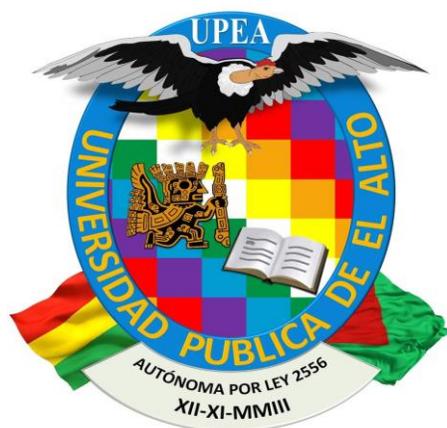


# UNIVERSIDAD PÚBLICA DE EL ALTO

## CARRERA INGENIERÍA DE SISTEMAS



## PROYECTO DE GRADO

### “SISTEMA DE EDUCACION VIRTUAL (E-LEARNING) COMO HERRAMIENTA DE ENSEÑANAZA”

**CASO: TECHSBOL**

**Para Optar al Título de Licenciatura en Ingeniería de Sistemas**

**MENCIÓN: INFORMÁTICA Y COMUNICACIONES**

**Postulante: Rudy David Castillo**

**Tutor Metodológico: Ing. Enrique Flores Baltazar**

**Tutor Revisor: Lic. Katya Maricela Pérez Martínez**

**Tutor Especialista: Lic. Roberto David Mayta Aliga**

**EL ALTO – BOLIVIA**

**2020**

## **DEDICATORIA**

*El presente proyecto de grado va dedicado en primer lugar a Dios, por haberme permitido llegar hasta este punto, haberme dado salud y por iluminar mi mente en cada paso que doy, A mis padres Hilaria y Juan Carlos por ser el pilar fundamental en todo lo que soy, en mi educación, tanto académica, como de la vida, por su incondicional apoyo, a mis compañeros de carrera por sus valiosas sugerencias; si no fuera por ellos, no hubiera llegado hasta este punto de mi vida, agradezco por su ayuda, apoyo desprendimiento y desprendimiento.*

## **AGRADECIMIENTOS**

Primero y antes que nada quisiera agradecer a Dios por la oportunidad que me brindo para realizar este proyecto y aprender de él.

A mis padres ya que me brindaron apoyo incondicional en mis estudios y a lo largo de mi vida.

Un agradecimiento a mi tutor metodológico, el Ing. Enrique Flores Baltazar, por haberme brindado su apoyo y conocimiento, su experiencia y consejos sin los cuales no hubiera sido posible la culminación del trabajo.

De la misma manera un agradecimiento muy sincero a mi tutor especialista Lic. Roberto David Mayta Aliga, por guiarme en las diferentes etapas de mi proyecto y por su asesoramiento.

Un agradecimiento muy sincero a mi tutor revisor Lic. Katya Maricela Pérez Martínez por su colaboración y consejos, por sus observaciones y seguimiento constante, pero sobre todo por su paciencia y tiempo necesario para poder concluir este trabajo.

## **RESUMEN**

El presente Proyecto de Grado consiste en implementar un SISTEMA DE EDUCACION VIRTUAL (E-LEARNING) COMO HERRAMIENTA DE ENSEÑANAZA CASO: TECHSBOL, surge del gran inconveniente que sufre TECHSBOL, que no disponen de oportunidades para publicar y promover su producción intelectual y capacitación en el mercado. El proyecto se centró en la venta del producto intelectual (CURSOS), desde el registro de los cursos, hasta reportes de ingresos a TECHSBOL de la venta de los mismos, administración de cursos, creación de carrito de compras, utilización de la pasarela de pagos incluyéndose Paypal y autenticación para controlar acceso al sistema. En la parte introductoria se demuestra los antecedentes y actividades que realiza la institución, también se muestra el análisis de los problemas y los objetivos propuestos. Para el desarrollo del presente proyecto se aplicó la metodología de desarrollo ágil SCRUM apoyándose junto a la metodología KANBAN y la metodología de desarrollo UWE para el modelado del diseño. El Sistema es un producto de calidad de acuerdo a la métrica de calidad Web-Site QEM. Para funcionamiento pleno del proyecto se toma como tarea final implementar la seguridad. Finalmente se concluye que los objetivos planteados fueron alcanzados y que el Sistema cumple con los requerimientos establecidos por el cliente.

Palabras clave: Metodología ágil Kanban, UWE, Web-Site QEM, Sistema Web, Linux.

## **SUMMARY**

This Degree Project consists of implementing a VIRTUAL EDUCATION SYSTEM (E-LEARNING) AS A TEACHING TOOL. CASE: TECHSBOL, arises from the great inconvenience suffered by TECHSBOL, that they do not have opportunities to publish and promote their intellectual production and training in the market. The project focused on the sale of the intellectual product (COURSES), from the registration of the courses, to reports of income to TECHSBOL from the sale of the same, administration of courses, creation of a shopping cart, use of the payment gateway including Paypal and authentication to control access to the system. In the introductory part, the background and activities carried out by the institution are shown, the analysis of the problems and the proposed objectives is also shown. For the development of this project, the SCRUM agile development methodology was applied, supported by the KANBAN methodology and the UWE development methodology for modeling the design. The System is a quality product according to the Web-Site QEM quality metric. For the full operation of the project, the final task is to implement security. Finally, it is concluded that the proposed objectives were achieved and that the System meets the requirements established by the client.

**Keywords:** Agile Kanban Methodology, UWE, Web-Site QEM, Web System, Linux.

## INDICE

1. MARCO PRELIMINAR .....	2
1.1. Introducción.....	2
1.2. ANTECEDENTES .....	3
1.2.1. antecedentes institucionales .....	3
1.2.2. Internacional.....	3
1.2.3. Nacional .....	4
1.3. PLANTEAMIENTO DEL PROBLEMA .....	5
1.3.1. Problema principal .....	5
1.3.2. Problemas secundarios.....	5
1.4. OBJETIVOS .....	5
1.4.1. Objetivo General .....	5
1.4.2. Objetivos Específicos .....	6
1.5. JUSTIFICACIÓN .....	6
1.5.1. Técnica.....	6
1.5.2. Económica .....	7
1.5.3. Social .....	7
1.6. METODOLOGÍA.....	7
1.6.1. Método científico .....	7
1.6.2. Método de ingeniería .....	7
1.7. HERRAMIENTAS.....	9
1.8. LÍMITES Y ALCANCES.....	10
1.8.1. Limites.....	10
1.8.2. Alcances.....	10
1.9. APORTES .....	11
2. MARCO TEÓRICO.....	13

2.1. INTRODUCCIÓN .....	13
2.2. SISTEMA .....	13
2.3. INFORMACIÓN.....	13
2.4. SISTEMAS DE INFORMACIÓN.....	13
2.5. EDUCACIÓN.....	14
2.6. EDUCACIÓN A DISTANCIA .....	14
2.7. E-LEARNING .....	15
2.8. TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN .....	15
2.9. INGENIERÍA DE SOFTWARE .....	16
2.10. INGENIERÍA WEB .....	16
2.11. SCRUM .....	17
2.11.1. Características .....	17
2.11.2. Principios básicos .....	17
2.11.3. Prácticas de scrum.....	18
2.11.4. Roles Scrum.....	19
2.11.5. Eventos de Scrum.....	20
2.11.6. Fases de scrum.....	22
2.12. KANBAN .....	25
2.12.1. Características .....	26
2.12.2. Roles Kanban.....	26
2.12.3. Tablero Kanban.....	27
2.12.4. Fases Kanban .....	28
2.13. Combinación de scrum y Kanban SCRUMBAN .....	30
2.13.1. Diferencias básicas entre scrum y Kanban .....	30
2.13.2. Beneficios de la integración .....	30

2.14. MÉTRICA DE CALIDAD.....	31
2.14.1. Fases de Web Qem .....	31
2.14.2. Definición de Las Metas de Evaluación y Selección del Perfil de Usuario .....	31
2.14.3. Definición de los requerimientos de calidad/costo .....	32
2.14.4. Definición de Criterios Elementales y Procedimientos de Medición .....	33
2.14.5. Características de WebQem .....	34
2.14.6. Funcionalidad.....	34
2.14.7. Confiabilidad .....	35
2.14.8. Usabilidad .....	36
2.14.9. Mantenibilidad .....	37
2.14.10. Portabilidad .....	38
2.15. SEGURIDAD.....	38
2.15.1. Seguridad por niveles .....	39
2.15.2. Seguridad a nivel de aplicación.....	39
2.15.3. Seguridad a nivel de la base de datos .....	39
2.15.4. Seguridad a nivel del servidor .....	40
2.16. HARDWARE .....	41
2.16.1. EQUIPO DE DESARROLLO A USAR .....	41
2.17. SOFTWARE.....	41
2.17.1. Sistema operativo .....	41
2.17.2. IDE (Integrated Development Environment) de desarrollo.....	42
2.18. LENGUAJE UNIFICADO DE MODELADO (U.M.L.) .....	43
2.19. patrones de diseño.....	44
2.19.1. modelo vista controlador .....	45

2.20. tecnologías web .....	46
2.20.1. Php.....	46
2.20.2. Framework Codeigniter .....	46
2.20.3. JavaScript .....	47
2.20.4. JQuery.....	47
2.20.5. Bootstrap.....	47
2.20.6. Gestor De Base de Datos MySQL .....	48
3. MARCO APLICATIVO .....	50
3.1. INTRODUCCIÓN .....	50
3.2. ESQUEMA DEL SISTEMA.....	50
3.3. FASE PREGAME .....	51
3.3.1. Concepción y exploración .....	51
3.3.2. Roles SCRUMBAN .....	52
3.3.3. Arquitectura del software.....	52
3.3.4. Obtención de requerimientos .....	53
3.3.5. Producto-backlog .....	54
3.3.6. Inicialización.....	56
3.4. FASE DEVELOPMENT .....	66
3.4.1. Desarrollo de los Sprint.....	66
3.5. PRIMER SPRING: MODULO DE REGISTRO Y LOGIN DE USUARIOS .....	67
3.5.1. Etapa de análisis.....	67
3.5.2. etapa de diseño.....	69
3.5.3. Etapa de implementación.....	70
3.5.4. Resultados .....	72
3.6. Segundo spring: módulo de administración de cuentas .....	74

3.6.1. Etapa de análisis.....	74
3.6.2. Etapa de implementación.....	76
3.6.3. Resultado.....	78
3.7. Tercer Spring: modulo carrito de compras.....	79
3.7.1. Etapa de análisis.....	79
3.7.2. Etapa de diseño.....	80
3.7.3. Etapa de implementación.....	82
3.7.4. Resultado.....	83
3.8. Cuarto Sprint: Módulo De Administración De Catálogo Y Cursos.....	84
3.8.1. Etapa de análisis.....	84
3.8.2. Etapa de diseño.....	86
3.8.3. Etapa de implementación.....	87
3.8.4. Resultados.....	88
3.9. Postgame y fase de estabilización, pruebas y reparaciones.....	94
3.9.1. PRUEBAS Y REPARACIONES.....	95
3.9.2. Pruebas de caja negra.....	95
4. PRUEBAS DE EVALUACIÓN Y RESULTADO.....	98
4.1. INTRODUCCIÓN.....	98
4.2. CALIDAD DE SOFTWARE.....	98
4.2.1. Definiendo metas de evaluación.....	98
4.2.2. Especificaciones requerimientos de calidad.....	99
4.2.3. Especificación de características de calidad.....	99
4.3. SEGURIDAD.....	104
4.3.1. TIPOS DE SEGURIDAD.....	105
4.4. COSTO.....	106

4.4.1. Técnica.....	106
4.4.2. Proceso .....	107
4.4.3. Verificación y Validación .....	108
4.4.4. Consideraciones .....	108
5. CONCLUSIONES Y RECOMENDACIONES .....	110
5.1. CONCLUSIONES.....	110
5.2. RECOMENDACIONES .....	110
Bibliografía .....	112

## ÍNDICE DE TABLAS

Tabla 1.	Prácticas de la gestión ágil .....	18
Tabla 2.	Roles scrum .....	19
Tabla 3.	Fases Pregame.....	23
Tabla 4.	Característica Kanban.....	26
Tabla 5.	Elementos tablero Kanban.....	27
Tabla 6.	Tabla de ponderación en porcentajes .....	34
Tabla 7.	Equipo desarrollo .....	41
Tabla 8.	Tabla Scrumban.....	51
Tabla 9.	Obtención de requisitos .....	51
Tabla 10.	Fases de trabajo y responsables .....	52
Tabla 11.	Lista de requerimientos .....	53
Tabla 12.	Producto backlog .....	54
Tabla 13.	Aplicación .....	58
Tabla 14.	Categoría.....	58
Tabla 15.	Identificador de inicio de sesión.....	58
Tabla 16.	Comentario .....	59
Tabla 17.	Curso .....	59
Tabla 18.	Inscripcion .....	60
Tabla 19.	Lección .....	61
Tabla 20.	Mensajes .....	61
Tabla 21.	Ilo del mensaje.....	62
Tabla 22.	Pagos a Instructores.....	62
Tabla 23.	Pagos de cursos .....	63
Tabla 24.	Cuestionarios.....	63

Tabla 25.	Calificación del curso.....	64
Tabla 26.	Roles .....	64
Tabla 27.	Sección de cursos .....	64
Tabla 28.	Etiquetas.....	65
Tabla 29.	Usuarios .....	65
Tabla 30.	Versiones de desarrollo .....	66
Tabla 31.	Primer sprint modulo de registro y login .....	67
Tabla 32.	especificación del caso de uso inicio de sesión y registro .....	68
Tabla 33.	Administración de cuentas.....	74
Tabla 34.	Carrito de compras .....	79
Tabla 35.	Administración de catalogo de cursos .....	84
Tabla 36.	Comprensibilidad global del sitio .....	100
Tabla 37.	Aspectos de interfaces y estéticos.....	100
Tabla 38.	Misceláneas.....	101
Tabla 39.	TOTAL USABILIDAD.....	101
Tabla 40.	Aspectos de búsqueda y recuperación.....	102
Tabla 41.	Aspectos de dominio orientados al usuario .....	102
Tabla 42.	TOTAL, FUNCIONALIDAD .....	102
Tabla 43.	Evaluación de confiabilidad .....	103
Tabla 44.	Evaluación de desempeño .....	104

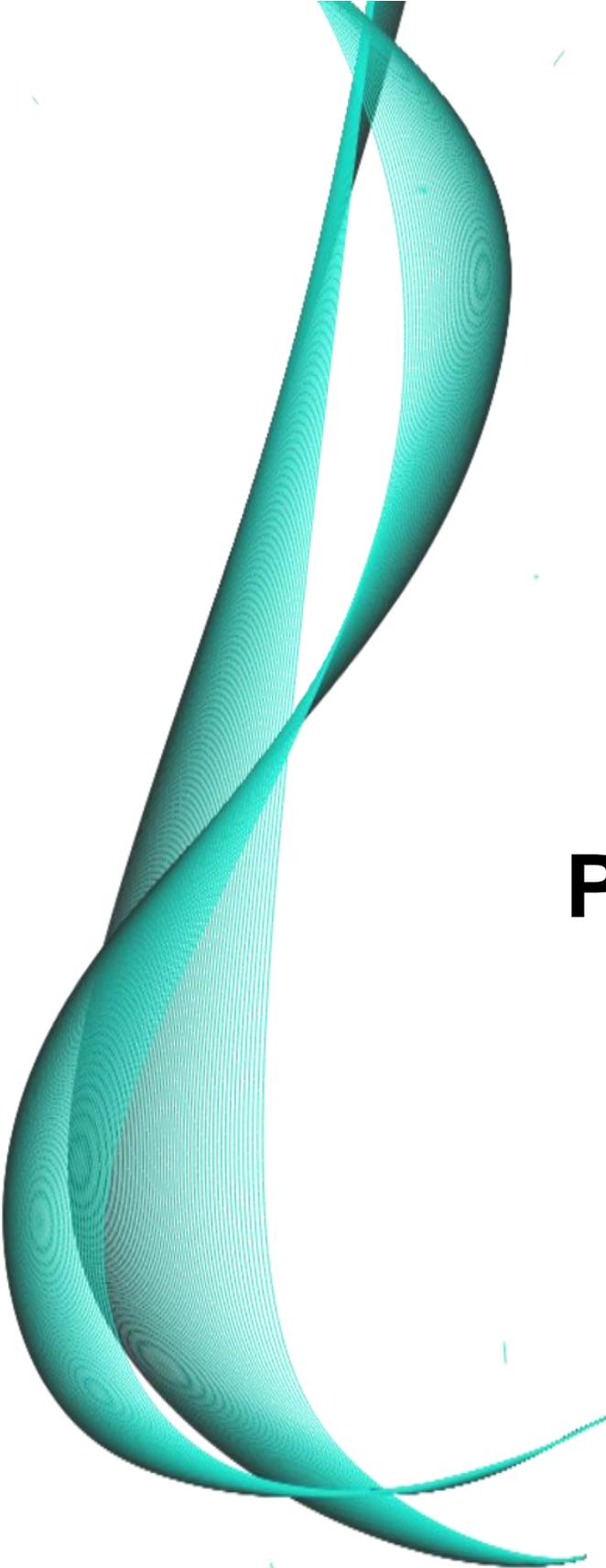
## ÍNDICE DE FIGURAS

Figura 1.	Fases scrum .....	23
Figura 2.	Tablero Kanban .....	27
Figura 3.	Panel Kanban .....	29
Figura 4.	Historia UML.....	44
Figura 5.	Esquema de sistema .....	50
Figura 6.	Esquema de arquitectura.....	53
Figura 7.	Base de datos.....	57
Figura 8.	diagrama caso de uso de inicio de sesión y registro .....	68
Figura 9.	diagrama de inicio de sesión y registro.....	69
Figura 10.	diagrama de inicio de sesión y registro .....	70
Figura 11.	diagrama de inicio de sesión y registro .....	71
Figura 12.	diagrama de inicio de sesión y registro .....	72
Figura 13.	interface de inicio de sesión .....	73
Figura 14.	interface de recuperación de contraseña .....	73
Figura 15.	interface de registro.....	74
Figura 16.	diagrama de estados del módulo de administración de usuario .....	76
Figura 17.	diagrama de estados del módulo de administración de usuario .....	77
Figura 18.	diagrama de estados del módulo de administración de usuario .....	78
Figura 19.	plataforma de administración .....	78
Figura 20.	diagrama de casos de uso del módulo del carrito .....	80
Figura 21.	diagrama de clases del módulo del carrito .....	81
Figura 22.	diagrama de estado del módulo del carrito.....	81
Figura 23.	diagrama de secuencia del módulo del carrito .....	82
Figura 24.	diagrama de secuencia del módulo del carrito .....	83

Figura 25.	diagrama de secuencia del módulo del carrito .....	84
Figura 26.	diagrama de caso de uso (include) del módulo del carrito .....	85
Figura 27.	diagrama de clases del módulo del carrito .....	86
Figura 28.	diagrama de estado del módulo del carrito.....	87
Figura 29.	diagrama de componentes del módulo del carrito.....	88
Figura 30.	Pantalla bienvenida .....	89
Figura 31.	Pantalla de cursos.....	89
Figura 32.	Pantalla vista previa curso.....	90
Figura 33.	Pantalla de login.....	91
Figura 34.	Pantalla de cursos adquiridos .....	92
Figura 35.	Pantalla avance de curso .....	93
Figura 36.	Pantalla de cursos adquiridos .....	94
Figura 37.	Vulnerabilidades de bibliotecas .....	95
Figura 38.	Prueba general de sistema.....	96
Figura 39.	estimación .....	107

## ÍNDICE ANEXOS

Anexo A manual usuario .....	116
------------------------------	-----



# **CAPÍTULO I**

## **MARCO**

### **PRELIMINAR**

## **1. MARCO PRELIMINAR**

### **1.1. Introducción**

E-learning (simplificación del anglicismo Electronic Learning) podría definirse como el proceso de enseñanza que se lleva a cabo o se desarrolla en internet y mediante la utilización de medios electrónicos. También se le denomina entre otras acepciones formación on-line, enseñanza virtual, tele formación o aprendizaje electrónico. En este sentido hay que hacer una distinción importante entre lo que se entiende por e-learning (el conocimiento se adquiere a través de internet) y el b-Learning o blended Learning (el conocimiento se adquiere combinando el aprendizaje a distancia y el aprendizaje presencial). (Moll, 2014)

Según (O'Reilly, 2005), Web 2.0 es la red como plataforma, extendiéndose a todos los dispositivos conectados: las aplicaciones Web 2.0 son aquellas que utilizan lo mejor de las ventajas intrínsecas de dicha plataforma: distribuyendo software como un servicio constantemente actualizado que es mejor cuanto más gente lo utiliza, consumiendo y remezclado datos de múltiples fuentes incluyendo usuarios individuales, mientras proporcionan sus propios datos y servicios de manera que permiten a otros Re mezclarlos, creando efectos de red a través de una "arquitectura de participación". (vazquez, 2014)

Por ello en el presente proyecto de grado se desarrollará el sistema de educación virtual(E-Learning) como herramienta de enseñanza en un entorno de LMS (Learning Management Systems, Sistemas de Gestión del Aprendizaje), que agrupará funcionalidades de gestión y distribución de contenido formativos, con herramientas de comunicación y utilidades para el seguimiento. Para este fin usaremos la metodología Kanban gestiona de manera general el desarrollo de las tareas gracias a su visualización del trabajo por fases, permitiendo evitar la sobrecarga en el desarrollo, a la vez que mide el tiempo estimado en el que se debería completar cada tarea. La institución TECHSBOL se ha incursionado en impartir cursos de nivelación y capacitación, impartiendo materias de matemáticas, diseño gráfico, programación básica, creación de páginas web, creación de aplicaciones Android, robótica para

niños, electrónica y otros, por esta razón dicha empresa requiere implementar SISTEMA DE EDUCACION VIRTUAL (E-LEARNING) COMO HERRAMIENTA DE ENSEÑANAZA.

El presente proyecto de grado tiene como fin el modelar, diseñar y desarrollar un o LMS (Learning Management Systems, Sistemas de Gestión del Aprendizaje) utilizando la metodología agile Kanban para la empresa TECHSBOL, el cual integrara el control de seguimiento académico, enseñanza virtual, control de docentes, conformidad con SCORM (Sharable Content Object Reference Model), Aprendizaje sincronizado y Aprendizaje asíncrono.

## **1.2. ANTECEDENTES**

### **1.2.1. antecedentes institucionales**

TECHSBOL es una empresa multiservicios con un lineamiento orientado a las TIC`s. Situado en Calle 22 Pedro Fernández 1514-1524, El Alto.

Inicio actividades a inicios del 2018, con la filosofía de ser una empresa de vanguardia en la tecnología y sus ramas de esta manera opto por dar cursos aleatorios de programación, robótica, electrónica, diseño grafico y otros. Siendo estos cursos de capacitación aun sin un certificado por curso ya q el mismo se encuentra en tramites con las instancias correspondientes.

Inicialmente la institución cuenta con un administrador o SEO, con cuatro encargados de área de desarrollo, marketing, servicios y área legal.

### **1.2.2. Internacional**

- (Espinoza Manchego, 2014) “ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE E-LEARNING PARA LA TIENDA POR DEPARTAMENTOS RIPLEY S.A.” ESCUELA PROFESIONAL DE INGENIERÍA DE COMPUTACIÓN Y SISTEMAS. La tienda por departamentos Ripley S.A., es una empresa dedicada a la venta minorista de productos de almacén, la cual presenta elevados costos de capacitación de los empleados en las áreas críticas para el desarrollo de la

empresa, que por su elevado costo y disponibilidad presenta un gran problema para la organización. Por lo tanto, la solución propuesta es la educación a distancia o E-learning.

- (Javier López Moratalla, 2003-2004) <e-Aula>: Desarrollo de un sistema e-learning basado en estándares IMS” Facultad de Informática Universidad Complutense de Madrid, Como se ha mencionado en la introducción, el proyecto de investigación tiene como uno de sus objetivos prioritarios el desarrollo de un entorno que permita evaluar diferentes propuestas de estandarización. Más concretamente, se pretende analizar el potencial de los estándares educativos en la construcción de entornos de aprendizaje personalizado, mediante un conjunto de sistemas que implementen la infraestructura básica (e.g. creación, acceso y reutilización de los contenidos). Para lograr estos objetivos se han desarrollado distintos entornos y varios cursos orientados a estudiantes universitarios. Se ha realizado una evaluación formativa tanto de los sistemas como de los cursos con estudiantes de informática de doctorado y de licenciatura.

### **1.2.3. Nacional**

- (Velasco, 2009) “aulas virtuales en la educación boliviana un nuevo paradigma”. El proyecto plantea el diseño implementación y gestión de aulas virtuales, que involucren a su vez, conferencias, consultorio y cursos en línea en la actitud de la web 2.0 “crear, compartir y coproducir”.
- (ZARATE, 2014 ) “TUTOR VIRTUAL B-LEARNING EN TIC CON NORMAS DE CALIDA” Capacitar a la población de Tumupasa en Tecnologías de Información y Comunicación bajo normas de calidad por medio de plataformas educativas virtuales introduciendo un Agente que apoye el proceso de aprendizaje en temas transversales de TIC. Para apoyar en la disminución del analfabetismo tecnológico a los habitantes de la comunidad de Tumupasa.

### **1.3. PLANTEAMIENTO DEL PROBLEMA**

TECHSBOL pasa por momentos de crisis desde el inicio de la cuarentena y el conflicto ocurrido en la anterior gestión. La institución se encuentra en dificultades a la hora de impartir sus cursos de manera regular y/o apertura de nuevos cursos.

El problema que aflige a TECHSBOL es de manera económica y social. La institución no dispone de oportunidades para la difusión de sus cursos y/o apertura de nuevos cursos de manera local.

#### **1.3.1. Problema principal**

¿Cómo una plataforma e-learning ayudaría a impartir los cursos de manera regular y/o crear nuevos cursos para TECHSBOL?

#### **1.3.2. Problemas secundarios**

- TECHSBOL no dispone de medios para la difusión de sus cursos, lo q ocasiona una disminución de ingresos.
- No existe una manera de promocionar los cursos nuevos o existentes, de manera que los usuarios no se encuentran informados.
- Los precios de los cursos solo se informan al momento de visitar la institución, lo que ocasiona la desconfianza del usuario.
- Inexistencia de reportes por ingresos de los cursos impartidos por TECHSBOL, lo que ocasiona un control inadecuado de los ingresos a la institución.

### **1.4. OBJETIVOS**

#### **1.4.1. Objetivo General**

Desarrollar un sistema de educación virtual (e-learning) como herramienta de enseñanza para TECHSBOL.

### **1.4.2. Objetivos Específicos**

- Implementar un medio de difusión y venta de cursos, para el incremento de ingresos a TECHSBOL.
- Brindar un sitio/espacio digital de visualización de cursos que muestren los precios y descripción.
- Expandir el mercado de los cursos de TECHSBOL de manera digital y on-line de manera que los cursos siempre estén disponibles
- Implementar sección de control de usuarios y cursos
- Generar reporte de los ingresos por los cursos de TECHSBOL.

### **1.5. JUSTIFICACIÓN**

En Bolivia, la modalidad de Educación Virtual está ingresando poco a poco, siendo este un impulsor para TECHSBOL hace que requiera implementar la educación virtual en su institución.

A continuación, se detalla la justificación económica, social y tecnológica para llevar adelante el proyecto.

#### **1.5.1. Técnica**

Los contenidos virtuales se justifican técnicamente, ya que se hace uso de la tecnología Web, se utilizará el lenguaje de programación PHP junto a MariaDB como base de datos, apoyándonos en diferentes módulos que se lleguen a desarrollar para el fácil aprendizaje. Herramientas a utilizar:

- A. Computador.
- B. Internet.
- C. Lenguajes de programación.
- D. Editores de código.
- E. S.O Manjaro Linux (derivada de Arch Linux).
- F. Se utilizará un hosting servidor.

### **1.5.2. Económica**

El desarrollo de la plataforma virtual para TECHSBOL, será una forma de ingreso económica ya que este proporcionará contenidos educativos de nivelación y capacitación de paga o de forma gratuita.

### **1.5.3. Social**

La plataforma virtual colaborará de forma directa con TECHSBOL en la capacitación y nivelación de sus estudiantes, y de forma indirecta con la sociedad ya que esta podrá ingresar a los cursos impartidos desde cualquier punto de la ciudad o a nivel nacional, teniendo una educación diferente a lo tradicional y así apoyando al crecimiento educativo tecnológico mejorando la calidad de vida.

## **1.6. METODOLOGÍA**

### **1.6.1. Método científico**

Para el presente proyecto de grado se utilizará el método científico, con un enfoque cuantitativo, de tipo descriptivo.

El método científico está conformado por etapas, las cuales son:

- Observación sistemática obteniendo conocimiento valido.
- Hipótesis la cual es una suposición que forma parte de una posible solución.
- Experimentación y emisión de conclusiones.

Un enfoque cuantitativo, ya que se seguirá un conjunto de procesos secuenciales y probatorios, además el proyecto es delimitado y concreto. Se considera lo que se ha investigado anteriormente.

### **1.6.2. Método de ingeniería**

#### **1.6.2.1. Metodologías de desarrollo de software**

- **Scrum Kamban** La metodología ágil que se empleara para el desarrollo es Kanban combinada con Scrum. Kanban trabaja a base de tarjetas, estas tarjetas de tareas son pegadas en la pared que muestran el estado actual de la tarea y Scrum ofrece Gestión regular de las expectativas del cliente, basada en resultados tangibles, resultados anticipados, flexibilidad y adaptación respecto

a las necesidades del cliente, cambios en el mercado, mitigación sistemática de los riesgos del proyecto (Henrik Kniberg, 2010).

- **UML:** Además de utilizar UML para el modelado de datos, ya que mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código. Provee especificaciones gráficas formales para un proceso de diseño completo que puede ser asistido por herramientas de diseño visuales (teniente, 2005).
- **OpenUP** es un proceso de desarrollo de software mínimamente suficiente, esto quiere decir que incluye solo el contenido fundamental, esto es que no provee orientación sobre temas en los que el proyecto tiene que lidiar, como son: el tamaño del equipo, el cumplimiento, seguridad, orientación tecnológica entre otras. Sin embargo, OpenUP es completa en el sentido de que manifiesta por completo el proceso de construir un sistema. Para atender las necesidades que no están cubiertas en su contenido OpenUp es extensible a ser utilizado como base sobre la cual se pueden añadir o adaptarse a contenido de otro proceso que sea necesario.

#### **1.6.2.2. Métricas de Calidad del software**

Se utilizará la metodología Web QEM (Quality Evaluation Method) se va a utilizar para evaluar la calidad de sitios web. Esta metodología parte de un modelo de calidad que proporciona un enfoque cuantitativo y sistemático para evaluar y comparar productos web tanto en la fase operativa como en la fase de desarrollo del ciclo de la vida de un producto. El principal objetivo es evaluar y determinar el nivel de cumplimiento de los siguientes factores de la calidad descritos en el estándar ISO 9126.

#### **1.6.2.3. Ingeniería de costos del software**

Planning Poker es una técnica de estimación utilizada en proyectos ejecutados con metodologías ágiles, como por ejemplo Scrum. Es una técnica para medir los elementos de pila de producto (Product Backlog) en un proyecto ágil, que no son más que el trabajo a realizar para desarrollar el software desglosado en componentes del software, funcionalidades específicas, en inclusive partes de funcionalidades.

## 1.7. HERRAMIENTAS

Para el desarrollo del SISTEMA DE EDUCACION VIRTUAL (E-LEARNING) COMO HERRAMIENTA DE ENSEÑANAZA utilizaremos las siguientes herramientas:

- **Codeigniter** es un framework (Oscar, Acosta, Mata, Bachmann, & Vallejos, 2012) para aplicaciones web de código abierto para crear sitios web dinámicos con PHP. «Su objetivo es permitir que los desarrolladores puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero, brindando un conjunto de bibliotecas para tareas comunes, así como una interfaz simple y una estructura lógica para acceder esas bibliotecas. (Technology, 2011).
- **MariaDB Server** es una de las bases de datos relacionales de código abierto más populares. Está hecho por los desarrolladores originales de MySQL y se garantiza que seguirá siendo de código abierto. Es parte de la mayoría de las ofertas en la nube y es el predeterminado en la mayoría de las distribuciones de Linux.
- **PHP** es un lenguaje de programación de uso general que se adapta especialmente al desarrollo web. Fue creado inicialmente por el programador danés-canadiense Rasmus Lerdorf en 1994. En la actualidad, la implementación de referencia de PHP es producida por The PHP Group. PHP originalmente significaba Personal Home Page (Página personal), pero ahora significa el inicialismo recursivo PHP: Hypertext Preprocessor. PHP suele ser procesado en un servidor web por un intérprete PHP implementado como un módulo, un Daemon o como un ejecutable de interfaz de entrada común (CGI). En un servidor web, el resultado del código PHP interpretado y ejecutado que puede ser cualquier tipo de datos, como el HTML generado o datos de imágenes binaria formaría la totalidad o parte de una respuesta HTTP. Existen diversos sistemas de plantillas, sistemas de gestión de contenidos y framework que pueden emplearse para organizar o facilitar la generación de esa respuesta. Por otra parte, PHP puede utilizarse para muchas tareas de programación fuera del contexto de la web, como aplicaciones gráficas autónomas y el control de

drones. También se puede interpretar y ejecutar un código PHP cualquiera a través de una interfaz de línea de comandos (CLI).

- **Visual Studio Code** Es un editor de código fuente desarrollada por Microsoft para Windows Linux y Mac incluye soporte para la depuración control de git resaltado de sintaxis finalización inteligente de código fragmentos y factorización de código. Tiene gran soporte de lenguajes como Php, JavaScript, java, typescript y otros (Microsoft, 2020).

## 1.8. LÍMITES Y ALCANCES

### 1.8.1. Limites

El presente proyecto se delimitará por las siguientes características:

- Inscripción al curso en línea
- Visualización del contenido impartido
- Panel administración para la institución
- Panel de administración para el docente
- Evaluaciones en línea por parte del docente
- Subido de contenido multimedia y de archivos
- Categorización de temas
- Calificaciones de los cursos impartidos
- Abra cursos gratuitos y de paga

### 1.8.2. Alcances

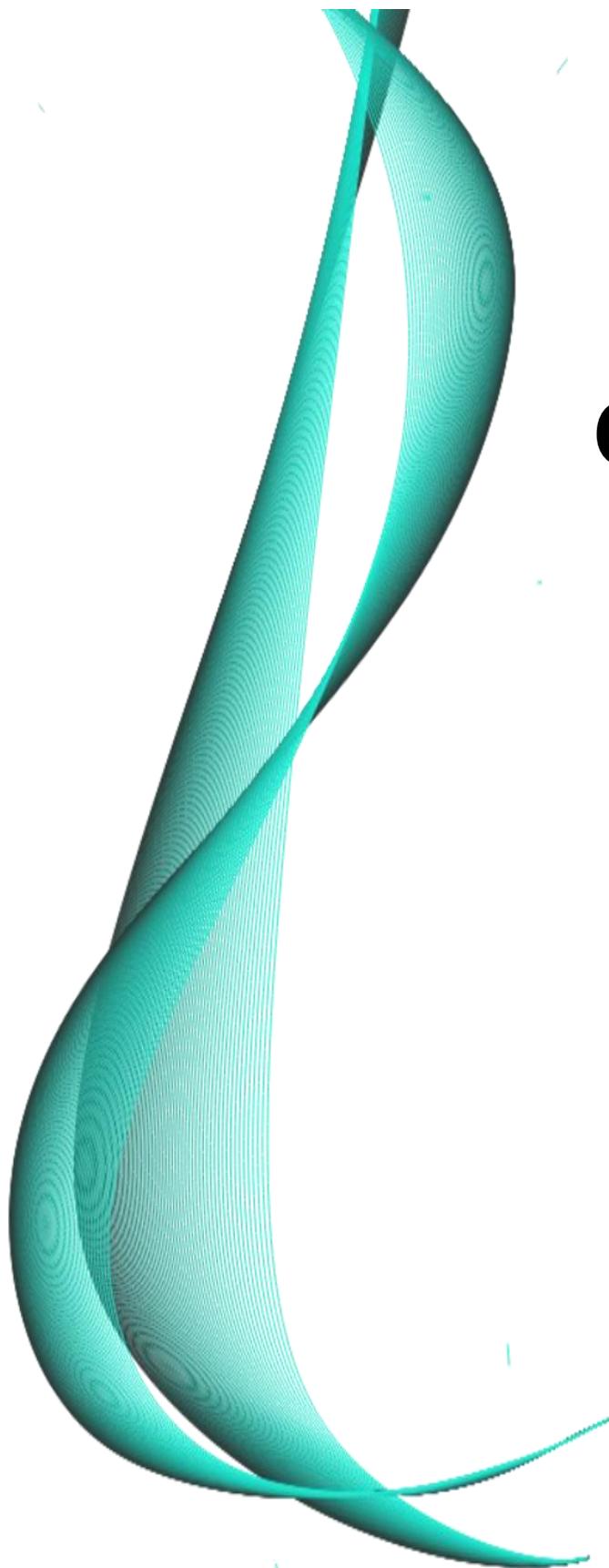
El sistema incluirá los siguientes módulos:

- Módulo de instructores
- Módulo registro
- Módulo de recuperación de contraseñas
- Módulo reportes de curso
- Módulo estudiante
- Módulo categorías
- Módulo de mensajes

- Módulo de lenguajes
- Módulo de pagos

### **1.9. APORTES**

Una Plataforma E-learning es un entorno informático basado en la web para el cumplimiento en la entrega de contenidos y servicios. En palabras prácticas y fáciles de comprender, es la capacitación realizada a través de computadoras o dispositivos electrónicos que no limitan al desempeño de tareas simultáneas. E-learning está considerado como uno de los más grandes aportes a la "Cultura Digital", enfocado específicamente en el campo educacional y proyectos. Esta metodología implica mantener una comunicación telemática, es decir cualquier tipo de comunicación por Internet, la cual puede ser asincrónica o sincrónica.



# **CAPÍTULO II**

## **MARCO**

### **TEÓRICO**

## **2. MARCO TEÓRICO**

### **2.1. INTRODUCCIÓN**

El propósito del marco teórico es establecer un marco teórico, el cual nos ayudara a conocer los rasgos optados para la implementación del proyecto.

### **2.2. SISTEMA**

En informática (Raffino, Informática, 2020), se entiendo por un sistema a un conjunto de datos ordenados conforme a una serie de instrucciones o algoritmos, que permiten su ubicación y recuperación rápida y simple.

Eso es un sistema de información o informático (Dangel, 2010), concepto que también emplean otras ciencias de la información como la bibliotecología, pero que en el caso de la informática está administrado de manera automática por un computador (Julián Pérez Porto, Definición de computadora, 2012).

### **2.3. INFORMACIÓN**

Según Idalberto Chiavenato, información "es un conjunto de datos con un significado, o sea, que reduce la incertidumbre o que aumenta el conocimiento de algo. En verdad, la información es un mensaje con significado en un determinado contexto, disponible para uso inmediato y que proporciona orientación a las acciones por el hecho de reducir el margen de incertidumbre con respecto a nuestras decisiones" (Idalberto, 2006)

### **2.4. SISTEMAS DE INFORMACIÓN**

Un sistema de información es un conjunto de elementos interrelacionados con el propósito de prestar atención a las demandas de información de una organización, para elevar el nivel de conocimientos que permitan un mejor apoyo a la toma de decisiones y desarrollo de acciones (Peña, 2006).

Otros autores como Peralta (Manuel, 2008), de una manera más acertada define sistema de información como: conjunto de elementos que interactúan entre sí con el

fin de apoyar las actividades de una empresa o negocio. Teniendo muy en cuenta el equipo computacional necesario para que el sistema de información pueda operar y el recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema.

Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información (Manuel, 2008).

## **2.5. EDUCACIÓN**

Se denomina educación a la facilitación del aprendizaje o de la obtención de conocimientos, habilidades, valores y hábitos en un grupo humano determinado, por parte de otras personas más versadas en el asunto enseñado y empleando diversas técnicas de la pedagogía: la narración, el debate, la memorización o la investigación.

La educación es un proceso complejo en la vida del ser humano, que ocurre fundamentalmente en el seno de la familia y luego en las distintas etapas de la vida escolar o académica que el individuo transite (desde el kindergarten hasta la universidad). (Raffino, Educación, 2020)

## **2.6. EDUCACIÓN A DISTANCIA**

El término educación a distancia cubre un amplio espectro de diversas formas de estudio y estrategias educativas, que tienen en común el hecho de que ellas no se cumplen mediante la tradicional contigüidad física continua, de profesores y alumnos en locales especiales para fines educativos; esta nueva forma educativa incluye todos los métodos de enseñanza en los que debido a la separación existente entre estudiantes y profesores, las fases interactiva y pre-activa de la enseñanza son conducidas mediante la palabra impresa, y/o elementos mecánicos o electrónicos. (ARMENGOL, 1982)

La educación a distancia es una estrategia educativa basada en la aplicación de la tecnología al aprendizaje sin limitación del lugar, tiempo, ocupación o edad de los estudiantes. Implica nuevos roles para los alumnos y para los profesores, nuevas actitudes y nuevos enfoques metodológicos. (LLAMAS, 1986)

## **2.7. E-LEARNING**

Aunque e-Learning no es un término castellano, su uso se ha generalizado de tal forma que es el más extendido a nivel mundial. Existen otros términos, que significan prácticamente lo mismo y a veces se usan como sinónimos, tales como: tele formación, formación on-line, enseñanza virtual, etc.

Podemos entender e-Learning como:

Procesos de enseñanza-aprendizaje que se llevan a cabo a través de Internet, caracterizados por una separación física entre profesorado y estudiantes, pero con el predominio de una comunicación tanto síncrona como asíncrona, a través de la cual se lleva a cabo una interacción didáctica continuada. Además, el alumno pasa a ser el centro de la formación, al tener que autogestionar su aprendizaje, con ayuda de tutores y compañeros. (SEVILLA, 2007)

## **2.8. TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN**

Las TIC son el conjunto de tecnologías desarrolladas en la actualidad para una información y comunicación más eficiente, las cuales han modificado tanto la forma de acceder al conocimiento como las relaciones humanas.

TIC es la abreviatura de Tecnologías de la Información y la Comunicación.

Las TIC han transformado los parámetros de obtención de información por medio de las tecnologías de la comunicación (diario, radio y televisión), a través del desarrollo de Internet y de los nuevos dispositivos tecnológicos como la computadora, la tableta y el smartphone, así como las plataformas y softwares disponibles.

Las TIC se reconocen como productos innovadores donde la ciencia y la ingeniería trabajan en conjunto para desarrollar aparatos y sistemas que resuelvan los problemas del día a día. Ellas sintetizan elementos de las llamadas tecnologías de la comunicación o TC (radio, prensa y TV) con las tecnologías de la información.

La información se refiere en este contexto a la transferencia de datos de un modo innovador, los cuales abarcan textos, imágenes y audio.

La comunicación se refiere a las herramientas que permiten que el mensaje enviado por el emisor sea correctamente descifrado por el receptor. Por ejemplo, las plataformas de información al usuario.

Existen dudas sobre la escritura correcta de las siglas, ya que refieren a un sustantivo plural. Las personas se debaten entre la grafía TIC, TIC's o TICS. Para indicar el plural, basta añadir el artículo femenino "las". Por ejemplo: "Las TIC han revolucionado el campo de la educación". Por lo tanto, lo correcto es escribir TIC, ya que la "s" minúscula suele ser percibida como un anglicismo (TIC's) y la "S" mayúscula suele ser percibida como otra sigla del término, lo que crea confusión. (Chen, 2019)

## **2.9. INGENIERÍA DE SOFTWARE**

Ingeniería de Software trata del establecimiento de los principios y métodos de la Ingeniería a fin de obtener software de modo rentable que sea fiable y trabaje en máquinas reales.

La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación (funcionamiento) y mantenimiento del software; es decir, la aplicación de ingeniería al software. (Pressman, 2010).

## **2.10. INGENIERÍA WEB**

La ingeniería web es un área que abarca procesos, técnicas y modelos orientados a los entornos Web. Consiste en la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones web de alta calidad. (MORA, 2003)

La ingeniería Web toma prestado muchos de los conceptos y principios básicos de la ingeniería del software, dando importancia a las mismas actividades técnicas y de gestión. Existen diferencias sutiles en la forma en que se llevan a cabo estas actividades, pero la filosofía primordial es idéntica dado que dicta un enfoque disciplinado para el desarrollo de un Sistema basado en computadora. (DÍAZ, 2005).

## **2.11. SCRUM**

La metodología es un marco diseñado para lograr la colaboración eficaz de equipos en proyectos, que emplea un conjunto de reglas y artefactos y define roles que generan la estructura necesaria para su correcto funcionamiento.

### **2.11.1. Características**

- Se divide en pequeños equipos, interdisciplinarios y organizados.
- Se divide el trabajo en pequeñas listas de enfoques concretos ordenados por prioridad y estimación de esfuerzo.
- Divide el tiempo en iteraciones cortas de longitud fija (generalmente de 1 a 4 semanas), con código potencialmente entregable y demostrado después de cada iteración.
- Se optimiza el plan de entrega actualizando las prioridades en colaboración del cliente.

### **2.11.2. Principios básicos**

Scrum propone el trabajo en ciclos sobre entregas parciales de un producto final amplio. Este tipo de trabajo permite distribuir mejor el tiempo y evita que un proyecto amplio se quede estancado.

Con Scrum es posible desglosar estos proyectos amplios en una lista de tareas y, de esa manera, el trabajo se vuelve más ágil. Al comenzar cada ciclo se definen qué tareas se van a realizar a lo largo del mismo y al finalizar se entregan resultados concretos. Además, plantea una división de roles entre el equipo de trabajo lo que fomenta la cooperación con nuestros compañeros (Wingu, 2016).

Scrum utiliza un enfoque incremental que tiene como fundamento la teoría de control empírico de procesos. Esta teoría se fundamenta en transparencia, inspección y adaptación; la transparencia, que garantiza la visibilidad en el proceso de las cosas que pueden afectar el resultado; la inspección, que ayuda a detectar variaciones indeseables en el proceso; y la adaptación, que realiza los ajustes pertinentes para minimizar el impacto de las mismas (Alliance, 2018).

Los integrantes del equipo tienen todos los conocimientos necesarios (por ser multifuncionales) para llevar a cabo el trabajo. La entrega del producto se hace en iteraciones; cada iteración crea nuevas funcionalidades o modifica las que el dueño del producto requiera (Navarro Cadavid, 2013).

### 2.11.3. Prácticas de scrum

Palacio (2014) refiere que SCRUM controla de forma empírica y adaptable la evolución del proyecto, empleando prácticas de la gestión ágil, ver tabla

**Tabla 1. Prácticas de la gestión ágil**

<b>Prácticas de la gestión ágil</b>	<b>Descripción</b>
<b>Revisión de las iteraciones</b>	Al finalizar cada iteración (normalmente 30 días) se lleva a cabo una revisión con todas las personas implicadas en el proyecto. Este es el periodo máximo que se tarda en reconducir una desviación en el proyecto o en las circunstancias del producto
<b>Desarrollo Incremental</b>	Durante el proyecto, las personas implicadas no trabajan con diseños o abstracciones. El desarrollo incremental implica que al final de cada iteración se dispone de una parte del producto operativa que se puede inspeccionar y evaluar
<b>Desarrollo evolutivo</b>	Los modelos de gestión ágil se emplean para trabajar en entornos de incertidumbre e inestabilidad de requisitos. Intentar predecir en las fases iniciales cómo será el producto final, y sobre dicha predicción desarrollar el diseño y la arquitectura del producto no es realista, porque las circunstancias obligarán a remodelarlo muchas veces.
<b>Auto organización</b>	Durante el desarrollo de un proyecto son muchos los factores impredecibles que surgen en todas las áreas y niveles. La

	gestión predictiva confía la responsabilidad de su resolución al gestor de proyectos
<b>Colaboración</b>	Las prácticas y el entorno de trabajo ágiles facilitan la colaboración del equipo. Ésta es necesaria, porque para que funcione la autoorganización como un control eficaz cada miembro del equipo debe colaborar de forma abierta con los demás, según sus capacidades y no según su rol o su puesto

**Fuente: (Palacio, 2014).**

#### 2.11.4. Roles Scrum

Schwaber & Sutherland (2011) describe tres roles en la tabla

**Tabla 2. Roles scrum**

<b>Nombre de rol</b>	<b>Descripción</b>
<b>Scrum master</b>	El Scrum Master es el responsable en asegurar que se entienda y se adopte Scrum. Los Scrum Masters hacen esto asegurándose de que el Equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum.
<b>Dueño del producto</b>	Es una sola persona y representa a los interesados, es el responsable de maximizar el valor del producto y el trabajo del equipo de desarrollo; tiene entre sus funciones gestionar la lista ordenada de funcionalidades requeridas o Product Backlog.
<b>Equipo de desarrollo</b>	Tiene como responsabilidad convertir lo que el cliente quiere, el Product Backlog, en iteraciones funcionales del producto; el equipo de desarrollo no tiene jerarquías, todos sus miembros tienen el mismo nivel y

	cargo: desarrollador. El tamaño óptimo del equipo está entre tres y nueve personas.
--	---

**Fuentes: (Schwaber & Sutherland, 2011)**

### **2.11.5. Eventos de Scrum**

Existen diferentes eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas en scrum. Todo evento es de compartimiento o periodos limitados (Sutherland, 2016).

Una vez que comienza un Sprint, la duración es fija y no puede acortarse o alargarse. Otros eventos pueden terminar siempre que se alcance el objetivo del evento, asegurando el empleo de una cantidad apropiada de tiempo sin permitir desperdicio en el proceso.

Además del propio Sprint, que es un contenedor de eventos, cada uno de los eventos de Scrum constituye una oportunidad formal para la inspección y adaptación en algún aspecto. Estos eventos se diseñaron específicamente para habilitar los pilares vitales de transparencia e inspección. La falta de algunos eventos da como resultado una reducción de la transparencia y constituye una pérdida de inspección y adaptación (Sutherland, 2016) .

#### **2.11.5.1. El sprint**

El corazón de Scrum es el Sprint, es un compartimiento o periodo de tiempo (time-box) de un mes o menos durante el cual se crea un incremento de producto “Terminado” utilizable y potencialmente desplegable. Lo más conveniente si la duración de los Sprints es consistente a lo largo de todo el esfuerzo de desarrollo. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint anterior (K., 2012).

Durante el Sprint:

- No realizan cambios que puedan afectar al objetivo del Sprint (Sprint Goal).
- No se disminuyen los objetivos de calidad.

- El alcance puede clarificarse y renegociarse entre el Propietario del Producto (Product Owner) y el Equipo de desarrollo a medida que el proyecto avanza más.

Cada Sprint puede considerarse un proyecto con un horizonte no mayor de un mes. Al igual que los proyectos, los Sprints se usan para alcanzar algo. Cada Sprint tiene una definición de lo que se construirá, un diseño y un plan flexible que guiará su construcción, el trabajo del equipo y el producto resultante (K., 2012).

Los Sprints se limitan a un mes calendario. Cuando el horizonte de un Sprint es demasiado grande la definición de lo que se está construyendo podría cambiar, la complejidad podría incrementarse y el riesgo podría aumentar. Los Sprints habilitan la predictibilidad al asegurar la inspección y adaptación del progreso al menos en cada mes calendario. Los Sprints también limitan el riesgo del coste a un mes calendario.

#### **2.11.5.2. Planificación de Sprint**

El trabajo a realizar durante el Sprint se planifica en la reunión de Planificación del Sprint (Sprint Planning). Este plan se crea mediante el trabajo colaborativo de todo el Equipo Scrum (Sutherland, 2016).

La Planificación del Sprint (Sprint Planning) tiene una duración máxima de ocho horas para un Sprint de un mes. Para Sprints más cortos el evento es usualmente más corto. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito. El Scrum Master enseña al Equipo Scrum a mantenerse dentro del periodo de tiempo.

La Planificación de Sprint responde a las siguientes preguntas:

¿Qué puede entregarse en el Incremento resultante del Sprint que comienza? ¿Cómo se conseguirá hacer el trabajo necesario para entregar el Incremento?

#### **2.11.5.3. Scrum diario**

El Scrum Diario es una reunión con una duración máxima de tiempo de 15 minutos para que el Equipo de Desarrollo (Development Team) sincronice sus actividades y

Cree un plan para las siguientes 24 horas. Esto se lleva a cabo inspeccionando el trabajo avanzado desde el último Scrum Diario (Daily Scrum) y haciendo una proyección acerca del trabajo que podría completarse antes del siguiente (Sutherland, 2016).

El Scrum Diario (Daily Scrum) se realiza a la misma hora y en el mismo lugar todos los días para reducir la complejidad. Durante la reunión, cada miembro del Equipo de Desarrollo (Development Team) explica:

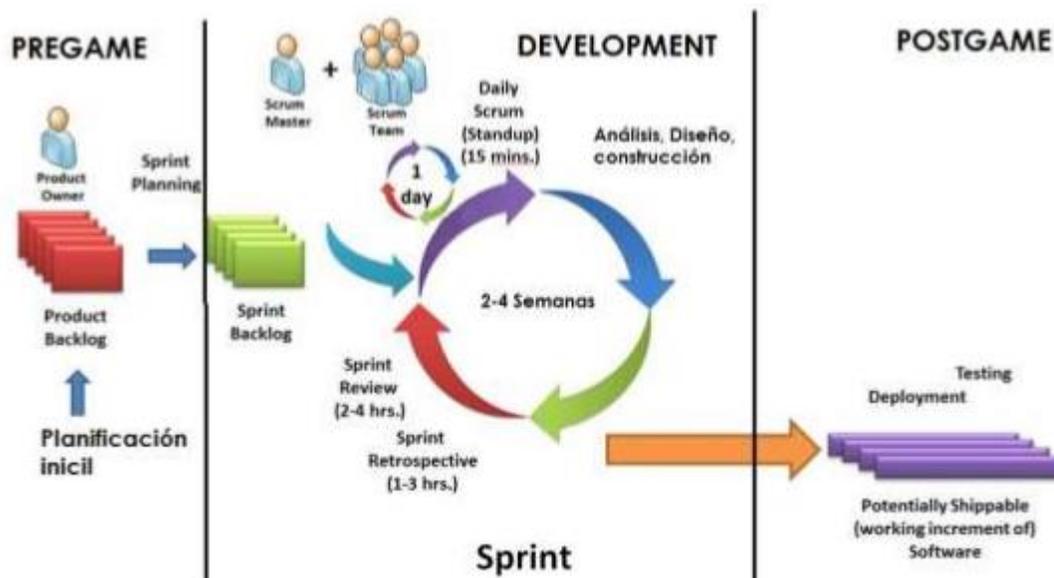
- ¿Qué hice ayer para ayudar al Equipo de Desarrollo (Development Team) a lograr el Objetivo del Sprint?
- ¿Qué haré hoy para ayudar al Equipo de Desarrollo (Development Team) a lograr el Objetivo del Sprint?
- ¿Detecto algún impedimento que evite que el Equipo de Desarrollo (Development Team) o yo terminen el Objetivo del Sprint?

El Equipo de Desarrollo (Development Team) utiliza el Scrum Diario (Daily Scrum) para evaluar el progreso hacia el objetivo del Sprint y para evaluar qué tendencia sigue este progreso hacia la finalización del trabajo contenido en la Pila del Sprint (Sprint Backlog). Cada día, el Equipo de Desarrollo (Development Team) debería entender cómo intenta trabajar en conjunto como un equipo autoorganizado para lograr el Objetivo del Sprint y crear el Incremento esperado hacia el final del Sprint. El Equipo de Desarrollo a menudo se vuelve a reunir inmediatamente después del Scrum Diario (Daily Scrum), para tener discusiones detalladas, o para adaptar o replanificar el resto del trabajo del Sprint.

#### **2.11.6. Fases de scrum**

Scrum consta de tres fases: Pregame, Development y Postgame (Beedle, 202). En la figura 1 se muestra los elementos de scrum en su respectiva fase:

Figura 1. Fases scrum



Fuente: (Beedle, 202)

### 2.11.6.1. Pregame

La fase de Pregame incluye dos subfases: planning y architecture, que se muestra en la tabla.

Tabla 3. Fases Pregame

Subface	Descripción
Planificación	Incluye la definición del sistema que se está desarrollando. Se crea una lista product backlog que contiene todos los requisitos que son actualmente conocido. Los requisitos pueden originarse del cliente, ventas y división de marketing, atención al cliente o desarrolladores de software. Los requisitos son prioritarios y el esfuerzo necesario para su implementación es estimado. La lista product backlog se actualiza constantemente con nuevos y más artículos detallados, así como con

	<p>estimaciones más precisas y nueva prioridad pedidos. La planificación también incluye la definición del equipo del proyecto, herramientas y otros recursos, evaluación de riesgos y cuestiones de control, necesidades de capacitación y aprobación de la gestión de verificación. En cada iteración, el product backlog es actualizado y la acumulación es revisada por el scrum team para obtener su compromiso para la próxima iteración.</p>
<p><b>Arquitectura</b></p>	<p>Se planifica en función de los elementos actuales en la lista product backlog. En el caso de una mejora de un sistema existente, se identifican los cambios necesarios para implementar los ítems de backlog junto con los problemas que pueden causar. Se realiza una reunión de revisión de diseño para revisar las propuestas para la implementación y las decisiones se toman sobre la base de esta revisión. Además, se preparan planes preliminares para los contenidos de las publicaciones.</p>

**Fuente: (Beedle, 202)**

### **2.11.6.2. Development**

La fase de Development también llamada Game Phase es la parte ágil del enfoque de Scrum. Esta fase se trata como una caja negra donde se espera lo impredecible. Las diferentes variables ambientales y técnicas (como marco de tiempo, calidad, requisitos, recursos, tecnologías y herramientas de implementación e incluso métodos de desarrollo) identificadas en Scrum, que pueden cambiar durante el proceso, se observan y controlan mediante diversas prácticas de Scrum durante el proceso. En lugar de tomar estos asuntos en consideración solo al comienzo del proyecto de desarrollo de software, Scrum apunta a controlarlos constantemente para poder

adaptarse de manera flexible a los cambios. En la fase de desarrollo, el sistema se desarrolla en Sprints (Figura 1). Los sprints son ciclos iterativos donde la funcionalidad se desarrolla o mejora para producir nuevos incrementos. Cada Sprint incluye las fases tradicionales de desarrollo de software: fases de requisitos, análisis, diseño, evolución y entrega (Figura 1). La arquitectura y el diseño del sistema evolucionan durante el desarrollo de Sprint. Se espera que un Sprint dure de una semana a un mes. Puede haber, por ejemplo, de tres a ocho Sprints en un proceso de desarrollo de sistemas antes de que el sistema esté listo para su distribución. También puede haber más de un equipo construyendo el incremento.

#### **2.11.6.3. Postgame**

La fase Postgame contiene el cierre del lanzamiento. Esta fase es ingresada cuando se ha llegado a un acuerdo que las variables ambientales como los requisitos se completan. En este caso, no se pueden encontrar más elementos y problemas ni se pueden inventar nuevos. El sistema ahora está listo para el lanzamiento y la preparación para esto se hace durante la fase Postgame, incluidas las tareas como la integración, las pruebas del sistema y la documentación (Figura 1).

#### **2.12. KANBAN**

El término viene del japonés: Kan, visual, y ban, tarjeta. Kanban permite visualizar el flujo de trabajo en una barra de tareas a través de tarjetas. Propone distribuir las mismas en una serie de columnas. Kanban trabaja con tableros que pueden ser tanto físicos como digitales y permite una visualización clara de todas las tareas a realizar, en qué etapa está cada una y quién es el encargado de las mismas (Rocher, 2013), 2013).

El Kanban es un sistema de gestión del trabajo en curso, que sirve principalmente para asegurar una producción continua y sin sobrecargas en el equipo de producción multimedia. El Kanban es un sistema de gestión donde se produce exactamente aquella cantidad de trabajo que el sistema es capaz de asumir (Fitzgerald, 1999).

### 2.12.1. Características

Según Ibarra Guzmán & Castañeda Islas (2014) las principales características de Kanban se lo pueden observar en la tabla.

**Tabla 4. Característica Kanban**

<b>Características</b>	<b>Descripción</b>
<b>Visualiza el flujo de trabajo</b>	Divide el trabajo en bloques, escribe cada elemento en una tarjeta y se coloca en el tablero. Utiliza columnas con nombre para ilustrar dónde está cada elemento en el flujo de trabajo
<b>Limita el trabajo en curso</b>	Asigna límites concretos al número de elementos que pueden estar en progreso en cada estado del flujo de trabajo.
<b>Mide el tiempo de ciclo medio para completar un elemento</b>	Optimiza el proceso para que el lead time sea tan pequeño y predecible como sea posible.

**Fuente: (Isla & Castañeda, 2014)**

### 2.12.2. Roles Kanban

Kanban, a diferencia de otras metodologías ágiles como Scrum, no prescribe roles o reuniones. Pero existen una serie de roles que pueden ser adoptados en Kanban, estos han sido creados a partir de la observación en organizaciones que han utilizado el método y lo han evolucionado de forma colaborativa (J., 2016).

- **Service Request Manager:** Se encarga de gestionar la demanda y los requisitos dentro del sistema Kanban, manejando las relaciones con los stakeholders y fomentando la transparencia dentro del sistema en torno a la priorización del trabajo. Alternativamente, este se puede llamar Product Owner.
- **Service Delivery Manager:** Es responsable del flujo de trabajo dentro de un sistema Kanban y/o determinados ítems de trabajo y facilita el Kanban Meeting y el Delivery Planning. Algunos nombres alternativos son Flow Manager, Delivery Manager o incluso Flow Master.

### 2.12.3. Tablero Kanban

Los tableros Kanban, que se desprenden de la metodología del mismo nombre, son herramientas visuales para la organización de las tareas.

**Figura 2. Tablero Kanban**



**Fuente: (Velázquez & Cornejo, 2015)**

Un tablero de Kanban en su concepción más simple es un diagrama o rectángulo dividido en columnas, las cuales mantienen un número finito de tareas en ejecución (WIP) y las tarjetas, las cuales emulan las actividades que realiza cada miembro del equipo de trabajo (Cornejo Velázquez, 2015). La tabla muestra los elementos del tablero Kanban.

**Tabla 5. Elementos tablero Kanban**

<b>Elementos</b>	<b>Descripción</b>
<b>Tareas o actividades a realizar</b>	Se representan en el tablero mediante tarjetas haciendo alusión al origen etimológico de la palabra Kanban. Las tarjetas que representan cada tarea o actividad se mueven a través de las

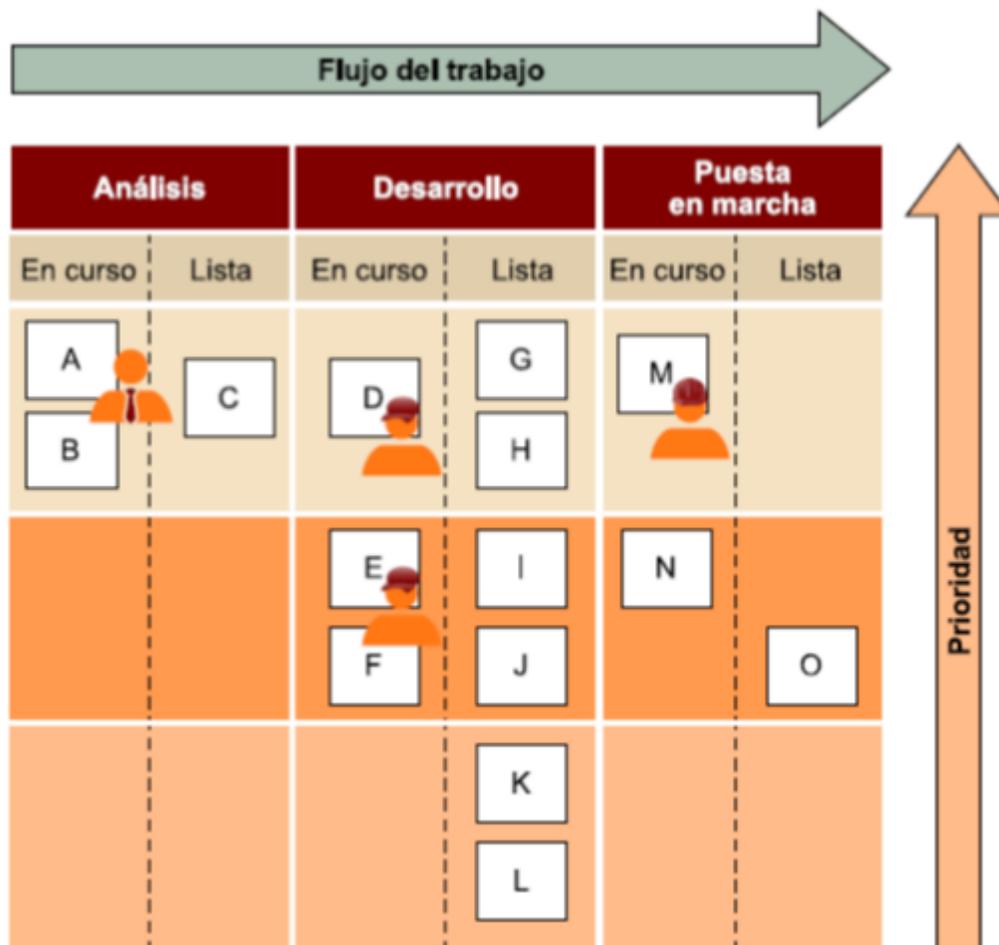
	columnas describiendo el estado en el que se encuentra.
<b>Columnas</b>	Estas representan las fases o etapas por las que debe pasar una actividad antes de considerarse terminada. Definir el número de columnas que debe tener un tablero Kanban siempre dependerá de las características del proyecto al que se quiera aplicar y del manual de procesos que se tenga en cada ente organizacional, ya que, como se mencionó anteriormente, la intención del Kanban no es contaminar ni cambiar el flujo y las formas de trabajo sino aportar una representación visual al mismo.
<b>WIP</b>	No es propiamente un elemento tangible dentro del tablero, pero sin el cual no se podría tener un control sobre las actividades máximas soportadas por el equipo de trabajo. Esto es tal vez lo que más dificultad presenta al momento de implantar un tablero Kanban o la metodología por sí misma debido a que nunca se tiene definido el número finito de tareas o actividades que se deben realizar por cada fase que compone el proceso de desarrollo del proyecto

**Fuente: (Skarin, 2010).**

#### **2.12.4. Fases Kanban**

La figura 2.2 muestra un tablero Kanban constituido por 3 columnas, que representan las diferentes fases por las que una tarea tiene que pasar para ser terminada.

Figura 3. Panel Kanban



Fuente: (Bermejo, 2010)

En la imagen se puede observar un panel constituido por tres columnas, que representan las diferentes fases por las que una tarea tiene que fluir para ser desarrollada (análisis, desarrollo y puesta en marcha).

Cada fase se encuentra subdividida en dos estados, que son en curso y lista, para pasar a la siguiente fase; esta división está representada por la línea discontinua de cada fase. El estado en curso significa que el equipo está actualmente trabajando en esta tarea, en esta fase y el estado lista significa que el equipo ya ha acabado el trabajo que tenía que ejecutar en esta fase y la tarea está esperando a que el sistema pueda

asumirla para la siguiente fase. Esta división nos ayuda a localizar atascos en el proceso de producción (Morales, 2015).

Las filas podrán representar diferentes proyectos en los que la empresa trabaja, siendo lo habitual que las filas indiquen prioridad, donde las tareas más superiores son las más prioritarias.

### **2.13. Combinación de scrum y Kanban SCRUMBAN**

Scrumban combina las mejores características de ambos métodos. Reúne la naturaleza preceptiva de Scrum y la capacidad de mejora del proceso de Kanban, permitiendo a los equipos moverse hacia el desarrollo Agile y a mejorar constantemente sus procesos. Scrumban se está haciendo especialmente popular en industrias en las que el desarrollo del proyecto y el mantenimiento van juntos.

#### **2.13.1. Diferencias básicas entre scrum y Kanban**

El Kanban no prescribe roles ni iteraciones fijas. Las entregas se pueden definir en función de las necesidades de la empresa: de manera regular (todos los viernes) o cada vez que se acabe un servicio y se ponga en producción. Scrum no prescribe límite WIP o, como mínimo, no directamente. Scrum limita el trabajo en curso al trabajo comprometido en la iteración actual, es decir, si en el sprint planning meeting se decide llevar a cabo diez funcionalidades en dos semanas, el primer día del sprint, el equipo podría tener en desarrollo las diez a la vez (Bermejo, 2010).

Scrum no permite la entrada de trabajo nuevo cuando la iteración ha empezado. Si en el sprint se han comprometido diez tareas, Scrum no permite cambiarlas ni añadir otras nuevas a media iteración, por lo tanto en equipos donde las tareas estén definidas a corto plazo, iteraciones muy cortas son casi obligatorias. Si en la empresa pueden entrar tareas de hoy para mañana, estas no pueden ser asumidas por Scrum puesto que sirve desarrollos en iteraciones fijas de como mínimo una semana

#### **2.13.2. Beneficios de la integración**

- Permite conocer en estado real el proceso de ejecución del proyecto.

- Introduce soluciones oportunas ante eventuales errores.
- Permite un mayor análisis de tareas realizadas.
- Mejora la interacción entre los miembros de un grupo en las reuniones periódicas.
- Aumenta la productividad de proyectos complejos o multiproyectos.
- Favorece una mayor adaptabilidad de las herramientas a las exigencias del proyecto.

## **2.14. MÉTRICA DE CALIDAD**

Se utilizará la metodología WebQem para medir la calidad del sistema.

### **2.14.1. Fases de Web Qem**

A continuación, se describe las fases de la metodología WebQem que se utilizara para realizar los cálculos que ayudara a comprobar que el sistema es confiable, eficiente y de calidad.

### **2.14.2. Definición de Las Metas de Evaluación y Selección del Perfil de Usuario**

En esta fase se consideran dos pasos primordiales y comunes en toda evaluación de calidad siguiendo la metodología WebQem.

#### **2.14.2.1. Metas de evaluación**

En esta fase se define las metas de evaluación y selección del perfil de usuario, los evaluadores deben definir las metas y establecer el alcance del proyecto. Las metas a llegar a cumplir con la evaluación del sistema son:

- Conocer la percepción de la calidad que tienen los usuarios finales sobre el sistema.
- Lograr que la calidad del sistema este por encima de las expectativas del usuario final.

#### **2.14.2.2. Selección del perfil de usuario**

Para el diseño e implementación de un caso de estudio de evaluación de calidad en uso, una meta muy utilizada es determinar el cumplimiento de requerimientos

elementales, parciales y globales de calidad para una aplicación Web operativa, considerando el perfil de encargado de requerimiento de pedidos y compras. Los encargados de realizar los pedidos y compras son los usuarios que interactuarán con el sistema de forma recurrente y a diario estos usuarios conocen la lógica del negocio y tuvieron capacitación sobre el manejo del sistema.

### **2.14.3. Definición de los requerimientos de calidad/costo**

En esta fase, teniendo en cuenta los aspectos definidos en la fase anterior respecto a metas de evaluación, selección de perfil de usuario, se deben establecer atributos y subconceptos (características) de calidad cuantificables que, agrupados jerárquicamente, representen un modelo de calidad apropiado para el perfil de usuario seleccionado. Para esto se realizará un árbol de requisitos:

#### **2.14.3.1. Usabilidad**

- Comprensibilidad del Sistema.
- El sistema debe ser amigable para el usuario.
- Aspectos de Interfaces y Estéticos.
- Los datos deben estar ordenados de la forma correcta para una mejor comprensión.
- Las interfaces deben ser clara.

#### **2.14.3.2. Funcionalidad**

- Aspectos de Búsqueda
  - Mecanismo de Búsqueda
  - Búsqueda parcial bajo diversos parámetros
  - Mostrar todos los resultados coincidentes con los parámetros ingresados
  - Validar todos los parámetros de ingreso la búsqueda.
- Aspectos de Navegación y Exploración
  - Navegabilidad Local
  - Navegación solo por interfaces asignados al rol del usuario.

- Funciones de registro de compras y pedidos.
  - Formulario comprensible
  - Indicar campos obligatorios
  - Validar campos del formulario
  - Ver estado actual de la compra o pedido.
  
- Funciones de registro de movimientos de madera.
  - Formulario de registro comprensible
  - Validar campos del formulario

#### **2.14.3.3. Confiabilidad**

- No Deficiencia
  - Errores de Enlaces
  - Enlaces Rotos
  - Enlaces Inválidos
  - Enlaces no Implementados

#### **2.14.3.4. Eficiencia**

- Accesibilidad de Información
  - Visualización de la información solicitada de forma clara
  - Eficiencia a la hora de realizar un registro
  
- Performance
  - Páginas Rápidas

#### **2.14.4. Definición de Criterios Elementales y Procedimientos de Medición**

En esta es la etapa donde se definirá una base de criterios para evaluación elemental para realizar el proceso de medición y puntuación de los valores, para medir la calidad del software en tal caso se definirá los siguientes criterios:

**Tabla 6. Tabla de ponderación en porcentajes**

Características de calidad	Mala	Regular	Buena	Muy bueno
funcionalidad	0-30%	31-50 %	51-90 %	91-100 %
Confiabilidad	0-30%	31-50 %	51-90 %	91-100 %
Usabilidad	0-30%	31-50 %	51-90 %	91-100 %
Mantenibilidad	0-30%	31-50 %	51-90 %	91-100 %
Portabilidad	0-30%	31-50 %	51-90 %	91-100 %
Total	0-30%	31-50 %	51-90 %	91-100 %

**Fuente: elaboración propia**

La tabla mostrada es un indicador de la calidad según el rango en el que se encuentre.

#### **2.14.5. Características de WebQem**

La metodología WebQem toma las métricas del modelo de calidad ISO 9126-1 la cual da referencia a las siguientes características, funcionalidad, Confiabilidad, Usabilidad, Mantenibilidad y Portabilidad.

#### **2.14.6. Funcionalidad**

Funcionalidad es una métrica orientada a la función del sistema y al proceso por el cual se desarrolla. Se centra en la funcionalidad o utilidad del programa.

- Entradas del usuario, se toma en cuenta cada entrada del usuario que el sistema proporciona a medida que ingresa al sistema.
- Salidas del usuario, se refleja las salidas que tiene el sistema tanto reportes como estadísticas que tiene el sistema.
- Número de peticiones del usuario, una petición se define como una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida.
- Número de archivos, se define cada archivo lógico.

- Número de interfaces externas, se definen todas aquellas interfaces legibles por el ordenador que solicitan transmitir información a otro sistema.

Para calcular los puntos función se usó las siguientes formulas:

$$PF = \text{Cuenta Total} * (\text{confiabilidad proyecto} + \text{error min} * \Sigma Fi)$$

$$\text{Funcionalidad} = \left( \frac{PF}{PF \text{ Maximo}} \right)$$

Donde:

**PF:** Medida de funcionalidad.

**PFmaximo:** Medida de funcionalidad con su valor máximo.

**Cuenta Total:** Es la suma de los siguientes datos: Número de entradas, número de salidas, número de peticiones, número de archivos y número de interfaces externas.

**Confiabilidad proyecto:** Confiabilidad del proyecto, varia de 1% al 100% (0 a 1).

**Error min:** Error mínimo aceptable de complejidad.  $\Sigma Fi$ : Son los valores de ajuste de complejidad, donde  $(1 \leq i \leq 14)$ .

### 2.14.7. Confiabilidad

Es la probabilidad de operación libre de fallos en un programa en un entorno determinado y durante un tiempo específico se toma en cuenta:

A) Enlaces

- Enlaces Rotos
- Enlaces Inválidos
- Enlaces no Implementados

B) Paginas

- Páginas Muertas

- Páginas bajo Construcción
- Errores de Ortografía

Para calcular los puntos función se usa la siguiente formula:

$$\% \text{ enlaces rotos} = \frac{\text{CERI} + \text{CERE}}{\text{CTE}} * 100$$

Donde:

CERI: Cantidad de enlaces rotos internos

CERE: Cantidad de enlaces rotos externos

CTE: Cantidad total de enlaces

Porcentaje de presencia de propiedad:

$$\% \text{ precia Alt} = \frac{\text{cantidadImágenesAlt}}{\text{cantidadTotalImágenes}} * 100$$

Y la confiabilidad se calcularía con la siguiente formula:

$$\text{Confiabilidad} = \text{PorcentajeEnlaces} + \text{PorcentajePresenciaALT}$$

### **2.14.8. Usabilidad**

Toma en cuenta la capacidad del software para ser comprendido, utilizado y atractivo para el usuario en determinadas condiciones.

Se realiza una tabla que toma en cuenta los siguientes puntos:

A) Comprensión Global del Sitio

i. Esquema de Organización Global

- Tabla de Contenidos
- Mapa del Sitio

- Índices (Alfabéticos, Temáticos, Híbridos...)

ii. Visita Guiada (convencional y/o virtual)

iii. Mapa de Imagen

B) Aspectos de Interfaces y Estéticos

i. Permanencia y Estabilidad en la Presentación de los Controles Principales

- Controles Directos
- Controles Indirectos
- Estabilidad

C) Mantenimiento del Color de Los Enlaces

Se toma en cuenta el diseño que tiene el sistema.

#### **2.14.9. Mantenibilidad**

La mantenibilidad es la facilidad con que una modificación puede ser realizada. Está indicada por los siguientes sub atributos.

- Facilidad de análisis
- Facilidad de cambio
- Estabilidad
- Facilidad de prueba

El índice de madurez del software se calcula con la siguiente formula:

$$IMS = \frac{[Mt - (Fa + Fb + Fc)]}{Mt}$$

Donde:

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.

#### **2.14.10. Portabilidad**

La portabilidad es la capacidad que tiene el sistema para ser trasladado de un entorno a otro.

Para poder medir la portabilidad del sistema usaremos la siguiente formula que indica el grado de portabilidad que tiene un software.

$$GP = 1 - \frac{ET}{ER}$$

Donde:

ET: Es la medida de los recursos necesarios para llevar el sistema a otro entorno.

ER: Es la medida de los recursos necesarios para crear el sistema en el entorno residente.

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.

#### **2.15. SEGURIDAD**

Los problemas de seguridad para sitios y sistemas web deben ser contemplados desde el momento del diseño lógico.

Los peligros se identifican al principio del proceso de desarrollo de software y las características de su diseño se especifican de modo que los eliminen o controlen

### **2.15.1. Seguridad por niveles**

Los niveles de seguridad proporcionan acceso a los diferentes módulos del sistema.

### **2.15.2. Seguridad a nivel de aplicación**

Los ataques a nivel de aplicación son una amenaza en constante aumento contra la seguridad web.

Utilizan una gran variedad de medios para paralizar un sitio Web e introducirse en él, lo que provoca resultados que varían desde un menor rendimiento del sitio Web hasta robos de datos y una desprotección de la infraestructura.

La seguridad web se divide en:

- Integridad, consistente en que el activo de información no ha sido alterado de manera no autorizada, además de garantizar que los datos sean los que se supone que son.
- Confidencialidad, asegurar que solo los individuos autorizados tengan acceso a los recursos que se intercambian.
- Disponibilidad, garantizar el correcto funcionamiento de los sistemas de información.
- Autenticación, asegurar que solo los individuos autorizados tengan acceso a los recursos.
- Trazabilidad, consiste en que las actualizaciones de una entidad pueden ser imputadas exclusivamente a dicha entidad.

### **2.15.3. Seguridad a nivel de la base de datos**

La información almacenada en la base de datos es importante para realizar el estudio correspondiente, es por ello que a nivel de base de datos se tomó en cuenta la validación de los campos a ingresar, con el propósito de evitar cualquier ataque.

La seguridad en las bases de datos y la aplicación web se encuentran profundamente relacionados, en aplicaciones web toda entrada al sistema debe ser filtrada y toda

salida escapada. Lo mismo aplica cuando las entradas o salidas son de o hacia una base de datos.

- Descubrimiento de información acerca de los datos de conexión al servidor (usuario y clave), información sensible almacenada en la base de datos o información sobre la estructura de la base de datos.
- Modificación de las instrucciones SQL enviadas al servidor, construidas de forma dinámica a partir de datos recibidos del usuario y por tanto potencialmente peligrosos (Inyección SQL).

Por lo tanto, para eliminar estas vulnerabilidades de seguridad, se tiene un gestor de base de datos ya implementado para los diferentes sub sistemas que comprenden al sistema.

#### **2.15.4. Seguridad a nivel del servidor**

La seguridad a nivel del servidor en el desarrollo de una aplicación web requiere de una serie de herramientas del lado del servidor como: servidores web, servidores de aplicación, servidores de bases de datos, lenguajes de servidor, los cuales comprometen a la aplicación si no se elimina:

- Vulnerabilidades debidas al uso de versiones no actualizadas.
- Configuraciones por defecto inadecuadas.
- Activación de cuentas por defecto.

En consecuencia, para eliminar estas vulnerabilidades se toman acciones para evaluar la vulnerabilidad y la configuración del servidor de aplicaciones web y del gestor de base de datos.

## 2.16. HARDWARE

### 2.16.1. EQUIPO DE DESARROLLO A USAR

**Tabla 7. Equipo desarrollo**

CARACTERÍSTICAS	DETALLE
Procesador	Core i3
RAM	8 gigas
Disco hdd	500gigas
pantalla	15,5 pulgadas
marca	Aspire acer

**Fuente: elaboración propia**

## 2.17. SOFTWARE

### 2.17.1. Sistema operativo

- **Manjaro** (basado en Arch Linux) Es un sistema operativo accesible, amigable y de código abierto. Brinda todos los beneficios del software de vanguardia combinado con un enfoque en comenzar rápidamente, herramientas automatizadas para requerir menos intervención manual y ayuda disponible cuando sea necesario. Manjaro es adecuado tanto para principiantes como para usuarios experimentados de computadoras.

A diferencia de los sistemas operativos propietarios, usted tiene un control total sobre su hardware, sin restricciones. Esto lo hace ideal para las personas que desean aprender cómo funciona Linux y en qué se diferencia de otros sistemas operativos. Desde esta perspectiva, también es adecuado para principiantes, de forma similar a como un Arduino es un excelente punto de entrada al desarrollo de hardware integrado.

### 2.17.2. IDE (Integrated Development Environment) de desarrollo

Un entorno de desarrollo integrado (IDE) es un sistema de software para el diseño de aplicaciones que combina herramientas del desarrollador comunes en una sola interfaz gráfica de usuario. Por lo general un IDE cuenta con estas características:

- **Editor de código fuente:** editor de texto que ayuda a escribir el código de software con funciones de resaltado de sintaxis con indicadores visuales, el relleno automático específico del lenguaje y la comprobación de errores a medida que se escribe el código.
- **Automatización de compilación local:** herramientas que automatizan tareas sencillas e iterativas como parte de una compilación local del software para el uso del desarrollador, como la compilación del código fuente de la computadora en un código binario, el empaquetado del código binario y la ejecución de pruebas automatizadas.
- **Depurador:** programa que sirve para probar otros programas y mostrar la ubicación de un error en el código original de forma gráfica.

#### 2.17.2.1. Bootstrap Studio

Bootstrap studio es un entorno de desarrollo de diseño y prototipado de sitios web. Esta incluye gran cantidad de componentes integrados, para ensamblar páginas webs receptivas las cuales son construidas en base a framework Bootstrap incluye una exportación limpia y semántica.

#### 2.17.2.2. Visual Studio Code

Es un editor de código fuente desarrollada por Microsoft para Windows Linux y Mac incluye soporte para la depuración control de git resaltado de sintaxis finalización inteligente de código fragmentos y factorización de código. Tiene gran soporte de lenguajes como Php, JavaScript, java, typescript y otros.

#### 2.17.2.3. phpMyAdmin

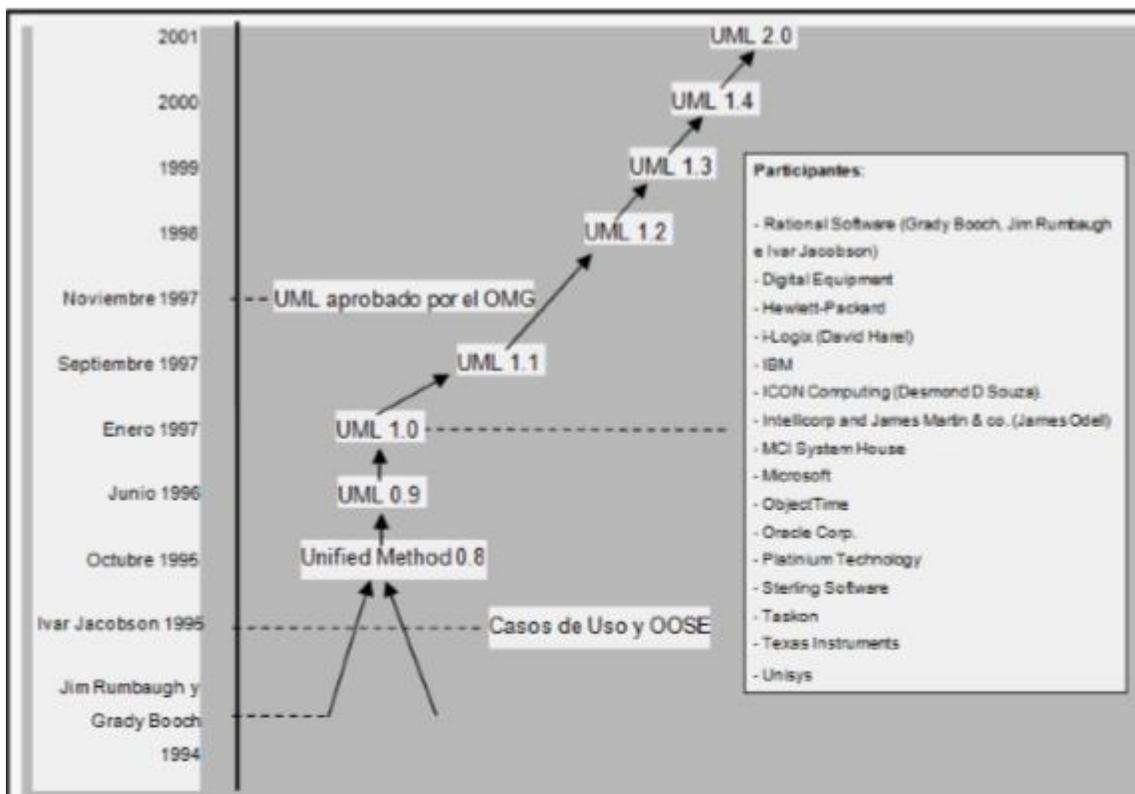
esta es una herramienta escrita en Php con la finalidad del manejo y administración de MySQL a través de páginas web. Actualmente puede crear, eliminar base de datos,

crear eliminar y alterar tablas, borrar editar y añadir campos ejecutar cualquier sentencia SQL administrar claves en campos administrar privilegios exportar datos en varios formatos.

## **2.18. LENGUAJE UNIFICADO DE MODELADO (U.M.L.)**

UML se define como un "lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software" (Booch, 1998). Es un lenguaje notacional (que, entre otras cosas, incluye el significado de sus notaciones) destinado a los sistemas de modelado que utilizan conceptos orientados a objetos. El UML es un estándar para construir modelos orientados a objetos. Nació en 1994 por iniciativa de Grady Booch y Jim Rumbaugh para combinar sus dos famosos métodos: el de Booch y el OMT (Object Modeling Technique, Técnica de Modelado de Objetos). Más tarde se les unió Ivar Jacobson, creador del método OOSE (Object-Oriented Software Engineering, Ingeniería de Software Orientada a Objetos). En respuesta a una petición del OMG (Object Management Group, Grupo de Administración de Objetos) para definir un lenguaje y una notación estándar del lenguaje de construcción de modelos, en 1997 propusieron el UML como candidato (Larman C. 1999).

**Figura 4. Historia UML**



**Fuente: (Letelier Torres, 2002)**

## 2.19. patrones de diseño

Los patrones son soluciones simples y elegantes a problemas específicos en el desarrollo de software. Son soluciones basadas en la experiencia y que se han demostrado que funcionan.

Cada patrón se adecua para ser adaptado a un cierto tipo de problema y con eso se pretende:

- Proporcionar catálogos de elementos reusables en el diseño de sistemas de software, lo que significa que debe ser aplicable a diferentes problemas de diseño en distintas circunstancias.

- Evitar la reiteración en la búsqueda de soluciones ya conocidos y solucionados anteriormente.
- Estandarizar el modo en que se realiza el diseño.
- Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento existente

### 2.19.1. modelo vista controlador

EL Modelo Vista Controlador (MVC), es un patrón de diseño que separa en capas bien definidas el desarrollo de una aplicación, esas partes son tres, el Modelo encargado de la lógica del negocio y la persistencia de los datos, las Vistas son las responsables de mostrar al usuario el resultado que obtienen del modelo a través del controlador, el Controlador encargado es el encargado de gestionar las peticiones del usuario, procesarlas invocando al modelo y mostrarlas al usuario a través de las vistas (Leff & Rayfield, 2001).

A continuación, se describirá los componentes:

- **Modelo:** es la representación lógica de la información y describe la funcionalidad del sistema, por lo tanto, se encarga de gestionar los permisos para ingresar a dicha información, también se encarga de realizar recuperación, actualización, inserción y eliminación de datos a través de un conjunto de datos.
- **Vista:** esta es la interfaz de usuario, siendo una representación visual del modelo, siendo visibles los datos y el estado del modelo. También contiene elementos que permiten al usuario interactuar con el programa, tales como botoneras, menús y submenús. Dejando en claro que no depende de la vista como se comportan estos elementos.
- **Controlador:** es el intermedio entre la vista y el controlador, responde a eventos generados por el usuario el cual se constituyen a través de llamados al modelo o la vista. o en pocas palabras el componente que hace posible tener la lógica de negocio con la vista es el controlador.

## **2.20. tecnologías web**

### **2.20.1. Php**

PHP acrónimo de Hypertext Preprocessor es un lenguaje de programación del lado del servidor diseñado originalmente para la generación de páginas web dinámicas. Es un lenguaje de programación interpretado o de script que permite insertar fragmentos de código dentro de una página HTML y realizar determinadas acciones de forma fácil y eficiente.

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en una herramienta ideal al momento de crear una página web dinámica:

- Fácil de mantener y mejorar el código desarrollado.
- Soporte para una gran cantidad de base de datos como MySQL, PostgreSQL, Oracle, MariaDB, MS SQL Server, entre otras.
- Facilidad de integración con varias bibliotecas externas utilizando Composer.
- Amplia comunidad de desarrolladores, por ser de código abierto, goza del aporte de gran número de programadores.
- Permite aplicar técnicas de programación orientada a objetos.

### **2.20.2. Framework Codeigniter**

Codeigniter es un conjunto de herramientas para desarrollar aplicaciones web usando PHP. Su objetivo es permitir desarrollar proyectos mucho más rápido, al proveer un rico conjunto de bibliotecas para tareas comúnmente necesarias, tanto como una interfaz sencilla y una estructura lógica para acceder a esas bibliotecas. Permite enfocarse en el proyecto al minimizar la cantidad de código necesario, está liberado bajo licencias open source como Apache/BSD, así que puede usarse donde sea necesario.

El núcleo de Codeigniter es bastante ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. La mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, sólo cuando se van a utilizar realmente.

Usa el patrón MVC, que permite una separación entre la lógica y la presentación. Es particularmente bueno para proyectos en los que los diseñadores trabajan en sus archivos de plantillas, ya que el código en estos archivos será mínimo:

- El Modelo representa sus estructuras de datos. Típicamente sus clases del modelo contendrán funciones que los ayudarán a devolver, insertar y actualizar información de su base de datos.
- La Vista es la información que se presenta al usuario. Una vista será normalmente una página web, pero en CodeIgniter, una vista también puede ser un fragmento de página como el encabezado o pie de página.
- El Controlador sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la solicitud HTTP y generar una página web.

### **2.20.3. JavaScript**

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden ser interpretados por los navegadores.

### **2.20.4. JQuery**

JQuery es un framework de JavaScript, permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejo de eventos desarrollar animaciones y agregar interacción con la técnica AJAX en páginas web

### **2.20.5. Bootstrap**

Bootstrap es un framework de CSS, de código abierto, en otras palabras, es un conjunto de archivos CSS que se incluye en una página y se puede empezar a maquetar el sitio web en minutos, sin tocar una sola línea de CSS, esto agiliza el

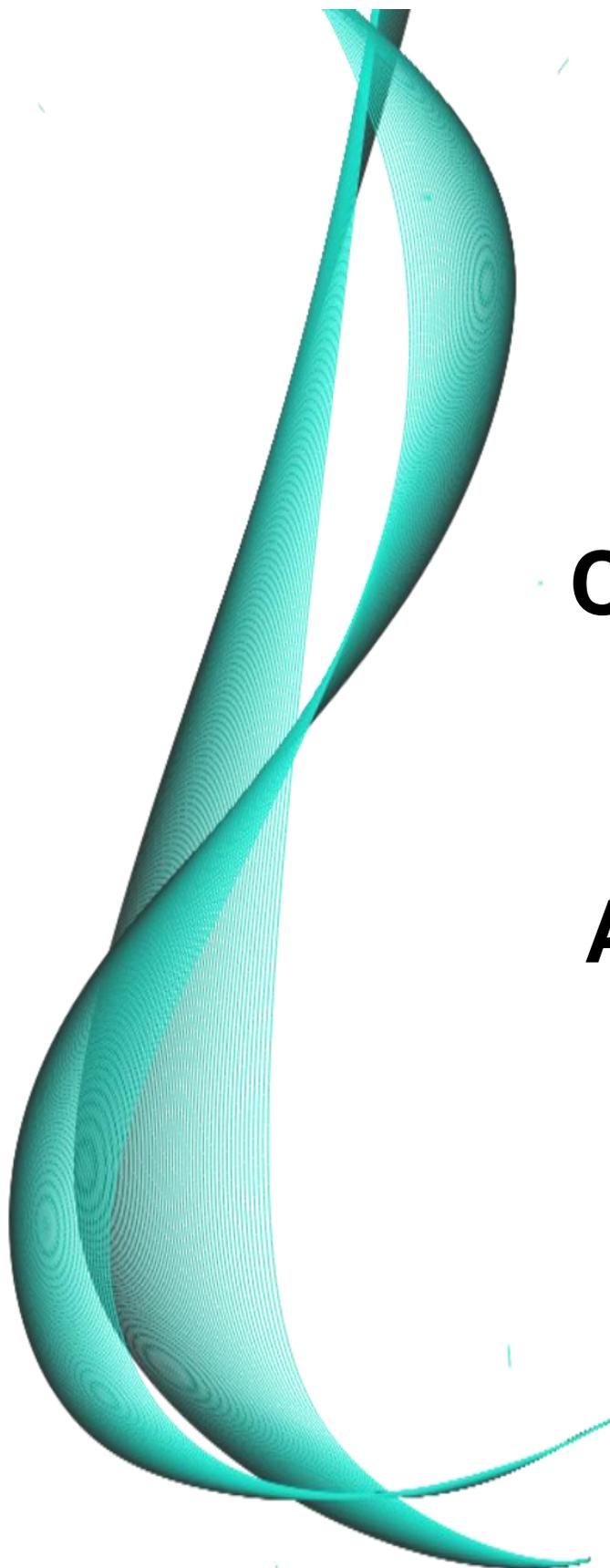
desarrollo de aplicaciones web quitándonos de encima toda la carga del diseño y dándonos un diseño elegante y bueno gracias a sus clases ya predefinidas.

### **2.20.6. Gestor De Base de Datos MySQL**

MySQL es un sistema gestor de bases de datos (SGBD, DBMS por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo.

Son muchas las razones para escoger MySQL como solución para la administración de datos:

- Coste: El coste de MySQL es gratuito para la mayor parte de los usos y su servicio de asistencia resulta económico.
- Asistencia: Existe una nutrida y activa comunidad MySQL.
- Facilidad de uso: MySQL resulta fácil de utilizar y de administrar. Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso.
- Velocidad: MySQL es mucho más rápido que la mayor parte de sus rivales.
- Funcionalidad: MySQL dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID, compatibilidad para la mayor parte de SQL ANSI, volcados online, duplicación, funciones SSL e integración con la mayor parte de los entornos de programación.
- Portabilidad: MySQL se ejecuta en la inmensa mayoría de sistemas operativos y, la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.



# **CAPÍTULO III**

## **MARCO**

### **APLICATIVO**

### 3. MARCO APLICATIVO

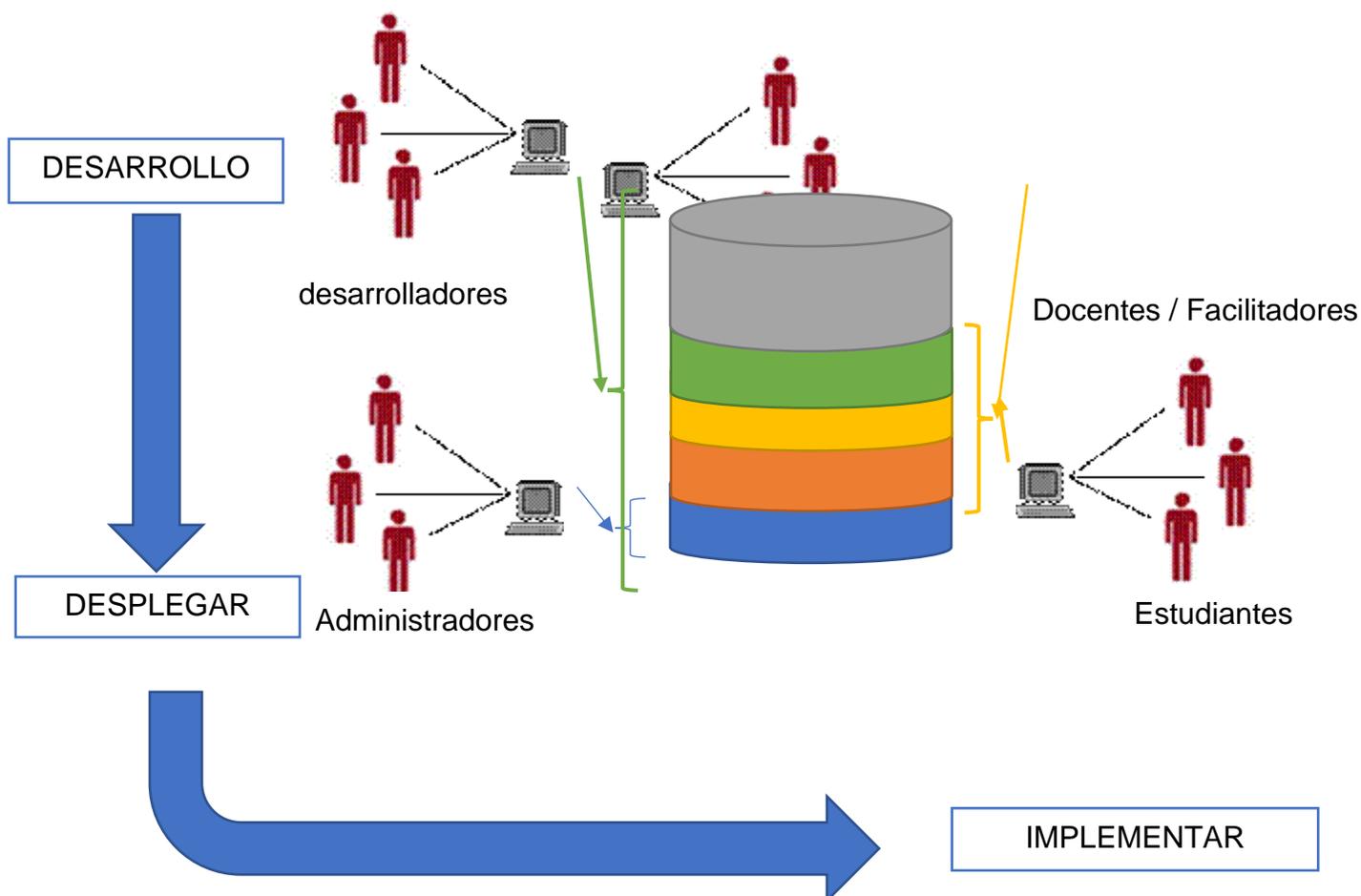
#### 3.1. INTRODUCCIÓN

En este capítulo se describe la forma de organización y métodos de trabajo del sistema, se hará uso de las metodologías y herramientas descritas anteriormente, las mismas que nos servirán en el desarrollo del sistema y todos sus módulos.

En la Tabla 8 se apreciar el modelo de proceso de desarrollo que se utilizó en el presente proyecto, en esta tabla se modela la combinación de la metodología SCRUMBAN, además de pasos que son importantes para la implementación del producto final.

#### 3.2. ESQUEMA DEL SISTEMA

Figura 5. Esquema de sistema



Fuente: elaboración propia

**Tabla 8. Tabla Scrumban**

HACER			Siguien e	HACIENDO			Por aprob ar	Sprint hech o
Storie s	backlo g	sprin t		Análisi s	Diseñ o	Implementació n		

Fuente: Elaboración propia

### 3.3. FASE PREGAME

Esta fase inicial es la más importante en scrum el objetivo del Pregame es el de construir el producto-backlog.

En caso de la metodología Kanban, la fase de exploración se identifican a los usuarios involucrados en el desarrollo del producto incluyendo los requerimientos que presentan, las tareas a realizar y la planeación del desarrollo del producto.

#### 3.3.1. Concepción y exploración

Esta fase se realizarán tres tareas de manera paulatinas para la obtención de requisitos los q se observarán.

**Tabla 9. Obtención de requisitos**

Tarea	Descripción de tareas
<b>Entrevistas personales</b>	Se realizaron las entrevistas a estudiantes que realizan sus cursos de capacitación de manera frecuente en TECHSBOL
<b>Observación</b>	Se observo las dificultades que tienen los estudiantes para llegar al centro de capacitación por la distancia que conlleva llegar a la ubicación de TECHSBOL.

	Además, se observó que no existía un control adecuado por los ingresos monetarios a TECHSBOL por dichos cursos además que se pretende vender cursos en línea.
<b>Documentación</b>	Se documentaron las observaciones y se genero un flujo de procesos para facilitar el estudio y toma de clases en TECHSBOL.

**Fuente: elaboración propia**

### 3.3.2. Roles SCRUMBAN

En la tabla 3.3 se mostrará las fases de trabajo en el tablero Kanban, el límite (WIP) y los responsables del equipo Kanban.

**Tabla 10. Fases de trabajo y responsables**

<b>Fase</b>	<b>Límite</b>	<b>Responsable</b>
<b>Pedido</b>	-	<b>Flow manager:</b> Rudy David Castillo
<b>Análisis</b>	3	<b>Analista:</b> Rudy David Castillo
<b>Diseño</b>	4	<b>Diseñador:</b> Rudy David Castillo
<b>Implementar</b>	3	<b>Desarrollador:</b> Rudy David Castillo
<b>Producción</b>	-	<b>Flow manager:</b> Rudy David Castillo

**Fuente: elaboración propia**

### 3.3.3. Arquitectura del software

La esencia de la arquitectura del software es la de dividir las funciones del sistema E-Learning en módulos de venta de cursos, administración e instructores de los mismos haciendo que los módulos se manejen y/o conecten al sistema de forma lógica.

El servidor Web proporcionara un catálogo de los cursos, incluyendo una orden de pedidos. Es decir, que el servidor se encarga de las ventas en modo contra pagó pudiendo existir en un periodo de mediano o largo plazo algunos métodos de pago en línea.

**Figura 6. Esquema de arquitectura**



**Fuente: elaboración propia**

En la figura 6 el estudiante puede ver una lista del catálogo de cursos y adquirirlas por un carito de compras el cual le emitirá un recibo de compra del curso por medio de un sistema de seleccionado el cual se presentará en un proceso de comprobación de pago del curso procediendo así a su habilitación en el mismo.

### 3.3.4. Obtención de requerimientos

A continuación, e la tabla 11 se mostrará cada de uno de los requerimientos que fueron descritos por el cliente.

**Tabla 11. Lista de requerimientos**

<b>Requerimientos</b>	<b>Descripción</b>
<b>Administración de estudiantes</b>	Crear, listar, modificar y eliminar registro de estudiantes
<b>Administración de cursos</b>	Crear, listar, modificar y eliminar registro de los cursos, una configuración básica de los cursos crear describir y subir archivos si son necesarios
<b>Clasificación por categorías de los cursos</b>	Crear, listar, modificar y eliminar registro de las categorías a la vez el permitir añadir una o más subcategorías a cada categoría.
<b>Administración de instructores</b>	Crear, listar, modificar y eliminar registro de Instructores
<b>Administración de inscripción</b>	Inscribir a un estudiante en un curso u eliminarlo del mismo

<b>Registro de estudiantes inscritos en cursos</b>	Se tendrá un historial de los estudiantes inscritos en cada curso
<b>Registro de instructores y cursos subidos</b>	Apartado de instructor para crear, listar, modificar y eliminar los cursos teniendo opciones básicas de configuración de los cursos
<b>Reporte de ingresos administrativos</b>	graficar los ingresos monetarios a la empresa, poder imprimirlos.
<b>Entorno gráfico amigable para el usuario</b>	El usuario deberá sentirse cómodo con el manejo del sistema

**Fuente: elaboración propia**

### 3.3.5. Producto-backlog

De acuerdo a la lista básica de requerimientos de la plataforma descrita en la tabla 9, se realizó el análisis correspondiente y en coordinación con el personal de TECHSBOL se construyó el producto-backlog como se observa en la tabla.

**Tabla 12. Producto backlog**

<b>Id</b>	<b>Nombre</b>	<b>Prioridad</b>	<b>Modulo</b>
R1	Diseño de registro, login y logout	Media	Módulo de registro y login de usuario
R2	Creación de controladores y desarrollo de crud	Alta	Módulo de registro y login de usuario
R3	Autenticación local	Alta	Módulo de registro y login de usuario
R4	Asignación de roles	Media	Módulo administración de cuentas de usuario
R5	Autenticación en línea con roles pre establecidos anteriormente	Alta	Módulo administración de cuentas de usuario
R6	Recuperación de contraseñas	Media	Módulo administración de cuentas de usuario

R7	Validar rol	Alta	Módulo administración de cuentas de usuario
R8	Validar credenciales	Media	Módulo administración de cuentas de usuario
R9	Creación de controladores y desarrollo de crud	Alta	Módulo de carrito de compras
R10	Creación de cursos	Alta	Módulo de carrito de compras
R11	Diseño de carritos de compras	Alta	Módulo de carrito de compras
R12	Adicionar cursos al carito	Alta	Módulo de carrito de compras
R13	Eliminar cursos al carrito	Alta	Módulo de carrito de compras
R14	Comprar curso	Alta	Módulo de carrito de compras
R15	Adicionar pagos PayPal	Media	Módulo de carrito de compras
R16	Generar recibo	Media	Módulo de carrito de compras
R17	Añadir catálogo de cursos	Alta	Módulo de administración de catálogo y cursos
R18	Eliminar catalogó de cursos	Alta	Módulo de administración de catálogo y cursos
R19	Búsqueda especializada	Media	Módulo de administración de catálogo y cursos

R20	Ver curso en detalle	Media	Módulo de administración de catálogo y cursos
R21	Ver cursos principales	Media	Módulo de administración de catálogo y cursos
R22	Enviar mensajes por correo	Media	Módulo de administración de catálogo y cursos
R23	Crud de cursos	Alta	Módulo de administración de catálogo y cursos
R24	Crud de evaluaciones en línea	Media	Módulo de administración de catálogo y cursos

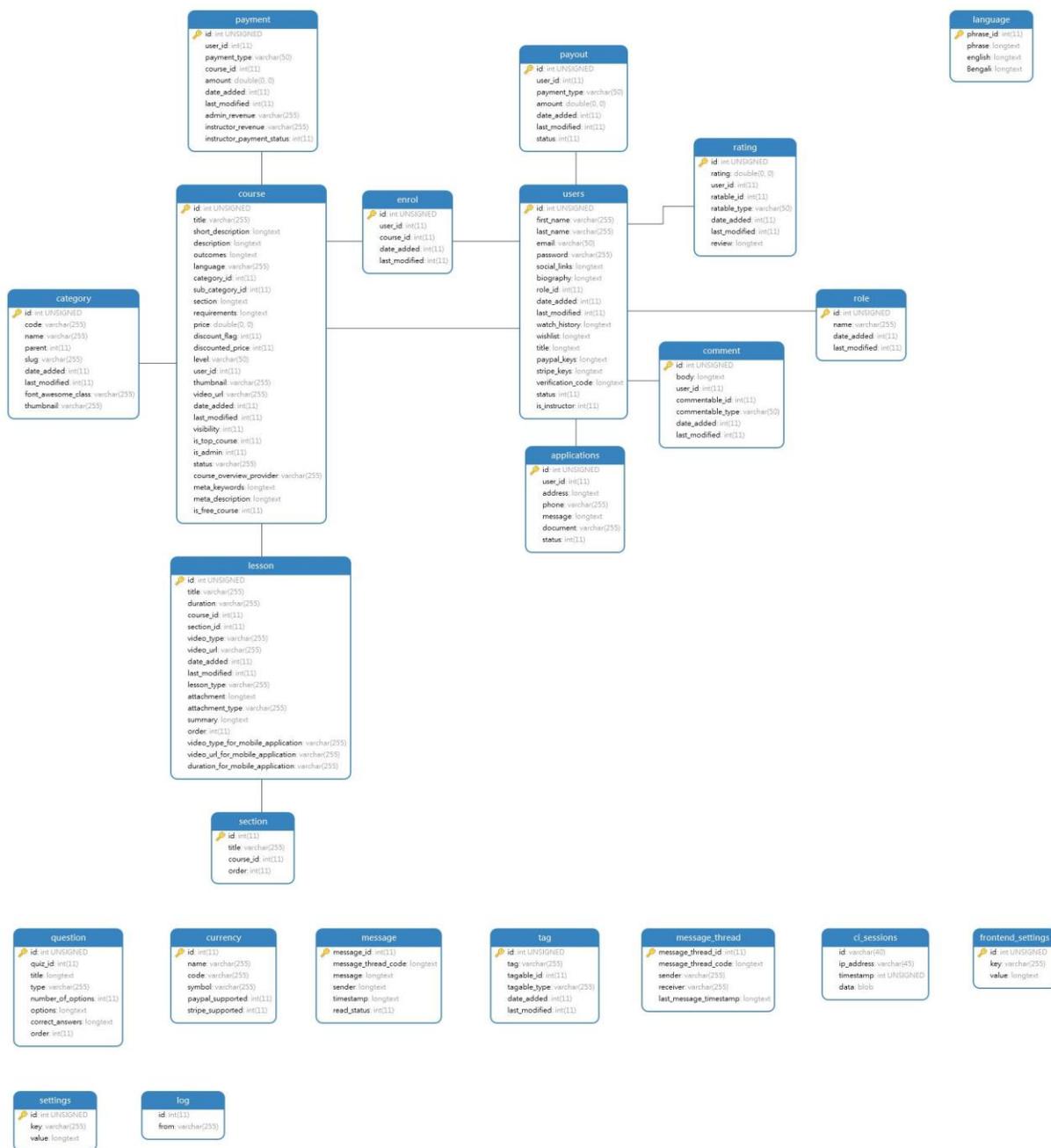
**Fuente: elaboración propia**

### 3.3.6. Inicialización

Esta fase será adoptada dentro de la fase Preamble para el desarrollo del sistema. Se considera dentro de la metodología Kanban, en la que se preparan y verifican asuntos críticos que involucrarán el desarrollo del sistema junto con la configuración para el proyecto.

Ya definido el producto-backlog como también la arquitectura de la plataforma, se procede al modelado de la base de datos, figura 9.

Figura 7. Base de datos



Fuente: elaboración propia

Después realizado el modelo relacional se obtuvo el diccionario de datos:

**Tabla 13. Aplicación**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id ( <i>Primaria</i> )	int(11)	No		
user_id	int(11)	Sí	NULL	
address	longtext	Sí	NULL	
phone	varchar(255)	Sí	NULL	
message	longtext	Sí	NULL	
document	varchar(255)	Sí	NULL	
status	int(11)	No	0	

**Tabla 14. Categoría**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id ( <i>Primaria</i> )	int(11)	No		
code	varchar(255)	Sí	NULL	
name	varchar(255)	Sí	NULL	
parent	int(11)	Sí	0	
slug	varchar(255)	Sí	NULL	
date_added	int(11)	Sí	NULL	
last_modified	int(11)	Sí	NULL	
font_awesome_class	varchar(255)	Sí	NULL	
thumbnail	varchar(255)	Sí	NULL	

**Tabla 15. Identificador de inicio de sesión**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id	varchar(40)	No		
ip_address	varchar(45)	No		

timestamp	int(10)	No	0	
data	blob	No		

**Tabla 16. Comentario**

	Tipo	Nulo	Predeterminado	Comentarios
id ( <i>Primaria</i> )	int(11)	No		
body	longtext	Sí	NULL	
user_id	int(11)	Sí	NULL	
commentable_id	int(11)	Sí	NULL	
commentable_type	varchar(50)	Sí	NULL	
date_added	int(11)	Sí	NULL	
last_modified	int(11)	Sí	NULL	

**Tabla 17. Curso**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id ( <i>Primaria</i> )	int(11)	No		
title	varchar(255)	Sí	NULL	
short_description	longtext	Sí	NULL	
description	longtext	Sí	NULL	
outcomes	longtext	Sí	NULL	
language	varchar(255)	Sí	NULL	
category_id	int(11)	Sí	NULL	
sub_category_id	int(11)	Sí	NULL	
section	longtext	Sí	NULL	
requirements	longtext	Sí	NULL	
price	double	Sí	NULL	
discount_flag	int(11)	Sí	0	

discounted_price	int(11)	Sí	NULL	
level	varchar(50)	Sí	NULL	
user_id	int(11)	Sí	NULL	
thumbnail	varchar(255)	Sí	NULL	
video_url	varchar(255)	Sí	NULL	
date_added	int(11)	Sí	NULL	
last_modified	int(11)	Sí	NULL	
visibility	int(11)	Sí	NULL	
is_top_course	int(11)	Sí	0	
is_admin	int(11)	Sí	NULL	
status	varchar(255)	Sí	NULL	
course_overview_provider	varchar(255)	Sí	NULL	
meta_keywords	longtext	Sí	NULL	
meta_description	longtext	Sí	NULL	
is_free_course	int(11)	Sí	NULL	

**Tabla 18. Inscripcion**

<b>Columna</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Predeterminado</b>	<b>Comentarios</b>
id ( <i>Primaria</i> )	int(11)	No		
user_id	int(11)	Sí	NULL	
course_id	int(11)	Sí	NULL	
date_added	int(11)	Sí	NULL	
last_modified	int(11)	Sí	NULL	

**Tabla 19. Lección**

Columna	Tipo	Nulo	Predeterminado	Comentarios
<i>id (Primaria)</i>	int(11)	No		
title	varchar(255)	Sí	NULL	
duration	varchar(255)	Sí	NULL	
course_id	int(11)	Sí	NULL	
section_id	int(11)	Sí	NULL	
video_type	varchar(255)	Sí	NULL	
video_url	varchar(255)	Sí	NULL	
date_added	int(11)	Sí	NULL	
last_modified	int(11)	Sí	NULL	
lesson_type	varchar(255)	Sí	NULL	
attachment	longtext	Sí	NULL	
attachment_type	varchar(255)	Sí	NULL	
summary	longtext	Sí	NULL	
order	int(11)	No	0	
video_type_for_mobile_application	varchar(255)	Sí	NULL	
video_url_for_mobile_application	varchar(255)	Sí	NULL	
duration_for_mobile_application	varchar(255)	Sí	NULL	

**Tabla 20. Mensajes**

Columna	Tipo	Nulo	Predeterminado	Comentarios
<i>id (Primaria)</i>	int(11)	No		
title	varchar(255)	Sí	NULL	
duration	varchar(255)	Sí	NULL	
course_id	int(11)	Sí	NULL	
section_id	int(11)	Sí	NULL	

video_type	varchar(255)	Sí	NULL	
video_url	varchar(255)	Sí	NULL	
date_added	int(11)	Sí	NULL	
last_modified	int(11)	Sí	NULL	
lesson_type	varchar(255)	Sí	NULL	
attachment	longtext	Sí	NULL	
attachment_type	varchar(255)	Sí	NULL	
summary	longtext	Sí	NULL	
order	int(11)	No	0	
video_type_for_mobile_application	varchar(255)	Sí	NULL	
video_url_for_mobile_application	varchar(255)	Sí	NULL	
duration_for_mobile_application	varchar(255)	Sí	NULL	

**Tabla 21. Ilo del mensaje**

Columna	Tipo	Nulo	Predeterminado	Comentarios
message_thread_id ( <i>Primaria</i> )	int(11)	No		
message_thread_code	longtext	Sí	NULL	
sender	varchar(255)	Sí		
receiver	varchar(255)	Sí		
last_message_timestamp	longtext	Sí	NULL	

**Tabla 22. Pagos a Instructores**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id ( <i>Primaria</i> )	int(11)	No		
user_id	int(11)	Sí	NULL	
payment_type	varchar(50)	Sí	NULL	
course_id	int(11)	Sí	NULL	

amount	double	Sí	NULL	
date_added	int(11)	Sí	NULL	
last_modified	int(11)	Sí	NULL	
admin_revenue	varchar(255)	Sí	NULL	
instructor_revenue	varchar(255)	Sí	NULL	
instructor_payment_status	int(11)	Sí	0	
transaction_id	varchar(255)	Sí	NULL	
session_id	varchar(255)	Sí	NULL	

**Tabla 23. Pagos de cursos**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id ( <i>Primaria</i> )	int(11)	No		
user_id	int(11)	Sí	NULL	
payment_type	varchar(50)	Sí	NULL	
course_id	int(11)	Sí	NULL	
amount	double	Sí	NULL	
date_added	int(11)	Sí	NULL	
last_modified	int(11)	Sí	NULL	
admin_revenue	varchar(255)	Sí	NULL	
instructor_revenue	varchar(255)	Sí	NULL	
instructor_payment_status	int(11)	Sí	0	
transaction_id	varchar(255)	Sí	NULL	
session_id	varchar(255)	Sí	NULL	

**Tabla 24. Cuestionarios**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id ( <i>Primaria</i> )	int(11)	No		
quiz_id	int(11)	Sí	NULL	

title	longtext	Sí	NULL	
type	varchar(255)	Sí	NULL	
number_of_options	int(11)	Sí	NULL	
options	longtext	Sí	NULL	
correct_answers	longtext	Sí	NULL	
order	int(11)	No	0	

**Tabla 25. Calificación del curso**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id ( <i>Primaria</i> )	int(11)	No		
rating	double	Sí	NULL	
user_id	int(11)	Sí	NULL	
ratable_id	int(11)	Sí	NULL	
ratable_type	varchar(50)	Sí	NULL	
date_added	int(11)	Sí	NULL	
last_modified	int(11)	Sí	NULL	
review	longtext	Sí	NULL	

**Tabla 26. Roles**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id ( <i>Primaria</i> )	int(11)	No		
name	varchar(255)	Sí	NULL	
date_added	int(11)	Sí	NULL	
last_modified	int(11)	Sí	NULL	

**Tabla 27. Sección de cursos**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id ( <i>Primaria</i> )	int(11)	No		
name	varchar(255)	Sí	NULL	
date_added	int(11)	Sí	NULL	
last_modified	int(11)	Sí	NULL	

**Tabla 28. Etiquetas**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id ( <i>Primaria</i> )	int(11)	No		
tag	varchar(255)	Sí	NULL	
tagable_id	int(11)	Sí	NULL	
tagable_type	varchar(255)	Sí	NULL	
date_added	int(11)	Sí	NULL	
last_modified	int(11)	Sí	NULL	

**Tabla 29. Usuarios**

Columna	Tipo	Nulo	Predeterminado	Comentarios
id ( <i>Primaria</i> )	int(11)	No		
first_name	varchar(255)	Sí	NULL	
last_name	varchar(255)	Sí	NULL	
email	varchar(50)	Sí	NULL	
password	varchar(255)	Sí	NULL	
social_links	longtext	Sí	NULL	
biography	longtext	Sí	NULL	
role_id	int(11)	Sí	NULL	
date_added	int(11)	Sí	NULL	
last_modified	int(11)	Sí	NULL	

watch_history	longtext	Sí	NULL	
wishlist	longtext	Sí	NULL	
title	longtext	Sí	NULL	
paypal_keys	longtext	Sí	NULL	
stripe_keys	longtext	Sí	NULL	
status	int(11)	Sí	NULL	
is_instructor	int(11)	Sí	0	

Previo al desarrollo, se configuro el ambiente de trabajo en los aplicativos para el desarrollo, en la siguiente tabla, se podrá observar la base de configuración y versiones utilizadas para el desarrollo.

**Tabla 30. Versiones de desarrollo**

Nº	Configuración	Versión
1	MariaDB	10.5.6
2	Php	7.4.12
3	Codeigniter	3
4	JQuery	3.5.1
5	Bootstrap	4.5.3
6	JavaScript	1.1

**Fuente: elaboración propia**

### 3.4. FASE DEVELOPMENT

En la siguiente fase se describe el desarrollo de la aplicación web, donde se describe cada de uno de los Sprint.

#### 3.4.1. Desarrollo de los Sprint

El proceso de desarrollo se sujetará a la presentación de cuatro sprints, que somn los siguientes:

- Primer spring: módulo de registro y login

- Segundo spring: módulo de administración de cuentas
- Tercer spring: modulo carrito e compras
- Cuarto spring: módulo de administración de catálogo y cursos.

### 3.5. PRIMER SPRING: MODULO DE REGISTRO Y LOGIN DE USUARIOS

#### 3.5.1. Etapa de análisis

La tabla muestra el sprint asignado para el modulo, que contiene tareas.

**Tabla 31. Primer sprint modulo de registro y login**

Sprint o Iteraciones		Inicio	Fin	Duración
		03/08/2020	08//08/2020	5
Nº	Tarea	Duración tarea		Estado
1	Interface y lógica de registro de usuario	3 horas		Completado
2	Interface y lógica de inicio de sesión	4 horas		Completado
3	Interface y lógica de cerrar sesión	2horas		Completado
4	Interface y lógica de recuperación de contraseña	5 horas		Completado
5	Implementación y desarrollo de recuperación de información	6 horas		Completado
6	Autenticación	1 día		Completado

**Fuente: elaboración propia**

#### 3.5.1.1. Diagrama de casos de uso

Se diseño el diagrama de casos de uso con los requisitos: interfeace y lógica de registroi de usuario, Interface y lógica de inicio de sesión, Interface y lógica de cerrar sesión, Interface y lógica de recuperación de contraseñas, Implementación y desarrollo de recuperación de información, Autenticación. El cual se visualisa en la figura 3.3 y en la tabla 3.22 se muestra especificaciones del caso de uso.

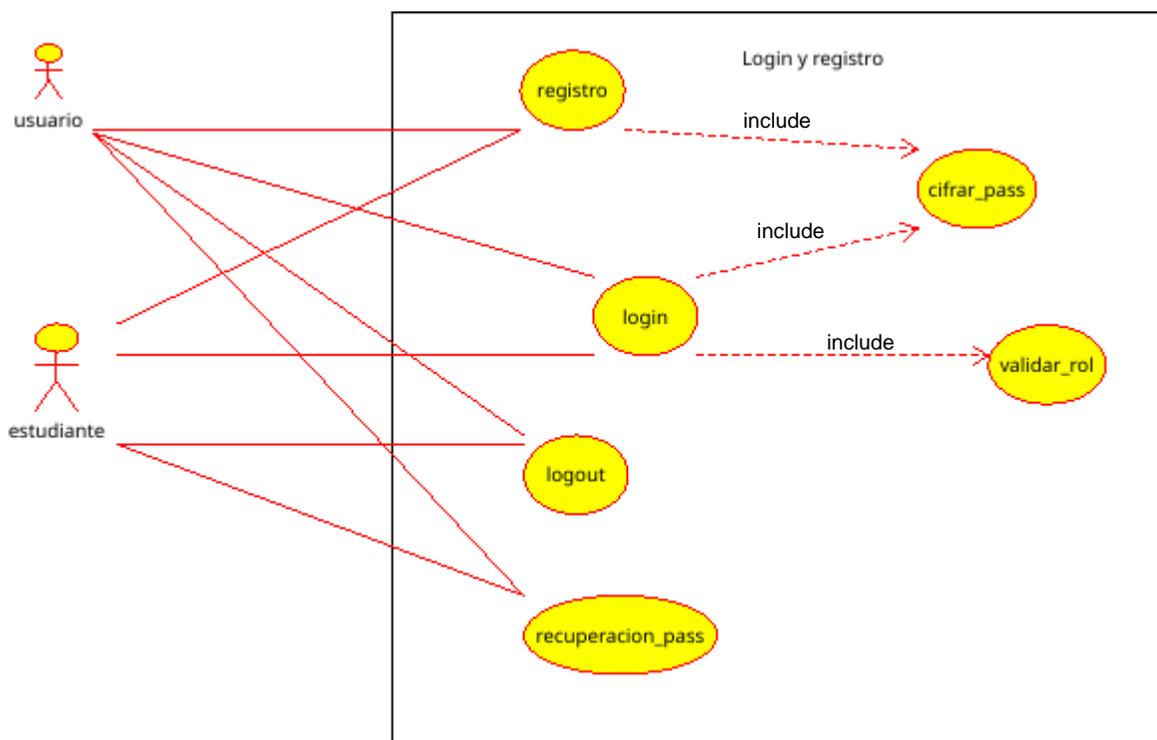


Figura 8. diagrama caso de uso de inicio de sesión y registro

Fuente elaboración propia

Tabla 32. especificación del caso de uso inicio de sesión y registro

<b>Caso de uso</b>	Caso de uso de inicio de sesión y registro
<b>Actores</b>	Administrador, estudiante, instructor.
<b>Descripción</b>	Permite al estudiante crear una cuenta en el sistema para su respectivo acceso al sistema
<b>Precondiciones</b>	Los actores deben contar con conexión a internet
<b>Flujo normal</b>	1) Los actores ingresan a la pagina principal.

	<p>2) El sistema accede a una nueva pagina por medio de un botón de registro.</p> <p>3) Se pide un correo electrónico y contraseña.</p> <p>4) el actor oprime el botón de crear y se logea.</p>
<b>Postcondiciones</b>	El sistema realiza la acción solicitada por el actor y muestra los cambios realizado.

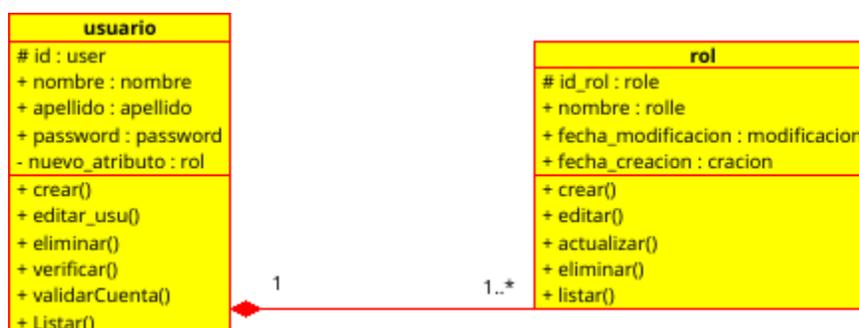
**Fuente: elaboración propia**

### 3.5.2. etapa de diseño

en esta fase de diseño se utilizó los diagramas de clases y de estado.

#### 3.5.2.1. Diagrama de clases

Se diseño el diagrama de clases para los requisitos: interface y lógica de registro de usuario, Interface y lógica de inicio de sesión, Interface y lógica de cerrar sesión, Interface y lógica de recuperación de contraseñas, Implementación y desarrollo, Autenticación el cual se visualiza en la figura.

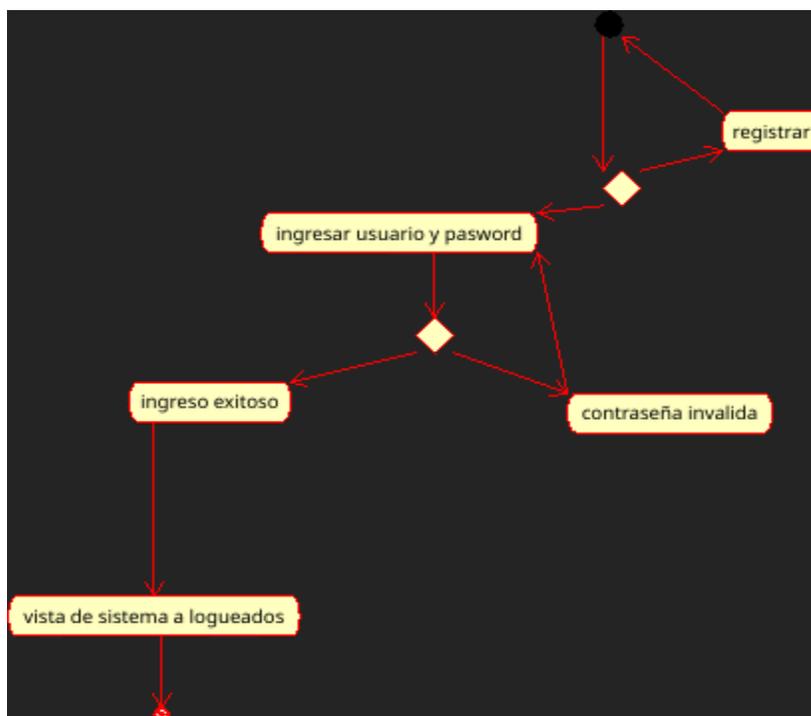


**Figura 9. diagrama de inicio de sesión y registro**

**Fuente elaboración propia**

### 3.5.2.2. Diagrama de estado

Se diseñó el diagrama de estado para los requisitos: interface y lógica de registro de usuario, Interface y lógica de inicio de sesión, Interface y lógica de cerrar sesión, Interface y lógica de recuperación de contraseñas, Implementación y desarrollo, Autenticación el cual se visualiza en la figura.



**Figura 10. diagrama de inicio de sesión y registro**

**Fuente elaboración propia**

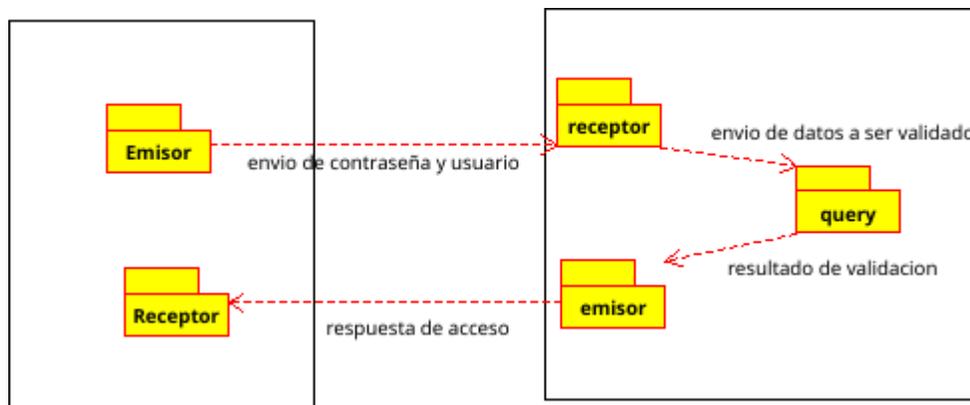
### 3.5.3. Etapa de implementación

En la etapa de diseño se utilizó los diagramas de secuencia y de componentes.

#### 3.5.3.1. Diagrama de secuencias

Se diseñó el diagrama de secuencias para los requisitos de login registro de usuarios y roles el cual se visualiza en la figura.





**Figura 12. diagrama de inicio de sesión y registro**

**Fuente elaboración propia**

#### **3.5.4. Resultados**

Para esto se desarrolló el inicio de sesión y registro como se puede observar en la captura de:

- Página de inicio de sesión figura 13
- Página de recuperación de contraseña figura 14
- Página de registro figura 15

Que se muestran capturas de pantalla.

### Iniciar sesión

Proporcione sus credenciales de inicio de sesión válidas.

✉ Correo electrónico:

🔒 Contraseña:

[Iniciar sesión](#)

[O olvidé la contraseña](#)  
[¿No tiene una cuenta? Regístrate](#)

**Figura 13. interface de inicio de sesión**

**Fuente elaboración propia**

### Se te olvidó tu contraseña

Proporcione su dirección de correo electrónico para obtener la contraseña.

✉ Correo electrónico:

Proporcione su dirección de correo electrónico para obtener la contraseña.

[Restablecer la contraseña](#)

[¿Quiero regresar? Iniciar sesión](#)

**Figura 14. interface de recuperación de contraseña**

**Fuente elaboración propia**

## Formulario de inscripción

Regístrate y empieza a aprender.

 Nombre de pila:

 Apellido:

 Correo electrónico:

 Contraseña:

Regístrate

[¿Ya tienes una cuenta? Iniciar sesión](#)

**Figura 15. interface de registro**

Fuente elaboración propia

### 3.6. Segundo spring: módulo de administración de cuentas

#### 3.6.1. Etapa de análisis

La tabla muestra el sprint asignado para este módulo que contiene tareas.

**Tabla 33. Administración de cuentas**

Sprint 2		Inicio	Fin	Duración
		10/08/2020	26/08/2020	16
Nº	Tarea	Duración tarea		Estado
1	Validar el registro del usuario directo lado cliente	2 días		Completado

2	Validar el registro del usuario directo lado cliente	2 horas	Completado
3	Solicitar confirmación de email con smtp	3 horas	Completado
4	Configuración del servidor para envío de correos electrónicos	2 días	Completado
5	Validar email con Ajax	4 días	Completado
6	Crud de usuarios	3 días	Completado

Fuente elaboración propia

### 3.6.1.1. Diagrama de casos de uso

Se diseñó el diagrama de clases para los requisitos: Validar el registro del usuario directo lado cliente, Validar el registro del usuario directo lado cliente, Solicitar confirmación de email con smtp, Configuración del servidor para envío de correos electrónicos, Validar email con Ajax, Crud de usuarios. El cual se visualisara en la figura.

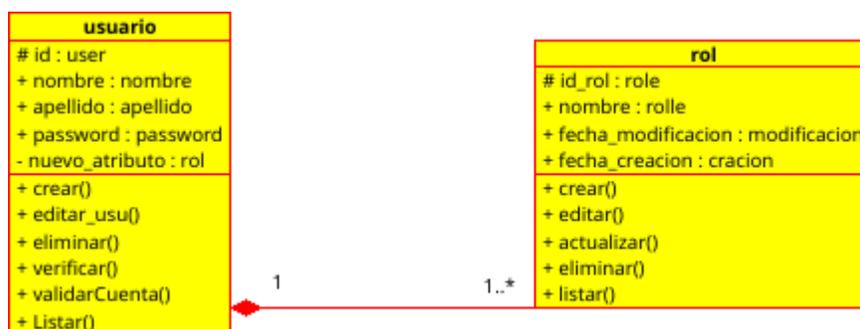


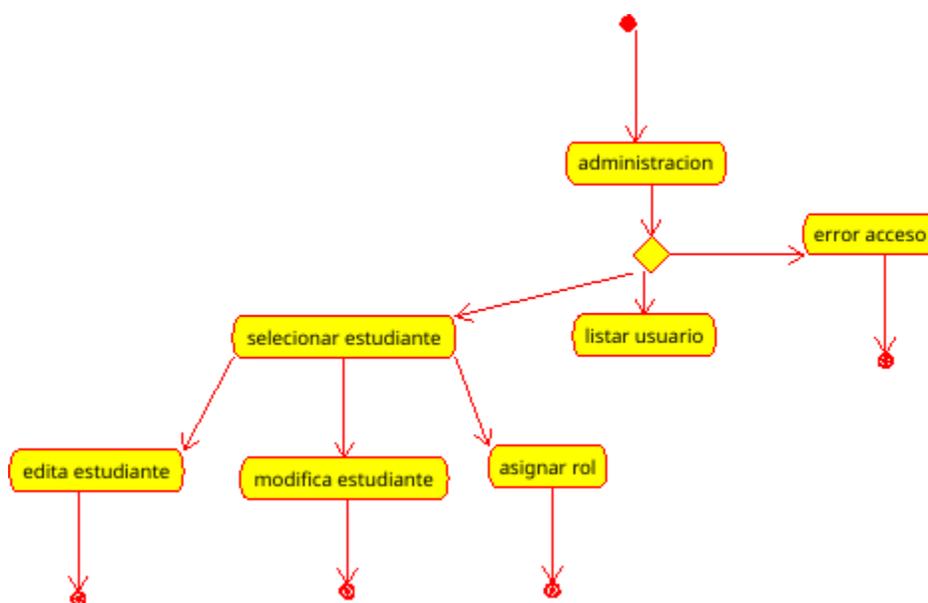
diagrama de estados del módulo de administración de usuario

Fuente elaboración propia

### 3.6.1.2. Diagrama de estados

Se diseñó el diagrama de clases para los requisitos: Validar el registro del usuario directo lado cliente, Validar el registro del usuario directo lado cliente, Solicitar confirmación de email con smtp, Configuración del servidor para envío de correos

electrónicos, Validar email con Ajax, Crud de usuarios. El cual se visualisara en la figura 3.12.



**Figura 16. diagrama de estados del módulo de administración de usuario**

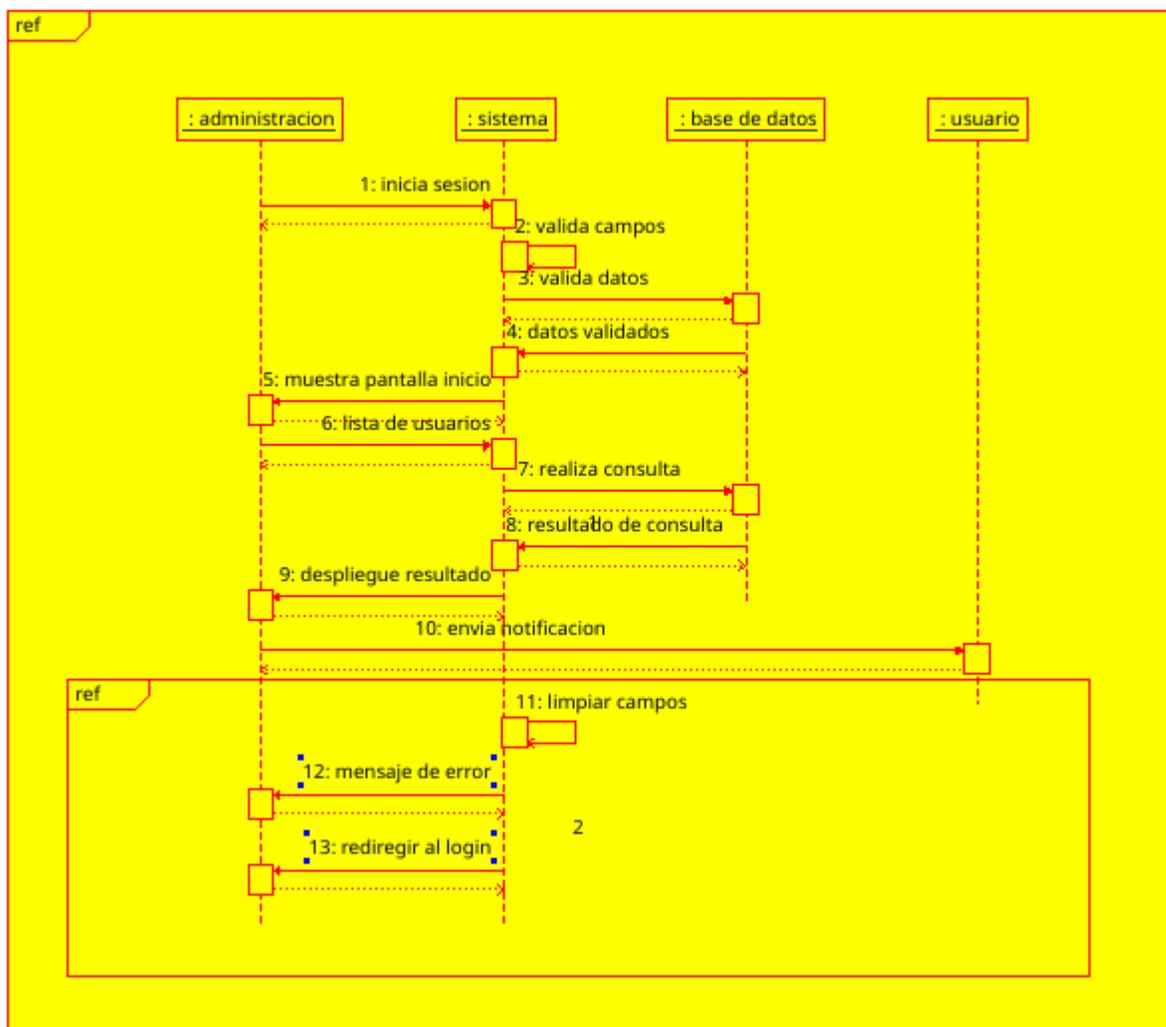
**Fuente elaboración propia**

### 3.6.2. Etapa de implementación

En la etapa de diseño se utilizó los diagramas de secuencia y de componentes.

#### 3.6.2.1. Diagrama de secuencias

Se diseño el diagrama de clases para los requisitos: Validar el registro del usuario directo lado cliente, Validar el registro del usuario directo lado cliente, Solicitar confirmación de email con smtp, Configuración del servidor para envío de correos electrónicos, Validar email con Ajax, Crud de usuarios. El cual se visualizará en la figura.

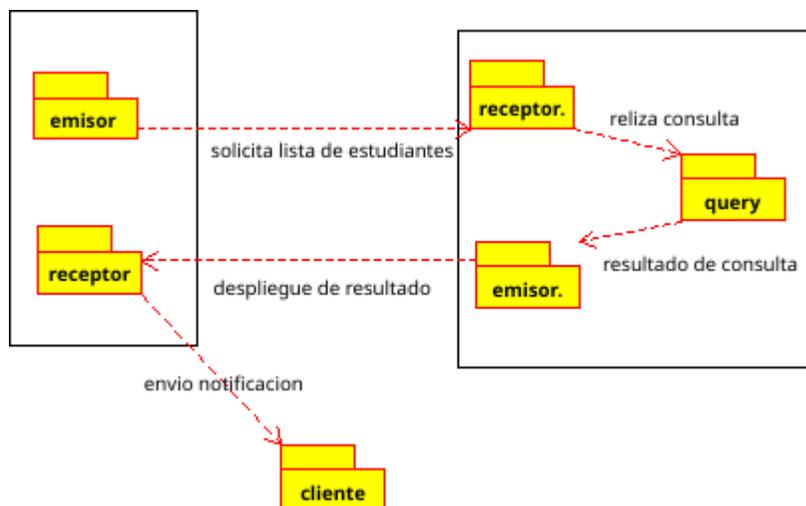


**Figura 17. diagrama de estados del módulo de administración de usuario**

**Fuente elaboración propia**

### 3.6.2.2. Diagrama de componentes

Se diseñó el diagrama de clases para los requisitos: Validar el registro del usuario directo lado cliente, Validar el registro del usuario directo lado cliente, Solicitar confirmación de email con smtp, Configuración del servidor para envío de correos electrónicos, Validar email con Ajax, Crud de usuarios. El cual se visualizará en la figura.



**Figura 18. diagrama de estados del módulo de administración de usuario**

**Fuente elaboración propia**

### 3.6.3. Resultado

Para esta parte se desarrolló el módulo de administración de usuarios, que se puedes observar en la figura que se muestra en la captura de pantalla.

ESTUDIANTE						
mostrar 10 entradas		Buscar: <input type="text"/>				
#	Foto	Nombre	Correo electrónico	Cursos inscritos	Comportamiento	
1		David Castillo	david@gmail.com			
2		rolando tola	rolando@gmail.com			

Mostrando 1 a 2 de 2 entradas

**Figura 19. plataforma de administración**

**Fuente elaboración propia**

### 3.7. Tercer Spring: modulo carrito de compras

#### 3.7.1. Etapa de análisis

**Tabla 34. Carrito de compras**

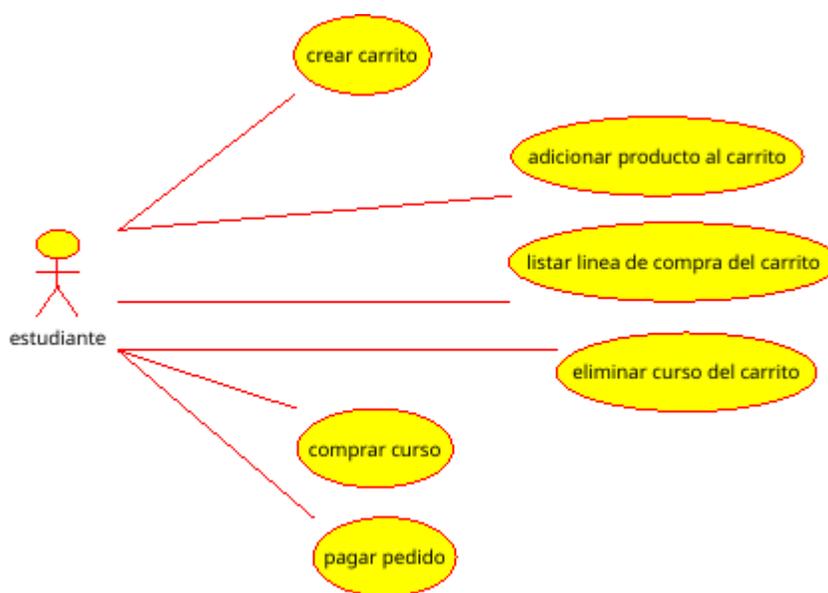
Sprint 3		Inicio	Fin	Duración
		26/08/2020	27/09/2020	32 días
Nº	Tarea	Duración de tarea	Estado	
1	Maquetación de carrito de compras	3 días	Completado	
2	Agregando los cursos con categorías en localStorage	1 día	Completado	
3	Hacer dinámico el carrito de compras	2 días	Completado	
4	Borrar cursos del carrito de compras	2 días	Completado	
5	Actualizar precio subtotal	1 día	Completado	
6	Actualizar la cesta cuando cambia la cantidad	3 días	Completado	
7	Agregar estilos al checkout	3 días	Completado	
8	Evaluar tasa de impuesto	4 días	Completado	
9	Evaluar total de la compra	1 día	Completado	
10	Pasarelas de pagos	6 días	Completado	
11	Aprobación de compras	2 días	Completado	
12	Creación de ítem de comentarios después de haber concluido el curso	3 días	Completado	

**Fuente: elaboración propia**

##### 3.7.1.1. Diagrama de casos de uso

Se diseñó el diagrama de casos de uso para los requisitos: Maquetación de carrito de compras, Agregando los cursos con categorías en localStorage, Hacer dinámico el

carrito de compras, Borrar cursos del carrito de compras, Adicionar cursos al carrito el cual se visualiza en la figura.



**Figura 20. diagrama de casos de uso del módulo del carrito**

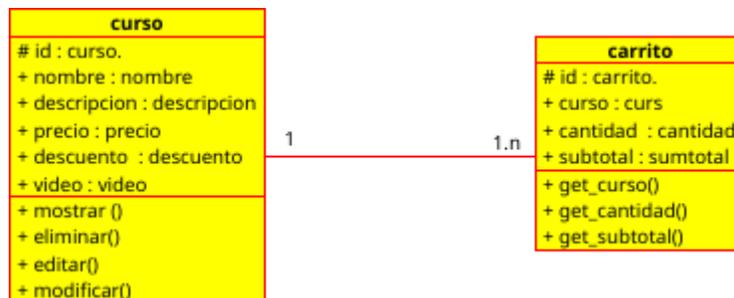
**Fuente elaboración propia**

### **3.7.2. Etapa de diseño**

En la fase de diseño se utilizó los diagramas de clases y de estado.

#### **3.7.2.1. Diagrama de clases**

Se diseño el diagrama de casos de uso para los requisitos: Maquetación de carrito de compras, Agregando los cursos con categorías en localStorage, Hacer dinámico el carrito de compras, Borrar cursos del carrito de compras, Adicionar cursos al carrito el cual se visualiza en la figura.

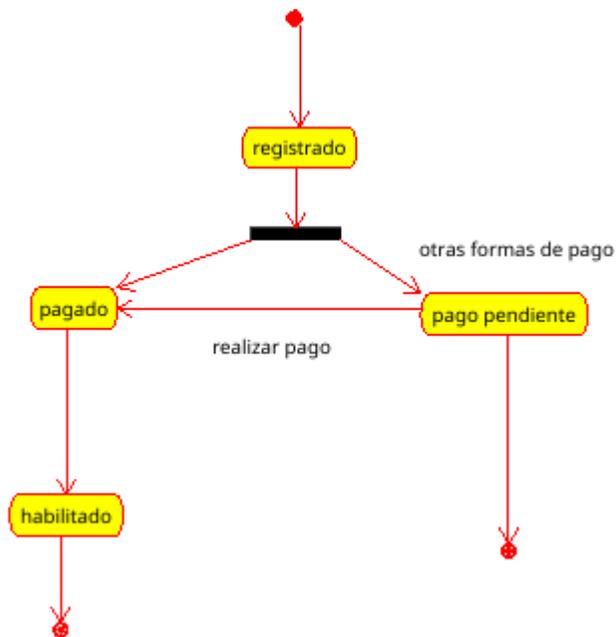


**Figura 21. diagrama de clases del módulo del carrito**

Fuente elaboración propia

### 3.7.2.2. Diagrama de estado

Se diseñó el diagrama de casos de uso para los requisitos: Maquetación de carrito de compras, Agregando los cursos con categorías en localStorage, Hacer dinámico el carrito de compras, Borrar cursos del carrito de compras, Adicionar cursos al carrito el cual se visualiza en la figura.



**Figura 22. diagrama de estado del módulo del carrito**

Fuente elaboración propia

### 3.7.3. Etapa de implementación

En la fase de diseño se utilizó los diagramas de clases y de estado.

#### 3.7.3.1. Diagrama de secuencia

Se diseñó el diagrama de casos de uso para los requisitos: Maquetación de carrito de compras, Agregando los cursos con categorías en localStorage, Hacer dinámico el carrito de compras, Borrar cursos del carrito de compras, Adicionar cursos al carrito el cual se visualiza en la figura.

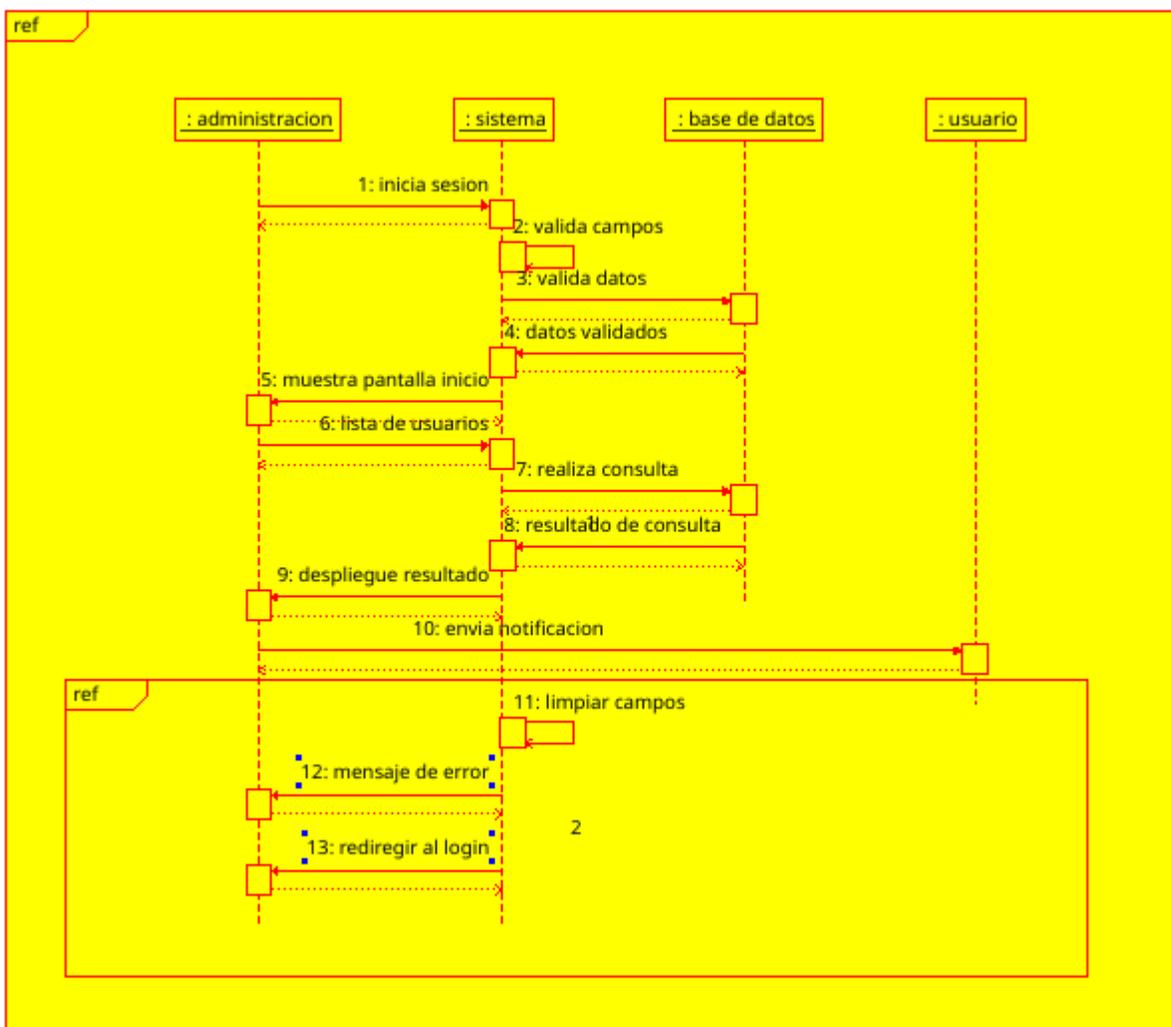
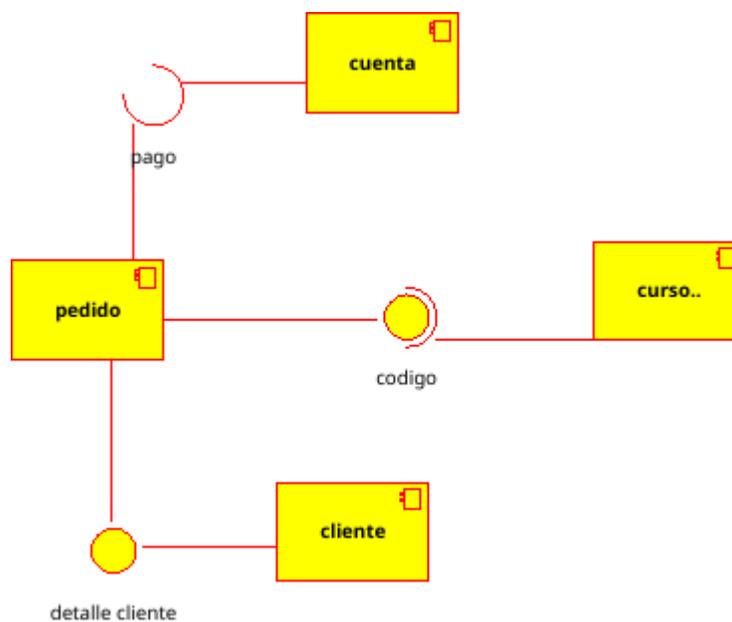


Figura 23. diagrama de secuencia del módulo del carrito

Fuente elaboración propia

### 3.7.3.2. Diagrama de componentes

Se diseñó el diagrama de casos de uso para los requisitos: Maquetación de carrito de compras, Agregando los cursos con categorías en localStorage, Hacer dinámico el carrito de compras, Borrar cursos del carrito de compras, Adicionar cursos al carrito el cual se visualiza en la figura.

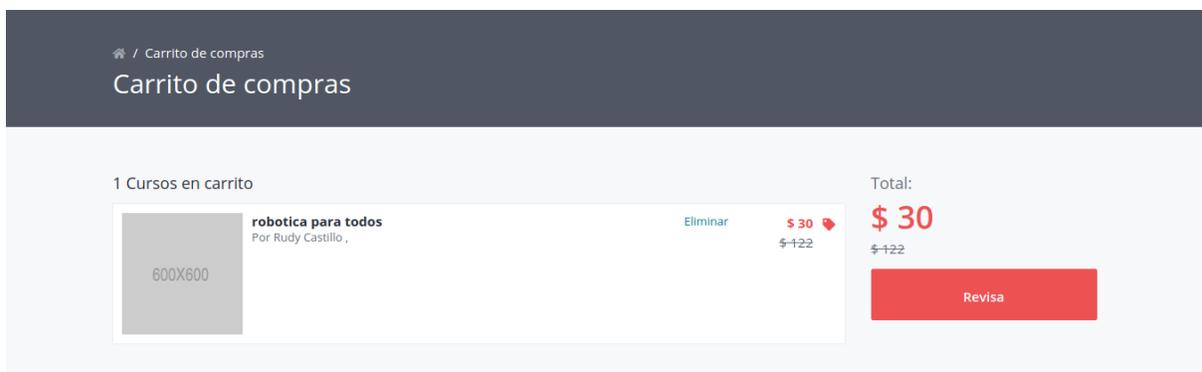


**Figura 24. diagrama de secuencia del módulo del carrito**

**Fuente elaboración propia**

### 3.7.4. Resultado

Para esta parte del desarrollo el módulo de carrito, se puede observar en figura q se muestra la captura de pantalla.



**Figura 25. diagrama de secuencia del módulo del carrito**

**Fuente elaboración propia**

### 3.8. Cuarto Sprint: Módulo De Administración De Catálogo Y Cursos

#### 3.8.1. Etapa de análisis

La tabla muestra el spring asignado para este módulo, mostrando las tareas que tiene.

**Tabla 35. Administración de catalogo de cursos**

Sprint 4		Inicio	Fin	Duración
		01/10/2020	31/10/2020	31 días
Nº	Tarea	Duración de tarea		Estado
1	Maquetación de la vitrina de cursos y catalogo	4 días		Completado
2	maquetación de los cursos en forma rejilla	4 días		Completado
3	Datos dinámicos de los cursos	4 días		Completado
4	Diseño e implementación del buscador	2 días		Completado
5	Creación de vistas previas del curso	5 días		Completado
6	Maquetar sección de comentarios	3 días		Completado

7	Maquetación de la pagina de cursos con datos dinámicos	5 dias	Completado
8	Diseño e implementación de estudiantes por curso	4 dias	Completado

Fuente elaboración propia

### 3.8.1.1. Diagrama de casos de uso

Se diseño el diagrama de casos de uso para los requisitos: Maquetación de la vitrina de cursos y catálogo, maquetación de los cursos en forma rejilla, Datos dinámicos de los cursos, Diseño e implementación del buscador, Creación de vistas previas del curso, Maquetar sección de comentarios, Maquetación de la página de cursos con datos dinámicos, Diseño e implementación de estudiantes por curso, la cual se visualizará en la figura.

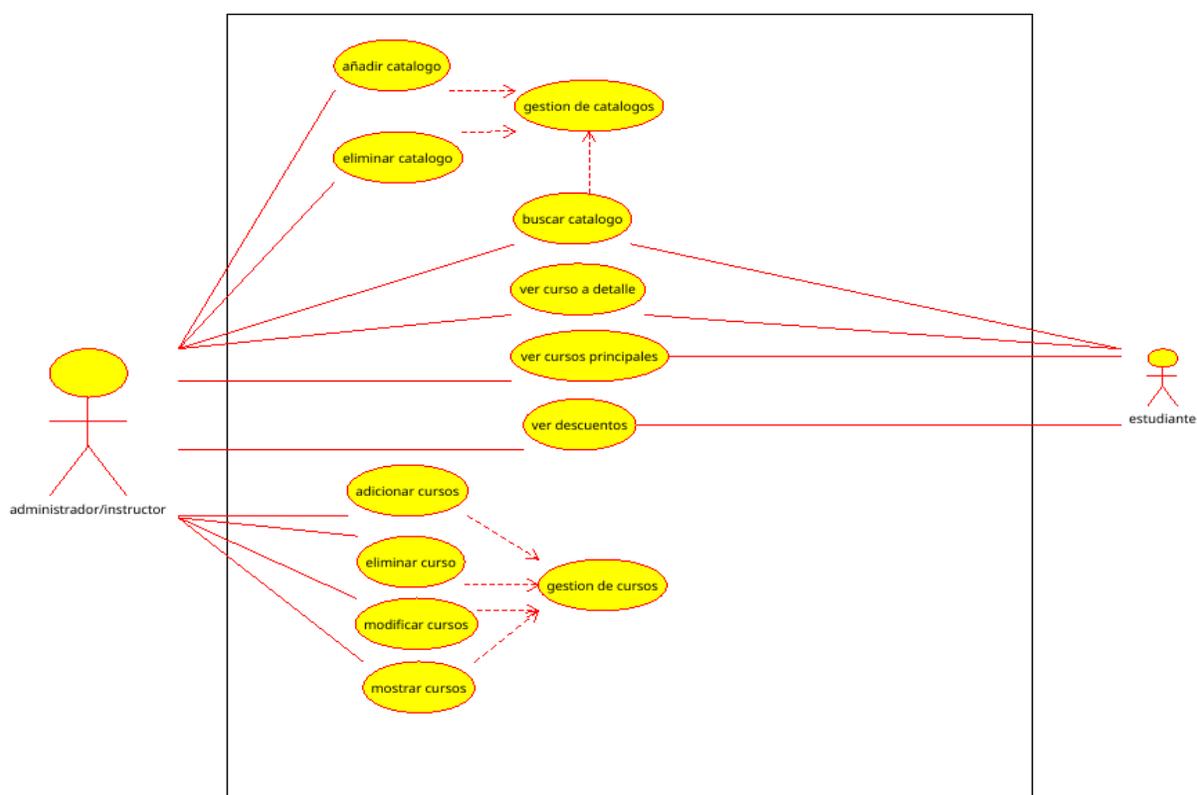


Figura 26. diagrama de caso de uso (include) del módulo del carrito

Fuente elaboración propia

### 3.8.2. Etapa de diseño

En la fase de diseño se utilizó los diagramas de clases y estados.

#### 3.8.2.1. Diagrama de clases

Se diseño el diagrama de clases para los requisitos: Maquetación de la vitrina de cursos y catálogo, maquetación de los cursos en forma rejilla, Datos dinámicos de los cursos, Diseño e implementación del buscador, Creación de vistas previas del curso, Maquetar sección de comentarios, Maquetación de la página de cursos con datos dinámicos, Diseño e implementación de estudiantes por curso, la cual se visualizará en la figura.

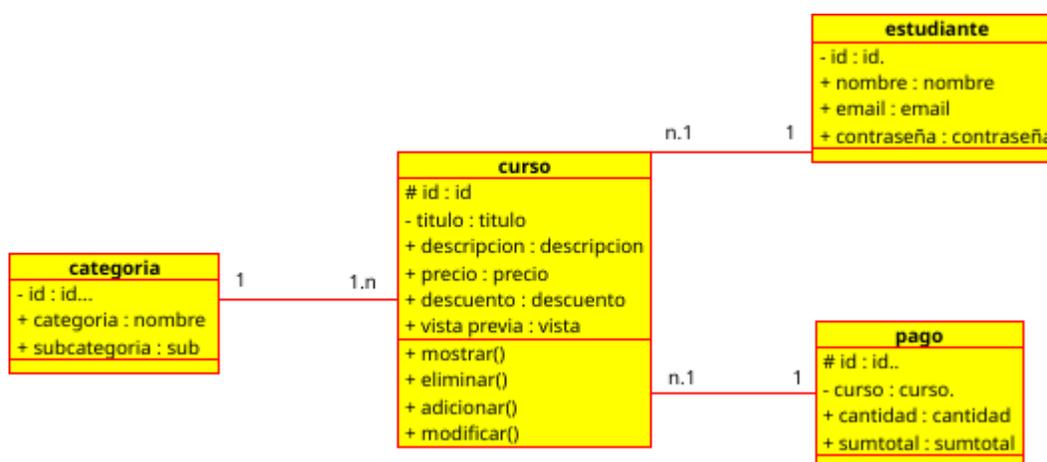
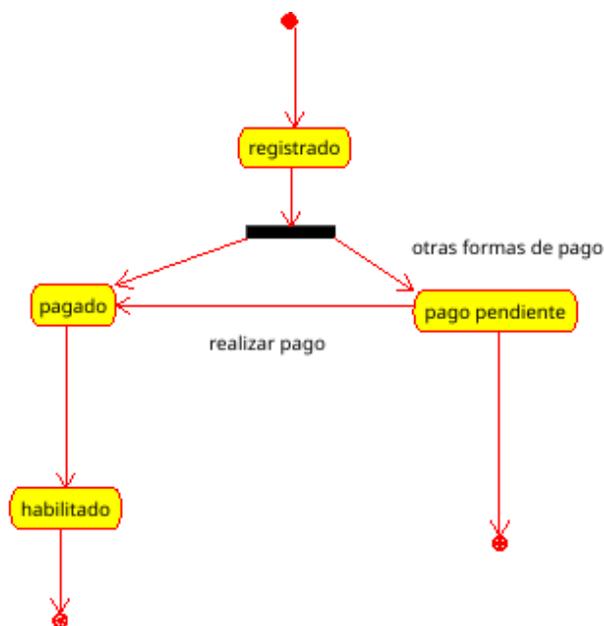


Figura 27. diagrama de clases del módulo del carrito

Fuente elaboración propia

#### 3.8.2.2. Diagrama de estado

Se diseño el diagrama de estado para los requisitos: Maquetación de la vitrina de cursos y catálogo, maquetación de los cursos en forma rejilla, Datos dinámicos de los cursos, Diseño e implementación del buscador, Creación de vistas previas del curso, Maquetar sección de comentarios, Maquetación de la página de cursos con datos dinámicos, Diseño e implementación de estudiantes por curso, la cual se visualizará en la figura.



**Figura 28. diagrama de estado del módulo del carrito**

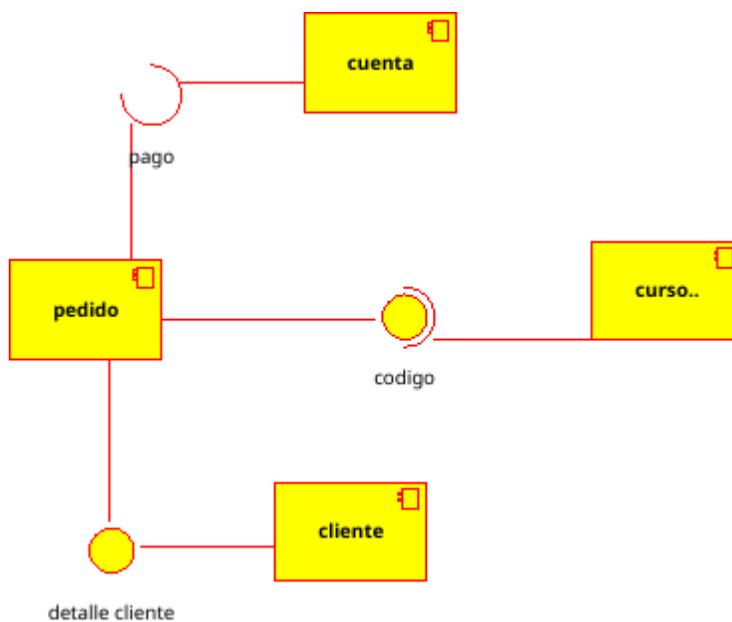
**Fuente elaboración propia**

### 3.8.3. Etapa de implementación

En la fase de diseño se utilizó los diagramas de secuencia y de componentes.

#### 3.8.3.1. Diagrama de componentes

Se diseñó el diagrama de estado para los requisitos: Maquetación de la vitrina de cursos y catálogo, maquetación de los cursos en forma rejilla, Datos dinámicos de los cursos, Diseño e implementación del buscador, Creación de vistas previas del curso, Maquetar sección de comentarios, Maquetación de la página de cursos con datos dinámicos, Diseño e implementación de estudiantes por curso, la cual se visualizará en la figura.



**Figura 29. diagrama de componentes del módulo del carrito**

**Fuente elaboración propia**

#### **3.8.4. Resultados**

Para esta parte se desarrolló el módulo de administración de catálogo, se puede observar en las figuras presentes.



Figura 30. Pantalla bienvenida

Fuente elaboración propia

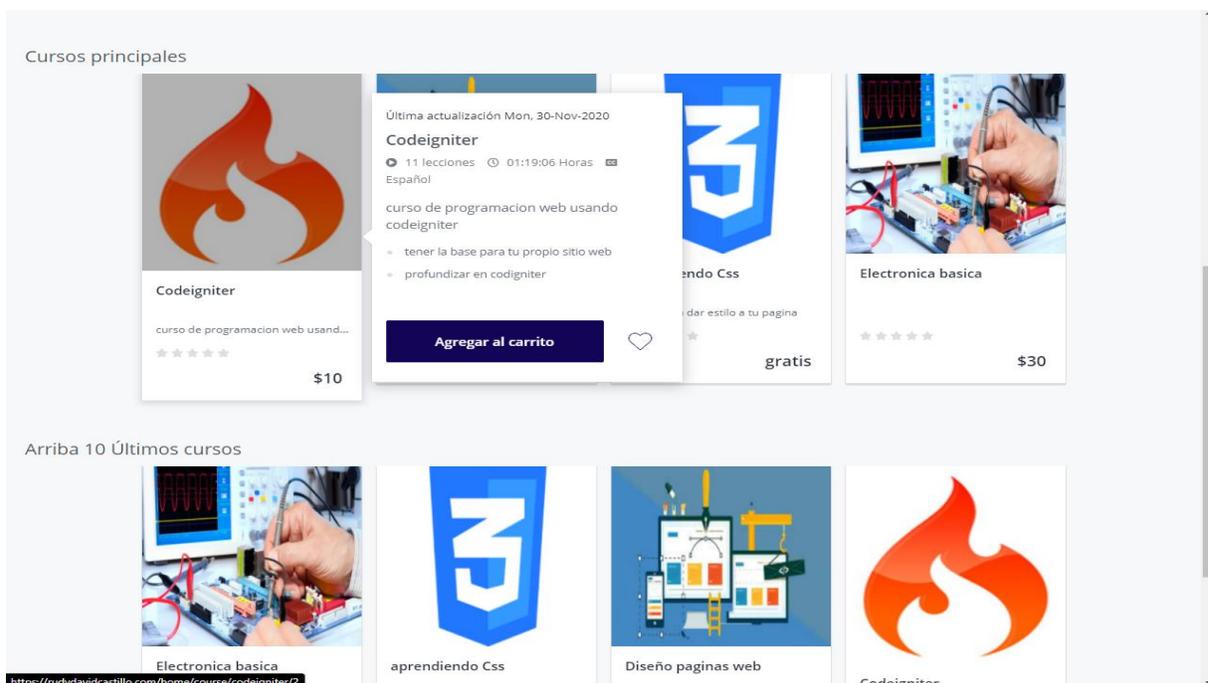


Figura 31. Pantalla de cursos

Fuente elaboración propia

The screenshot shows a course preview page for 'Diseño paginas web' on the 'cademy' platform. The page is divided into several sections:

- Header:** Includes the 'cademy' logo, navigation links for 'Cursos' and 'Buscar cursos', a search icon, a shopping cart icon with '0' items, and buttons for 'Iniciar sesión' and 'Registrarse'.
- Course Title and Description:** The title is 'Diseño paginas web' with the subtitle 'en aqui aprenderas lo basico del diseño de paginas web'. It is labeled as 'PRINCIPIANTE' (beginner) and has 0 ratings from 3 students. It was created by Rudy Castillo and last updated on Mon, 30-Nov-2020 in Spanish.
- Video Preview:** A large video player showing a preview of the course content, with a play button in the center and the text 'Vista previa de este curso' below it.
- Price and Purchase Options:** The price is '\$100'. There are three buttons: 'Agregar a la lista de deseos' (Add to wishlist), 'Compre ahora' (Buy now), and 'Agregar al carrito' (Add to cart).
- Includes:** A list of what is included in the course: 01:53:45 Hours Videos a pedido, 11 lecciones, Acceso de por vida completo, and Acceso en dispositivos móviles y TV.
- Learning Objectives:** A section titled 'Qué aprenderé?' (What I will learn) with two items: 'creacion paginas web basicas' and 'base para diseños front end', both with a checkmark and the text 'superación propia' (self-achievement).
- Course Plan:** A section titled 'Plan de estudios para este curso' (Course plan for this course) showing 11 lessons and a total duration of 01:53:45 Hours. The lessons are:
 

Lesson	Duration
- Iniciando	01:14:18 Horas
• Etiquetas Básicas (HTML)	00:09:43
• Estilos de textos	00:15:18
• Insertar fondos e imagenes	00:09:26
• crear tablas	00:13:53
• material de apoyo	
• creando botones y enlaces	00:15:14

**Figura 32. Pantalla vista previa curso**

**Fuente elaboración propia**

The image shows a web browser window displaying the login page of a website named 'Academy'. The page has a dark blue header with the 'Academy' logo on the left, a search bar with the text 'Cursos' and 'Buscar cursos', and two buttons on the right: 'Iniciar sesión' and 'Registrarse'. Below the header, there is a dark blue navigation bar with a home icon and the text 'Inicio sesión' and 'Usuario registrado'. The main content area is a light gray box with a white background. It features a title 'Iniciar sesión' and a subtitle 'Proporcione sus credenciales de inicio de sesión válidas.'. There are two input fields: 'Correo electrónico:' and 'Contraseña:'. Below the input fields is a green button labeled 'Iniciar sesión'. At the bottom of the form, there is a link 'o Olvidé mi contraseña' and another link 'No tengo una cuenta? Registrarse'. The footer of the page contains the 'Academy' logo, the text 'Dev. Rudy David Castillo', and a series of links: 'Acerca de', 'Política de privacidad', 'Términos y condiciones', and 'Inicio sesión'.

academy Cursos Buscar cursos Iniciar sesión Registrarse

Inicio sesión Usuario registrado

**Iniciar sesión**  
Proporcione sus credenciales de inicio de sesión válidas.

✉ Correo electrónico:  
Correo electrónico

🔒 Contraseña:  
Contraseña

**Iniciar sesión**

o Olvidé mi contraseña  
No tengo una cuenta? Registrarse

academy Dev. Rudy David Castillo Acerca de Política de privacidad Términos y condiciones Inicio sesión

**Figura 33. Pantalla de login**

**Fuente elaboración propia**

The screenshot displays the 'Mis cursos' (My courses) page on the Academy website. The page features a navigation bar with the Academy logo, 'Cursos', and a search bar. Below the navigation bar, there are tabs for 'Cursos', 'Listas de deseos', 'Mensajes', 'Historial de compras', and 'Perfil'. The main content area includes a filter section with 'Filtrar por' and 'Categories' dropdown, and a search bar for 'Buscar mis cursos'. Four course cards are displayed in a grid:

- aprendiendo Css**: 100% Completado, 5 stars rating, 'Editar calificación' button.
- Diseño paginas web**: 10% Completado, 5 stars rating, 'Editar calificación' button.
- Codeigniter**: 0% Completado, 5 stars rating, 'Editar calificación' button.
- Electronica basica**: 0% Completado, 5 stars rating, 'Editar calificación' button.

Each course card has a 'Detalles del curso' and 'Comenzar lección' button at the bottom.

**Figura 34. Pantalla de cursos adquiridos**

**Fuente elaboración propia**

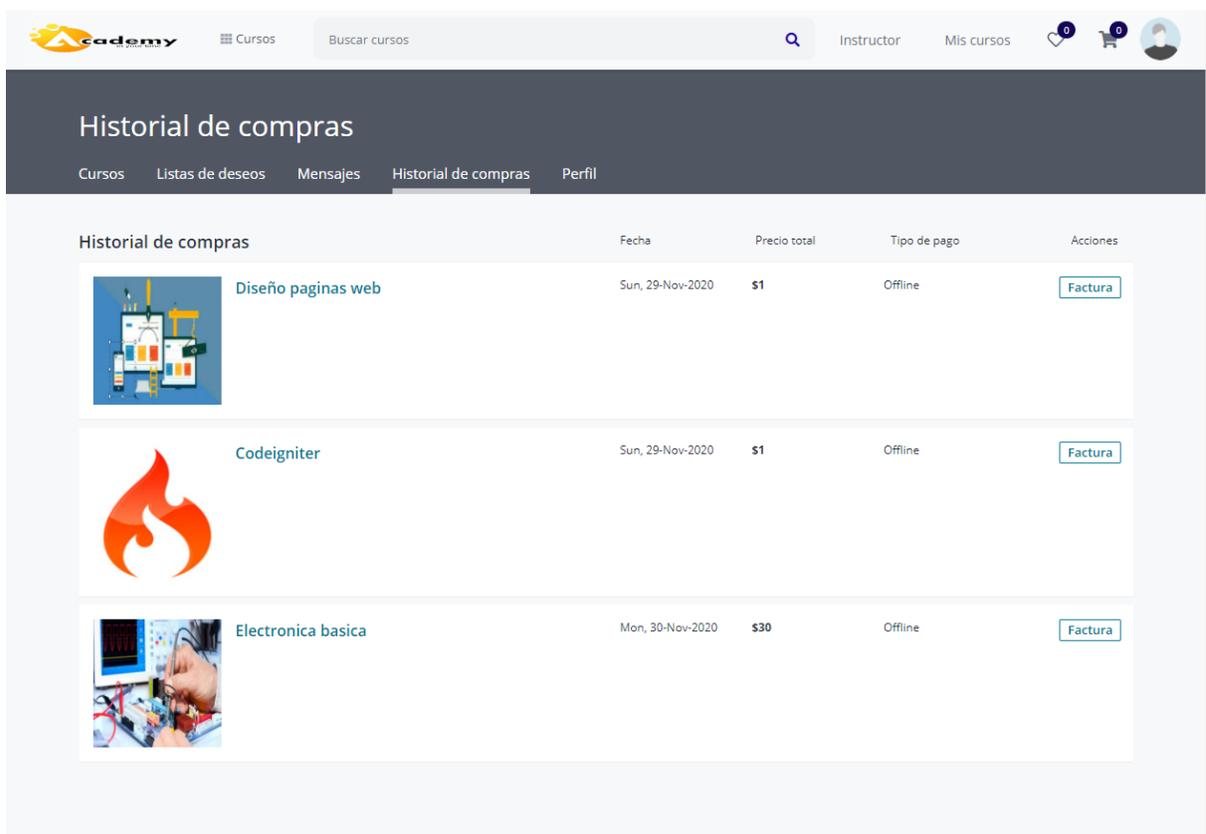
The screenshot shows a video player interface for a course titled 'CURSO DE CSS'. The video content features the CSS logo, the text 'CURSO DE CSS', and a large number '1'. The video player has a progress bar at the bottom showing '-09:02' and various control icons. To the right of the video player is a sidebar titled 'Contenido del curso' (Course Content) with a 'lecciones' (lessons) tab. Under 'Sección 1 primeros pasos', there is a list of 10 lessons, each with a checkmark and a duration:

- 1: Introduccion (10 Min)
- 2: Primer paso (8 Min)
- 3: Selectores (11 Min)
- 4: Colores (9 Min)
- 5: Textos (8 Min)
- 6: Id & clases (19 Min)
- 7: Contenedores (14 Min)
- 8: modelado de cajas (16 Min)
- 9: tipos de elemento (18 Min)
- 10: Seudo-Clases

Below the video player, there is a 'Nota:' (Note) section with the text: 'empecemos con intro de lo que es css'.

**Figura 35. Pantalla avance de curso**

**Fuente elaboración propia**



Historial de compras		Fecha	Precio total	Tipo de pago	Acciones
	Diseño paginas web	Sun, 29-Nov-2020	\$1	Offline	<a href="#">Factura</a>
	Codeigniter	Sun, 29-Nov-2020	\$1	Offline	<a href="#">Factura</a>
	Electronica basica	Mon, 30-Nov-2020	\$30	Offline	<a href="#">Factura</a>

**Figura 36. Pantalla de cursos adquiridos**

**Fuente elaboración propia**

### **3.9. Postgame y fase de estabilización, pruebas y reparaciones**

En esta fase se finaliza el desarrollo de la plataforma, para este cometido se realizan pruebas necesarias y se darán solución a las mismas asegurando la calidad del producto desarrollado utilizando pruebas de aceptación del producto

### 3.9.1. PRUEBAS Y REPARACIONES

Al realizar las pruebas a los sistemas, se encontraron errores y observaciones, los cuales se describen en la imagen.

BIBLIOTECA VULNERABLE	SE DETECTÓ UNA VERSIÓN VULNERABLE
<b>M</b> bootstrap	4.0.0
<b>M</b> bootstrap	4.0.0
<b>M</b> bootstrap	4.0.0
<b>M</b> jquery	3.2.1

**Figura 37. Vulnerabilidades de bibliotecas**

### 3.9.2. Pruebas de caja negra

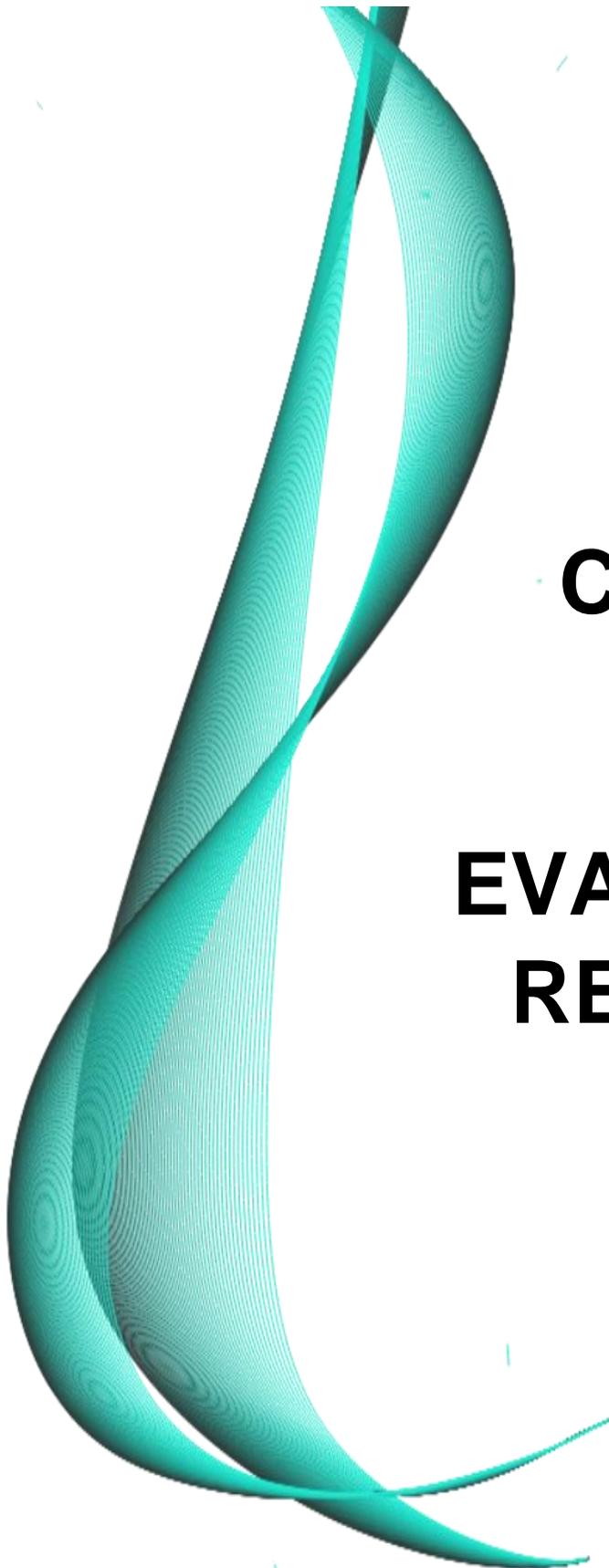
Las pruebas de caja negra se centran principalmente en lo que se quiere de un módulo, específica de un software, es decir, es una manera de encontrar casos específicos en ese modulo que atiendan a su especificación

Las pruebas de caja negra son, ni más ni menos que, pruebas funcionales dedicadas a “mirar” en el exterior de lo que se prueba y pruebas unitarias. Estas pruebas se denominan de varias formas, pruebas de caja “opaca”, pruebas de entrada/salida, pruebas inducidas por datos...los sinónimos son muchos y muy variados. En el presente proyecto las pruebas de caja negra se la realizan bajo los siguientes niveles:

- Pruebas unitarias, se realiza al momento del desarrollo de cada uno de los módulos del sistema verificando su funcionalidad.
- Pruebas de integración, esta prueba se realiza cuando todos los módulos están diseñados y desarrollados, para su posterior prueba en general.



**Figura 38. Prueba general de sistema**



# **CAPÍTULO IV**

## **PRUEBAS EVALUACION Y RESULTADOS**

## **4. PRUEBAS DE EVALUACIÓN Y RESULTADO**

### **4.1. INTRODUCCIÓN**

En este capítulo se hará análisis posterior al desarrollo e implementación del **SISTEMA DE EDUCACION VIRTUAL (E-LEARNING) COMO HERRAMIENTA DE ENSEÑANAZA CASO: TECHSBOL**, en este análisis se comprobará la calidad del software, seguridad y costo mediante análisis.

### **4.2. CALIDAD DE SOFTWARE**

Con el desarrollo de aplicaciones cada vez más complejas orientadas a la web se ha hecho necesario adoptar metodologías de desarrollo de software especialmente enfocadas a este medio, siempre teniendo como objetivo esencial la calidad, Web-Site QEM, define un enfoque integral, sistemático y cuantitativo para evaluar y comparar productos Web, tanto en la fase operativa como en la fase de desarrollo del ciclo de vida Web.

Según Olcina 1999, Web-Site QEM, es esencialmente integral, flexible y robusto, y cubre la mayor parte de las actividades en el proceso de evaluación, comparación, y selección de artefactos Web

Una de las metas principales de la evaluación y comparación de calidad de artefactos Web, radica en comprender el grado de cumplimiento de un conjunto de características y sub características con respecto a los requerimientos de calidad establecidos.

#### **4.2.1. Definiendo metas de evaluación**

Como detalla Olsina 1999, en esta fase deben definirse y refinar las metas y el alcance del proceso de evaluación. Para la evaluación del presente sistema web se ha escogido la fase operativa, la meta principal consiste en “comprender la calidad global de un sitio web desde el punto de vista del visitante”.

#### **4.2.2. Especificaciones requerimientos de calidad**

Como especifica Olsina 1999, en esta fase los evaluadores deben acordar y especificar las características, sub-características y atributos de calidad agrupándolas en un árbol de requerimientos. Respecto de las características de calidad de más alto nivel, se sigue la misma clasificación conceptual que la prescrita en el estándar ISO [ISO/IEC 9126]. Estas características de alto nivel son: usabilidad, funcionalidad, confiabilidad, eficiencia, portabilidad, y mantenibilidad, donde se deben seleccionar el subconjunto de características de primer nivel, conforme a las metas y el perfil de usuario seleccionado, en presente caso se dejaron fuera las características de Mantenibilidad y Portabilidad por la poca relevancia que tienen para el perfil seleccionado.

En la siguiente tabla puede apreciarse el árbol de requerimientos de calidad. Mencionar que para la construcción de este árbol se toma como base las características, sub-características y atributos citados por Olsina 1999, que es un modelo básicamente estándar para utilizarlo en distintos dominios

#### **4.2.3. Especificación de características de calidad**

Para la medida de calidad se especificarán a continuación las características de usabilidad, funcionalidad, confiabilidad, eficiencia y mantenibilidad.

##### **4.2.3.1. Usabilidad**

Es una característica de calidad de producto de alto nivel, que se la puede medir mediante cálculo a partir de métricas directas e indirectas, representa la capacidad o potencialidad del producto para ser utilizado, comprendido y operado por los usuarios, además de ser atractivo para cualquiera.

El criterio de evaluación es un criterio binario, discreto y absoluto. Solo se pregunta si está disponible representado por 1 y si no está disponible con un 0. Según Olsina de 1999 para evaluar la usabilidad se debe considerar las siguientes características

- **Comprensibilidad global del sitio:** Es una característica que representa a todas aquellas facilidades que permiten la audiencia, tener una rápida comprensión tanto de la estructura organizativa, como el contenido del sitio web como un todo, facilitando el rápido acceso y recorrido del mismo con sus componentes. Por tal razón, los atributos y sub-características se hallan principalmente en la página principal o en los primeros niveles del sitio.

**Tabla 36. Comprensibilidad global del sitio**

Característica: Compresibilidad Global del Sitio		
Nº		Resultado
1.	Esquema organizacional global	0.8
1.1.	Mapa del sitio	1
1.2.	Tabla de contenido	1
1.3.	Índice	1
2.	Calidad del sistema de etiquetado	1
3.	Visita guía orientada al usuario	0.8
4.	Mapa de imagen	1
TOTAL		0.94

**Fuente elaboración propia**

- **Aspectos de interfaces y estéticos:** Son factores y elementos relativos a la interacción del usuario, enfocados a un entorno o dispositivos concretos cuyo resultado es la generación de una percepción positiva o negativa de dicho servicio, producto o dispositivo. El diseño de los elementos de la interfaz debe facilitar la interacción del usuario con la funcionalidad, debe generar y formalizar documentos comprensibles, interactivos, navegables y facilitar su visualización.

**Tabla 37. Aspectos de interfaces y estéticos**

Característica: Aspectos de Interfaces y Estéticos		
Nº	Sub características	resultados
1	Agrupar los objetos del menú	1
2	Permanencia y estabilidad en la presentación	1
2.1	Estabilidad	1
3	Aspectos de estilo	1
3.1	Uniformidad en colores de enlaces	1
3.2	Uniformidad en estilo global	1
4	Preferencia estética	1
Total		1

- **Misceláneas:** Este atributo modela el número de lenguajes extranjeros soportados por un sitio (sitios de dominios de aplicación de índole académica, museos, comercio electrónico y otros). Además, especifica el nivel de soporte de cada lenguaje: Total (todas las páginas del sitio), parcial (algunos sub-sitios del sitio), o mínimo (algunas páginas o documentos de algunos sub-sitios). No se computa obviamente el lenguaje nativo del sitio web.

**Tabla 38. Misceláneas**

Misceláneas		
Nº	Sub-características	resultado
1	Soporte de lenguaje extranjero	1
2	Sección de presentación nuevos	1
total		1

**Fuente elaboración propia**

La usabilidad de la aplicación se determinará por el promedio de las anteriores características, como se muestra en la siguiente tabla.

**Tabla 39. TOTAL, USABILIDAD**

Total, de Usabilidad		
Nº	criterio	resultado
1	Comprensibilidad global del sitio	0.94
2	Aspectos de interfaces y estéticos	1
3	Misceláneas	1
Total		0.98

**Fuente elaboración propia**

#### **4.2.3.2. Funcionalidad**

Para determinar la calidad de la funcionalidad de la aplicación se debe analizar la búsqueda y exploración de contenidos. El criterio de evaluación es un criterio binario, discreto y absoluto. Solo se pregunta si está disponible representado por 1 y si no está disponible con un 0. Según Olsina de 1999 para evaluar la funcionalidad se debe considerar las siguientes características:

- Aspectos de búsqueda y recuperación: Es una característica que modela el mecanismo que permite tener un modo directo de encontrar información.

**Tabla 40. Aspectos de búsqueda y recuperación**

Característica: Aspectos de Búsqueda y Recuperación		
Nº	Sub características	Resultado
1	Mecanismo de búsqueda en el sitio web	1
1.1	Búsqueda por filtros	1
1.1.1	Curso gratis	1
1.1.2	Cursos de paga	1
1.1.3	Cursos por categoría	1
1.2	Búsqueda global	1
2	Opción de recuperación	1
2.1	Nivel de personalización	1
2.2	Retroalimentación en la recuperación	1
total		1

**Fuente elaboración propia**

- Aspectos de dominio orientados al usuario: Es la información que el usuario puede tener y la información de aprobación de operaciones realizadas por este.

**Tabla 41. Aspectos de dominio orientados al usuario**

Característica: Aspectos de Dominio Orientados al Usuario		
Nº	Sub categorías	Resultado
1.	Importancia del contenido	1
1.1	Información de los cursos	1
1.2	Información de los costes	1
2.	Servicio on-line	1
Total		1

**Fuente elaboración propia**

- La funcionalidad de la aplicación se determinará por el promedio de las anteriores características, como se muestra en la siguiente tabla.

**Tabla 42. TOTAL, FUNCIONALIDAD**

Total Funcionalidad		
Nº	Criterio	Resultado
1	Aspectos de búsqueda y recuperación	1
2	Aspectos de navegación y exploración	0.98

3	Aspectos del dominio orientados al usuario	1
Total		0.99

**Fuente elaboración propia**

#### 4.2.3.3. CONFIABILIDAD

La medición de esta característica está definida por el complemento de los casos de deficiencia encontrados en la aplicación. El criterio elemental es uno de variable normalizada, continuo y absoluto; en donde si BL = Número de enlaces rotos encontrados. TL = Número total de enlaces del sitio. La fórmula para computar la variable es:

$$X = 100 - \left( BL * \frac{100}{TL} \right) * 10$$

Donde, si  $X < 0$  entonces  $X = 0$ .

- No deficiencia: Este atributo representa básicamente a los enlaces encontrados que conducen a nodos destinos ausentes (también llamados enlaces ausentes o pendientes).

**Tabla 43. Evaluación de confiabilidad**

Característica: Confiabilidad		
Nº	Sub-característica/s	resultado
1.	No deficiencia	1
1.1	Errores de enlace	0
1.1.1	Enlaces rotos	0
1.1.2	Enlaces inválidos	0
1.1.3	Enlaces no implementados	0
1.2	Errores o deficiencias varias	0
1.2.1	Permanecía de los controles contextuales	0
1.2.2	Deficiencias o resultados inesperados	0
1.2.3	Nodo destino en construcción	0
Confiabilidad total		1

**Fuente elaboración propia**

#### 4.2.3.4. EFICIENCIA

Es una característica de calidad de producto de alto nivel que se la puede medir mediante cálculo a partir de métricas directas e indirectas y principalmente representa a la relación entre el grado de performance del artefacto y la cantidad de recursos (tiempo, espacio, etc.) usados bajo ciertas condiciones. El criterio de evaluación es un criterio binario, discreto y absoluto. Solo se pregunta si está disponible representado por 1 y si no está disponible con un 0. Según Olsina de 1999 para evaluar la eficiencia se debe considerar las siguientes características:

- Desempeño: Se mide el tamaño de todas las páginas (estáticas) del sitio web considerando todos sus componentes gráficos, tabulares y contextuales. El tamaño de cada página se especifica como una función del tiempo de espera y de la velocidad mínima establecida para una línea de comunicación.

**Tabla 44. Evaluación de desempeño**

Característica: Desempeño		
Nº	Sub características	Resultado
1	Páginas de acceso rápido	1
Total		1

**Fuente elaboración propia**

#### 4.3. SEGURIDAD

Los problemas más frecuentes de seguridad en una aplicación web, suelen venir de la configuración de las herramientas que son utilizadas para el desarrollo, también pueden venir de un posible fallo en el diseño lógico.

Entre los problemas más comunes se encuentran.

- Ingreso de usuarios no valido.
- Control de Acceso roto.
- Administración de sesión y autenticación rota.
- Inyecciones de código.
- Manejo de errores inadecuados.
- Administración de seguridad insegura.

### 4.3.1. TIPOS DE SEGURIDAD

Existen cuatro tipos de seguridad en los sistemas web:

- Seguridad en el cliente.
- Seguridad en el servidor
- Seguridad en la comunicación.
- Seguridad en la aplicación

#### 4.3.1.1. Seguridad en el cliente

El mecanismo de seguridad viene siendo la validación por el lado del cliente. Mecanismo que se encarga de validar la información antes de llegar al servidor.

#### 4.3.1.2. Seguridad en el servidor

Es necesaria la realización de controles por lado del servidor de la aplicación y el servidor de la base de datos. Generalmente un servidor de aplicaciones proporciona varios servicios que son innecesarios para el funcionamiento de la aplicación, que pueden producir problemas de seguridad, para el presente caso se tomó en cuenta la desactivación de las opciones no utilizadas para el funcionamiento del servidor. En cuanto a la seguridad en el servidor de base de datos, el mayor problema son las inyecciones SQL siendo un ataque realizado en la consulta de la base de datos. En PHP, usamos la función **mysql\_real\_escape\_string ()** para evitar esto junto con otras técnicas, pero CodeIgniter proporciona funciones y bibliotecas incorporadas para evitar esto. Podemos evitar la inyección SQL en CodeIgniter de las siguientes tres formas:

- Consultas de escape
- Consulta de consultas
- Clase de registro activo

#### 4.3.1.3. Seguridad en la comunicación

Como se hizo la utilización del framework CodeIgniter que tiene las siguientes características en cuanto a seguridad.

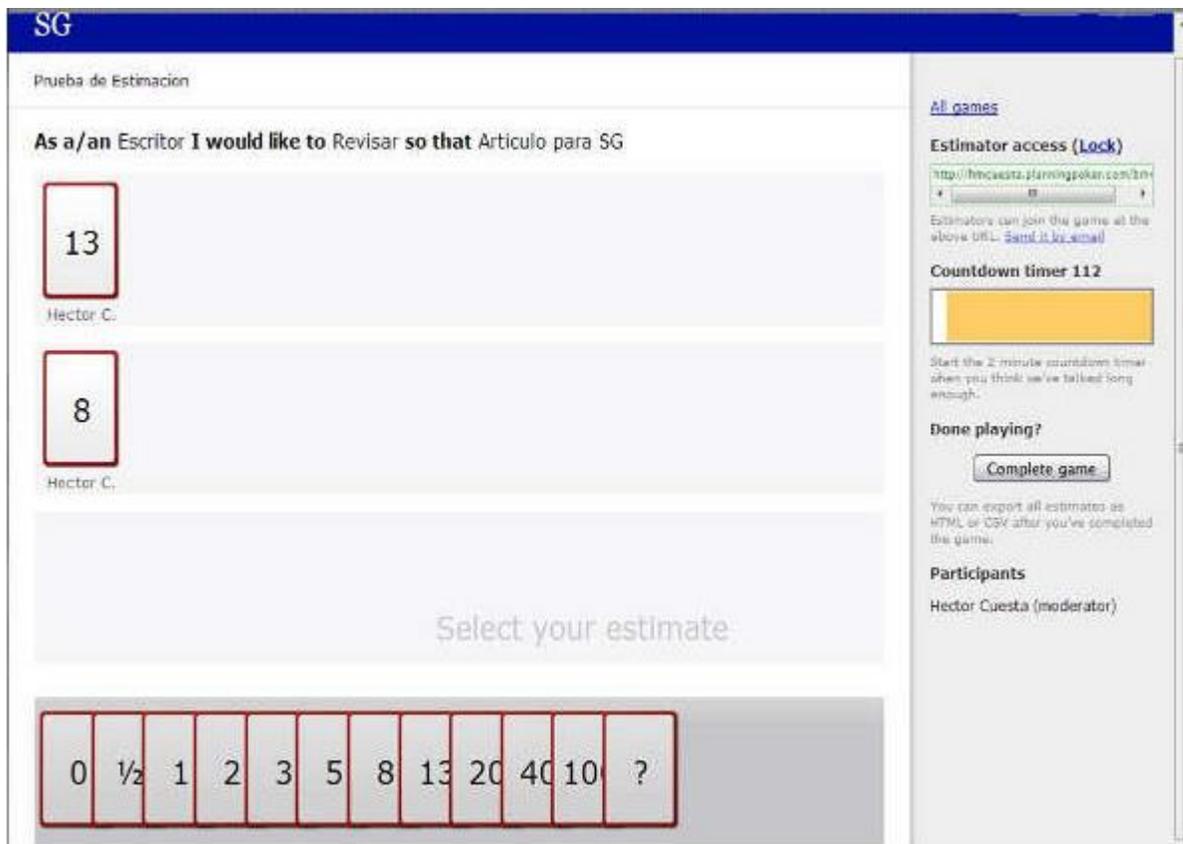
- HTTP Middleware: Middleware se encarga de analizar y filtrar las llamadas HTTP en tu servidor. Puedes instalarlo para que se encargue de verificar que se trate de un usuario registrado, de evitar problemas de tipo Cross-SiteScripting (XSS) y otras medidas de seguridad.
- Autenticación: CodeIgniter viene listo para implementar autenticación de usuarios de forma nativa e incluye la opción de “recordar” al usuario. Además, permite incluir parámetros adicionales, lo que nos asegurará, por ejemplo, si se trata de un usuario activo.
- Encriptación: CodeIgniter provee de una clase\librería llamada Encrypt Class que permite encriptar y desencriptar cadenas de texto en base a una llave (key) de codificación; con cadenas de textos se quiere decir nuestras contraseñas, códigos de tarjetas de crédito, etc.

#### **4.4. COSTO**

Para planear el desarrollo de un software es necesario estimar de una u otra forma su tamaño y por supuesto su costo. Existen diversos mecanismos para estimar el tamaño de un software y el esfuerzo requerido para construirlo, en las páginas de SG anteriormente se han publicado artículos sobre métodos como la estimación por puntos de función, así como el recurrir a información histórica de proyectos anteriores.

##### **4.4.1. Técnica**

Consiste en crear una reunión de trabajo con el grupo de expertos. Se requiere de un moderador que dirigirá la actividad. Primero se expone el proyecto que se va a desarrollar. Luego se determinan las actividades requeridas para ejecutar el proyecto, y entonces para cada una de las actividades los distintos miembros del equipo realizan una estimación empírica y se conjuntan y discuten los resultados con el fin de cerrar la brecha y converger en un valor estimado para cada tarea. La información obtenida con la estimación de cada tarea se utiliza para realizar el plan general del proyecto y conocer el esfuerzo total.



**Figura 39. estimación**

**Fuente [planningpoker.com](http://www.planningpoker.com)**

#### **4.4.2. Proceso**

A continuación, mostramos el proceso en mayor detalle que se realiza para cada historia de usuario a estimar.

- El moderador lee y explica una historia o requerimiento ante los miembros del equipo. Se podrán hacer preguntas para dejar claro qué se desea alcanzar.
- Los participantes anotan en una tarjeta su estimación para dicha tarea y la entregan al moderador de manera anónima, es decir que no se sepa qué tarjeta corresponde a qué persona.
- El moderador captura los resultados en un pizarrón o en alguna herramienta de software tal como la de <http://www.planningpoker.com>.

- Una vez que se conoce los integrantes deberán discutirlos, haciendo énfasis en el razonamiento detrás de los valores extremos (el más bajo y el más alto).
- Tomando en cuenta el conocimiento obtenido por esta discusión, cada participante hace una nueva estimación para la historia y la anota en una tarjeta.
- El moderador captura las estimaciones ajustadas y en caso de no haber consenso se vuelven a discutir.

#### 4.4.3. Verificación y Validación

Es recomendable que al terminar el proyecto y realizar el post-mortem de éste se compare la estimación contra el esfuerzo real, de manera que esto se tome en cuenta para futuras estimaciones.

Para esto podemos recurrir al cálculo del balance relativo de error (BRE) con la siguiente ecuación:

$$BRE = \frac{X - Y}{\text{Min}(X, Y)}$$

Donde X = real y Y = estimado

#### 4.4.4. Consideraciones

Esta técnica requiere de un moderador con gran pericia para lograr que rápidamente haya convergencia en las estimaciones. También hay que tener cuidado cuando uno de los integrantes tiene demasiada autoridad, ya que puede sesgar la estimación. No es posible estimar un proyecto entero de esta forma, sino que se recomienda estimar por historia o sprint si se trabaja con SCRUM.

Uno de los beneficios de esta técnica es que permite considerar distintos puntos de vista. Adicionalmente, ayuda a hacer amenizar una tarea que puede ser tediosa. Resulta ser una técnica mejor que Delphi y que grupos no estructurados, tal vez por la discusión cara a cara entre los participantes. A medida que se practica Planner Poker se podrán hacer mejores estimaciones.



# **CAPÍTULO IV**

# **CONCLUSIONES Y**

# **RECOMENDACIONES**

## **5. CONCLUSIONES Y RECOMENDACIONES**

### **5.1. CONCLUSIONES**

Luego de plantear el problema, diseñar, desarrollar y probar el sistema web para TECHSBOL, aplicando todas las metodologías de análisis y diseño de software, se logró cumplir con los objetivos planteados en un inicio, y el desarrollo del sistema fue exitoso. Tomando en cuenta los objetivos inicialmente planteados se llegó a las siguientes conclusiones:

- Se logró implementar un medio de sus cursos para incrementar las ganancias de TECHSBOL.
- Se logró brindar sitio/espacio digital de visualización de cursos que muestren los precios y descripción.
- Se logró expandir el mercado de los cursos de TECHSBOL de manera digital y on-line de manera que los cursos siempre estén disponibles.
- Se logró proporcionar información de cursos novedosos.
- Se logró centralizar la información de los estudiantes.
- Se mejoró la administración de cursos.
- Se mejoró la gestión de educación que se desarrolla en TECHSBOL.

De esta manera se alcanzó el objetivo general y los objetivos específicos, planteado para mejorar el acceso a la de educación mediante cursos en línea, de manera que las personas sean capaces de tomar sus cursos de manera normal sin contratiempos, además que tengan la posibilidad de tomar cualquier otro curso sin necesidad de apersonarse en la institución haciendo de esto una facilidad a la hora de capacitarse en el área deseada.

### **5.2. RECOMENDACIONES**

A la culminación del presente proyecto se efectúan las siguientes recomendaciones:

- El sistema puede ser complementado con la opción de poder generar reportes en Excel.

- Desarrollar una aplicación móvil para mayor comodidad de los clientes, que navegan desde dispositivos móviles.
- Aumentar al módulo de autenticación un CAPTCHA que aparezca cuando el usuario falle muchas veces en su autenticación, para evitar robo de cuentas.
- Realizar copias de seguridad de la base de datos si es posible diariamente o en su defecto semanalmente, ya que la información que contiene es muy importante para el buen funcionamiento del sistema.

## Bibliografía

ARMENGOL, M. C. (1982).

available!, P. 5. (21 de junio de 2020.). *Perl 5.32.0 is now available!* Obtenido de Perl 5.32.0 is now available!: [www.nntp.perl.org](http://www.nntp.perl.org)

CALDEIRO, G. (2014). *La incidencia de la configuración digital en los estilos comunicacionales sobre los que se construyen las Dinámicas Colaborativas Mediadas por Tecnología. Virtualidad, Educación y Ciencia.* <http://www.revistas.unc.edu.ar/index.php/vesc/article/view/7470>.

Chen, C. (2019). *TIC (Tecnologías de la información y la comunicación)*. Chile.

Dangel, A. D. (2010). *Sistemas de Información*. Peralta.

DÍAZ, P. y. (2005). *INGENIERIA DE LA WEB Y PATRONES DE DISEÑO*. España: Prentice Hall.

Espinoza Manchego, M. S. (2014).

Flathub. (s.f.). *Flathub*. Obtenido de <https://github.com/flathub/com.visualstudio.code.oss>

Flatpak. (s.f.). *Flatpak*. Obtenido de <https://es.wikipedia.org/wiki/Flatpak>

Foundation, T. A. ( 2007). *Apache License and Distribution FAQ*.

GARCÍA ARETIO, L. (1999). *Historia de la Educación a Distancia*. RIED. *Revista Iberoamericana de Educación a Distancia*.

here, T. C. (2019-10-23). *The Codelgniter Foundation is finally here*.

Husaković, A. (2010 de 06 de 29). <https://mariadb.org/about/#team>. Obtenido de <https://mariadb.org/about/#team>: <https://mariadb.org>

Idalberto, C. (2006). *Introducción a la Teoría General de la Administración*. McGraw-Hill Interamericana.

- Jardines, F. (2009). *Desarrollo histórico de la educación a distancia (Historical development of distance education)*. UANL.
- Javier López Moratalla, I. M. (2003-2004).
- Julián Pérez Porto, A. G. (2009). *Definición de ingeniería de software*.
- Julián Pérez Porto, A. G. (2012). *Definición de computadora*.
- LLAMAS, J. L. (1986).
- Manuel, P. (2008).
- McAnally-Salas, L. (2007). *La educación en línea, su complejidad y las instituciones de educación*. Virtual Educa Brasil.
- Moll, S. (2014).
- MORA, S. L. (2003). *Programación de aplicaciones web: historia, principios básicos y clientes web*. Alicante: Club Universitario.
- O'Reilly, T. (2005).
- Oscar, L. R., Acosta, J. C., Mata, L. E., Bachmann, N. G., & Vallejos. (1 de enero de 2012). Aprendizaje combinado, aprendizaje electrónico centrado en el alumno y nuevas tecnologías.
- Peña. (2006). *Sistemas de Información Geográfica aplicados a la gestión del territorio*. Chile: Editorial Club Universitario.
- Prentice-hall. (1981).
- Preprocesso, P. H. (04 de 06 de 2020). *PHP: Hypertext Preprocesso*. Obtenido de [www.php.net](http://www.php.net)
- Pressman, R. S. (2005). *Ingeniería del Software. Vol.*
- Raffino, M. E. (2020). *Educación*. Argentina.
- Raffino, M. E. (2020). *Informática*. Argentina.

Schrum, L. (1998). *On-Line Education: A Study of Emerging Pedagogy*.

Seidler, K. '. (2006). *xampp*. Obtenido de <https://www.apachefriends.org/es/about.html>

SEVILLA, U. D. (2007). *e-Learning. Definición y Características*. Sevilla (España).

Technology, B. I. (enero de 2011). *BCIT COMPUTING*. Obtenido de BCIT  
COMPUTING: <https://codeigniter.com/>

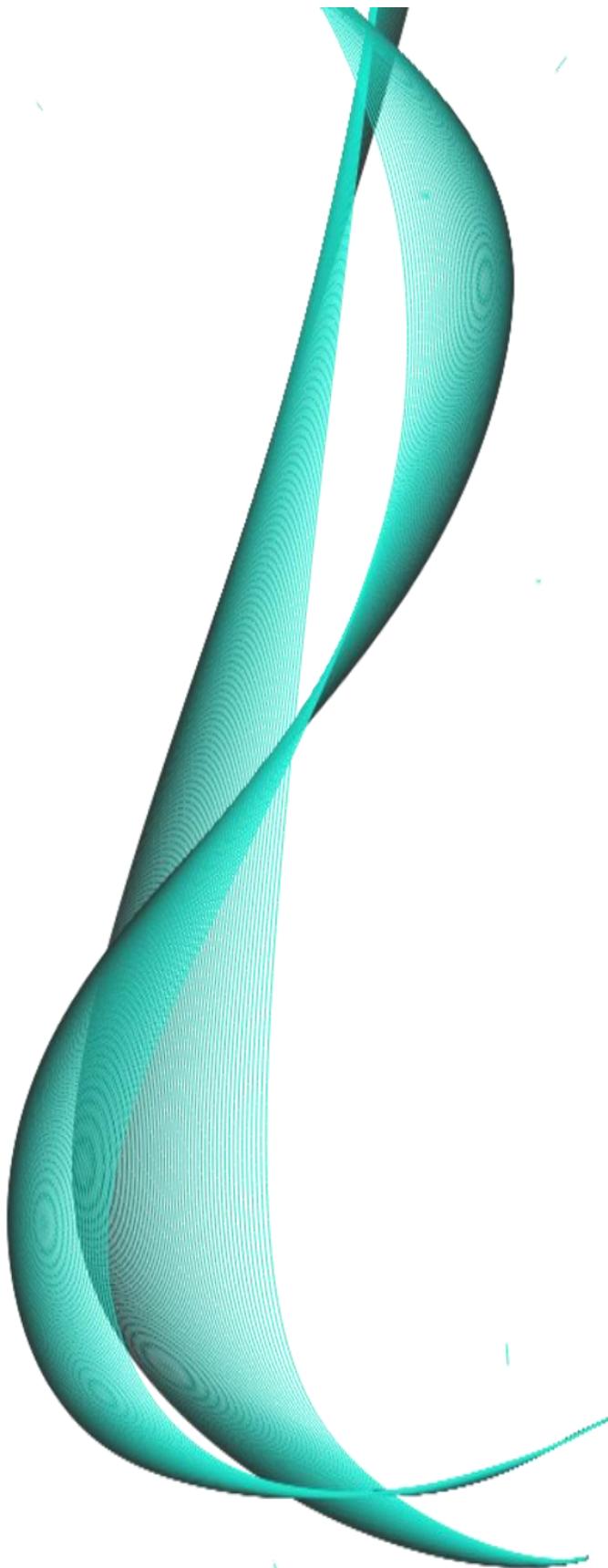
vazquez, M. (2014). *e-Lis e-prints in library & information science*.

Velasco, R. A. (2009).

Vogelgesang, K. (2002). *xampp*. Obtenido de *xampp*:  
<https://www.apachefriends.org/es/about.html>

wikipedia. (2020). *wikipedia*. Obtenido de <https://es.wikipedia.org/wiki/DBeaver>

ZARATE, J. M. (2014 ). *TUTOR VIRTUAL B-LEARNING EN TIC CON NORMAS DE CALIDAD*.



# **ANEXOS**

**ANEXO A.** MANUAL USUARIO

# **ACADEMIA | TECHSBOL**

**Guía de usuario de administrador**

Copyright 2018 Rudy David Castillo. Todos los derechos reservados

## Índice

### Contenido panel administrativo

- Tablero
- Categorías
  - Categorías
  - Agregar categorías
- Cursos
- Instructores
  - Lista de instructores
  - Configuración del instructor
  - Solicitud del instructor
- Estudiantes
- Matricula
  - Historial de inscripciones
  - Inscribir a estudiante
- Informe
  - Ingresos de administrador
  - Ingresos del instructor
- Pago sin conexión
  - Solicitud pendiente
  - Solicitud aceptada
  - Solicitud suspendida
- Mensaje
- Configuración
  - Configuración del sistema
  - Configuración del sitio web
  - Configuración de pago
- Administrar perfil

## 1. Tablero

- a. Resumen del sistema que se muestra en la página de inicio. Número total de cursos, número total de lecciones, número total de En el panel de control se muestran las inscripciones, el número total de estudiantes.

## 2. Categorías

- a. Categoría
- b. Agregar nueva categoría

**¿Cómo crear una categoría?** - Desde el menú de navegación del panel de administración, vaya a Categorías. El administrador podrá ver una lista de categorías que el administrador ha creado. En la parte superior de la lista hay un botón llamado "+ Agregar nueva categoría". Al hacer clic en ese botón, el administrador verá un formulario para crear categorías.

**¿Cómo crear una subcategoría?** - Para crear subcategorías, el administrador puede crear una subcategoría para la página Categoría. Tiene que seleccionar el botón llamado "+ Agregar nueva categoría". Al hacer clic en ese botón, el administrador verá un formulario de creación de categorías dentro del mismo encontrará un botón desplegable con el nombre de categoría padre en el cual deberá seleccionar a la categoría, haciendo que el mismo sea una sub categoría de la categoría padre.

## 3. Cursos

- a. Cursos

**¿Cómo crear cursos?** - Para crear un curso, el administrador debe crear primero una categoría y una subcategoría. El administrador podrá ver todos los cursos en la página del curso. Vaya a la opción Cursos del menú de navegación izquierdo del administrador, se mostrará una lista de todos los Cursos creados. Al seleccionar la categoría y subcategoría deseadas, el administrador puede filtrar la lista de cursos. Aparecerá un formulario después de hacer clic en el botón "+ Agregar curso" en la parte superior. El

administrador puede crear nuevos cursos proporcionando todos los datos necesarios.

Al crear un curso, el administrador puede proporcionar información valiosa como meta palabra clave, meta descripción para ese curso específico para que sea más fácil para los motores de búsqueda encontrarlo y organizarlo. Lo que llamamos Optimización de motores de búsqueda o SEO.

**¿Cómo gestionar las secciones?** - Cada curso debe tener al menos una sección. El administrador puede seleccionar un curso para administrar secciones del mismo. Solo ve al

Página del curso desde el menú de navegación del administrador, seleccione un curso, haga clic en el menú desplegable Acción, elija Administrar secciones. Aparecerá una lista de las secciones creadas para ese curso específico. El administrador puede crear nuevas secciones simplemente haciendo clic en el botón "+ Agregar sección". El administrador puede editar, eliminar y serializar esas secciones eligiendo las opciones del menú desplegable Acción.

**¿Cómo gestionar las lecciones?** - El administrador puede seleccionar un curso para administrar las lecciones del mismo. Simplemente vaya a Cursos desde el menú de navegación del administrador, seleccione un curso, haga clic en el menú desplegable Acción y elija Administrar lección. Aparecerá una lista de lecciones creadas para ese curso específico. El administrador puede crear nuevas lecciones simplemente haciendo clic en el botón "+ Agregar lección". El administrador puede filtrar lecciones seleccionando el curso en el menú desplegable del curso. El administrador también puede Editar, Eliminar esas lecciones eligiendo las opciones del menú desplegable Acción.

El administrador puede crear diferentes tipos de lecciones, como URL de video, archivos de texto, imágenes de archivos PDF, etc. Primero debe elegir el tipo de lección al crear una lección.

#### 4. Estudiante

El administrador podrá ver toda la lista de estudiantes que se inscribieron. Simplemente vaya a Estudiante desde el menú de navegación del administrador. El administrador también podrá crear estudiantes por sí mismo. También puede editar y eliminar alumnos de la página de alumnos.

#### 5. Matricula

El administrador podrá ver la lista de todos los estudiantes inscritos que aparecerán en la página Historial de inscripciones. El administrador puede encontrar esto desde el menú de navegación del administrador hasta el Historial de inscripciones.

El administrador puede inscribir a un estudiante manualmente. Desde la opción Inscribir a un estudiante.

#### 6. Mensaje

El administrador puede encontrar una opción de Mensaje en el menú de navegación del administrador. Donde pueda iniciar o continuar una conversación individual entre él y sus alumnos.

#### 7. Configuración

- a. configuración del sistema
- b. configuración del sitio web
- c. configuración de pago

**¿Qué es la configuración del sistema?** - La configuración es el componente clave de cualquier sistema de gestión. La flexibilidad de un sistema se define por su configuración fácil de manejar. La "Configuración del sistema" permitirá al administrador cambiar la configuración básica del sistema editando el campo de datos requerido. También brinda la posibilidad de cambiar el logotipo, la información de contacto y otros aspectos importantes de la configuración.

En la configuración del sistema, el administrador ahora puede proporcionar información valiosa como meta palabra clave, meta descripción y nombre del autor y facilita a los motores de búsqueda encontrarlo y organizarlo. Lo que llamamos Optimización de motores de búsqueda o SEO

**¿Qué es la configuración del sitio web?** - La "Configuración del sitio web" permitirá al administrador cambiar la configuración básica del sitio web editando el campo de datos requerido. También ofrece la posibilidad de cambiar el logotipo de la interfaz, el banner de inicio, Acerca de, Política de privacidad y Términos y condicionar otros aspectos importantes de la configuración.

**¿Qué es la configuración de pago?** - La "Configuración de pago" permitirá al administrador cambiar la configuración de pago, como identificadores de cliente, claves secretas, claves públicas de diferentes tipos de pasarelas de pago. También da la posibilidad de cambiar el modo de prueba. Desde la configuración de pago, el administrador puede establecer la moneda. El administrador puede configurar la moneda predeterminada del sistema, otro para la moneda de PayPal. Para mantener la coherencia, el administrador debe mantener todas las divisas iguales.

## **8. Administración de perfil**

La información del perfil se puede editar haciendo clic en "Imagen de perfil de administrador" en el encabezado. La información se puede editar y guardar usando este "Perfil de actualización".