

Para calcular el Punto Función ajustado (PFA) se utilizará la siguiente formula dada (IFPUG) International Function Point User Group:

$$PFA = PFSA * (X + Min(Y) * \sum Fi)$$

Donde:

PFA: Punto de función ajustado.

PFSA: Resultado del conteo de puntos de función sin ajustar de la tablas de Factor de ponderación evaluada según su tipo y su complejidad.

X: valor establecido de 0.65

Min(Y): Error mínimo aceptable a la complejidad, el margen de error es igual a 0.01

Fi: Valores de ajuste de la complejidad.

Tabla 4.8

Funciones según su tipo y su complejidad

tipo	complejidad	baja	media	alta
EI	Número de Entradas externas	3 PF	4 PF	6 PF
EO	Número de Salidas externas	4 PF	5 PF	7 PF
EQ	Número de Consultas externas	3 PF	4 PF	6 PF
ILF	Número de archivos lógico interno	7 PF	10 PF	15 PF
EIF	Numero de interfaces externas	5 PF	7 PF	10 PF

Nota: Valores estándar (IFPUG) International Function Point User Group

Se tomó el tipo de complejidad media para el software.

A continuación realizamos el conteo de los módulos del sistema y los requerimientos funcionales de las mismas. Las cuales cada módulo cuenta con cierta tarea.

Módulos del sistema:

Web de administración:

Tabla 4.9

Módulo inicio de sesión

funcionalidad	tipo	Cantidad
Iniciar sesión	(EI)	1

Nota: (EI) Número de Entradas externas.

Tabla 4.10

Módulo tablero de información.

funcionalidad	tipo	Cantidad
estadísticas	(EO)	1

Nota: (EO) Número de Salidas externas.

Tabla 4.11

Módulo de usuarios

funcionalidad	tipo	Cantidad
Buscar	(EQ)	1
insertar	(EI)	1
actualizar	(EI)	1
listar	(EO)	1
Eliminar	(EI)	1
reportes	(EO)	1
cambiar estado	(EI)	1

Nota: (EI) Número de Entradas externas, (EO) Número de Salidas externas, (EQ) Número de Consultas externas.

Tabla 4.12

Módulo de roles y permisos.

funcionalidad	tipo	Cantidad
Buscar	(EQ)	1
actualizar	(EI)	1
listar	(EO)	1
reportes	(EO)	1

Nota: (EI) Número de Entradas externas, (EO) Número de Salidas externas, (EQ) Número de Consultas externas.

Tabla 4.13

Módulo de categoría.

funcionalidad	tipo	Cantidad
Buscar	(EQ)	1
insertar	(EI)	1
actualizar	(EI)	1
listar	(EO)	1
Eliminar	(EI)	1
reportes	(EO)	1
cambiar estado	(EI)	1

Nota: (EI) Número de Entradas externas, (EO) Número de Salidas externas, (EQ) Número de Consultas externas.

Tabla 4.14

Módulo de autores.

funcionalidad	tipo	Cantidad
Buscar	(EQ)	1
insertar	(EI)	1
actualizar	(EI)	1
listar	(EO)	1
Eliminar	(EI)	1
reportes	(EO)	1
cambiar estado	(EI)	1

Nota: (EI) Número de Entradas externas, (EO) Número de Salidas externas, (EQ) Número de Consultas externas.

Tabla 4.15

Módulo de revistas y publicación.

funcionalidad	tipo	Cantidad
Buscar	(EQ)	1
insertar	(EI)	1

actualizar	(EI)	1
listar	(EO)	1
Eliminar	(EI)	1
reportes	(EO)	1
cambiar estado	(EI)	1

Nota: (EI) Número de Entradas externas, (EO) Número de Salidas externas, (EQ) Número de Consultas externas.

Tabla 4.16

Módulo de proveedor.

funcionalidad	tipo	Cantidad
Buscar	(EQ)	1
insertar	(EI)	1
actualizar	(EI)	1
listar	(EO)	1
Eliminar	(EI)	1
reportes	(EO)	1
cambiar estado	(EI)	1

Nota: (EI) Número de Entradas externas, (EO) Número de Salidas externas, (EQ) Número de Consultas externas.

Tabla 4.17

Módulo de compra.

funcionalidad	tipo	Cantidad
Buscar	(EQ)	1
insertar	(EI)	1
listar	(EO)	1
reportes	(EO)	1

Nota: (EI) Número de Entradas externas, (EO) Número de Salidas externas, (EQ) Número de Consultas externas.

Tabla 4.18*Módulo de clientes.*

funcionalidad	tipo	Cantidad
Buscar	(EQ)	1
insertar	(EI)	1
actualizar	(EI)	1
listar	(EO)	1
Eliminar	(EI)	1
reportes	(EO)	1

Nota: (EI) Número de Entradas externas, (EO) Número de Salidas externas, (EQ) Número de Consultas externas.

Tabla 4.19*Módulo de ventas.*

funcionalidad	tipo	Cantidad
Buscar	(EQ)	1
insertar	(EI)	1
listar	(EO)	1
reportes	(EO)	1

Nota: (EI) Número de Entradas externas, (EO) Número de Salidas externas, (EQ) Número de Consultas externas.

Tabla 4.20*Módulo de ventas en línea.*

funcionalidad	tipo	Cantidad
Buscar	(EQ)	1
insertar	(EI)	1
listar	(EO)	1
reportes	(EO)	1
cambiar estado	(EI)	1

Nota: (EI) Número de Entradas externas, (EO) Número de Salidas externas, (EQ) Número de Consultas externas.

Web público:

Tabla 4.21

Módulo inicio.

funcionalidad	tipo	Cantidad
Listar de revistas	(EO)	1

Nota: (EO) Número de Salidas externas.

Tabla 4.22

Módulo revistas

funcionalidad	tipo	Cantidad
Buscar	(EQ)	1
Filtro categorías	(EQ)	1
Listar de revistas	(EO)	1

Nota:(EO) Número de Salidas externas, (EQ) Número de Consultas externas.

Tabla 4.23

Módulo detalle de las revistas.

funcionalidad	tipo	Cantidad
Detalle de revistas	(EO)	1
Agregar al carrito	(EI)	1
Agregar a favoritos	(EI)	1

Nota: (EI) Número de Entradas externas, (EO) Número de Salidas externas.

Tabla 4.24

Módulo inicio de sesión.

funcionalidad	tipo	Cantidad
Inicio de sesión	(EI)	1

Nota: (EI) Número de Entradas externas.

Tabla 4.25

Módulo registro de usuario.

funcionalidad	tipo	Cantidad
Inicio de sesión	(EI)	1

Nota: (EI) Número de Entradas externas.

Tabla 4.26

Módulo perfil de usuario.

funcionalidad	tipo	Cantidad
Actualizar información de perfil	(EI)	1

Nota: (EI) Número de Entradas externas.

Tabla 4.27

Módulo cambiar contraseña.

funcionalidad	tipo	Cantidad
Actualizar información de la contraseña	(EI)	1

Nota: (EI) Número de Entradas externas.

Tabla 4.28

Módulo favorito del usuario.

funcionalidad	tipo	Cantidad
Listar revistas	(EO)	1

Nota: (EO) Número de Salidas externas.

Tabla 4.29

Módulo compras del usuario.

funcionalidad	tipo	Cantidad
listar	(EO)	1
Descarga de revista	(EO)	1
Ver revistas	(EI)	1

Nota: (EI) Número de Entradas externas, (EO) Número de Salidas externas.

Tabla 4.30

Módulo carrito de compras del usuario.

funcionalidad	tipo	Cantidad
Listar las compras a realizar	(EO)	1

Nota: (EO) Número de Salidas externas.

Módulos totales: 23

Cantidad total de tipo (EI) = 36

Cantidad total de tipo (EO) = 28

Cantidad total de tipo (EQ) = 12

Cantidad total de tipo (ILF) = 25

Cantidad total de tipo (EIF) = 0

En la siguiente tabla se calcula el punto de función, los cuales miden el software desde una perspectiva del usuario, dejando de lado los detalles de programación.

Tabla 4.31

Factor de ponderación

TIPO	complejidad	Cantidad	factor de ponderación media	total
EI	Número de Entradas externas	36	4	144
EO	Número de Salidas externas	28	5	140
EQ	Número de Consultas externas	12	4	48
ILF	Número de archivos lógico interno	25	10	250
EIF	Numero de interfaces externas	0	7	0
	punto de función sin ajustar PFSA			1164

Nota: elaboración según en base a criterios (Pressman R. , 2010)

El cálculo de los valores para Fi se obtiene de los resultados de la siguiente tabla, bajo las ponderaciones descritos en la escala.

Tabla 4.32

Tabla de Valores de ajuste de la complejidad

ponderación	0	1	2	3	4	5	
Factor de ajuste	Sin importancia	Incremental	Moderado	Medio	Significativo	Esencial	FI
1.- ¿Requiere el sistema copias de seguridad y de recuperación fiable?						X	5
2.- ¿Se requiere comunicación de datos?						X	5
3.- ¿Existe funciones de procesos distribuidos?						X	5
4.- ¿Es crítico el rendimiento?			X				2
5.- ¿El sistema web será ejecutado el SO. Actual. ?					X		4
6.- ¿Se requiere una entrada interactiva para el sistema?					X		4
7.- ¿Se requiere que el sistema tenga entradas a datos con múltiples ventanas?						X	5
8.- ¿Se actualiza los archivos de forma interactiva?						X	5
9.- ¿Son complejas las entradas, salidas, los archivos o las peticiones?				X			3
10.- ¿Es complejo el procesamiento interno del sistema?					X		4
11.- ¿Se ha diseñado el código para ser reutilizado?						X	5
12.- ¿Se ha diseñado el sistema para facilitar al usuario el trabajo y ayudarlos a encontrar la información?						X	5

13.- ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizados por el usuario?						X	5
Factor de ajuste	$\sum(Fi) = 57$						

Calculando el punto de función ajustado

Remplazado en la siguiente ecuación:

$$\mathbf{PFA = PFSA * (X + Min(Y) * \sum Fi)}$$

Donde:

$$PFSA = 1164$$

$$X = 0.65$$

$$Min(Y) = 0.01$$

$$\sum Fi = 57$$

$$\mathbf{PFA = 1164 * (0.65 + 0.01 * 57)}$$

$$\mathbf{PFA = 1164 * 1.22}$$

$$\mathbf{PFA = 1.420,08}$$

4.5.2. COCOMO II

El Modelo de Construcción de Costo COCOMO (Constructive Cost Model), se utiliza para la estimación de costos de un software. Incluye tres sub-modelos, cada uno ofrece un nivel de detalle y aproximación. Estos modelos utilizan las tres siguientes ecuaciones:

$$E = a (KLDC)^b * m(X), \text{ en personas/mes}$$

$$Tdev = c (E)^d, \text{ en meses}$$

$$P = \frac{E}{Tdev}, \text{ en personas}$$

E: Esfuerzo requerido por el proyecto, en meses.

Tdev: Tiempo requerido por el proyecto, en meses.

P: Número de personas requeridos por el proyecto.

a, b, c y d: Constantes con valores definidos, según cada sub-modelo.

KLDC: Cantidad de líneas de código, en miles.

m(X): Multiplicador que depende de 15 atributos.

A la vez cada modelo se divide en modos, estos modos son:

- **Modo orgánico:** es un pequeño grupo de programadores experimentados desarrollando software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas docenas de miles (medio).
- **Modo semi – libre o semi – acoplado:** Corresponde a un esquema intermedio entre el orgánico y el rígido, el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.
- **Modo rígido o empotrado:** El proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único, siendo difícil basarse en la experiencia puesto que puede no haberla.

Tabla 4.33

Coficiente a y c y los exponentes b y d

modos	a	b	c	d
Orgánico	2.4	1.05	2.2	0.38
Semi - acoplado	3.0	1.12	2.5	0.35
Empotrado	3.80	1.20	2.5	0.32

4.5.2.1. Costo del Desarrollo de Software

Punto de función no ajustado encontrado es **PFSA = 1164**

Este resultado se debe convertir a KLDC (Kilos de Líneas de Código), para ello se utiliza la siguiente la tabla.

Tabla 4.34

Conversión de puntos de Función a KDLC

LENGUAJE	NIVEL	FACTOR LDC/PF
C	2.5	128
ANSI BASIC	5	64
JAVA	6	53
PL/I	4	80
ANSI COBOL 74	3	107
VISUAL BASIC	7.00	46
ASP	9.00	36
PHP	11.00	29
VISUAL C++	9.50	34

Nota: (Pressman R. , 2010)

Se tomara como dato el factor LDC/PF que es 29

Calculando las líneas de código en la siguiente ecuación tenemos:

$$\text{LDC} = \text{PFSA} * \text{Factor LDC / PF}$$

$$\text{LDC} = 1164 * 29$$

$$\text{LDC} = 33756$$

Calculando el número de líneas distribuidas en el sistema KLCD se tiene:

$$\text{KLDC} = \text{LDC} / 1000$$

$$KLDC = 33756/1000$$

$$KLDC = 33.756$$

Calculando el esfuerzo necesario para la programación del sistema.

$$E = a * (KLDC)^b, \text{ en personas /mes}$$

Donde:

E: Es el esfuerzo expresado en personas por mes.

a y b: Constantes con valores definidos.

KLDC: 33.756

Como nuestro proyecto es del tipo Orgánico, utilizaremos $a = 2.4$ y $b = 1.05$.
Reemplazando estos valores en la ecuación, tenemos:

$$E = 2.4 * (33.756)^{1.05}$$

$$E = 40.25 \left(\frac{\text{personas}}{\text{mes}} \right)$$

Calculando el tiempo requerido **Tdev** con $c = 2.2$ y $d = 0.38$ tenemos:

$$Tdev = c (E)^d, \text{ en meses}$$

$$Tdev = 2.2 * (40.25)^{0.38}$$

$$Tdev = 8,95 \cong 9 \text{ (meses)}$$

El proyecto deberá tener un desarrollo aproximadamente de 9 meses.

Calculando el personal requerido en este caso el Número de programadores para el desarrollo es:

$$P = \frac{E}{Tdev}, \text{ en personas}$$

$$P = \frac{40.25}{8.95}$$

$$P = 4.49 \cong 4 \text{ (programadores)}$$

El salario promedio de un programador junior oscila entre los 2000 Bs y 4000 Bs., en nuestro caso tomaremos un promedio con un valor de 3000 Bs, a partir de este monto podemos calcular el costo del software desarrollado por mes:

$$\text{Costo del software desarrollado por mes} = \text{Persona} * \text{salario de un programador}$$

$$\text{Costo del software desarrollado por mes} = 4 * 3000$$

$$\text{Costo del software desarrollado por mes} = 12000 \text{ bs}$$

Como el desarrollo de software se lo estima en 9 meses tendremos:

Costo total de desarrollo = Costo del software desarrollado por mes * Numero de meses

$$\text{Costo total de desarrollo} = 12000 * 9$$

$$\text{Costo total de desarrollo} = 108000 \text{ bs}$$

Tabla 4.35

Resumen de Cocomo II

Símbolo	Descripción	Valor	Unidad
E	Esfuerzo	40.25	Personas/Mes
Tdev	Tiempo	9	Meses
P	Persona	4	Persona
CM	Costo Mes	12000	Bolivianos
CT	Costo total	108000	Bolivianos

Nota: (E) esfuerzo de 4 personas, Tdev tiempo de 9 meses, (CT) costo total 108000 Bs.

4.5.2.2. Costo de Implementación del software

Las herramientas de software son PHP y PostgreSQL y Vue js, son de uso libre y las mismas que son utilizadas por muchas empresas por ser software libre.

Así que los únicos gastos son los detallados a continuación:

Tabla 4.36*Costos de Implementación*

DETALLE	IMPORTE (Bs)
Dominio	0
servidor	0
TOTAL	0

4.5.2.3. Costo de Elaboración del software

Los costos de elaboración del proyecto se refieren principalmente a los gastos que se realizan a lo largo de las diferentes fases de la metodología UWE, estos son detallados a continuación:

Tabla 4.37*Costos de Elaboración*

DETALLE	IMPORTE (Bs)
Análisis y diseño del proyecto	1000
Material de escritorio	250
Internet	1500
Otros	200
TOTAL	2950

4.5.2.4. Costo Total del software

El costo total del software se lo obtiene de la sumatoria del costo de: desarrollo, implementación y elaboración del proyecto:

Tabla 4.38

Costos de Elaboración

DETALLE	IMPORTE (Bs)
Costo de desarrollo	108000 bs
Costo de implementación	0 bs
Costo de elaboración	2950 bs
TOTAL	110950 bs

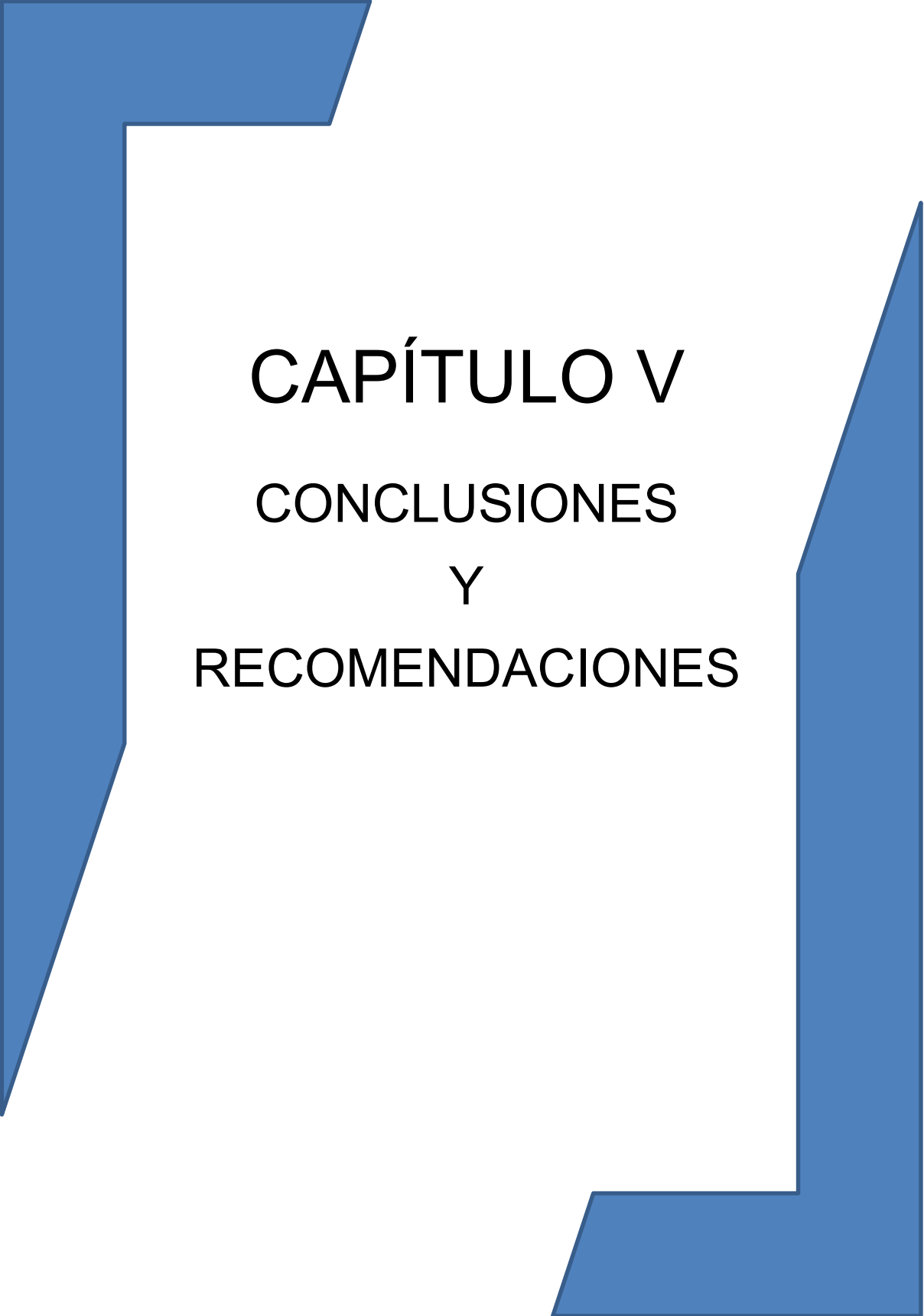
Nota: Por lo tanto, el costo total del proyecto es 110950 Bs.

4.6. ANTES Y DESPUES DEL SISTEMA

Tabla 4.39

Antes y después del sistema

ANTES	DESPUES - SISTEMA
Reporte 1hrs.	Reporte 2.3 seg.
Registros en Excel de las compras de revistas.	Registros centralizados en base de datos.
Publicación con afiches	Publicaciones con más alcance en internet.

A large, blue, stylized frame surrounds the text. The frame is composed of several geometric shapes: a top horizontal bar with a slanted right edge, a left vertical bar with a slanted bottom edge, a right vertical bar with a slanted top edge, and a bottom horizontal bar with a slanted left edge. The text is centered within this frame.

CAPÍTULO V
CONCLUSIONES
Y
RECOMENDACIONES

5.1. INTRODUCCIÓN

En este capítulo se dará a conocer algunas observaciones y recomendaciones se deben tener en cuenta al terminar el sistema.

5.1.1. CONCLUSIONES

Una vez finalizado el proyecto de grado titulado “Sistema de información web para la administración y publicación digital de revistas científicas” para la dirección de Posgrado de la Universidad Pública de El Alto, se logró alcanzar el objetivo principal planteado, cumpliendo con las necesidades requeridas.

Tomando en cuenta los objetivos planteados se llega a las siguientes conclusiones:

- Se desarrolló el sistema web en su totalidad, con todos los módulos requeridos de acuerdo al análisis de la situación actual de la dirección de posgrado.
- La utilización de la metodología UWE fue práctica para el desarrollar e implementar del sistema Web basado en los requerimientos funcionales y no funcionales.
- Se aplicó las pruebas al sistema usando las técnicas de prueba al software como la caja blanca y caja negra.
- Se aplicó la ISO/IEC 25000 para la calidad del software obteniendo 81% de satisfacción.
- Se aplicó la evaluación de costos COCOMO II obteniendo los siguiente:

Esfuerzo = 4 personas, Tiempo = 9 meses Costo = 108000 bs.

5.1.2. RECOMENDACIONES

A partir del presente trabajo se propone las siguientes recomendaciones, con el fin de buscar el mejoramiento del sistema:

- Se recomienda al usuario administrador, dar de alta a los usuarios como editor y encargado de venta, con los privilegios mínimos y contraseñas con cantidad mínima de caracteres, así como de discreción después de su asignación.

- Se recomienda realizar copias de seguridad de la base de datos periódicamente, para la restauración si se lo requiere.
- Se recomienda hacer una actualización a las ventas en línea ya que el modo de pago es mediante comprobante de pago de banco.
- Se recomienda actualizar el envío de archivos al correo. por un gestor de contenido de archivos como (mega, google drive, etc.). Por qué el envío depende mucho de la velocidad conexión de internet.

BIBLIOGRAFÍA

- Acibeiro, M. (07 de marzo de 2022). *lucushost*. Obtenido de lucushost: <https://www.lucushost.com/blog/que-es-node-js/>
- ACyT, A. C. (2010). *TEBEOSFERA*. Obtenido de TEBEOSFERA: https://www.tebeosfera.com/conceptos/publicacion_digital.html
- Adriana Gómez, M. d. (2021). *UN MODELO DE ESTIMACION DE PROYECTOS DE SOFTWARE*. abril: <https://blogadmi1.files.wordpress.com/2010/11/cocom0llfull.pdf>.
- Aguado, L. E. (5 de abril de 2022). *wikipedia.org*. Obtenido de wikipedia.org: https://es.wikipedia.org/wiki/Revista_cient%C3%ADfica
- Alvarez, c. ., (2017). *Implementacion iso- 27001*.
- Andalucía. (22 de 03 de 2021). *Junta de Andalucía*. Obtenido de Junta de Andalucía: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/264>
- Bembibre, C. (febrero de 2013). *Definición ABC*. Obtenido de Definición ABC: <https://www.definicionabc.com/general/publicacion.php>
- C.E.S.U.M.A. (23 de septiembre de 2022). *UNIVERSIDAD INTERNACION DEL TALENTO*. Obtenido de UNIVERSIDAD INTERNACION DEL TALENTO: <https://www.cesuma.mx/blog/que-es-la-informacion-en-informatica-y-otras-ciencias.html>
- Carlos Yañez. (08 de noviembre de 2020). *CEAC. Planeta Formación y Universidades*. Obtenido de CEAC. Planeta Formación y Universidades: <https://www.ceac.es/blog/tipos-de-seguridad-informatica>
- Catedracosgaya. (30 de 03 de 2021). *www.catedracosgaya.com.ar*. Obtenido de www.catedracosgaya.com.ar: <http://www.catedracosgaya.com.ar/tipoblog/2015/como-nacio-la-revista/#:~:text=naci%C3%B3%20la%20revista%3F-Se%20piensa%20que%20la%20revista%20naci%C3%B3%20hacia%20finales%20del%20siglo,tipo%20de%20lector%20en%20particular>.
- Cecilia Bembibre, C. (febrero de 2013). *Definición ABC* . Obtenido de Definición ABC : <https://www.definicionabc.com/general/publicacion.php>
- Civantos, M. (21 de mayo de 2021). *tribalyte*. Obtenido de tribalyte: <https://tech.tribalyte.eu/blog-que-es-una-api-rest>
- CodeIgniter. (05 de septiembre de 2022). *CodeIgniter4*. Obtenido de CodeIgniter4: https://codeigniter.com/user_guide/intro/index.html
- Crear Imagen Agencia Digital. (17 de 04 de 2021). *paginaswebbogota*. Obtenido de [paginaswebbogota](https://www.paginaswebbogota.pro): <https://www.paginaswebbogota.pro/administracion-web/#:~:text=El%20concepto%20de%20administraci%C3%B3n%20web,funcionamiento%20de%20una%20p%C3%A1gina%20web>.
- Denzer, P. (2002). *PostgreSQL*.
- EcuRed. (25 de septiembre de 2022). *EcuRed*. Obtenido de EcuRed: <https://www.ecured.cu/JavaScript>

EcuRed. (25 de septiembre de 2022). *EcuRed*. Obtenido de EcuRed: <https://www.ecured.cu/PostgreSQL>

EduRed. (2022). *EduRed*. Obtenido de EduRed: <https://www.ecured.cu/MagicDraw>

Estatuto Orgánico Upea. (10 de 11 de 2017). *abc la upea*. Obtenido de <https://upea.reyqui.com/2017/11/upea-estatuto-organico-reglamentos.html>

Etecé. (22 de octubre de 2021). *concepto*. Obtenido de concepto: <https://concepto.de/sistema/>

Florencia Ucha. (septiembre de 2008). *Definición ABC*. Obtenido de Definición ABC: <https://www.definicionabc.com/comunicacion/revista.php>

Friends, A. (08 de 09 de 2022). *Apache Friends*. Obtenido de Apache Friends: <https://www.apachefriends.org/es/index.html>

Gaikwad, S. S. (2019). A review paper on bootstrap framework. En S. S. Gaikwad, *A review paper on bootstrap framework*.

Galiano, I. (octubre de 2012). *V-INF-3T*. Obtenido de V-INF-3T: <http://elproyectodeluisgaliano.blogspot.com/>

Galiano, I. (noviembre de 2012). *V-INF-3T*. Obtenido de V-INF-3T: <http://elproyectodeluisgaliano.blogspot.com/>

Gardey, J. P. (2019). *definicion.de*. Obtenido de definicion.de: <https://definicion.de/publicacion/>

Gdr, S. (22 de 03 de 2021). *Gdr soluciones*. Obtenido de Gdr soluciones: <https://facturacion.gdrsoluciones.com/knowledgebase/26/iQue-es-APACHE.html?language=spanish>

Gómez, A. M. (2021). *UN MODELO DE ESTIMACION DE PROYECTOS DE SOFTWARE*. abril: <https://blogadmi1.files.wordpress.com/2010/11/cocom0llfull.pdf>.

González, J. (17 de Agosto de 2018). *OpenWebinars*. Obtenido de OpenWebinars: <https://openwebinars.net/blog/caracteristicas-importantes-de-postgresql/>

González, J. (17 de Agosto de 2018). *OpenWebinars*. Obtenido de OpenWebinars: <https://openwebinars.net/blog/caracteristicas-importantes-de-postgresql/>

Google Sites. (17 de abril de 2021). *Google Sites*. Obtenido de Google Sites: <https://sites.google.com/site/sistemasdecalidadenti/iso-iec-9126>

Group, T. p. (2021). *Hypertext Pre-Processor "PHP"*. Obtenido de Group, The php: [https://www.php.net/manual/es/intro-what-is.php#:~:text=PHP%20\(acr%C3%B3nimo%20recursivo%20de%20PHP,puede%20ser%20incrustado%20en%20HTML](https://www.php.net/manual/es/intro-what-is.php#:~:text=PHP%20(acr%C3%B3nimo%20recursivo%20de%20PHP,puede%20ser%20incrustado%20en%20HTML).

ISO25000, i. (30 de 08 de 2022). *NORMA ISO 25000*. Obtenido de NORMA ISO 25000: <https://iso25000.com/index.php/normas-iso-25000>

ISO27000. (22 de abril de 2021). *normasiso net*. Obtenido de normasiso net: <https://www.normasiso.net/wp-content/uploads/2016/10/iso-27000.pdf>

Ludwig Maximilians. (10 de agosto de 2016). *UWE – UML-based Web Engineering*. Obtenido de UWE – UML-based Web Engineering: <https://uwe.pst.ifi.lmu.de/teachingTutorialRequirementsSpanish.html>

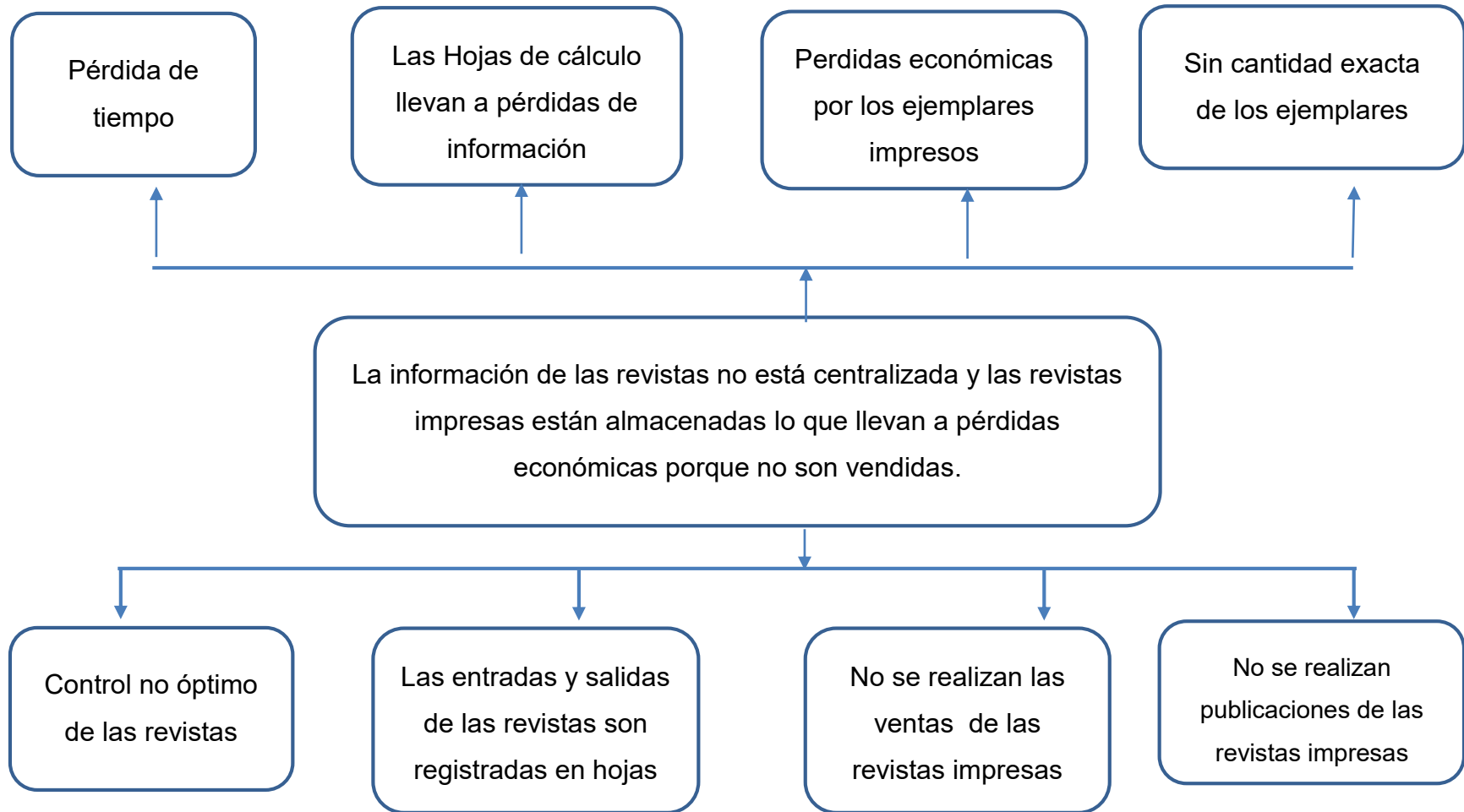
- Mamani, M. L. (26 de noviembre de 2019). *encora*. Obtenido de encora: <https://www.encora.com/es/blog/qu%C3%A9-es-vue.js-y-c%C3%B3mo-lo-usamos>
- Maximilians, L. (10 de agosto de 2016). *UWE – UML-based Web Engineering*. Obtenido de UWE – UML-based Web Engineering: <https://uwe.pst.ifi.lmu.de/teachingTutorialRequirementsSpanish.html>
- Micro Focus. (4 de octubre de 2022). *Micro Focus*. Obtenido de Micro Focus: <https://www.microfocus.com/es-es/what-is/performance-testing>
- Mora, S. L. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. Editorial Club Universitario.
- Muñoz, P. L. (2018). *Introducción, conceptos, creación de sitios Web, diseño de páginas WEB*,. Lima, Perú.
- Murillo Danny, S. D. (11 de 01 de 2016). *UTP-Ridda2*. Obtenido de <https://ridda2.utp.ac.pa/handle/123456789/3104>
- Narvaez, P. B. (2016). *EXPERIENCIA DE DESARROLLO DE UNA APLICACIÓN WEB UTILIZANDO LA METODOLOGIA UWE*. ECUADOR.
- NodeJs. (08 de 09 de 2022). *NodeJs*. Obtenido de NodeJs: <https://nodejs.org/es/>
- Nora Koch, A. K. (10 de junio de 2002). *scholar google*. Obtenido de scholar google: https://scholar.google.com/citations?user=GEh-NxkAAAAJ&hl=de#d=gs_md_cita-d&u=%2Fcitations%3Fview_op%3Dview_citation%26hl%3Dde%26user%3DGEh-NxkAAAAJ%26citation_for_view%3DGEh-NxkAAAAJ%3Au5HHmVD_uO8C%26tzom%3D240
- Oropeza, D. K. (2018). *La competencia económica en el comercio electrónico y su protección en el sistema jurídico mexicano*. Mexico.
- Patricio Denzer. (2002). *PostgreSQL*.
- Pérez, J. E. (2019). *introduccion a JavaScript*. En J. E. Pérez, *introduccion a JavaScript*.
- Pineda, J. M. (03 de NOVIEMBRE de 2016). *coriaweb*. Obtenido de coriaweb: <https://www.coriaweb.hosting/codeigniter-cuales-algunas-ventajas/>
- PostgreSQL. (08 de 09 de 2022). *postgresql*. Obtenido de postgresql: <https://www.postgresql.org/>
- Pressman. (2010). *Ingeniería del software un enfoque práctico*. Mexico.
- Pressman, R. (2010). *Ingeniería del software un enfoque práctico*. México: McGraw Hill.
- Pressman, R. (2010). *Ingeniería del software un enfoque práctico*. México: McGraw Hill.
- Roc, I. (19 de marzo de 2019). *luis roc*. Obtenido de <https://luisroc.com/plataformas-web-soluciones-a-la-medida/>
- Rockcontent. (12 de abril de 2020). *rockcontent*. Obtenido de rockcontent: <https://rockcontent.com/es/blog/bootstrap/>
- Roger S. Pressman. (2010). *Ingeniería del software un enfoque práctico*. México: McGraw Hill.
- Ruiz, R. L. (30 de 03 de 2021). *www.catedracosgaya.com.ar*. Obtenido de <http://www.catedracosgaya.com.ar/tipoblog/2015/como-nacio-la-revista/#:~:text=naci%C3%B3%20la%20revista%3F->

,Se%20piensa%20que%20la%20revista%20naci%C3%B3%20hacia%20finales%20del%20sig
lo, tipo%20de%20lector%20en%20particular.

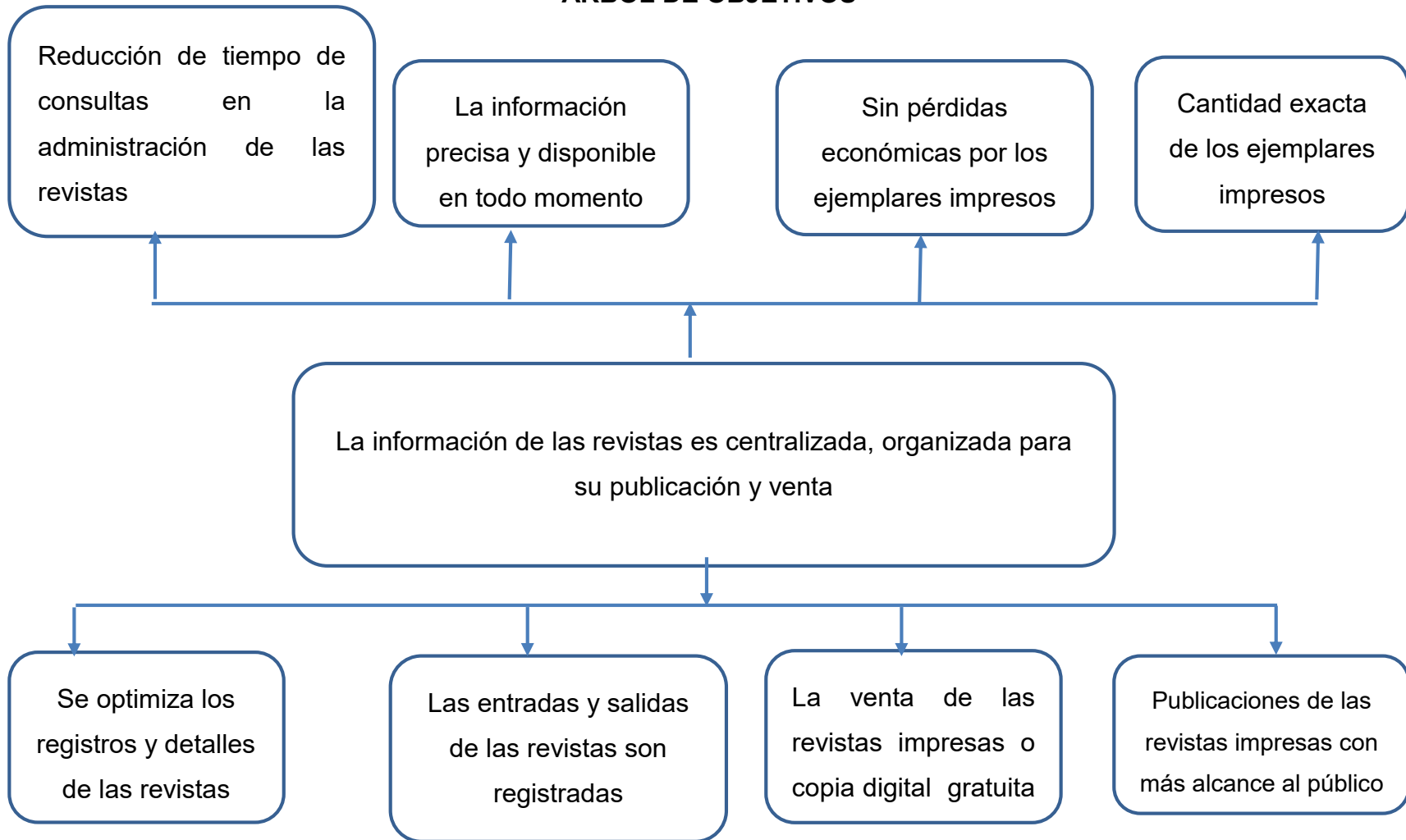
- Salamanca Ticona Ramiro. (30 de 09 de 216). *Ri-UMSA*. Obtenido de <https://repositorio.umsa.bo/handle/123456789/17662>
- Sánchez, P. J. (2015). *Pruebas de Software. Fundamentos y Técnicas*. Obtenido de upm: <https://oa.upm.es/40012/>
- Schiaffarino, A. (12 de Marzo de 2019). *Infranetworking Internacional*. Obtenido de Infranetworking Internacional: <https://blog.infranetworking.com/modelo-cliente-servidor/>
- scielo Bolivia. (13 de abril de 2021). *scielo Bolivia*. Obtenido de scielo Bolivia: <http://www.scielo.org.bo/>
- Sebastián Díaz Becerro. (17 de 04 de 2021). *Google Sites*. Obtenido de Google Sites: <https://sites.google.com/site/jorgeluiscordovadiatic/unidad-de-aprendizaje-iii/definicion-y-caracteristicas-de-plataforma-educativa>
- Sites Google. (20 de abril de 2021). *sites google*,. Obtenido de sites google,: <https://sites.google.com/site/phpinfoiutepi/-que-es-php>
- Thompson, I. (Octubre de 2008). Definición de Información. pág. 3. Obtenido de <https://www.itmerida.mx/panel/posgrado/archivos/mga/PRUEBA.pdf>
- Torres, I. M. (1996). *METODOS DE RECOLECCION DE DATOS PARA UNA*. España.
- Upea. (30 de 03 de 2021). *www.upea.bo*. Obtenido de [www.upea.bo](https://enlaupea.blogspot.com/p/historia-de-la-upea.html#.YGKIEFVKjDc): <https://enlaupea.blogspot.com/p/historia-de-la-upea.html#.YGKIEFVKjDc>
- Uriarte. (29 de marzo de 2020). Obtenido de <https://www.caracteristicas.co/sistema-informatico/>
- Uriarte. (10 de marzo de 2020). *Caracteristicas*. Obtenido de Caracteristicas: <https://www.caracteristicas.co/sistemas/>
- UTN. (06 de junio de 2015). *slideshare*. Obtenido de slideshare: <https://es.slideshare.net/omar1023/uwe-49070035>
- Vázquez, S. (05 de marzo de 2022). *digital editorial*. Obtenido de digital editorial: <https://digital-editorial.com/revistas-digitales/>
- Vuejs. (05 de octubre de 2022). *vuejs*. Obtenido de vuejs: <https://es.vuejs.org/v2/guide/>
- Webempresa. (20 de abril de 2021). *Webempresa*. Obtenido de Webempresa: <https://www.webempresa.com/hosting/que-es-servidor-apache.html>
- Wikipedia.org. (11 de Febrero de 2020). *Calidad de software*. Obtenido de Wikipedia: https://es.m.wikipedia.org/wiki/Calidad_de_software

ANEXOS

ÁRBOL DE PROBLEMAS



ÁRBOL DE OBJETIVOS



El Alto, Noviembre de 2022

Señor(a)

M.Sc.Ing. David Carlos Mamani Quispe
DIRECTOR DE LA CARRERA DE INGENIERIA DE SISTEMAS

Presente:

REF: AVAL DE CONFORMIDAD

Mediante la presente tengo el bien comunicarle mi conformidad del proyecto de grado ***“Sistema de Información web para la Administración y Publicación digital de Revistas Científicas”*** para la Dirección de Posgrado U.P.E.A. que propone el postulante **Gabriel Limachi Misme**, con cedula de identidad **9104987 L.P.** y registro universitario **14000996**, para su defensa publica, evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente:



M.Sc. Ing. Enrique Flores Baltazar

TUTOR METODOLOGICO

El Alto, Noviembre de 2022

Señor(a)

M.Sc. Ing. Enrique Flores Baltazar

TUTOR METODOLOGICO TALLER DE GRADO II
CARRERA INGENIERIA DE SISTEMAS
UNIVERSIDAD PÚBLICA DE EL ALTO

Presente:

REF.: AVAL DE CONFORMIDAD

De mi consideración.

Mediante la presente tengo el bien comunicarle mi conformidad del proyecto de grado **"Sistema de Información web para la Administración y Publicación digital de Revistas Científicas"** caso: Dirección de Posgrado U.P.E.A. que propone el postulante **Gabriel Limachi Misme**, con cedula de identidad **9104987 L.P.** y registro universitario **14000996**, para su defensa publica, evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la carrera Ingeniería de sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente:



.....
Lic. Mario Torrez Cupiticona
TUTOR REVISOR

El Alto, Noviembre de 2022

Señor(a)

M.Sc. Ing. Enrique Flores Baltazar

TUTOR METODOLOGICO TALLER DE GRADO II
CARRERA INGENIERIA DE SISTEMAS
UNIVERSIDAD PÚBLICA DE EL ALTO

Presente:

REF.: AVAL DE CONFORMIDAD

De mi consideración.

Mediante la presente tengo el bien comunicarle mi conformidad del proyecto de grado **"Sistema de Información web para la Administración y Publicación digital de Revistas Científicas"** caso: Dirección de Posgrado U.P.E.A. que propone el postulante **Gabriel Limachi Misme**, con cedula de identidad **9104987 L.P.** y registro universitario **14000996**, para su defensa publica, evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la carrera Ingeniería de sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente:



Ing. Walter Emilio Paco Siles
TUTOR ESPECIALISTA

El Alto, Noviembre de 2022

Señor(a)

M.Sc.Ing. Enrique Flores Baltazar
TUTOR METODOLOGICO TALLER DE GRADO II
CARRERA INGENIERIA DE SISTEMAS
UNIVERSIDAD PÚBLICA DE EL ALTO

Presente:

REF.: AVAL DE IMPLEMENTACIÓN

Distinguido ingeniero.

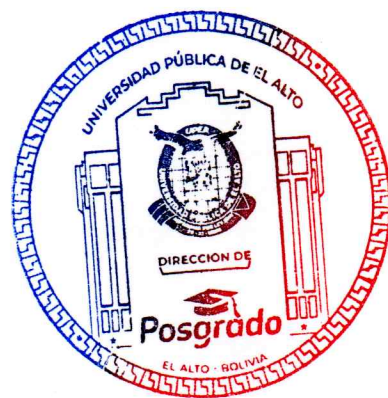
Mediante la presente tengo el bien comunicarle mi conformidad del proyecto de grado **“Sistema de Información web para la Administración y Publicación digital de Revistas Científicas”** caso: Dirección de Posgrado U.P.E.A. por parte del universitario **Gabriel Limachi Misme**, con cedula de identidad **9104987 L.P.** y registro universitario **14000996**, de haber realizado e implementado satisfactoriamente con éxito el sistema web mencionado.

Sin otro particular, reciba saludos cordiales.

Atentamente:



.....
M. Sc. Richard Jorge Torrez Juaniquina
DIRECTOR DE POSGRADO
UNIVERSIDAD PÚBLICA DE EL ALTO



MANUAL DE USUARIO

SISTEMA DE INFORMACIÓN WEB PARA LA ADMINISTRACIÓN Y PUBLICACIÓN DIGITALES DE REVISTAS CIENTÍFICAS

Por: Gabriel Limachi Misme



INICIO DE PÁGINA PRINCIPAL

The screenshot shows the main page of the Posgrado website. At the top left is the Posgrado logo. In the center, there are links for 'Inicio' and 'Revistas'. On the top right, there are links for 'Login' and 'Registrarme'. Below these is a large banner with the text 'Revistas Científicas' and 'Universidad Pública de El Alto'. A search bar is located below the banner with the placeholder text 'buscar titulo de revista'. Underneath the banner, there is a section titled 'Todas Las Revistas' which displays four journal covers. The covers are for 'REVISTA CIENTIFICA POSTDOCTORAL', 'GESTIÓN EDUCATIVA', 'EDUCACIÓN', and 'REVISTA CIENTIFICA DOCTORAL'. Two red circles with arrows point to the 'Login' and 'Registrarme' links, labeled '1' and '2' respectively. A blue arrow points to a '6' on the 'REVISTA CIENTIFICA DOCTORAL' cover. A blue arrow points to an upward arrow icon in the bottom right corner.

- 1: INICIAR SESIÓN: ingreso de usuario lector con su registro anterior.
- 2: REGISTRO DE USUARIO: registro de nuevos usuarios.

INICIAR SESIÓN

The screenshot shows the 'Iniciar sesión' login form. The form is enclosed in a blue border and contains the following elements: a title 'Iniciar sesión', a label 'Correo Electronico' above a text input field containing 'ejemplo@gmail.com', a label 'Contraseña' above a password input field, a blue button labeled 'Ingresar', a link '¿No tienes una cuenta? Registrarte aquí', a link '¿Olvidaste la contraseña? Haz clic aquí', and a link 'regresar a inicio'. Three red circles with arrows point to the email input field (labeled 'A'), the password input field (labeled 'B'), and the 'Ingresar' button (labeled 'C').

A: CORREO ELECTRONICO: ingreso el correo electrónico para ingresar en el sistema.

B: CONTRASEÑA: ingreso de contraseña del usuario.

C: BOTÓN DE INGRESAR AL SISTEMA: Dar clic al llenar el correo electrónico y contraseña.

REGISTRO DE USUARIO

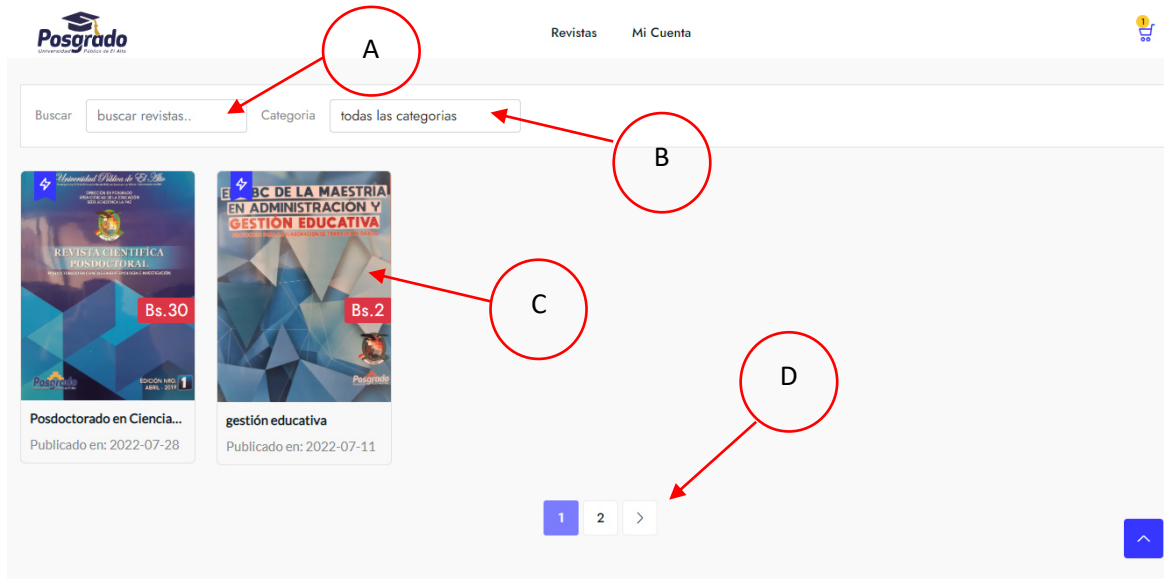
The image shows a registration form titled "Registro" with the following fields:

- Nombre
- Apellido paterno
- Apellido materno
- género (dropdown menu: "Seleccione un elemento")
- pais (dropdown menu: "Seleccione un elemento")
- celular (input field: "opcional")
- Numero de cedula de identidad
- expedido (dropdown menu: "Seleccione un elemento")
- Fecha de Nacimiento (calendar icon)
- Dirección de correo electrónico
- Nombre de Usuario
- Usuario Contraseña

At the bottom of the form is a blue button labeled "Crear Cuenta". A red circle containing the letter "A" has an arrow pointing to the right side of the form.

A: FORMULARIO DE REGISTRO: Llenar el formulario con todos los datos posteriormente dar clic crear cuenta.

REVISTAS DISPONIBLES



A: Entrada de texto para buscar

B: filtro de categorías disponibles.

C: clic para ver DETALLE DE LA REVISTA.

D: paginación de las revista.

DETALLE DE REVISTAS



A: agregar favoritos.

B: agregar al carrito.

C: información básica.

PERFIL DE USUARIO

Revistas Mi Cuenta

Posgrado

MI CUENTA gabriel

Mi Perfil

Mis favoritos

Mis Compras

Salir

Configuración de perfil

Primer Apellido* llimachi

Segundo Apellido* mismo

Nombres* gabriel

Nombre de Usuario* gabriel

Dirección de correo electrónico* gabriel_misme@hotmail.com

[Cambiar contraseña](#)

Actualizar Perfil

A

A: Perfil del usuario con sus datos personales, botón para actualizar y cambiar contraseña.

MIS FAVORITOS

Revistas Mi Cuenta

Posgrado

MI CUENTA gabriel

Mi Perfil

Mis favoritos

Mis Compras

Salir

Mis Favoritos

titulo	categoria
Posdoctorado en Ciencias en epistemología e investigación Bs 30	revista científica posdoctoral
revista científica doctoral Bs 25	educación

A

A: ítem favorito del usuario, clic para ver los detalles.

MIS COMPRAS

Posgrado
UNIVERSIDAD PÚBLICA DE EL ALTO

Revistas Mi Cuenta

MI CUENTA
gabriel

Mi Perfil
Mis favoritos
Mis Compras

Salir

Mis Compras

producto	fecha de compra	estado	acciones
revista cientifica doctoral Bs 25	2022-10-20	pagado	

A: Datos mínimo de la revista.

B: Estado de la compra.

C: Acciones de la revista, ver el archivo y la descarga del archivo.

CARRITO DE COMPRAS

Posgrado
UNIVERSIDAD PÚBLICA DE EL ALTO

Revistas Mi Cuenta

Carrito De Compras

REVISTA CIENTIFICA DOCTORAL
DOCTORADO EN CIENCIAS DE LA EDUCACION E INVESTIGACION

revista cientifica doctoral
Bs. 25

total de la compra: **Bs.25**

continuar a la caja

las revistas no son fisicas son una copia digital!

A: Eliminar de la lista del carrito.

B: Botón para continuar a la caja.

MODO DE PAGO

Modos de pagos

1. Realizar el depósito en las cuentas disponibles:
2. El llenado de sus datos realicelo con cuidado:

banco: banco union
numero de cuenta: 5235235235532
nombre: erik flores mamani
ci: 235223532 lp

banco: banco mercantil santa cruz
numero de cuenta: 31356848
nombre: luis flores
ci: 52524624 lp

Total de la compra a pagar en bolivianos

DATOS PERSONALES

DATOS PERSONALES

Nombre	Apellido paterno
<input type="text" value="gabriel"/>	<input type="text" value="liimachi"/>
Apellido materno	género
<input type="text" value="misme"/>	<input type="text" value="Hombre"/>
pais	celular
<input type="text" value="Bolivia"/>	<input type="text" value="75232024"/>
Dirección de correo electrónico	
<input type="text" value="gabriel_misme@hotmail.com"/>	

INFORMACIÓN DE PAGO

INFORMACIÓN DE PAGO

seleccione el banco al que realizo el pago

Numero de deposito

fecha de pago

monto de pago en bolivianos

total de la compra en bolivianos

una fotografia o captura del deposito

[Anterior](#) [Finalizar](#)

FINAL DEL PROCESO DE COMPRA

FINAL DEL PROCESO

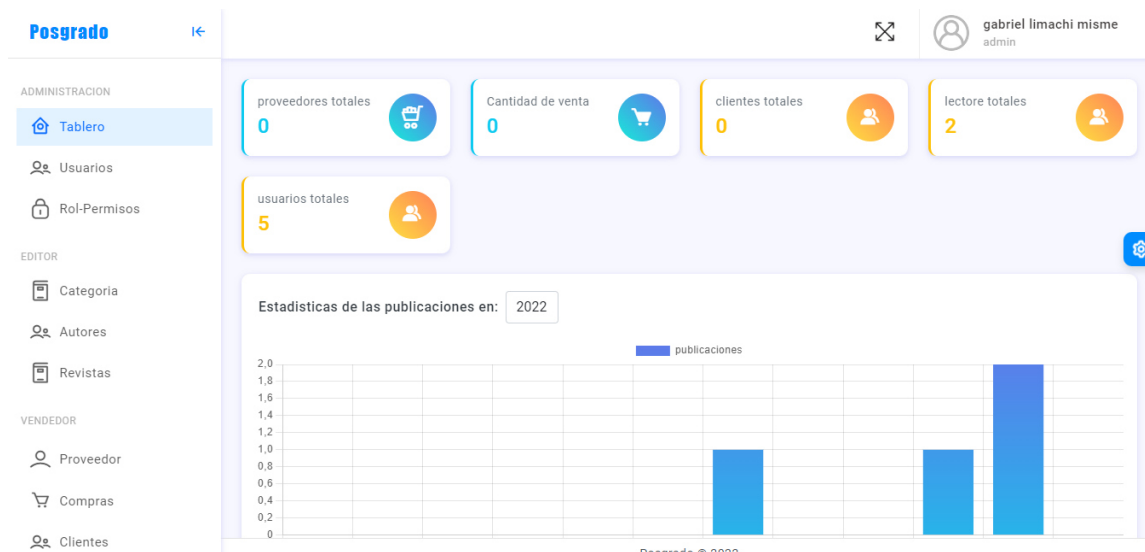
1.se procesara su compra una vez verificado los datos enviados esto puede durar unas horas.
2.podra ver sus compras en la seccion de su perfil de cuenta, las copias de los archivos digitales estaran disponibles para su descarga.

[Ver Mas Revistas](#)

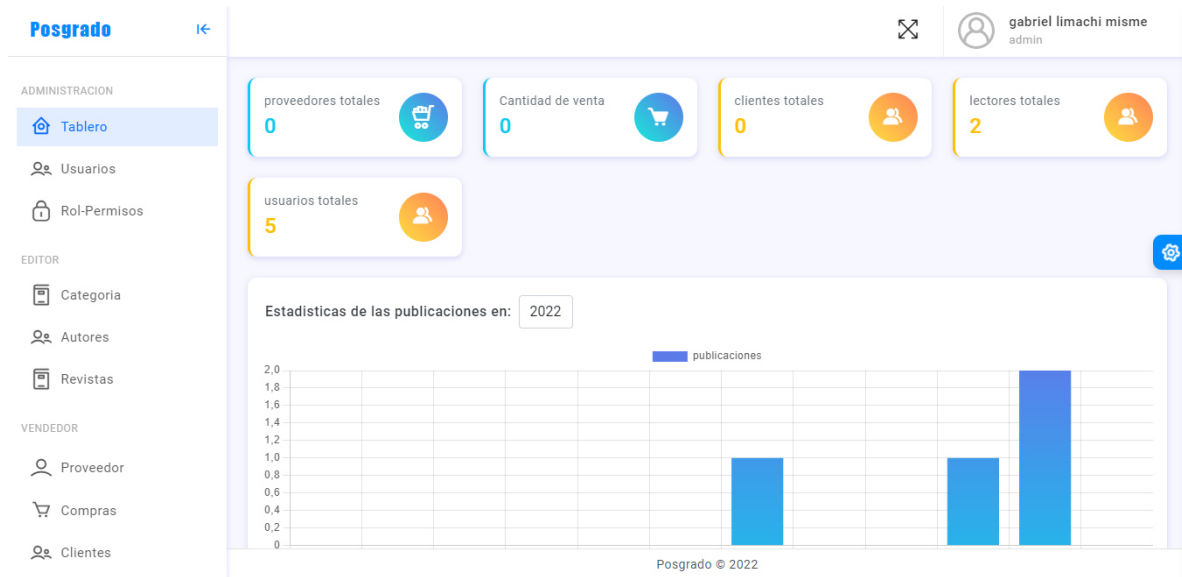
MANUAL DE ADMINISTRACIÓN

SISTEMA DE INFORMACIÓN WEB PARA LA ADMINISTRACIÓN Y
PUBLICACIÓN DIGITALES DE REVISTAS CIENTÍFICAS

Por: Gabriel Limachi Misme



PAGINA DE INICIO



USUARIOS

The 'USUARIOS' page includes a search bar and a table of users. Annotations A through E point to specific elements:

- A: Botón 'reporte en pdf'.
- B: Botón 'Nuevo'.
- C: Botón de configuración.
- D: Botón de paginación '10'.
- E: Botón de acción (checkmark) en la fila del usuario 'admin'.

#	nombre usuario	estado	rol	creado en	actualizado en	Acciones
1	vendedor	ACTIVO	vendedor	2022-06-17 13:30:01	2022-06-29 02:58:33	[check] [trash] [confirm]
2	gabriel	ACTIVO	lector	2022-06-19 23:38:17	2022-11-20 01:03:23	[check] [trash] [confirm]
3	editor	ACTIVO	editor	2022-07-08 02:18:58	2022-07-08 02:18:58	[check] [trash] [confirm]
4	gabo	ACTIVO	lector	2022-10-31 21:26:32	2022-11-17 03:38:29	[check] [trash] [confirm]
5	admin	ACTIVO	admin		2022-11-17 23:43:27	[check] [trash] [confirm]

A: Botón reporte en pdf descarga.

B: Botón crear nuevo usuario

Crear Nuevo Usuario

Información Personal

Nombre	Apellido Paterno	Apellido Materno
<input type="text"/>	<input type="text"/>	<input type="text"/>
<small>campo requerido</small>	<small>campo requerido</small>	<small>campo requerido</small>
Correo Electronico	Celular	Genero
<input type="text"/>	<input type="text"/>	Seleccione un elemei <input type="text"/>
<small>campo requerido</small>	<small>campo requerido</small>	<small>campo requerido</small>
Numero C.I.	Exp	Fecha Nacimiento
<input type="text"/>	Seleccione un elemei <input type="text"/>	dd/mm/aaaa <input type="text"/>
<small>campo requerido</small>	<small>campo requerido</small>	<small>campo requerido</small>
Direccion	Pais	
<input type="text"/>	<input type="text"/>	

Cancelar Guardar

C: Botón cambiar estado.

Cambiar Estado

Desea Cambiar el Estado del usuario?

Cancelar cambiar estado

D: Botón editar los datos del usuario.

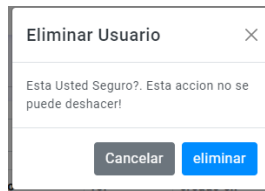
editar Usuario

Información Personal

Nombre	Apellido Paterno	Apellido Materno
vendedor ✓	vendedor ✓	vendedor ✓
Correo Electronico	Celular	Genero
vendedor@gmail.com ✓	2313123 ✓	Hombre ✓
Numero C.I.	Exp	Fecha Nacimiento
1412412 ✓	Potosi - PT ✓	24/06/2022 ✓
Direccion	Pais	
El Alto ✓	Bolivia ✓	

Cancelar Actualizar

E: Botón eliminar usuario.



ROL Y PERMISOS

Posgrado

ADMINISTRACION

- Tablero
- Usuarios
- Rol-Permisos**

EDITOR

- Categoría
- Autores
- Revistas

VENDEDOR

- Proveedor
- Compras
- Clientes

SEGUIDAD

gabriel limachi mismo admin

Buscar por nombre

#	rol	creado en	actualizado en	Acciones
1	vendedor	2022-10-13 18:26:45	2022-07-08 13:45:32	[editar] [reporte en pdf]
2	editor	2022-10-13 18:26:45	2022-10-23 21:00:26	[editar]
3	admin	2022-10-13 18:26:45	2022-07-19 02:40:16	[editar]
4	lector	2022-10-13 18:26:45	2022-06-29 02:05:43	[editar]

10

Posgrado © 2022

A: Botón reporte en pdf descarga.

B: Botón editar permiso del usuario

Permisos

Activar todo:

#	activar	menu	submenu
1	<input checked="" type="checkbox"/>	categorias	crear
2	<input checked="" type="checkbox"/>	categorias	editar
3	<input type="checkbox"/>	categorias	eliminar
4	<input checked="" type="checkbox"/>	categorias	listar
5	<input checked="" type="checkbox"/>	categorias	cambiar-estado
6	<input type="checkbox"/>	categorias	reportes
7	<input checked="" type="checkbox"/>	autores	crear
8	<input checked="" type="checkbox"/>	autores	editar
9	<input type="checkbox"/>	autores	eliminar
10	<input checked="" type="checkbox"/>	autores	listar

Cancelar Actualizar

CATEGORIA

Posgrado

ADMINISTRACION

- Tablero
- Usuarios
- Rol-Permisos

EDITOR

- Categoría**
- Autores
- Revistas

VENDEDOR

- Proveedor
- Compras
- Clientes

CATEGORIA

gabriel limachi mismo admin

Buscar por nombre

#	nombre categoria	estado	creado en	actualizado en	Acciones
1	desarrollo en tecnología ia	INACTIVO	2022-11-19 20:32:22	2022-11-20 00:57:26	[edit] [delete] [check] [settings]
2	educación	ACTIVO	2022-06-19 21:37:07	2022-06-19 21:37:14	[edit] [delete] [check]
3	revistas gratis	ACTIVO	2022-07-02 23:54:05	2022-07-02 23:55:14	[edit] [delete] [check]
4	revista científica posdoctoral	ACTIVO	2022-07-28 15:04:53	2022-07-28 20:49:02	[edit] [delete] [check]

10

Posgrado © 2022

A: Descargar reporte en pdf

B: Crear una nueva categoría

Crear Nueva categoría

Nombre de categoría

campo requerido

Cancelar Guardar

C: Cambiar estado de la categoría

Cambiar Estado

Desea Cambiar el Estado?

Cancelar cambiar estado

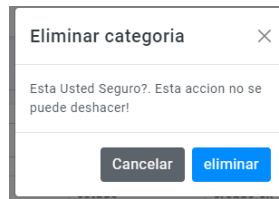
D: Editar categoría

editar categoría

Nombre de categoría

Cancelar Actualizar

E: Eliminar categoría



AUTORES

The screenshot shows the 'AUTORES' management interface. The left sidebar contains navigation options: ADMINISTRACION (Tablero, Usuarios, Rol-Permisos), EDITOR (Categoria, Autores, Revistas), and VENDEDOR (Proveedoror, Compras, Clientes). The main area displays a table of authors with columns: #, formacion, nombre, apellido paterno, apellido materno, genero, estado, and Acciones. The 'Acciones' column contains icons for edit, delete, and check. Annotations A through E point to specific elements: A points to the 'reporte en pdf' button, B points to the 'Nuevo' button, C points to the 'Eliminar' icon in the actions column, D points to the 'Editar' icon, and E points to the 'Eliminar' icon. A search bar and a dropdown menu are also visible.

A: Descargar reporte en pdf

B: Crear un nuevo autor

Crear Nuevo Autor

Formacion: Seleccione un element (required)

Nombre: (required)

Apellido Paterno: (required)

Apellido Materno: (required)

Genero: Seleccione un element (required)

Cancelar Guardar

C: Cambiar estado del autor

Cambiar Estado ✕

Desea Cambiar el Estado?

Cancelar cambiar estado

D: Editar datos del autor

editar Autor ✕

Formacion: Licenciada.(a) ✓

Nombre: Betty Nely ✓

Apellido Paterno: Arcani ✓

Apellido Materno: Arcani ✓

Genero: Mujer ✓

Cancelar Actualizar

E: Eliminar el autor

Eliminar Autor ✕

Esta Usted Seguro?. Esta accion no se puede deshacer!

Cancelar eliminar

REVISTAS

The screenshot shows the 'REVISTAS' management interface. On the left is a sidebar with navigation options: Posgrado, Usuarios, Rol-Permisos, EDITOR (Categoria, Autores, Revistas), and VENDEDOR (Proveedor, Compras, Clientes, Ventas, Ventas en linea). The main area displays a table of publications with columns: revista estado, fecha de publicacion, publicacion estado, creado en, actualizado en, detalle de publicacion, and Acciones. Annotations A through G point to specific elements: A points to the 'reporte en pdf' button, B to the 'Nuevo' button, C to the search bar, D to the 'publicacion estado' column, E to the 'detalle de publicacion' column, F to the 'Acciones' column, and G to the trash icon in the actions column.

revista estado	fecha de publicacion	publicacion estado	creado en	actualizado en	detalle de publicacion	Acciones
ACTIVO	2022-11-19	PUBLICADO	2022-07-03 21:18:29	2022-11-19 20:35:59	ver detalle	[icon] [icon] [icon]
ACTIVO	2022-10-23	PUBLICADO	2022-06-26 00:14:05	2022-10-23 21:44:49	ver detalle	[icon] [icon] [icon]
ACTIVO	2022-07-28	PUBLICADO	2022-07-28 20:35:25	2022-07-28 22:42:15	ver detalle	[icon] [icon] [icon]
ACTIVO	2022-11-06	PUBLICADO	2022-07-03 00:14:11	2022-11-06 02:12:55	ver detalle	[icon] [icon] [icon]

F: Editar la revista

editar revista

Información de la revista

titulo
revista científica doctoral ✓

resumen
El rendimiento académico se constituye en un pilar fundamental para el logro de la tan

cantidad de paginas: 23 ✓ numero de Volumen: 6 ✓ año de edición: 2019 ✓

idioma: español ✓ categoria: educación ✓

autores
Lic Betty Nely Arcani Arcani

Cancelar Actualizar

G: Eliminar la revista

Eliminar revista

Esta Usted Seguro?. Esta accion no se puede deshacer!

Cancelar Eliminar

PROVEEDOR

The screenshot shows the 'PROVEEDOR' management interface. On the left is a sidebar with navigation options: Posgrado, Usuarios, Rol-Permisos, EDITOR (Categoria, Autores, Revistas), and VENDEDOR (Proveedor, Compras). The main area displays a table of providers with columns: nit/ci, celular, direccion, email, estado, creado en, actualizado en, and Acciones. A search bar is at the top. Annotations A through E point to specific elements: A points to the 'Reporte en pdf' button; B points to the 'Nuevo' button; C points to the 'Reporte en pdf' button in the actions column; D points to the trash icon in the actions column; E points to the settings gear icon in the actions column.

nit/ci	celular	direccion	email	estado	creado en	actualizado en	Acciones
412412545754	325467854	El Alto	imprentasol@gmail.com	ACTIVO	2022-11-24 12:34:43	2022-11-24 12:34:43	[Reporte en pdf] [Nuevo] [Reporte en pdf] [Eliminar] [Check] [Settings]

A: Descargar reporte en pdf

B: Crear un nuevo proveedor

Nuevo Proveedor

Información empresa

Nombre empresa Nit/ci

telefono/celular direccion

Correo Electronico

Cancelar Guardar

C: Editar proveedor

editar Proveedor

Información empresa

Nombre empresa Nit/ci

telefono/celular direccion

Correo Electronico

Información del representante legal

Cancelar Actualizar

D: Eliminar proveedor

Eliminar Proveedor

Esta Usted Seguro?. Esta accion no se puede deshacer!

Cancelar eliminar

E: Cambiar estado del proveedor

Cambiar Estado

Desea Cambiar el Estado?

Cancelar cambiar estado

COMPRAS

Posgrado ←

gabriel limachi mismo
admin

COMPRAS

Buscar por nombre

#	proveedor	fecha	total	tipo de documento	Acciones
1	Imprenta sol	2022-11-24	791	factura	detalles

10

1

VENDEDOR

- Proveedor
- Compras**
- Cientes
- Ventas
- Ventas en linea
- Almacen

A: Nueva compra

compra de revistas

INGRESO DE DATOS DE LA COMPRA

N° Factura: Ingrese el número del documento
fecha del documento: dd/mm/aaaa
tipo de documento: Factura
empresa: Proveedor

REGISTRAR DETALLE DE LA COMPRA

agregar un producto o varios

Buscar productos...

PRODUCTO	CANTIDAD	PRECIO UNITARIO - COMPRA	PRECIO UNITARIO - VENTA	IMPORTE	
Sub total	0.00	IVA	0.00	Total	0.00

Registrar

D: Detalle de compra

Detalle de compra

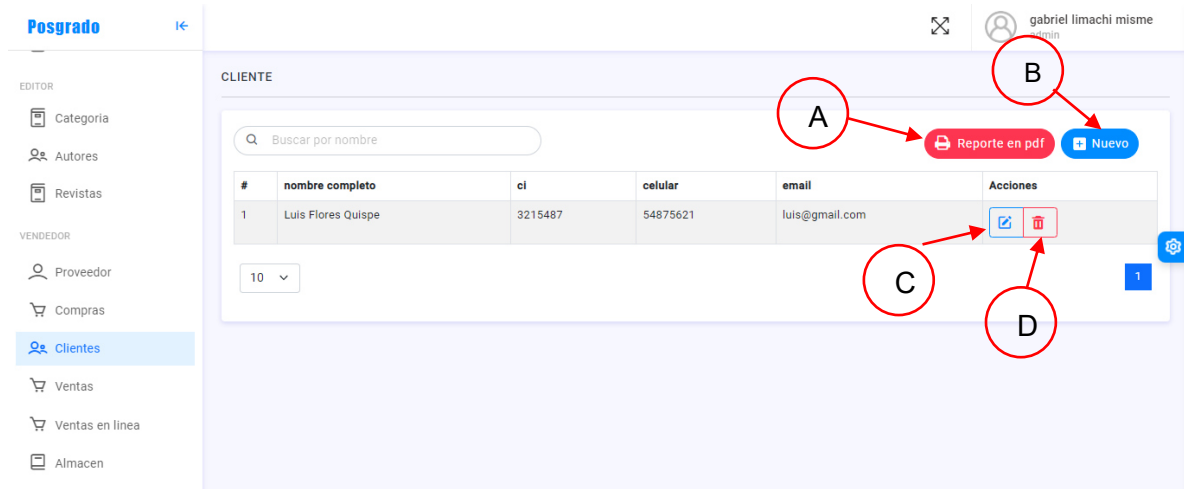
reporte en pdf

Proveedor: imprenta sol
NIT/CI: 412412545754
Email: imprentasol@gmail.com
Teléfono/Celular: 325467854

Tipo de documento: factura
Fecha: 2022-11-24

Cantidad	Producto	Precio unitario	Importe
50	gestión educativa	2 bs	100.00 bs
20	Posdoctorado en Ciencias en epistemología e investigación	30 bs	600.00 bs
Subtotal			700
Iva			91
Total		791	bs

CLIENTES



A: Descargar reporte en pdf

B: Crear nuevo cliente

Nuevo cliente

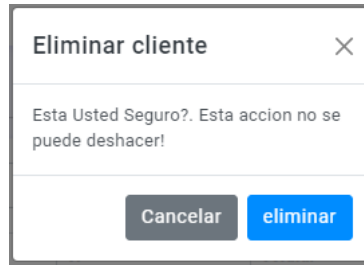
Nombre	Apellido Paterno	Apellido Materno
<input type="text" value="Luis"/>	<input type="text" value="Flores"/>	<input type="text" value="Quispe"/>
Numero C.I.	Exp	Pais
<input type="text" value="3215487"/>	<input type="text" value="La Paz - LP"/>	<input type="text" value="Bolivia"/>
Genero	Celular	Direccion
<input type="text" value="Hombre"/>	<input type="text" value="54875621"/>	<input type="text" value="El Alto"/>
Correo Electronico		
<input type="text" value="luis@gmail.com"/>		

C: Editar datos del cliente

editar cliente

Nombre	Apellido Paterno	Apellido Materno
<input type="text" value="Luis"/>	<input type="text" value="Flores"/>	<input type="text" value="Quispe"/>
Numero C.I.	Exp	Pais
<input type="text" value="3215487"/>	<input type="text" value="La Paz - LP"/>	<input type="text" value="Bolivia"/>
Genero	Celular	Direccion
<input type="text" value="Hombre"/>	<input type="text" value="54875621"/>	<input type="text" value="El Alto"/>
Correo Electronico		
<input type="text" value="luis@gmail.com"/>		

D: Eliminar datos del cliente



VENTAS DE REVISTAS

The dashboard shows a sidebar with navigation options: Posgrado, EDITOR (Categoria, Autores, Revistas), and VENDEDOR (Proveedor, Compras, Clientes, Ventas, Ventas en linea, Almacen). The main area is titled 'VENTAS' and contains a search bar, a 'Nuevo' button, and a table of sales records. A red circle 'A' highlights the 'Nuevo' button, and a red circle 'B' highlights the 'detalles' link in the 'Acciones' column of the first row.

#	cliente	fecha	sub total	total	comprobante	Acciones
1	Luis Flores Quispe	2022-11-24	55.00	62.15	factura	detalles

A: Nueva venta

venta de revistas

Buscar productos... Luis Flores Quispe Factura

PRODUCTO	CANTIDAD	PRECIO UNITARIO	STOCK	IMPORTE
Posdoctorado en Ciencias en epistemologia e investigación	1	30	97	30.00
revista cientifica doctoral	1	25	164	25.00

Sub total: 55.00 IVA: 7.15 Total: 62.15 descuento: 0.00 Registrar

B: Detalle de la venta

Detalle de venta

reporte en pdf

Cliente: Luis Flores Quispe
C#:3215487
Email: luis@gmail.com
Teléfono/Celular:54875621

Tipo de documento:factura
Fecha: 2022-11-24

Cantidad	Producto	Precio	Importe
1	Posdoctorado en Ciencias en epistemologia e investigación	30	30.00
1	revista cientifica doctoral	25	25.00

Subtotal: 55.00
Iva: 7.15
descuento:
Total: 62.15

VENTAS EN LINEA

Posgrado

VENTAS EN LINEA

Buscar numero de deposito

#	fecha de la venta	total	fecha de pago	numero de deposito	imagen del pago	banco	estado del pago	Acciones
1	2022-10-20	25	2022-10-20	54356653454		banco union	PAGADO	

10

1

A: Ver detalle de la venta en línea

Detalle de venta en línea

enviar comprobante al email | reporte en pdf

Posgrado

Cliente: gabriel limachi mismo Fecha: 2022-10-20
CI: 12345678
Email: gabriel.mismo@hotmail.com
Teléfono/Celular: 75232024

Cantidad	Producto	Precio	Importe
1	revista cientifica doctoral	25	25

descuento 0

Total 25

B: Información del pago

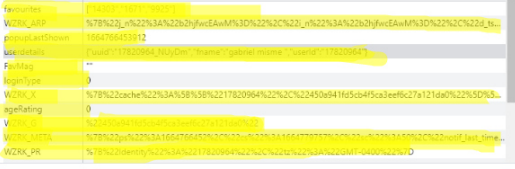
información del pago

numero de deposito: 54356653454 banco: banco union

fecha de pago: 2022-10-20

estado del pago: pagado

imagen del pago:



actualizar estado

ALMACEN

Posgrado

gabriel limachi mismo admin

ALMACEN

Buscar por titulo

#	titulo	Existencia	Precio venta	Precio compra
1	educación	88	0	15
2	gestión educativa	57	2	2
3	revista científica doctoral	164	25	25
4	Posdoctorado en Ciencias en epl...	97	30	30

10

1