

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

SISTEMA WEB PARA LA GESTIÓN DE INVENTARIOS Y PRESTAMOS EN LOS LABORATORIOS Y TALLERES DE LA CARRERA DE ELECTRICIDAD INDUSTRIAL

CASO: INSTITUTO TECNOLÓGICO MARCELO QUIROGA SANTA CRUZ

Para optar al título de Licenciatura en Ingeniería de Sistemas

Mención: INFORMÁTICA Y COMUNICACIONES

POSTULANTE: Jesús Manuel Tambo Limachi

TUTOR METODOLÓGICO: Ing. Marisol Arguedas Balladares

TUTOR ESPECIALISTA: Ing. Jaime Nestor Ticona Huariste

TUTOR REVISOR: Ing. Fanny Helen Perez Mamani

EL ALTO – BOLIVIA

2020

Dedicatoria:

Este trabajo de grado lo dedico principalmente a Dios, por darme la vida y permitirme llegar hasta este momento tan importante de mi formación profesional.

A mis queridos padres Natalia y Valentín, por haberme apoyado siempre.

A Fabio y Ubaldo mis hermanos y amigos quienes iluminan mi camino.

A mi familia quienes fueron la motivación para seguir adelante.

Agradecimientos:

Al Ing. Jaime Nestor Ticona Huariste por su colaboración desinteresada en llevar adelante experiencias de aula.

A la Universidad Pública de El Alto, por habernos acogido en sus aulas y permitirme dar un paso más en mi formación profesional.

Al Instituto “Tecnológico Marcelo Quiroga Santa Cruz”, por brindarme facilidades para elaborar este trabajo de investigación.

RESUMEN EJECUTIVO

En la actualidad, el Instituto Tecnológico Marcelo Quiroga Santa Cruz, no cuenta en sus laboratorios con sistema automatizado para el control de su inventario de equipos, instrumentos y herramientas en la carrera de electricidad industrial, además que el seguimiento de préstamos realizados a estudiantes y docentes no eficiente ocasionando confusión y un flujo de información no confiable.

La metodología web empleada en el presente proyecto es OOHDM cuyas fases o etapas son 4: Análisis de requerimientos, diseño conceptual, diseño navegacional y diseño de interfaces abstractas e implementación. La metodología de desarrollo de software empleada es SCRUM una metodología ágil de la cual fue dividida después de la recopilación de información y entrevistas fue dividida en 4 sprints o iteraciones, cumpliendo las 3 fases requeridas por dicha metodología (pre juego, juego y pos juego).

Para la evaluación de calidad del software se realizó mediante la norma ISO 9126 que se basa en los parámetros de usabilidad, funcionalidad, confiabilidad, mantenibilidad, portabilidad, para las pruebas del sistema se recurrió a la prueba de caja negra y la seguridad del sistema se utilizó algoritmos como MD5 para la encriptación de datos y finalmente para la estimación del costo del producto se usó COCOMO II con el modelo de post arquitectura. En cuanto al inventario se empleó el modelo de valuación PEPS mediante las alertas al usuario.

Para el desarrollo de software las herramientas que se utilizaron fueron: el lenguaje de programación PHP, servidor de base de datos MariaDB, Framework Laravel, y en la parte de diseño HTML, JavaScript, Bootstrap, Ajax y CSS.

INDICE

	Pág.
CAPÍTULO I	
MARCO PRELIMINAR.....	1
1.1 INTRODUCCIÓN.....	2
1.2. ANTECEDENTES.....	3
1.2.1 Antecedentes Institucionales.....	3
1.2.2. Antecedentes de trabajos afines.....	7
1.3. PLANTEAMIENTO DEL PROBLEMA.....	9
1.3.1. Descripción de la situación problemática.....	9
1.3.2. Problema Principal.....	10
1.3.3. Problemas Secundarios.....	10
1.4. PLANTEAMIENTO DE OBJETIVOS.....	11
1.4.1. Objetivo Principal.....	11
1.4.2. Objetivos Secundarios.....	11
1.5. JUSTIFICACIÓN.....	11
1.5.1. Justificación Técnica.....	11
1.5.2. Justificación Económica.....	12
1.5.3. Justificación Social.....	12
1.6. METODOLOGÍA.....	12
1.6.1. Método de Ingeniería.....	13
1.7. HERRAMIENTAS.....	14
1.8. LÍMITES Y ALCANCES.....	15
1.8.1. Límites.....	15
1.8.2. Alcances.....	16
1.9. APORTES.....	16
1.9.1. Practico.....	16
1.9.2. Teórico.....	16
CAPÍTULO II	
MARCO TEÓRICO.....	18

INDICE

	Pág.
2.1. INTRODUCCIÓN.	19
2.2. DEFINICIÓN DE SISTEMAS DE INFORMACIÓN.	19
2.3. DEFINICIÓN DE INVENTARIO.....	20
2.3.1. Tipos de inventario	21
2.3.2. Control de inventario de equipos.	22
2.3.3. Método de valuación de inventarios.	23
2.3.3.1. PEPS.....	24
2.4. INGENIERÍA WEB.	25
2.4.1. Proceso de la Ingeniería Web.....	26
2.5. INGENIERÍA DE REQUERIMIENTOS.	27
2.5.1. Definición de Ingeniería de Requerimientos.	27
2.6. OBJECT ORIENTED HIPERMEDIA DESING METHODOLOGY (OOHDM).....	28
2.6.1. Fases de OOHDM.	29
2.6.1.1. Obtención de Requerimientos.	30
2.6.1.2. Diseño Conceptual.	32
2.6.1.3. Diseño Navegacional.....	32
2.6.1.4. Diseño de Interfaces Abstractas.....	32
2.6.1.5. Implementación.	32
2.7. METODOLOGÍA AGIL SCRUM.	35
2.7.1. Componentes de Scrum.	36
2.7.1.1. Reuniones.	36
2.7.1.2. Roles o responsabilidades.....	36
2.7.2. Elementos de Scrum.....	38
2.7.3. Fases de Scrum.	39

INDICE

	Pág.
2.7.3.1 PreGame.	40
2.7.3.2 Game.....	41
2.7.3.3 Post Game.....	42
2.8. HERRAMIENTAS.....	42
2.8.1. PHP 7.....	42
2.8.2. Framework Laravel	42
2.8.3. CSS.	44
2.8.4. MySQL.....	44
2.9. CALIDAD DE SOFTWARE ISO 9126.	44
2.10. MODELO DE ESTIMACIÓN DE COSTOS COCOMO II.	48
2.10.1. Estimación de esfuerzo.	50
2.10.1.1. Modelo de composición de aplicación.	50
2.10.1.2. Modelo de diseño anticipado.	51
2.10.1.3. Modelo de post arquitectura.	52
2.11. SEGURIDAD DEL SOFTWARE.....	55
2.11.1. Algoritmo MD5.....	55
2.12. PRUEBAS DEL SOFTWARE.....	56
2.12.1. Pruebas de caja negra.....	57
2.12.1.1. Análisis del valor limite.....	57
CAPÍTULO III	
MARCO APLICATIVO.....	60
3.1. INTRODUCCIÓN.	61
3.2. ANALISIS DE LA SITUACION ACTUAL.	61
3.3. PRE GAME (antes del juego).....	61
3.3.1. Planeación.....	61

INDICE

	Pág.
3.3.2. Análisis de Requerimientos (Product Backlog).....	62
3.3.3. Identificación de Usuarios.....	63
3.3.4. Definición de Roles de los Actores	63
3.3.5. Requerimientos de Usuario.	64
3.3.6. Planificación de la Entrega.	68
3.2.6.1. Análisis de Riesgo.	68
3.4. GAME (Juego)	69
3.4.1. Iteraciones (Sprints).....	69
3.4.1.1. Primera Iteración – Módulo Usuarios.....	70
3.4.1.1.1. Diseño de la Base de Datos.	70
3.4.1.1.2. Diseño Conceptual – Módulo Usuarios.....	71
3.4.1.1.3. Diseño Navegacional – Módulo Usuarios.....	72
3.4.1.1.4. Diseño Interfaz Abstracta – Módulo Usuarios.....	73
3.4.1.1.5. Implementación - Módulo De Usuarios.....	73
3.4.1.2. Segunda iteración - Módulo de Inventario.	77
3.4.1.2.1. Diseño conceptual – Módulo de inventario.	79
3.4.1.2.2. Diseño Navegacional – Módulo de inventario.....	80
3.4.1.2.3. Diseño de Interfaces Abstractas – modulo inventario.....	81
3.4.1.2.4. Implementación – Módulo inventario.	81
3.4.1.3. Tercera Iteración – Módulo de prestamos	82
3.4.1.3.1. Diseño Conceptual – Módulo de préstamos.	83
3.4.1.3.2. Diseño Navegacional - Módulo de préstamos.	84
3.4.1.3.3. Diseño de Interfaces abstractas – Módulo de préstamos.	85
3.4.1.3.4. Diseño de Implementación – Módulo de préstamos.....	86

INDICE

Pág.

3.4.1.4. Cuarta iteración - Módulo de reportes y búsquedas.....	89
3.4.1.4.1. Diseño Conceptual – Módulo de reportes.....	89
3.4.1.4.2. Diseño Navegacional - Módulo de reportes.....	90
3.4.1.4.3. Diseño de Interfaces abstractas – Módulo de reportes.....	91
3.4.1.4.4. Diseño de Implementación – Módulo de reportes.....	91
3.5. POST GAME (después del juego)	94
3.5.1. PRUEBAS DEL SISTEMA.	94
3.5.2. Prueba de Caja Negra.....	95

CAPÍTULO IV

CALIDAD DE SOFTWARE Y SEGURIDAD.....	97
4.1. INTRODUCCIÓN.....	98
4.2. Calidad ISO/ IEC 9126.	98
4.2.1. Funcionalidad.	98
4.2.2. Confiabilidad.....	103
4.2.3. Usabilidad.....	105
4.2.4. Capacidad de Mantenimiento.....	107
4.2.5. Portabilidad.....	109

CAPÍTULO V

ANÁLISIS DE COSTOS.....	111
5.1. INTRODUCCIÓN.	117
5.2. ANÁLISIS DE COSTOS.	117
5.3. COCOMO II.....	117
5.3.1. Líneas de Código.....	117

INDICE

	Pág.
5.3.2. Factor de Escala.....	117
5.3.3. Esfuerzo Nominal (PM).....	117
5.3.4. Multiplicadores de Esfuerzo.....	117
 CAPÍTULO VI	
CONCLUSIONES Y RECOMENDACIONES.....	116
6.1. CONCLUSIONES.....	117
6.1.1. Respecto al Proyecto.....	117
6.1.2. Respecto a los Objetivos Específicos.....	117
6.1.3. Respecto al Objetivo General.....	118
6.2. RECOMENDACIONES.....	118
6.2.1. Respecto al software.....	118
 BIBLIOGRAFÍA.....	 119
WEBGRAFIA.....	121
 ANEXOS	

INDICE DE TABLAS

Pág.

Tabla 1.1. Carreras del Tecnológico Marcelo Quiroga Santa Cruz	5
Tabla 2.1. Fase de implementación de OOHDM.....	34
Tabla 2.2. Principales características y ventajas de Laravel	43
Tabla 2.3. Características ISO 9126.	45
Tabla 2.4. Aplicaciones en COCOMO II.....	45
Tabla 2.5. Productividad para el modelo Composición de Aplicación.	51
Tabla 2.6. Condiciones para la prueba de Caja Negra.	58
Tabla 3.1. Historia de usuario - 1.	64
Tabla 3.2. Historia de usuario - 2.	65
Tabla 3.3. Historia de usuario - 3.	65
Tabla 3.4. Historia de usuario - 4.	66
Tabla 3.5. Historia de usuario - 5.	66
Tabla 3.6. Requerimientos de usuario.	67
Tabla 3.7. Cronograma de entrega	68
Tabla 3.8. Análisis de riesgo	69
Tabla 3.9. Primera iteración.	70
Tabla 3.10. Segunda iteración.	78
Tabla 3.11. Tercera iteración.	83
Tabla 3.12. Cuarta iteración.....	89
Tabla 3.13. Prueba equivalencia - Inicio de sesión.....	95
Tabla 3.14. Prueba caja negra – Control de inventario.....	96
Tabla 4.1. Calculo del punto función.....	101
Tabla 4.2. Valores de ajuste.....	102
Tabla 4.3. Escala de valores.....	106
Tabla 4.4. Preguntas de evaluación.....	106
Tabla 5.1. Factores de Escala.....	109
Tabla 5.2. Multiplicadores de esfuerzo.....	114
Tabla 5.3. Sueldo de programadores.....	115

INDICE DE FIGURAS

	Pág.
Figura 1.1. Estructura Orgánica del Tecnológico Marcelo Quiroga Santa Cruz.....	7
Figura 2.1. Proceso del control de inventarios	23
Figura 2.2. Relación entre objetos conceptuales, de navegación y de interfaz en OOHDM..	28
Figura 2.3. Etapas de OOHDM, para el desarrollo de software hipermedia	29
Figura 2.4. El proceso de obtención de requerimientos.	31
Figura 2.5. Flujo del proceso SCRUM.....	35
Figura 2.6. Ciclo de desarrollo Scrum	39
Figura 2.7. El proceso o las tres fases de Scrum: Pregame, Development y Postgame	40
Figura 3.1. Planeación del desarrollo del proyecto.	62
Figura 3.2. Diseño de la base de datos del sistema.....	71
Figura 3.3. Diseño Conceptual - Módulo de Usuario.....	72
Figura 3.4. Diseño Navegacional - Módulo de Usuarios..	72
Figura 3.5. Diseño Abstracto - Módulo Usuarios.....	73
Figura 3.6. Diseño Abstracto - Registro de Usuario Persona.....	73
Figura 3.7. Interfaz - Inicio de Sesión.....	74
Figura 3.8. Interfaz - Vista Home	74
Figura 3.9. Interfaz - Registro Usuario y Asignación de rol..	75
Figura 3.10. Interfaz - Listado Usuario	75
Figura 3.11. Interfaz - Roles listado de docentes.....	76
Figura 3.12. Interfaz - Registro de persona.....	76
Figura 3.13. Interfaz - Listado estudiantes	77
Figura 3.14. Diseño conceptual - Modulo Inventario.....	79
Figura 3.15. Diseño Navegacional - Módulo de Inventario.....	80
Figura 3.16. Diseño Abstracta - Vista Inventario.....	80
Figura 3.17. Diseño abstracto - Registro de Inventario.....	81

INDICE DE FIGURAS

	Pág.
Figura 3.18. Interfaz - Registro de Equipo.	81
Figura 3.19. Interfaz - Listado de Equipos.	82
Figura 3.20. Interfaz - Baja de Equipo.....	82
Figura 3.21. Diseño Conceptual - Módulo de Préstamos.....	84
Figura 3.22. Diseño Navegacional - Módulo de Préstamos..	85
Figura 3.23. Diseño abstracto - módulo de préstamos.....	85
Figura 3.24. Diseño abstracto - Listado de solicitudes pendientes..	86
Figura 3.25. Interfaz – lista y registro de solicitud de préstamo de equipo.....	86
Figura 3.26. Interfaz - Listado de solicitudes pendientes.	87
Figura 3.27. Interfaz –Registro de aprobación de préstamo..	87
Figura 3.28. Interfaz –Registro de devolución de equipo.....	88
Figura 3.29. Diseño conceptual - modulo reportes.....	90
Figura 3.30. Diseño navegacional - Modulo de reportes.....	90
Figura 3.31. Diseño abstracto - modulo reportes..	91
Figura 3.32. Interfaz – Listado de reportes de equipos, instrumentos y herramientas según la fecha y ubicación	91
Figura 3.33. Interfaz - Reporte de baja de equipos.....	92
Figura 3.34. Interfaz – Reporte trimestral de equipos.	92
Figura 3.35. Interfaz - Reporte semestral de equipos.	93
Figura 3.36. Interfaz - Reporte anual de equipos.....	93
Figura 3.37. Grafica de Equipos Prestados	94
Figura 3.38. Prueba de Caja Negra - Inicio de Sesión.....	95
Figura 3.39. Prueba de Caja Negra – Control de inventario.	96

CAPÍTULO I

MARCO PRELIMINAR

1.1. INTRODUCCIÓN.

En el mundo cambiante en el que vivimos la tecnología y la informática toma un papel tan importante, surge la necesidad de optimizar y modernizar algunos de los procesos al interior de las organizaciones, dentro de esta tendencia la informática funciona como un eje transversal que sirve de apoyo a la gestión y administración de todos estos procesos facilitando el mejoramiento continuo en toda organización.

En la actualidad los sistemas informáticos hacen un gran aporte al mundo entero, facilitando el manejo de la información, resolviendo las necesidades y problemas que aquejan a las personas. En esta sociedad los mundos globalizados de la informática se constituyen en una necesidad básica para el funcionamiento óptimo de cualquier organización, ya que un sistema de información proporciona grandes beneficios, así como compartición de información, compartición de hardware y software, soporte administrativo, reducción de costos, y aumento de la cobertura geográfica.

La tecnología informática nos aporta valiosas herramientas para mejorar o perfeccionar los trabajos y actividades que desempeñamos a diario, por lo tanto, es importante que las instituciones de formación profesional se apropien de dichos recursos para brindar un mejor servicio a la comunidad administrativa, docente y estudiantil.

Es por ello que la carrera de Electricidad Industrial del Instituto Tecnológico Marcelo Quiroga Santa, actualmente tiene dos laboratorios para diferentes áreas, un taller de instalaciones eléctricas y un taller electromecánico, destinado a la formación académica de estudiantes. Pero lamentablemente la atención en los dos laboratorios y talleres a los docentes y estudiantes se realiza de forma manual. Debido a que se hace uso de cuadernos registros para el control de inventario, control de estudiantes, control docente y control de préstamos.

No cuenta con un sistema información automatizada y actualizada, que ayude a estas tareas a optimizar la atención de manera ágil.

Por lo anteriormente dicho se plantea este proyecto titulado “SISTEMA WEB PARA LA GESTION DE LABORATORIOS Y TALLERES, CASO: CARRERA DE ELECTRICIDAD INDUSTRIAL DEL I.T.M.Q.S.C.”, el cual tiene como finalidad el estudio, diseño y desarrollo de un sistema de información para satisfacer las necesidades detectadas en cuanto al manejo y tratamiento de la información de gestión de laboratorios y talleres. Permitiendo agilizar los procesos, mejorar el tratamiento de los datos y mantener una base de datos actualizada que beneficie a todos los actores participantes de la comunidad académica. Evitar daños de equipos, instrumentos y herramientas.

Para el presente proyecto de grado la metodología de desarrollo de software que se hace uso es Scrum y OOHDM para el desarrollo de aplicaciones hipermedia. Las herramientas que se emplean con respecto a la página web son PHP, Laravel como framework y con respecto a la base de datos MySQL.

1.2. ANTECEDENTES.

1.2.1. Antecedentes Institucionales.

➤ Contexto General

El presente proyecto de grado se desarrolla para la carrera de Electricidad Industrial de la sede central del Instituto Tecnológico “Marcelo Quiroga Santa Cruz”. La institución cuenta con una sede central y sus subsedes.

La sede central del Instituto Tecnológico “Marcelo Quiroga Santa Cruz” se halla funcionando desde el mes de abril del 2009 en la Zona de Villa Nuevo Potosí, entre calle final Luciano Alcoveza y Av. 9 de abril, perteneciente al Macro Distrito I Cotahuma de la ciudad de La Paz.

Acoge a una población estudiantil de las ciudades de La Paz y El Alto, en las carreras de Sistemas Informativos, Electricidad Industrial y Construcción Civil. Se encuentra ubicado en el distrito I de la ciudad de La Paz, entre calle final Luciano Alcoreza y Av. 9 de abril de la Zona de Villa Nuevo Potosí. Que se halla funcionando desde el mes de abril de 2009 para estudiantes externos, no dependientes de las penitenciarias.

La institución cuenta con una infraestructura nueva, mobiliario adecuado con todos los ambientes que exige la norma para el funcionamiento de una institución de Educación Superior Industrial y/o Comercial y equipamiento de laboratorios y talleres.

Las subsedes del Instituto Tecnológico “Marcelo Quiroga Santa Cruz” están ubicados en los Centro Penitenciario de San Pedro (frente a la plaza Sucre_entre Cañada Strongest y la Avenida 20 de Octubre), Centro Penitenciario de Mujeres Miraflores (entre la Avenida Héctor Ormachea y la calle 7) y Centro Penitenciario Chonchocoro (en la provincia de Ingavi, al noroeste de la ciudad de La Paz, aproximadamente a 30 kilómetros). La institución cuenta con más de 34 años de vida, en los centros penitenciarios.

➤ **Fundamento Legal**

Fundado el 11 de abril de 1986 y amparado en la Resolución Ministerial 1293 del 25 de septiembre de 1991, como Instituto Superior de Educación Comercial Mcal. “Antonio José de Sucre”.

Actualmente con una nueva Resolución Ministerial 039/13 por cambio de nombre a “Tecnológico Marcelo Quiroga Santa Cruz”.

Se identifica plenamente como una Institución Estatal de Educación Técnica Superior.

➤ Oferta Curricular

Formadora de profesionales en las carreras de Sistemas Informáticos, Electricidad Industrial, Construcción Civil, Contaduría General, Mercadotecnia, Secretariado Ejecutivo, y cursos de capacitación, para satisfacer las demandas pertinentes y funcionales del mercado ocupacional. Cuya oferta curricular es la siguiente:

Tabla 1.1. Carreras del Tecnológico Marcelo Quiroga Santa Cruz.

CÓDIGO Y SIGLA	CARRERAS	NIVEL ACADÉMICO	REGIMEN ACADÉMICO	TIEMPO DE ESTUDIO	CARGA HORARIA
(04-SIS)	Sistemas Informáticos	Técnico Superior	Anualizado	3 Años	3600 horas
(02-EIN)	Electricidad Industrial	Técnico Superior	Semestralizado	6 Semestres	3600 horas
(02-CCC)	Construcciones Civiles	Técnico Superior	Semestralizado	6 Semestres	3600 horas
(03-CNG)	Contaduría General	Técnico Superior	Anualizado	3 Años	3600 horas
(03-MER)	Mercadotecnia	Técnico Superior	Anualizado	3 Años	3600 horas
(03-SEJ)	Secretariado Ejecutivo	Técnico Superior	Anualizado	3 Años	3600 horas

Fuente: Ministerio de Educación R. M. N° 066 (2020).

➤ Visión

El Tecnológico Marcelo Quiroga Santa Cruz se constituye en una institución de formación profesional de alto nivel competitivo y productivo, ofreciendo carreras de educación técnica superior de acuerdo a las demandas de la sociedad, en áreas industriales y comerciales, superando viejas dicotomías, para ofrecer una formación integral donde los estudiantes puedan satisfacer las demandas del mundo actual y necesidades de la futura sociedad.

➤ **Misión**

La misión del Tecnológico Marcelo Quiroga Santa Cruz es formar y capacitar profesionales íntegros, éticos, críticos-reflexivos y competitivos, sin tomar en cuenta el límite de edad, color, religión, género u otra situación social, comprometidos con el desarrollo económico, político, social y cultural de su comunidad y con el uso sostenible de los recursos naturales, a través de carreras industrial y comercial a nivel técnico superior y cursos de capacitación y especialización en beneficio del bien común y de su entorno en general.

➤ **Población Estudiantil**

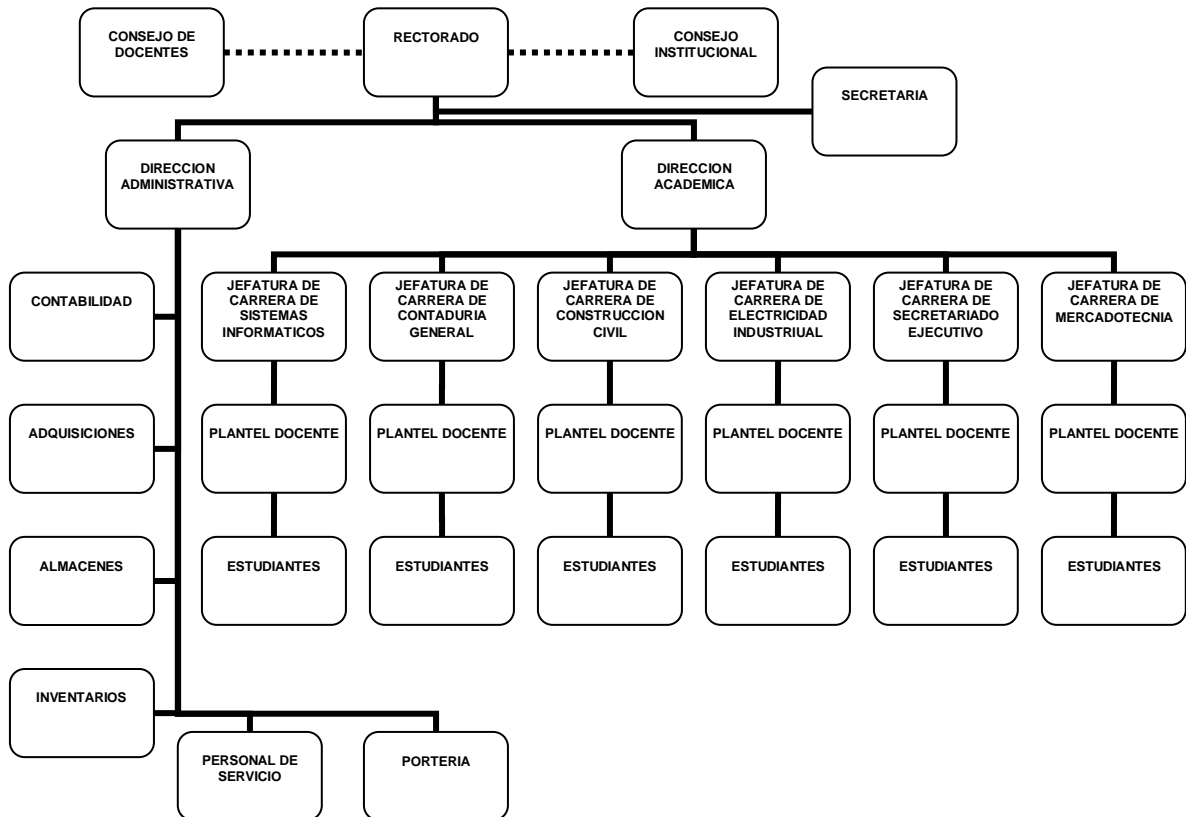
Por estar ubicada la sede central del Tecnológico Marcelo Quiroga Santa Cruz en una zona intermedia entre la ciudad de La Paz y la ciudad de El Alto, acoge una población estudiantil de ambas ciudades, en dos turnos: mañana y noche. Y en sus anexos o subsedes Centro Penitenciario de San Pedro, Centro Penitenciario de Chonchocoro y Centro Penitenciario de Mujeres Miraflores.

Acoge a estudiantes reclusos privados de libertad o personas que se encuentran recluidos en la cárcel. Beneficiándolos para que desarrollen actividades educativas de formación profesional durante su permanencia en prisión y de esa forma reducir su condena.

Según la Ley 2298 (2001), una de las recompensas establecidas es la redención, también conocida como 2X1. Según esta recompensa, cada dos días que el interno haya trabajado o estudiado dentro del penal, podrá redimir un día de pena.

➤ Estructura Organizacional

Figura 1.1. Estructura Orgánica del Tecnológico Marcelo Quiroga Santa Cruz.



Fuente: FODA_ Tecnológico Marcelo Quiroga Santa Cruz (2020).

1.2.2. Antecedentes de trabajos afines.

Los antecedentes referentes al diagnóstico y monitoreo de trabajos similares que se hallaron en la revisión bibliográfica del presente trabajo, los cuales son:

Castillo y Guzmán (2019), presentaron una propuesta de tesis denominada “Sistema Web para la mejora de la gestión administrativa de los laboratorios de cómputo en la Universidad Nacional de Trujillo”.

El propósito de este trabajo es desarrollar un Sistema Web para la mejora de la Gestión Administrativa de los Laboratorios de Cómputo, logrando que la disponibilidad, acceso y transferencia de archivos se realicen de una manera rápida, confiable y en tiempo real, lo que redundará en una disminución notable de tiempos en reportes y consultas, mejorando así el procesamiento de información en los laboratorios de cómputo. Sigue un estudio basado en la metodología RUP (Proceso Unificado de Rational), el cual permite desarrollar el software de manera iterativa, identificando los riesgos tempranamente, presentando un proyecto ordenado y de calidad. En la implementación del sistema se utilizó el lenguaje de programación JAVA con el framework Spring y base de datos PostgreSQL. Para dar solución a los procesos que realizan manualmente, generando pérdidas de tiempo al querer acceder a la información.

Horstmeier (2014), presenta una propuesta denominada “Implementación de sistema web de gestión de laboratorios de computación de la Facultad de Ciencias Empresariales”.

El propósito fundamental del proyecto es implementar un sistema web de gestión de laboratorios, que ayude al manejo de datos para las estadísticas obtenidas del uso de los laboratorios (herramientas, uso de computadores, insumos, horarios, eventos y clases). Implementar sistema con una interfaz gráfica que sea amigable para el usuario. Vincula las asignaturas con los diferentes laboratorios a través de una nueva metodología de trabajo y fomentando una utilización adecuada a los requerimientos de las asignaturas. Crea un modelo de base de datos para los datos del sistema web. Implementa un sistema de “feedback” para dar a conocer las falencias y observaciones realizadas a los laboratorios, considerando los resultados de la comisión evaluadora. La metodología que se utilizara es XP (Extreme Programming), las herramientas a utiliza para la construcción de desarrollo del sistema son MySQL para el modelado de la Base de Datos, Apache para el servidor Web, HTML para el diseño de página, PHP Para los procesos, cálculos y funciones del sistema y CSS para el entorno gráfico de la página web.

Buitrago y Gonzales (2013), realizó una propuesta denominada “Portal Web PARA la gestión de instrumentos de laboratorio de la Universidad Católica de Colombia”.

El propósito fundamental del trabajo es diseñar e implementar un portal Web que permita el ingreso de usuarios específicos (Auxiliares de laboratorio), para gestionar los préstamos de los instrumentos de laboratorio de la facultad de ingeniería de la Universidad Católica de Colombia así mismo como brindar otras funciones que mejoren la administración de los laboratorios. donde los usuarios de laboratorio pueden ingresar por medio de un nombre de usuario y una contraseña, para realizar la entrega de instrumentos a los estudiantes, y con la ayuda de un lector de código de barras identificar tanto el código del alumno como el de los instrumentos involucrados. Debido a que los préstamos de instrumentos para laboratorio se vienen realizando a mano, por parte del personal de laboratorio, lo cual ha motivado a innovar el sistema y darle un toque de vanguardia de acuerdo con la tecnología que actualmente se encuentra disponible y al alcance de todos. Se diseñó un prototipo de un Portal Web especializado que se acople con la gestión de los instrumentos, teniendo en cuenta las soluciones de tipo LAMP y basada en la funcionalidad de los CMS. De tal forma que refleje la usabilidad del sitio a los usuarios.

1.3. PLANTEAMIENTO DEL PROBLEMA.

1.3.1. Descripción de la situación problemática.

De acuerdo al diagnóstico realizado empleando la técnica de observación directa, la entrevista y la investigación documental, en la carrera de Electricidad industrial del Tecnológico Marcelo Quiroga Santa Cruz, actualmente existen falencias en el registro de inventariado de equipamiento, registro estudiantil y docente para la solicitud de préstamos de equipos, instrumentos y herramientas de los laboratorios, taller de instalaciones eléctricas y taller electromecánico. Servicio de atención para préstamos de equipos, instrumentos y herramientas a los docentes y estudiantes con demora.

Desconocimiento de información de parte del estudiantado de equipos, instrumentos y herramientas, para su préstamo. Además, para el inventariado trimestral el jefe de carrera y plantel docente deben realizar grupos de trabajo, para la revisión de equipos, instrumentos y herramientas con ayuda del cuaderno de registros, lo cual genera demora.

1.3.2. Problema Principal.

El registro manual del inventario de equipamiento, registro manual docente estudiantil y servicios de préstamo manual a los docentes y estudiantes generan una deficiente atención a la población estudiantil y docente de la carrera de Electricidad Industrial del Instituto Tecnológico Marcelo Quiroga Santa Cruz, provocando pérdida de tiempo en la espera, molestias, desmotivación, errores en informes de descargo y pérdidas económicas.

1.3.3. Problemas Secundarios.

- El acceso manual al inventariado genera errores y pérdida de tiempo.
- El registro manual de estudiantes y docentes provoca molestias y desmotivación, debido a la búsqueda de datos.
- El servicio manual de préstamo de equipos, instrumentos y herramientas para los laboratorios, taller de instalaciones eléctricas y taller electromecánico provocan pérdida de tiempo en los docentes y estudiantes.
- El registro manual de informes referentes a altas y bajas del equipamiento y servicio de préstamo, provoca demora en su envío a la autoridad correspondiente (Dirección Administrativa de la institución).
- La información desorganizada de inventariado de equipamiento y servicio de préstamos ocasiona una toma de decisiones no oportunas.

1.4. PLANTEAMIENTO DE OBJETIVOS.

1.4.1. Objetivo Principal.

Desarrollar e implementar un Sistema de información basado en la web para la gestión de inventarios y prestamos de los laboratorios y taller de instalaciones eléctricas, que proporciona información oportuna para la obtención de datos eficiente y confiable para la toma de decisiones en la carrera de Electricidad Industrial del Instituto Tecnológico Marcelo Quiroga Santa Cruz.

1.4.2. Objetivos Secundarios.

- Diagnosticar el actual proceso de movimiento del equipamiento en los laboratorios, taller de instalaciones eléctricas y taller electromecánico.
- Realizar un registro de inventario adecuado con el propósito de concretar el equipamiento asignado a la carrera de electricidad industrial.
- Diseñar una interfaz web para el control de inventario y servicios de préstamo de equipos, instrumentos y herramientas que pueda ser amigable para los usuarios.
- Desarrollar informes automatizados con datos confiables.
- Realizar pruebas del sistema de información web y evaluar resultados.

1.5. JUSTIFICACIÓN.

1.5.1. Justificación Técnica.

Para el desarrollo del presente proyecto se cuenta con software y herramientas adecuadas, con equipos necesario para la implementación del proyecto. También se tiene la predisposición de mejorar continuamente los equipos de la carrera de Electricidad Industrial.

1.5.2. Justificación Económica.

El sistema es justificable económicamente, porque ayudará a mejorar la gestión de los laboratorios, taller de instalaciones eléctricas, taller electromecánico e inventariado de equipamiento, permitiendo efectuar procesamiento de datos eficiente y oportuno para la toma de decisiones adecuada en la carrera de Electricidad Industrial del Instituto Tecnológico Marcelo Quiroga Santa Cruz. Además, en la fase de desarrollo del sistema de información web se usará por completo software libre y software privativo, esto quiere decir que no se incurrirá en gastos extras como por ejemplo los costos por pago de licencias.

1.5.3. Justificación Social.

El sistema Web para la gestión de los laboratorios, taller de instalaciones eléctricas y taller electromecánico, permitirá efectuar procesamiento de datos eficiente y oportuno para la toma de decisiones en la carrera de Electricidad Industrial del Instituto Tecnológico Marcelo Quiroga Santa Cruz. La implementación del Sistema información Web para la gestión de los laboratorios, taller de instalaciones eléctricas y taller electromecánico, se justifica socialmente ya que la población estudiantil, docente y personal administrativo se beneficiarán con un mejor servicio. Teniendo en cuenta que el inventariado actualizado de equipamiento beneficiara en el préstamo de equipos, instrumentos y herramientas a los estudiantes y docentes.

1.6. METODOLOGÍA.

La metodología de investigación es descriptiva, conocida también como la investigación estadística, ya que describen los datos, el objetivo de esta investigación consiste en llegar a conocer las situaciones, costumbres y actitudes a través de la descripción exacta de las actividades y procesos, el objetivo principal es saber él porque y para que se está realizando.

1.6.1. Método de Ingeniería.

La metodología de desarrollo de software que se usara es Scrum, una metodología ágil, que se basa en el control de procesos empíricos con un conjunto de prácticas y roles.

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

Según Huanca (2015), la metodología Scrum cumple un proceso que realiza entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto, scrum también se utiliza cuando es necesario identificar y solucionar ineficiencias.

Este método de trabajo promueve la innovación, motivación que forma parte de este proyecto, los beneficios que se tiene esta metodología son el cumplimiento de las expectativas, flexibilidad en cambios, mayor calidad de software, reducción de riesgos.

En el proceso de Scrum se desarrolla iteraciones llamadas "Sprint", el equipo de desarrollo decide que funcionalidad incluir o no, el tiempo necesario para terminar el proyecto. Conlleva tres fases: Pre game, Game, Post game.

Por otra parte, se trabajará con la metodología OOHDM que propone el desarrollo de aplicaciones hipermedia a través de un proceso compuesto por cuatro etapas: diseño conceptual, diseño navegacional, diseño de interfaces e implementación. OOHDM propone el desarrollo de aplicaciones hipermedia a través de un proceso compuesto por cuatro Fases: diseño conceptual, diseño navegacional, diseño de interfaces abstractas e implementación.

1.7. HERRAMIENTAS.

Las herramientas que se emplearan para el desarrollo del presente proyecto de grado son:

- **PHP**, 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. PHP es un lenguaje de programación de servidores y una poderosa herramienta para crear páginas web dinámicas e interactivas. PHP es una alternativa eficiente, gratuita y ampliamente utilizada frente a competidores como ASP de Microsoft.
- **Gestor de Base de Datos MariaDB**: MariaDB es un sistema gestor de bases de datos (SGBD), es decir, un conjunto de programas que permiten modificar, almacenar, y extraer información de una base de datos. Disponiendo de otro tipo de funcionalidades como la administración de usuarios, y recuperación de la información si el sistema se corrompe, entre otras.
- **Laravel**, es un framework PHP. Es uno de los frameworks más utilizados y de mayor comunidad en el mundo de Internet. Como framework resulta bastante moderno y ofrece muchas utilidades potentes a los desarrolladores, que permiten agilizar el desarrollo de las aplicaciones web. El objetivo de Laravel es el de ser un framework que permita el uso de una sintaxis refinada y expresiva para crear código de forma sencilla, evitando el «código espagueti» y permitiendo multitud de funcionalidades.
- **HTML (lenguaje de marcas de hipertexto)**: hace referencia al lenguaje de marcado para la elaboración de páginas web. Define una estructura básica y un código para la definición de contenido de una página web, como texto, imágenes, videos, entre otros.

- **JAVASCRIPT:** JavaScript es un lenguaje de scripting multiplataforma y orientado a objetos. Es un lenguaje pequeño y liviano. JavaScript contiene una librería estándar de objetos, tales como Array, Date, y Math, y un conjunto central de elementos del lenguaje, tales como operadores, estructuras de control, y sentencias.
- **BOOTSTRAP:** es un framework o conjunto de herramientas de Código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones.
- **AJAX:** es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores.
- **CSS:** Son hojas de Estilo en Cascada – que es un lenguaje que descubre la presentación de los documentos estructurados en hojas de estilo para mostrar un documento en pantalla.

1.8. LÍMITES Y ALCANCES.

1.8.1. Límites.

Los límites del sistema de información son los siguientes:

- El sistema estará restringido a los tipos de usuarios autorizados para realizar cambios en el sistema.
- El sistema solo se enfocará en la información de los inventarios, registro de estudiantes, registro de docentes, prestamos de equipos, instrumentos y materiales; por lo tanto, no contará con conexiones a otros sistemas de información que tuviera el instituto.
- El sistema no evaluara a los docentes y estudiantes.

- El sistema no estará encargado del registro de notas del estudiante, ni la asistencia.

1.8.2. Alcances.

Los alcances del sistema web se definen a través de los siguientes módulos:

- Módulo de inicio de sesión.
- Módulo de estudiantes y docentes.
- Módulo de usuarios.
- Módulo de control de inventarios.
- Módulo de seguimiento de préstamos.
- Módulo de reportes y módulo de estadística.

1.9. APORTES.

Los principales aportes del presente proyecto de grado se detallan a continuación:

1.9.1. Practico.

El principal aporte practico es poder brindar un óptimo control y seguimiento del flujo de información al movimiento de inventario y prestamos en la carrera de Electricidad Industrial del Instituto Tecnológico Marcelo Quiroga Santa Cruz, automatizando los procesos de dicha información, mediante el empleo de una interfaz web.

1.9.2. Teórico.

El aporte teórico va relacionado en la adaptabilidad que conlleva incorporarse a la web 2.0., ya que estará desarrollado con una metodología ágil y una metodología orientada a la Web.

Además de llevar herramientas actualizadas como las hojas de estilo CSS, con framework Laravel entre otros. Se verán los movimientos como actualización, inserción y eliminación de datos que han realizado los usuarios del sistema sobre este, gracias a un módulo de trazabilidad, guardando en tablas los movimientos generados.

CAPÍTULO II

MARCO TEÓRICO

2.1 INTRODUCCIÓN.

En este capítulo veremos los conceptos para la realización de proyecto, así mismo se puntualizará sobre las metodologías, métodos y técnicas, de esta manera brindar la facilidad y entendimiento para interactuar con la metodología aplicada al proyecto.

2.2. DEFINICIÓN DE SISTEMAS DE INFORMACIÓN.

En cuanto los sistemas de información, Hernández afirma que:

Los Sistemas de información han ido evolucionando durante los últimos años hasta constituir los denominados sistemas de información estratégicos. Primeramente, los Sistemas de Información empresariales eran considerados como un instrumento simplificador de las distintas actividades de la empresa, una herramienta con la cual se facilitaban los tramites y reducía la burocracia. Su finalidad era básicamente llevar la contabilidad y el procesamiento de los documentos que a nivel operativo. Posteriormente el desarrollo de la informática y las telecomunicaciones permitieron incrementar la eficacia en la realización de las tareas, ahorrar tiempo en el desarrollo de las actividades y almacenar la mayor cantidad de información en el menor espacio posible, lo cual aumentó en las organizaciones el interés en los sistemas de información. (Hernandez, 2020, pág. 4)

Según Andreu, Ricart y Valor (1991) un sistema de información es un conjunto formal de procesos que, operando sobre una colección de datos estructurada de acuerdo a las necesidades de la empresa, recopila, elabora y distribuyen selectivamente la información necesaria para la operación de dicha empresa y para las actividades de dirección y control correspondientes, apoyando, al menos en parte, los procesos de toma de decisiones necesarios para desempeñar funciones de negocio de la empresa de acuerdo con su estrategia.

Pressman afirma que: "...los componentes estándares para un sistema de información son subsistemas, módulos, pantallas, reportes, programas interactivos, programas en lote, archivos, LOC e instrucciones en el nivel objeto. El planificador del proyecto estima el número de ocurrencias de cada componente estándar y luego usa datos de proyecto históricos para estimar el tamaño entregado por componente estándar". (Pressman, 2010, pág. 600)

2.3. DEFINICIÓN DE INVENTARIO.

Pressman define al inventario como:

El inventario puede ser nada más que un modelo de hojas de cálculo que contenga información que ofrezca una descripción detallada (por ejemplo, tamaño, edad, importancia empresarial) de cada aplicación activa. Al ordenar esta información de acuerdo con importancia empresarial, longevidad, mantenibilidad actual, soportabilidad y otros importantes criterios locales, aparecen los candidatos para reingeniería. Entonces pueden asignarse recursos a esas aplicaciones. Es importante observar que el inventario debe revisarse con regularidad. (Pressman, 2010, pág. 662)

Muller afirma que: "Los inventarios de una compañía están constituidos por sus materias primas, sus productos en proceso, los suministros que utiliza en sus operaciones y los productos terminados. Un inventario puede ser algo tan elemental como una botella de limpiador de vidrios empleada como parte del programa de mantenimiento de un edificio, o algo más complejo, como una combinación de materias primas y sub ensamblajes que forman parte de un proceso de manufactura." (Muller, 2005, pág. 1)

El inventario llega a ser el conjunto de bienes tangibles, propios y disponibles a la venta ya sea en su estado original o cuando fue transformado para ser vendida. (Moreno, 2009)

2.3.1. Tipos de Inventario.

De acuerdo a (Lacalle, 2013) de acuerdo a las características de las empresas existen cinco tipos de inventarios:

- a) Inventario Periódico: Lo constituyen todos aquellos bienes que le pertenecen a la empresa bien sea comercial o mercantil, los cuales los compran para luego venderlos sin ser modificados.
- b) Inventario de productos terminados: Son todos aquellos bienes adquiridos por las empresas manufactureras o industriales, los cuales son transformados para ser vendidos como productos elaborados.
- c) Inventario de Productos en Proceso de Fabricación: Lo integran todos aquellos bienes adquiridos por las empresas manufactureras o industriales, los cuales se encuentran en proceso de manufactura.
- d) Inventario de Materias Primas: Lo conforman todos los materiales con los que se elaboran los productos, pero que todavía no han recibido procesamiento.
- e) Inventario de Suministros de Fábrica: Son los materiales con los que se elaboran los productos, pero que no pueden ser cuantificados de una manera exacta (Pintura, lija, clavos, lubricantes, etc.).

Asimismo, pueden tipificarse según la forma en que se crearon existen cuatro tipos de inventarios: de ciclo, de seguridad, de previsión y en tránsito. Los cuales se desarrollan a continuación:

- Inventario de ciclo: son los inventarios necesarios para satisfacer la demanda promedio durante el tiempo entre aprovisionamientos sucesivos.
- Inventario de seguridad: se refiere al excedente del inventario que se mantiene para protegerse contra los cambios esperados de la demanda, el tiempo de espera y cambios en abastecimiento.

- Inventario de previsión: es utilizado para absorber las irregularidades que se presenten en las tasas de demanda y oferta, acumulando inventario en los periodos de baja demanda, con el fin de no incrementarlos cuando la demanda alcance sus puntos máximos.
- Inventario en tránsito: es el inventario que se mueve de un punto a otro en flujo continuo de materiales. (Krajewski, Malhotra y Ritzman, 2008)

Según Muller (2005) los inventarios pueden llegar a dividirse en las siguientes categorías generales: materias primas, productos terminados y productos en proceso. De acuerdo a su manera funcional divide el inventario en: artículos de consumo artículos para servicio y reparación.

2.3.2. Control de inventario de equipos.

Los sistemas de control de inventarios son necesarios en las empresas ya que su finalidad es conseguir un nivel de almacenamiento que minimice los costos totales relacionados con el inventario, manteniendo a su vez bajo control la posibilidad de que el cliente o el proceso receptor, queden desabastecidos. Existen dos sistemas básicos de control de inventarios (Taha, 2004)

“El inventario representa uno de los activos más importantes en muchas empresas; además, constituye la mayor parte de la fuente de ingresos, tanto de las empresas comerciales como industriales, de modo que influye significativamente en el resultado de un período y en la situación financiera de una empresa, y es, por lo general, el activo circulante mayor en su balance general” (Villa, 2008, pág. 227)

Asimismo, Sierra, J, et.al. (2015) afirma que el: “Control de Inventarios: Es el dominio que se tiene sobre los haberes o existencias pertenecientes a una organización. En la práctica el control de inventarios no resulta tan fácil como su definición. Por sí mismo el Control de Inventarios es un sistema que está subordinado a otros sistemas mayores que tienen como fin último operar para el logro de los objetivos generales de toda la organización.” (p.8)

Figura 1.1. Proceso del Control de Inventarios.



Fuente: Sierra (2015).

2.3.3. Método de valuación de inventarios.

Para Rentería (2005) el sistema de inventarios perpetuos la determinación del costo varía de acuerdo con el método de valuación de inventarios seleccionado y aunque las adquisiciones y los gastos sobre compra se registran por igual en los métodos, las salidas de almacén varían en cada uno de los 4 principales métodos: primeras entradas, primeras salidas; últimas entradas, primeras salidas; precio promedio y detallista.

Los métodos de valoración o **métodos de valuación de inventarios** son técnicas utilizadas con el objetivo de seleccionar y aplicar una base específica para valuar los **inventarios** en términos monetarios. La valuación de inventarios es un proceso vital cuando los precios unitarios de adquisición han sido diferentes. (Salazar, 2019)

Respecto la valuación de los inventarios Fuertes afirma lo siguiente:

“Con el desarrollo de nuevas normas contables, la valuación de los inventarios en el estado de situación financiera, y la recuperabilidad de estos elementos también afecta al análisis del rendimiento financiero de la entidad. Es decir, tanto el importe que se reconoce inicialmente en los inventarios se deriva de la necesidad de que los costos necesarios se reconozcan cuando el ingreso producto de la venta se haya realizado, así como el importe que se reconoce en el costo de ventas se deriva de la valuación del activo y de la recuperabilidad del mismo”. (Fuertes, 2015, pág. 49)

2.3.3.1. PEPS

El método PEPS (Primeras entradas, primeras salidas) consiste en tener identificados los productos que ingresaron primero para darle salida inmediata del almacén ya sea por venta o traspaso.

En cualquiera de los métodos las compras no tienen gran importancia, puesto que estas ingresan al inventario por el valor de compra y no requiere procedimiento especial alguno.

En el caso de existir devoluciones de compras, esta se hace por el valor que se compró al momento de la operación, es decir se la de salida del inventario por el valor pagado en la compra.

Si lo que se devuelve es un producto vendido a un cliente, este se ingresa al inventario nuevamente por el valor en que se vendió, pues se supone que cuando se hizo la venta, a esos productos se les asignó un costo de salida según el método de valuación de inventarios manejado por la empresa.

Este método puede ser controlado o supervisado mediante un Kardex en donde se especifiquen campos necesarios de los productos o de la compra realizada, algunos de estos campos pueden ser:

- Fecha.
- Detalle de producto.
- Cantidad (Número de productos que ingresaron al almacén)
- Costo unitario del producto.
- Total en moneda de las entradas.
- Cantidad (Número de productos que salen de tu almacén)
- Costo unitario del producto.
- Total en moneda de las salidas.
- Cantidad (Número de productos existentes)
- Costo unitario de tus existencias actuales.
- Total en peso de las existencias.

2.4. INGENIERÍA WEB.

En la Ingeniería de Software tradicional, hay tareas que forman parte en el proceso como lo es el control y garantía de la calidad, que implican actividades como: supervisión de estándares, revisiones técnicas, análisis, seguimiento, entre otras, estas pueden ser igualmente aplicadas a la Ingeniería Web. Sin embargo, en la Ingeniería Web tiene otros aspectos para valorar la calidad como la usabilidad, accesibilidad, seguridad, eficiencia y mantenibilidad. (Pressman, 2010)

“La ingeniería Web es el proceso con el que se crean las WebApps de alta calidad. La Ingeniería Web no es un clon perfecto de la ingeniería de software, pero utiliza muchos conceptos y principios fundamentales de ella”. (Garcia, 2013, pág. 10)

Pinzon destaca que: “...en la Ingeniería Web tiene otros aspectos para valorar la calidad como la usabilidad, accesibilidad, seguridad, eficiencia y mantenibilidad. Las aplicaciones Web están dirigida frecuentemente a grupo de usuarios desconocidos haciendo que el desarrollo sea más desafiante a diferencia de, Aplicaciones de software convencionales que se desarrollan generalmente para un grupo de usuarios previamente conocido” (Pinzon, 2017, pág. 2)

2.4.1. Proceso de la Ingeniería Web.

Según Pressman (2010) este llega enumera actividades que forman parte del proceso de la Ingeniería Web y que son aplicables a cualquier aplicación Web independientemente de su tamaño y complejidad, los cuales son:

- La Formulación identifica objetivos y establece el alcance de la primera entrega.
- La Planificación genera estimación de costo, la evaluación de riesgo y el calendario del desarrollo y fechas de entrega.
- El Análisis especifica los requerimientos e identifica el contenido.
- La Modelización Consta de dos partes: diseño y producción del contenido.
- Diseño de la arquitectura, navegación e interfaz del usuario.
- En la Generación de Páginas se integran arquitectura, navegación e interfaz para la creación más visible del proyecto, que son las páginas.
- El Test son pruebas en busca de errores en todos los niveles: contenido, funcional, navegación, etc.
- El resultado final es sometido a Evaluación del Cliente.

El proceso de ingeniería web tiene sus características como inmediatez y transformación y el crecimiento continuo, que posibilita que el usuario se involucre, facilitando el desarrollo de productos que se ajustan mucho lo que éste busca y necesita. Los pasos más resaltantes del proceso de la ingeniería web: enunciación, planificación análisis, modelización, generación de páginas, test y evaluación del cliente. (Mi carrera universitaria, s/a)

2.5. INGENIERÍA DE REQUERIMIENTOS.

2.4.1. Definición de Ingeniería de Requerimientos.

Pressman afirma que: “El espectro amplio de tareas y técnicas que llevan a entender los requerimientos se denomina *ingeniería de requerimientos*. Desde la perspectiva del proceso del software, la ingeniería de requerimientos es una de las acciones importantes de la ingeniería de software que comienza durante la actividad de comunicación y continúa en la de modelado. Debe adaptarse a las necesidades del proceso, del proyecto, del producto y de las personas que hacen el trabajo. (Pressman, 2010, pág. 102)

Por otra parte, la ingeniería de requerimientos es “El proceso para establecer los servicios que el sistema debería proveer y las restricciones bajo las cuales debería operar y ser desarrollado, se llama Ingeniería de Requerimientos” (Cristián, 2014, pág. 3)

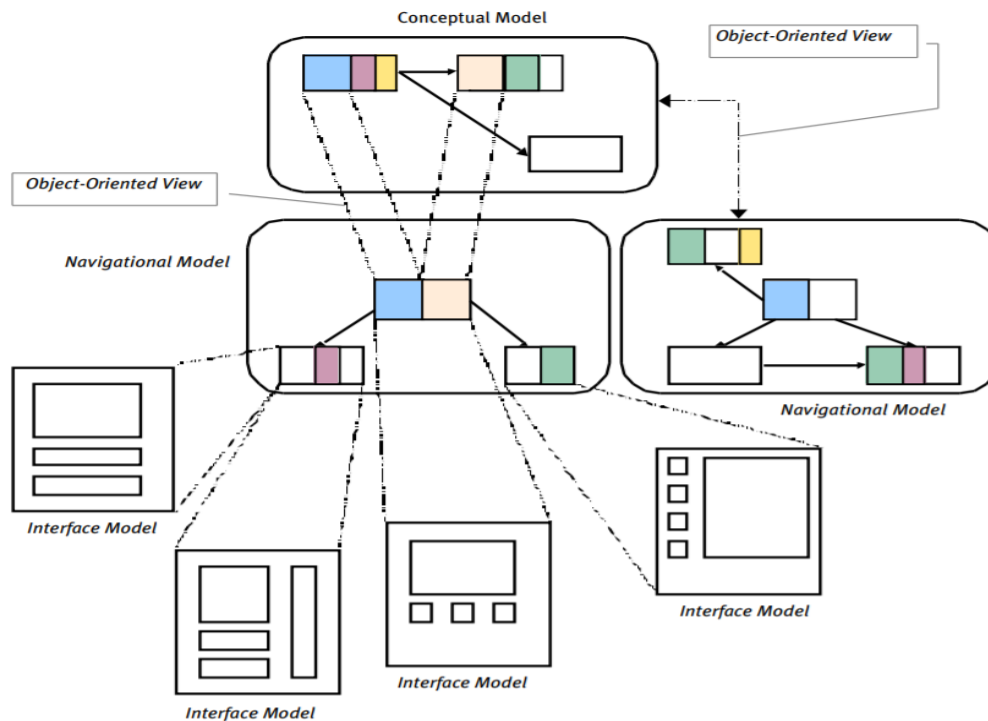
La ingeniería de requerimientos proporciona los mecanismos adecuados para comprender qué Espero que los clientes analicen sus necesidades, evalúen la viabilidad y negocien soluciones razonables. Especifique claramente las soluciones, verifique las especificaciones y gestione los requisitos Cuando se transforman en sistemas funcionales. (Pressman, 2010)

2.6. OBJECT ORIENTED HIPERMEDIA DESING METHODOLOGY (OOHDM).

Cárdenas afirma que Object Oriented Hipermedia Desing Methodology es “...un lenguaje de modelado orientado a aplicaciones hipermedia; sin embargo, al analizar su metodología se sintetiza en un proceso de desarrollo, relacionado con UML que es un lenguaje de modelado independiente, OOHDM lo usa en su proceso, muy similar a la secuencia de RUP (Rational Unified Process), con la diferencia de que RUP está orientado para el desarrollo de cualquier aplicación, mientras que OOHDM hace énfasis en la organización de los objetos y los aspectos de la navegación para cumplir su propósito” (Cárdenas, 2011, pág. 105)

En la siguiente figura se muestran las relaciones entre el esquema conceptual, navegacional y los objetos de interfaz en OOHDM.

Figura 2.2. Relación entre objetos conceptuales, de navegación y de interfaz en OOHDM.



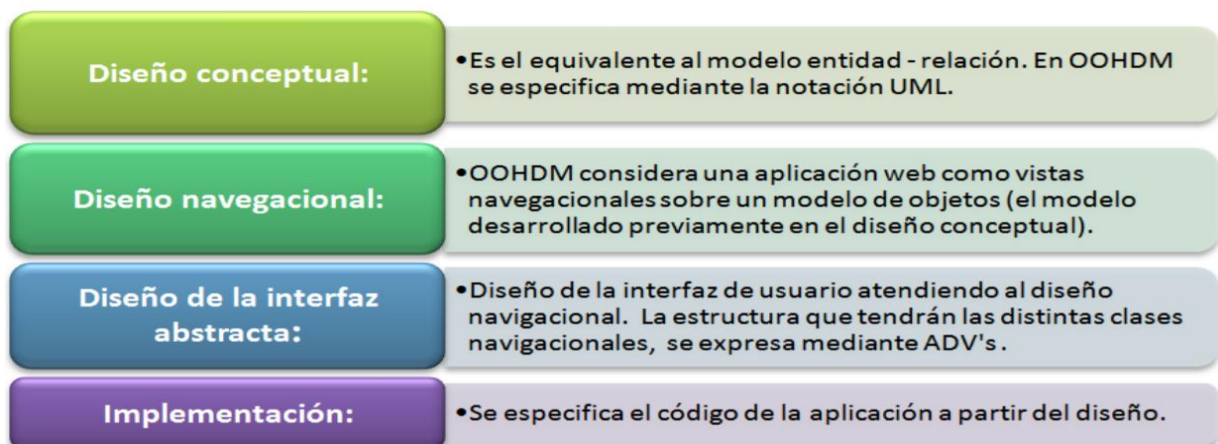
Fuente: Schwabe, D. y Rossi, G. (1997).

Según Schwabe y Rossi (1995) El método de desarrollo de hipertexto OOHDM introduce el modelado orientado a objetos en el desarrollo de hipertexto. En OOHDM se modela la navegación a través del diagrama de clases navegacionales y del diagrama de contextos.

2.6.1. Fases de OOHDM.

Vilariño afirma que: “OOHDM propone 4 etapas (fases) y consideran el paradigma del proceso incremental. Cada una de las etapas se enfoca en un diseño particular de una preocupación, en donde el resultado será un modelo OO. OOHDM es una extensión de HDM, pero lo que la distingue claramente es el proceso de concepción orientado a objetos” (Vilariño, 2010, pág. 34)

Figura 2.3. Etapas de OOHDM, para el desarrollo de software hipertexto.



Fuente: Rossi, G., Schmid, A., y Lyardet, F. (2003).

Según Lamarca (2013) OOHDM propone el desarrollo de aplicaciones hipertexto mediante un proceso de 4 etapas: diseño conceptual, diseño navegacional, diseño de interfaces abstractas e implementación. En la primera etapa se construye un esquema conceptual representado por los objetos de dominio o clases y las relaciones entre dichos objetos. Se puede usar un modelo de datos semántico estructural. El modelo OOHDM propone como esquema conceptual basado en clases, relaciones y subsistemas.

En la segunda etapa, el diseñador define clases navegacionales tales como nodos, enlaces y estructuras de acceso (índices y visitas guiadas) inducidas del esquema conceptual. Los enlaces derivan de las relaciones y los nodos representan ventanas lógicas (views) sobre las clases conceptuales. A continuación, el diseñador describe la estructura navegacional en términos de contextos navegacionales. Durante esta etapa, es posible adaptar los objetos navegacionales para cada contexto, de forma similar a las perspectivas de HDM. (Lamarca, 2013)

OOHDM no propone un modelo enriquecido para el dominio de la aplicación, por lo que deja libre al diseñador para elegir el modelo de especificación del dominio. La tercera etapa está dedicada a la especificación de la interfaz abstracta. Así, se define la forma en la cual deben aparecer los contextos navegacionales. También se incluye aquí el modo en que dichos objetos de interfaz activarán la navegación y el resto de funcionalidades de la aplicación, esto es, se describirán los objetos de interfaz y se los asociará con objetos de navegación. La separación entre el diseño navegacional y el diseño de interfaz abstracta permitirá construir diferentes interfaces para el mismo modelo navegacional. (Lamarca, 2013)

Y finalmente la cuarta etapa, dedicada a la puesta en práctica, es donde se hacen corresponder los objetos de interfaz con los objetos de implementación. (Lamarca, 2013)

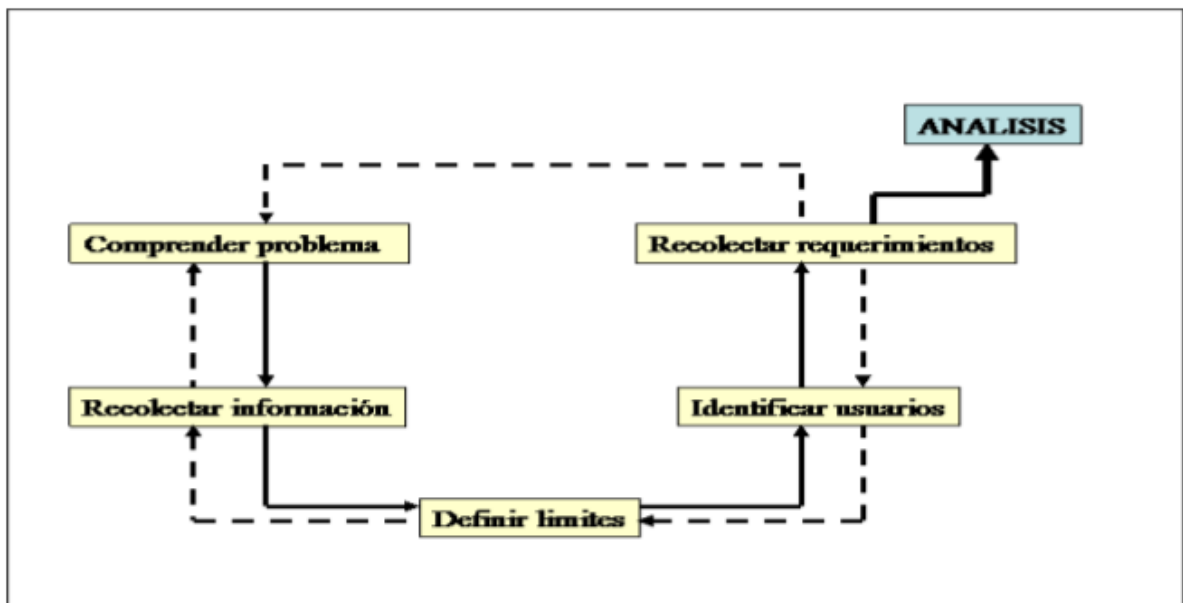
2.6.1.1. Obtención de requerimientos.

Pinto, J. (2015) afirma que: “La obtención de requerimiento se refiere a “La herramienta en la cual se fundamenta esta fase son los diagramas de casos de usos, los cuales son diseñados por escenarios con la finalidad de obtener de manera clara los requerimientos y acciones del sistema” (p.6)

Para Mejía (2009) en la obtención de los requerimientos existen las siguientes tareas a seguir (referencia):

- Comprender el problema que se va a resolver, para lo cual es necesario estudiar el dominio o entorno en el que el sistema va a operar.
- Buscar y recolectar información acerca del sistema a desarrollar, de manuales de operación y mantenimiento, de manuales organizacionales y políticas de operación.
- Definir los límites y restricciones del sistema para determinar con precisión qué es lo que el sistema va a hacer y también especificar lo que no va a hacer.
- Identificar a las personas o usuarios interesados en el sistema, ya que ellos conocen el medio ambiente en que operará el sistema y pueden ayudar describiendo sus necesidades.
- Recolectar y clasificar requerimientos, los desarrolladores pueden iniciar definiendo un bosquejo general del sistema, su funcionamiento básico y estableciendo su alcance. (Mejía, 2009, pág. 70)

Figura 2.4. El proceso de Obtención de Requerimientos.



Fuente: Mejía, P. (2009).

2.6.1.2. Diseño Conceptual.

French (1984) dice que es la fase donde la ciencia de ingeniería, el conocimiento práctico, los métodos de producción y los temas comerciales necesitan unirse y donde se toman las decisiones más importantes.

La fase conceptual es la etapa en la que se exigen más al diseñador, en la que se unen la ciencia de ingeniería, el conocimiento práctico, los métodos de producción, así como aspectos comerciales y donde se toman las decisiones más importantes (French, 1984)

Para Horváth (2005) indica que no existe una definición precisa carente de ambigüedades acerca de lo que es el diseño conceptual, dado que éste tiene diferentes fines y aparece de diferentes maneras en varias subdisciplinas, como la arquitectura, el diseño mecánico, diseño de interiores o diseño industrial. No obstante, todos estos poseen elementos comunes, y por tanto podemos resumir el proceso de diseño conceptual, como el conjunto de tareas encaminadas a obtener una solución a un problema planteado a partir de las especificaciones, requisitos y necesidades. El diseño conceptual sintetiza dicha solución en forma de conceptos, expresados en forma de esquemas.

2.6.1.3. Diseño Navegacional.

Se puede definir al diseño navegacional en OOHDM como "...un paso muy importante en el diseño de aplicaciones Web, un modelo navegacional está constituido como una vista sobre un diseño conceptual, admitiendo la construcción de modelos diferentes de acuerdo a los diferentes perfiles de usuario y como se efectuará la navegación entre ellos. (Martínez, 2012, pág. 32)

Por otro lado, para Soliz y Morales (2014) en OOHDM una aplicación se ve a través de un sistema de navegación. En esta fase se debe diseñar la aplicación teniendo en cuenta las tareas que el usuario va a realizar sobre el sistema. Para ello, hay que partir del esquema conceptual desarrollado en la fase anterior. Además, que sobre un mismo esquema conceptual se pueden desarrollar diferentes modelos navegacionales.

De esa manera para realizar un diseño navegacional Torres afirma que: “A la hora de realizar un modelo navegacional, hay que tener en cuenta qué objetos del modelo básico van a ser navegables, qué tipo de relaciones y estructuras de composición hay entre estos objetos navegables y las conexiones entre los distintos objetos y las posibilidades de navegación de un nodo a otro”. (Torres, J., et al, pág. 4)

2.6.1.4. Diseño de Interfaces Abstractas.

“La acción de diseño abstracto de la interfaz especifica los objetos de la interfaz que el usuario ve cuando ocurre una interacción con la webapp” el modelo Vista de Datos Abstractos expone “la relación entre objetos de interfaz y de navegación, así como las características de comportamiento de los objetos de interfaz.” (Pressman, 2010, pág. 333).

Una vez definida la estructura navegacional, hay que prepararla para que sea perceptible por el usuario y esto es lo que se intenta en esta fase. Esto consiste en definir qué objetos de interfaz va a percibir el usuario, y en particular el camino en el cuál aparecerán los diferentes objetos de navegación, qué objeto de interfaz actuarán en la navegación, la forma de sincronización de los objetos multimedia y el interfaz de transformaciones. Al haber una clara separación entre la fase anterior y esta fase, para un mismo modelo de navegación se pueden definir diferentes modelos de interfaces, permitiendo, así que el interfaz se ajuste mejor a las necesidades del usuario. (Schwabe & Rossi, pág. 15)

El diseño de interfaz abstracta permite definir los objetos que va a percibir el usuario, tanto el camino como la sincronización, empleándose en un modelo conocido como la vista de datos abstractos de manera ajustada para el usuario.

2.6.1.5. Implementación.

La actividad de implementación del MDHOO representa una iteración del diseño específica del ambiente en el que opera la webapp. Las clases, navegación e interfaz se caracterizan cada una en forma tal que pueden construirse para el ambiente cliente-servidor, sistemas operativos, software de apoyo, lenguajes de programación, y otras características ambientales que son relevantes para el problema. (Pressman, 2010, pág. 334)

Dado el cumplimiento de las cuatro fases de OOHDM, es decir “una vez obtenido el modelo conceptual, el modelo de navegación y el modelo de interfaz abstracta, sólo queda llevar los objetos a un lenguaje concreto de programación, para obtener así la implementación ejecutable de la aplicación” (Soliz D & Morales O, 2014, pág. 8).

Esta quinta fase permite llevar los objetivos a un lenguaje adecuado de programación, de una aplicación ejecutable, a continuación, se presenta el resumen de esta fase:

Tabla 1.1. Fase de Implementación de OOHDM.

Fase	Implementación
Productos	Aplicación ejecutable
Herramientas	El entono del lenguaje de programación
Mecanismos	Los ofrecidos por el lenguaje
Objetivo de diseño	Obtener la aplicación ejecutable

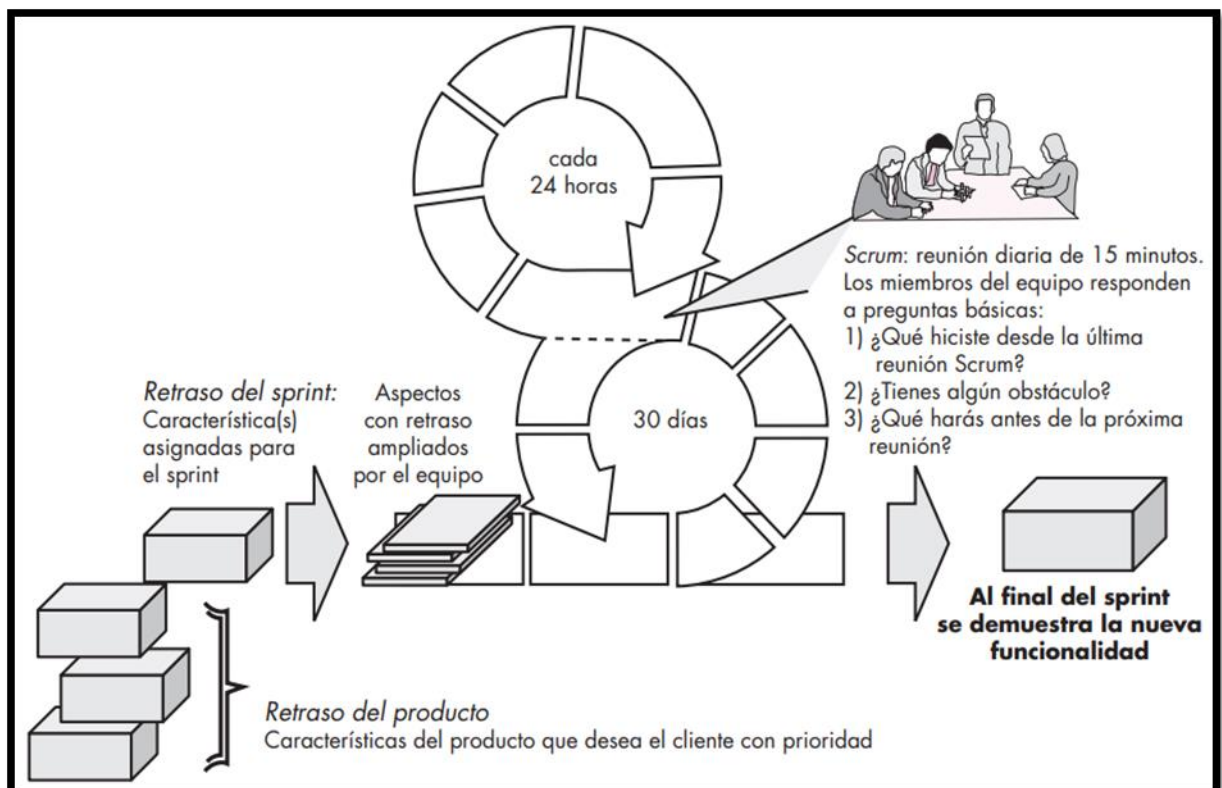
Fuente: Departamento de lenguajes y sistemas informáticos, Universidad SEVILLA (2001).

2.7. METODOLOGÍA AGIL SCRUM.

La metodología SCRUM “es un método de desarrollo ágil de software concebido por Jeff Sutherland y su equipo de desarrollo a principios de la década de 1990” (Pressman, 2010, pág. 69), los principios coadyuvan a las actividades estructurales a través de un patrón del proceso (sprint):

- Requerimientos
- Análisis
- Diseño
- Evolución
- Entrega

Figura 2.5. Flujo del Proceso SCRUM.



Fuente: Ingeniería del Software (Pressman, 2010).

2.7.1. Componentes de Scrum.

Los componentes de Scrum, cuenta con los componentes de reuniones y roles, mismos que se dividen en fases:

2.7.1.1. Reuniones.

Las reuniones forman parte de los artefactos de esta metodología junto con los roles y los elementos que lo forman.

Según Trigás Gallego (2012), define tres fases de las reuniones:

- a) Planificación de Backlog. Se definirá un documento en el que se reflejarán los requisitos del sistema por prioridades. En esta fase se definirá también la planificación del Sprint 0, en la que se decidirá cuáles van a ser los objetivos y el trabajo que hay que realizar para esta interacción.
- b) Seguimiento del Sprint. Se hacen reuniones diarias en las que las 3 preguntas principales para evaluar el avance de las tareas serán:
 - ¿Qué trabajo se realizó desde la reunión anterior?
 - ¿Qué trabajo se hará hasta una nueva reunión?
 - Inconvenientes que han surgido y qué hay que solucionar para poder continuar.
- c) Revisión del Sprint. Cuando se finaliza el Sprint se realizará una revisión del incremento que se ha generado. Se presentarán los resultados finales y una demo o versión, esto ayudará a mejorar el feedback.(pág. 35)

Es decir que el componente de las reuniones se divide en tres fases, primero la planificación (se debe obtener listas de tareas a realizar), segundo el seguimiento de sprint lo que permitirá evaluar y solucionar inconvenientes surgidos durante el desarrollo, y por último la tercera donde se presentan los resultados.

2.7.1.2. Roles o Responsabilidades.

Los roles o responsabilidades están divididos en dos, según Trigás Gallego (2012) estos son cerdos y gallinas:

1. Los cerdos. Son personas que están comprometidas con el proyecto y el proceso de Scrum:
 - Product Owner. Es la persona que toma las decisiones, y es la que realmente conoce el negocio del cliente y su visión del producto. Se encarga de escribir las ideas del cliente, las ordena por prioridad y las coloca en el Product Backlog.
 - ScrumMaster. Es el encargado de comprobar que el modelo y la metodología funcionan. Eliminará todos los inconvenientes que hagan que el proceso no fluya e interactuará con el cliente y con los gestores.
 - Equipo de desarrollo. Suele ser un equipo pequeño de unas 5 – 9 personas y tienen autoridad para organizar y tomar decisiones para conseguir su objetivo. Está involucrado en la estimación del esfuerzo de las tareas del Backlog.

2. Las gallinas. Aunque no son parte del proceso Scrum, es necesario que parte de la retroalimentación de la salida del proceso y así poder revisar y planear cada Sprint.
 - Usuarios. Es el destinatario final del producto.
 - Stakeholders. Las personas a las que el proyecto les produzca un beneficio. Participa durante las revisiones del sprint.
 - Managers. Toma las decisiones finales participando en la selección de los objetivos y de los requisitos.(pág.36)

Los roles se dividen en dos grupos: los que están comprometidos con el proyecto y proceso de SCRUM y los que no son parte del proceso pero que se necesitan para la realimentación de la salida de los procesos y el planeamiento del sprint. (Tymkiw, Bournissen, & Tumino, 2020, pág. 82)

El componente de roles o responsabilidades está dividido por dos grupos de los cuales cuentan tareas, el primer grupo tiene las tareas de Product Owner, ScrumMaster.y el equipo de desarrollo; en el segundo grup estan las tareas de los usuarios, Stakeholders y managers.

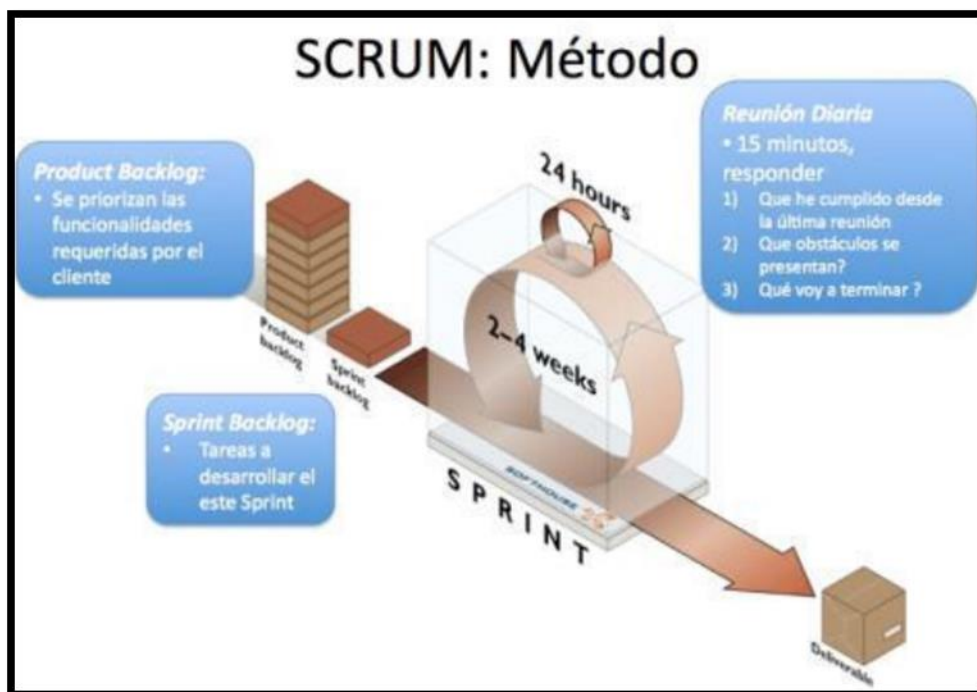
2.7.2. Elementos de Scrum.

Los elementos formales de Scrum para un proyecto de desarrollo, según Alaimo (2013) son:

- Product Backlog. El primero de los elementos, y principal de Scrum, es el Backlog del producto o también conocido como Pila del Producto o Product Backlog, es un listado de ítems (Producti Backlog Items, PBIs) o características del producto a construir, mantenido y priorizado por el Producto Owner. (pág. 33)
- Sprint Backlog. Es el conjunto de PBIs que fueron seleccionados para trabajar en ellos durante un cierto Sprint, conjuntamente con las tareas que el equipo de desarrollo ha identificado que debe realizar para poder crear un incremento funcional potencialmente entregable al finalizar el Sprint. (pág.403)
- Incremento funcional porque es una característica funcional nueva (o modificada) de un producto que está siendo construido de manera evolutiva. El producto crece con cada Sprint. (pág. 41)

Scrum cuenta con tres elementos el primero es “Product Backlog son la lista de necesidades del cliente; Sprint Backlog son la lista de tareas que se realizan en un Sprint; Incremento es la parte desarrollada en un Sprint, es una parte terminada y totalmente operativa” (Trigás Gallego, 2012, pág. 36), es decir que en los tres elementos de Scrum contiene los requerimientos del cliente más conocido como historias del usuario mismos que son seleccionados para el sprint, y el producto esperado por el cliente que debe ser de calidad, funcional y nuevo.

Figura 2.6. Ciclo de Desarrollo Scrum.

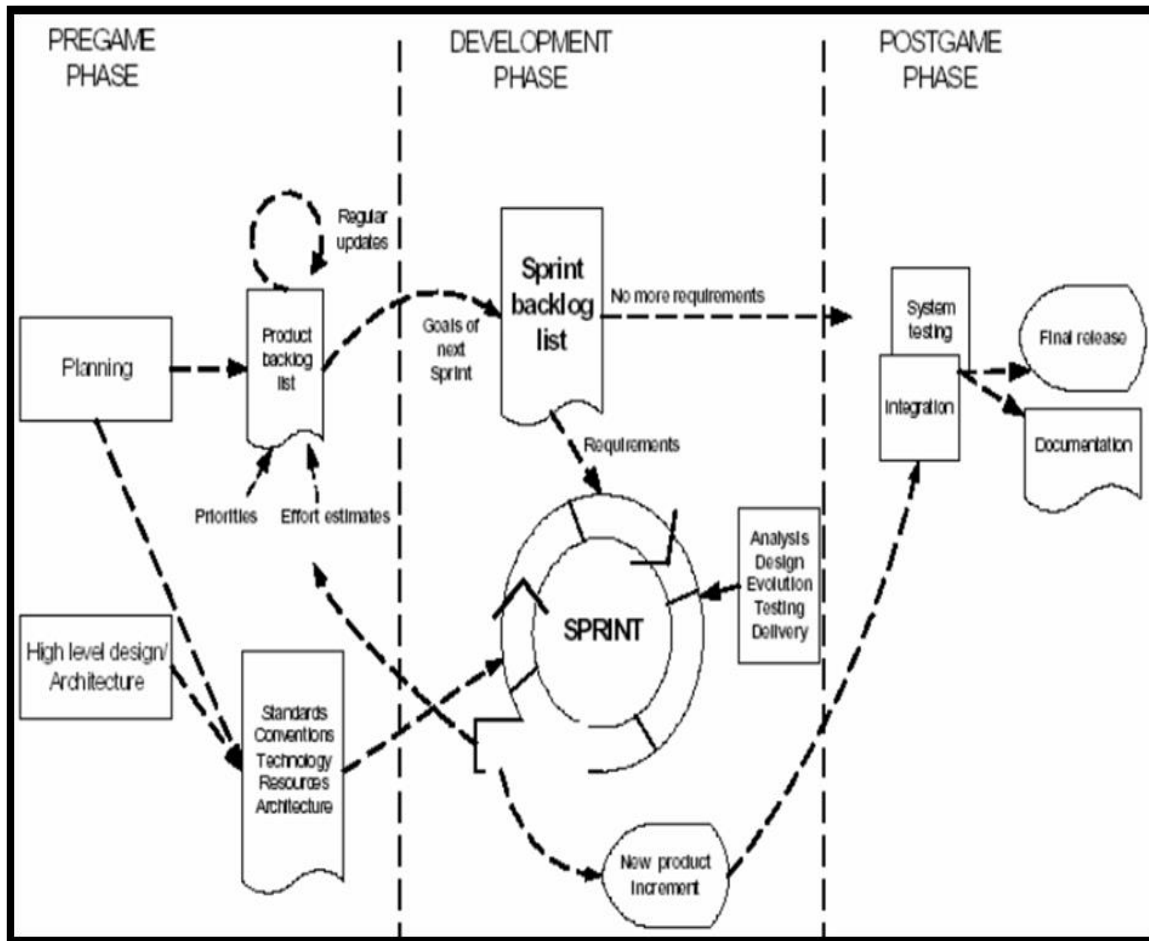


Fuente: Trigás Gallego (2012).

2.7.3. Fases de Scrum.

Las fases de Scrum consiste en el inicio y desarrollo del proyecto, antes del desarrollo, desarrollo y el cierre, a continuación, se presenta estas fases:

Figura 2.7. El proceso o las tres fases de Scrum: Pregame, Development y Postgame.



Fuente: Hernandez (2020).

2.7.3.1. PreGame.

De acuerdo a Peralta (2003) esta fase cuenta con dos subfases:

- **Planning.** Consiste en la definición del sistema que será construido. Para esto se crea la lista Product Backlog a partir del conocimiento que actualmente se tiene del sistema. La Product Backlog List es actualizada constantemente con ítems nuevos y más detallados, con estimaciones más precisas y cambios en la prioridad de los ítems.

- **Architecture / High level Design.** El diseño de alto nivel del sistema se planifica a partir de los elementos existentes en la Product Backlog List. En caso de que el producto a construir sea una mejora a un sistema ya existente, se identifican los cambios necesarios para implementar los elementos que aparecen en la lista Product Backlog. (pág. 6)

Esta fase esta dividida en el planeamiento que consiste en establecer la vision, el presupuesto, forma de financiamiento y el backlog del producto; y la arquitectura que consiste en la conceptualizacion y analisis. Si el proyecto se trata de la mejora de un nuevo sistema, solo se realiza un analisis limitado. Se realiza un diseño de alto nivel para actualizar los modelos del dominio y reflejar el contexto del nuevo sistema y los requerimientos y las modificaciones necesarias de la arquitectura del sistema. (Caso, 2004, pág. 3)

En resumen, la primera fase de Scrum consiste en la planificación, donde se establece y evalúa las herramientas de desarrollo, así como el análisis y diseño.

2.7.3.2. Game.

En esta fase se espera que ocurran cosas impredecibles. Para evitar el caos Scrum define prácticas para observar y controlar las variables técnicas y del entorno, así también como la metodología de desarrollo que hayan sido identificadas y puedan cambiar. Este control se realiza durante los Sprints. Dentro de variables de entorno encontramos: tiempo, calidad, requerimientos, recursos, tecnologías y herramientas de implementación. (Hernandez, 2020, pág. 7)

La segunda dase de Scrum es el desarrollo a partir de las interacciones más conocidas como el proceso de Sprints para el desarrollo del Software desde el requerimiento, análisis, diseño y desarrollo, hasta la entrega.

2.7.3.3. Post Game.

Contiene el cierre del release. Para ingresar a esta fase se debe llegar a un acuerdo respecto a las variables del entorno por ejemplo que los requerimientos fueron completados. El sistema está listo para ser liberado y es en esta etapa en la que se realiza integración, pruebas del sistema y documentación. (Hernandez, 2020, pág. 7)

La fase de post game, también es denominada la etapa de cierre es cuando “el equipo de management decide que las variables de entorno, tales como los requerimientos se han completado, se genera la documentación final, se realiza el testing pre-lanzamiento y el lanzamiento propiamente dicho” (Caso, 2004, pág. 4). Es decir que el sistema se encuentra listo para ser liberado para que posteriormente se realice la integración, pruebas y documentación.

2.8. HERRAMIENTAS.

2.8.1. PHP 7.

PHP es considerada como un lenguaje “se ejecuta del lado del servidor, el código PHP se incluye en una página HTML normal”, así también este lenguaje permite “realizar páginas web cuyo contenido puede generar total o parcialmente en el momento de la llamada página” (Heurtel, 2016, pág. 15).

En resumen, PHP 7 es un lenguaje de programación de la versión 7 de código abierto, orientado a objetivos, su importancia radica en el desarrollo web ajustado en HTML.

2.8.2. Framework Laravel.

Laravel es un marco de referencia de código abierto para el desarrollo de aplicaciones web en PHP 5, que posee una sintaxis simple, expresiva y elegante.

Fue creado en 2011 por Taylor Otwell, inspirándose en Ruby on Rails y Symfony, de los cuales ha adoptado sus principales ventajas. Laravel es un framework de código abierto para el desarrollo de aplicaciones web más usados de PHP, es también útil para crear código de manera sencilla. (Gallego Sánchez, s.f, pág. 5)

Tabla 2.2. Principales características y ventajas de Laravel.

Característica o ventaja	Descripción
El patrón MVC (modelo - vista - controlador)	Centrándose en la correcta separación y modularización del código. Lo que facilita el trabajo en equipo, así como la claridad, el mantenimiento y la reutilización del código
Integra un sistema ORM de mapeado	Permite la construcción de consultas directas a base de datos mediante su Consultor de construcción
Bases de datos	Permite la gestión y la manipulación de tablas desde código, manteniendo un control de versiones de las mismas mediante su sistema de Migraciones
sistema de plantillas para las vistas llamado Blade	Hace uso de la cache para darle mayor velocidad. Blade facilita la creación de vistas mediante el uso Delaware diseños , herencia y secciones
Extensión de funcionalidad mediante paquetes o librerías externas	es muy sencillo añadir paquetes que nos faciliten el desarrollo de una aplicación y nos ahorren mucho tiempo de programación
Línea de comandos llamados Artesano	Ayuda en varias tareas rutinarias como la creación de distintos componentes de código, trabajo con la base de datos y migraciones, gestión de rutas, cachés, colas, tareas programadas, etc.

Fuente: Gallego Sánchez (2012).

2.8.3. CSS.

CSS, por sus siglas en inglés Cascading Style Sheets, es el lenguaje de hojas de estilo de cascada “es un lenguaje que define los estilos, aspecto y formato de distintos elementos, generalmente, pertenecientes a un archivo HTML. Tal cual explica su nombre, la sintaxis de CSS se va organizando en forma de cascada” (Dias Cortez & Ocampo, 2016, pág. 1).

Esta herramienta sirve para organizar la presentación y aspecto de una página web, por lo que CSS determina instrucciones que definen los elementos HTML. CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. (Eguíluz Pérez, 2008, pág. 5)

2.8.4. MySQL.

MySQL es un sistema gestor de bases de datos. Es un sistema de libre distribución y de código abierto. Significa que se puede descargar libremente de Internet y que cualquier programador puede remodelar el código de la aplicación para mejorarlo. (Sánchez, 2003, pág. 2)

Es un sistema de gestión o administración de base de datos, es también un software libre de código abierto por la confiabilidad, velocidad y conectividad.

2.9. CALIDAD DE SOFTWARE ISO 9126.

De acuerdo con Largo y Marín (2005), en la Guía técnica para la evaluación de software, menciona que la norma Internacional ISO 9126 fue publicada en 1992, la cual ha sido y es usada para la evaluación de la calidad de software, llamado también “Information technology - Software product evaluation - Quality characteristics and guidelines for their use”. (Largo y Marín, 2005, pág. 9)

El estándar ISO-9126 establece que cualquier componente de la calidad del software puede ser descrito en términos de una o más de seis características básicas, las cuales son: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad; cada una de las cuales se detalla a través de un conjunto de subcaracterísticas que permiten profundizar en la evaluación de la calidad de productos de software. (Abud, 2012)

Tabla 2.3. Características ISO 9126.

Características	Pregunta central
Funcionalidad	¿Las funciones y propiedades satisfacen las necesidades explícitas e implícitas; esto es, el qué ?
Confiabilidad	¿Puede mantener el nivel de rendimiento, bajo ciertas condiciones y por cierto tiempo?
Usabilidad	¿El software es fácil de usar y de aprender?
Eficiencia	¿Es rápido y minimalista en cuanto al uso de recursos?
Mantenibilidad	¿Es fácil de modificar y verificar?
Portabilidad	¿Es fácil de transferir de un ambiente a otro?

Fuente: Abud, M. (2012).

Este estándar describe 6 características generales que son definidas a continuación:

Funcionalidad, en este grupo se conjunta una serie de atributos que permiten calificar si un producto de software maneja en forma adecuada el conjunto funciones que satisfagan las necesidades para las cuales fue diseñado (Abud, 2012). Para este propósito se establecen los siguientes atributos:

- **Adecuación.** Se enfoca a evaluar si el software cuenta con un conjunto de funciones apropiadas para efectuar las tareas que fueron especificadas en su definición.

- Exactitud. Este atributo permite evaluar si el software presenta resultados o efectos acordes a las necesidades para las cuales fue creado.
- Interoperabilidad. Permite evaluar la habilidad del software de interactuar con otros sistemas previamente especificados.
- Conformidad. Evalúa si el software se adhiere a estándares, convenciones o regulaciones en leyes y prescripciones similares.
- Seguridad. Se refiere a la habilidad de prevenir el acceso no autorizado, ya sea accidental o premeditado, a los programas y datos.

Confiabilidad, conjunto de atributos que se refieren a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido (Abud, 2012). Las subcaracterísticas que el estándar sugiere son:

- Nivel de Madurez. Permite medir la frecuencia de falla por errores en el software.
- Tolerancia a fallas. Se refiere a la habilidad de mantener un nivel específico de funcionamiento en caso de fallas del software o de cometer infracciones de su interfaz específica.
- Recuperación. Se refiere a la capacidad de restablecer el nivel de operación y recobrar los datos que hayan sido afectados directamente por una falla, así como al tiempo y el esfuerzo necesarios para lograrlo.

Usabilidad, consiste de un conjunto de atributos que permiten evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema. (Abud, 2012)

- Comprensibilidad. Se refiere al esfuerzo requerido por los usuarios para reconocer la estructura lógica del sistema y los conceptos relativos a la aplicación del software.
- Facilidad de Aprender. Establece atributos del software relativos al esfuerzo que los usuarios deben hacer para aprender a usar la aplicación.

- Operabilidad. Agrupa los conceptos que evalúan la operación y el control del sistema.

Eficiencia, esta característica permite evaluar la relación entre el nivel de funcionamiento del software y la cantidad de recursos usados. Los aspectos a evaluar son:

- Comportamiento con respecto al Tiempo. Atributos del software relativos a los tiempos de respuesta y de procesamiento de los datos.
- Comportamiento con respecto a Recursos. Atributos del software relativos a la cantidad de recursos usados y la duración de su uso en la realización de sus funciones.

Mantenibilidad, se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidad (Abud, 2012). En este caso, se tienen los siguientes factores:

- Capacidad de análisis. Relativo al esfuerzo necesario para diagnosticar las deficiencias o causas de fallas, o para identificar las partes que deberán ser modificadas.
- Capacidad de modificación. Mide el esfuerzo necesario para modificar aspectos del software, remover fallas o adaptar el software para que funcione en un ambiente diferente.
- Estabilidad. Permite evaluar los riesgos de efectos inesperados debidos a las modificaciones realizadas al software.
- Facilidad de Prueba. Se refiere al esfuerzo necesario para validar el software una vez que fue modificado.

Portabilidad, se refiere a la habilidad del software de ser transferido de un ambiente a otro, y considera los siguientes aspectos:

- Adaptabilidad. Evalúa la oportunidad para adaptar el software a diferentes ambientes sin necesidad de aplicarle modificaciones.
- Facilidad de Instalación. Es el esfuerzo necesario para instalar el software en un ambiente determinado.
- Conformidad. Permite evaluar si el software se adhiere a estándares o convenciones relativas a portatibilidad.
- Capacidad de reemplazo. Se refiere a la oportunidad y el esfuerzo usado en sustituir el software por otro producto con funciones similares.

2.10. MODELO DE ESTIMACIÓN DE COSTOS COCOMO II.

Según Un poco de java, en el artículo modelos de estimación menciona que COCOMO II: "... permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de coste y de escala. En los factores de coste se incluyen aspectos relacionados con la naturaleza del sistema, equipo, y características propias del proyecto. El factor de escala incluye la parte de escala producida a medida que un proyecto de software incrementa su tamaño". (Un poco de java, 2012)

COCOMO II posee tres modelos: Composición de Aplicación, Diseño Temprano y Post-Arquitectura.

Cada uno de estos modelos está orientado a sectores específicos del mercado de desarrollo de software y a las distintas etapas del desarrollo de software.

La sectorización de aplicaciones en COCOMO II es esta:

Tabla 2.4. Aplicaciones en COCOMO II.

Aplicaciones desarrolladas por usuarios finales		
Generadores de Aplicaciones	Aplicaciones con Componentes	Sistemas Integrados
Infraestructura		

Fuente: Un poco de java (2012).

- **Aplicaciones desarrolladas por Usuarios Finales:** En este sector se encuentran las aplicaciones generadas directamente por usuarios finales, mediante la utilización de generadores de aplicaciones tales como hojas de cálculo, sistemas de consultas, etc. Estas aplicaciones surgen debido al uso masivo de estas herramientas, conjuntamente con la presión actual para obtener soluciones rápidas y flexibles.
- **Generadores de Aplicaciones:** En este sector están los módulos pre-empaquetados que serán usados por usuarios finales y programadores.
- **Aplicaciones con Componentes:** Sector en el que se están aquellas aplicaciones que se resuelven por soluciones pre empaquetadas, pero son lo suficientemente simples para ser construidas a partir de componentes interoperables. Por ejemplo: interfaces gráficos, administradores de bases de datos, buscadores inteligentes de datos, ... Estas aplicaciones son generadas por un equipo reducido de personas, en pocas semanas o meses.
- **Sistemas Integrados:** Sistemas de gran escala, con un alto grado de integración entre sus componentes, sin antecedentes en el mercado que se puedan tomar como base. Partes de estos sistemas pueden ser desarrolladas a través de la composición de aplicaciones.
- **Infraestructura:** Área que comprende el desarrollo de sistemas operativos, protocolos de redes, sistemas administradores de bases de datos.

2.10.1. Estimación de Esfuerzo.

El esfuerzo necesario para concretar un proyecto de desarrollo de software, cualquiera sea el modelo empleado, se expresa en meses/persona (PM) y representa los meses de trabajo de una persona full-time, requeridos para desarrollar el proyecto. (Gómez et al, 2010, pág. 28)

2.10.1.1. Modelo de Composición de Aplicación.

La fórmula propuesta en este modelo es la siguiente:

$$PM = NOP / PROD$$

Donde:

NOP (Nuevos Puntos Objeto): Tamaño del nuevo software a desarrollar expresado en Puntos Objeto y se calcula de la siguiente manera:

$$NOP = OP \times (100 - \%reuso)/100$$

OP (Puntos Objeto): Tamaño del software a desarrollar expresado en Puntos Objeto
%reuso: Porcentaje de reuso que se espera lograr en el proyecto.

PROD: Es la productividad promedio determinada a partir del análisis de datos de proyectos en Banker 1994 (como se citó en Gómez, 2010).

Tabla 2.5. Productividad para el modelo Composición de Aplicación.

Capacidad de los desarrolladores	Muy Bajo	Bajo	Normal	Alto	Muy Alto
Madurez y Capacidad del ICASE	Muy Bajo	Bajo	Normal	Alto	Muy Alto
PROD	4	7	13	25	50

Fuente: Gomez (2010).

2.10.1.2. Modelo de Diseño Anticipado.

Este modelo se usa en las etapas tempranas de un proyecto de software, cuando se conoce muy poco del tamaño del producto a ser desarrollado, de la naturaleza de la plataforma, del personal a ser incorporado al proyecto o detalles específicos del proceso a utilizar. Este modelo podría emplearse tanto en productos desarrollados en sectores de Generadores de Aplicación, Sistemas Integrados o Infraestructura. (Gómez et al, 2010, pág. 28)

El modelo de Diseño Temprano ajusta el esfuerzo nominal usando siete factores de costo. La fórmula para el cálculo del esfuerzo es la siguiente:

$$PM_{estimado} = PM_{nominal} \times \prod_{i=1}^7 EM_i$$

$$PM_{nominal} = A \times (KSLOC)^B$$

$$B = 1.01 + 0.01 \times \sum_{j=1}^5 W_j$$

Donde:

- **PM Estimado** es el esfuerzo Nominal ajustado por 7 factores, que reflejan otros aspectos.
- propios del proyecto que afectan al esfuerzo necesario para la ejecución del mismo.
- **KSLOC** es el tamaño del software a desarrollar expresado en miles de líneas de código fuente.
- **A** es una constante que captura los efectos lineales sobre el esfuerzo de acuerdo a la variación del tamaño, (**A=2.94**).
- **B** es el factor exponencial de escala, toma en cuenta las características relacionadas con las economías y deseconomías de escala producidas cuando un proyecto de software incrementa su tamaño.
- **EMi** corresponde a los factores de costo que tienen un efecto multiplicativo sobre el esfuerzo, llamados Multiplicadores de Esfuerzo (Effort Multipliers). Cada factor se puede clasificar en seis niveles diferentes que expresan el impacto del multiplicador sobre el esfuerzo de desarrollo. Esta escala varía desde un nivel Extra Bajo hasta un nivel Extra Alto. Cada nivel tiene un peso asociado. El peso promedio o nominal es 1.0.

2.10.1.3. Modelo de Post Arquitectura.

Es el modelo de estimación más detallado y se aplica cuando la arquitectura del proyecto está completamente definida. Este modelo se aplica durante el desarrollo y mantenimiento de productos de software incluidos en las áreas de Sistemas Integrados, Infraestructura y Generadores de Aplicaciones. (Gómez et al, 2010, pág. 30)

El esfuerzo nominal se ajusta usando 17 factores multiplicadores de esfuerzo.

El mayor número de multiplicadores permite analizar con más exactitud el conocimiento disponible en las últimas etapas de desarrollo, ajustando el modelo

de tal forma que refleje fielmente el producto de software bajo desarrollo. La fórmula para el cálculo del esfuerzo es la siguiente:

$$PM_{estimado} = PM_{nominal} \times \prod_{i=1}^{17} EM_i$$

➤ Modelos de estimación

En la estimación del tamaño de software COCOMO II utiliza tres técnicas:

- Puntos Objeto.
- Puntos Función No Ajustados.
- Líneas de Código Fuente.

Además, se emplean otros parámetros relativos al tamaño que contemplan aspectos tales como: reuso, reingeniería, conversión y mantenimiento.

➤ Puntos objeto

El procedimiento para determinar Puntos Objeto en un proyecto software se resume en:

- **Determinar Cantidad de Objetos:** Estimar la cantidad de pantallas, reportes, componentes que contendrá la aplicación.
- **Clasificar cada instancia de un objeto según sus niveles de complejidad** (simple, media o difícil).
- **Dar el peso a cada objeto según el nivel de complejidad.** Los pesos reflejan el esfuerzo relativo requerido para implementar una instancia de ese nivel de complejidad.
- Determinar la cantidad de Puntos Objeto, sumando todos los pesos de las instancias de los tipos de objetos especificados.

➤ Puntos Función

El modelo COCOMO II usa Puntos Función y/o Líneas de Código Fuente (SLOC) como base para medir tamaño en los **modelos de estimación de Diseño Temprano y Post-Arquitectura**.

Los puntos función están basados en información disponible en las etapas tempranas del ciclo de vida del desarrollo de software.

COCOMO II considera solamente UFP (Puntos Función no ajustados).

$$FP = UFP \times TCF$$

Donde:

UFP: Puntos Función no Ajustados

TCF: Factor de Complejidad Técnica

Para calcular los UFP, se deben identificar los siguientes elementos:

- **Entradas Externas (Inputs):** Entrada de datos del usuario o de control que ingresan desde el exterior del sistema para agregar y/o cambiar datos a un archivo lógico interno.
- **Salidas Externas (Outputs):** Salida de datos de usuario o de control que deja el límite del sistema de software.
- **Archivo Lógicos Internos (Archivos):** Incluye cada archivo lógico, es decir cada grupo lógico de datos que es generado, usado, o mantenido por el sistema de software.
- **Archivos Externos de Interfase (Interfases):** Archivos transferidos o compartidos entre sistemas de software.

- **Solicitudes Externas (Queries):** Combinación única de entrada-salida, donde una entrada causa y genera una salida inmediata, como un tipo de solicitud externa.

Una vez identificados los elementos se clasifican de acuerdo al grado de complejidad en: bajo, promedio o alto. Se asigna un peso a cada ítem según el tipo y el grado de complejidad correspondiente. Finalmente, los UFP son calculados sumando los pesos de todos los ítems identificados.

➤ **Líneas de Código Fuente (SLOC)**

El objetivo es medir la cantidad de trabajo intelectual puesto en el desarrollo de un programa. Definir una línea de código es difícil debido a que existen diferencias conceptuales cuando se cuentan sentencias ejecutables y de declaraciones de datos en lenguajes diferentes.

➤ **Conversión de Puntos Función a Líneas de Código Fuente (SLOC)**

Para determinar el esfuerzo nominal en el modelo COCOMO II los puntos función no ajustados tienen que ser convertidos a líneas de código fuente considerando el lenguaje de implementación.

2.11. SEGURIDAD DEL SOFTWARE.

2.11.1. Algoritmo MD5.

Según el artículo Herramientas web para la enseñanza de protocolos de comunicación extraído de la página neo.lcc.uma.es, (s.f), menciona que el algoritmo de hash más utilizado en estos momentos es el MD5 (Algoritmo de Resumen del Mensaje 5). Este algoritmo fue diseñado por Ronald Rivest en 1995 y está basado en dos algoritmos anteriores MD2 y MD4. Todos estos protocolos producen un número de 128 bits a partir de un texto de cualquier longitud.

MD5 comienza rellorando el mensaje de una longitud en módulo $448 \bmod 512$. El relleno consiste en un bit 1 seguido por unos cuantos bits en 0 sean necesarios. La longitud original del mensaje es almacenada en los últimos 64 bits de relleno.

Adicionalmente se inicializa con un valor fijo, un buffer de 128 bits. Este buffer puede verse de 4 registros de 32 bits (ABCD) y son inicializados con los siguientes valores hexadecimales.

A= 67452301; B= EFC DAB89; C= 98BADCFE; D= 103225476

Durante varias rondas de procesamiento el algoritmo toma bloques de 512 bits de la entrada y lo mezcla con los 128 bits del buffer. Este proceso es repetido hasta que todos los bloques de entrada han sido consumidos. El valor resultante en el buffer es el hash del mensaje 512 bits del mensaje.

2.12. PRUEBAS DEL SOFTWARE.

Las pruebas de sistema se llevan a cabo cuando todo el desarrollo ha sido culminado y tenemos una versión preliminar del sistema que saldrá a producción. Esta etapa de pruebas consiste en probar un sistema integrado con el objeto de comprobar el cumplimiento de requisitos especificados. (Campos, 2015, pág. 13)

Las pruebas se hacen con un enfoque desde el punto de vista del usuario.

Las pruebas de sistema se desarrollan utilizando casos de prueba funcionales y no funcionales. Campos (2015) afirma: “Las pruebas funcionales confirman que los requisitos para un uso específico previsto han sido cumplidos”. (p. 13)

2.12.1. Pruebas de Caja Negra.

La técnica de caja negra, se enfoca en probar el sistema sin tomar en cuenta la estructura interna del mismo, su objetivo es validar que las salidas sean las esperadas. (Campos, 2015, pág. 23)

Se centra en encontrar las circunstancias en las que el sistema no se comporta conforme a las especificaciones establecidas.

Las técnicas más comunes de caja negra son:

- **Partición de equivalencia** (posibles valores divididos en clases, valores de entrada y valores de salida). Se agrupan todos los valores para los cuales se espera que el programa tenga un comportamiento común (rango de valores), y esa es una clase de equivalencia, existen clases de equivalencia válidas y clases de equivalencias inválidas.
- **Valores límite.** Complementan a la partición equivalente, se debe prestar mucha atención en que los límites deben estar correctamente definidos y programados. (Campos, 2015, pág. 24)
- **Transición de estado.** La transición de estados nos enuncia que todo sistema se mueve por transiciones de un paso a otro, nos podemos guiar por las transacciones válidas y las transiciones inválidas.
- **Tablas de decisión.** Estas pruebas consideran que para encontrar el resultado esperado deben estar en conjunto varias condiciones que son los detonadores del resultado, llamado como causa y efecto.

Ejemplo: Se tiene el siguiente formato de registro

Número De Empleado	Nombre	Meses Trabajando	Directivo
--------------------	--------	------------------	-----------

Dónde:

- Número de empleado es un campo de tres dígitos enteros positivos, sin incluir el 000.
- Nombre es un campo alfanumérico de máximo 30 caracteres.
- Meses trabajando es un campo que indica el número de meses que lleva el empleado en la empresa. Entero positivo de 3 dígitos, incluyendo el 000.
- Directivo es un campo de un solo carácter que puede ser “+” para indicar que el empleado es un directivo o “-“ para indicar que no lo es.

De acuerdo a lo anterior, se tiene la siguiente tabla:

Tabla 2.6. Condiciones para la prueba de Caja Negra.

Condición	Clases válidas	Clases inválidas
Número de empleado	-Número de tres dígitos, enteros positivos $000 > x \leq 999$	-Número menor a tres dígitos -Número mayor a tres dígitos -x = 000 -Números negativos -Números decimales -Caracteres no numéricos -Cadena nula
Meses trabajando	-Número de tres dígitos, enteros positivos $000 \geq x \leq 999$	-Número mayor a tres dígitos -Números negativos -Números decimales -Caracteres no numéricos -Cadena nula
Directivo	-Cadena de 1 carácter x = + x = -	-Cadena nula -Cadena de más de 1 carácter -Carácter diferente a “+” o “-“

Fuente: Campos (2015).

2.11.1.1 Análisis del Valor Límite.

Es el mayor número de errores que se presenta en los límites del dominio de entrada en el centro. Sánchez (2015), esta técnica de diseño de casos de prueba complementa la técnica partición de equivalencia u otras pruebas de caja negra. (Sánchez, 2015, pág. 41)

Los valores máximos y mínimos de una partición son sus valores límite. Las pruebas pueden diseñarse para cubrir tanto los valores límites válidos como no válidos. Las directrices para el análisis del valor límite son similares a las de partición equivalencia:

- Si una condición de entrada especifica un rango limitado por los valores a y b, los casos de prueba deben diseñarse con esos valores, además de los que se encuentran por encima y por debajo de ellos.
- Si una condición de entrada especifica diversos valores, deben desarrollarse casos de prueba que ejerciten los números máximo y mínimo. También se prueban los valores ubicados por encima y por debajo de estos máximos y mínimos.
- Aplicar directrices 1 y 2 a las condiciones de salida.
- Si la estructura interna de datos del programa tiene límites prescritos debe diseñarse un caso de prueba para ejercitar los límites de la estructura de datos.

CAPÍTULO III

MARCO APLICATIVO

3.1. INTRODUCCIÓN.

En este capítulo se realizará el análisis y diseño del sistema para ello se hará el uso de la metodología SCRUM, que requiere esfuerzo continuo para la adaptación y la evolución del proyecto. Al comenzar cada iteración (sprints) se determinará que partes se van a construir, tomando como criterios la prioridad para el desarrollo del sistema web y la cantidad de procesos que podrá abordar durante la iteración. Para el desarrollo de la metodología SCRUM se complementará con la metodología OOHDM, para las etapas de desarrollo de cada iteración que abarca el proyecto. Además, que se realizara las pruebas de software mediante la aplicación de la prueba de caja negra.

3.2. ANÁLISIS DE LA SITUACION ACTUAL.

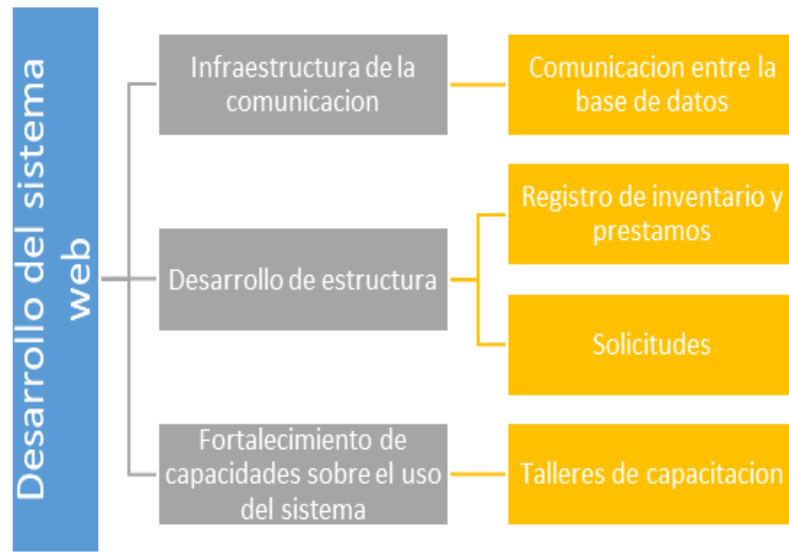
De acuerdo al diagnóstico realizado empleando la técnica de observación directa, la entrevista y la investigación documental, en la carrera de Electricidad industrial del Tecnológico Marcelo Quiroga Santa Cruz, actualmente existen falencias en el registro de inventario de equipamiento, registro estudiantil y docente para la solicitud de préstamos de equipos, instrumentos y herramientas de los laboratorios, taller de instalaciones eléctricas y taller electromecánico. Servicio de atención para préstamos de equipos, instrumentos y herramientas a los docentes y estudiantes con demora.

3.3. PRE GAME (antes del juego).

3.3.1. Planeación.

El presente proyecto de grado se constituye en una solución para mejorar y optimizar los procesos de la gestión de inventarios y préstamos. Para la planeación se llevaron a cabo varios requerimientos que fueron analizados desde distintos puntos de vista. Para ello se planifico las siguientes etapas de desarrollo.

Figura 3.1. Planeación del Desarrollo del Proyecto.



Fuente: Elaboración propia (2020)

El proyecto a desarrollar se constituye como una herramienta para el control y seguimiento de los procesos de inventario de equipamiento y el servicio de préstamos de los mismos, a continuación, se detallan dichos procesos:

- Los formularios de registro y control deben ser realizados con una interface amigable para el usuario y de fácil interacción.
- Contará con una bandeja de entrada para el inventario y solicitud de préstamo, juntos con los respectivos reportes.
- Contará con una ventana para el login de usuario, con la cual el usuario podrá ingresar al sistema de acuerdo con el permiso de acceso que tenga.

3.3.2. Análisis de Requerimientos (Product Backlog).

En este punto se planteará a grandes rasgos las herramientas, tecnologías y prácticas que se utilizarán en el proyecto, y se explorarán las posibilidades de la arquitectura del sistema.

Las herramientas de desarrollo son las siguientes:

- Lenguaje de Programación: PHP 5, JavaScript, Ajax, JQuery.
- MySQL como gestor de Base de Datos.
- Mozilla Firefox o Google Chrome como Navegador.
- Servidor Xampp.
- Windows 7 a Windows 10 como sistema operativo.

3.3.3. Identificación de Usuarios.

- **Usuario Administrador:** Es el que tiene todos los privilegios que puede proporcionar el sistema, el acceso a la información registrada es total, su rol en el sistema es de administrador y este es quien designa permisos para acceder al sistema.
- **Usuario personas:** Conformado por dos tipos de usuarios, docentes y estudiantes, ambos con acceso al sistema, como usuario normal y con restricciones solo para solicitar los préstamos.

3.3.4. Definición de Roles de los Actores.

A continuación, se describen a los actores y roles que representan con el que se interactúan con el sistema.

- **Personas:** representado por los estudiantes y docentes del ITMQSC, quienes son los que realizan la solicitud de préstamos de equipamiento para los laboratorios.
- **Scrum Master:** Estará representado por Univ. Jesús Tambo., quien será el único intercesor entre las personas y el equipo de trabajo.
- **Product Owner:** Estará representado por ya que es la persona encargado del almacén quien controla el inventario y préstamo de equipos.
- **Team:** Estará representado por Univ. Jesús Tambo. como único programador del equipo de trabajo.

3.3.5. Requerimientos de Usuario.

La metodología Scrum específica los requerimientos a través de historias de usuario, estos requerimientos servirán para la elaboración de diagramas de Interacción del Usuario, luego presiden la creación de las pruebas de aceptación una vez implementadas.

Desde el punto de vista del número de Historias de Usuario, se obtuvo inicialmente un total de doce tareas principales que serán necesarias para el inicio del desarrollo del sistema las cuales fueron de gran utilidad para la estimación de los tiempos requeridos para el desarrollo del proyecto.

Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración. A continuación, se describe cada historia de usuario como se detalla en las siguientes tablas:

Tabla 3.1. Historia de usuario - 1.

Historia de Usuario Número:	1	ITMQSC
Nombre Historia de Usuario: Diseño de Base de datos	Prioridad: alta	
Usuario: Ing. Elvis Rubén De Celis	PRIMERA ITERACION	
Historia: La Base de Datos que tendrá el nombre de inventario debe tener fácil acceso a la los del detalle de los equipos y para la solicitud de préstamos de los mismos equipos.		
Objetivo: Diseñar una base de datos que facilite el acceso a la información		
Desarrollador: Jesús Tambo Limachi		
Estimación: 1 semana		

Fuente: Elaboración propia (2020)

Tabla 3.2. Historia de usuario - 2.

Historia de Usuario Número:	2	ITMQSC
Nombre Historia de Usuario: Administración de Usuarios y personas	Prioridad: media	
Usuario: Ing. Elvis Rubén De Celis	SEGUNDA ITERACIÓN	
Historia: El registro usuarios es importante para asignar roles dar permisos de acceso al sistema tanto para el encargado de realizar el control de equipos y para los docentes y estudiantes que solicitan el préstamo de dichos equipos.		
Objetivo: Desarrollar un módulo de administración de usuarios donde se registre, actualice o elimine a los usuarios, se asigne un respectivo rol según la importancia con la que interactúe con el sistema permitir los accesos para iniciar sesión dentro del sistema. Además que se encuentra el registro de personas en este caso docentes y estudiantes.		
Desarrollador: Jesús Tambo Limachi		
Estimación: 1 semana		

Fuente: Elaboración propia (2020)

Tabla 3.3. Historia de usuario - 3.

Historia de Usuario Número:	3	ITMQSC
Nombre Historia de Usuario: Control de inventario	Prioridad: Alta	
Usuario: Ing. Elvis Rubén De Celis	TERCERA ITERACIÓN	
Historia: En la entrevista que se tuvo con el jefe de carrera se enfatizó sobre la automatización del inventario de equipos y herramientas dentro de la carrera de electrónica ya que todos los registros son manuales.		
Objetivo: realizar el módulo de inventario en cual se detalle la descripción de cada equipo, el laboratorio y el tipo de equipo.		
Desarrollador: Jesús Tambo Limachi		
Estimación: 1 semana		

Fuente: Elaboración propia (2020)

Tabla 3.4. Historia de usuario - 4.

Historia de Usuario Número:	4	ITMQSC
Nombre Historia de Usuario: Gestión de préstamos.	Prioridad: alta	
Usuario: Ing. Elvis Rubén De Celis	TERCERA ITERACION	
Historia: en la entrevista realizada con el jefe de carrera se determinó la automatización de los préstamos de equipos, ya que su control es moroso y poco fiable.		
Objetivo: realizar el módulo de préstamos donde se pueda hacer el seguimiento mediante el registro de solicitud de préstamos por parte del docente y del alumno.		
Desarrollador: Jesús Tambo Limachi		
Estimación: 1 seman		

Fuente: Elaboración propia (2020)

Tabla 3.5. Historia de usuario - 5.

Historia de Usuario Número:	5	ITMQSC
Nombre Historia de Usuario: Reportes	Prioridad: media	
Usuario: Ing. Elvis Rubén De Celis	CUARTA ITERACION	
Historia: los reportes son indispensables para realizar los seguimientos ya que no se cuenta con informes manuales.		
Objetivo: Desarrollar el módulo de reportes, donde se pueda visualizar informes de inventario y préstamos.		
Desarrollador: Jesús Tambo Limachi		
Estimación: 1 seman		

Fuente: Elaboración propia (2020)

A continuación, en la tabla 3.6, hacemos la recolección de los requerimientos de usuario que fueron extraídos de las historias de usuario.

Tabla 3.6. Requerimientos de usuario.

ID.	Descripción	Modulo	Prioridad	Estado
R1	Diseñar un base de datos para el desarrollo de procesos.	Bases de datos	Alta	Terminado
R2	Desarrollar la interface que permita el registro, modificación y eliminación datos de inventario.	Administración	Alta	Terminado
R3	Unificar las descripciones de los equipos, instrumentos en un sistema de información basado en la web.	Administración	Media	Terminado
R4	Desarrollar una interface de búsqueda con criterios múltiples para fechas o dependencias.	Búsqueda	Alta	Terminado
R5	Generar el formulario para el préstamo de equipamiento.	Solicitudes	Alta	Terminado
R6	Desarrollar una interface para el control y gestión de las solicitudes de préstamos por parte de los usuarios del sistema.	Solicitudes	Media	Terminado
R7	Diseñar una interface de acceso al sistema condicionado por el código de acceso mediante permisos y roles del sistema.	Usuario	Media	Terminado
R8	Desarrollar una interface que permita el registro, edición, eliminación de usuarios por parte del administrador del sistema.	Usuarios	Media	Terminado
R9	Adaptar la base de datos a la interfaz web.	Comunicación	Media	Terminado
R10	Elaboración del plan de capacitación.	Comunicación	Baja	Terminado
R11	Realización de taller de capacitación.	Administración	Baja	Terminado

Fuente: Elaboración propia (2020)

3.3.6. Planificación de la Entrega.

El cronograma de entrega se define en base al ciclo de la metodología SCRUM. En la **tabla 3.7**, se detalla todo el trabajo que se realizara:

Tabla 3.7. Cronograma de entrega.

CRONOGRAMA DE ENTREGA												
ID	Descripción	Duración en semanas	Septiembre			Octubre			Noviembre			
R1	Diseñar la base de datos que facilite el acceso al inventario de equipamiento.	1	■									
R2	Desarrollar la interface que permita el registro, edición y eliminación de los datos del inventario de equipamiento, resguardados en el sistema.	1			■							
R3	Unificar las descripciones de los equipos, instrumentos en un sistema de información basado en la web.	1				■						
R4	Desarrollar una interface de búsqueda con criterios múltiples para fechas o dependencias.	2				■	■					
R5	Generar el formulario para el préstamo de equipamiento.	1						■				
R6	Desarrollar una interface para el control y gestión de las solicitudes de préstamos por parte de los usuarios del sistema.	2						■	■			
R7	Diseñar una interface de acceso al sistema condicionado por el código de acceso mediante permisos y roles del sistema.	1		■								
R8	Desarrollar una interface que permita el registro, edición, eliminación de usuarios por parte del administrador del sistema. También el registro de docente y estudiante.	1			■							
R9	Adaptar la base de datos a la interfaz web.	2								■	■	
R10	Elaboración del plan de capacitación.	1										■
R11	Realización de taller de capacitación.	1										■

Fuente: Elaboración propia (2020)

3.3.6.1. Análisis de Riesgo.

Para realizar este análisis es necesario contemplar 2 tipos de riesgos: uno es riesgo del proyecto y otro es riesgo del producto. En la siguiente tabla se describen todos los puntos analizados.

Tabla 3.8. Análisis de riesgo

Riesgo	Tipo	Descripción	Probabilidad	Estado	Estrategia
Cuando no se cumplen la fechas establecidas en el cronograma.	Proyecto	Es muy probable que la fechas descritas en el cronograma no se cumplan en dichos plazos	Alta	Tolerable	Realizar otro cronograma más flexible.
Cambio en los requerimientos del cliente	Producto	Riesgo que se de cambios en los requerimientos con respecto a los módulos.	Moderado	Tolerable	Realizar una revisión constante a los requerimientos
No se cuenta con la infraestructura estable para la implementación del sistema	Proyecto	Riesgo de que la institución tenga problemas con alguno de los requerimientos para la implementación del sistema	Moderado	Tolerable	Solicitar con anticipación o contar con un equipo portátil

Fuente: Elaboración propia (2020)

3.4. GAME (Juego).

3.4.1. Iteraciones (Sprints).

En este punto se realiza varias iteraciones sobre el sistema antes de ser entregado. Cada iteración está conformada por tablas de requerimientos conocidas como product backlog correspondientes a cada historia de usuario descrita en la fase de pre game.

Las iteraciones se desarrollarán combinando las fases de diseño navegacional, diseño conceptual y el diseño de interfaz abstracta propuesta por la metodología OOHDM.

3.4.1.1. Primera Iteración – Módulo Usuarios.

Durante la primera iteración se desarrolló el diseño de la base de datos que contara con una emulación de una estructura jerárquica para su funcionalidad respecto al cuadro de clasificación, como se ve en la tabla 15, que representa a la primera iteración.

Tabla 3.9. Primera Iteración.

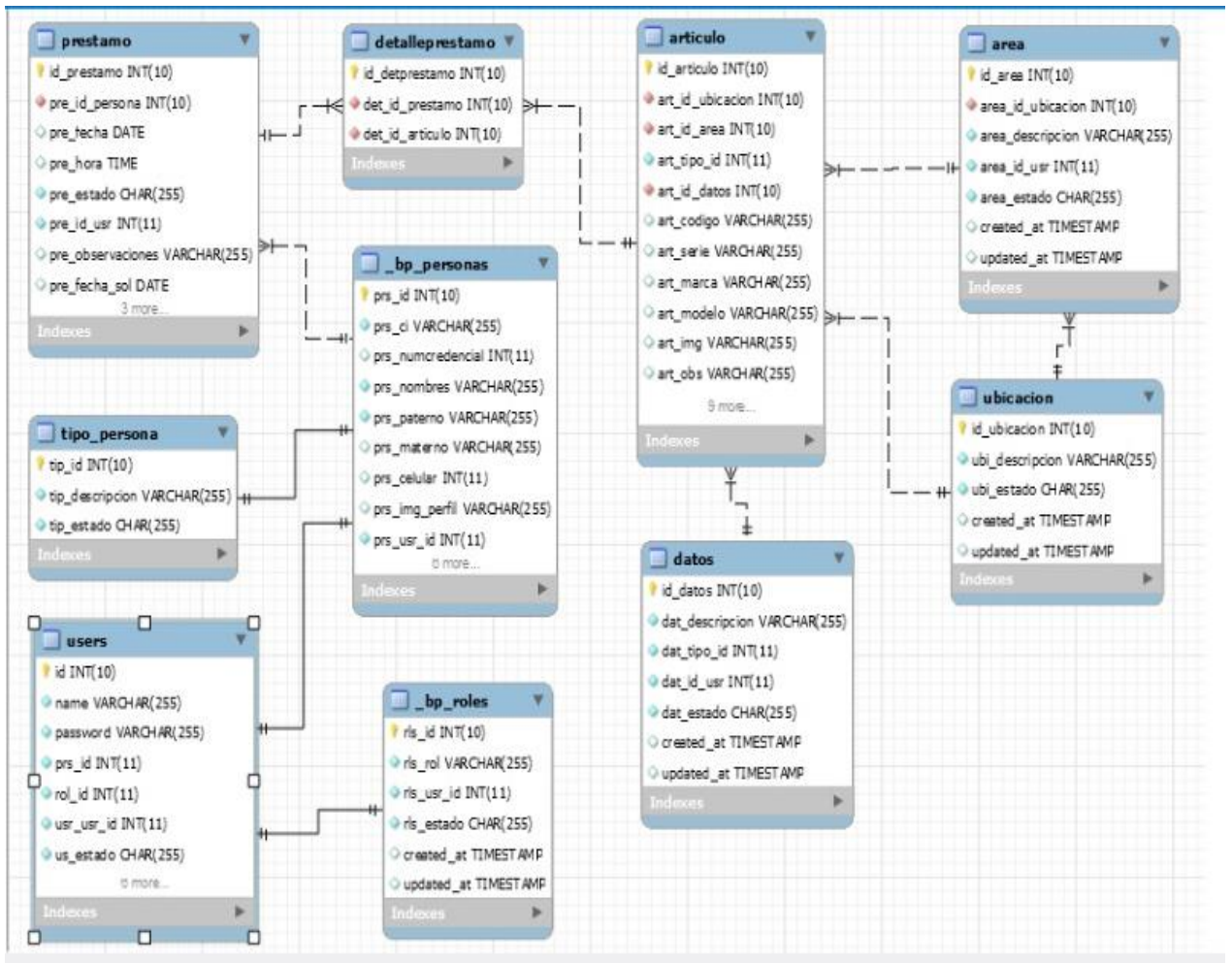
CICLO DE TRABAJO		SPRINT	INICIO	DURACIÓN
		1	03/09/2020	13 días
ID		Tipo	Días de trabajo	Estado
1.	Realizar la planificación de la iteración.	Planificación	3	Terminado
2.	Analizar los requerimientos de backlog del producto.	Planificación	2	Terminado
3.	Análisis de riesgos de backlog del producto.	Planificación	1	Terminado
4.	Diseño de la base de datos.	Modelado	3	Terminado
5.	Construcción de diseño navegacional.	Modelado	1	Terminado
6.	Construcción del diseño conceptual.	Modelado	1	Terminado
7.	Construcción del diseño de interfaz abstracta.	Modelado	1	Terminado
8.	Desarrollar la interfaz de ingreso al sistema	Desarrollo	1	Terminado
9.	Creación del módulo de registro de usuario	Desarrollo	1	Terminado

Fuente: Elaboración propia (2020)

3.4.1.1.1. Diseño de la Base de Datos.

El diseño de la base de datos se la realiza analizando los requerimientos de usuario, es así que después de hacer un análisis exhaustivo se tiene la figura 16, que representa a la base de datos del sistema.

Figura 3.2. Diseño de la base de datos del sistema.



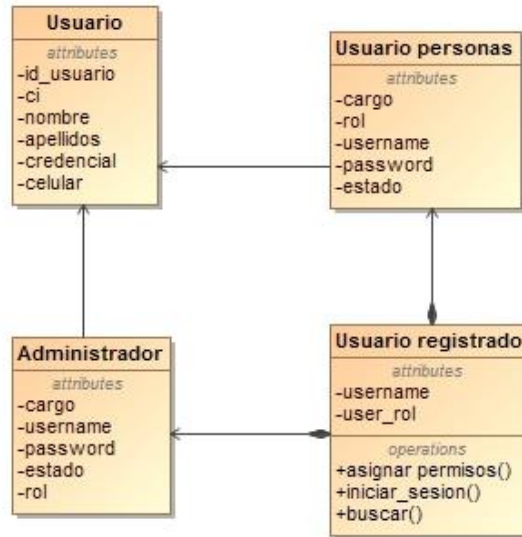
Fuente: Elaboración propia (2020)

Luego de haber diseñado la base de datos se procede con el diseño Conceptual, diseño Navegacional y Diseño de Interfaz Abstracta.

3.4.1.1.2. Diseño Conceptual – Módulo Usuarios.

El diseño conceptual de la figura representa la etapa de la metodología OOHDM en la cual se elabora a partir de los requerimientos obtenidos del sistema.

Figura 3.3. Diseño Conceptual - Módulo de Usuario.

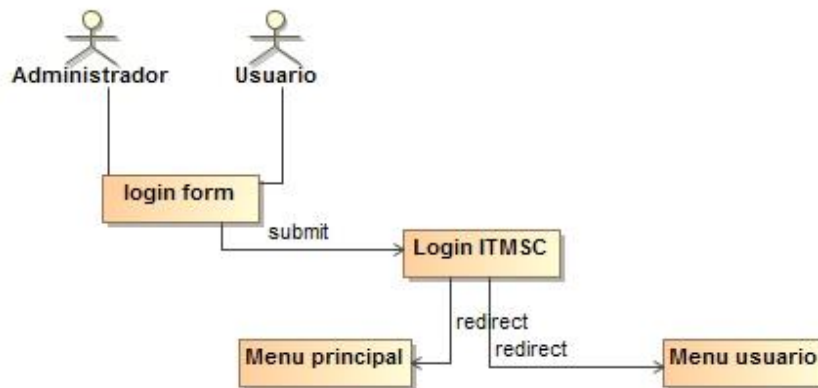


Fuente: Elaboración propia (2020)

3.4.1.1.3. Diseño Navegacional – Módulo Usuarios.

En esta etapa se define clases navegacionales tales como nodos, enlaces y estructuras de acceso inducidas del esquema conceptual, seguidamente se realiza el diagrama de contextos navegacionales representadas en la Figura 3.4.

Figura 3.4. Diseño navegacional - Módulo de Usuarios.



Fuente: Elaboración propia (2020)

3.4.1.1.4. Diseño Interfaz Abstracta – Módulo Usuarios.

Una vez que las estructuras navegacionales son definidas, se pasa a la siguiente etapa de la metodología OOHDM donde se especifica los objetos navegacionales y cuál de estos activaran la navegación, como se ve en la figura 3.5.

Figura 3.5. Diseño Abstracto - Módulo Usuarios.

ADV INGRESO

Usuario:

Contraseña:

Fuente: Elaboración propia (2020)

Figura 3.6. Diseño Abstracto - Registro de Usuario Persona.

ADV REGISTRO PERSONA (DOCENTE, ESTUDIANTE)

ADV DATOS

NOMBRE:

CI. MATRICULA:

TIPO_PERSONA:

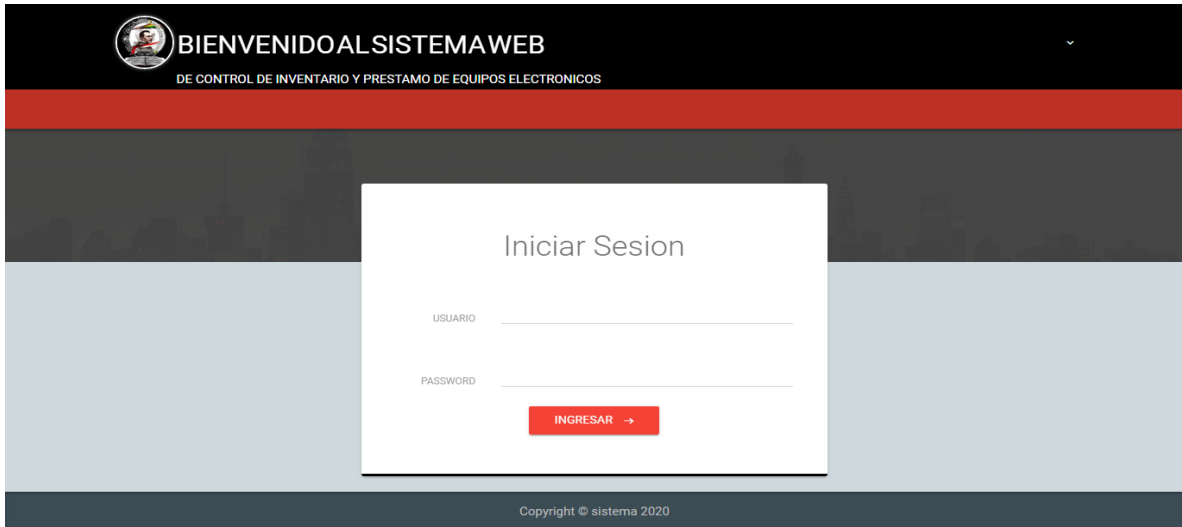
CORREO: CELULAR:

Fuente: Elaboración propia (2020)

3.4.1.1.5. Implementación - Módulo De Usuarios.

La Figura 3.7, muestra la interfaz de inicio de sesión.

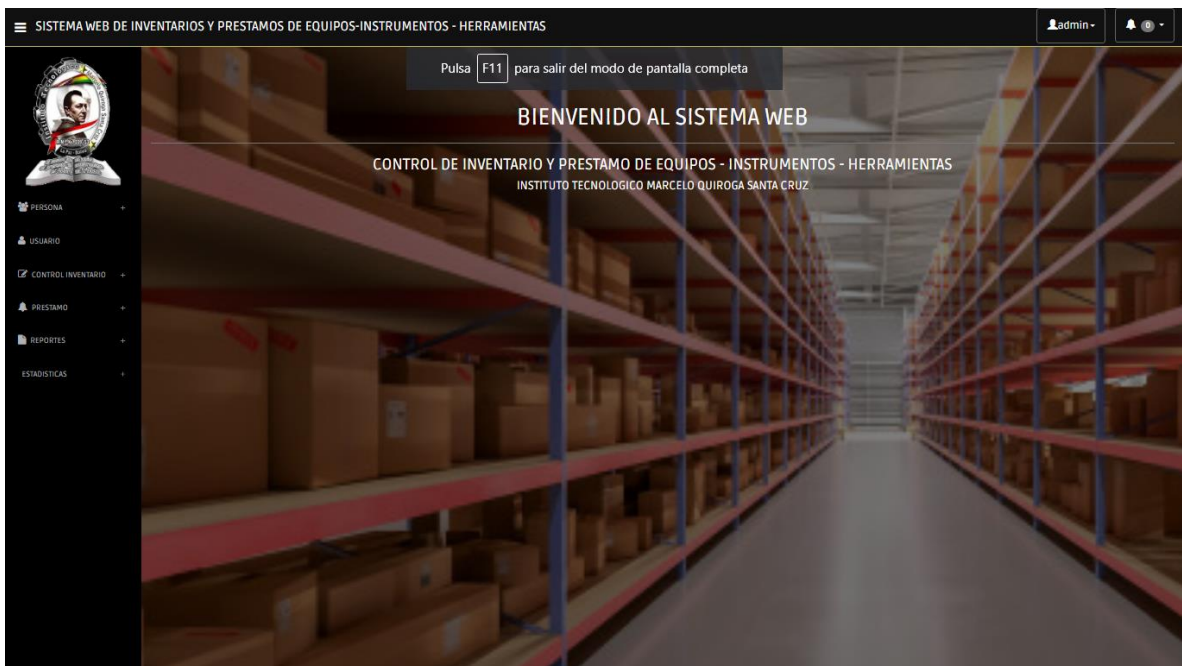
Figura3.7. Interfaz - Inicio de sesión.



Fuente: Elaboración propia (2020)

En la siguiente figura se puede observar la interfaz de bienvenida al sistema después de haberse logueado.

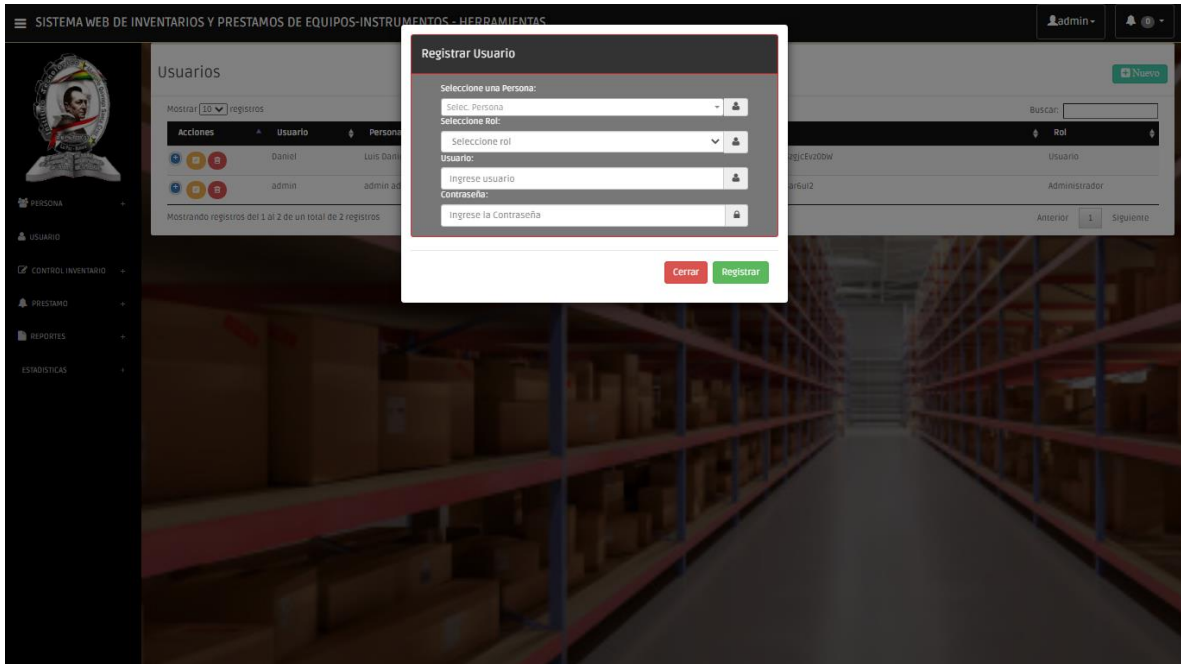
Figura 3.8. Interfaz - Vista Home.



Fuente: Elaboración propia (2020)

En la figura 3.9, se muestra el formulario de registro de usuario con los respectivos campos para llenar más el nombre de usuario y contraseña, además de la asignación del rol de usuario por parte del admin.

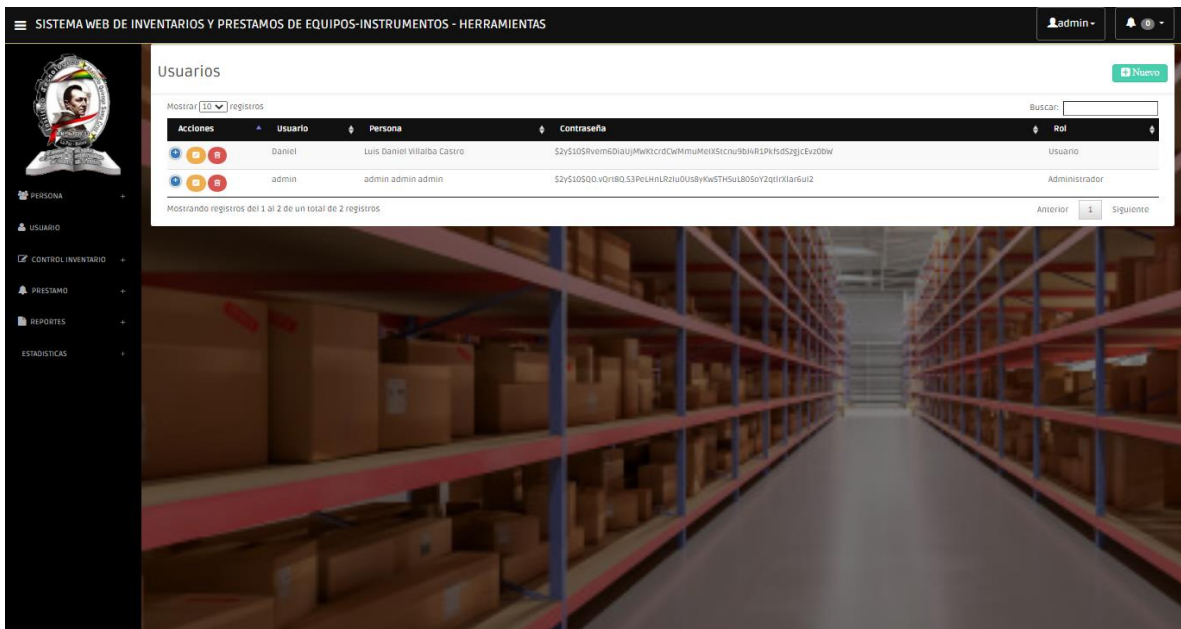
Figura 3.9. Interfaz - Registro Usuario y Asignación de rol.



Fuente: Elaboración propia (2020)

La figura 3.10. Muestra el listado de usuario en la vista principal.

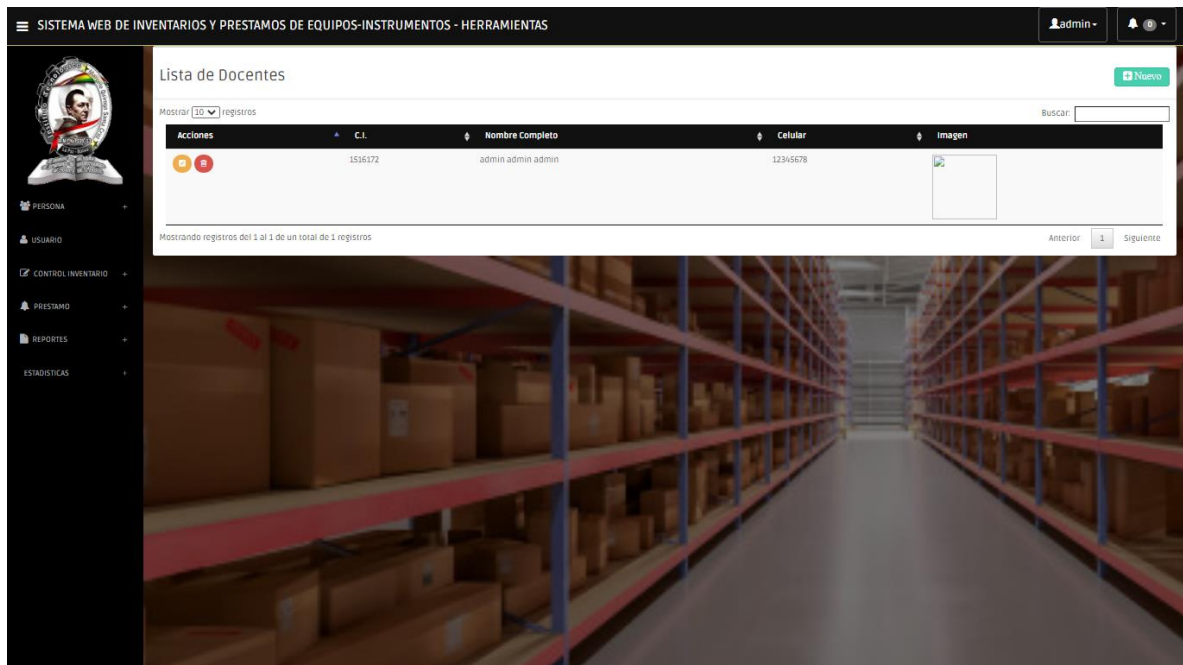
Figura 3.10. Interfaz - Listado Usuario.



Fuente: Elaboración propia (2020)

En la figura 3.11 se muestra la interfaz del listado de docentes.

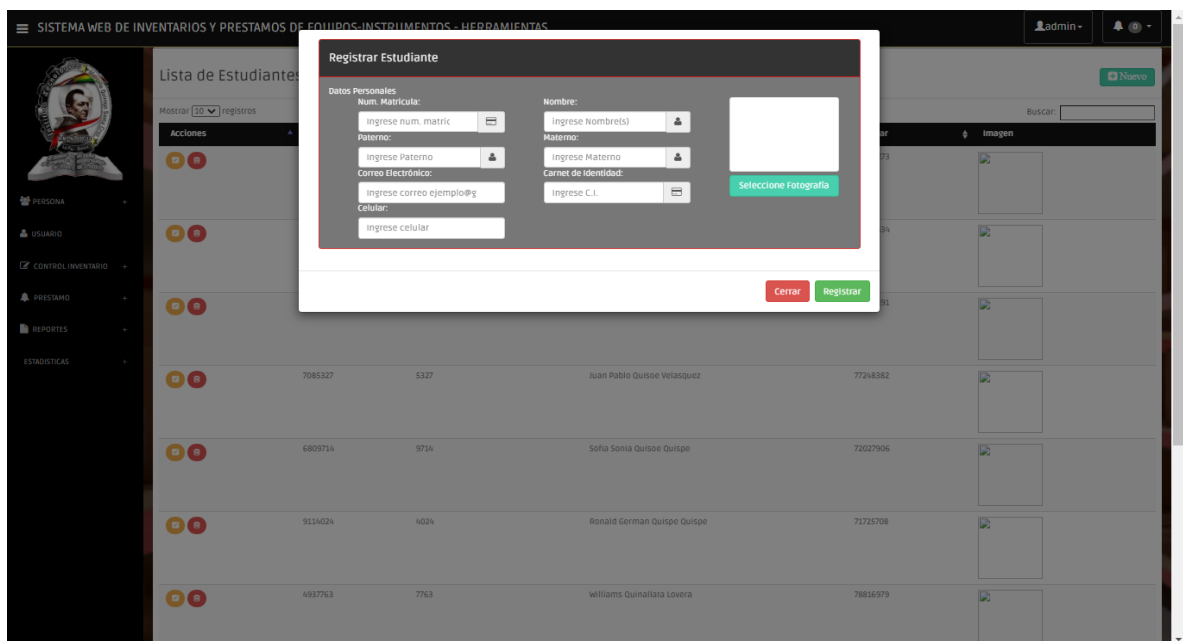
Figura 3.11. Interfaz - Roles listado de Docentes.



Fuente: Elaboración propia (2020)

En la figura 3.12, se muestra el formulario de registro de estudiante con los campos que permitan la el llenado y adición de fotografía del estudiante.
















Figura 3.12. Interfaz - Registro de Persona.



Fuente: Elaboración propia (2020)

La figura 3.13. muestra el listado de estudiantes del sistema.

Figura 3.13. Interfaz - Listado Estudiantes.

Acciones	C.I.	N° matricula	Nombre Completo	Celular	Imagen
 	4810892	938	Carlos Andres Torrez Colmena	72265454	
 	4855033	941	Miguel Angel Rodriguez Calama	78020055	
 	4931431	940	Benigno Gregorio Gutierrez Chura	66243412	
 	4744541	939	José Luis Julián Tito	60054322	
 	4810892	938	Carlos Andres Torrez Colmena	72265454	

Fuente: Elaboración propia (2020)

3.4.1.2. Segunda Iteración - Módulo de Inventario.

En la segunda iteración se continuo con los diseños OOADM, además que se realizó el módulo de control de inventario y el registro de bajas de equipos.

Tabla 3.10. Segunda iteración.

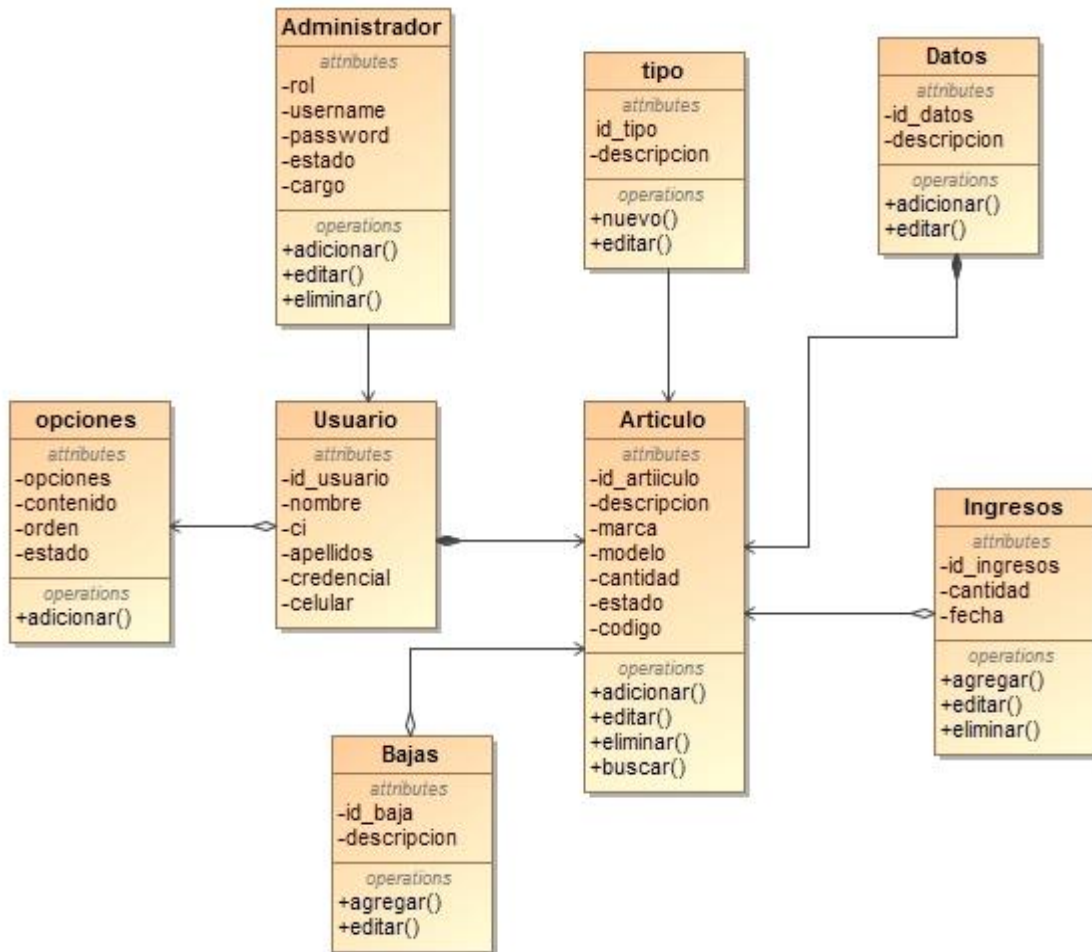
CICLO DE TRABAJO		SPRINT	INICIO	DURACIÓN
		2	20/09/2020	12 días
ID		Tipo	Días de trabajo	Estado
1.	Realizar la planificación de la iteración.	Planificación	1	Terminado
2.	Analizar los requerimientos de backlog del producto.	Planificación	1	Terminado
3.	Análisis de riesgos de backlog del producto.	Planificación	1	Terminado
4.	Complementar el diseño navegacional.	Modelado	1	Terminado
5.	Complementar diseño conceptual.	Modelado	1	Terminado
6.	Complementar diseño de interfaz abstracta.	Modelado	1	Terminado
7.	Desarrollar el módulo de registro de inventario	Desarrollo	3	Terminado
8.	Desarrollar el registro de baja de equipos.	Desarrollo	2	Terminado

Fuente: Elaboración propia (2020)

3.4.1.2.1. Diseño Conceptual - Módulo de Inventario.

El diseño conceptual es la etapa de la metodología OOHDM en la cual se elabora a partir de los requerimientos obtenidos del sistema. La siguiente figura 3.14, representa el diseño conceptual del módulo de administración que contendrá el sistema.

Figura 3.14. Diseño conceptual - Módulo Inventario.

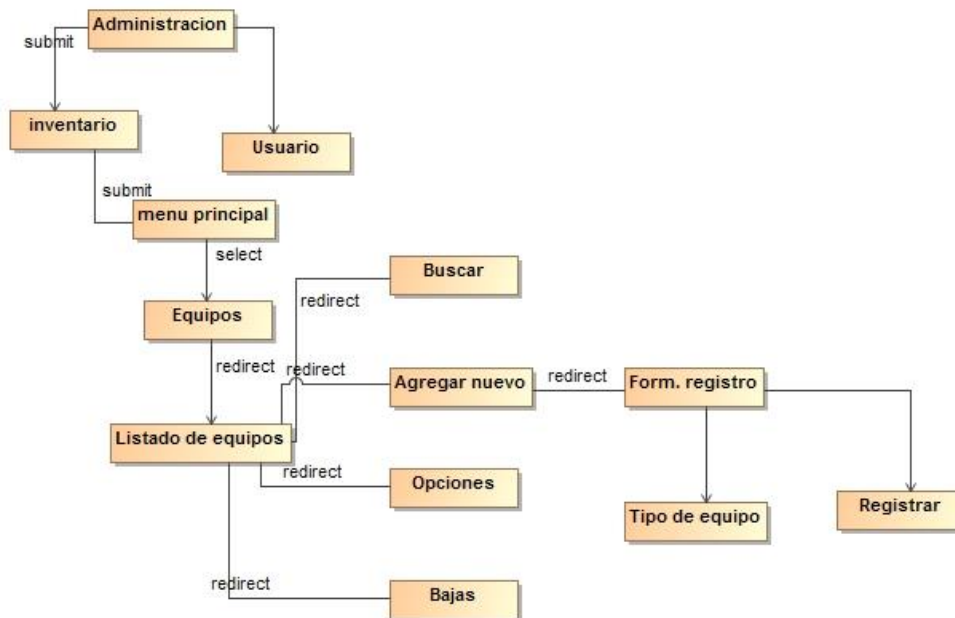


Fuente: Elaboración propia (2020)

3.4.1.2.2. Diseño Navegacional – Módulo de Inventario.

El diseño navegacional del módulo de inventario constituye en el registro, listado, búsquedas, tipo y bajas de equipos.

Figura 3.15. Diseño Navegacional - Módulo de Inventario.



Fuente: Elaboración propia (2020)

3.4.1.2.3. Diseño de Interfaces Abstractas – Módulo de Inventario.

El diseño de interfaz abstracta da a conocer el modelo como se desarrolla el sistema.

Figura 3.16. Diseño Abstracta - Vista Inventario.



Fuente: Elaboración propia (2020)

Figura 3.17. Diseño Abstracto - Registro de Inventario.

ADV REGISTRO DE EQUIPO

CODIGO	<input type="text"/>
UBICACION	1
AREA	1
EQUIPO	1
MARCA	<input type="text"/>
MODELO	<input type="text"/>
<input type="button" value="REGISTRAR"/>	

ADV TIPO DE EQUIPO

NOBRE DE EQUIPO

ESTADO

Fuente: Elaboración propia (2020)

3.4.1.2.4. Implementación – Módulo Inventario.

En la figura 3.18, se muestra la interfaz del formulario de registro de equipo con todos los campos para seleccionar y llenar el registro de un nuevo equipo.

Figura 3.18. Interfaz - Registro de Equipo.

SISTEMA WEB DE INVENTARIOS Y PRESTAMOS DE EQUIPOS-INSTRUMENTOS - HERRAMIENTAS

Registro de Equipo

Codigo: ITM05C-0000 Serie: serie

Ubicacion: Seleccione ubicacion

Area: Seleccione Area

Tipo: Seleccione Tipo

Nuevo Equipo...

Equipo: Seleccione Equipo

Marca: Modelo:

Costo De Equipo: costo 1111.00 Fecha Compra: 14/12/2020

Años De Garantia: garantia

Depreciación imagen Estado

Español (España)
Teclado Español

Para cambiar entre métodos de entrada, presiona la tecla Windows+Espacio.

Fuente: Elaboración propia (2020)

La figura 3.19. muestra el listado de equipos ya registrados.

Figura 3.19. Interfaz - Listado de Equipos.

The screenshot shows a web interface for equipment management. At the top, there is a header with the system name 'SISTEMA WEB DE INVENTARIOS Y PRESTAMOS DE EQUIPOS-INSTRUMENTOS - HERRAMIENTAS' and a user profile 'Admin'. Below the header, there are search filters for 'BUSQUEDA POR CODIGO', 'BUSQUEDA POR SERIE', and 'BUSQUEDA POR DESCRIPCION'. A 'Mostrar 10 registros' dropdown is visible. The main content is a table with the following columns: Opciones, Código, Serie, Descripción, Ubicación, Area, Tipo, Fecha Compra, Precio Bs., Año Garantía, Depreciación, Imagen, and Estado. The table contains five rows of equipment data, each with a 'MODIFICAR' and 'BAJA' button.

Opciones	Código	Serie	Descripción	Ubicación	Area	Tipo	Fecha Compra	Precio Bs.	Año Garantía	Depreciación	Imagen	Estado
MODIFICAR BAJA	ITMQSC-00005	LOG4321	PLC SIEMENS 1200M	LABORATORIO 2	LAB DE CONTROLES LOGICOS PROGRAMABLES	EQUIPO	2020-12-11	700	1	2021-12-11		ALTA
MODIFICAR BAJA	ITMQSC-00004	m678	GENERADORES DE FUNCIONES EW44 GAN1290	LABORATORIO 1	LAB DE CIRCUITOS ELECTRICOS I	EQUIPO	2020-12-11	1300	1	2021-12-11		
MODIFICAR BAJA	ITMQSC-00003	CT1203-67	CABLE PULLER CAMSCO 12MM	LABORATORIO 2	LAB DE AUTOMATISMOS ELECTRICOS II	HERRAMIENTAS	2020-12-11	1.9	1	2021-12-11		BAJA
MODIFICAR BAJA	ITMQSC-00002	12434	AUTOTRANSFORMADORES MONOFASICOS GEMRO 12MM	LABORATORIO 1	LAB DE CIRCUITOS ELECTRICOS I	EQUIPO	2020-12-11	3456	1	2021-12-11		
MODIFICAR	ITMQSC-		AMPERIMETRO ANALOGICO CA	LABORATORIO	LAB DE CIRCUITOS INSTRUMENTOS							

Fuente: Elaboración propia (2020)

En la figura 3.20. se muestra el formulario para dar de baja a un equipo.

Figura 3.20. Interfaz - Baja de Equipo.

The screenshot shows the 'BAJA DE EQUIPO' form overlaid on the equipment list. The form contains the following fields: CODIGO (ITMQSC-00005), SERIE (LOG4321), UBICACION (LABORATORIO 2), EQUIPO (PLC), AREA (LAB DE CONTROLES LOGICOS PROGRAMABLES), TIPO (EQUIPO), MARCA (SIEMENS), MODELO (1200M), FECHA BAJA (2020-12-14), and RAZON DE LA BAJA (empty text area). At the bottom right of the form, there are two buttons: 'Cerrar' and 'DAR BAJA'.

Fuente: Elaboración propia (2020)

3.4.1.3. Tercera Iteración – Módulo de Préstamos.

En esta iteración se hará la implementación del módulo de seguimiento de préstamos mediante el desarrollo de registro de solicitudes para dicho préstamo, aplicando los diseños OOHDM.

Tabla 3.11. Tercera Iteración.

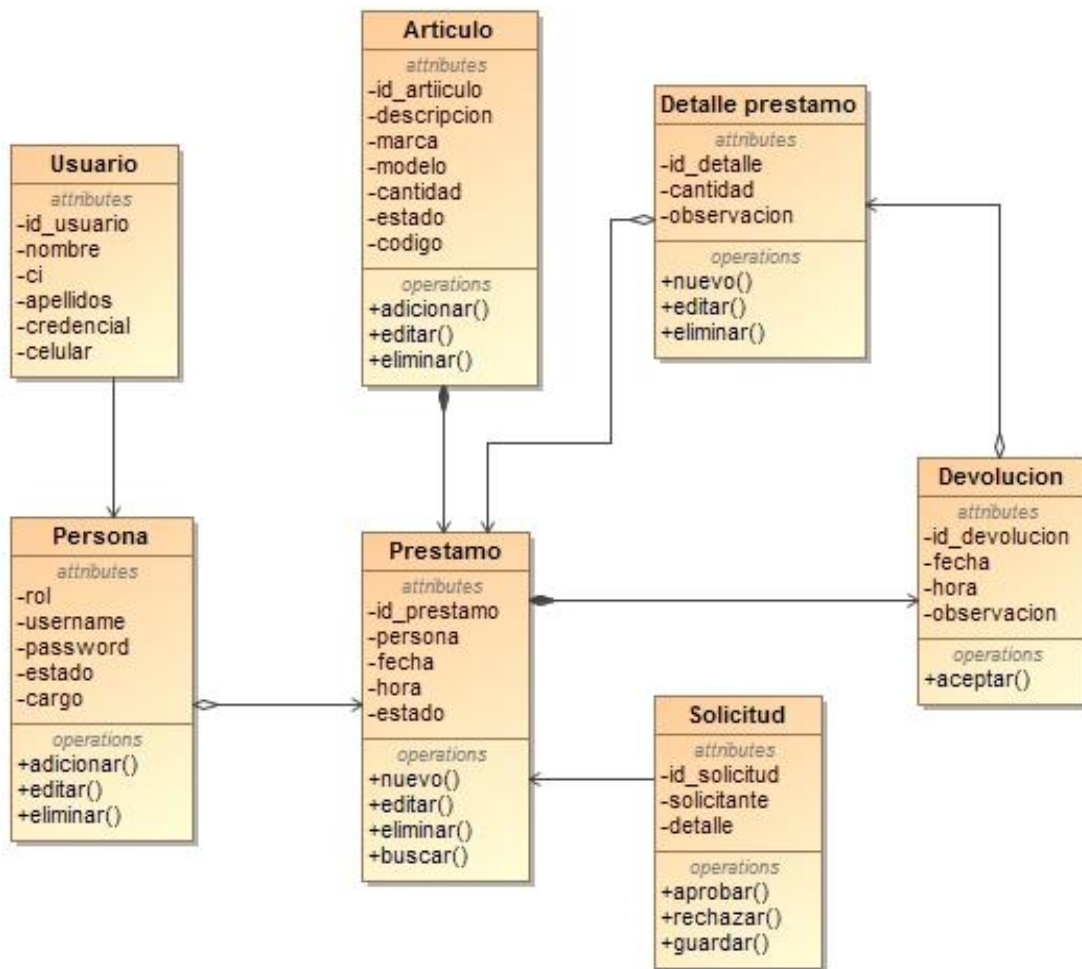
CICLO DE TRABAJO		SPRINT	INICIO	DURACIÓN
		3	05/10/2020	15 días
ID		Tipo	Días de trabajo	Estado
1.	Realizar la planificación de la iteración.	Planificación	1	Terminado
2.	Analizar los requerimientos de backlog del producto.	Planificación	1	Terminado
3.	Análisis de riesgos de backlog del producto.	Planificación	1	Terminado
4.	Complementar el diseño navegacional.	Modelado	1	Terminado
5.	Complementar diseño conceptual.	Modelado	2	Terminado
6.	Complementar diseño de interfaz abstracta.	Modelado	2	Terminado
7.	Desarrollar el módulo de registro de préstamos.	Desarrollo	3	Terminado
8.	Desarrollar las solicitudes y registro de devoluciones	Desarrollo	3	Terminado

Fuente: Elaboración propia (2020)

3.4.1.3.1. Diseño Conceptual – Módulo de Préstamos.

El diseño conceptual del módulo de préstamos refleja como el detalle de préstamos y solicitudes.

Figura 3.21. Diseño Conceptual - Módulo de Préstamos.

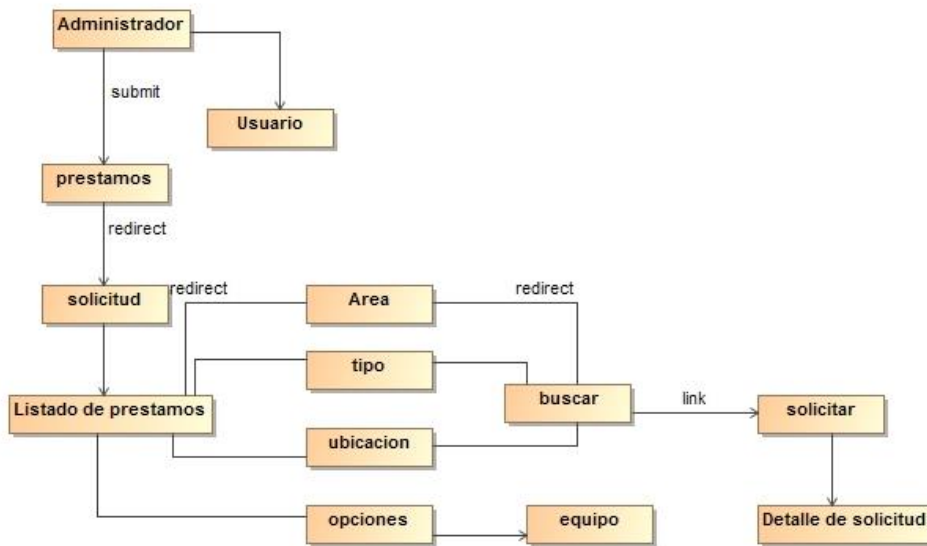


Fuente: Elaboración propia (2020)

3.4.1.3.2. Diseño Navegacional - Módulo de Préstamos.

En el diseño navegacional del módulo de préstamos se desarrolla en base a la solicitud con las características de área, tipo, ubicación y búsqueda.

Figura 3.22. Diseño Navegacional - Módulo de Préstamos

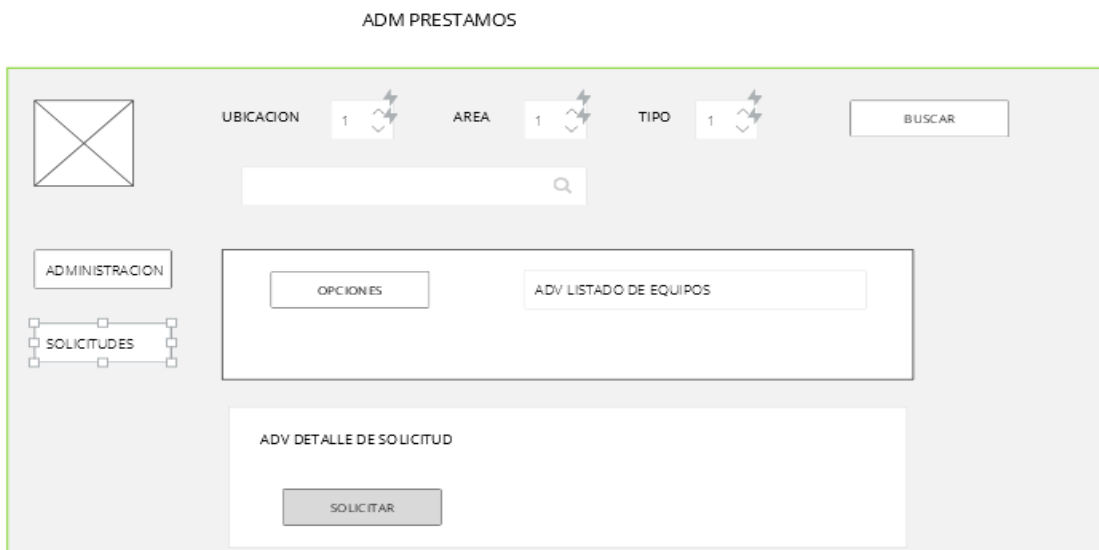


Fuente: Elaboración propia (2020)

3.4.1.3.3. Diseño de Interfaces Abstractas – Módulo de Préstamos.

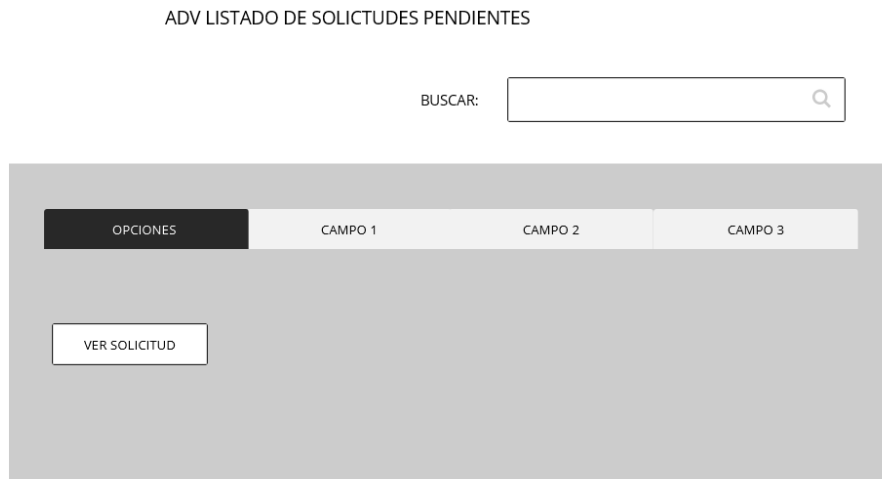
Para el diseño de interfaz de abstractas del módulo de préstamos se diseñó la opción para solicitar un equipo y el detalle del mismo, además del listado pendientes.

Figura 3.23. Diseño Abstracto - Módulo de Préstamos.



Fuente: Elaboración propia (2020)

Figura 3.24. Diseño abstracto - Listado de solicitudes pendientes.

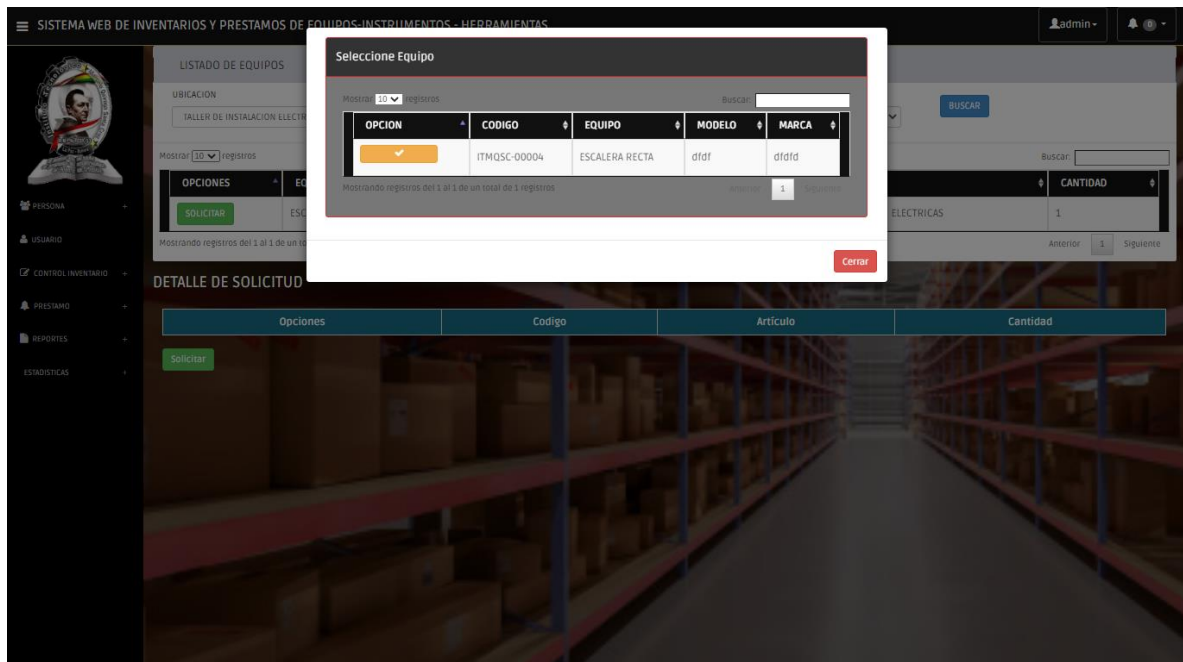


Fuente: Elaboración propia (2020)

3.4.1.3.4. Diseño de Implementación – Módulo de Préstamos.

En la figura 3.25 se muestra el registro para la solicitud de préstamo de equipos.

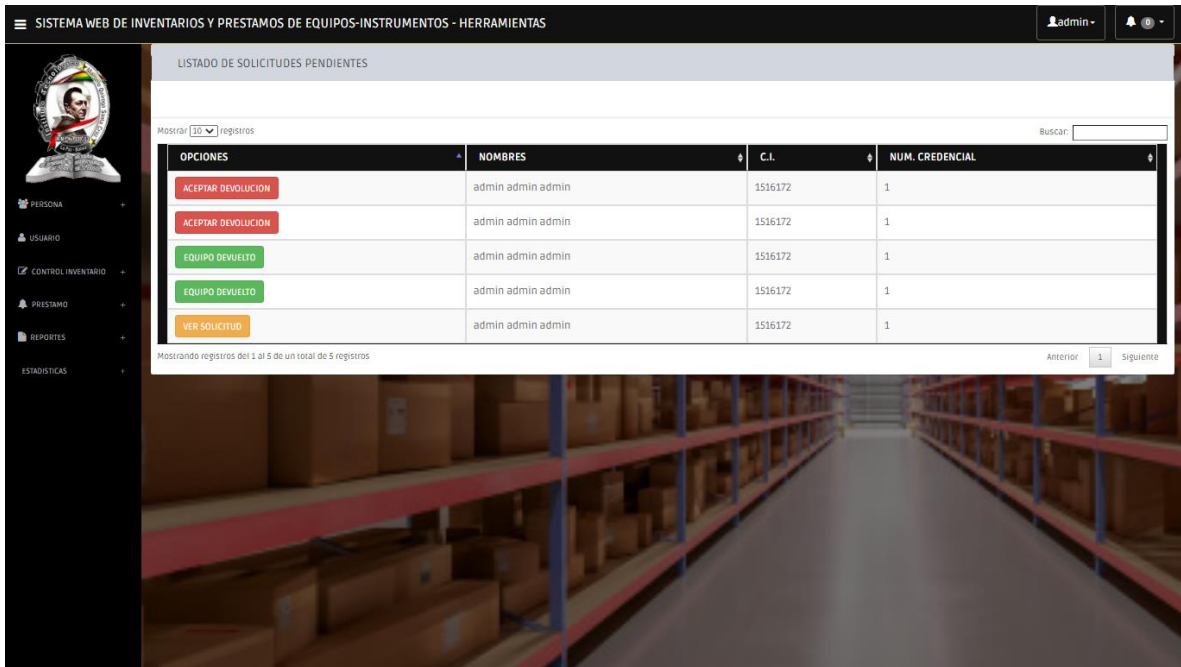
Figura 3.25. Interfaz – Lista y registro de solicitud de préstamo de equipo.



Fuente: Elaboración propia (2020)

La figura 36 muestra el listado de todas las solicitudes pendientes por aprobar.

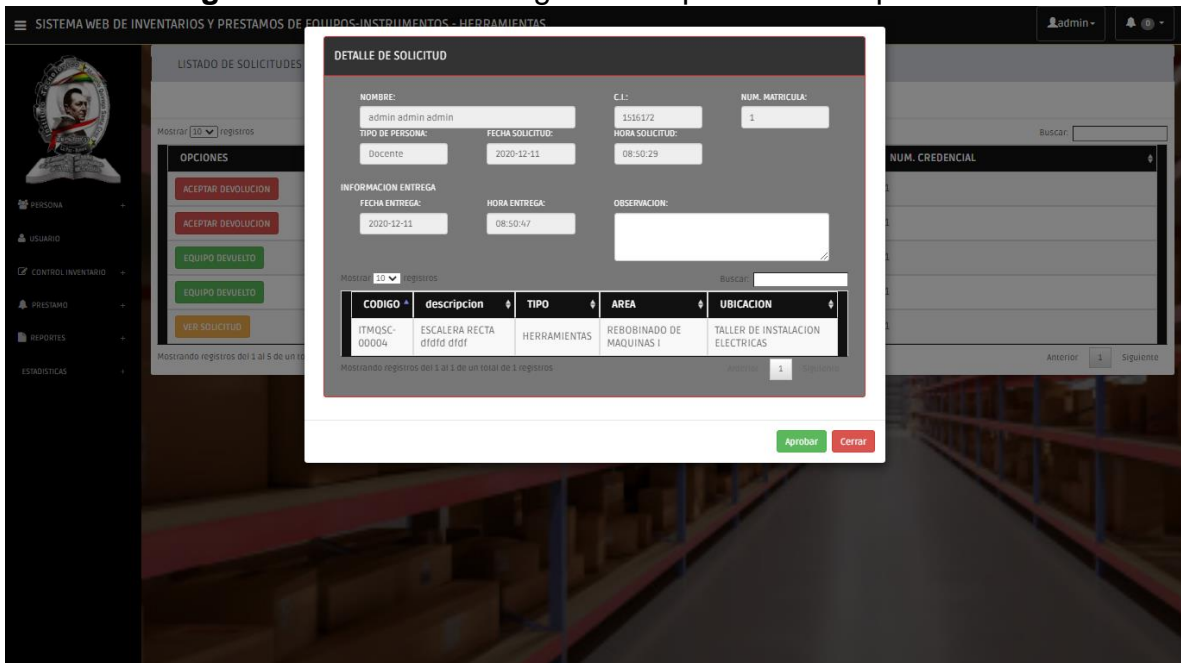
Figura 3.26. Interfaz - Listado de solicitudes pendientes.



Fuente: Elaboración propia (2020)

En la figura 37 se muestra la interfaz para la aprobación del préstamo de equipo.

Figura 3.27. Interfaz –Registro de aprobación de préstamo.



Fuente: Elaboración propia (2020)

La figura 3.28 se observa el formulario de registro de devolución de equipo con todos los detalles.

Figura 3.28. Interfaz – Registro de devolución de equipo.

DETALLE DE DEVOLUCION

NOMBRE: admin admin admin C.I.: 1516172 NUM. MATRICULA: 1

TIPO DE PERSONA: Docente FECHA SOLICITUD: 2020-12-11 HORA SOLICITUD: 04:32:36

INFORMACION ENTREGA REALIZADO ANTERIORMENTE

FECHA ENTREGA: 2020-12-11 HORA ENTREGA: 04:32:43 OBSERVACION:

DEVOLUCION

FECHA ENTREGA: 2020-12-11 HORA ENTREGA: 08:36:49 OBSERVACION DEVOLUCION:

CODIGO	EQUIPO	TIPO	AREA	UBICACION
ITMOSC-00005	AMPERIMETRO ANALOGICO CA sdsds sdsds	INSTRUMENTOS DE MEDIDA	LAB DE AUTOMOTISMOS ELECTRICOS II	LABORATORIO 2

Mostrando registros del 1 al 1 de un total de 1 registros

Devolucion Cerrar

Fuente: Elaboración propia (2020)

3.4.1.4. Cuarta Iteración - Módulo de Reportes y Búsquedas.

Tabla 3.12. Cuarta Iteración.

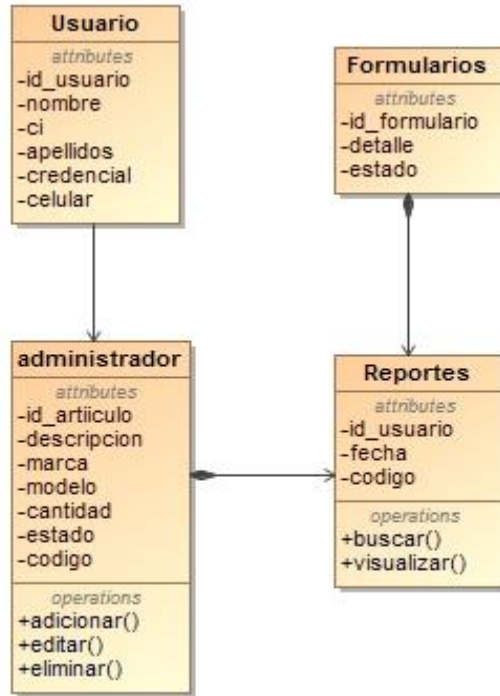
CICLO DE TRABAJO		SPRINT	INICIO	DURACIÓN
		4	22/10/2020	10 días
ID		Tipo	Días de trabajo	Estado
1.	Realizar la planificación de la iteración.	Planificación	1	Terminado
2.	Analizar los requerimientos de backlog del producto.	Planificación	1	Terminado
3.	Complementar el diseño navegacional.	Modelado	1	Terminado
4.	Complementar diseño conceptual.	Modelado	1	Terminado
5.	Complementar diseño de interfaz abstracta.	Modelado	1	Terminado
6.	Desarrollar el módulo de registro de reportes .	Desarrollo	2	Terminado
7.	Desarrollar la página de búsquedas.	Desarrollo	2	Terminado
8.	Implementación del sistema web.	Desarrollo	1	Terminado

Fuente: Elaboración propia (2020)

3.4.1.4.1. Diseño Conceptual – Módulo de Reportes.

Para el diseño conceptual del módulo de reportes se contempla la búsqueda de fechas con detalle para préstamos e inventario.

Figura 3.29. Diseño Conceptual - Módulo Reportes.

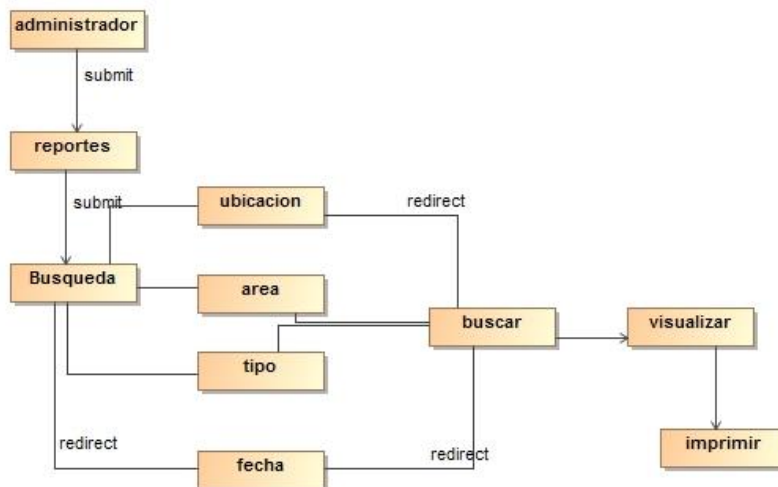


Fuente: Elaboración propia (2020)

3.4.1.4.2. Diseño Navegacional - Módulo de Reportes.

El diseño navegacional del módulo de reportes contempla la búsqueda por área, ubicación y fecha para poder visualizar y también imprimir.

Figura 3.30. Diseño Navegacional - Módulo de Reportes.



Fuente: Elaboración propia (2020)

3.4.1.4.3. Diseño de Interfaces Abstractas – Módulo de Reportes.

Figura 3.31. Diseño Abstracto - Módulo Reportes.



Fuente: Elaboración propia (2020)

3.4.1.4.4. Diseño de Implementación – Módulo de Reportes.

En la figura 3.32 se muestra el listado de reportes de equipos, instrumentos y herramientas según la fecha y ubicación con la finalidad de visualizar el en pantalla la información.

Figura 3.32. Interfaz – Listado de reportes de equipos, instrumentos y herramientas según la fecha y ubicación.

Código	Serie	Descripción	Ubicación	Área	Tipo	Garantía	Fecha Compra cantidad	Precio Bs	Cantidad	Estado
ITMGSC-00001	43566	AUTOTRANSFORMADORES MONOFASICOS m3ac modelo	LABORATORIO 1	LAB DE CIRCUITOS ELECTRICOS I	EQUIPO	1	2020-12-10	450.5	1	
ITMGSC-00002	678989	AMPERIMETRO ANALOGICO DC 055ds 555ds	LABORATORIO 2	LAB DE AUTOMATISMOS ELECTRICOS I	INSTRUMENTOS DE MEDIDA	5	2020-12-10	670.9	1	
ITMGSC-00004	7279588	ESCALERA RECTA 01010 0101	TALLER DE INSTALACION ELECTRICAS	REBORINADO DE MAQUINAS I	HERRAMIENTAS	3	2020-12-10	790.9	1	
Total Equipos:									3	

Fuente: Elaboración propia (2020)

La figura 3.34 nos muestra el listado de reporte de equipos dados de baja por fecha.

Figura 3.33. Interfaz - Reporte de Baja de Equipos.

Codigo	Serie	Descripcion	Ubicacion	Area	Tipo	Precio Bs	Garantia	Justificacion	Estado
ITMOSC-00005	501961	AMPERIMETRO ANALOGICO CA 500-50505	LABORATORIO 2	LAB DE AUTOMOTISMOS ELECTRICOS II	INSTRUMENTOS DE MEDIDA	340.8	2		BAM

Fuente: Elaboración propia (2020)

La figura 3.35 muestra el listado del reporte trimestral de equipos según lo solicitado por el instituto.

Figura 3.34. Interfaz – Reporte Trimestral de Equipos.

Codigo	Serie	Descripcion	Ubicacion	Area	Tipo	Fecha Compra	Costo Bs.
ITMOSC-00001	45566	AUTOTRANSFORMADORES MONOFASICOS	LABORATORIO 1	LAB DE CIRCUITOS ELECTRICOS I	EQUIPO	2020-12-10	450.5
ITMOSC-00002	678989	AMPERIMETRO ANALOGICO DC	LABORATORIO 2	LAB DE AUTOMOTISMOS ELECTRICOS I	INSTRUMENTOS DE MEDIDA	2020-12-10	670.9
ITMOSC-00004	7879988	ESCALERA RECTA	TALLER DE INSTALACION ELECTRICAS	REBOBINADO DE MAQUINAS I	HERRAMIENTAS	2020-12-10	790.9

Fuente: Elaboración propia (2020)

La figura 3.36 muestra el listado para reporte semestral de equipos.

Figura 3.35. Interfaz - Reporte Semestral de Equipos.

Codigo	Serie	Descripción	Ubicación	Area	Tipo	Fecha Compra	Costo Bs.
ITMQSC-00001	45566	AUTOTRANSFORMADORES MONOFASICOS	LABORATORIO 1	LAB DE CIRCUITOS ELECTRICOS I	EQUIPO	2020-12-10	450.5
ITMQSC-00002	678989	AMPERIMETRO ANALOGICO DC	LABORATORIO 2	LAB DE AUTOMOTISMOS ELECTRICOS I	INSTRUMENTOS DE MEDIDA	2020-12-10	670.9
ITMQSC-00004	7879988	ESCALERA RECTA	TALLER DE INSTALACION ELECTRICAS	REBOBINADO DE MAQUINAS I	HERRAMIENTAS	2020-12-10	790.9

Fuente: Elaboración propia (2020)

En la figura 3.37 se observa la interfaz del listado de reporte anual de equipos.

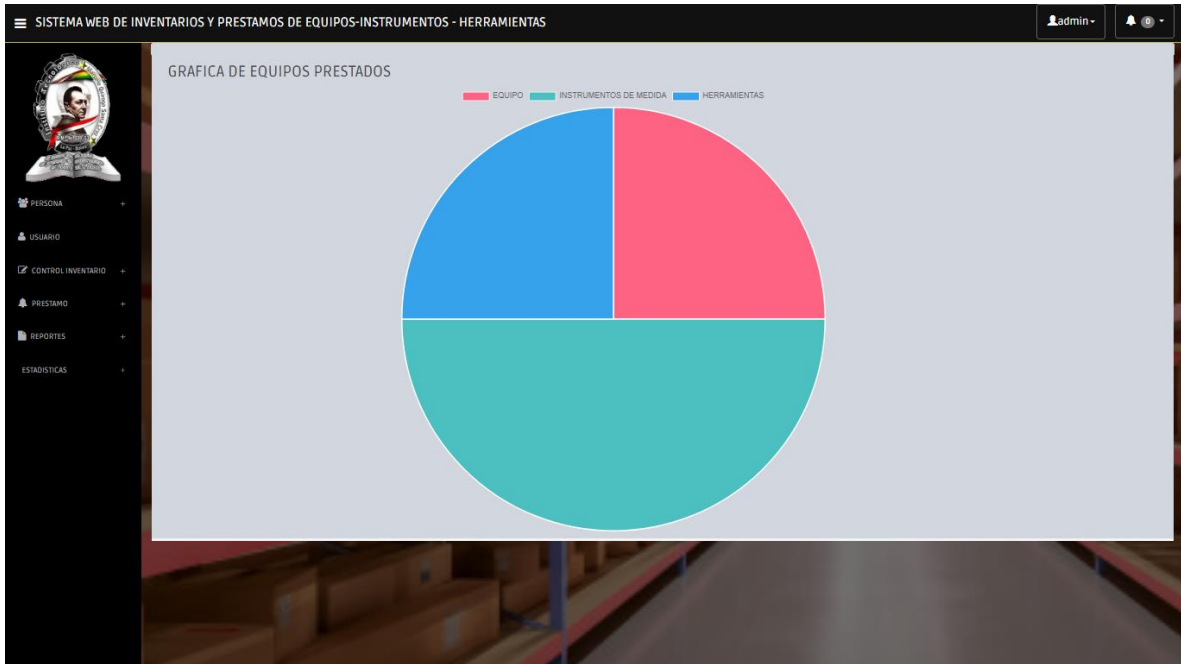
Figura 3.36. Interfaz - Reporte Anual de Equipos.

Codigo	Serie	Descripción	Ubicación	Area	Tipo	Fecha Compra	Costo Bs.
ITMQSC-00001	45566	AUTOTRANSFORMADORES MONOFASICOS	LABORATORIO 1	LAB DE CIRCUITOS ELECTRICOS I	EQUIPO	2020-12-10	450.5
ITMQSC-00002	678989	AMPERIMETRO ANALOGICO DC	LABORATORIO 2	LAB DE AUTOMOTISMOS ELECTRICOS I	INSTRUMENTOS DE MEDIDA	2020-12-10	670.9
ITMQSC-00004	7879988	ESCALERA RECTA	TALLER DE INSTALACION ELECTRICAS	REBOBINADO DE MAQUINAS I	HERRAMIENTAS	2020-12-10	790.9

Fuente: Elaboración propia (2020)

En la figura 3.38 nos muestra el grafico estadístico de los equipos, instrumentos y herramientas prestados para la toma de decisiones.

Figura 3.37. Grafica de Equipos Prestados.



Fuente: Elaboración propia (2020)

3.5. POST GAME (después del juego).

3.5.1. Pruebas del Sistema.

Para mantener la seguridad en el manejo de las contraseñas de los usuarios del sistema, en este caso el administrador y los usuarios con acceso al portal de administración se debe encriptar las contraseñas.

3.5.2. Prueba de Caja Negra.

Figura 3.38. Prueba de Caja Negra - Inicio de Sesión.

Fuente: Elaboración propia (2020)

La tabla 3.13 describe el detalle de las posibles entradas de datos para el inicio de sesión con la finalidad de demostrar la prueba de caja negra en casos del sistema.

Tabla 3.13. Prueba equivalencia - Inicio de Sesión.

POSIBLES ENTRADAS		SALIDAS ESPERADAS	COMENTARIO
Usuario	Contraseña	Mensaje	
“ ”	“ ”	Ingrese el usuario Ingrese contraseña	
Juan	*****	Ingrese usuario	Debe introducir un usuario existente sin caracteres especiales para evaluar la contraseña.
R1121	“ ”	Ingrese la contraseña	Debe introducir la contraseña para evaluar al usuario
123XX	*****	La contraseña o el usuario son incorrectas	Si los datos son incorrectos al comprobar en la BD
Admin123	*****	Acceso permitido	Ingreso al sistema

Fuente: Elaboración propia (2020)

Figura 29. Prueba de Caja Negra – Control de inventario.

Fuente: Elaboración propia (2020)

Tabla 3.14. Prueba Caja Negra – Control de Inventario.

Modulo inventario	Entradas	Prioridad baja	Prioridad media	Prioridad alta	Conclusión
Ingreso de código	Alfanumérico	Validación mayúsculas	Validación números	Validación alfanumérico	Validar entradas varchar
Serie	Alfanumérico	Validación mayúsculas y minúsculas	Validación números	Validación alfanumérico	Validar entradas varchar
Ubicación	Selección		Media		Verificar selección
Tipo	Selección		Media		Verificar selección
Equipo	Selección		Media		Verificar selección
Marca	Texto	Números	Texto	Texto	Verificar text
Modelo	Texto	Números	Texto	Texto	Verificar text
Costo de equipo	Decimal	Numero	Decimal	Decimal	Verificar float
Años de garantía	Numero	Alfanumérico	Entero	Entero	Verificar int
Fecha	Seleccionar		Media		Verificar selección

Fuente: Elaboración propia (2020)

CAPÍTULO IV

CALIDAD DE SOFTWARE

Y SEGURIDAD

4.1 INTRODUCCIÓN.

En este capítulo se realizan la medición de las métricas de calidad ya que mediante ellas se puede testear la calidad de software, la medición de calidad de software se realiza a través de métricas de control de calidad, para medir los factores como ser: funcionalidad, confiabilidad, mantenibilidad, portabilidad, entre otros.

4.2. CALIDAD ISO/ IEC 9126.

La calidad del software se evalúa teniendo en cuenta la etapa del desarrollo, se deben fijar las metas de la calidad tanto para el software final como para desarrollos incompletos y tener en cuenta que es imposible que las metas y criterios sean iguales para un software pequeño y un gran software empresarial.

Se establecen categorías para las cualidades de la calidad externa e interna y calidad en uso del software, teniendo en cuenta estos 7 indicadores: funcionalidad, confiabilidad, utilidad, eficiencia, capacidad de mantenimiento, portabilidad y calidad en uso, que se subdividen a su vez en varios indicadores estas se pueden medir por métrica interna o externa.

4.2.1. Funcionalidad.

Funcionalidad es la capacidad del software de cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas. Ejemplo, lo que hace el software para satisfacer las necesidades sin tener en cuenta de cómo lo hace y cuando.

La funcionalidad se divide en 5 criterios:

- **Adecuación:** La capacidad del software para proveer un adecuado conjunto de funciones que cumplan las tareas y objetivos especificados por el usuario.

- **Exactitud:** La capacidad del software para hacer procesos y entregar los resultados solicitados con precisión o de forma esperada.
- **Interoperabilidad:** La capacidad del software de interactuar con uno o más sistemas específicos.
- **Seguridad:** La capacidad del software para proteger la información y los datos de manera que los usuarios o los sistemas no autorizados no puedan acceder a ellos para realizar operaciones, y la capacidad de aceptar el acceso a los datos de los usuarios o sistemas autorizados.
- **Conformidad de la funcionalidad:** La capacidad del software de cumplir los estándares referentes a la funcionalidad.

La funcionalidad de un sistema no puede ser medido directamente, entonces corresponde derivar mediante otras medidas directas como el punto función, para esto se tiene la siguiente relación:

$$PF = \text{Cuenta_Total} \times [0,65 + 0,01 \times \sum (Fi)]$$

Número de entradas del usuario. se toma en cuenta cada entrada del usuario, estas entradas pueden ser: actualizar, agregar y borrar datos del sistema. En el sistema se tiene los siguientes registros:

- Registro, edición y eliminación de usuarios.
- Registro, edición y eliminación de personas.
- Registro, edición y eliminación de artículos y equipos.
- Registro y modificación de préstamos.
- Asignación de permisos a usuarios.
- Registro de bajas de equipos.
- Registro de la devolución de artículos.

Número de salidas de usuario. Se toma en cuenta cada salida que brinda información al usuario. Aplicando al sistema se tiene:

- Reporte de usuarios.
- Reporte de equipos.
- Reporte de préstamos.

Número de peticiones de usuario. Son la cantidad de entradas interactivas que producen la generación de salidas inmediatas del sistema. Se menciona las siguientes peticiones.

- Listado de inventario.
- Listado de préstamos.
- Listado de bajas de equipos.
- Listado de devoluciones.
- Listado de usuarios, personas, roles, opciones, y accesos.

Numero de archivos:

- Cantidad de tablas de la base de datos. (10)
- Menus principales y emergentes. (12)
- Archivos lógicos de formularios. (7)
- Archivos de programa. (20)

Numero de interfaces externas:

- Backups.
- Impresoras.
- Conexión de red.

Habiendo obtenido estos datos se procede al cálculo de los puntos de función, asignado un factor de ponderación, como se muestra en la tabla.

Tabla 4.1. Calculo del Punto Función.

Parámetros de Medida	Cuenta	Factor de Peso			Total
		Simple	Medio	Complejo	
Número de Entradas del Usuario	8	3	4	6	32
Número de Salidas del usuario	4	4	5	7	20
Número de Peticiones de Usuario	6	3	4	6	36
Número de Archivos	49	7	10	15	343
Número de Interfaces	3	5	7	10	21
Total					452

Fuente: Elaboración propia (2020)

$\Sigma (Fi)$ puede ser de 1 a 14 los valores de ajuste de la complejidad según las respuestas a las siguientes preguntas:

Tabla 4.2. Valores de Ajuste.

Preguntas para los valores de ajuste	Valor
1. ¿Requiere el sistema de copias de seguridad y recuperación fiables?	4
2. ¿Se requiere comunicación de datos?	5
3. ¿Existen funciones de procesamiento distribuido?	3
4. ¿Es crítico el rendimiento?	2
5. ¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado?	5
6. ¿Requiere el sistema entrada de datos interactiva?	4
7. ¿Requiere la entrada de datos interactiva que las transacciones de datos se lleven a cabo sobre múltiples pantallas u operaciones?	2
8. ¿Se actualizarán los archivos maestros de forma interactiva?	4
9. ¿Son complejas las entradas, las salidas, los archivos o las peticiones?	2
10. ¿Es complejo el procesamiento interno?	2
11. ¿Se ha diseñado código para que sea reutilizable?	4
12. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	4
13. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente usada por el usuario?	5
Total	46

Fuente: Elaboración propia (2020)

Cada una de las preguntas anteriores es respondida usando la escala de rangos desde 0 (No importante o aplicable) hasta 5 (Absolutamente esencial).

Una vez que se han calculado los puntos de función se usa de forma analógica a las LDC como forma de normar a la productividad, calidad y otros atributos del software.

0	1	2	3	4	5
Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial

$\Sigma (F_i)$: (i=1 hasta 14) son los valores de ajuste de complejidad.

1=5, 2=3, 3=4, 4=3, 5=5, 6=4, 7=4, 8=5, 9=3, 10=4, 11=3, 12=1, 13=2, 14=3

$\Sigma (F_i) = 49$

Si consideramos el máximo valor de complejidad como $\Sigma (F_i) = 46$, se tiene:

$PF = 452 * (0.65 + (0.01 * 46))$

$PF = 501.72$

Hallamos los puntos de función máximo.

$PF \text{ máximo} = 452 * (0.65 + (0.01 * 49))$

$PF \text{ máximo} = 515.28$

Entonces si $\Sigma (F_i)$ es considerada como el 100%, la relación obtenida entre los puntos será:

$(PF / PF \text{ máximo}) * 100$

$PF = 501.72 / 515.28 = 0.96$

$PF = 96\%$

Por lo tanto, la funcionalidad que tiene el sistema es del 96%, tomando en cuenta el punto de función máximo.

4.2.2. Confiabilidad.

La confiabilidad se divide en 4 criterios:

- **Madurez:** La capacidad que tiene el software para evitar fallas cuando encuentra errores. Ejemplo, la forma como el software advierte al usuario cuando no encuentra espacio suficiente el disco duro donde esta almacenando los datos.
- **Tolerancia a errores:** La capacidad que tiene el software para mantener un nivel de funcionamiento en caso de errores.
- **Recuperabilidad:** La capacidad que tiene el software para restablecer su funcionamiento adecuado y recuperar los datos afectados en el caso de una falla.
- **Conformidad de la fiabilidad:** La capacidad del software de cumplir a los estándares o normas relacionadas a la fiabilidad.

La confiabilidad del sistema es directamente proporcional a la calidad de sus componentes, por tal razón se tomó las siguientes consideraciones:

El sistema comienza su trabajo en un tiempo inicial, denominado t_0 ; con $t_0 = 0$, donde se observa a t como tiempo de falla.

Se designa una variable aleatoria T , que representa el tiempo de trabajo del sistema sin fallas, la probabilidad de que un sistema falle en un tiempo será: $P[T > t] = F(t)$ y la probabilidad de funcionamiento del sistema será: $P[T > t] = 1 - F(t)$

Se denominará la confiabilidad del sistema como:

$R(t)$: Función de confiabilidad de un componente en el tiempo t .

F : es la funcionalidad del sistema ya calculado (0,97)

λ : Taza constante de fallo o error en el sistema. (0,03)

t : Periodo de operación de tiempo. (12 meses)

El margen de error λ es de 3/10 que se calculó realizando 10 ejecuciones en un mes que se produzca fallas en el sistema.

Calculando para una gestión de un año, junto a al valor de los puntos de función. A continuación, se realiza los siguientes cálculos.

$$P[T > t] = F(t)$$

Donde:

$$F(t) = PF * e^{\left(-\frac{\lambda}{10}\right)} ; \text{ con } \lambda > 0$$

Calculamos la probabilidad del sistema sin fallas.

$$F(t) = 0,96 * e^{\left(-\frac{3}{10} * 12\right)} ; t=12$$

$F(12) = 0,02$; entonces se tiene $0,02 * 100 = 2\%$

La probabilidad que el sistema tenga fallas es de 3%

Calculo del sistema sin fallas:

$$F(t) = 1 - 0,96 * e^{\left(-\frac{3}{10} * 12\right)}$$

$$F(12) = 1 - 0,02 = 0,98$$

Por tanto, el sistema tiene un grado de confiabilidad de 98% y que seguirá funcionando en un año.

4.2.3. Usabilidad.

La usabilidad es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva. Algunos criterios de funcionalidad, fiabilidad y eficiencia afectan la usabilidad, pero para los propósitos de la ISO/IEC 9126 ellos no clasifican como usabilidad.

La usabilidad está determinada por los usuarios finales y los usuarios indirectos del software, dirigidos a todos los ambientes a la preparación del uso y el resultado obtenido.

Las métricas de usabilidad son:

- **Entendimiento:** La capacidad que tiene el software para permitir al usuario entender si es adecuado, y de una manera fácil como ser utilizado para las tareas y las condiciones particulares de la aplicación. En este criterio se debe tener en cuenta la documentación y de las ayudas que el software entrega.
- **Aprendizaje:** La forma como el software permite al usuario aprender su uso. También es importante considerar la documentación.
- **Operabilidad:** La manera como el software permite al usuario operarlo y controlarlo.
- **Atracción:** La presentación del software debe ser atractivo al usuario. Esto se refiere a las cualidades del software para hacer más agradable al usuario, ejemplo, el diseño gráfico.
- **Conformidad de uso:** La capacidad del software de cumplir los estándares o normas relacionadas a su usabilidad.
- **Tasa de errores:** De acuerdo con la confiabilidad obtenida en función a un número de fallas en un rango de pruebas hechas. La tasa de errores se ajusta a la siguiente relación.

Tasa de errores = (numero fallas/ numero pruebas) = 5/15 = 0.3

Para calcular la usabilidad del sistema se realizó la siguiente escala de valores como se muestra en la tabla 21.

Tabla 4.3. Escala de Valores.

Escala	Valor
Muy buena	5
Buena	4
Regular	3
Malo	2
Pésimo	1

Fuente: Elaboración propia (2020)

Tabla 4.4. Preguntas de Evaluación.

Pregunta	Evaluación
¿el sistema es fácil de utilizar?	5
¿ Los registros de la página web son eficientes y seguros?	5
¿ El sistema facilita el proceso de préstamo que usted realiza?	5
¿Se ha satisfecho todos los requerimientos establecidos?	5
¿Cómo considera los formularios que elabora el sistema?	4
¿El sistema tien la seguridad necesaria?	5
¿Cómo considera el ingreso de datos al sistema?	5
¿La generación de resultados le ayuda al proceso de toma de decisiones?	4
Total	38

Fuente: Elaboración propia (2020)

Tomando en cuenta que el número de preguntas son 8 y los consultores que realizaron las pruebas son 5 personas se tiene:

$$FU = \left[\sum Xi/n * 100 \right] / N$$

$$FU = [(38/8)*100]/5$$

$$FU = 95$$

En conclusión, la facilidad de uso es del 95%.

4.2.4. Capacidad de Mantenimiento.

La capacidad de mantenimiento es la cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, a cambios en el entorno, y especificaciones de requerimientos funcionales.

- **Capacidad de ser analizado:** La forma como el software permite diagnósticos de deficiencias o causas de fallas, o la identificación de partes modificadas.
- **Cambiabilidad:** La capacidad del software para que la implementación de una modificación se pueda realizar, incluye también codificación, diseño y documentación de cambios.
- **Estabilidad:** La forma como el software evita efectos inesperados para modificaciones del mismo.
- **Facilidad de prueba:** La forma como el software permite realizar pruebas a las modificaciones sin poner el riesgo los datos.
- **Conformidad de facilidad de mantenimiento:** La capacidad que tiene el software para cumplir con los estándares de facilidad de mantenimiento.

Para calcular la mantenibilidad del sistema utilizaremos las medidas directas proporcionadas por la IEEE 982.1 – 1998 el cual sugiere un índice de madurez del sistema, que consiste en los cambios que producen en cada versión del producto para lo cual se tiene la siguiente relación.

$$\text{IMS} = [\text{MT} - (\text{Fa} + \text{Fb} + \text{Fc})] / \text{MT}$$

Dónde:

MT = Número de módulos en la versión actual.

Fa = Número de módulos en la versión actual que se han cambiado.

Fb = Número de módulos en la versión actual que se han añadido.

Fc = Número de módulos en la versión anterior que se han borrado en la versión actual.

Si el valor de IMS se aproxima a 1, el sistema empieza a estabilizarse.

Tabla 4.5. Versiones del Sistema.

Versión	MT	Fa	Fb	Fc	IMs%
Versión 1	10	2	3	0	50%
Versión 2	15	5	1	0	60%
Versión 3	20	1	2	0	85%
Versión 4	23	1	1	0	91%

Fuente: Elaboración propia (2020)

Por tanto, se concluye que el índice de madurez del sistema es del 91% al momento de realizar el mantenimiento.

4.2.5. Portabilidad.

Es la capacidad que tiene el software para ser trasladado de un entorno a otro, cuenta con las siguientes métricas de calidad:

- **Adaptabilidad:** Es como el software se adapta a diferentes entornos especificados (hardware o sistemas operativos) sin que implique reacciones negativas ante el cambio. Incluye la escalabilidad de capacidad interna (Ejemplo: Campos en pantalla, tablas, volúmenes de transacciones, formatos de reporte, etc.).
- **Facilidad de instalación:** La facilidad del software para ser instalado en un entorno específico o por el usuario final.
- **Coexistencia:** La capacidad que tiene el software para coexistir con otro o varios softwares, la forma de compartir recursos comunes con otro software o dispositivo.
- **Reemplazabilidad:** La capacidad que tiene el software para ser reemplazado por otro software del mismo tipo, y para el mismo objetivo. Ejemplo, la reemplazabilidad de una nueva versión es importante para el usuario, la propiedad de poder migrar los datos a otro software de diferente proveedor.
- **Conformidad de portabilidad:** La capacidad que tiene el software para cumplir con los estándares relacionados a la portabilidad.

Además de estos aspectos también podemos considerar la facilidad de instalación, ajuste y adaptación al cambio dado por la siguiente formula:

$$GP=1-[CT/CRD]$$

Dónde:

GP= Grado de Portabilidad

CT= Costo de Transportar

CRD= Costo de Re- desarrollo

Si $GP > 0$, la portabilidad es más rentable que el re-desarrollo

Si $GP=1$, la portabilidad es perfecta

Si $GP < 0$, el re-desarrollo es más rentable que la portabilidad.

$$GP=1-[75/1000]$$

$$GP= 0.9625$$

Por lo tanto, el sistema puede transportarse de un entorno a otro en un 96% más rentable que el desarrollo.

CAPÍTULO V

ANÁLISIS DE COSTOS

5.1. INTRODUCCIÓN.

En este capítulo se describen el costo - beneficio que conlleva realizar el presente proyecto de grado, mediante la estimación de costos que se efectuara después del desarrollo de software, haciendo uso de del modelo post – arquitectura, calculando el esfuerzo y costo COCOMO II.

5.2. ANÁLISIS DE COSTOS.

Uno de los trabajos más importantes al momento de desarrollar un Sistema es la estimación de costos, la cual consiste en determinar con cierto grado de certeza los recursos de hardware y software, costo, tiempo y esfuerzo necesario para el desarrollo de los mismos.

5.3. COCOMO II.

Cocomo II nos permite realizar estimaciones en función al tamaño del software, para este caso se utilizará el modelo Post-Arquitectura.

Para ello se encontrará el valor de los puntos de función, el cual ya fue obtenido anteriormente al obtener cuyo valor es el siguiente:

$$PF = 0,96 (96\%)$$

5.3.1. Líneas de Código.

Es necesario calcular las líneas de código (LDC) para ello el Factor LDC/PF, para JavaScript el valor es de 47 porque es una de las librerías que se utilizó en el sistema.

$$LDC = PFA*(Factor LDC/PF)$$

$$LDC = 96*47$$

$$LDC = 4.512$$

Para calcular KLDC.

$$KLDC = LDC/1000$$

$$KLDC = 4.512/1000$$

$$KLDC = 4,512$$

5.3.2. Factor de Escala.

A continuación, se debe calcular la sumatoria de los Factores de Escala (ΣW_i).

Tabla 5.1. Factores de Escala.

Factor de Escala	Valor
PREC	1.24
FLEX	3.04
RESL	5.65
TEAM	2.19
PMAT	1.56
Total, Factores de Escala (ΣW_j).	13,63

Fuente: Elaboración propia (2020)

Calculamos (B) Factor de escala, basado en factores de escala.

$$B = 0.91 + 0.01 * (\Sigma W_j)$$

$$B = 0.91 + 0.01 * (13.63)$$

$$B = 1.0463$$

5.3.3. Esfuerzo Nominal (PM).

Se obtiene el Esfuerzo Nominal reemplazando los datos en la siguiente ecuación:

$$PM_{nominal} = A * (KLDC)^B$$

$$PM_{nominal} = 2.94 * (4,512)^{1.0463}$$

$$PM_{nominal} = 14.22$$

5.3.4. Multiplicadores de Esfuerzo.

Tabla 5.2. Multiplicadores de Esfuerzo.

Conductores de Costo	Nro.	Multiplicadores de Esfuerzo	Valor
PRODUCTO	1	RELY	1
	2	DATA	0.94
	3	CPLX	0.85
PLATAFORMA	6	TIME	1
	7	STOR	1
	8	VIRT	0.87
		TURN	1
PERSONAL	9	ACAP	1
	10	PCAP	0.86
	12	AEXP	0.91
	13	VEXP	0.90
	14	LEXP	0.95
PROYECTO	15	TOOL	1
	16	MOPD	0,91
	17	SCED	1
TOTAL PRODUCTO			0.4233

Fuente: Elaboración propia (2020)

El Esfuerzo estimado se calcula según la siguiente fórmula:

$$PM_{\text{estimado}} = PM_{\text{nominal}} * \Sigma EM$$

$$PM_{\text{estimado}} = 14,22 * 0.4233$$

$$PM_{\text{estimado}} = 6.019 \text{ (Personas/Mes)}$$

Por tanto, para el desarrollo del software en un mes se requerirá de 6 programadores.

Tabla 5.3. Sueldo de Programadores.

Empresa	Salario Bs
Jalasoft	4.176
Assuresoft	5,568
Viva	2,784
Promedio	4.176

Fuente: Elaboración propia (2020)

Sumando los salarios en Bs de las nueve personas por mes seria.

$$\text{Costo de Desarrollo} = 6 * 4176$$

$$\text{Costo de Desarrollo} = 25.056$$

Se llega a la conclusión que el desarrollo tanto del Sistema Web un costo de Bs 25.056.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

6.1.1. Respecto al proyecto.

Se logró cumplir con el objetivo de desarrollar un Sistema Web para la Gestión de Inventarios y Prestamos en los laboratorios y talleres de la carrera de electricidad industrial del Instituto Tecnológico Marcelo Quiroga Santa Cruz de esta manera llevar un adecuado control y gestión de dicha información para la administración.

6.1.2. Respecto a los Objetivos Específicos.

Las conclusiones se realizan de acuerdo a los objetivos específicos que se ha propuesto para el desarrollo del proyecto.

- Se realizó un diagnóstico del proceso de movimiento del equipamiento en los laboratorios, talleres de instalaciones eléctricas y taller electromecánico, con el fin de conocer todos los detalles y requerimientos que podrían interferir con el desarrollo del proyecto.
- Se ha realizado el registro de inventario con el propósito de concretar todo el equipamiento asignado a la carrera de electricidad, y así llevar un adecuado control.
- Se ha realizado una interfaz web amigable para los usuarios que intervengan los procesos de control de inventario y prestamos de equipos.
- Se logró realizar informes automatizados mediante la elaboración de reportes dentro del sistema con respecto a los detalles de los prestamos e inventario.
- Se pudo realizar la evaluación y pruebas del sistema web para poder implementarlo y así cumpla con los objetivos y automatización de información.

6.1.3. Respecto al Objetivo General.

Se logró se desarrolló un sistema de información orientado a la web con la cual se pudo cubrir las necesidades para la gestión de inventario y prestamos de los equipos en la carrera de electricidad industrial del Instituto Tecnológico Mariscal Santa Cruz, logrando alcanzar las expectativas del instituto mediante el desarrollo de todos los módulos requeridos.

6.2. RECOMENDACIONES.

6.2.1. Respecto al software.

- Se debe realizar backups mensualmente de la base de datos, se debe proteger las claves de acceso al sistema.
- Se debe revisar el disco duro para que no colapse, también la revisión del servidor web donde se alojará el sistema.
- Se puede sistematizar otras áreas administrativas dentro del instituto para tener un sistema completo de todo el instituto.

BIBLIOGRAFÍA

Andreu, R., Ricart J. y Valor, J. (1991). Estrategia y Sistemas de Información. Madrid: Mc Graw-Hill.

Amo y Alonso. (2005). Introducción a la ingeniería del software, Madrid.

Cárdenas, J. (2011). La utilización de la ingeniería de software en hipermedia. Revista Ciencia UNEMI. 6 (1), 102-117.

Caso, N. G. (2004). SCRUM development process. Buenos Aires: Universidad Tecnológica Nacional.

French, M.J., 1984. Conceptual design for engineers. Springer

Fuertes, A. (2015). Métodos, técnicas y sistemas de valuación de inventarios. Un enfoque global. Revista de la Agrupación Joven Iberoamericana de Contabilidad y Administración de Empresas (AJOICA). 14 (1), 48-65.

Gómez, C. (2011). Notas del curso: análisis de requerimientos. México: Universidad Autónoma Metropolitana.

Heurtel , O. (2016). PHP 7 Desarrollar un sitio web dinámico e interactivo. Barcelona: Ángel BELINCHON CALLEJA.

Tymkiw, N., Bournissen, J. M., & Tumino, M. C. (2020). SCRUM como Herramienta Metodológica para el Aprendizaje de la Programación. Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología (26), 79 - 87. Doi: 10.24215/18509959.26.e9

Horváth, I., 2005. On some crucial issues of computer support of conceptual design, in: Product Engineering. Springer, (1), pp. 123–142

Huanca. (2015). Sistema Web de control de pagos y citas médicas, Universidad Mayor de San Andrés, Carrera de Informática.

Kendall & Kendall. (2000). Análisis y Diseño de Sistemas. 3ª. Edición.

Krajewski L., Malhotra M., Ritzman L. (2008). Administración de Operaciones: Procesos y Cadenas de Valor. México: Pearson.

Lacalle, G. (2013). Operaciones administrativas de compraventa. (s/l): Editex.

Martinez, M. (2012). Diseño y construcción de un sistema de geoposición en visual.net ambientada como Aplicación Web para el manejo de información general de pólizas de seguros que integre el manejo de una Base de Datos textual en MySQL y una Base de Datos Gráfica (AutoCAD). (Tesis licenciatura). Universidad Politécnica Salesiana, Quito.

Menzisky, A., López, G., y J. Castillo, (2016). "Scrum Manager". Recuperado de: <http://www.scrummanager.net>

Moreno, A. (2009). Fundamentos de control interno. México: Cengage learning

Muller, M. (2005). Fundamentos de administración de inventarios. Colombia: Grupo Editorial Norma.

Pressman, R. (2005) Ingeniería del Software, 6ta Edición, McGraw-Hill inter americana editores, México, Bogotá, buenos Aires.

Pressman, R. (2010). Ingeniería del software. Un enfoque práctico. México: McGraw-Hill.

Rossi, G., Schmid, A., y Lyardet, F. (2003). Customizing business processes in web applications. In EC-Web, pp. 359–368.

Schwabe, D. y Rossi, G. (1995). Building hypermedia applications as navigational views of information models. In HICSS (3), pp. 231–240.

Taha, H. (2004). Investigación de Operaciones. México: Pearson Educación.

Vilariño, J. (2010). Modelo para la selección de la metodología de desarrollo web de una aplicación según sus características funcionales. (Tesis de maestría). Universidad Católica Andrés Bello, Caracas.

Villa, J. (2008). Contabilidad general 2. La Habana Cuba: Félix Varela.

WEBGRAFIA

Abud, M. (2012). Calidad en la Industria del Software. La Norma ISO-9126. Recuperado de: <http://www.nacionmulticultural.unam.mx/empresasindigenas/docs/2094.pdf>

Alaimo, M. D. (2013). Proyectos Ágiles con SCRUM. Buenos Aires: Ediciones Kleer. Recuperado de: <https://fliphtml5.com/uof/dlln/basic>

Bakken, S. et al. (2005). Manual de PHP. Editado por Rafael Martínez, publicada por la “Free Software Foundation”, recuperado de: <http://www.opencontent.org/openpub>.

Campos, C. (2015). Las pruebas de desarrollo de software (tesis de pregrado). Universidad Nacional Autónoma de México, México. Recuperado de: <http://132.248.52.100:8080/xmlui/bitstream/handle/132.248.52.100/7627/Las%20pruebas%20en%20el%20desarrollo%20de%20software.pdf?sequence=1>

Cristiá, M. (2014). Introducción a la ingeniería de requerimientos. Recuperado de:
<https://www.fceia.unr.edu.ar/~mcristia/publicaciones/ingreq-a.pdf>

Dias Cortez, & Ocampo. (2016). Diseño Gráfico por Computación. 1-4.
Recuperado de: http://catedraocampo.com.ar/wp-content/uploads/2016/05/dgpc_apunte_CSS.pdf

Eguíluz Pérez, J. (2008). Introducción a CSS. Creative Commons. Recuperado de:
https://www.jesusda.com/docs/ebooks/introduccion_css.pdf

Escalona Cuaresma, M. J. (2001). Metodologías para el desarrollo de sistemas de información global: análisis comparativo y propuesta. Sevilla: USEVILLA.
Recuperado de: <http://www.lsi.us.es/docs/informes/EstadoActual.pdf>

Gallego Sánchez, A. J. (s.f). Laravel. Recuperado de:
<https://www.academia.edu/32488225/Laravel>

Gallego, J., (2003). “Desarrollo Web con PHP y MySQL”. Ediciones Anaya Multimedia Grupo Anaya S. A. recuperado de:
<http://www.lsi.us.es/cursos/cursophp/apuntes/tema4.pdf>

García. I. (2013). Antología de ingeniería web. Recuperado de:
https://www.academia.edu/33443096/ANTOLOG%C3%8DAS_DE_INGENIER%C3%8DA_WEB.

Gómez, A., López, M., Migani, S., y Otazú, A. (2010). Cocomo - un modelo de estimación de proyectos de software. Recuperado de:
<https://blogadmi1.files.wordpress.com/2010/11/cocom0llfull.pdf>

Hernández, A. (2020). Los sistemas de información: evolución y desarrollo. Recuperado de: <https://dialnet.unirioja.es/descarga/articulo/793097.pdf>

Lamarca, J. (2013). Hipertexto, el nuevo concepto de documento en la cultura de la imagen. Recuperado de: <http://www.hipertexto.info/documentos/oohdm.htm#:~:text=OOHDM%20propone%20el%20desarrollo%20de,dise%C3%B1o%20de%20interfaces%20abstractas.>

Largo, C. y Marín, E. (2005). Guía técnica para la evaluación de software. Recuperado de: https://jrvargas.files.wordpress.com/2009/03/guia_tecnica_para_evaluacion_de_software.pdf

Mejía, P. (2009). Ingeniería de software. Recuperado de: <http://delta.cs.cinvestav.mx/~pmalvarez/softeng/curso-2009/Obtencion-requerimientos.pdf>

Mi carrera universitaria (s/a). Ingeniería web: Qué es, características y todo lo que debes saber. Recuperado de: [https://micarrerauniversitaria.com/c-ingenieria/ingenieria-web/.](https://micarrerauniversitaria.com/c-ingenieria/ingenieria-web/)

Peralta, A. (2003). Metodología SCRUM. Uruguay: Universidad ORT. Recuperado de: <https://fi.ort.edu.uy/innovaportal/file/2021/1/scrum.pdf>

Pinto, J. (2015). Metodología OOHDM. Recuperado de: <https://pintojairo.files.wordpress.com/2011/09/metodologc3ada-oohdm1.pptx>

Pinzon, O., (2017). Ingeniería Web: Una Metodología para el Desarrollo de Aplicaciones Web Escalables y Sostenibles. Recuperado de: http://www.laccei.org/LACCEI2017-BocaRaton/student_Papers/SP277.pdf

Quiroga, P., (2017). Requerimientos funcionales y no funcionales. Recuperado de: <http://www.electrohulla.com.co/Portals/0/UpDocuments/0b530417-2986-450e-bd92-34928a11e2f5.pdf>.

Rentería, F. (2005). Contabilidad II. Recuperado de: <http://fcasua.contad.unam.mx/apuntes/interiores/docs/2005/contaduria/2/1257.pdf>

Salazar, B. (2019). Métodos de valoración de inventarios. Recuperado de: <https://www.ingenieriaindustrialonline.com/gestion-de-inventarios/metodos-de-valoracion-de-inventarios/>

Sánchez, J. (2003). Manual de MySQL. Recuperado de: <https://www.cartagena99.com/recursos/programacion/apuntes/mysql.pdf>

Sánchez, J. (2015). Pruebas de software. Fundamentos y técnicas. Universidad Politécnica de Madrid, Departamento de Teoría de la señal y Telecomunicaciones, Madrid. Recuperado de: http://oa.upm.es/40012/1/PFC_JOSE_MANUEL_SANCHEZ_PENO_3.pdf

Schwabe, D. y Rossi, G. (1997). An Object Oriented Approach to Web-Based Application Design. Recuperado de: <http://www-di.inf.puc-rio.br/schwabe/papers/TAPOSRevised.pdf>

Schwabe, D., & Rossi, G. (s.f.). Un enfoque orientado a objetos para el diseño de aplicaciones basadas en web. Obtenido de di.inf.puc-rio.br: <http://www-di.inf.puc-rio.br/schwabe/papers/TAPOSRevised.pdf>

Schwaber, k. y J. Sutherland, (2017). "Guía del Scrum". Recuperado de:
<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-SouthAmerican.pdf>.

Sierra, J., et.al. (2015). Administración de almacenes y control de inventarios.
Recuperado de: <https://www.eumed.net/libros-gratis/2015/1444/index.htm>

Soliz D, R. D., & Morales O, F. A. (2014). OOHDM (Método de diseño Hipermedia objeto orientado) & Normativa ISO 9126. Barinas: Universidad Nacional Experimental de los Llanos Occidentales. Recuperado de:
<https://darjelingsilva.files.wordpress.com/2018/05/5-metd-oohdm.pdf>

Soliz, R. y Morales, F. (2014). OOHDM (método de diseño hipermedia objeto orientado) & normativa ISO 9126. Recuperado de:
<https://darjelingsilva.files.wordpress.com/2018/05/5-metd-oohdm.pdf>.

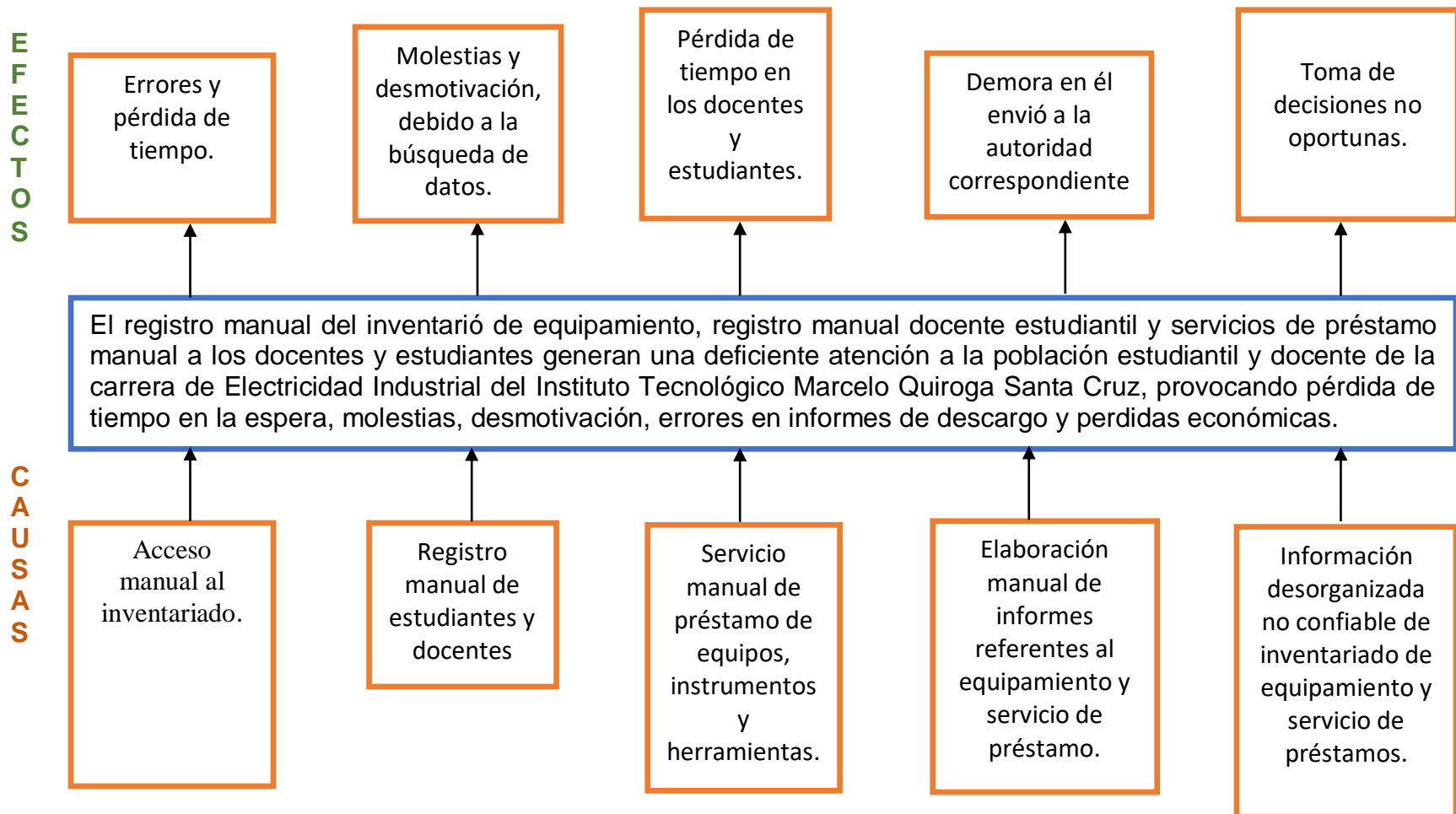
Torres, J. (2000). Diseño del Modelo Navegacional para Sistemas de Tratamiento de Bibliotecas Digitales. Recuperado de:
https://idus.us.es/bitstream/handle/11441/56135/02_2000.pdf?sequence=1&isAllowed=y

Trigás Gallego, M. (2012). Metodología Scrum. España: Universitat Oberta de Catalunya. Recuperado de:
<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>

ANEXOS

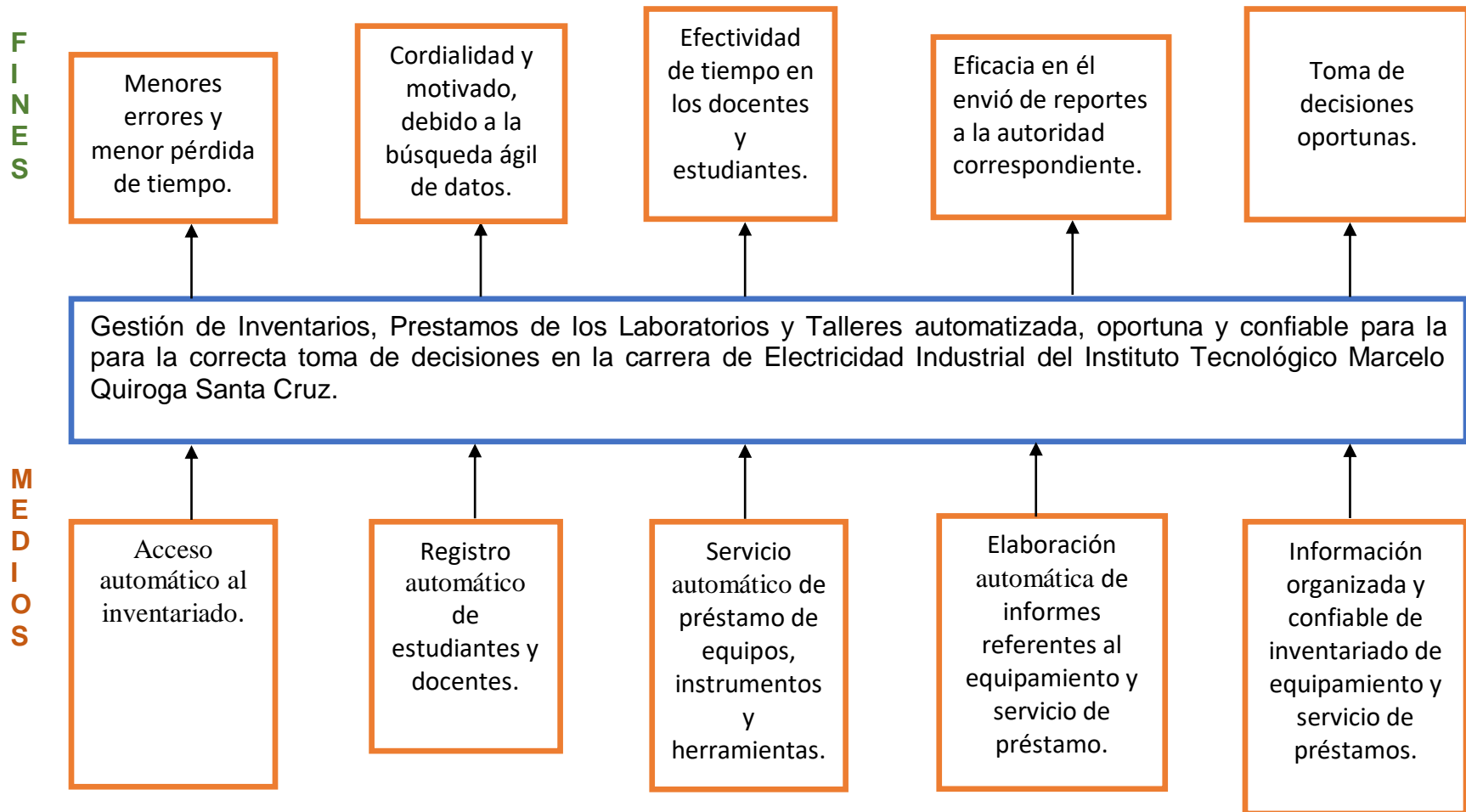
ANEXO A

ARBOL DE PROBLEMAS:



ANEXO B

ARBOL DE OBJETIVOS:



ANEXO C

GUÍA DE ENTREVISTA

1. Datos generales

Nombre: Ing. Elvis Rubin de Celis

Cargo: Jefe de Carrera de Electricidad Industrial

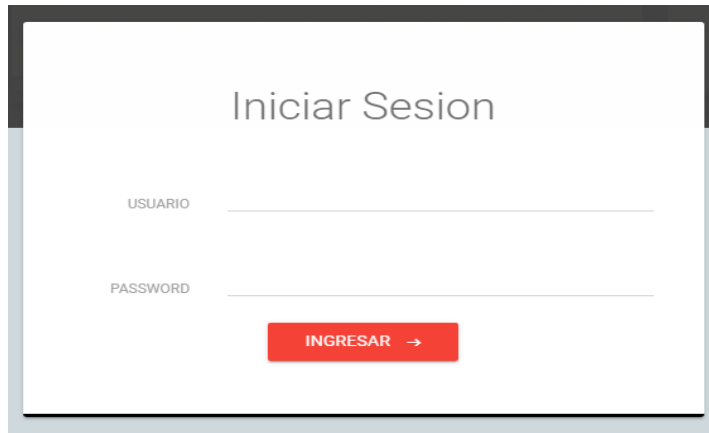
1. ¿En la actualidad cómo es el proceso de atención llevado a cabo en laboratorios y talleres de la carrera de Electricidad Industrial del I? T.M.Q.S.C?
2. ¿Cómo considera usted, la atención que se da a los estudiantes al momento de utilizar los laboratorios y talleres de la carrera de Electricidad Industrial?
3. ¿Por qué opina de esa manera?
4. ¿Cómo considera usted el control del inventario?
5. ¿Qué se necesita mejorar en el control del inventario?
6. ¿Cómo aprecia el empleo del formato orden de trabajo para la solicitud del préstamo de un equipo de los laboratorios o talleres de la carrera de electricidad industrial?

7. ¿Por qué el procedimiento a pesar del avance de la tecnología, sigue siendo manual?
8. ¿Normalmente, qué dificultades hay en el proceso de préstamos de equipos por parte de los estudiantes?
9. ¿Durante el tiempo que estuvo trabajando en la carrera hubo alguna pérdida de equipos dentro de los laboratorios y talleres?
10. ¿Cómo juzga en base a su experiencia la operatividad de los equipos actualmente?
11. ¿Cómo considera usted que se puede mejorar la atención a través del registro del inventario, a partir del desarrollo de un Sistema de Información basado en la web?
12. ¿Usted está de acuerdo con un Sistema de Información basado en la web para el control del inventario y servicios de préstamo de equipos, instrumentos y herramientas que pueda ser amigable para los usuarios?
13. ¿Qué apreciación tiene usted respecto a la organización en el control del inventario en los laboratorios y talleres de la carrera de Electricidad Industrial?

ANEXO D MANUAL DE USUARIO

SISTEMA WEB PARA GESTIÓN DE INVENTARIOS Y PRESTAMOS EN LOS LABORATORIOS Y TALLERES DE LA CARRERA DE ELECTRICIDAD INDUSTRIAL CASO: INSTITUTO TECNICO MARCELO QUIROGA SANTA CRUZ

Ingreso al sistema (Back end). El Administrador debe ingresar su Usuario y contraseña:



Iniciar Sesión

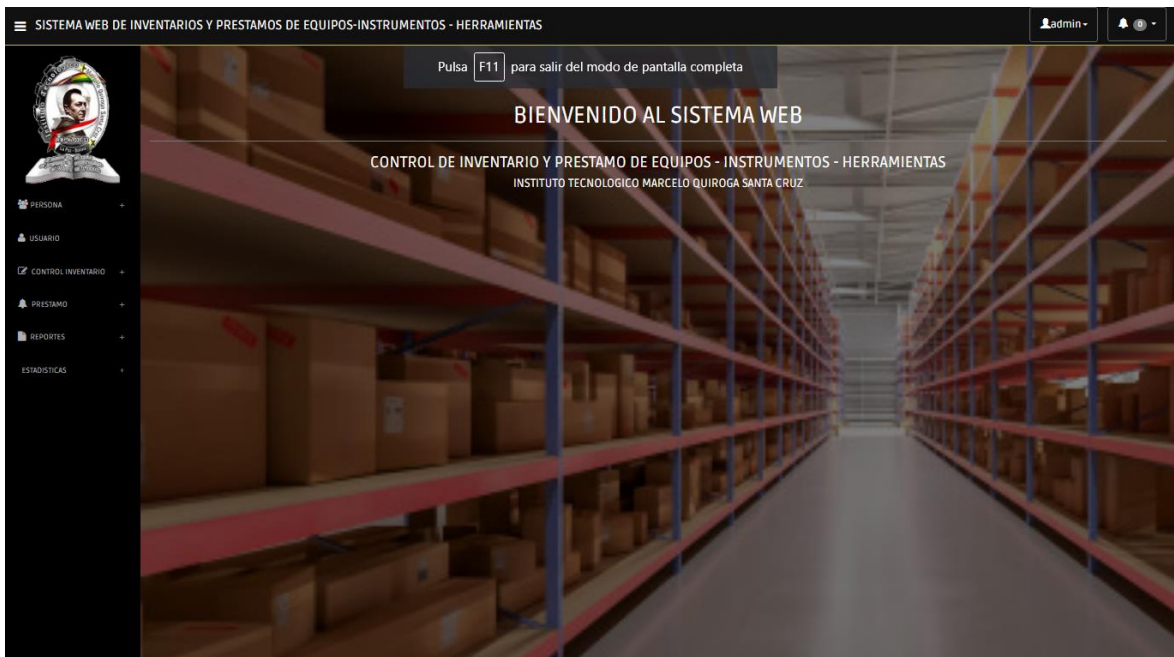
USUARIO

PASSWORD

INGRESAR →

Sistema.

Home. Es la vista de bienvenida al sistema Web, el administrador tiene acceso al menú (personas, usuarios, control de inventario, préstamo, reportes y estadísticas).







Menú. Nos ayuda a administrar los módulos dentro del sistema.

Sitio. Se cuenta con el sitio del instituto <https://instec.mqsc.edu.bo/equipo.php>

Salir. Una opción que nos permite salir del sistema.

Usuario. Debemos llenar sus datos personales del usuario y al mismo tiempo podemos crear nuevos usuarios otorgando una contraseña y un nombre de usuario, además de asignar el rol de usuario.

Listado de usuario. Nos ayuda a visualizar los datos del usuario (nombre de usuario, tipo de usuario, contraseña, rol y las opciones).

Acciones	Usuario	Persona	Contraseña	Rol
 	Daniel	Luis Daniel Villaiba Casero	\$2y\$10\$Pveem6Daj9Mw3cra8Ca9MmuMoz5Scnu96u1u1Plu5tdzgcuv20k	usuario
 	admin	admin admin admin	\$2y\$10\$00.v0n180.53PelHnRtzu0uskyxw5THsuL805oy2Qlrx1ar6u2	Administrador

Roles/cargos. Podemos asignar roles en este caso (admin, docente, estudiante)

Registrar Usuario

Seleccione una Persona:

Selec. Persona  

Seleccione Rol:

Seleccione rol  

Usuario:

Ingrese usuario 

Contraseña:

Ingrese la Contraseña 

Listado Estudiante. Al seleccionar en el menú la opción de persona tenemos el listado de estudiante y docente. La vista del listado del estudiante resalta los datos (C.I., número de matrícula, nombre de estudiante, celular e imagen), además de la opción para el registro de estudiante.

Acciones	C.I.	N° matrícula	Nombre Completo	Celular	Imagen
	4810892	938	Carlos Andres Torrez Colmena	72265454	
	4855033	941	Miguel Angel Rodriguez Calama	78020055	
	4933431	940	Benigno Gregorio Gutierrez Chura	66243412	
	4746541	939	José Luis Julián Tito	60054322	
	4810892	938	Carlos Andres Torrez Colmena	72265454	

Registrar Estudiante

Datos Personales

Num. Matriculación:

Nombre:

Apellido Paterno:

Apellido Materno:

Correo Electrónico:

Carnet de Identidad:

Celular:

Registro de estudiante. El menú de registro de estudiante permite llenar el formulario cuyos datos son: número de matrícula, nombre, apellido paterno, materno, correo electrónico, carnet de identidad, celular y la opción para agregar la fotografía del estudiante. En el formulario se tiene la opción de registro para guardarlo en la base de datos.

Listado Docente. En el menú, la opción personas en la selección del docente se visualiza los datos del docente junto a la opción para el registro de un nuevo docente.










SISTEMA WEB DE INVENTARIOS Y PRESTAMOS DE EQUIPOS-INSTRUMENTOS - HERRAMIENTAS

admin -

Lista de Docentes

Mostrar 10 registros

Acciones C.I. Nombre Completo Celular Imagen

 	4240053	Javier Flores Torrejón	73350543	
 	4478064	Eliás Sanchez vera	73240453	
 	4355033	Miguel Angel Cussi Cabrera	73020055	

Control de inventario. Se visualiza la opción para la vista del listado, baja de equipos.

Listado de equipos. Muestra los datos del equipamiento como ser: código, serie, descripción, área, tipo, fecha, imagen, etc. además de contar con las opciones para modificar, eliminar y agregar un nuevo equipo.

SISTEMA WEB DE INVENTARIOS Y PRESTAMOS DE EQUIPOS-INSTRUMENTOS - HERRAMIENTAS

















admin -

LISTADO DE EQUIPOS

AGREGAR EQUIPO

BUSQUEDA POR CODIGO: BUSQUEDA POR SERIE: BUSQUEDA POR DESCRIPCION:

Mostrar 10 registros

Opciones	Codigo	Serie	Descripcion	Ubicacion	Area	Tipo	Fecha Compra	Precio Bs.	Año Garantía	Depreciación	Imagen	Estado
  	ITMOSC-00005	50101	AMPERIMETRO ANALOGICO CA sdsds sdsds	LABORATORIO 2	LAB DE AUTOMOTISMOS ELECTRICOS II	INSTRUMENTOS DE MEDIDA	2020-12-10	340.8	2	2022-12-10		ALTA
  	ITMOSC-00004	7879988	ESCALERA RECTA disto disto	TALLER DE INSTALACION ELECTRICAS	REBOBINADO DE MAQUINAS I	HERRAMIENTAS	2020-12-10	790.9	3	2023-12-10		ALTA
  	ITMOSC-00002	670389	AMPERIMETRO ANALOGICO DC sdsds sdsds	LABORATORIO 2	LAB DE AUTOMOTISMOS ELECTRICOS I	INSTRUMENTOS DE MEDIDA	2020-12-10	670.9	5	2025-12-10		ALTA
  	ITMOSC-00001	45566	AUTOTRANSFORMADORES MONOFASICOS mtrac modelo	LABORATORIO 1	LAB DE CIRCUITOS ELECTRICOS I	EQUIPO	2020-12-10	450.5	1	2021-12-10		ALTA

Mostrando registros del 1 al 4 de un total de 4 registros

Anterior 1 Siguiente

Registro de equipo. El formulario de registro de equipo permite el llenado de datos tales como: código, serie, ubicación, área, tipo, selección de equipo, marca, modelo, costo de equipo, años de garantía, fecha y foto del equipo.

Registro de Equipo

Código: ITMQSC-0000 Serie: serie

Ubicación: Selecciona ubicación

Área: Selecciona Área

Tipo: Selecciona Tipo

Nuevo Equipo...

Equipo: Selecciona Equipo

Marca: marca Modelo: modelo

Costo De Equipo: costo 1111,00 Fecha Compra: 11/12/2020

Años De Garantía: garantía

Selecciona Fotografía

Cerrar Registrar

Baja de Equipo. En la opción de baja de equipo nos muestra el formulario para dar de baja un equipo para ellos se llena los siguientes datos: código, serie, ubicación, área, tipo, equipo, marca, modelo, fecha y razón de la baja.

BAJA DE EQUIPO

CODIGO: ITMQSC-00004 SERIE: 7879988 UBICACION: TALLER DE INSTALACION ELECTRICAS

EQUIPO: ESCALERA RECTA

AREA: REBOBINADO DE MAQUINAS I TIPO: HERRAMIENTAS

MARCA: dfdfd MODELO: dfdf

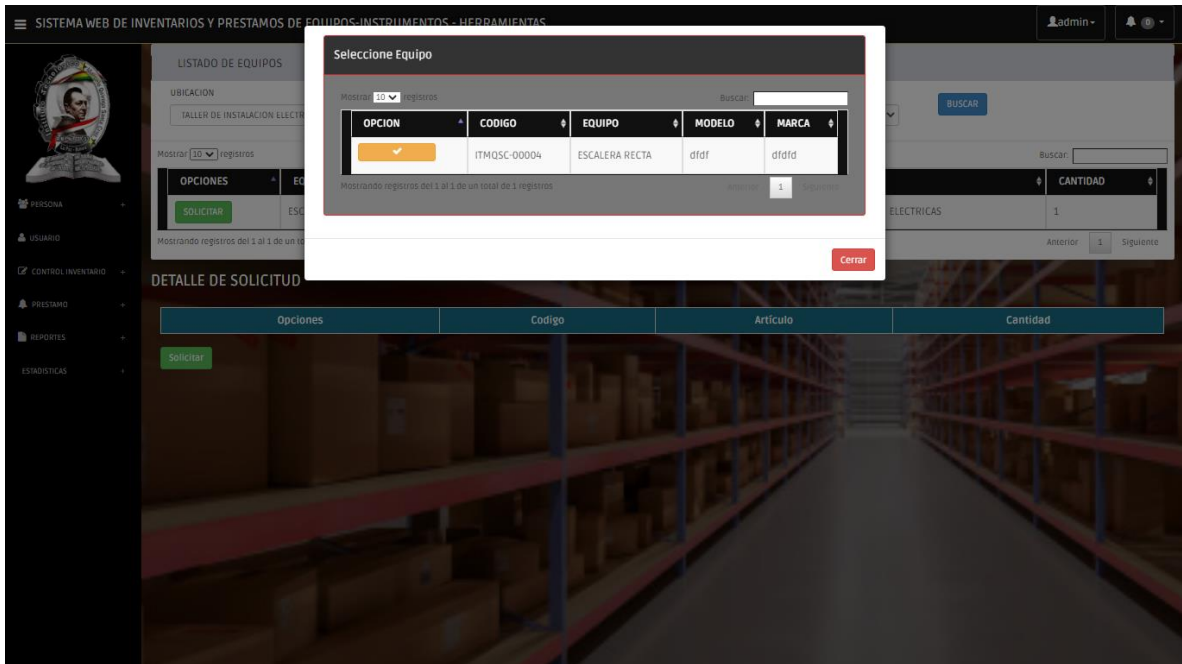
FECHA BAJA: 2020-12-11

RAZON DE LA BAJA:

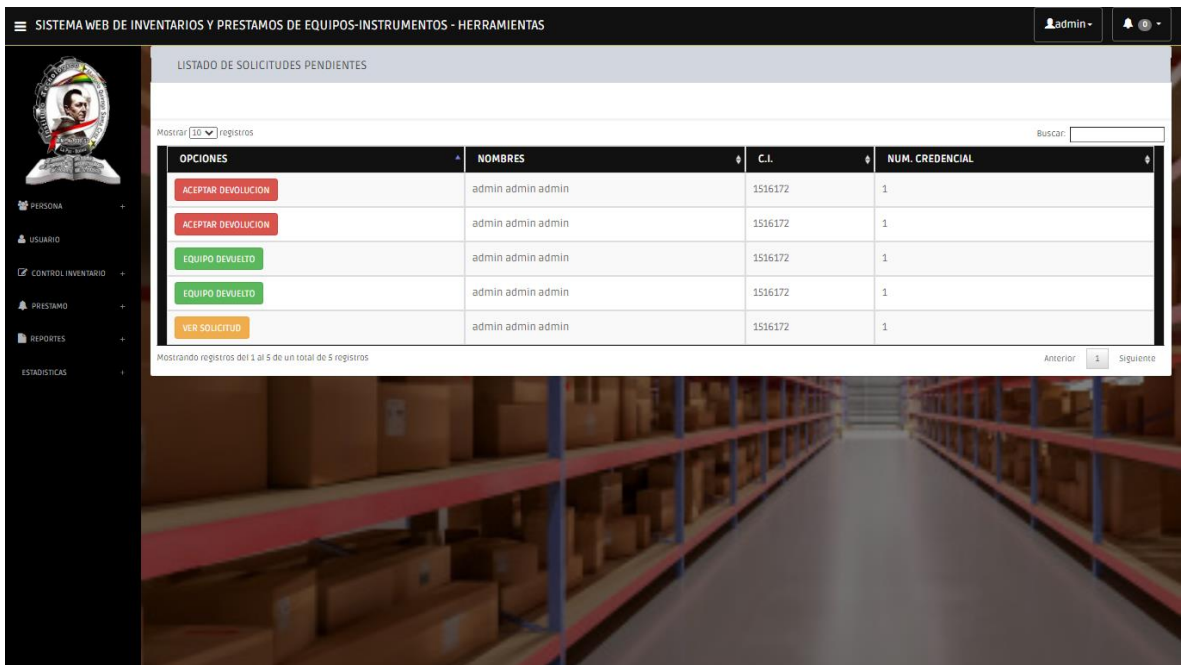
Cerrar DAR BAJA

Prestamos. La opción para préstamos de equipos nos permite ver en el menú (registro de solicitud de préstamos, listado de solicitudes, registro de aprobación y devolución).

Lista y Registro de Solicitud Préstamo Equipo. Con respecto al registro de solicitud de préstamo de equipo se puede seleccionar el equipo para el préstamo.



Lista de solicitudes pendientes. Se visualiza el cómo nombre de usuario, carnet, numero de credencial y las opciones para equipo devuelto, aceptación de préstamo.



Registro de aprobación de préstamo. Al seleccionar en el menú la opción para el registro de aprobación de préstamo se resalta los datos como ser nombre del solicitante, C.I., tipo de persona, fecha de solicitud, número de matrícula, hora de entrega, observación y las opciones para aprobar y cerrar.

DETALLE DE SOLICITUD

NOMBRE: admin admin admin C.I.: 1516172 NUM. MATRICULA: 1

TIPO DE PERSONA: Docente FECHA SOLICITUD: 2020-12-11 HORA SOLICITUD: 08:50:29

INFORMACION ENTREGA

FECHA ENTREGA: 2020-12-11 HORA ENTREGA: 08:50:47 OBSERVACION:

Mostrar 10 registros Buscar:

CODIGO	descripcion	TIPO	AREA	UBICACION
ITMQSC-00004	ESCALERA RECTA ddfdf ddf	HERRAMIENTAS	REBOBINADO DE MAQUINAS I	TALLER DE INSTALACION ELECTRICAS

Mostrando registros del 1 al 1 de un total de 1 registros

Anterior 1 Siguiente

[Aprobar](#) [Cerrar](#)

Registro de Devolución de Equipo. El formulario de registro de devolución muestra el detalle cómo ser: nombre del solicitante, C.I., tipo de persona, fecha de solicitud, número de matrícula, hora de entrega, observación, fecha de devolución y observación en la devolución.

DETALLE DE DEVOLUCION

NOMBRE: admin admin admin C.I.: 1516172 NUM. MATRICULA: 1

TIPO DE PERSONA: Docente FECHA SOLICITUD: 2020-12-11 HORA SOLICITUD: 04:32:36

INFORMACION ENTREGA REALIZADO ANTERIORMENTE

FECHA ENTREGA: 2020-12-11 HORA ENTREGA: 04:32:43 OBSERVACION:

DEVOLUCION

FECHA ENTREGA: 2020-12-11 HORA ENTREGA: 08:36:49 OBSERVACION DEVOLUCION:

Mostrar 10 registros Buscar:

CODIGO	EQUIPO	TIPO	AREA	UBICACION
ITMQSC-00005	AMPERIMETRO ANALOGICO CA sdsds sdsds	INSTRUMENTOS DE MEDIDA	LAB DE AUTOMOTISMOS ELECTRICOS II	LABORATORIO 2

Mostrando registros del 1 al 1 de un total de 1 registros

Anterior 1 Siguiente

[Devolucion](#) [Cerrar](#)

Reportes. En el menú cuando se selecciona la opción de reportes nos muestra un listado total de equipos dentro de los cuales se puede seleccionar reportes trimestrales, semestrales y anuales.

Estadísticas. En el menú se obtiene el reporte grafico de préstamos esta es una opción que puede seleccionar el administrador.

LISTA Y REPORTE TOTAL DE EQUIPOS, INSTRUMENTOS Y HERRAMIENTAS SEGÚN FECHA Y UBICACIÓN:

LISTADO TOTAL DE EQUIPOS

Fecha de baja: 11/12/2020

Fecha de baja: 11/12/2020

UBICACION: Selecciona ubicacion

Buscar: []

Codigo	Serie	Descripcion	Ubicacion	Area	Tipo	Garantia	Fecha Compra cantidad	Precio Bs	Cantidad	Estado
ITMOSC-00001	45566	AUTOTRANSFORMADORES MONOFASICOS mivac modelo	LABORATORIO 1	LAB DE CIRCUITOS ELECTRICOS I	EQUIPO	1	2020-12-10	450.5	1	
ITMOSC-00002	678989	AMPERIMETRO ANALOGICO DC ddsds sdsds	LABORATORIO 2	LAB DE AUTOMOTISMOS ELECTRICOS I	INSTRUMENTOS DE MEDIDA	5	2020-12-10	670.9	1	
ITMOSC-00004	7879988	ESCALERA RECTA ddsds dsds	TALLER DE INSTALACION ELECTRICAS	REBOBINADO DE MAQUINAS I	HERRAMIENTAS	3	2020-12-10	790.9	1	
Total Equipos:									3	

Mostrando registros del 1 al 3 de un total de 3 registros

Anterior 1 Siguiente

LISTA Y REPORTES DE EQUIPOS DADOS DE BAJA POR FECHA:

LISTADO DE EQUIPOS BAJAS POR FECHA

Fecha de baja: 11/12/2020

Fecha de baja: 11/12/2020

UBICACION: Selecciona ubicacion

Buscar: []

Codigo	Serie	Descripcion	Ubicacion	Area	Tipo	Precio Bs	Garantia	Justificacion	Estado
ITMOSC-00005	SERIE1	AMPERIMETRO ANALOGICO CA Sdsds sdsds	LABORATORIO 2	LAB DE AUTOMOTISMOS ELECTRICOS II	INSTRUMENTOS DE MEDIDA	940.8	2		BAJA

Mostrando registros del 1 al 1 de un total de 1 registros

Anterior 1 Siguiente

LISTA Y REPORTE DE EQUIPOS TRIMESTRAL:

SISTEMA WEB DE INVENTARIOS Y PRESTAMOS DE EQUIPOS-INSTRUMENTOS - HERRAMIENTAS

admin


LISTADO DE EQUIPOS TRIMESTRAL

Gestion: 2020 Trimestre: Ato

Mostrar 10 registros

Codigo	Serie	Descripcion	Ubicacion	Area	Tipo	Fecha Compra	Costo Bs.
ITMQSC-00001	45566	AUTOTRANSFORMADORES MONOFASICOS	LABORATORIO 1	LAB DE CIRCUITOS ELECTRICOS I	EQUIPO	2020-12-10	450.5
ITMQSC-00002	678989	AMPERIMETRO ANALOGICO DC	LABORATORIO 2	LAB DE AUTOMOTISMOS ELECTRICOS I	INSTRUMENTOS DE MEDIDA	2020-12-10	670.9
ITMQSC-00004	787988	ESCALERA RECTA	TALLER DE INSTALACION ELECTRICAS	REBOBINADO DE MAQUINAS I	HERRAMIENTAS	2020-12-10	790.9

Mostrando registros del 1 al 3 de un total de 3 registros



LISTA Y REPORTE DE EQUIPOS SEMESTRAL:

SISTEMA WEB DE INVENTARIOS Y PRESTAMOS DE EQUIPOS-INSTRUMENTOS - HERRAMIENTAS

admin


LISTADO DE EQUIPOS SEMESTRAL

Gestion: 2020 Semestre: Semestre II

Mostrar 10 registros

Codigo	Serie	Descripcion	Ubicacion	Area	Tipo	Fecha Compra	Costo Bs.
ITMQSC-00001	45566	AUTOTRANSFORMADORES MONOFASICOS	LABORATORIO 1	LAB DE CIRCUITOS ELECTRICOS I	EQUIPO	2020-12-10	450.5
ITMQSC-00002	678989	AMPERIMETRO ANALOGICO DC	LABORATORIO 2	LAB DE AUTOMOTISMOS ELECTRICOS I	INSTRUMENTOS DE MEDIDA	2020-12-10	670.9
ITMQSC-00004	787988	ESCALERA RECTA	TALLER DE INSTALACION ELECTRICAS	REBOBINADO DE MAQUINAS I	HERRAMIENTAS	2020-12-10	790.9

Mostrando registros del 1 al 3 de un total de 3 registros



LISTA Y REPORTE DE EQUIPOS ANUAL:

SISTEMA WEB DE INVENTARIOS Y PRESTAMOS DE EQUIPOS-INSTRUMENTOS - HERRAMIENTAS

admin

LISTADO DE EQUIPOS ANUAL

Gestion: 2020

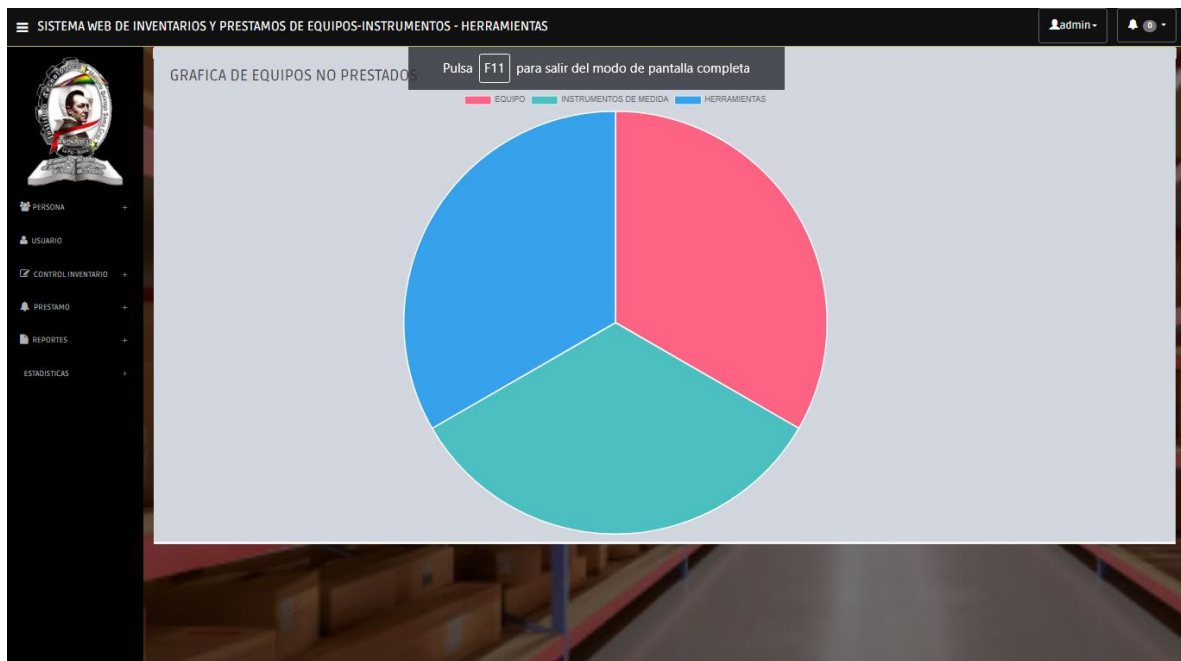
Mostrar 10 registros

Codigo	Serie	Descripcion	Ubicacion	Area	Tipo	Fecha Compra	Costo Bs.
ITMQSC-00001	45566	AUTOTRANSFORMADORES MONOFASICOS	LABORATORIO 1	LAB DE CIRCUITOS ELECTRICOS I	EQUIPO	2020-12-10	450.5
ITMQSC-00002	678989	AMPERIMETRO ANALOGICO DC	LABORATORIO 2	LAB DE AUTOMATISMOS ELECTRICOS I	INSTRUMENTOS DE MEDIDA	2020-12-10	670.9
ITMQSC-00004	7879988	ESCALERA RECTA	TALLER DE INSTALACION ELECTRICAS	REBOBINADO DE MAQUINAS I	HERRAMIENTAS	2020-12-10	790.9

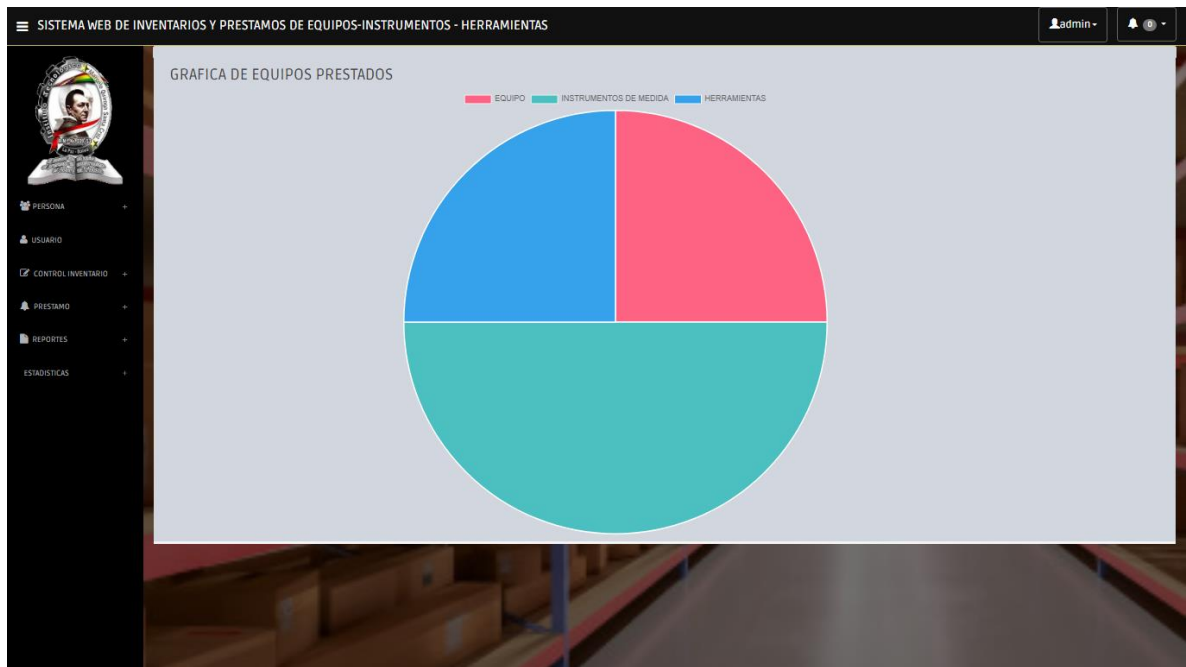
Mostrando registros del 1 al 3 de un total de 3 registros

Anterior | Siguiente

TORTA ESTADISTICA DE EQUIPOS, HERRAMIENTAS Y INSTRUMENTOS NO PRESTADOS:



TORTA ESTADISTICA DE EQUIPOS, HERRAMIENTAS Y INSTRUMENTOS PRESTADOS:



ANEXO E

MANUAL TÉCNICO

1. Instalación de XAMPP.

Descargar las versiones con PHP 5.5, 5.6 o 7 se pueden descargar gratuitamente desde la página del proyecto [Apache Friends](#).

2. Iniciar el asistente de instalación

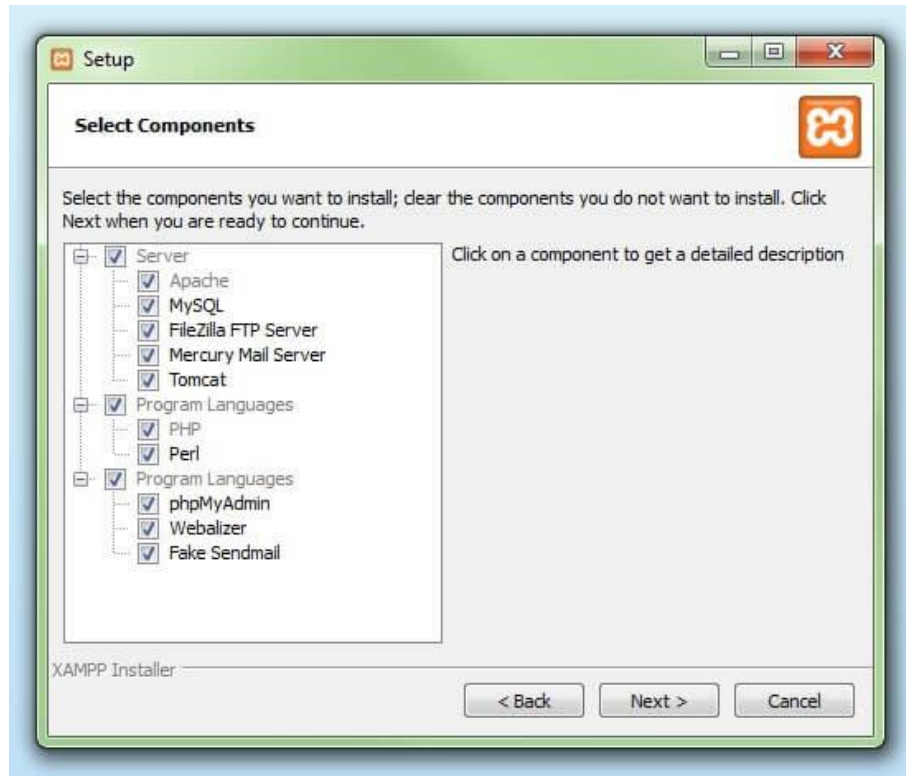
Una vez superados estos pasos, aparece la pantalla de inicio del asistente para instalar XAMPP. Para ajustar las configuraciones de la instalación se hace clic en “Next”.



Con la aparición de la pantalla de inicio del asistente da comienzo la instalación de XAMPP.

3. Selección de los componentes del software

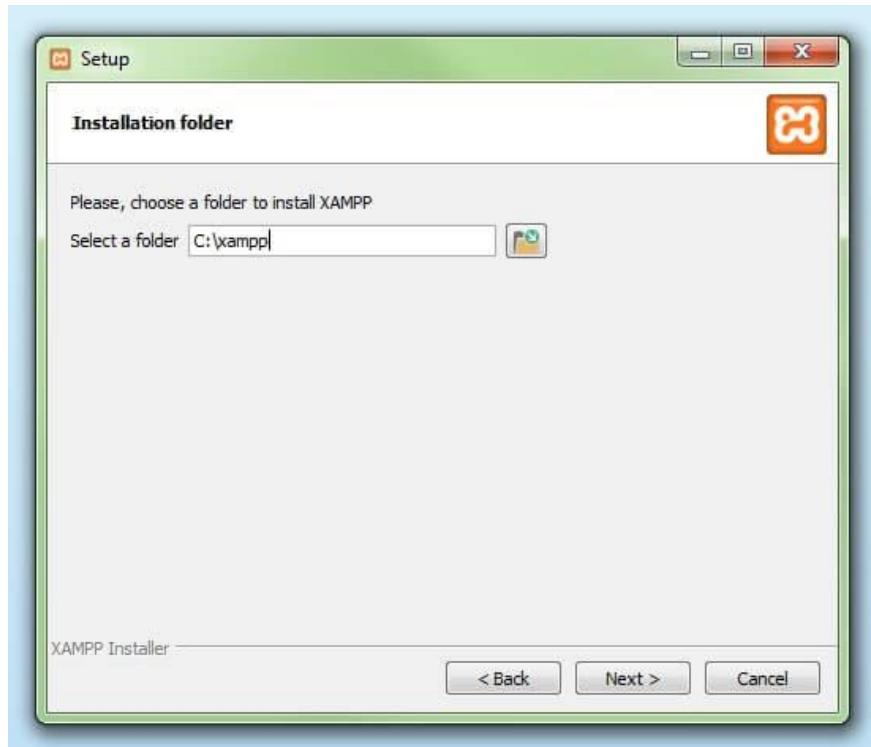
En la rúbrica “Select components” se pueden excluir de la instalación componentes aislados del paquete de software de XAMPP. Se recomienda la configuración estándar para un servidor de prueba local, con la cual se instalan todos los componentes disponibles. Confirma la selección haciendo clic en “Next”.



En el cuadro de diálogo “Select Components” se pueden seleccionar o deseleccionar los componentes que se instalarán.

4. Selección del directorio para la instalación

En este paso se escoge el directorio donde se instalará el paquete. Si se ha escogido la configuración estándar se creará una carpeta con el nombre XAMPP en C:\.



En un siguiente paso, se selecciona el directorio donde se instalarán los archivos

5. Iniciar el proceso de instalación

El asistente extrae los componentes seleccionados y los guarda en el directorio escogido en un proceso que puede durar algunos minutos. El avance de la instalación se muestra como una barra de carga de color verde.

Una vez instalado Xampp procedemos a la configuración de laravel.

5.1 Instalando PHP

Para instalar PHP en Windows tenemos varias herramientas, cada una con distintos beneficios.

Es tan simple como entrar a [su página](#), descargar el paquete [WAMP](#) (Windows, Apache, MySQL, PHP) y ejecutar el instalador.

Elegimos la ruta donde se instalará Laragon, y luego el instalador asociará PHP a nuestras variables de entorno (PATH) para que podamos usarlo desde consola.

Una de las ventajas de Laragon vs [Xampp](#) o [Wamp](#) es que instalar proyectos Laravel es muy sencillo, ya que tiene una opción específica para eso, que genera un nuevo proyecto Laravel con todos los paquetes instalados, el archivo .env creado, un Virtual Host para **Apache** o **nginx** configurado con el nombre del proyecto para que podamos acceder rápidamente en nuestro navegador (<http://ejemplo.dev>) y hasta puede generar una base de datos para el proyecto, todo desde un fácil acceso rápido en la barra de tareas.

Una vez instalado alguno de estos entornos, ya sea Laragon, Xampp, Wamp o cualquier otro, podremos comprobar que PHP se instaló correctamente comprobando por consola su versión:

```
php -v
```

Si PHP se instaló con éxito, debería mostrarnos algo similar a esto:

```
PHP 7.0.11 (cli) ( NTS )
```

```
Copyright (c) 1997-2016 The PHP Group  
Zend Engine v3.0.0, Copyright (c) 1998-2016 Zend Technologies  
with Zend OPcache v7.0.11, Copyright (c) 1999-2016, by Zend  
Technologies
```

Es importante instalar las versiones de PHP 7+, ya que las versiones 5 de PHP han quedado discontinuadas y muchos paquetes requieren de la última versión para ejecutarse.

5.2 Instalando Composer

Composer es el manejador de dependencias de PHP, es decir, se encarga de instalar las librerías que nuestros proyectos necesitan y de mantenerlas actualizadas, por lo que podemos reutilizar código de otros programadores en nuestras aplicaciones y ahorrarnos reescribir siempre las mismas funcionalidades.

PHP tuvo un antes y un después de la aparición de Composer, ya que permitió que la comunidad desarrollara paquetes especializados y que estos fueran probados por miles de aplicaciones, ahorrando tiempo a futuros programadores y mejorando la estabilidad de nuestros sistemas.

Para instalar Composer en Windows, debemos descargarlo desde su página:

<https://getcomposer.org/Composer-Setup.exe>

Luego, ejecutamos el instalador y seguimos sus pasos:

Una vez terminado, podemos abrir una consola de Windows y ejecutar composer. Si todo salió bien, nos mostrará una lista de comandos para ejecutar.

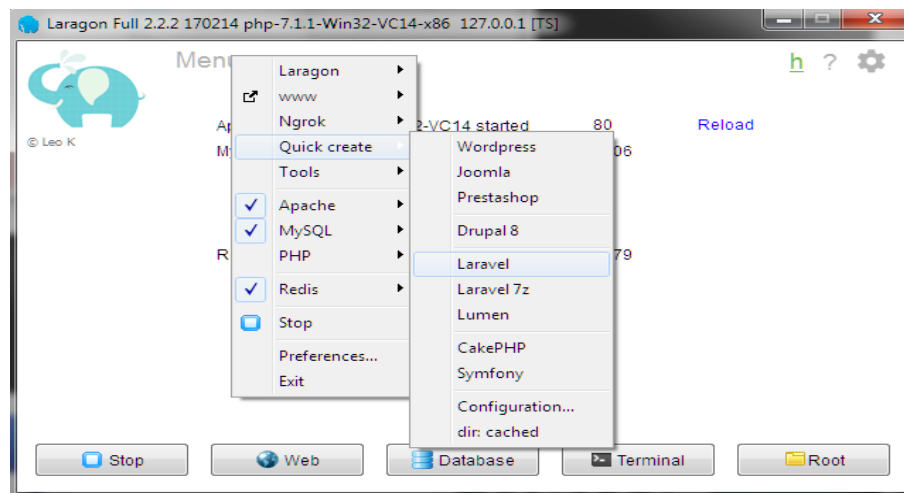
```
$ composer

Composer version 1.4.1 2017-03-10 09:29:45

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display this help message
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
  --ansi                    Force ANSI output
  --no-ansi                 Disable ANSI output
  -n, --no-interaction      Do not ask any interactive question
  --profile                 Display timing and memory usage information
  --no-plugins              Whether to disable plugins.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  -v|vv|vvv, --verbose     Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  about             Short information about Composer.
  archive           Create an archive of this composer package.
  browse            Opens the package's repository URL or homepage in your browser.
  clear-cache       Clears composer's internal package cache.
  clearcache        Clears composer's internal package cache.
  config            Set config options.
  create-project    Create new project from a package into given directory.
  depends           Shows which packages cause the given package to be installed.
  diagnose          Diagnoses the system to identify common errors.
  dump-autoload     Dumps the autoloader.
```



Estos comandos generarán una carpeta nuevo-proyecto que tendrá la instalación básica del framework. Si lo creamos con composer, tendremos que copiar el archivo .env.example a .env y generar una clave nueva para el proyecto.