

# UNIVERSIDAD PÚBLICA DE EL ALTO

## CARRERA INGENIERÍA DE SISTEMAS



## PROYECTO DE GRADO

**“CUADERNO PEDAGÓGICO WEB COMO HERRAMIENTA DEL MAESTRO  
PARA EL SEGUIMIENTO DE ACTIVIDADES PEDAGÓGICAS”**

**CASO: UNIDAD EDUCATIVA TÉCNICO HUMANÍSTICO JAPÓN B**

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

**MENCIÓN: INFORMÁTICA Y COMUNICACIONES**

**POSTULANTE:** Alfredo Callisaya Huanca

**TUTOR METODOLÓGICO:** Ing. Dionicio Henry Pacheco Ríos

**TUTOR ESPECIALISTA:** Lic. Santos Zenón Quispe Apaza

**TUTOR REVISOR:** Ing. Fanny Helen Pérez Mamani

**EL ALTO – BOLIVIA**

**2021**

## **Dedicatoria**

El presente proyecto de grado va dedicado a Dios todo poderoso, por haberme dado salud, esperanza y por iluminar mi mente en cada paso que doy.

A mi querida esposa Genara, por el apoyo incondicional y comprensión que me dio a lo largo de estos años de estudio.

A mis hijos Edilson y Emerson les quité tiempo de estar juntos perdón, me esforzaré por redimirlo.

A mis queridos padres, Daniel y María, quienes con su apoyo lograron alentar mi camino y continuar a pesar de las adversidades, que Dios le bendiga a mi familia.

¡Gracias!

## **Agradecimientos**

Quiero agradecer en a Dios, por guiarme en el camino y fortalecerme espiritualmente para empezar un camino lleno de éxito.

A mi tutor metodológico Ing. Dionicio Henry Pacheco Ríos por su valioso tiempo y dedicación en todo el proceso de desarrollo del proyecto de grado.

A mis tutores Lic. Santos Zenón Quispe Apaza y Ing. Fanny Helen Pérez Mamani por toda la sabiduría y experiencia profesional, por sus consejos, colaboración y paciencia a lo largo del desarrollo de este proyecto de grado, que es un gran paso en la vida de un estudiante Universitario.

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

¡Muchas gracias por todo!

## RESUMEN

El aumento de Unidades Educativas del sub sistema de educación regular y el crecimiento de la comunidad estudiantil, genera un gran volumen de información, que debe ser organizada y clasificada para su manejo de datos. Para realizar estas tareas muchos de las unidades educativas hacen uso de dispositivos móviles, computadoras internet, herramientas digitales para coadyuvar con el trabajo educativo.

El proyecto de grado titulado “CUADERNO PEDAGÓGICO WEB COMO HERRAMIENTA DEL MAESTRO PARA EL SEGUIMIENTO DE ACTIVIDADES PEDAGÓGICAS” se ha desarrollado con el objetivo de coadyuvar procesos educativos de los maestros de la unidad educativa Técnico Humanístico “Japón B”

Para iniciar con el proyecto se utilizó la metodología Ágil Scrum para el desarrollo y proceso del sistema; planificando las tareas de acuerdo a las historias de usuario que son contadas por los actores educativos, para el desarrollo del sistema web se utilizó el lenguaje de programación php complementado con el framework laravel, como gestor de base de datos MySql y composer para la gestión de paquetes.

Finalmente, los objetivos planteados en base a la problemática existente dentro de la unidad educativa, han sido alcanzados de manera satisfactoria, de forma que se puso en producción el sistema web que permite realizar inscripción de estudiantes, asignar maestros a las áreas correspondientes, controlar asistencias y calificaciones.

# ÍNDICE

<b>CAPITULO I</b> .....	1
1. MARCO PRELIMINAR .....	1
1.1. INTRODUCCIÓN .....	1
1.2. ANTECEDENTES .....	2
1.2.1. Antecedente Institucional .....	2
1.2.2. Objetivos de la institución.....	2
1.2.2.1. Objetivo general: .....	2
1.2.2.2. Objetivos Específicos .....	3
1.2.3. Misión Visión de la institución.....	3
1.2.3.1. Misión.....	3
1.2.3.2. <i>Visión</i> .....	3
1.2.4. <i>Antecedente Internacionales</i> .....	4
1.2.5. <i>Antecedente Nacionales</i> .....	5
1.3. Planteamiento del Problema .....	6
1.3.1. Problema Principal .....	6
1.3.2. Problemas Secundarios .....	7
1.4. OBJETIVO GENERAL .....	7
1.4.1. Objetivo General .....	7
1.4.2. Objetivo Especifico.....	7
1.5. JUSTIFICACIÓN .....	7
1.5.1. Justificación Técnica .....	7
1.5.2. Justificación Económica .....	9
1.5.3. Justificación Social .....	9
1.6. METODOLOGÍA .....	9
1.6.1. Scrum.....	9
1.6.2. Método de Ingeniería .....	11
1.6.2.1. Métricas de la Calidad de Software .....	11
1.6.2.2. <i>Características de la calidad del software</i> .....	11
1.7. COSTO Y BENEFICIO .....	13
1.7.1. <i>Cocoma II</i> .....	13
1.8. HERRAMIENTAS.....	15

1.8.1.	Plataforma del sistema .....	15
1.8.2.	Servidor.....	15
1.8.3.	Framework .....	16
1.8.4.	Herramientas de diseño .....	16
1.9.	LÍMITES Y ALCANCES.....	17
1.9.1.	Limites.....	17
1.9.2.	Alcances .....	17
1.10.	APORTES .....	17
<b>CAPITULO II</b>	.....	<b>19</b>
2.	MARCO TEÓRICO.....	19
2.1.	INTRODUCCIÓN .....	19
2.2.	CUADERNO PEDAGÓGICO .....	19
2.3.	METODOLOGÍA SRUM .....	20
2.3.1.1.	Componentes de Scrum.....	21
2.3.1.1.1.	Las reuniones.....	21
2.3.1.1.2.	Roles.....	22
2.3.1.2.	Elementos de Scrum .....	23
2.3.1.3.	Fases de Scrum .....	24
2.3.1.3.1.	Pre-game (Antes del Juego).....	25
2.3.1.3.2.	Development phase (Fase de desarrollo) .....	26
2.3.1.3.3.	Post-game (Después del Juego) .....	28
2.4.	UML - LENGUAJE UNIFICADO DE MODELADO .....	29
2.4.1.	Concepción del UML .....	30
2.4.2.	Diagramas UML .....	30
2.4.2.1.	Diagrama de clases.....	30
2.4.2.2.	Diagrama de Objetos .....	31
2.4.2.3.	Diagrama Casos de uso .....	31
2.4.2.4.	Diagrama de estados .....	32
2.4.2.5.	Diagrama de secuencias .....	32
2.4.2.6.	Diagrama de actividades .....	36
2.4.2.7.	Diagrama de colaboraciones.....	38
2.4.2.8.	Diagrama de componentes .....	39

2.4.2.9.	Arquitectura .....	39
2.4.3.	Ciclo de vida.....	39
2.5.	Herramientas de Implementación .....	41
2.5.1.	Algoritmos .....	41
2.5.2.	Lenguaje de Programación PHP .....	41
2.5.3.	HTML5 .....	42
2.5.4.	Xampp.....	43
2.5.5.	PHPMYADMIN .....	44
2.5.6.	Visual Studio Code .....	44
2.5.7.	Hoja de estilos CSS .....	45
2.5.8.	AdminLTE .....	45
2.5.9.	Bases de datos en la vida cotidiana .....	46
2.5.10.	Base de datos .....	48
2.5.10.1.	MySQL .....	48
2.5.10.2.	MariaDB .....	49
2.5.10.3.	Framework .....	49
2.5.10.4.	Laravel .....	50
2.5.10.5.	Composer .....	50
2.5.11.	Patron MVC.....	51
2.6.	CALIDAD DE SOFTWARE.....	52
2.7.	ISO/IEC 25010 .....	52
2.7.1.	Adecuación Funcional .....	53
2.7.2.	Eficiencia de desempeño .....	53
2.7.3.	Compatibilidad .....	54
2.7.4.	Usabilidad .....	54
2.7.5.	Fiabilidad.....	55
2.7.6.	Seguridad.....	55
2.7.7.	Mantenibilidad .....	56
2.7.8.	Portabilidad .....	56
2.8.	COCOMO II.....	57
2.8.1.	Estimación de esfuerzo .....	57
2.8.1.1.	<i>Modelo Composición de Aplicación.....</i>	<i>57</i>

2.8.1.2.	<i>Modelo de diseño temprano</i> .....	58
2.8.1.3.	<i>Modelo Post-Arquitectura</i> .....	60
2.8.2.	Estimación del Cronograma .....	60
2.8.3.	Métricas de Software.....	61
2.9.	SEGURIDAD.....	61
2.10.	Cifrador AES .....	62
2.10.1.1.	Matriz de Estado .....	63
<b>CAPITULO III</b>	.....	<b>65</b>
3.	MARCO APLICATIVO.....	65
3.1.	Recopilación de Información .....	65
3.1.1.	Concepción .....	65
3.1.2.	Indagación y elaboración.....	66
3.1.3.	Negociación .....	67
3.2.	Análisis de la Situación Actual.....	67
3.3.	PRE GAME (antes del juego) .....	67
3.3.1.	Planificación .....	67
3.3.2.	Análisis de Requerimiento (Product Backlog).....	68
3.3.3.	Identificación de Usuarios .....	69
3.3.4.	Identificación de Roles de los Actores .....	69
3.3.5.	Requerimientos de Usuario .....	70
3.3.6.	Análisis de Riesgo.....	74
3.4.	GAME (Juego) .....	75
3.4.1.	Iteraciones.....	75
3.4.1.1.	Primera Iteración .....	75
3.4.1.2.	Segunda Iteración .....	76
3.4.1.3.	Tercera Iteración .....	77
3.4.1.4.	Cuarta Iteración.....	78
3.4.2.	Modelado del Sistema.....	79
3.4.2.1.	Diagrama de caso de uso.....	79
3.4.2.2.	Diagrama de caso de uso General .....	80
3.4.2.3.	Diagrama de caso de uso: Registro de usuario .....	81
3.4.2.4.	Diagrama de caso de uso: Registro de Áreas y Campos .....	82



3.4.2.5.	Diagrama de caso de uso: Inscripción .....	83
3.4.2.6.	Diagrama de caso de uso: Asignar materia al Maestro .....	84
3.4.2.7.	Diagrama de caso de uso: Gestión de Asistencias .....	85
3.4.2.8.	Diagrama de caso de uso: Gestión de Notas .....	86
3.4.2.9.	Diagrama de caso de uso: Seguimiento del estudiante .....	87
3.4.2.10.	Descripción de casos de uso .....	88
3.4.2.11.	Diagrama de Actividades .....	94
3.4.2.12.	Diagrama de Clases .....	101
3.4.2.13.	Diagrama de Presentación .....	102
3.5.	POST-GAME .....	108
3.5.1.	Prueba del Sistema .....	108
3.5.1.1.	Pruebas de caja blanca .....	108
3.5.1.2.	Pruebas de caja negra .....	109
3.6.	MODELO DE IMPLANTACIÓN DEL PROGRAMA .....	110
<b>CAPITULO IV</b>	.....	<b>117</b>
4.	ANÁLISIS DE CALIDAD Y COSTOS .....	117
4.1.	MÉTRICAS DE CALIDAD .....	117
4.1.1.	Técnica ISO 9126 .....	117
4.1.1.1.	Funcionalidad .....	117
4.1.1.2.	Confiabilidad .....	121
4.1.1.3.	Mantenibilidad .....	122
4.1.1.4.	Portabilidad .....	123
4.1.1.5.	Usabilidad .....	123
4.1.2.	Resultados .....	124
4.2.	SEGURIDAD DE SOFTWARE .....	125
4.2.1.	Autenticación .....	125
4.2.2.	Cifrado .....	126
4.2.3.	Hash .....	127
4.2.4.	Restablecimiento de contraseñas .....	127
4.3.	COSTO DEL SISTEMA .....	128
4.3.1.	COCOMO II .....	128
4.3.2.	Tiempo del desarrollo del Software .....	129

4.3.3.	Costo del desarrollo del Software .....	131
4.3.4.	Costo de Elaboración del Proyecto .....	132
4.3.5.	Costo total de Proyecto .....	132
<b>CAPITULO V</b> .....		<b>134</b>
5.	CONCLUSIONES Y RECOMENDACIONES.....	134
5.1.	CONCLUSIONES.....	134
5.2.	RECOMENDACIONES .....	134
BIBLIOGRAFÍA.....		135
ANEXOS .....		137
MANUAL DE USUARIO.....		138
AVALES DE CONFORMIDAD .....		151

### Índice de tablas

Tabla 1.1.	Características del Equipo de la Institución .....	8
Tabla 1.2.	Características de Sistema Operativo .....	8
Tabla 1.3.	Aplicaciones desarrolladas por usuarios finales .....	14
Tabla 1.4.	Aplicaciones desarrolladas por usuarios finales .....	15
Tabla 1.5.	Los requerimientos de Hardware para el desarrollo .....	16
Tabla 2.1.	Productividad para el modelo Composición de Aplicación.....	58
Tabla 2.2.-	Número de rondas .....	62
Tabla 3.1.	Requerimientos funcionales del Software.....	66
Tabla 3.2.	Roles de Actores .....	69
Tabla 3.3.	Historia de usuario – 1 .....	70
Tabla 3.4.	Historia de usuario - 2 .....	70
Tabla 3.5.	Historia de usuario - 3 .....	71
Tabla 3.6.	Historia de usuario - 4 .....	71
Tabla 3.7.	Historia de usuario - 5 .....	72
Tabla 3.8.	Historia de usuario - 6 .....	72
Tabla 3.9.	Requerimiento de usuario .....	73
Tabla 3.10.	Requerimiento de usuario .....	74
Tabla 3.11.	Primera Iteración .....	75
Tabla 3.12.	Segunda Iteración .....	76

Tabla 3.13. Tercera Iteración .....	77
Tabla 3.14. Cuarta Iteración.....	78
Tabla 3.15. Descripción de actores .....	79
Tabla 3.16. Registro de usuario .....	88
Tabla 3.17. Registro de Áreas y Campo de Saberes y Conocimientos .....	89
Tabla 3.18. Inscripción de Estudiantes.....	89
Tabla 3.19. Asignar materias al maestro.....	90
Tabla 3.20. Gestión de asistencias .....	91
Tabla 3.21. Gestión de Notas .....	92
Tabla 3.22. Seguimiento del Estudiante.....	93
Tabla 3.23. Pruebas unitarias .....	108
Tabla 4.1. Entrada de datos para el Cálculo de Funcionalidad .....	118
Tabla 4.2. Cuenta total con fator de ponderación medio .....	118
Tabla 4.3. Cuenta total con fator de ponderación medio .....	119
Tabla 4.4. Información Requerida para el IMS .....	122
Tabla 4.5. Encuestas de usabilidad del sistema.....	124
Tabla 4.6. Resultados de usabilidad del sistema .....	124
Tabla 4.7. Coeficiente a y c y los exponentes b y d .....	129
Tabla 4.8. Factor LCD/PF de Lenguajes de Programación .....	130
Tabla 4.9. Costo de Elaboración del Proyecto .....	132
Tabla 4.10. Costo total del proyecto.....	132

### **Índice de Figuras**

Figura 1.1. Estructura de la Unidad Educativa .....	4
Figura 1.2. El ciclo de desarrollo de Scrum .....	10
Figura 1.3. Ciclo principal de Scrum.....	10
Figura 1.4. Dimensiones de calidad .....	12
Figura 1.5. Calidad Aplicación Web .....	13
Figura: 1.6 Diferencia entre Backend y Frontend .....	16
Figura 2.1. Flujo del Proceso SCRUM .....	20
Figura 2.2. Ciclo de desarrollo Scrum .....	23
Figura 2.3. El proceso o las tres fases de Scrum .....	24

Figura 2.4. Las actividades que se desarrollan durante del Sprint .....	26
Figura 2.5. Esquema de las fases de desarrollo en Scrum .....	28
Figura 2.6. El símbolo UML de una clase .....	30
Figura 2.7. El símbolo UML del Objeto.....	31
Figura 2.8. El símbolo UML del Objeto.....	31
Figura 2.9. Diagrama de estados UML.....	32
Figura 2.10. Estructura de la Unidad Educativa .....	33
Figura 2.11. Activación.....	33
Figura 2.12. Mensaje .....	34
Figura 2.13. Líneas de Vida .....	34
Figura 2.14. Destrucción de Objetos .....	35
Figura 2.15 Loops .....	35
Figura 2.16. Estados de Acción .....	36
Figura 2.17. Flujos de Acción.....	36
Figura 2.18. Flujo de Objetos .....	37
Figura 2.19 Estados Inicial y Final .....	37
Figura 2.20. Ramificación .....	37
Figura 2.21. Simplificación .....	38
Figura 2.22. Marco de responsabilidad .....	38
Figura 2.23 Diagrama de componente .....	39
Figura 2.24. Ciclo de vida del desarrollo software .....	40
Figura 2.25. Estructura de HTML 5 .....	42
Figura 2.26. Estructura de las etiquetas de HTML 5 .....	43
Figura 2.27. Etiqueta.....	43
Figura 2.28. Panel de control de Xampp .....	44
Figura 2.29. MariaDB y MySQL .....	49
Figura 2.30. Logo de laravel .....	50
Figura 2.31. logo de Composer .....	51
Figura 2.31. logo de Composer .....	63
Figura 2.31. logo de Composer .....	63
Figura 3.1: Planeación del Desarrollo del Proyecto.....	68
Figura 3.2. Caso de uso General .....	80

Figura 3.3. Diagrama de casos de uso – Registro de usuario .....	81
Figura 3.4. Diagrama de casos de uso - Registro de Áreas y Campos .....	82
Figura 3.5. Diagrama de casos de uso – Registro.....	83
Figura 3.6. Diagrama de casos de uso – Asignar materias .....	84
Figura 3.7. Diagrama de casos de uso – Asignar materias .....	85
Figura 3.8. Diagrama de casos de uso – Asignar materias .....	86
Figura 3.9. Diagrama de casos de uso – Seguimiento Estudiantes.....	87
Figura 3.10. Diagrama de actividades – Registro de usuario .....	94
Figura 3.11. Diagrama de actividades – Registro de Campo de Saberes .....	95
Figura 3.12. Diagrama de actividades – Inscripciones .....	96
Figura 3.13. Diagrama de actividades – Asignar Materia al Maestro.....	97
Figura 3.14. Diagrama de actividades – Registro de Asistencia.....	98
Figura 3.15. Diagrama de actividades – Registro de Calificaciones .....	99
Figura 3.16. Diagrama de actividades – Seguimiento del Estudiante .....	100
Figura 3.17. Diagrama de Clases.....	101
Figura 3.18. Diagrama de presentación – Ingreso al Sistema .....	102
Figura 3.19. Diagrama de presentación – Ingreso al Sistema .....	102
Figura 3.20. Diagrama de presentación –Estudiantes registrados .....	103
Figura 3.21. Diagrama de presentación – Formulario de registro.....	103
Figura 3.22. Diagrama de presentación – Formulario de registro de maestros	104
Figura 3.23. Diagrama de presentación – Registro de Área .....	104
Figura 3.24. Diagrama de presentación – Inscripción de estudiantes .....	105
Figura 3.25. Diagrama de presentación – Asignación de Maestros.....	105
Figura 3.26. Diagrama de presentación – Panel principal del Maestro.....	106
Figura 3.27. Diagrama de presentación – Registro de Asistencia .....	106
Figura 3.28. Diagrama de presentación – Registro de Calificaciones .....	107
Figura 3.29. Diagrama de presentación – Panel del Estudiante .....	107
Figura 3.30. Pruebas de autenticación.....	109
Figura 3.31. Ingreso al Sistema .....	110
Figura 3.32. Panel de Administración.....	110
Figura 3.33. Registro de Estudiantes .....	111
Figura 3.34. Registro de Estudiantes .....	111

Figura 3.35. Registro de Áreas .....	112
Figura 3.36. Inscripción de Estudiantes .....	112
Figura 3.37. Asignación de Maestros .....	113
Figura 3.38. Panel principal del Maestro .....	113
Figura 3.39. Gestión de Asistencia .....	114
Figura 3.40. Gestión de Notas .....	114
Figura 3.41. Gestión de Notas .....	115
Figura 3.42. Perfil del estudiante.....	115
Figura 3.43. El archivo de configuración de autenticación.....	125
Figura 4.2. Cifrado .....	126
Figura 4.2. Bcrypt.....	127
Figura 4.3. Argon .....	127
Figura 4.3. Argon .....	128

# **CAPÍTULO I**

---

## **MARCO PRELIMINAR**

# **CAPITULO I**

## **1. MARCO PRELIMINAR**

### **1.1. INTRODUCCIÓN**

Hoy en día la tecnología constituye en una de las más importantes contribuciones al ser humano que brinda soluciones educativas a partir de herramientas tecnológicas para el aprendizaje de los estudiantes.

Un aspecto muy importante en cualquier es la educación ya que de ello dependerá la calidad de vida de los bolivianos así también el desarrollo de la misma.

El colegio es la institución que favorece ese desarrollo de una manera más completa e integral, la cual le brinda herramientas necesarias para enfrentarse a la sociedad.

Todas las unidades educativas generan información a partir de las actividades pedagógicas que realizan cotidianamente llega a ser de gran importancia, a partir de esa información valiosa se toma decisiones muy importantes.

El constante crecimiento de las necesidades de aplicaciones web en las unidades educativas, surge la necesidad del desarrollo de sistemas de información Web, confiables y seguros que permitan a las unidades educativas optimizar en el manejo de la información.

Es así, que en el presente perfil de proyecto describe un cuaderno pedagógico web como herramienta del maestro / maestra y para el seguimiento de actividades pedagógicas de la unidad educativa técnico Humanístico Japón B. que facilita al maestro/a de los procesos pedagógicos que se realizan cotidianamente en la unidad educativa.



## **1.2. ANTECEDENTES**

### **1.2.1. Antecedente Institucional**

La Unidad Educativa Técnico Humanístico Japón “B” se encuentra ubicado calle Santos Atahuallpa, Av. Estructurante de la Zona Mercedes “A” Acribol, pertenecientes al Sub Distrito Municipal Nro. 8 de la ciudad de El Alto, altura carretera a Viacha, es una población que se dedica a la actividad comercial.

Al ser un Colegio Técnico Humanístico tiene como finalidad de implementar herramientas digitales e impartir educación de calidad con valores y actitudes, y habilidades que les permitan al estudiante estructurar un proyecto de vida hacia la Educación Superior.

Para una gestión eficiente la Unidad Educativa dependiente de la Dirección Distrital de Educación de El Alto-3, está organizada por comisiones:

- ✓ Técnico Pedagógico
- ✓ Convivencia Escolar
- ✓ Comisión Salud y Medio Ambiente
- ✓ Comisión Social Económica
- ✓ Comisión Deportiva
- ✓ Comisión BTH

### **1.2.2. Objetivos de la institución**

#### **1.2.2.1. Objetivo general:**

Desarrollar políticas educativas orientadas a la formación integral del estudiante, estimulando un desarrollo de competencias básicas esenciales y aplicadas que respondan con asertividad a exigencias de una sociedad competitiva donde el estudiante tenga una educación en valores y una nutrición saludable y se conviertan en herramientas eficaces para un aprendizaje significativo y de esa manera ser agentes de cambios en la sociedad.

### **1.2.2.2. Objetivos Específicos**

- Lograr una adecuada atención al estudiante mediante el mejoramiento del servicio técnico, pedagógico y administrativo.
- Lograr la implementación de un currículo actualizado (Según R.M. 001/2021), con metodologías y técnicas de Enseñanza-Aprendizaje, enmarcadas en la Ley 070.
- Mejorar y conservar la estructura física del colegio con el apoyo comprometido EL GAMEA, mediante la atención, priorizando la implementación, equipamiento y consolidación ESPACIOS Y AMBIENTES.
- Desarrollar el sentimiento de patriotismo, realzando los valores en las fechas cívicas correspondientes.

### **1.2.3. Misión Visión de la institución**

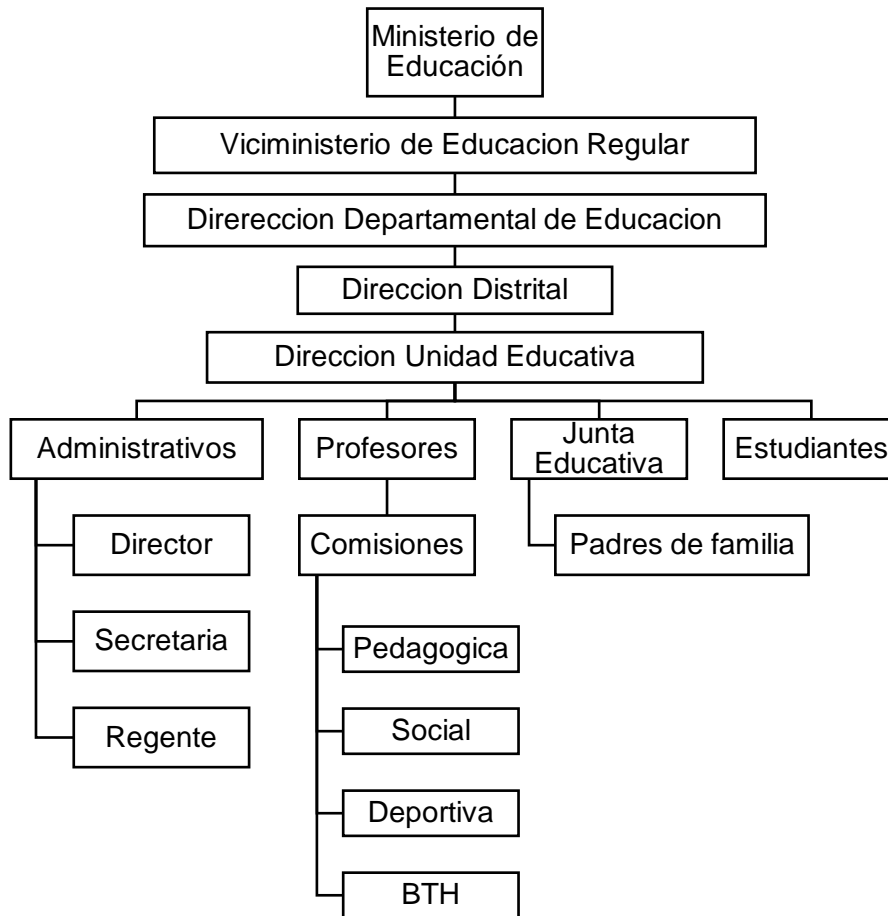
#### **1.2.3.1. Misión**

Garantizar una educación de calidad, tecnológica y productiva, fortaleciendo la implementación de infraestructura, mobiliario, equipamiento y la capacitación permanente de los recursos humanos, docentes, consejos comunitarios con el desarrollo integral y holístico de los estudiantes en base a un pensamiento crítico, acción transformadora, propositivo y con valores socio comunitarios, para VIVIR BIEN.

#### **1.2.3.2. Visión**

Ser una Unidad Educativa líder, con un servicio educativo de calidad, excelencia y eficiencia en base a las políticas educativas contextualizadas, productivas, innovadoras y tecnológicas, con docentes comprometidos para el cambio que garanticen la formación de estudiantes, críticos, reflexivos, investigadores, emprendedores, y propositivos, capaces de transformar y responder a las expectativas, necesidades y demandas de la sociedad con la aplicación del Modelo Educativo Socio comunitario Productivo.

**Figura 1.1.** Estructura de la Unidad Educativa



**Fuente:** Unidad Educativa “Japón B” (2020)

#### **1.2.4. Antecedente Internacionales**

E. M. (2015). Diseño de un sistema web para el seguimiento y evaluación de los alumnos con carta de permanencia en la facultad de ciencias contables, económicas y financieras (Tesis de grado). Universidad de San Martín de Porres, facultad de ingeniería y arquitectura - Escuela profesional de Ingeniería de Computación y Sistemas. Lima, Perú. Se trabajó con la Metodología Scrum, las herramientas que se utilizó fueron: Base de datos MySQL, Lenguaje de programación PHP.

J. C. (2011). Sistema de control escolar mediante una aplicación web utilizando software libre para las unidades educativas que conforman la casa salesiana

"Cristobal colón". (Tesis de grado) Universidad politécnica salesiana. Guayaquil, Ecuador. Se trabajó con la Metodología Web, las herramientas que se utilizó fueron: Base de datos Oracle, Lenguaje de programación Java.

E.D. (2016). Sistema de Información Web y su mejora en la gestión académica para el colegio privado hans kelsen del distrito de florencia de mora – Trujillo (Tesis de grado) Universidad Nacional del Trujillo, Facultad de Ingeniería, Carrera Ingeniería de Sistemas. Trujillo Perú. Se trabajó con la Metodología RUP y XP, las herramientas que se utilizó fueron: MySQL como base de datos, lenguaje de programación PHP y para aplicaciones Web el lenguaje de Modelado UML con las extensiones para aplicaciones Web.

### **1.2.5. Antecedente Nacionales**

W. D. (2009). Sistema de planificación y seguimiento académico. caso: Unidad Educativa RVDO. P. WALTER STRUB. (Proyecto de Grado). Universidad Mayor de San Andrés, Facultad de Ciencias Puras y Naturales, Carrera de Informática. La Paz, Bolivia. Se trabajó con la Metodología RUP, las herramientas que se utilizó fueron: Base de datos Postgres, Lenguaje de programación PHP.

I. O. (2018). Sistema integrado académico caso: Unidad Educativa San Sebastián B. (Proyecto de Grado). Universidad Mayor de San Andrés, Facultad de Ciencias Puras y Naturales, Carrera de Informática. La Paz, Bolivia. Se trabajó con la Metodología Scrum, las herramientas que se utilizó fueron: Base de datos Postgres, Lenguaje de programación, Javascript.

J. Y. (2018). Software de Control y Seguimiento de Actividades Pedagógicas caso: Unidad Educativa Simón Bolívar A. (Proyecto de Grado). Universidad Mayor de San Andrés, Facultad de Ciencias Puras y Naturales, Carrera de Informática. La Paz, Bolivia. Se trabajó con la Metodología Scrum, las herramientas que se utilizó fueron: Base de datos PostgreSQL, framework para aplicaciones web desarrollado en TypeScript.

### **1.3. Planteamiento del Problema**

#### **1.3.1. Problema Principal**

La Unidad Educativa Técnico Humanístico Japón B es una institución educativa público que desarrolla un conjunto de actividades académicos en la gestión educativa.

Durante los primeros días de clases los maestros realizan su propia lista a partir de los estudiantes asistentes. Posteriormente la comisión pedagógica en coordinación con la dirección de la Unidad Educativa, facilitan a los maestros un cuaderno pedagógico en una hoja de Excel para su posterior impresión.

El cuaderno pedagógico con el que cuentan los maestros en la unidad educativa es de forma impresa, y el registro de forma manual, por lo que los estudiantes no pueden ver el registro de su asistencia y las calificaciones de las actividades cotidianas.

Cuando un estudiante llega tarde o falta a las clases presenciales, el padre / madre de familia no tiene conocimiento, porque están desempeñando sus actividades cotidianas y no tienen tiempo para ir a la unidad educativa para consultar a su maestro sobre el aprovechamiento de su hijo.

La mayoría de los miembros de la comunidad educativa cuentan con un dispositivo móvil para la comunicación, pero no se aprovecha de todas las opciones que nos brinda el dispositivo móvil.

A partir de los problemas mencionados surge la siguiente interrogante

¿Cómo podrá el maestro realizar el seguimiento a la asistencia y calificaciones cotidianas de los estudiantes del nivel secundaria?

### **1.3.2. Problemas Secundarios**

- Información desorganizada, cada maestro con su propio formato de registro.
- Falta de un sistema de información para el manejo de datos.
- Ausencia de herramientas tecnológicas para la labor los Maestros.
- Deficiencia en el seguimiento de los estudiantes en el registro de asistencias.
- Dificultades en la centralización de notas al finalizar la gestión educativa

## **1.4. OBJETIVO GENERAL**

### **1.4.1. Objetivo General**

Desarrollar e implementar el sistema denominado, Cuaderno pedagógico Web como herramienta del maestro para el seguimiento de actividades pedagógicas de la Unidad Educativa Técnico Humanístico Japón B.

### **1.4.2. Objetivo Especifico**

- Analizar y priorizar los requerimientos de la institución Educativa.
- Gestionar el registro de estudiantes, maestros y áreas de saberes y conocimientos.
- Desarrollar un módulo de gestión que coadyuve en la labor del maestro.
- Crear un rol de estudiante para su seguimiento pedagógico.
- Incorporar reportes para centralizar calificaciones trimestrales.
- Elaborar un manual de usuario para el manejo del sistema Web.

## **1.5. JUSTIFICACIÓN**

### **1.5.1. Justificación Técnica**

Actualmente la tecnología está avanzando exponencialmente, llegando a ser parte indispensable en las instituciones públicas como privada. Las instituciones

Educativas no pueden dejar de lado el uso de las herramientas educativas en el proceso de enseñanza - aprendizaje.

La Unidad Educativa Técnico Humanístico “Japón B” cuenta con equipos de computación de última tecnología con conexión a internet, las características técnicas necesarias para la implementación del sistema, se detalla a continuación:

**Tabla 1.1.** Características del Equipo de la Institución

<b>HARDWARE</b>		
<b>Nombre</b>	<b>Características</b>	<b>Marca</b>
Monitor	LED 17 Pulgadas	LG
Case, Teclado, Mouse y Parlantes	Tipo Combo	Delux
Microprocesador	Core (TM) DUO - CPU 2.70 Ghz.	Intel(R)
Memoria RAM	4 GB	Kingston
Disco Duro	500 GB	Western Digital

**Fuente:** Unidad Educativa Republica del Japón B (2020)

**Tabla 1.2.** Características de Sistema Operativo

<b>SOFTWARE</b>	
<b>Tipo</b>	<b>Software Instalado</b>
Sistema operativo	Windows 7 PRO
Ofimática	Microsoft Office 2010
Antivirus	AvastFREE
Otras aplicaciones	WinRAR, Aboobe Acrobat, etc.

**Fuente:** Unidad Educativa Japón B (2020)

### **1.5.2. Justificación Económica**

El Cuaderno Pedagógico Web ahorrará tiempo a los padres de familia en el seguimiento de la asistencia y actividades diarias de sus hijos accediendo al sistema con el número de identificación del estudiante, los maestros optimizarán en el manejo de información cotidiana con el cuaderno pedagógico en línea y la administración optimizará los recursos económicos que son invertidos en la administración académica.

### **1.5.3. Justificación Social**

Uno de los principales pilares en los que se sustenta una sociedad es la educación, educar va más allá de formar. Se forma en conocimientos y se educa en la suma de conocimientos, valores y actitudes. Por lo tanto, el sistema beneficiará a toda la comunidad Educativa.

El Cuaderno Pedagógico Web podrá beneficiar directamente a los Maestros en el registro de las actividades cotidianas como asistencias, Padres de familia, estudiantes y la Administración de la Unidad Educativa, acortando tiempo y espacio.

## **1.6. METODOLOGÍA**

El presente proyecto será elaborado bajo algunos métodos que servirá para el análisis, diseño y posteriormente la implementación y la entrega del Cuaderno Pedagógico Web de la Unidad Educativa.

### **1.6.1. Scrum**

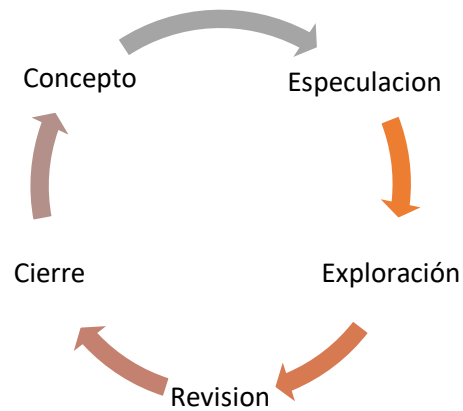
La Metodología Scrum es un proceso de desarrollo de software iterativo y creciente utilizado, comúnmente, en entornos basados en el desarrollo ágil de software.

Scrum al ser una metodología de desarrollo ágil tiene como base la idea de creación de ciclos breves para el desarrollo, que comúnmente se llaman iteraciones y que en Scrum se llamarán "Sprints".



Para entender el ciclo de desarrollo de Scrum es necesario conocer las fases que definen el ciclo de desarrollo ágil:

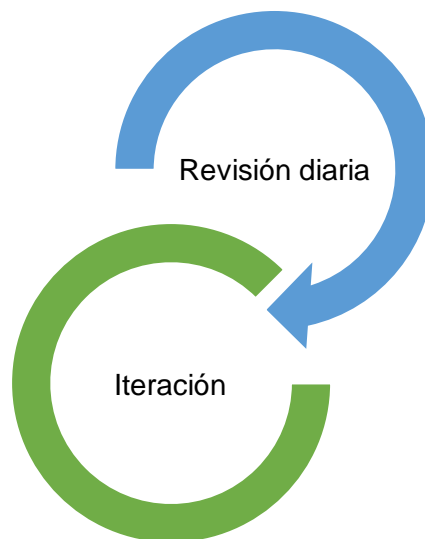
**Figura 1.2** El ciclo de desarrollo de Scrum.



**Fuente:** M. Trigas (2012)

Scrum gestiona estas iteraciones a través de reuniones diarias, uno de los elementos fundamentales de esta metodología.

**Figura 1.3.** Ciclo principal de Scrum



**Fuente:** M. Trigas (2012)

## **1.6.2. Método de Ingeniería**

### **1.6.2.1. Métricas de la Calidad de Software**

El objetivo de la ingeniería del software es desarrollar y producir software de alta calidad. Para lograr este objetivo, es fundamental aplicar métodos y herramientas efectivos dentro del contexto de un proceso maduro de desarrollo de software. Para las evaluaciones que se quieran obtener es necesario la utilización de medidas técnicas, que evalúan la calidad de manera objetiva. Métricas que definen la calidad del software: exactitud, estructuración o modularidad, pruebas, mantenimiento.

### **1.6.2.2. Características de la calidad del software**

Las características generales de la calidad del software se aplican a las aplicaciones Web y nos proporcionan una base útil para evaluar la calidad de los sistemas Web.

El modelo de calidad ISO/IEC 9216 (Internacional Standard “Information technology – Software Product Quality”) define 6 características de calidad que debe de cumplir toda aplicación Web para que sea un producto de calidad.

#### **1. Funcionalidad:**

Es la capacidad de un producto software de satisfacer los requisitos funcionales prescriptos y las necesidades implícitas de los usuarios.

#### **2. Fiabilidad:**

Es la capacidad de un producto software de mantener su nivel de desempeño, bajo condiciones establecidas, por un periodo de tiempo.

#### **3. Usabilidad:**

Es la capacidad de un producto software de ser comprendido, aprendido, usado, atractivo y conforme con las reglamentaciones y guías de usabilidad.

#### 4. Eficiencia:

Es la capacidad de un producto software de proporcionar un rendimiento apropiado, de acuerdo a la cantidad de recursos usados bajo condiciones establecidas.

#### 5. Mantenibilidad:

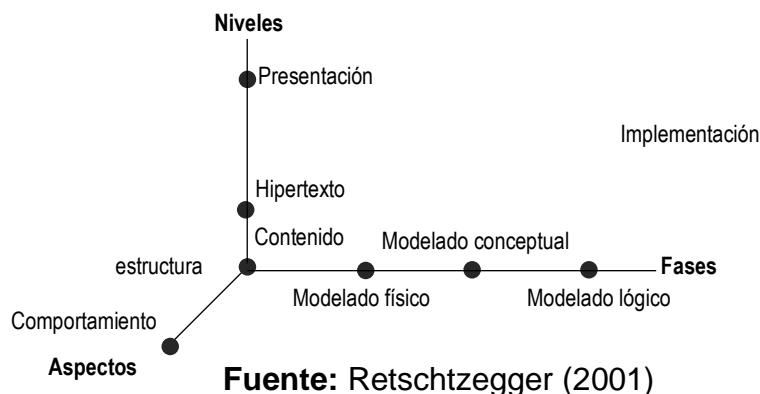
Es la capacidad de un producto software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, en los requisitos o en las especificaciones funcionales.

#### 6. Portabilidad:

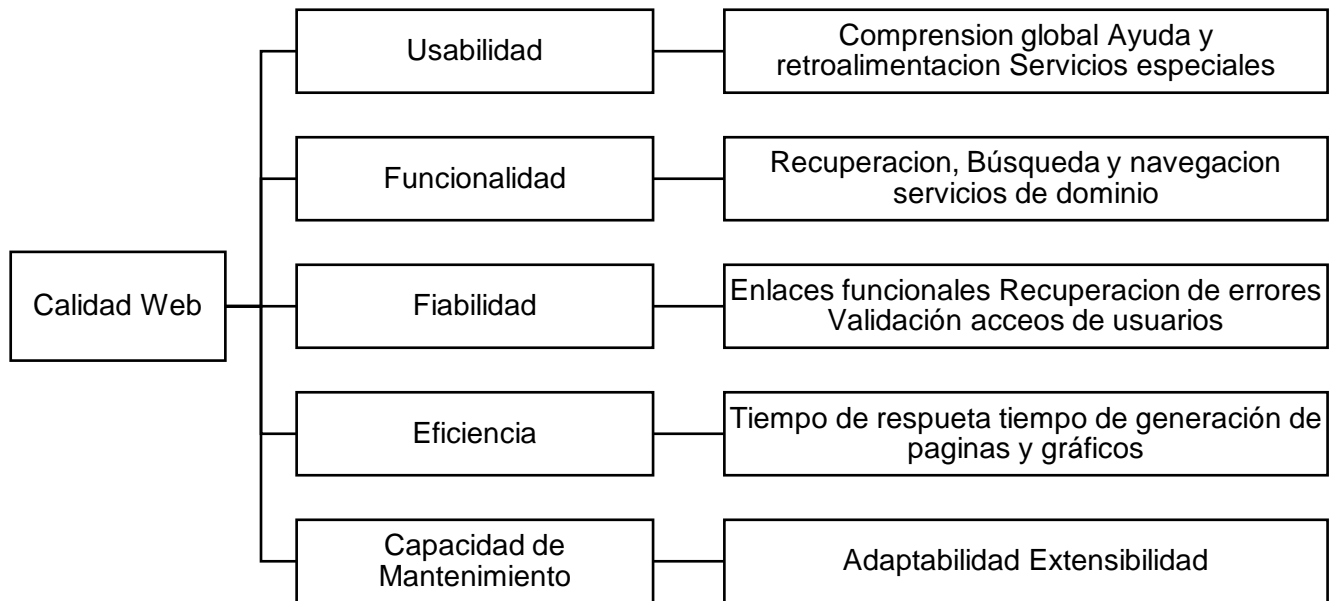
Es la capacidad de un producto software de ser transferido de un ambiente a otro. Nota: El ambiente puede ser organizacional, de software o de hardware.

Olsina et al. (2001) han desarrollado un árbol de requisitos de calidad que identifica un conjunto de atributos que conducen a aplicaciones Web de alta calidad

**Figura 1.4. Dimensiones de calidad**



**Figura 1.5. Calidad Aplicación Web**



**Fuente:** (OSL 99)

## 1.7. COSTO Y BENEFICIO

### 1.7.1. Cocomo II

Los objetivos principales que se tuvieron en cuenta para construir el modelo COCOMO II fueron:

- Desarrollar un modelo de estimación de costo y cronograma de proyectos de software que se adaptara tanto a las prácticas de desarrollo de la década del 90 como a las futuras.
- Construir una base de datos de proyectos de software que permitiera la calibración continua del modelo, y así incrementar la precisión en la estimación.
- Implementar una herramienta de software que soportara el modelo.
- Proveer un marco analítico cuantitativo y un conjunto de herramientas y técnicas que evaluaran el impacto de las mejoras tecnológicas de software sobre los costos y tiempos en las diferentes etapas del ciclo de vida de desarrollo.

COCOMO II está compuesto por tres modelos denominados:

- Composición de Aplicación
- Diseño Temprano
- Post-Arquitectura.

Éstos surgen en respuesta a la diversidad del mercado actual y futuro de desarrollo de software. Esta diversidad podría representarse con el siguiente esquema:

**Tabla 1.3.** Aplicaciones desarrolladas por usuarios finales

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

**Fuente:** Boehm (1995)

**Aplicaciones desarrolladas por Usuarios Finales:** En este sector se encuentran las aplicaciones de procesamiento de información generadas directamente por usuarios finales, mediante la utilización de generadores de aplicaciones tales como planillas de cálculo, sistemas de consultas, etc.

**Generadores de Aplicaciones:** En este sector operan firmas como Lotus, Microsoft, Novell, Borland con el objetivo de crear módulos pre empaquetados que serán usados por usuarios finales y programadores.

**Aplicaciones con Componentes:** Sector en el que se encuentran aquellas aplicaciones que son específicas para ser resueltas por soluciones pre-empaquetadas, pero son lo suficientemente simples para ser construidas a partir de componentes interoperables. Componentes típicas son constructores de interfases gráficas, administradores de bases de datos, buscadores inteligentes de datos, componentes de dominio-específico (medicina, finanzas, procesos industriales, etc.). Estas aplicaciones son generadas por un equipo reducido de personas, en pocas semanas o meses.

**Sistemas Integrados:** Sistemas de gran escala, con un alto grado de integración entre sus componentes, sin antecedentes en el mercado que se puedan tomar como base. Porciones de estos sistemas pueden ser desarrolladas a través de la composición de aplicaciones.

**Infraestructura:** Área que comprende el desarrollo de sistemas operativos, protocolos de redes, sistemas administradores de bases de datos, etc. Incrementalmente este sector direccionará sus soluciones, hacia problemas genéricos de procesamiento distribuido y procesamiento de transacciones. Firmas representativas son Microsoft, Oracle, SyBase, Novell y NeXT.

## 1.8. HERRAMIENTAS

Las herramientas para el desarrollo del Sistema “Registro pedagógico Web” se detalla a continuación:

### 1.8.1. Plataforma del sistema

Para un correcto funcionamiento del Cuaderno Pedagógico Web, el servidor Web debe estar instalado sobre un sistema Operativo compatible.

### 1.8.2. Servidor

El componente de los servidores que se utilizará para el desarrollo del sistema de Registro Pedagógico Web, son lo siguiente:

**Tabla 1.4.** Aplicaciones desarrolladas por usuarios finales

<i>Back END</i>	
Lenguaje de Programación	PHP
Servidor de base de datos	MariaDB
Servidor de plataforma	XAMPP
Servidor Web	Apache
Gestor gráfico	phpMyAdmin
<i>Front END</i>	
Bootstrap. La biblioteca HTML, CSS y JS y la Plantilla AdminLTE para el panel de administración	

**Fuente:** coders free (2019)

**Figura: 1.6.** Diferencia entre Backend y Frontend



**Fuente:** coders free (2021)

### 1.8.3. Framework

Para este proyecto se utilizará Laravel, que permite el uso de una sintaxis elegante y expresiva para crear código de forma sencilla y permitiendo multitud de funcionalidades.

Laravel, es un marco de trabajo o Framework gratuito (código abierto) que facilita el desarrollo de aplicaciones con el lenguaje PHP y Bases de Datos, Laravel se enfoca en el desarrollo de aplicaciones limpias y elegantes.

### 1.8.4. Herramientas de diseño

Para el tratamiento de imágenes digitales se utilizará la aplicación Adobe Photoshop CC, además se trabajará con el mejor panel de control de código abierto. Construido sobre Bootstrap, AdminLTE proporciona una gama de componentes receptivos, reutilizables y de uso común.

**Tabla 1.5.** Los requerimientos de Hardware para el desarrollo

Procesador	Core i3 o superior
Memoria RAM	4 GB o superior.
Disco Duro	300 Gb.

**Fuente:** Elaboración propia

## **1.9. LÍMITES Y ALCANCES**

### **1.9.1. Limites**

- El sistema denominado Cuaderno pedagógico web, no tendrá ninguna conexión con el sistema del Ministerio de Educación.
- No será posible registrar información de activos fijos de la unidad educativa
- El sistema no generará información para realizar trámites en la dirección distrital.
- No se enviará información a través de internet a ninguna otra instancia.

### **1.9.2. Alcances**

- El Cuaderno pedagógico web, centralizará las notas de todos los maestros
- El Sistema registrará las asistencias diarias para consultar desde cualquier dispositivo móvil o de escritorio.
- Generará un reporte centralizado de notas para facilitar al maestro
- El sistema será capaz de mostrar información pertinente a la dirección de la unidad educativa
- Todos los estudiantes de la unidad educativa tendrán acceso de visualización de sus asistencias y calificaciones diarias.

## **1.10. APORTES**

El aporte del sistema denominado Cuadernos pedagógico web, tiene como característica principal haciendo uso del internet y adaptable para todos los dispositivos móviles y de escritorio, aplicando todas las herramientas pertinentes para su desarrollo del Sistema, así mismo los niveles de seguridad para diferentes usuarios.



## **CAPÍTULO II**

---

### **MARCO TEÓRICO**

## **CAPITULO II**

### **2. MARCO TEÓRICO**

#### **2.1. INTRODUCCIÓN**

En el presente capítulo se abordará conceptos y definiciones generales referente al sistema denominado “Cuaderno Pedagógico Web como Herramienta del Maestro para el Seguimiento de Actividades Pedagógicas”.

La tecnología web es el resultado de la integración de dos plataformas tecnológicas: Internet y multimedia, lo cual ha requerido la posibilidad de comunicación, llevándolos a evolucionar desde la educación por correspondencia hasta los entornos educativos basados en la Web. Dando lugar a que estos últimos sean considerados como el recurso didáctico del futuro para la educación.

#### **2.2. CUADERNO PEDAGÓGICO**

Un cuaderno pedagógico escrito, con el formato de una narrativa personal, que va más allá de contar la tarea de investigación de los estudiantes, se centra en la descripción de las acciones de enseñanza.

##### **¿Qué expondrá el docente en el Cuaderno Pedagógico?**

El maestro da cuenta de cómo ha llevado adelante la actividad de investigación con sus estudiantes, por ejemplo: su planificación, asistencia y la elección del tema en términos curriculares, la forma en que llevó adelante la actividad, el tiempo empleado, su organización, etc.

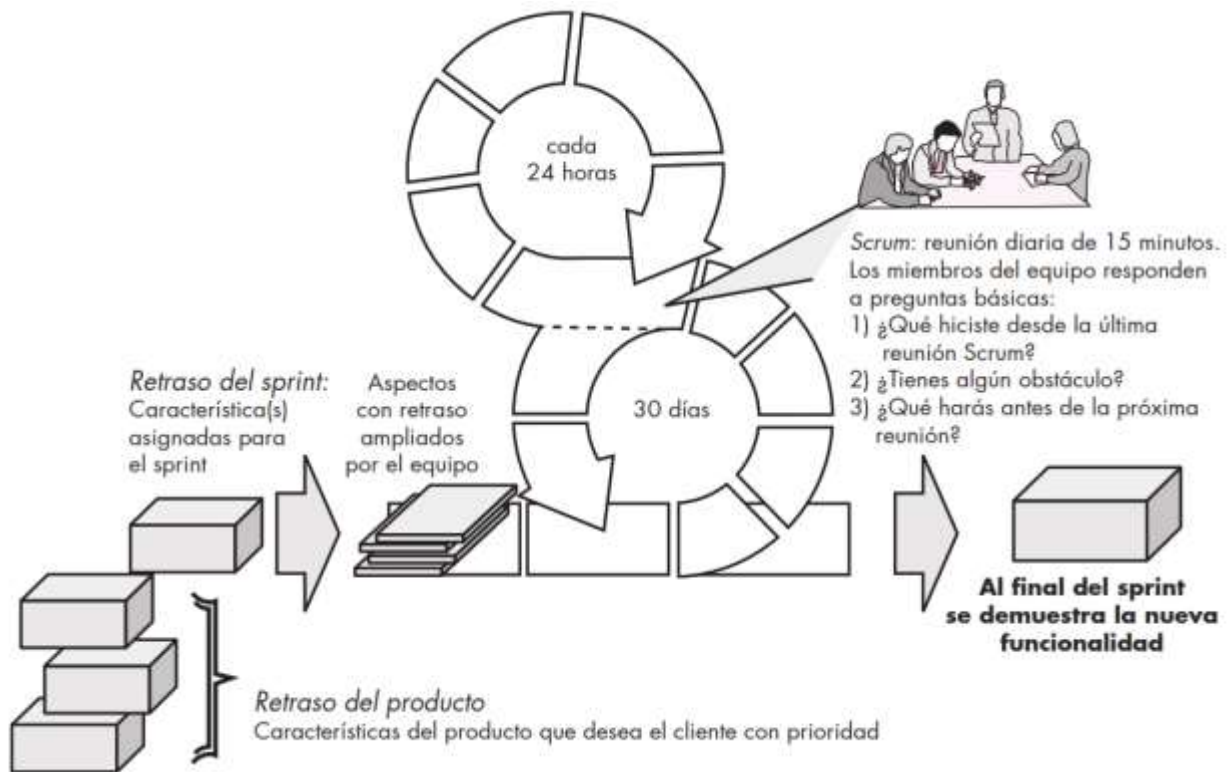
En el Cuaderno pedagógico, el maestro tiene que exponer sus opiniones sobre el proceso que ha llevado a cabo y los resultados obtenidos, como así también brindar un diagnóstico sobre el impacto del mismo en el aula y en los aprendizajes de su clase.

### 2.3. METODOLOGÍA SRUM

Scrum (nombre que proviene de cierta jugada que tiene lugar durante un partido de rugby) es un método de desarrollo ágil de software concebido por Jeff Sutherland y su equipo de desarrollo a principios de la década de 1990. En años recientes, Schwaber y Beedle han desarrollado más los métodos Scrum. (Pressman, 2010, pág. 69)

Los principios Scrum son congruentes con el manifiesto ágil y se utilizan para guiar actividades de desarrollo dentro de un proceso de análisis que incorpora las siguientes actividades estructurales: requerimientos, análisis, diseño, evolución y entrega.

**Figura 2.1.** Flujo del Proceso SCRUM



**Fuente:** Ingeniería del Software (Pressman, 2010)

### **2.3.1.1. Componentes de Scrum**

Scrum se puede dividir de forma general en 3 fases, que podemos entender como reuniones. Las reuniones forman parte de los artefactos de esta metodología junto con los roles y los elementos que lo forman. (Trigás Gallego, 2012, pág. 35)

#### **2.3.1.1.1. Las reuniones**

##### **1. Planificación del Backlog**

En esta fase se definirá también la planificación del Sprint 0, en la que se decidirá cuáles van a ser los objetivos y el trabajo que hay que realizar para esa iteración. Se obtendrá además en esta reunión un Sprint Backlog, que es la lista de tareas y que es el objetivo más importante del Sprint.

##### **2. Seguimiento de Sprint**

En esta fase se hacen reuniones diarias en las que las 3 preguntas principales para evaluar el avance de las tareas serán:

- ¿Qué trabajo se realizó desde la reunión anterior?
- ¿Qué trabajo se hará hasta una nueva reunión?
- Inconvenientes que han surgido y qué hay que solucionar para poder continuar.

##### **3. Revisión de Sprint**

Cuando se finaliza el Sprint se realizará una revisión del incremento que se ha generado. Se presentarán los resultados finales y una demo o versión, esto ayudará a mejorar el feedback con el cliente. (Trigás Gallego, 2012, pág. 35)

### 2.3.1.1.2. Roles

Los roles se dividen en 2 grupos: cerdos y gallinas, esto surge en el chiste sobre un cerdo y una gallina y su intención de poner un restaurante, según Trigás Gallego (2012)

#### 1. LOS CERDOS (COMPROMETIDOS)

Son las personas que están comprometidas con el proyecto y el proceso de Scrum.

**Product Owner:** Es la persona que toma las decisiones, y es la que realmente conoce el negocio del cliente y su visión del producto. Se encarga de escribir las ideas del cliente, las ordena por prioridad y las coloca en el Product Backlog.

**ScrumMaster:** Es el encargado de comprobar que el modelo y la metodología funciona. Eliminará todos los inconvenientes que hagan que el proceso no fluya e interactuará con el cliente y con los gestores.

**Equipo De Desarrollo:** suele ser un equipo pequeño de unas 5-9 personas y tienen autoridad para organizar y tomar decisiones para conseguir su objetivo. Está involucrado en la estimación del esfuerzo de las tareas del Backlog.

#### 2. LAS GALLINAS (INVOLUCRADOS)

Aunque no son parte del proceso de Scrum, es necesario que parte de la retroalimentación dé la salida del proceso y así poder revisar y planear cada sprint.

- **Usuarios:** Es el destinatario final del producto.
- **Stakeholders:** Las personas a las que el proyecto les producirá un beneficio. Participan durante las revisiones del Sprint.
- **Managers:** Toma las decisiones finales participando en la selección de los objetivos y de los requisitos.

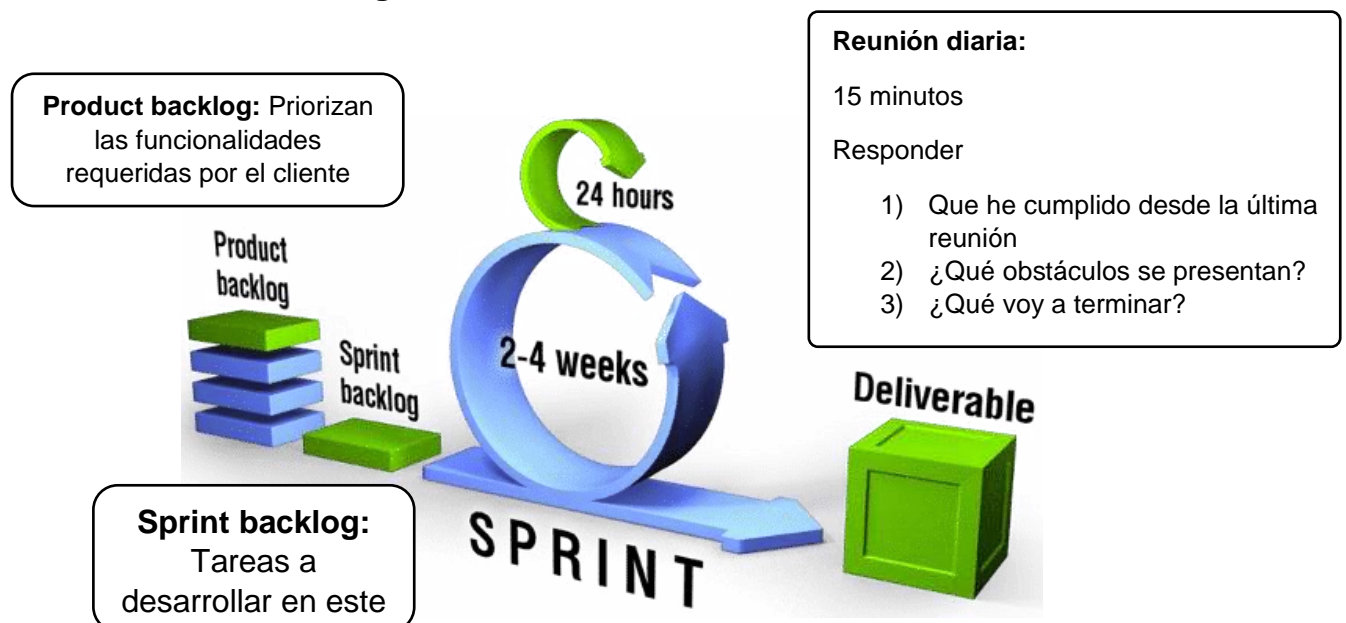
### 2.3.1.2. Elementos de Scrum

Los elementos que forman a Scrum son:

- **Product Backlog:** lista de necesidades del cliente.
- **Sprint Backlog:** lista de tareas que se realizan en un Sprint.
- **Incremento:** parte añadida o desarrollada en un Sprint, es una parte terminada y totalmente operativa.

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

**Figura 2.2.** Ciclo de desarrollo Scrum



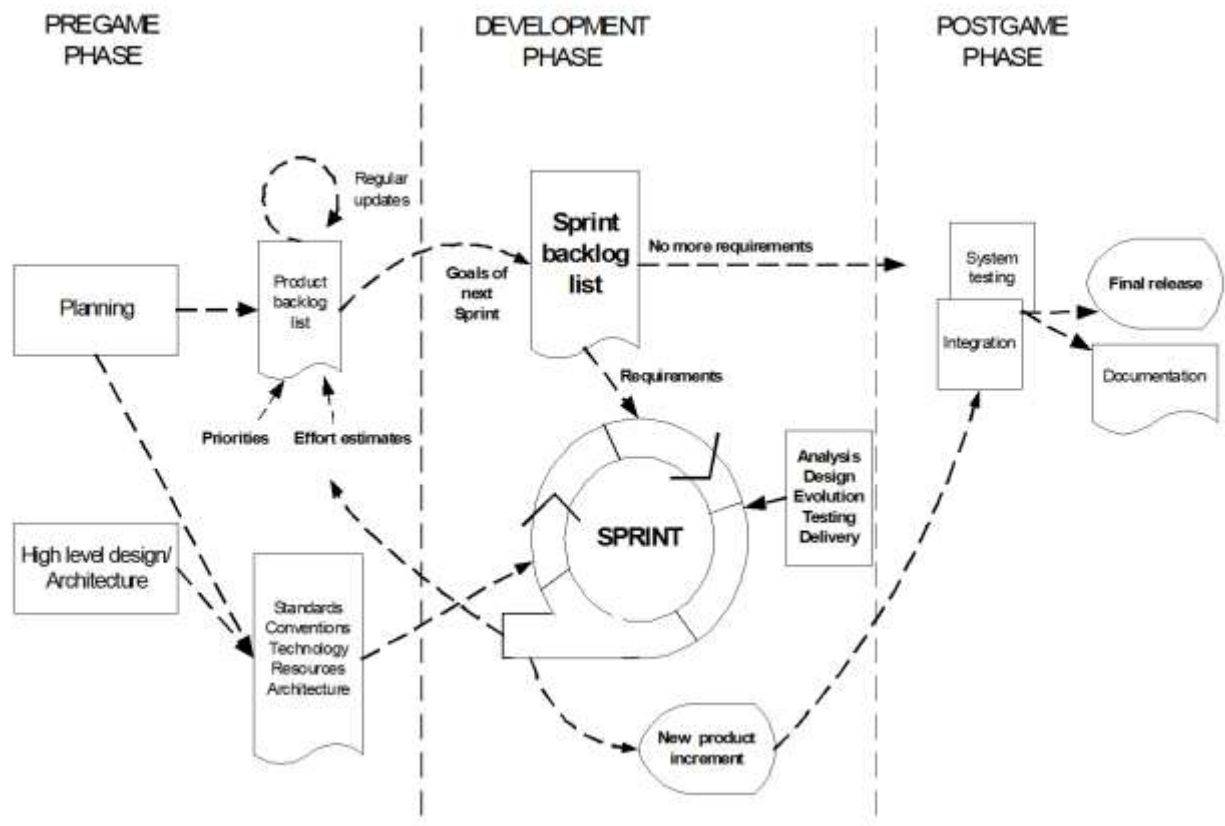
**Fuente:** Trigás Gallego (2012)

### 2.3.1.3. Fases de Scrum

Se ha realizado de esta manera una guía por todo el proceso de creación de un proyecto Scrum en el que se van realizando las diferentes fases en forma de ciclos, hasta completar todas las tareas del Backlog. Scrum consta de tres fases:

- ✓ Pregame (Antes del juego)
- ✓ Development (Desarrollo)
- ✓ Postgame (Después del juego)

**Figura 2.3.** El proceso o las tres fases de Scrum



**Fuente:** Adriana peralta (2003)

### **2.3.1.3.1. Pre-game (Antes del Juego)**

La fase previa al juego incluye dos subfases:

- Planificación
- Arquitectura / Diseño de alto nivel

#### **Planificación**

- Definición del sistema en desarrollo.
- Se crea una lista de Product Backlog que contiene todos los requisitos que se conocen actualmente.
- Se priorizan los requisitos y se estima el esfuerzo necesario para su implementación.
- La lista de Backlog de productos se actualiza constantemente con elementos nuevos y más detallados, así como con estimaciones más precisas y nuevos pedidos de prioridad.
- La planificación también incluye la definición del equipo del proyecto, las herramientas y otros recursos, la evaluación de riesgos y los problemas de control, las necesidades de capacitación y la aprobación de la gestión de verificación.

#### **Arquitectura**

- El diseño de alto nivel del sistema, incluida la arquitectura, se planifica sobre la base de los elementos actuales en el Product Backlog.
- En caso de una mejora en un sistema existente, los cambios necesarios para implementar los elementos del Backlog se identifican junto con los problemas que pueden causar.
- Se lleva a cabo una reunión de revisión del diseño para revisar las propuestas para la implementación y las decisiones se toman sobre la base de esta revisión.



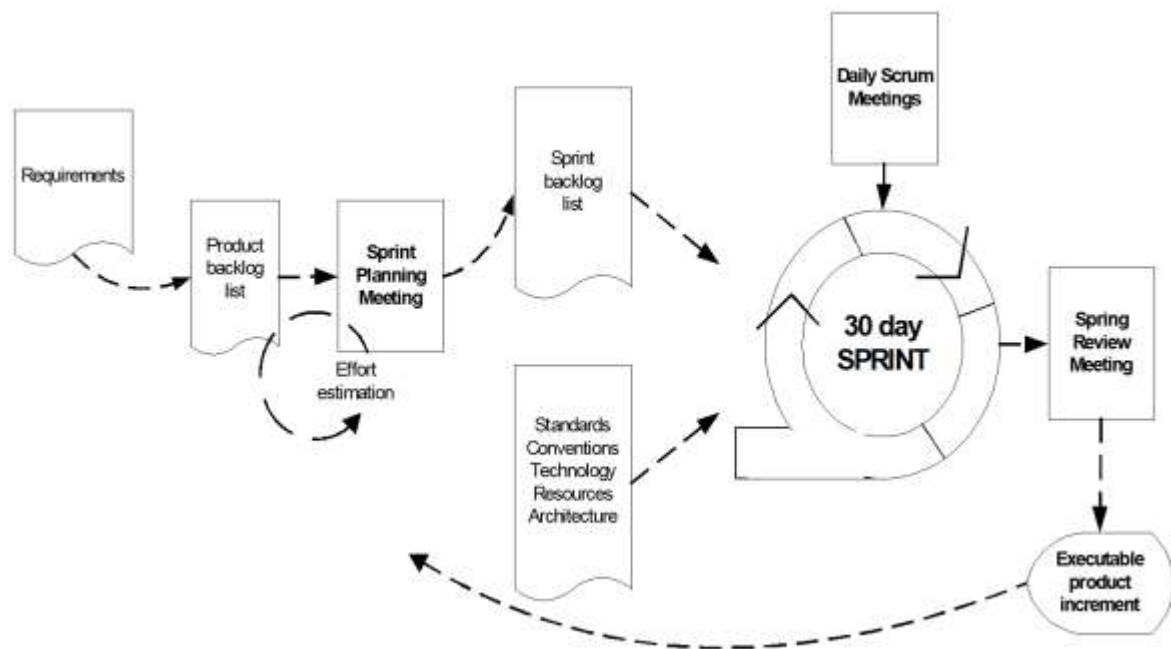
### 2.3.1.3.2. Development phase (Fase de desarrollo)

- Esta fase se trata como una "caja negra" donde se espera lo impredecible.
- El sistema se desarrolla en Sprints

#### Sprint

- 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: requisitos, análisis, diseño, evolución y fases de entrega.
- Está previsto que un Sprint dure de una semana a un mes.
- Las herramientas de trabajo del equipo son Sprint Planning Meetings, Sprint Backlog y Daily Scrum.

**Figura 2.4.** Las actividades que se desarrollan durante del Sprint



**Fuente:** Adriana peralta (2003)

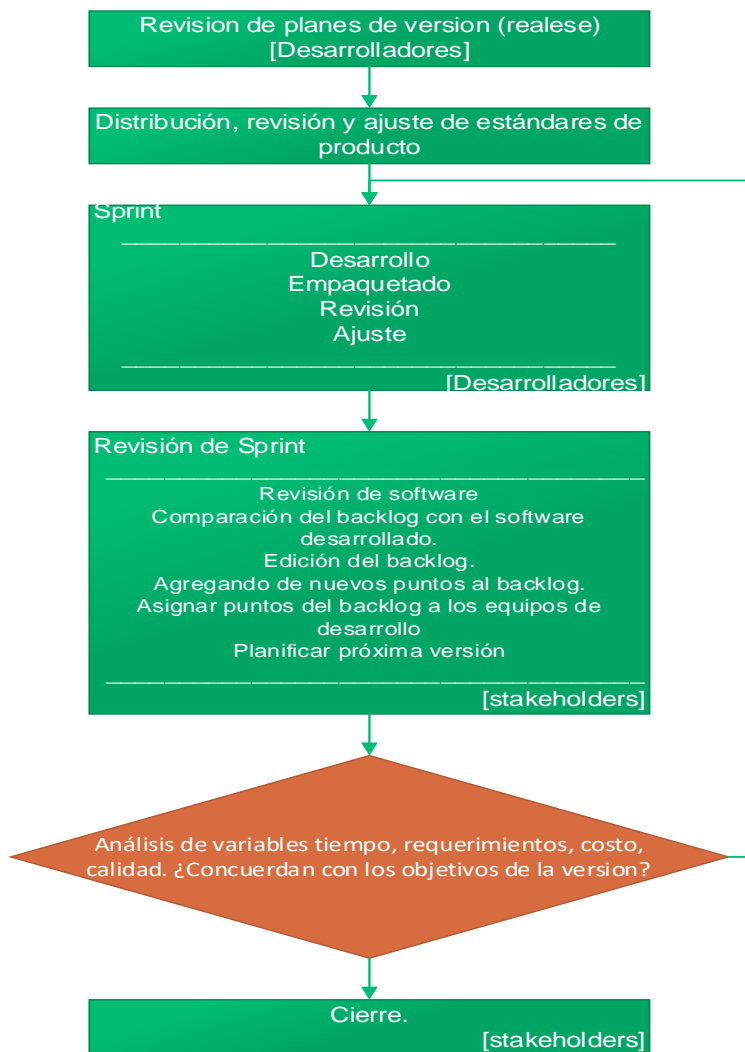
- **Reunión de planificación de Sprint:** una reunión de planificación de Sprint es una reunión de dos fases organizada por el Scrum Master. Los clientes, usuarios, administración, Product Owner y Scrum Team participan en la primera fase de la reunión para decidir los objetivos y la funcionalidad del próximo Sprint (ver Sprint Backlog a continuación). La segunda fase de la reunión la llevan a cabo el Scrum Master y el Scrum Team centrándose en cómo se implementa el incremento de producto durante el Sprint.
- **Sprint Backlog:** Sprint Backlog es el punto de partida para cada Sprint. Es una lista de elementos del Backlog de productos seleccionados para ser implementados en el próximo Sprint. Los elementos son seleccionados por el equipo Scrum junto con el Scrum Master y el propietario del producto en la reunión de planificación del Sprint, sobre la base de los elementos priorizados y los objetivos establecidos para el Sprint. A diferencia del Product Backlog, el Sprint Backlog es estable hasta que se completa el Sprint (es decir, 30 días). Cuando se completan todos los elementos del Sprint Backlog, se entrega una nueva iteración del sistema.
- **Reunión diaria de Scrum:** las reuniones diarias de Scrum se organizan para realizar un seguimiento del progreso del Scrum Team de forma continua y también sirven como reuniones de planificación: qué se ha hecho desde la última reunión y qué se debe hacer antes de la próxima. También se discuten y controlan problemas y otros asuntos variables en esta breve reunión (aproximadamente 15 minutos) que se realiza diariamente. Se busca, identifica y elimina cualquier deficiencia o impedimento en el proceso de desarrollo de sistemas o en las prácticas de ingeniería para mejorar el proceso. El Scrum Master dirige las reuniones de Scrum. Además del equipo Scrum, también la dirección, por ejemplo, puede participar en la reunión.
- **Reunión de revisión del Sprint:** En el último día del Sprint, el Equipo Scrum y el Scrum Master presentan los resultados (es decir, el incremento del producto de trabajo) del Sprint a la gerencia, los clientes, los usuarios y el Product Owner en una reunión informal. Los participantes evalúan el incremento de producto y toman la decisión sobre las siguientes actividades.

### 2.3.1.3.3. Post-game (Después del Juego)

Esta fase se ingresa cuando se ha llegado a un acuerdo de que se completan las variables ambientales, como los requisitos.

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 realiza durante la fase posterior al juego, incluidas las tareas como la integración, las pruebas del sistema y la documentación.

**Figura 2.5. Esquema de las fases de desarrollo en Scrum**



#### REVISIÓN DE PLANES DE VERSIÓN:

Se revisa que hay que hacer y en qué punto está la distribución actual.

#### SPRINT:

Es la fase de desarrollo iterativa.

Desarrollo: Análisis, implementación, testing.

*Empaquetar:* Generar paquetes ejecutables

*Revisión:* Resolución de problemas y se añaden

nuevos ítems.

*Ajustes:* Uso de los ajustes para mejorar el

producto.

#### SPRINT REVIEW:

Después del Sprint se hace una reunión con el ScrumMaster donde se revisa el producto del Sprint anterior y en el que se pueden añadir puntos nuevos al backlog.

#### Cierre:

En esta fase se encuentran las típicas actividades de fin de proyecto como, hacer una versión distribuible, testear, marketing etc....

Fuente: Trigás Gallego (2012)

## 2.4. UML - LENGUAJE UNIFICADO DE MODELADO

UML es un lenguaje estándar que sirve para escribir los planos del software, puede utilizarse para visualizar, especificar, construir y documentar todos los artefactos que componen un sistema con gran cantidad de software. UML puede usarse para modelar desde sistemas de información hasta aplicaciones distribuidas basadas en Web, pasando por sistemas empotrados de tiempo real. UML es solamente un lenguaje por lo que es sólo una parte de un método de desarrollo software, es independiente del proceso, aunque para que sea óptimo debe usarse en un proceso dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental.

UML es un lenguaje por que proporciona un vocabulario y las reglas para utilizarlo, además es un lenguaje de modelado lo que significa que el vocabulario y las reglas se utilizan para la representación conceptual y física del sistema.

UML es un lenguaje que nos ayuda a interpretar grandes sistemas mediante gráficos o mediante texto obteniendo modelos explícitos que ayudan a la comunicación durante el desarrollo ya que, al ser estándar, los modelos podrán ser interpretados por personas que no participaron en su diseño (e incluso por herramientas) sin ninguna ambigüedad. En este contexto, UML sirve para especificar, modelos concretos, no ambiguos y completos.

Debido a su estandarización y su definición completa no ambigua, y aunque no sea un lenguaje de programación, UML se puede conectar de manera directa a lenguajes de programación como Java, C++ o Visual Basic, esta correspondencia permite lo que se denomina como ingeniería directa (obtener el código fuente partiendo de los modelos) pero además es posible reconstruir un modelo en UML partiendo de la implementación, o sea, la ingeniería inversa.

UML proporciona la capacidad de modelar actividades de planificación de proyectos y de sus versiones, expresar requisitos y las pruebas sobre el sistema, representar todos sus detalles, así como la propia arquitectura. Mediante estas capacidades se

obtiene una documentación que es válida durante todo el ciclo de vida de un proyecto.

### 2.4.1. Concepción del UML

El UML es la creación de Grady Booch, James Rumbaugh e Ivar Jacobson. Estos caballeros, apodados recientemente “Los tres amigos”. Trabajan en empresas distintas durante la década de los años ochenta y principios de los noventa y cada uno diseñó su propia metodología para el análisis y diseño orientado a objetos. (Joseph Schmuller, pag. 8)

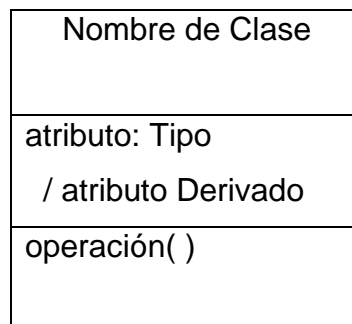
### 2.4.2. Diagramas UML

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. (Joseph Schmuller, pag. 8).

#### 2.4.2.1. Diagrama de clases

Muestran un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Estos diagramas son los más comunes en el modelado de sistemas orientados a objetos y cubren la vista de diseño estática o la vista de procesos estática (sí incluyen clases activas).

**Figura 2.6.** El símbolo UML de una clase.



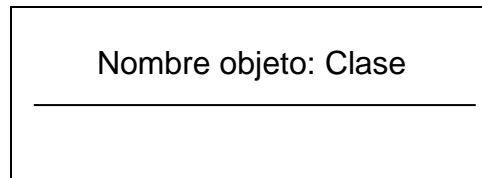
**Fuente:** Aprendiendo UML (Joseph Schmuller)

### 2.4.2.2. Diagrama de Objetos

Un objeto es una instancia de clase (una entidad que tiene valores específicos de los atributos y acciones).

La figura le muestra la forma en que el UML representa a un objeto. Vea que el símbolo es un rectángulo, como en una clase, pero el nombre está subrayado. El nombre de la instancia específica se encuentra a la izquierda de los dos puntos (:), y el nombre de la clase a la derecha.

**Figura 2.7.** El símbolo UML del Objeto



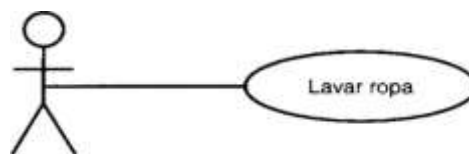
**Fuente:** Aprendiendo UML (Joseph Schmuller)

### 2.4.2.3. Diagrama Casos de uso

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Para los desarrolladores del sistema, ésta es una herramienta valiosa, ya que es una técnica de aciertos y errores para obtener los requerimientos del sistema desde el punto de vista del usuario. Esto es importante si la finalidad es crear un sistema que pueda ser utilizado por la gente en general (no sólo por expertos en computación).

La figura le muestra cómo representaría esto en un diagrama de casos de uso UML.

**Figura 2.8.** El símbolo UML del Objeto



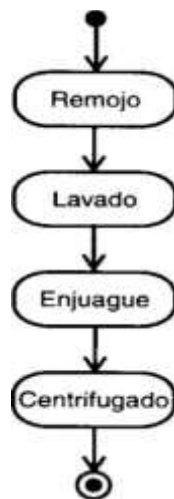
**Fuente:** Aprendiendo UML (Joseph Schmuller)

#### 2.4.2.4. Diagrama de estados

En cualquier momento, un objeto se encuentra en un estado particular, la luz está encendida o apagada, el auto en movimiento o detenido, la persona leyendo o cantando, etc. El diagrama de estados UML captura esa pequeña realidad.

El diagrama de estados UML, que aparece en la figura, captura esta pequeña realidad. La figura muestra las transiciones de la lavadora de un estado al otro.

**Figura 2.9.** Diagrama de estados UML

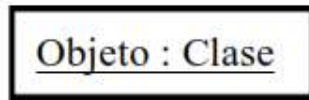


**Fuente:** Aprendiendo UML, pag. 11 (Joseph Schmuller)

#### 2.4.2.5. Diagrama de secuencias

Los diagramas de clases y los de objeto representan información estática. No obstante, en un sistema funcional los objetos interactúan entre sí, y tales interacciones suceden con el tiempo. El diagrama de secuencias UML muestra la mecánica de la interacción con base en tiempos. (Joseph Schmuller)

**Figura 2.10.** Rol de la Clase

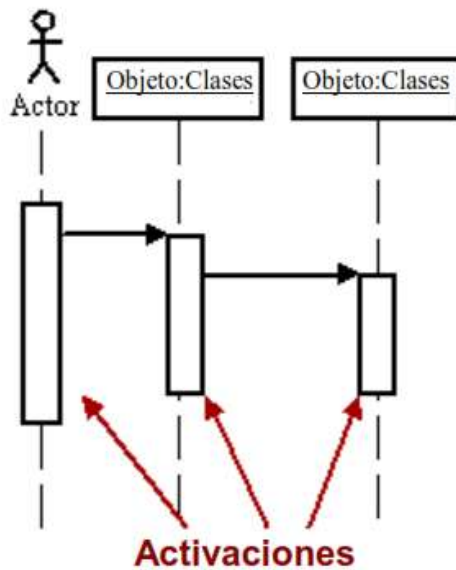


**Rol de la Clase**

El rol de la clase describe la manera en que un objeto se va a comportar en el contexto. No se listan los atributos del objeto.

**Fuente:** Catedra de proyecto

**Figura 2.11.** Activación



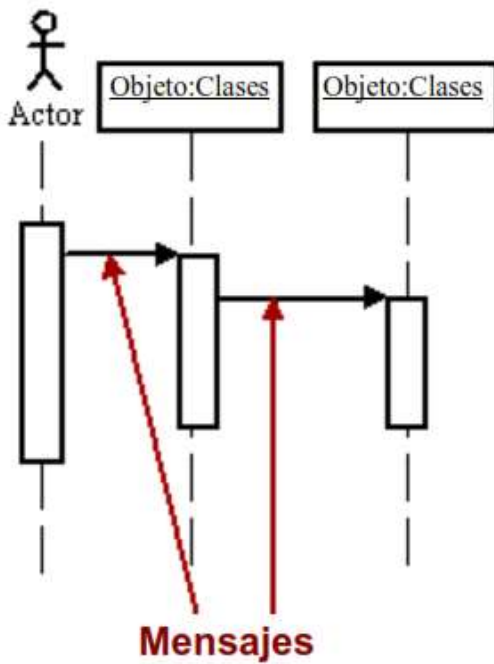
**Activación**

Los cuadros de activación representan el tiempo que un objeto necesita para completar una tarea.

**Fuente:** Catedra de proyecto



Figura 2.12. Mensaje



### Mensaje

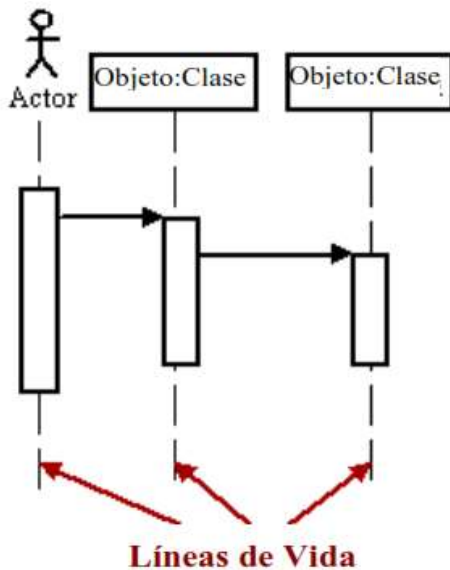
Los mensajes son flechas que representan comunicaciones entre objetos. Las medias flechas representan mensajes asincrónicos.

Los mensajes asincrónicos son enviados desde un objeto que no va a esperar una respuesta del receptor para continuar con sus tareas.

Flecha	Tipo de mensaje
	Simple
	Sincrónico
	Asincrónico
	Rechazado
	Time out

Fuente: Catedra de proyecto

Figura 2.13. Líneas de Vida

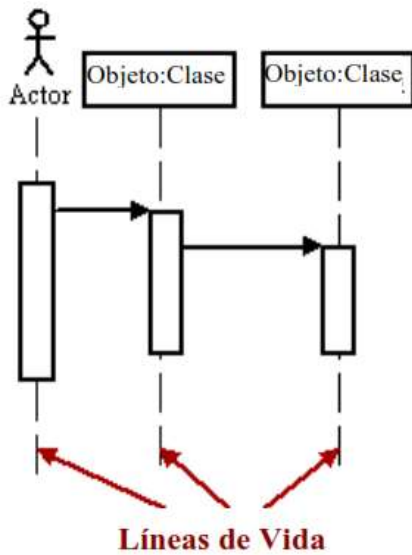


### Líneas de Vida

Las líneas de vida son verticales y en línea de puntos, ellas indican la presencia del objeto durante el tiempo.

Fuente: Catedra de proyecto

**Figura 2.14** Destrucción de Objetos

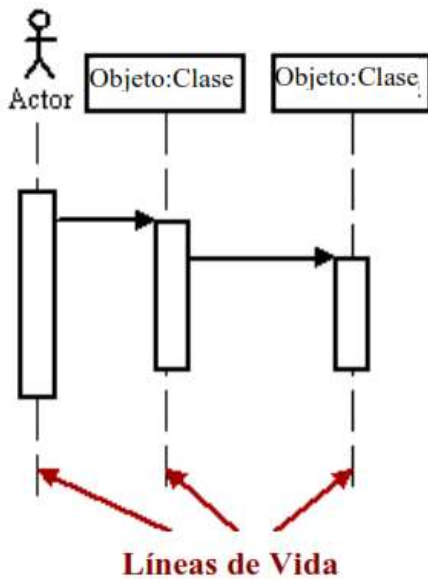


### **Destrucción de Objetos**

Los objetos pueden ser eliminados tempranamente usando una flecha etiquetada "<<destruir>>" que apunta a una X.

**Fuente:** Catedra de proyecto

**Figura 2.15.** Loops



### **Loops**

Una repetición o loop en un diagrama de secuencias, es representado como un rectángulo. La condición para abandonar el loop se coloca en la parte inferior entre corchetes [ ].

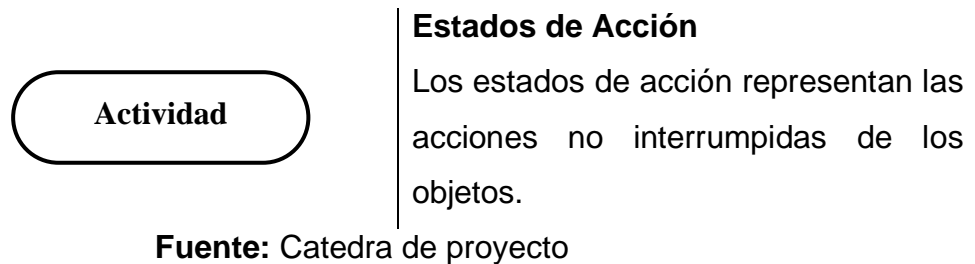
**Fuente:** Catedra de proyecto

#### 2.4.2.6. Diagrama de actividades

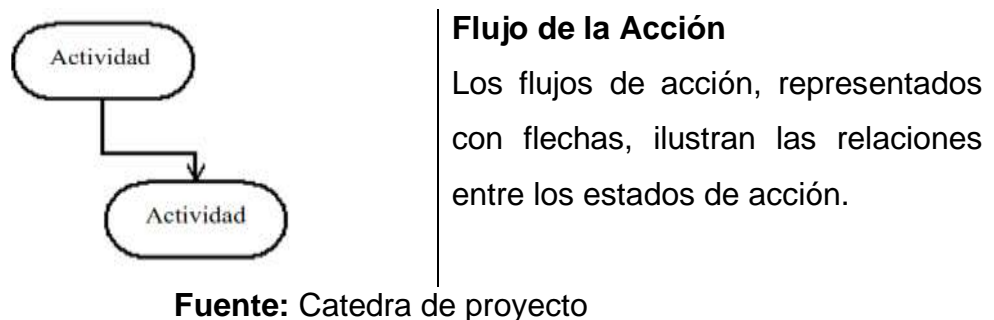
Las actividades que ocurren dentro de un caso de uso o dentro del comportamiento de un objeto se dan, normalmente en secuencia. (Joseph Schmuller, pag. 12)

ilustra la naturaleza dinámica de un sistema mediante el modelado del flujo ocurrente de actividad en actividad. Una actividad representa una operación en alguna clase del sistema y que resulta en un cambio en el estado del sistema. Típicamente, los diagramas de actividad son utilizados para modelar el flujo de trabajo interno de una operación.

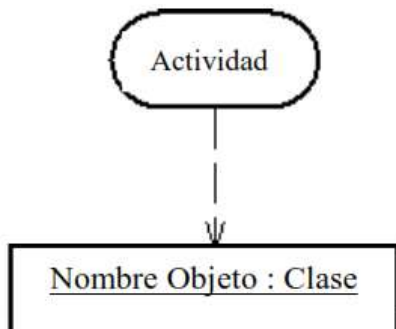
**Figura 2.16.** Estados de Acción



**Figura 2.17.** Flujos de Acción



**Figura 2.18.** Flujo de Objetos

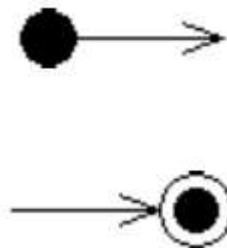


### Flujo de Objetos

El flujo de objetos se refiere a la creación y modificación de objetos por parte de actividades. Una flecha de flujo de objeto, desde una acción a un objeto, significa que la acción está creando o influyendo sobre dicho objeto. Una flecha de flujo de objeto, desde un objeto a una acción, indica que el estado de acción utiliza dicho objeto.

**Fuente:** Catedra de proyecto

**Figura: 2.19.** Estados Inicial y Final



### Flujo de la Acción

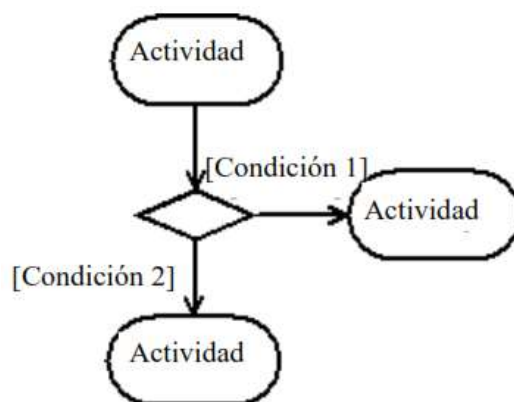
**Estado Inicial** Estado inicial de un estado de acción.

### Final State

Estado final de un estado de acción.

**Fuente:** Catedra de proyecto

**Figura 2.20.** Ramificación

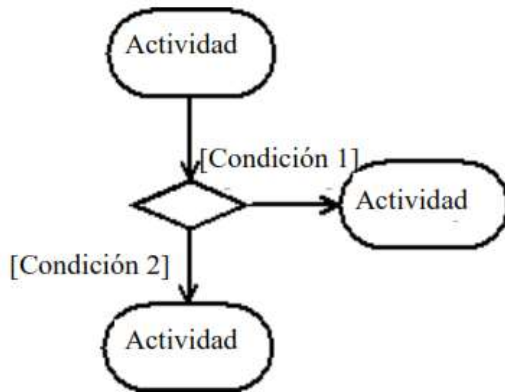


### Ramificación

Un rombo representa una decisión con caminos alternativos. Las salidas alternativas deben estar etiquetadas con una condición

**Fuente:** Catedra de proyecto

**Figura 2.21.** Simplificación

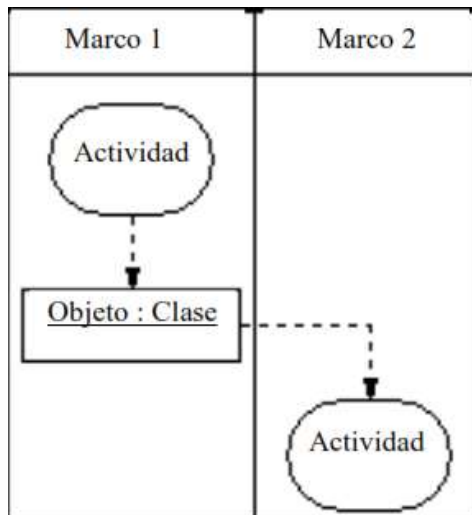


### Sincronización

Una barra de sincronización ayuda a ilustrar la ocurrencia de transiciones paralelas, así quedan representadas las acciones concurrentes.

**Fuente:** Catedra de proyecto

**Figura 2.22.** Marco de responsabilidad



### Marcos de Responsabilidad

Los marcos de responsabilidad agrupan a las actividades relacionadas en una misma columna.

**Fuente:** Catedra de proyecto

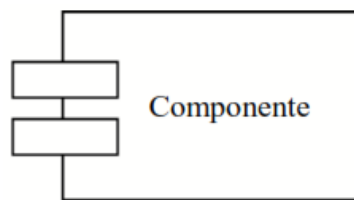
#### 2.4.2.7. Diagrama de colaboraciones

Los elementos de un sistema trabajan en conjunto para cumplir con los objetivos del sistema, y un lenguaje de modelado deberá contar con una forma de representar esto. El diagrama de colaboraciones UML (Joseph Schmuller, pag. 13)

### 2.4.2.8. Diagrama de componentes

Un diagrama de componentes describe la organización de los componentes físicos de un sistema.

**Figura 2.23.** Diagrama de componente UML



#### **Componente**

Un componente es un bloque de construcción física del sistema.

**Fuente:** Catedra de proyecto

### 2.4.2.9. Arquitectura

El desarrollo de un sistema con gran cantidad de software requiere que este sea visto desde diferentes perspectivas. Diferentes usuarios (usuario final, analistas, desarrolladores, integradores, jefes de proyecto...) siguen diferentes actividades en diferentes momentos del ciclo de vida del proyecto, lo que da lugar a las diferentes vistas del proyecto, dependiendo de qué interés más en cada instante de tiempo. (Alarcón, 2000, pag 24).

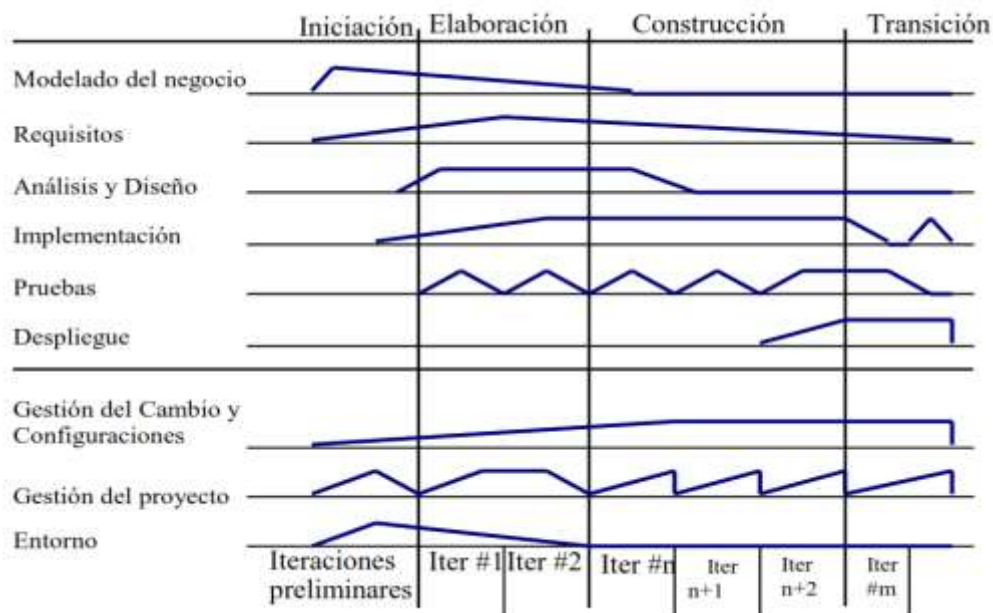
### 2.4.3. Ciclo de vida

Se entiende por ciclo de vida de un proyecto software a todas las etapas por las que pasa un proyecto, desde la concepción de la idea que hace surgir la necesidad de diseñar un sistema software, pasando por el análisis, desarrollo, implantación y mantenimiento del mismo y hasta que finalmente muere por ser sustituido por otro sistema. (Alarcón, 2000, pag 25)

Aunque UML es bastante independiente del proceso, para obtener el máximo rendimiento de UML se debería considerar un proceso que fuese:

- Dirigido por los casos de uso, o sea, que los casos de uso sean un artefacto básico para establecer el comportamiento del deseado del sistema, para validar la arquitectura, para las pruebas y para la comunicación entre las personas involucradas en el proyecto.
- Centrado en la arquitectura de modo que sea el artefacto básico para conceptualizar, construir, gestionar y hacer evolucionar el sistema.
- Un proceso iterativo, que es aquel que involucra la gestión del flujo de ejecutables del sistema, e incremental, que es aquel donde cada nueva versión corrige defectos de la anterior e incorpora nueva funcionalidad. Un proceso iterativo e incremental se denomina dirigido por el riesgo, lo que significa que cada nueva versión se ataca y reducen los riesgos más significativos para el éxito del proyecto.

**Figura 2.24.** Ciclo de vida del desarrollo software



**Fuente:** Alarcón R. (2000)

## **2.5. Herramientas de Implementación**

### **2.5.1. Algoritmos**

La palabra algoritmo se deriva de la palabra árabe alkhwarinmi, siendo el nombre de un matemático y astrónomo árabe que escribió un tratado sobre manipulación de números y ecuaciones en el siglo IX. (B. Ríos, 2014, pag 9)

- a) Un algoritmo es la solución a cualquier problema de computadora que involucra la ejecución de una serie de acciones en orden específico. Un procedimiento para resolver un problema en términos de:
  - a. Las acciones a ejecutarse
  - b. En orden en el cual estas acciones deben ejecutarse.
- b) Un algoritmo es el medio por el que se explica cómo puede resolverse un problema indicando paso a paso la solución
- c) Un algoritmo es una serie de operaciones detalladas y no ambiguas.
- d) Un algoritmo es un conjunto de reglas para resolver una cierta clase de problemas

### **2.5.2. Lenguaje de Programación PHP**

PHP es un lenguaje interpretado del lado del servidor que surge dentro de la corriente denominada código abierto (open source). Se caracteriza por su potencia, versatilidad, robustez y modularidad. Al igual que ocurre con tecnologías similares, los programas son integrados directamente dentro del código HTML. En este libro se explicará en detalle la sintaxis y el funcionamiento de este lenguaje, de momento se realiza a continuación una breve comparativa con las otras tecnologías del lado del servidor descritas previamente. (Cobo, Gómez, Pérez y Rocha, 2005, pag 99)

PHP viene de las palabras en inglés Hypertext Pre-Processor. Es considerado como un lenguaje de programación para aplicaciones web, se podría decir que su enfoque principal es desarrollar script que son interpretados por un servidor; es decir, es un lenguaje de programación interpretado. (Torrez, 2016, pag. 57)



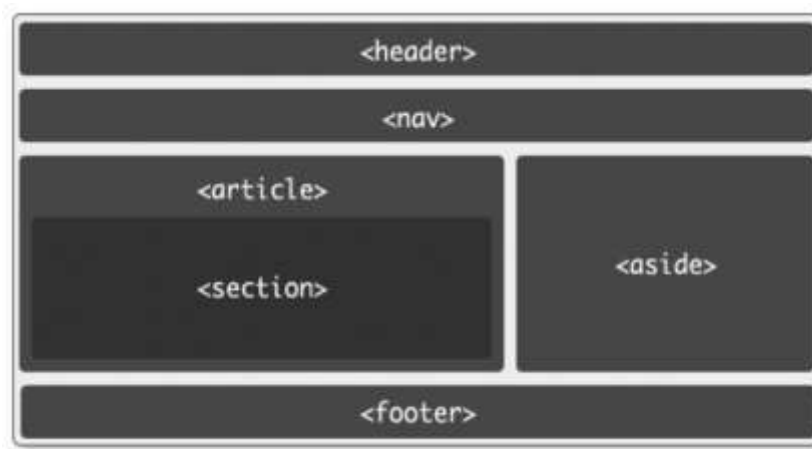
- Última versión: 8
- Fecha de lanzamiento: noviembre 2020
- Requisitos:
  - Analizador de PHP (módulo CGI)
  - Servidor web
  - Navegador web

### 2.5.3. HTML5

(Torrez, 2016, pag. 24) HTML5 es un lenguaje de etiquetas que permite diseñar documentos web estáticos; a diferencia de las versiones anteriores al HTML5, esta ofrece un conjunto de funciones que permitirá dar una nueva experiencia en el diseño web, veamos algunas de estas funciones:

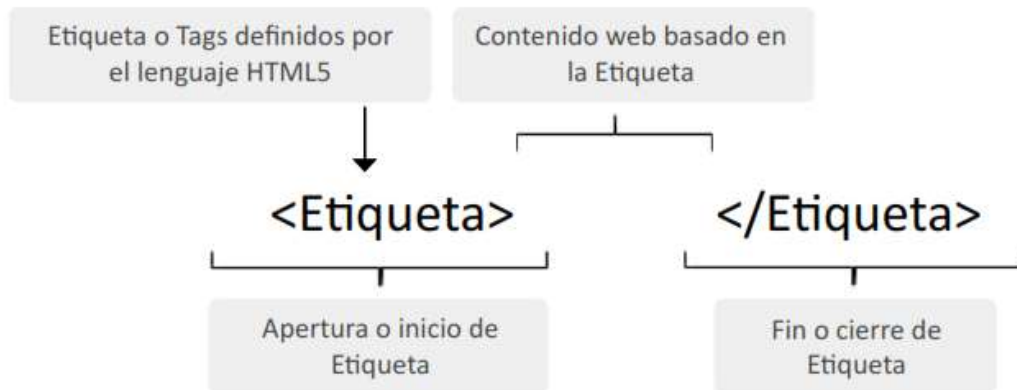
- **Semántica:** Mejora en la distribución de los elementos web, lo que da precisión al contenido.
- **Conectividad:** Permite comunicarse con un servidor web de manera ágil, sin consumir muchas capacidades.
- **Servicio local:** Permite navegar por un siti o web sin necesidad de estar en línea, también es llamado ejecución local desde el lado cliente.
- **Multimedia:** Permite asignar archivos de videos o música con una sola etiqueta de manera sencilla.

**Figura 2.25.** Estructura de HTML 5



**Fuente:** Torrez (2016)

**Figura 2.26.** Estructura de las etiquetas de HTML 5



**Fuente:** Torrez (2016)

### Atributos de una etiqueta HTML5

Se le denomina «atributos» a todas las opciones adicionales que puede tener una etiqueta; por ejemplo, asignar un estilo o definir un nombre a la etiqueta, etc.

**Figura 2.27.** Etiqueta

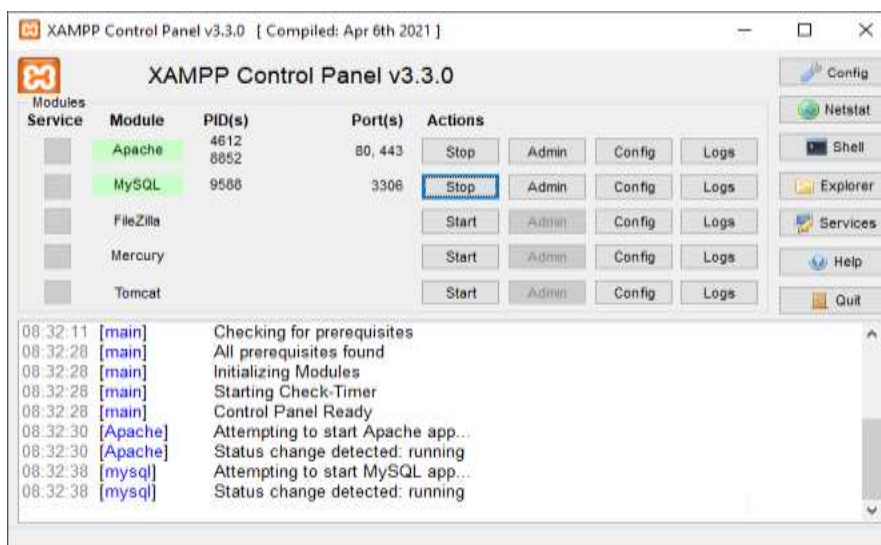


**Fuente:** Torrez (2016)

### 2.5.4. Xampp

Uno de los paquetes más usados para la implementación de aplicaciones con PHP, pues incorpora un conjunto de aplicaciones como el servidor Apache, el servidor de base de datos MySQL y el lenguaje de programación PHP. (Torrez, 2016, pag. 61)

**Figura 2.28.** Panel de control de Xampp



**Fuente:** Elaboración propia

## 2.5.5. PHPMYADMIN

phpMyAdmin es una herramienta de software gratuita escrita en PHP, destinada a manejar la administración de MySQL a través de la Web. phpMyAdmin admite una amplia gama de operaciones en MySQL y MariaDB. Las operaciones de uso frecuente (administración de bases de datos, tablas, columnas, relaciones, índices, usuarios, permisos, etc.) se pueden realizar a través de la interfaz de usuario, mientras aún tiene la capacidad de ejecutar directamente cualquier declaración SQL.

## 2.5.6. Visual Studio Code

Visual Studio Code es un editor de código redefinido y optimizado para crear y depurar aplicaciones web y en la nube modernas.

Los diferentes lenguajes de Visual Studio ofrecen distintos conjuntos de características y, en algunos casos, estas se comportan de forma diferente en función del lenguaje. Muchas de estas diferencias se especifican en las descripciones de las características, pero si quiere obtener más información, puede ver las secciones sobre lenguajes específicos de Visual Studio. (Microsoft, 2021).

### **2.5.7. Hoja de estilos CSS**

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. (Eguíluz, 2008, pag. 5)

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc. (Eguíluz, 2008, pag. 5)

### **2.5.8. AdminLTE**

Es el panel de administración de control de código abierto. Construido sobre Bootstrap, AdminLTE proporciona una gama de componentes receptivos, reutilizables y de uso común.

#### **Características:**

1. Tiene una documentación increíble.
2. Es adaptable a cualquier dispositivo o computador.
3. Está establecida en GitHub, por tanto, te permite hacer estudios o mejoras.
4. Incluye gráficos y status de todas las visitas de tu página web.

### **2.5.9. Bases de datos en la vida cotidiana**

En la actualidad, las bases de datos se usan tan ampliamente que se pueden encontrar en organizaciones de todos los tamaños, desde grandes corporaciones y agencias gubernamentales, hasta pequeños negocios e incluso en hogares. Las actividades diarias con frecuencia lo ponen en contacto con las bases de datos, ya sea directa o indirectamente. (Catherine M. Ricardo, 2009, pag. 2)

- Cuando visita un portal de Internet del consumidor que permite navegar y ordenar en línea bienes como libros o ropa, accede a una base de datos. La información acerca de los productos disponibles y los datos acerca del pedido se almacenan en una base de datos. También es posible que pueda ver los datos almacenados acerca de pedidos anteriores que haya levantado. Algunos sitios Web pueden usar información acerca de sus pedidos, o incluso sus actividades de navegación, para sugerir productos o servicios que es probable que le interesen.
- Cuando visita un sitio Web interactivo de servicio al cliente, como la página de inicio de una compañía de servicios o una aseguradora de salud, es capaz de acceder a información acerca de sus propios registros de servicios o productos proporcionados. Es posible que sea capaz de actualizar entradas en la base de datos con información personal como su dirección o número telefónico. Algunos sitios Web de servicios al cliente le permiten hacer cambios a los servicios a los que se suscribe. Por ejemplo, su proveedor de servicios telefónicos o compañía eléctrica pueden permitirle cambiar planes en línea.
- Si usa banca electrónica, puede recuperar registros de base de datos acerca de depósitos, retiros, pago de facturas y otras transacciones para sus cuentas. Puede transferir fondos, ordenar cheques y realizar muchas otras funciones, todas las cuales involucran el uso de una base de datos.
- Cuando usa una tarjeta de crédito, el vendedor por lo general espera la aprobación por computadora de su compra antes de presentarle un recibo para que lo firme. El proceso de aprobación consulta una base de datos para

verificar que su tarjeta no se perdió o la robaron y para encontrar su límite de crédito, saldo actual y cantidad de compras ya aprobadas. La base de datos se actualiza automáticamente para reflejar la nueva cantidad aprobada. Para una tarjeta de débito, se consulta la base de datos del banco para verificar su número de cuenta, su saldo actual y su saldo ajustado previo a la aprobación de la compra. La cantidad de compra se deduce en forma automática de su cuenta mientras la transacción se completa.

- Cuando compra bienes en un supermercado o tienda al menudeo, se usan escáneres para leer códigos universales de producto u otros identificadores de mercancía. Al usar el código escaneado, el sistema de base de datos puede identificar el artículo exacto y producir un recibo con el nombre del artículo y su precio, y toma en consideración cualquier precio de venta especial.
- Cuando hace planes para viajar, puede ingresar al sistema de reservaciones de una aerolínea en la que se usa una base de datos para rastrear los vuelos programados y las reservaciones de pasajeros. Dado que muchos viajeros pueden solicitar reservaciones de manera simultánea, el sistema debe ser capaz de manejar peticiones rápidamente, resolver conflictos y aceptar solicitudes hasta que se alcance el número máximo de asientos. Muchas cadenas hoteleras y compañías de renta de autos también tienen sistemas centralizados de reservaciones para aceptar reservaciones en cualquiera de sus ubicaciones, con el uso de un sistema de base de datos integrada.
- Si visita al médico, es posible que sus registros médicos y datos de facturación se conserven en una base de datos. Cuando le extienden una receta, probablemente el farmacéutico usará una base de datos para registrar información acerca de la prescripción, comprobar las interacciones con los medicamentos que use en la actualidad e imprimir la etiqueta y la receta. Tanto el médico como el farmacéutico pueden usar sus bases de datos para hacer cobranzas a terceras partes, que automáticamente verifican la cobertura y extienden las cobranzas del seguro para los gastos cubiertos, mientras que usted sólo paga el deducible. A todos los proveedores de salud

en Estados Unidos se les requiere proteger la privacidad durante estas transacciones, en concordancia con la legislación de privacidad de la Ley de Transportabilidad de Responsabilidad en Seguros de Salud (HIPAA, por sus siglas en inglés).

- Sus registros escolares tal vez se conservan en una base de datos que se actualiza cada periodo al registrar su inscripción, conclusión y calificación para cada clase.

### **2.5.10. Base de datos**

Una base de datos es un conjunto estructurado de datos que representa entidades y sus interrelaciones. La representación será única e integrada, a pesar de que debe permitir utilizaciones varias y simultáneas. (Camps, Casillas, Costal, Ginesta, Escofet, Mora, 2005, pag. 8)

Una base de datos es una colección organizada de información estructurada, o datos, típicamente almacenados electrónicamente en un sistema de computadora. Una base de datos es usualmente controlada por un sistema de gestión de base de datos (DBMS). En conjunto, los datos y el DBMS, junto con las aplicaciones que están asociados con ellos, se conocen como un sistema de base de datos, que a menudo se reducen a solo base de datos.

#### **2.5.10.1. 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. (Rafael Camps Paré, Luis Alberto Casillas Santillán, Dolors Costal Costa, Marc Gibert Ginestà, 2005, pag. 5)

### 2.5.10.2. MariaDB

MariaDB es una derivación (fork) de MySQL, que es desarrollada y mantenida por desarrolladores originales de MySQL developers organizados bajo la Fundación MariaDB. Se considera que es más abierto y está siendo distribuida como una Base de Datos compatible por defecto con MySQL por la mayoría de las distribuciones modernas de Linux.

**Figura 2.29.** MariaDB y MySQL



**Fuente:** Tec Guros 2021

### 2.5.10.3. Framework

(Vazquez, Nahuel, 2018, pag. 14) Podemos asociar el término framework a un subsistema y/o conjunto de librerías que proveen funcionalidades estándar a cualquier sistema; sin embargo, esta definición es incompleta, ya que además nos brinda:

- Una **estructura** de carpetas y archivos para organizar el código.
- Una **arquitectura** para desarrollar un proyecto.
- **Seguridad**, ya que los frameworks son actualizados frecuentemente para poder implementar medidas contra nuevas amenazas.
- **Robustez**, porque los frameworks son utilizados por muchos programadores en diversos proyectos: en consecuencia, cada framework está expuesto a un alcance mucho mayor al que podemos lograr escribiendo nuestro propio código.
- **Soporte**, ya que al ser utilizado por otros programadores, es muy fácil encontrar a alguien que haya tenido el mismo problema que podamos tener nosotros y que no logremos resolver.



- Un conjunto de **buenas prácticas** de programación para tener nuestro código lo más legible posible, de manera tal que podamos entender el código escrito por otro programador y, a la vez, hacer que el nuestro sea más entendible para los demás.

#### 2.5.10.4. Laravel

Laravel es un framework de código abierto para el desarrollo de aplicaciones web en PHP que posee una sintaxis simple, expresiva y elegante. Fue creado en 2011 por Taylor Otwell, inspirándose en Ruby on Rails y Symfony, de los cuales ha adoptado sus principales. (Documentación laravel)

Laravel es un marco de aplicación web con una sintaxis elegante y expresiva. Un marco web proporciona una estructura y un punto de partida para crear su aplicación, lo que le permite concentrarse en crear algo sorprendente mientras nos preocupamos por los detalles.

Laravel se esfuerza por proporcionar una experiencia de desarrollador increíble al tiempo que proporciona funciones poderosas como la inyección de dependencias exhaustiva, una capa de abstracción de base de datos expresiva, colas y trabajos programados, pruebas de integración y unidad, y más.

**Figura 2.30.** Logo de laravel



**Fuente:** Laravel LLC (2021)

#### 2.5.10.5. Composer

Composer es un administrador de dependencias para PHP (Similar a lo que npm es para JavaScript o pip para Python).

Es una aplicación PHP que ayuda a administrar las librerías desarrolladas por terceros que vas a incorporar a tu proyecto.

**Figura 2.31.** logo de Composer



**Fuente:** Nils Adermann y Jordi Boggiano (2021)

### **2.5.11. Patron MVC**

MVC es un patrón arquitectura, un modelo o guía que expresa cómo organizar y estructurar los componentes de un sistema software, sus responsabilidades y las relaciones existentes entre cada uno de ellos.

Su nombre, MVC, parte de las iniciales de Modelo-Vista-Controlador (Model-View-Controller, en inglés), que son las capas o grupos de componentes en los que organizaremos nuestras aplicaciones bajo este paradigma.

La arquitectura MVC propone, independientemente de las tecnologías o entornos en los que se base el sistema a desarrollar, la separación de los componentes de una aplicación en tres grupos (o capas) principales: el modelo, la vista, y el controlador, y describe cómo se relacionarán entre ellos para mantener una estructura organizada, limpia y con un acoplamiento mínimo entre las distintas capas. (Platzi)

## **2.6. CALIDAD DE SOFTWARE**

Hablar de calidad del software implica la necesidad de contar con parámetros que permitan establecer los niveles mínimos que un producto de este tipo debe alcanzar para que se considere de calidad. El problema es que la mayoría de las características que definen al software no se pueden cuantificar fácilmente; generalmente, se establecen de forma cualitativa, lo que dificulta su medición, ya que se requiere establecer métricas que permitan evaluar cuantitativamente cada característica dependiendo del tipo de software que se pretende calificar. (Antonela, Abud).

## **2.7. ISO/IEC 25010**

El modelo de calidad representa la piedra angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto. En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar las propiedades de un producto software determinado. (<https://iso25000.com/>)

La calidad del producto software se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios aportando de esta manera un valor. Son precisamente estos requisitos (funcionalidad, rendimiento, seguridad, mantenibilidad, etc.) los que se encuentran representados en el modelo de calidad, el cual categoriza la calidad del producto en características y subcaracterísticas.

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

1. Adecuación Funcional
2. Eficiencia de desempeño
3. Compatibilidad
4. Usabilidad
5. Fiabilidad
6. Seguridad

7. Mantenibilidad
8. Portabilidad

### 2.7.1. Adecuación Funcional

Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas. Esta característica se subdivide a su vez en las siguientes sub características:

- **Compleitud funcional.** Grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados.
- **Corrección funcional.** Capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido.
- **Pertinencia funcional.** Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

### 2.7.2. Eficiencia de desempeño

Esta característica representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes sub características:

- **Comportamiento temporal.** Los tiempos de respuesta y procesamiento y las ratios de throughput de un sistema cuando lleva a cabo sus funciones bajo condiciones determinadas en relación con un banco de pruebas (benchmark) establecido.
- **Utilización de recursos.** Las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas.
- **Capacidad.** Grado en que los límites máximos de un parámetro de un producto o sistema software cumplen con los requisitos.

### 2.7.3. Compatibilidad

Capacidad de dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno hardware o software. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- **Coexistencia.** Capacidad del producto para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes sin detrimento.
- **Interoperabilidad.** Capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

### 2.7.4. Usabilidad

Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- **Capacidad para reconocer su adecuación.** Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
- **Capacidad de aprendizaje.** Capacidad del producto que permite al usuario aprender su aplicación.
- **Capacidad para ser usado.** Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
- **Protección contra errores de usuario.** Capacidad del sistema para proteger a los usuarios de hacer errores.
- **Estética de la interfaz de usuario.** Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.
- **Accesibilidad.** Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

### 2.7.5. Fiabilidad

Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- **Madurez.** Capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.
- **Disponibilidad.** Capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.
- Tolerancia a fallos. Capacidad del sistema o componente para operar según lo previsto en presencia de fallos hardware o software.
- **Capacidad de recuperación.** Capacidad del producto software para recuperar los datos directamente afectados y reestablecer el estado deseado del sistema en caso de interrupción o fallo.

### 2.7.6. Seguridad

Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos. Esta característica se subdivide a su vez en las siguientes sub características:

- **Confidencialidad.** Capacidad de protección contra el acceso de datos e información no autorizados, ya sea accidental o deliberadamente.
- **Integridad.** Capacidad del sistema o componente para prevenir accesos o modificaciones no autorizados a datos o programas de ordenador.
- **No repudio.** Capacidad de demostrar las acciones o eventos que han tenido lugar, de manera que dichas acciones o eventos no puedan ser repudiados posteriormente.
- **Responsabilidad.** Capacidad de rastrear de forma inequívoca las acciones de una entidad.

- **Autenticidad.** Capacidad de demostrar la identidad de un sujeto o un recurso.

### 2.7.7. Mantenibilidad

Esta característica representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas. Esta característica se subdivide a su vez en las siguientes sub características:

- **Modularidad.** Capacidad de un sistema o programa de ordenador (compuesto de componentes discretos) que permite que un cambio en un componente tenga un impacto mínimo en los demás.
- **Reusabilidad.** Capacidad de un activo que permite que sea utilizado en más de un sistema software o en la construcción de otros activos.
- **Analizabilidad.** Facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos en el software, o identificar las partes a modificar.
- **Capacidad para ser modificado.** Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.
- **Capacidad para ser probado.** Facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.

### 2.7.8. Portabilidad

Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro. Esta característica se subdivide a su vez en las siguientes sub características:

- **Adaptabilidad.** Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso.

- **Capacidad para ser instalado.** Facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.
- **Capacidad para ser reemplazado.** Capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno.

## 2.8. COCOMO II

Permite realizar estimaciones en función del tamaño del software, y de un conjunto de actores de costo y de escala. (Gómez, López, Migani y Otazú).

### 2.8.1. Estimación de esfuerzo

Posee tres modelos: Composición de Aplicación, Diseño Temprano y Post-Arquitectura.

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

#### 2.8.1.1. Modelo Composición de Aplicación

La fórmula propuesta en este modelo es la siguiente

$$PM = NOP / PROD$$

Donde:

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

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

**OP** (Puntos Objeto): Tamaño del software a desarrollar expresado en Puntos Objeto

**%reuso** Porcentaje de reuso que se espera lograr en el proyecto.



**PROD:** Es la productividad promedio determinada a partir del análisis de datos de proyectos en [Banker 1994], mostrada en la siguiente tabla:

**Tabla 2.1.** Productividad para el modelo Composición de Aplicación

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

Fuente: Boehm (1995/2)

### 2.8.1.2. Modelo de diseño temprano

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

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

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

$$PM_{nominal} = AX(KSLOC)^B$$

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

Donde:

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

Clasificados en categorías, los 7 Multiplicadores de Esfuerzo son:

Del Producto

RCPX: Confiabilidad y Complejidad del producto

RUSE: Reusabilidad Requerida

De la Plataforma

PDIF: Dificultad de la Plataforma

Del Personal

PERS: Aptitud del Personal

PREX: Experiencia del Personal

Del Proyecto

FCIL: Facilidades

SCED: Cronograma de Desarrollo Requerido

### **2.8.1.3. Modelo Post-Arquitectura**

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

El esfuerzo nominal se ajusta usando 17 factores multiplicadores de esfuerzo. El mayor número de multiplicadores permite analizar con más exactitud el conocimiento disponible en las últimas etapas de desarrollo, ajustando el modelo de tal forma que refleje fielmente el producto de software bajo desarrollo. La fórmula para el cálculo del esfuerzo es la siguiente:

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

### **2.8.2. Estimación del Cronograma**

La versión inicial de COCOMO II provee un modelo de estimación del cronograma similar al presentado en COCOMO' 81 y ADA COCOMO. La ecuación inicial para los tres modelos de COCOMO II es:

$$TVEV = \left[ 3.0 \times PM^{(0.33)+0.2x(B-1.01)} \right] \times \frac{SCED\%}{100}$$

Donde:

*TDEV* es el tiempo calendario en meses que transcurre desde la determinación de los requerimientos a la culminación de una actividad que certifique que el producto cumple con las especificaciones.

*PM* \* es el esfuerzo expresado en meses personas, calculado sin tener en cuenta el multiplicador de esfuerzo SCED.

*B* es el Factor de Escala

*SCED*% es el porcentaje de compresión/expansión del cronograma.

### **2.8.3. Métricas de Software**

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

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

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

## **2.9. SEGURIDAD**

Cualquier organización que expone sus servicios informáticos a redes de acceso tendrán que realizar un esfuerzo significativo para asegurar que la información y recursos están protegidos. Internet es un factor primordial en la comunicación y también un evidente riesgo potencial de acceso y mal uso de los servicios e información disponibles. Obviamente, se catalogan sistemas más críticos que otros donde su seguridad debe de ser muy significativa, pero en general todas las aplicaciones Web deben de estar protegidas y aseguradas ante los principales ataques.

En una aplicación web, dividimos la seguridad en:

**Disponibilidad:** Propiedad o característica de los activos consistentes en que las entidades o procesos autorizados tienen acceso a los mismos cuando lo requieren.

**Autenticidad:** Propiedad o característica consistente en que una entidad es quien dice ser o bien que garantiza la fuente de las que producen los datos.

**Integridad:** Propiedad o característica consistente en que el activo de información no ha sido alterado de manera no autorizada.

**Confidencialidad:** Propiedad o característica consistente en que las actuaciones de una entidad pueden ser imputadas exclusivamente a dicha entidad.

## 2.10. Cifrador AES

Se trata de un algoritmo de clave simétrica y que cifra por bloques. El tamaño de bloque es de 128 bits siempre mientras que el tamaño de la clave puede variar entre los 128, 192 y 256 bits. Sin embargo, si aumenta el tamaño de la clave, también lo hace el número de rondas necesarias para cifrar. (Martínez, pag, 45).

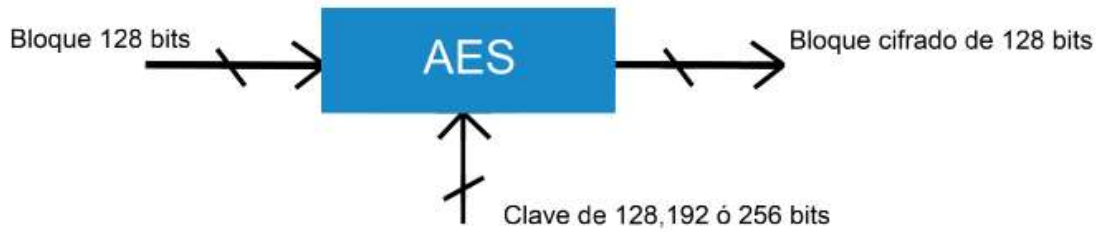
**Tabla 2.2.-** Número de rondas

Tamaño de la clave	Número de rondas
128	10
192	12
256	14

**Fuente:** Martínez (2016)

AES es el cifrador de clave simétrica más importante, la NSA (National Security Agency) permite encriptar información clasificada como TOP SECRET usando AES con claves de 192 o 256 bits.

**Figura 2.31. Bloque cifrado**



**Fuente:** Martínez (2016)

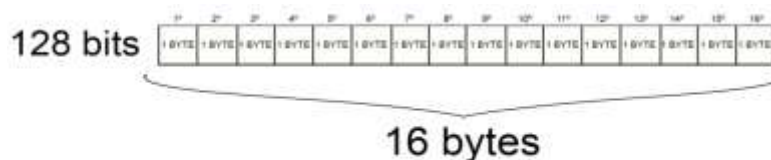
Para entender el funcionamiento de AES es necesario tener claro que AES cifra por bloques de 128 bits, esto quiere decir que se cogerán los primeros 128 bits del texto a cifrar, se cifrarán y una vez cifrados se cifrarán los siguientes 128 bits y ese proceso se repetirá hasta que se cifre el mensaje completo.

### 2.10.1.1. Matriz de Estado

En AES y dentro de cada ronda la entrada es un bloque de 128 bits y la salida también, a estos 128 bits que van cambiando a lo largo de la ronda y de todo el proceso se les llama estado.

La matriz de estado contiene los 16 bytes puestos por columnas.[2]

**Figura 2.31. Matriz de huella**



**Matriz de estado**

1 <sup>a</sup>	5 <sup>a</sup>	9 <sup>a</sup>	13 <sup>a</sup>
2 <sup>a</sup>	6 <sup>a</sup>	10 <sup>a</sup>	14 <sup>a</sup>
3 <sup>a</sup>	7 <sup>a</sup>	11 <sup>a</sup>	15 <sup>a</sup>
4 <sup>a</sup>	8 <sup>a</sup>	12 <sup>a</sup>	16 <sup>a</sup>

**Fuente:** Martínez (2016)

Al igual que los estados, la clave y las subclaves pueden representarse como matrices y trabajar de esa forma con ellas.

## **CAPÍTULO III**

---

### **MARCO APLICATIVO**

## CAPITULO III

### 3. MARCO APLICATIVO

#### 3.1. Recopilación de Información

En este capítulo se hará uso de la metodología Scrum, la cual se apoya en la notación UML para modelar el software. El objetivo fundamental es determinar el análisis, diseño y desarrollo del sistema “CUADERNO PEDAGÓGICO WEB COMO HERRAMIENTA DEL MAESTRO PARA EL SEGUIMIENTO DE ACTIVIDADES PEDAGÓGICAS” en el cual se desarrollará las siguientes actividades:

- Planificación del proyecto de Software orientado a la Web.
- Recolección de información pertinente
- Establecer tiempos, tecnologías, herramientas de desarrollo y la definición de numero de Sprints.
- Realizar las correspondientes pruebas de funcionalidad del sistema.

Para esta primera fase, se asignaron días con el fin de obtener los requerimientos necesarios del sistema web.

##### 3.1.1. Concepción

En este apartado se cumplió diferentes actividades para la obtención de los requisitos de software, con el objetivo de obtener información pertinente acerca del manejo del Cuaderno Pedagógico del maestro, que a diario registran actividades en la unidad educativa.

- **Observación:** Se pudo evidenciar que cada inicio de gestión educativa el personal administrativo realiza inscripción de estudiantes nuevos y antiguos, además de asignar cursos a los maestros.
- **Entrevistas personales:** Se realizaron frecuentes entrevistas con el personal docente de la Unidad Educativa, para socializar acerca del Cuaderno Pedagógico.



- **Documentación:** Mediante la coordinación con el director de la Unidad Educativa fue posible obtener copias de la documentación como ser RUDE (Registro Único de Estudiante), RDA (Registro Docente Administrativo) copia de boletín de calificaciones y Cuaderno pedagógico.

Esta información será útil, porque existe un estándar en el almacenamiento de la información al sistema Web.

### 3.1.2. Indagación y elaboración

Una vez concluido las tareas para la obtención de requerimientos, se procederá a realizar el respectivo análisis de los requerimientos que se encontraron:

- Registrar datos Relevantes de los Estudiantes.
- Registrar datos Relevantes de los Maestros.
- Registrar datos de Asistencia de los Estudiantes.
- Registrar calificaciones diarias de los Estudiantes.
- Generar Reporte de Calificaciones de forma Trimestral.

A continuación, se realizará la clasificación de requerimientos funcionales (Product Backlog) que se muestran en las siguientes tablas:

**Tabla 3.1.** Requerimientos funcionales del Software

<b>Id.</b>	<b>Requerimientos funcionales</b>
<b>1</b>	Análisis, diseño y creación de la Base de datos del sistema.
<b>2</b>	Registro de docentes y estudiantes
<b>3</b>	Inscripción de estudiantes nuevos
<b>4</b>	Registro de Campo de saberes y conocimientos
<b>5</b>	Asignar cursos y paralelo
<b>6</b>	Asignar maestros
<b>7</b>	Generar Reporte

**Fuente:** Elaboración propia

### **3.1.3. Negociación**

Es normal que los clientes y usuarios pidan más de lo que puede lograrse dado lo limitado de los recursos del negocio (Pressman, 2010).

Por ello se aclararon ciertos requerimientos con los usuarios (director, Maestros y personal Administrativo de la Unidad Educativa), llegando en los siguientes acuerdos:

- El sistema web será capaz de almacenar datos relevantes de los estudiantes y maestros.
- Para el ingreso al sistema se proporcionará usuario y contraseña a los Administradores, Maestros y Estudiantes.
- Se hará uso del gestor de base de datos MySQL para el almacenamiento de la información, servidor web y un navegador de internet para el acceso al sistema.

### **3.2. Análisis de la Situación Actual**

De acuerdo al diagnóstico realizado empleando la técnica de la observación directa, entrevistas y la investigación documental, la Unidad Educativa Republica del Japón B, existe varias necesidades una de ellas es la falencia en el registro de asistencias y calificaciones de las actividades pedagógicas diarias que realizan los maestros.

### **3.3. PRE GAME (antes del juego)**

#### **3.3.1. Planificación**

El presente proyecto de grado se construye en una solución satisfactoria para innovar, mejorar y optimizar el registro de asistencia y actividades cotidianas del maestro. Para esta tarea de planeación se trabajó con los actores más importantes de la Unidad Educativa.

**Figura 3.1:** Planeación del Desarrollo del Proyecto.



**Fuente:** Elaboración propia (2021)

El proyecto Web constituye en una herramienta web para el control y seguimiento de las asistencias y actividades diarias que realiza el maestro, para ello se detallan los siguientes procesos:

- El sistema contará con tres roles de usuarios (administrador, maestros y estudiantes), con la cual podrá ingresar al sistema de acuerdo a los roles asignados.
- Los formularios de registro de estudiantes deben ser realizados con una interface amigable para el administrador.
- Los datos de entrada del registro de asistencia serán de fácil interacción para el maestro.

### **3.3.2. Análisis de Requerimiento (Product Backlog)**

En esta fase se tomará en cuenta en términos generales, las herramientas tecnológicas que se utilizarán en el proyecto web.

***Las herramientas de desarrollo son las siguientes:***

- Lenguaje de Programación: PHP 7, JavaScript, Ajax, JQuery.
- MySQL como gestor de Base de Datos.

- Navegador como Google Chrome o Mozilla Firefox
- Sistema operativo Windows

### 3.3.3. Identificación de Usuarios

- **Administrador:** Es el usuario que tiene el privilegio de registrar datos de los estudiantes, maestros, registrar áreas / campos de conocimientos y saberes, inscripciones y asignar cursos a los maestros.
- **Usuario Maestro:** Es el usuario que tiene el privilegio de registrar las asistencias y notas de los cursos y paralelos asignados.
- **Usuario Estudiantes:** Es el usuario que tiene el privilegio de realizar el seguimiento de todas las actividades registradas.

### 3.3.4. Identificación de Roles de los Actores

A continuación, se describen a los actores importantes y roles que desempeñaran en la interacción con el sistema Web.

**Tabla 3.2.** Roles de Actores

ROL	TAREA	NOMBRE
PRODUCT OWNER	Responsable de PRODUCT OWNER delinear el Producto	Lic. Nilo Manuel Villanueva Quispe
SCRUM MASTER	Líder del Equipo	Univ. Alfredo Callisaya Huanca
DEVELOPMENT TEAM (Equipo de desarrollo)	<ul style="list-style-type: none"> <li>✓ Análisis</li> <li>✓ Diseño</li> <li>✓ Desarrollo</li> <li>✓ Pruebas</li> </ul>	Univ. Alfredo Callisaya Huanca
CUSTOMERS	Beneficiarios en hacer uso del Sistema	Maestros y estudiantes que realizaran el uso del sistema

**Fuente:** Elaboración propia

### 3.3.5. Requerimientos de Usuario

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

**Tabla 3.3.** Historia de usuario – 1

<b>Número:</b> 1	<b>Historia de Usuario:</b> Cliente
<b>Nombre Historia de Usuario:</b> Diseño de Base de datos	
<b>Prioridad:</b> Alta	<b>Riesgo en desarrollo:</b> Bajo
<b>Tiempo estimado:</b> 1 semana	<b>Iteración:</b> 1
<b>Programador responsable:</b> Alfredo Callisaya Huanca	
<b>Historia:</b> La base de datos debe almacenar toda la información pertinente y debe ser de fácil acceso a las consultas.	
<b>Objetivo:</b> Diseñar una base de datos que facilite el acceso a la información.	

**Fuente:** Elaboración propia (2021)

**Tabla 3.4.** Historia de usuario - 2

<b>Número:</b> 2	<b>Historia de Usuario:</b> Cliente
<b>Nombre Historia de Usuario:</b> Administración de usuarios	
<b>Prioridad:</b> Media	<b>Riesgo en desarrollo:</b> Bajo
<b>Tiempo estimado:</b> 1 semana	<b>Iteración:</b> 2
<b>Programador responsable:</b> Alfredo Callisaya Huanca	
<b>Historia:</b> El registro de usuarios es muy importante para asignar roles y permisos correspondiente para el acceso al sistema web.	
<b>Objetivo:</b> Desarrollar un módulo de administración de usuarios donde registre, actualice o elimine a los usuarios.	

**Fuente:** Elaboración propia (2021)

**Tabla 3.5.** Historia de usuario - 3

<b>Número:</b> 3	<b>Historia de Usuario:</b> Cliente
<b>Nombre Historia de Usuario:</b> Registro de Asistencia	
<b>Prioridad:</b> Alta	<b>Riesgo en desarrollo:</b> Bajo
<b>Tiempo estimado:</b> 1 semana	<b>Iteración:</b> 3
<b>Programador responsable:</b> Alfredo Callisaya Huanca	
<b>Historia:</b> De acuerdo a las entrevistas con el personal administrativo se enfatizó sobre la implementación del Cuaderno Pedagógico Web como herramienta del maestro ya que cada maestro registra de forma manual.	
<b>Objetivo:</b> Implementar un módulo para la gestión de asistencias diarias.	

**Fuente:** Elaboración propia (2021)

**Tabla 3.6.** Historia de usuario - 4

<b>Número:</b> 4	<b>Historia de Usuario:</b> Cliente
<b>Nombre Historia de Usuario:</b> Registro de la Heteroevaluación	
<b>Prioridad:</b> Alta	<b>Riesgo en desarrollo:</b> Bajo
<b>Tiempo estimado:</b> 1 semana	<b>Iteración:</b> 4
<b>Programador responsable:</b> Alfredo Callisaya Huanca	
<b>Historia:</b> De acuerdo a las entrevistas con los maestros de la Unidad Educativa, indicaron que se realiza la calificación de acuerdo a las dimensiones del Ser, saber, hacer y decidir.	
<b>Objetivo:</b> Implementar un módulo para la gestión de tareas cotidianas del maestro.	

**Fuente:** Elaboración propia (2021)

**Tabla 3.7.** Historia de usuario - 5

<b>Número:</b> 5	<b>Historia de Usuario:</b> Cliente
<b>Nombre Historia de Usuario:</b> Registro de la Heteroevaluación	
<b>Prioridad:</b> Media	<b>Riesgo en desarrollo:</b> Bajo
<b>Tiempo estimado:</b> 1 semana	<b>Iteración:</b> 5
<b>Programador responsable:</b> Alfredo Callisaya Huanca	
<b>Historia:</b> De acuerdo a las entrevistas con los estudiantes hicieron referencia al registro manual del maestro, indicando que no se puede realizar el seguimiento de sus calificaciones cotidianas como ser: trabajos, exposiciones, cuadernos, etc.	
<b>Objetivo:</b> Implementar un módulo del estudiante para el seguimiento de las calificaciones.	

**Fuente:** Elaboración propia (2021)

**Tabla 3.8.** Historia de usuario - 6

<b>Número:</b> 6	<b>Historia de Usuario:</b> Cliente
<b>Nombre Historia de Usuario:</b> Reportes	
<b>Prioridad:</b> Media	<b>Riesgo en desarrollo:</b> Bajo
<b>Tiempo estimado:</b> 1 semana	<b>Iteración:</b> 6
<b>Programador responsable:</b> Alfredo Callisaya Huanca	
<b>Historia:</b> Los reportes trimestrales son indispensables para la Unidad Educativa, ya que se maneja de forma impresa dificultando en la centralización.	
<b>Objetivo:</b> Implementar un módulo de reportes, donde se pueda visualizar de forma trimestral.	

**Fuente:** Elaboración propia (2021)

**Tabla 3.9.** Requerimiento de usuario

<b>ID</b>	<b>Descripción</b>	<b>Módulo</b>	<b>Prioridad</b>	<b>Estado</b>
R1	Diseñar un base de datos para el desarrollo de procesos.	Bases de datos	Alta	Terminado
R2	Desarrollar la interface del administrador que permita el registro, modificación y eliminación de estudiantes y maestros	Administración	Alta	Terminado
R3	Desarrollar la interface del Docente que permita el registro, eliminación y modificación de las asistencias y notas.	Maestros	Alta	Terminado
R4	Desarrollar una interface para mostrar reportes de asistencia y notas trimestrales	Maestros	Alta	Terminado
R5	Desarrollar una interface para la inscripción de estudiantes	Administración	Alta	Terminado
R6	Desarrollar la interface del estudiante que permita el seguimiento cotidiano	Estudiantes	Media	Terminado
R7	Elaborar manual de usuario para Maestros y Estudiantes	Comunicación	Baja	Terminado
R8	Realizar un taller de Capacitación a los maestros	Comunicación	Baja	Terminado
R9	Alojar la base de datos en un hosting con un dominio personalizado	Administración	Alta	Terminado
R10	Realizar videos explicativos para su manejo adecuado	Comunicación	Baja	

**Fuente:** Elaboración propia (2021)



### 3.3.6. Análisis de Riesgo

Es importante tomar en cuenta dos tipos de riesgos:

- Primero riesgo del proyecto.
- Segundo riesgo del producto.

En la siguiente tabla se describen todos los puntos analizados.

**Tabla 3.10.** Requerimiento de usuario

Riesgo	Tipo	Descripción	Probabilidad	Estado	Estrategia
Cuando no se cumplen con las fechas establecidas	Proyecto	Es probable que las fechas descritas en el cronograma no se cumplan en su totalidad	Alta	Tolerable	Ajustar las fechas establecidas
Cambio de los requerimientos del cliente	Producto	Existe riesgo cuando hay cambios en los módulos	Alta	Tolerable	Revisión constante a los requerimientos
Ausencia de infraestructura para la implementación del sistema	Proyecto	Es posible que la institución no cuente con herramientas para su implementación	Moderado	Tolerable	Solicitar con anticipación o contar con equipo computacional.

**Fuente:** Elaboración propia (2021)

### 3.4. GAME (Juego)

#### 3.4.1. Iteraciones

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

##### 3.4.1.1. Primera Iteración

Durante la primera iteración se desarrollaron los elementos necesarios para el desarrollo del sistema Cuaderno Pedagógico Web. Las tareas realizadas durante esta iteración se observan en la tabla que constituye el Blacklog del Sprint.

**Tabla 3.11.** Primera Iteración

ID	TAREA	SPRINT	INICIO	DURACIÓN
		1	2-08-2021	15 días
		TIPO	DÍAS DE TRABAJO	ESTADO
1.1	Realizar la planificación de la iteración.	Planificación	2	Terminado
1.2	Analizar los requerimientos del Blacklog del producto.	Planificación	2	Terminado
1.3	Analizar los requerimientos de la iteración de casos de uso.	Diseño	2	Terminado
1.4	Construir diagrama de actividades.	Diseño	1	Terminado
1.5	Construir diagrama de clases.	Diseño	1	Terminado
1.6	Construir diagrama físico	Diseño	1	Terminado
1.7	Construir diagrama de presentación.	Diseño	1	Terminado
1.8	Diseñar y construir la base de datos del sistema.	Desarrollo	2	Terminado
1.9	Desarrollar el módulo de administrador.	Desarrollo	3	Terminado

**Fuente:** Elaboración propia (2021)

Funcionalidades correspondientes al incremento de la iteración son:

- Base de datos independientes del sistema.
- Página de ingreso con control de acceso a los usuarios.
- Módulo de administración del sistema.

### 3.4.1.2. Segunda Iteración

En la segunda iteración se desarrollan los módulos del administrador registro e inscripciones.

**Tabla 3.12.** Segunda Iteración

ID	TAREA	SPRINT	INICIO	DURACIÓN
		1	16-8-2021	30 días
		TIPO	DÍAS DE TRABAJO	ESTADO
2.1	Realizar la planificación de la iteración.	Planificación	3	Terminado
2.2	Analizar los requerimientos del Blacklog del producto.	Planificación	2	Terminado
2.3	Análisis de riesgos los requerimientos de la iteración de casos de uso.	Diseño	2	Terminado
2.4	Complementar diagrama de actividades.	Diseño	2	Terminado
2.5	Complementar diagrama físico	Diseño	4	Terminado
2.6	Construir diagrama de Diseño presentación.	Diseño	3	Terminado
2.7	Construir diagrama de clases.	Diseño	3	Terminado
2.8	Construir diagrama físico.	Diseño	2	Terminado
2.9	Desarrollar registro y módulo de inscripciones.	Desarrollo	4	Terminado
2.10	Probar los módulos elaborados en la iteración con los actores.	Revisión	3	Terminado

**Fuente:** Elaboración propia (2021)

En la segunda iteración se desarrollaron las siguientes funcionalidades:

- Módulo de usuarios manejo de roles
- Módulo de administración, registro de estudiantes y maestros
- Módulo de inscripciones y asignación de maestros.

### 3.4.1.3. Tercera Iteración

En la tercera iteración se desarrollan los módulos del maestro.

**Tabla 3.13.** Tercera Iteración

ID	TAREA	SPRINT	INICIO	DURACIÓN
		1	1-09-2021	30 días
		TIPO	DÍAS DE TRABAJO	ESTADO
2.1	Realizar la planificación de la iteración.	Planificación	3	Terminado
2.2	Analizar los requerimientos del Blacklog del producto.	Planificación	2	Terminado
2.3	Análisis de riesgos los requerimientos de la iteración de casos de uso.	Diseño	2	Terminado
2.4	Complementar diagrama de actividades.	Desarrollo	2	Terminado
2.5	Complementar diagrama de clases.	Desarrollo	2	Terminado
2.6	Complementar diagrama físico	Desarrollo	4	Terminado
2.7	Construir diagrama de Diseño presentación	Desarrollo	3	Terminado
2.8	Desarrollar el módulo de asistencia	Desarrollo	2	Terminado
2.9	Desarrollar el módulo de notas	Desarrollo	4	Terminado
2.10	Probar los módulos elaborados en la iteración con los actores.	Revisión	3	Terminado

**Fuente:** Elaboración propia (2021)

En la tercera iteración se desarrollaron las siguientes funcionalidades:

- Módulo gestión y administración de asistencias diarias.
- Módulo de gestión de calificaciones cotidianas del maestro.

### 3.4.1.4. Cuarta Iteración

Finalmente, durante la cuarta iteración se desarrollaron los requerimientos finales de sistema web, como la generación de reportes, pruebas del sistema y alojamiento en un hosting.

**Tabla 3.14.** Cuarta Iteración

ID	TAREA	SPRINT	INICIO	DURACIÓN
		4	4-10-2021	30 días
		TIPO	DÍAS DE TRABAJO	ESTADO
2.1	Realizar la planificación de la iteración.	Planificación	3	Terminado
2.2	Analizar los requerimientos del Blacklog del producto.	Planificación	2	Terminado
2.3	Análisis de riesgos los requerimientos de la iteración de casos de uso.	Diseño	2	Terminado
2.4	Complementar diagrama de actividades.	Desarrollo	2	Terminado
2.5	Complementar diagrama de clases.	Desarrollo	2	Terminado
2.6	Complementar diagrama físico	Desarrollo	4	Terminado
2.7	Construir diagrama de Diseño presentación	Desarrollo	3	Terminado
2.8	Desarrollar módulo para el estudiante.	Desarrollo	2	Terminado
2.9	Implementación del módulo de reportes.	Desarrollo	4	Terminado
2.10	Probar los módulos elaborados en la iteración con los actores.	Revisión	3	Terminado

**Fuente:** Elaboración propia (2021)

En la cuarta iteración se desarrollaron las siguientes funcionalidades:

- Módulo de reportes y pruebas del sistema en la Web.

### 3.4.2. Modelado del Sistema

#### 3.4.2.1. Diagrama de caso de uso

Es un modelo del sistema que contiene actores, casos de uso y sus respectivas relaciones.

En la siguiente tabla se describen las características de los actores principales identificados en el manejo e implementación del sistema.

**Tabla 3.15.** Descripción de actores

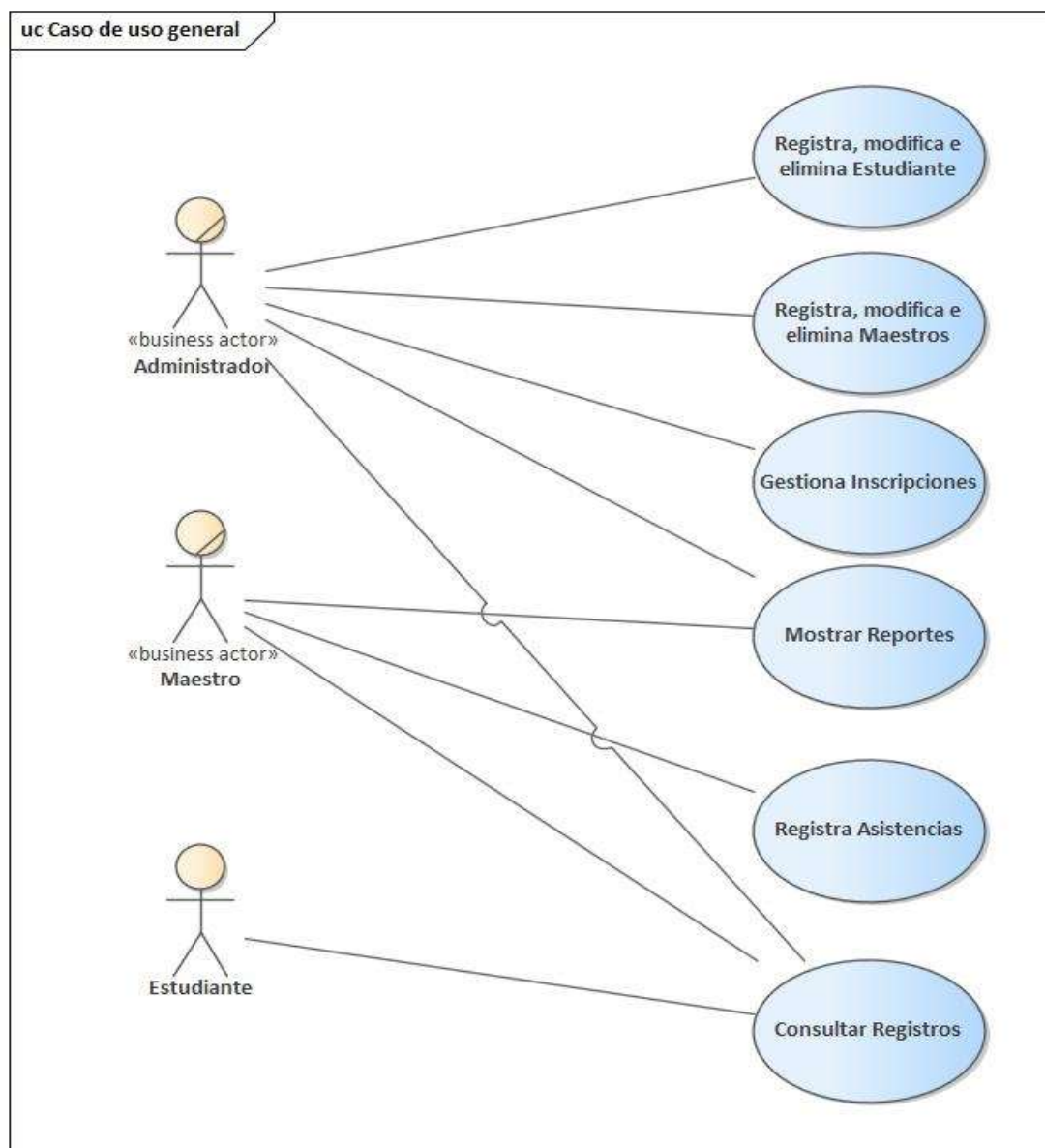
ACTORES	DESCRIPCIÓN
Administrador	Es el personal administrativo de la unidad educativa que tendrá todos los privilegios sobre el manejo y administración del sistema web.
Maestros	Son los Maestros de la Unidad Educativa que son los actores principales en el registro de las actividades pedagógicas.
Estudiantes	Es la comunidad estudiantil de la Unidad Educativa.

**Fuente:** Elaboración propia (2021)

### 3.4.2.2. Diagrama de caso de uso General

Este modelo presenta mediante diagramas de Casos de uso el cuál serán las principales funcionalidades que el sistema web debe permitir llevar a cabo, mientras que los actores representan los roles o papeles que actúan recíprocamente con los procesos.

**Figura 3.2.** Caso de uso General

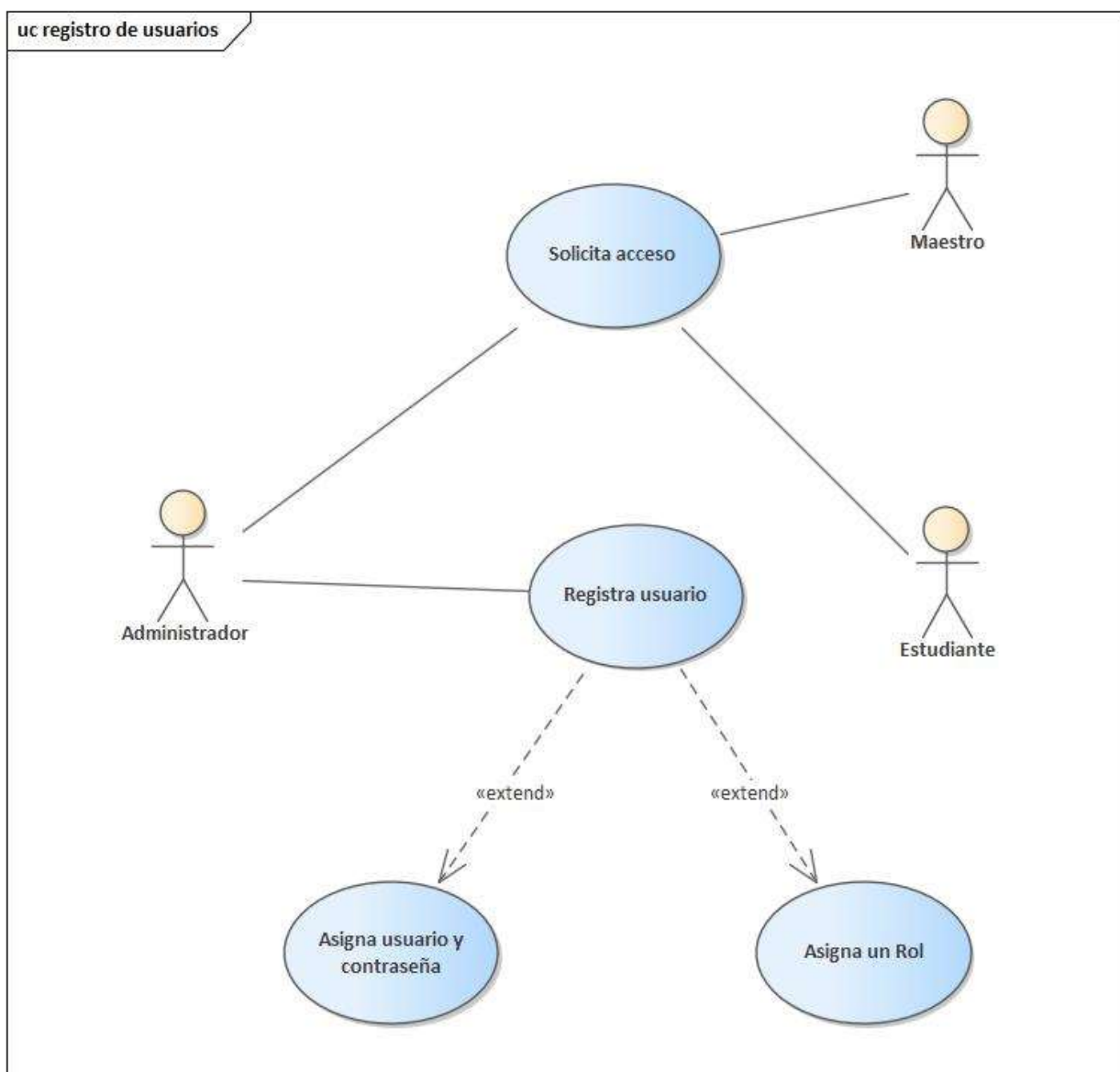


**Fuente:** Elaboración propia

### 3.4.2.3. Diagrama de caso de uso: Registro de usuario

Para el registro de usuarios al sistema web el actor principal es el administrador (director de la Unidad Educativa), se encarga de llenar un formulario de datos del nuevo usuario, dar un rol, habilitar o en algún caso eliminar al usuario como se muestra en la siguiente figura.

**Figura 3.3.** Diagrama de casos de uso – Registro de usuario en el sistema



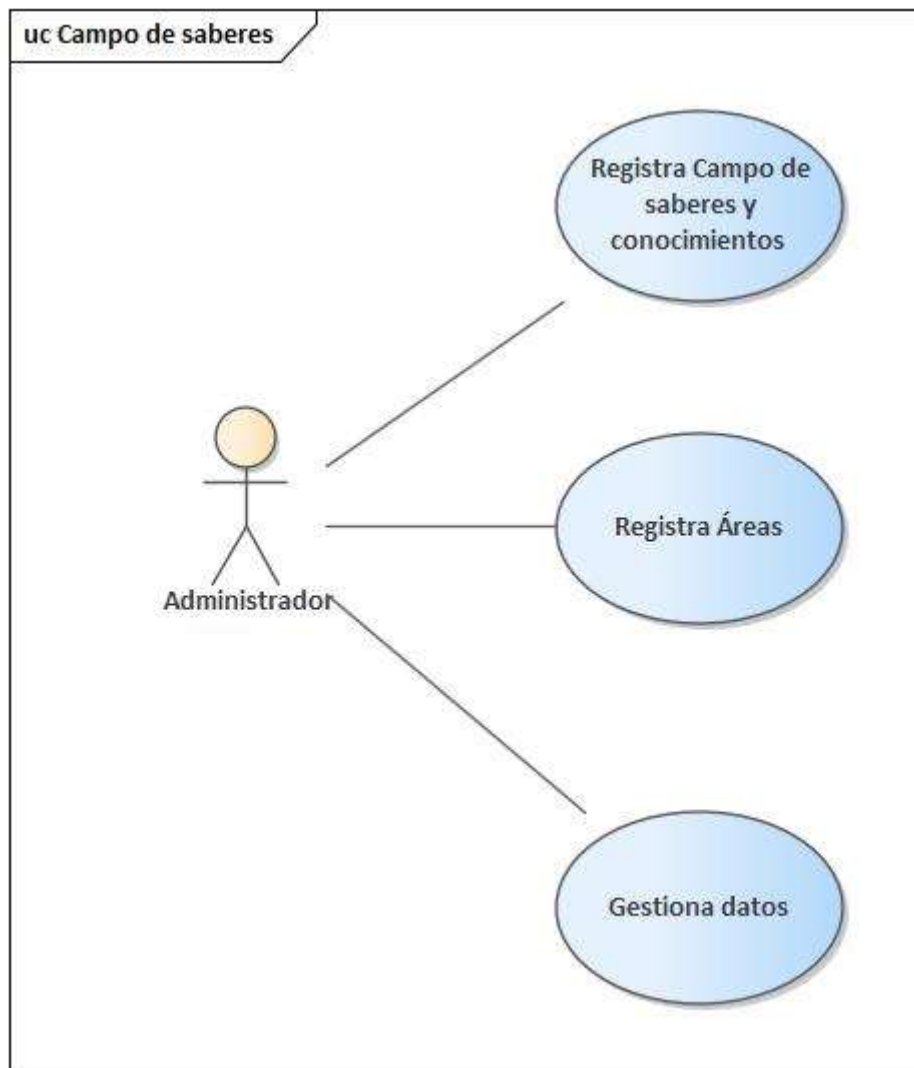
**Fuente:** Elaboración propia



### 3.4.2.4. Diagrama de caso de uso: Registro de Áreas y Campos

Para el registro de áreas, campo de saberes y conocimientos solo los actores principales pueden realizar esa acción.

**Figura 3.4.** Diagrama de casos de uso – Registro de Áreas y Campos.

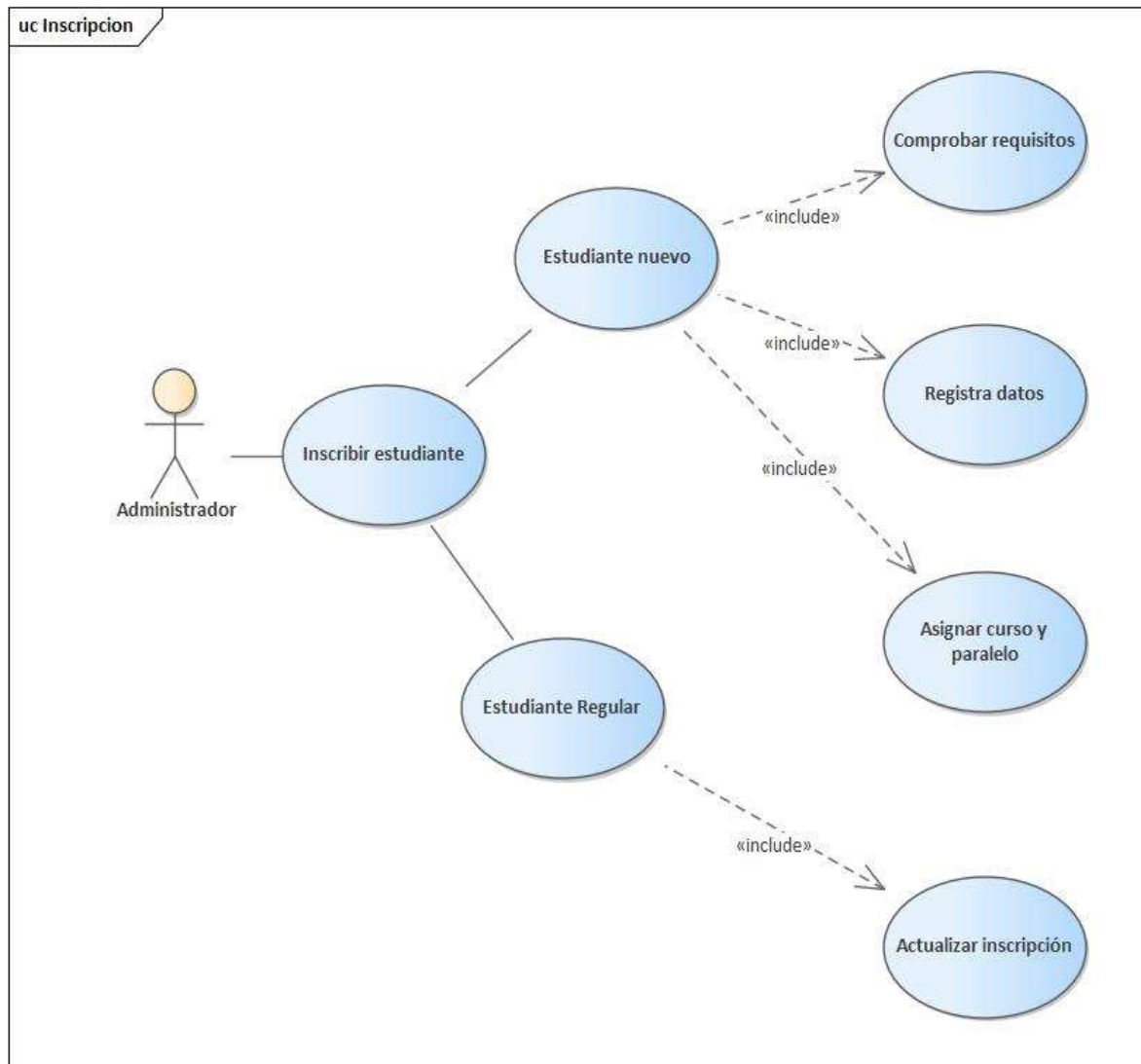


**Fuente:** Elaboración propia

### 3.4.2.5. Diagrama de caso de uso: Inscripción

Para la inscripción de estudiantes solo los actores principales pueden realizar esa acción.

**Figura 3.5.** Diagrama de casos de uso – Registro

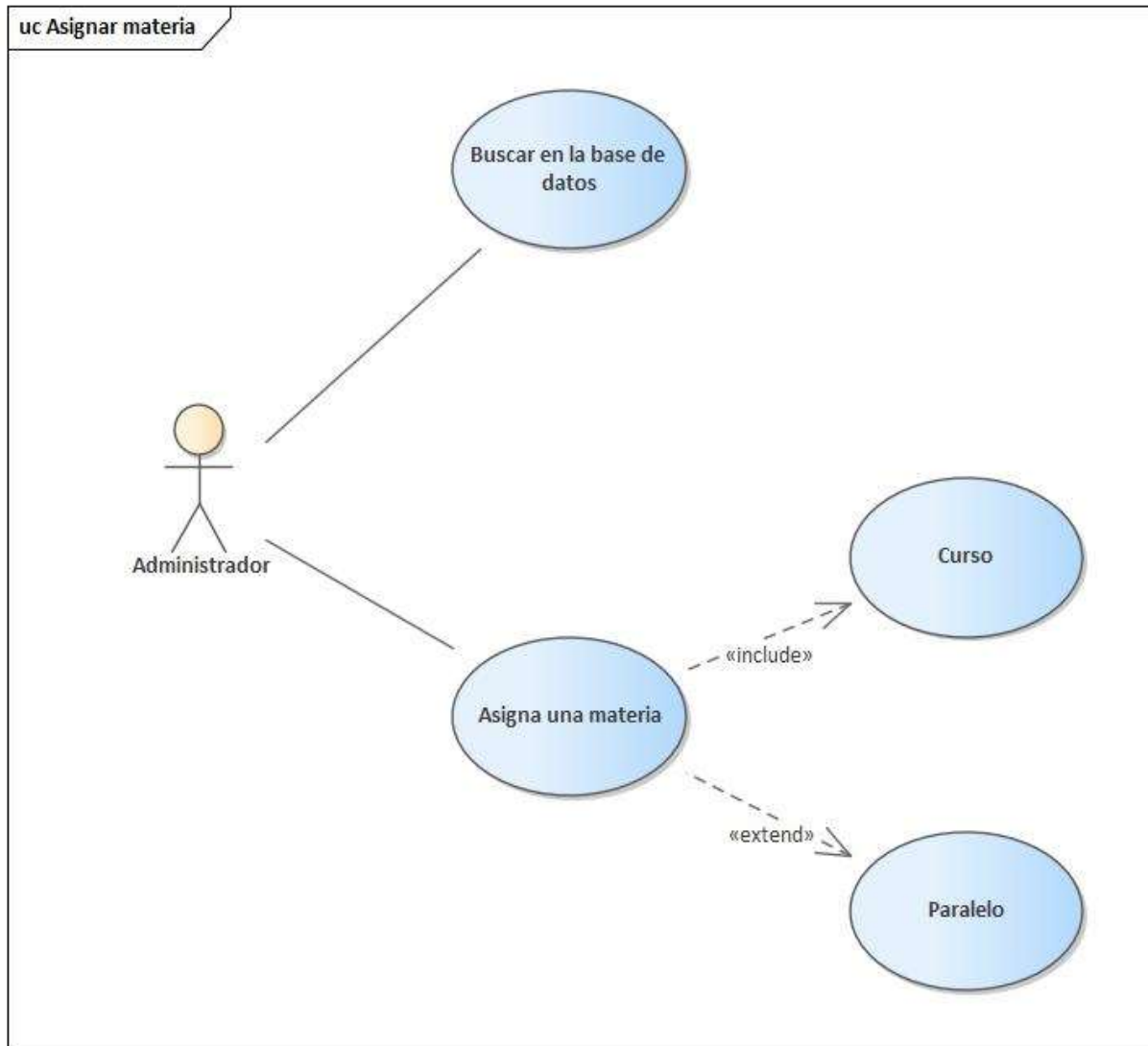


**Fuente:** Elaboración propia

### 3.4.2.6. Diagrama de caso de uso: Asignar materia al Maestro

Para asignar asignaturas al Maestros solo los actores principales pueden realizar esa acción.

**Figura 3.6.** Diagrama de casos de uso – Asignar materias

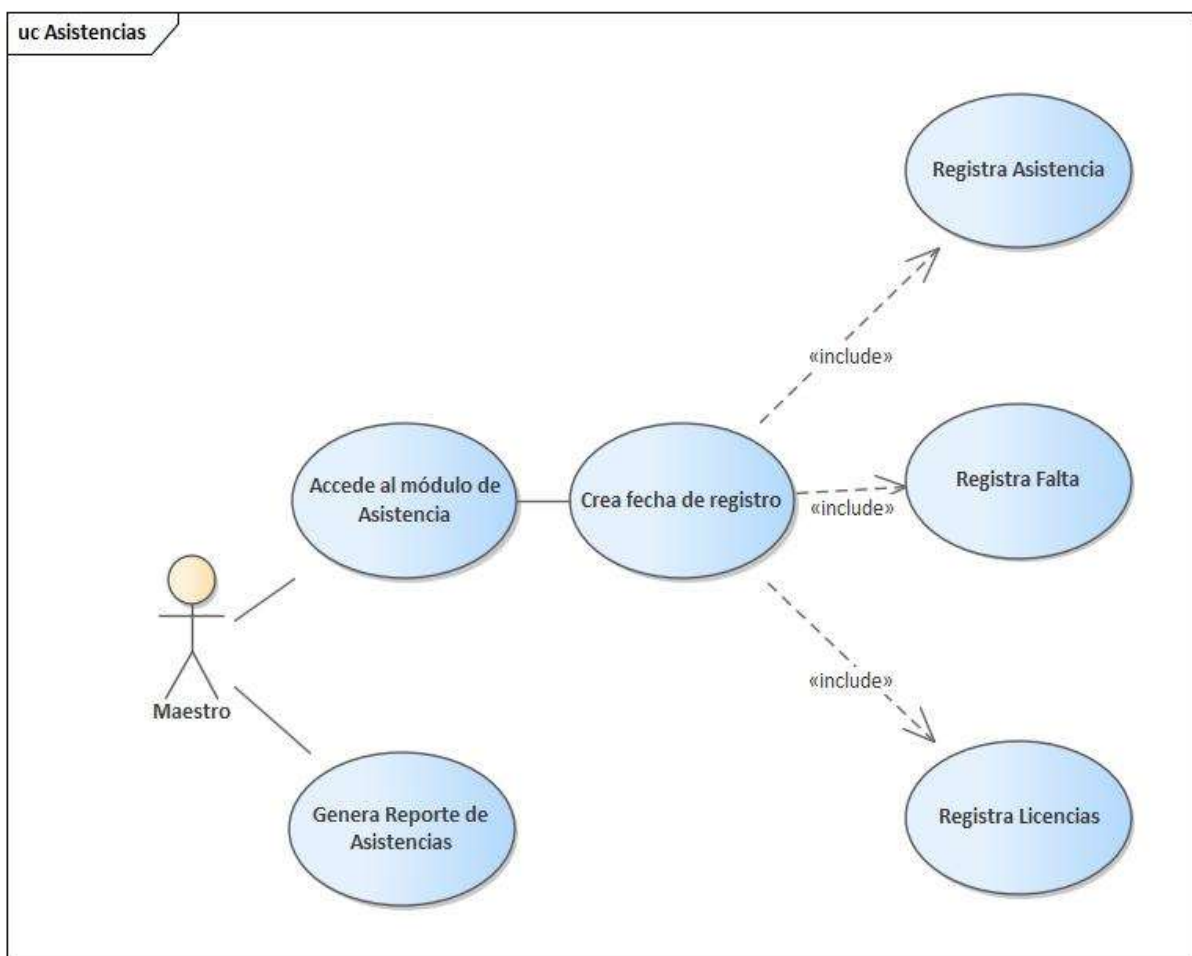


**Fuente:** Elaboración propia

### 3.4.2.7. Diagrama de caso de uso: Gestión de Asistencias

Para la gestión de asistencias, inasistencias y faltas, solo los maestros de la Unidad Educativa pueden realizar esa acción.

**Figura 3.7.** Diagrama de casos de uso – Asignar materias

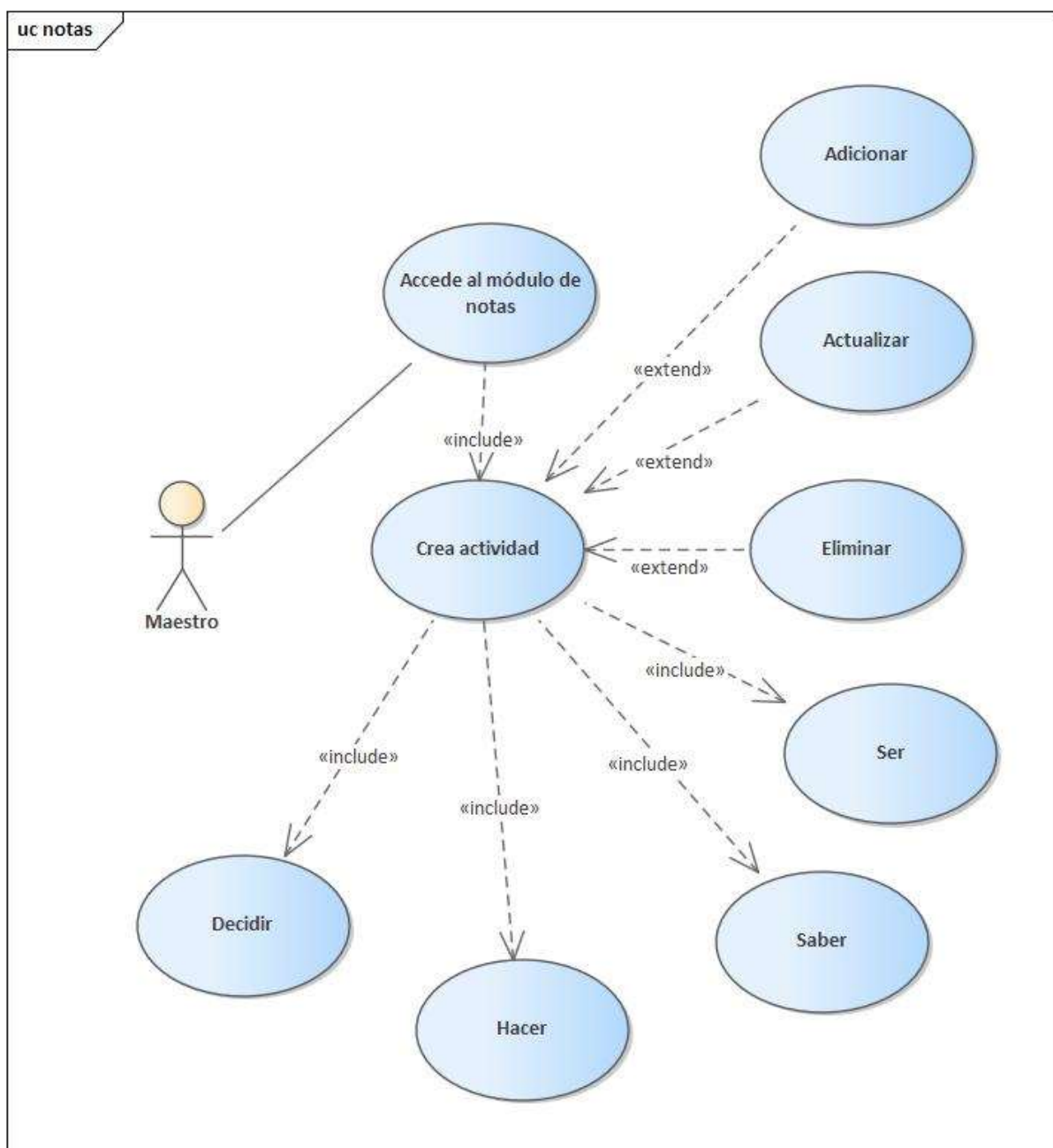


**Fuente:** Elaboración propia

### 3.4.2.8. Diagrama de caso de uso: Gestión de Notas

Para la gestión de calificaciones, solo los maestros de la Unidad Educativa pueden realizar esa acción.

Figura 3.8. Diagrama de casos de uso – Asignar materias

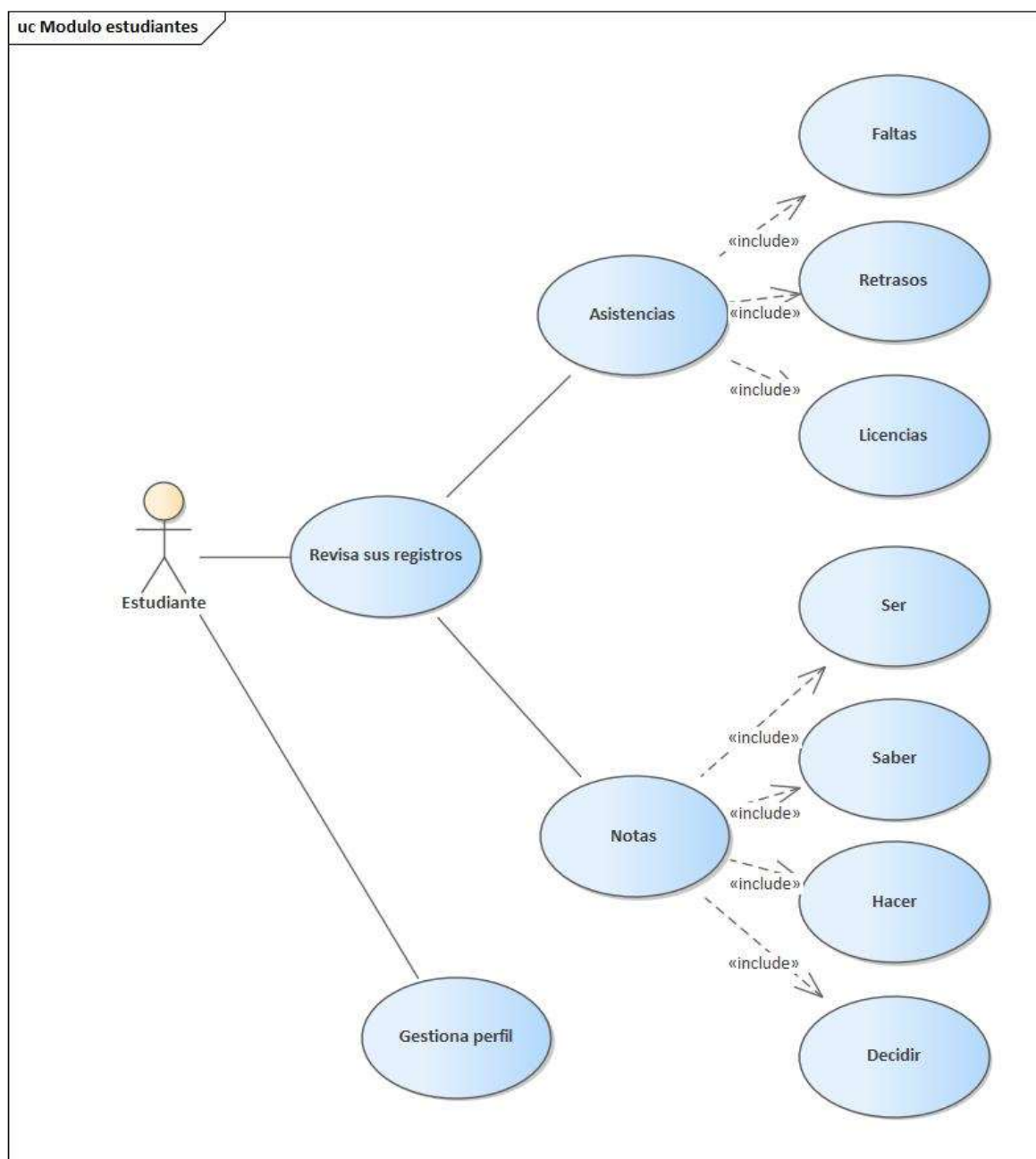


Fuente: Elaboración propia

### 3.4.2.9. Diagrama de caso de uso: Seguimiento del estudiante

Para el seguimiento de las asistencias, faltas y licencias, los estudiantes de la Unidad Educativa tienen el rol para visualizar.

**Figura 3.9.** Diagrama de casos de uso – Seguimiento Estudiantes



**Fuente:** Elaboración propia

### 3.4.2.10. Descripción de casos de uso

En esta sección se describe los casos de uso que fueron descritos en las figuras anteriores.

**Tabla 3.16.** Registro de usuario

<b>Caso de Uso</b>	<b>Registro de usuario</b>
<b>Actor</b>	Administrador, Maestros, estudiantes
<b>Pre-Condición</b>	El administrador debe estar autenticado en el sistema y los maestros y estudiantes debe ser registrado por el administrador y asignar un rol, con usuario contraseña.
<b>Escenarios básicos</b>	<p>El administrador registra a los Estudiantes.</p> <ol style="list-style-type: none"> <li>1. El administrador selecciona la opción de registro de estudiantes.</li> <li>2. El sistema despliega la interfaz de registro del estudiante nuevo.</li> <li>3. El administrador ingresa datos solicitados.</li> <li>4. El administrador guarda los datos introducidos.</li> <li>5. El sistema automáticamente asigna por defecto su número de documento de identidad como usuario y contraseña.</li> <li>6. Despliega alerta de conformidad.</li> <li>7. El administrador modifica los datos en caso de error.</li> </ol> <p>El administrador registra a los maestros.</p> <ol style="list-style-type: none"> <li>8. El administrador selecciona la opción de registro de Maestros.</li> <li>9. El sistema despliega la interfaz de registro del maestro nuevo.</li> <li>10. El administrador ingresa datos solicitados.</li> <li>11. El administrador guarda los datos introducidos.</li> <li>12. El sistema automáticamente asigna por defecto su número de documento de identidad como usuario y contraseña.</li> <li>13. Despliega alerta de conformidad.</li> <li>14. El administrador modifica los datos en caso de error.</li> </ol>
<b>ESCENARIOS ITERATIVOS</b>	
<b>Alternativa 1</b>	- Si en el escenario 5 y 12 del escenario básico el sistema encuentra datos duplicados en usuario, entonces el sistema despliega un mensaje de alerta
<b>Post Condición</b>	- Los datos de registro del nuevo usuario están almacenados en la Base de Datos

**Fuente:** Elaboración propia

**Tabla 3.17.** Registro de Áreas y Campo de Saberes y Conocimientos

<b>Caso de Uso</b>		<b>Registro de usuario</b>
<b>Actor</b>		Administrador
<b>Pre-Condición</b>		El administrador debe estar autenticado en el sistema
<b>Escenarios básicos</b>		<p>En el caso de uso el administrador requiere registrar Áreas y Campo de Saberes y Conocimientos.</p> <ol style="list-style-type: none"> <li>1. El administrador selecciona la opción de Campo de Saberes y Conocimientos.</li> <li>2. El sistema despliega la interfaz de registro de Campo de Saberes y Conocimientos.</li> <li>3. El Administrador ingresa los datos requeridos.</li> <li>4. El Administrador pulsa el botón guardar.</li> <li>5. Despliega un mensaje de conformidad</li> </ol>
<b>ESCENARIOS ITERATIVOS</b>		
<b>Alternativa 1</b>	-	Si en el escenario 4 del escenario básico el sistema encuentra datos duplicado, entonces el sistema despliega un mensaje de alerta
<b>Post Condición</b>	-	Los datos de registro del nuevo usuario están almacenados en la Base de Datos

**Fuente:** Elaboración propia

**Tabla 3.18.** Inscripción de Estudiantes

<b>Caso de Uso</b>		<b>Registro de usuario</b>
<b>Actor</b>		Administrador
<b>Pre-Condición</b>		El administrador debe estar autenticado en el sistema
<b>Escenarios básicos</b>		<p>En el caso de uso el administrador requiere registrar Áreas y Campo de Saberes y Conocimientos.</p> <ol style="list-style-type: none"> <li>1. El administrador selecciona la opción de Campo</li> </ol>



---

de Saberes y Conocimientos.

2. El sistema despliega la interfaz de registro de Campo de Saberes y Conocimientos.
3. El Administrador ingresa los datos requeridos.
4. El Administrador pulsa el botón guardar.
5. Despliega un mensaje de conformidad

---

### ESCENARIOS ITERATIVOS

**Alternativa 1** - Si en el punto 4 del escenario básico el sistema encuentra datos duplicado, entonces el sistema despliega un mensaje de alerta

---

**Post Condición** - Los datos de registro del nuevo usuario están almacenados en la Base de Datos

---

**Fuente:** Elaboración propia

**Tabla 3.19.** Asignar materias al maestro

Caso de Uso	Registro de usuario
<b>Actor</b>	Administrador
<b>Pre-Condición</b>	El administrador debe estar autenticado en el sistema
<b>Escenarios básicos</b>	En el caso de uso el administrador requiere asignar asignaturas al maestro. <ol style="list-style-type: none"><li>1. El administrador selecciona la opción para asignar materias a los maestros.</li><li>2. El sistema despliega la interfaz para la asignación.</li><li>3. El Administrador selecciona maestro, curso y asignatura.</li><li>4. El Administrador pulsa el botón guardar.</li><li>5. Despliega un mensaje de conformidad</li></ol>

---

### ESCENARIOS ITERATIVOS

**Alternativa** - Si en el punto 4 del escenario básico el sistema

---

<b>1</b>	encuentra datos duplicado, entonces el sistema despliega un mensaje de alerta
<b>Post Condición</b>	- Los datos de registro del nuevo usuario están almacenados en la Base de Datos

**Fuente:** Elaboración propia

**Tabla 3.20.** Gestión de asistencias

<b>Caso de Uso</b>	<b>Registro de usuario</b>
<b>Actor</b>	Maestro
<b>Pre-Condición</b>	El maestro debe estar autenticado en el sistema
<b>Escenarios básicos</b>	<p>En el caso de uso el maestro requiere registrar asistencias del estudiante.</p> <ol style="list-style-type: none"> <li>1. El Maestro selecciona la opción para registrar asistencias.</li> <li>2. El sistema despliega la interfaz para crea una fecha de asistencia</li> <li>3. El Administrador selecciona (falta, licencia o Retraso).</li> <li>4. El Administrador pulsa el botón guardar.</li> <li>5. Despliega un mensaje de conformidad</li> </ol>

#### **ESCENARIOS ITERATIVOS**

<b>Alternativa 1</b>	- Si en el punto 4 del escenario básico el sistema encuentra datos duplicado, entonces el sistema despliega un mensaje de alerta
<b>Post Condición</b>	<ul style="list-style-type: none"> <li>- Los datos de registro del nuevo usuario están almacenados en la Base de Datos</li> <li>- Genera un reporte de asistencias</li> </ul>

**Fuente:** Elaboración propia

**Tabla 3.21.** Gestión de Notas

<b>Caso de Uso</b>		<b>Registro de usuario</b>
<b>Actor</b>		Maestro
<b>Pre-Condición</b>		El Maestro debe estar autenticado en el sistema
<b>Escenarios básicos</b>		<p>En el caso de uso el maestro requiere registrar calificaciones de las actividades pedagógicas.</p> <ol style="list-style-type: none"> <li>1. El Maestro selecciona la opción para registrar calificación.</li> <li>2. El sistema despliega la interfaz para crea la actividad.</li> <li>3. El Administrador selecciona fecha, actividad y la dimensión (ser, saber, hacer y decidir)</li> <li>4. El Administrador pulsa el botón guardar.</li> <li>5. Despliega un mensaje de conformidad</li> </ol>
<b>ESCENARIOS ITERATIVOS</b>		
<b>Alternativa 1</b>	-	Si en el punto 4 del escenario básico el sistema encuentra datos duplicado, entonces el sistema despliega un mensaje de alerta
<b>Post Condición</b>	-	<ul style="list-style-type: none"> <li>- Los datos de registro del nuevo usuario están almacenados en la Base de Datos</li> <li>- Genera un reporte de notas de forma trimestral</li> </ul>
<b>Fuente:</b> Elaboración propia		

**Tabla 3.22.** Seguimiento del Estudiante

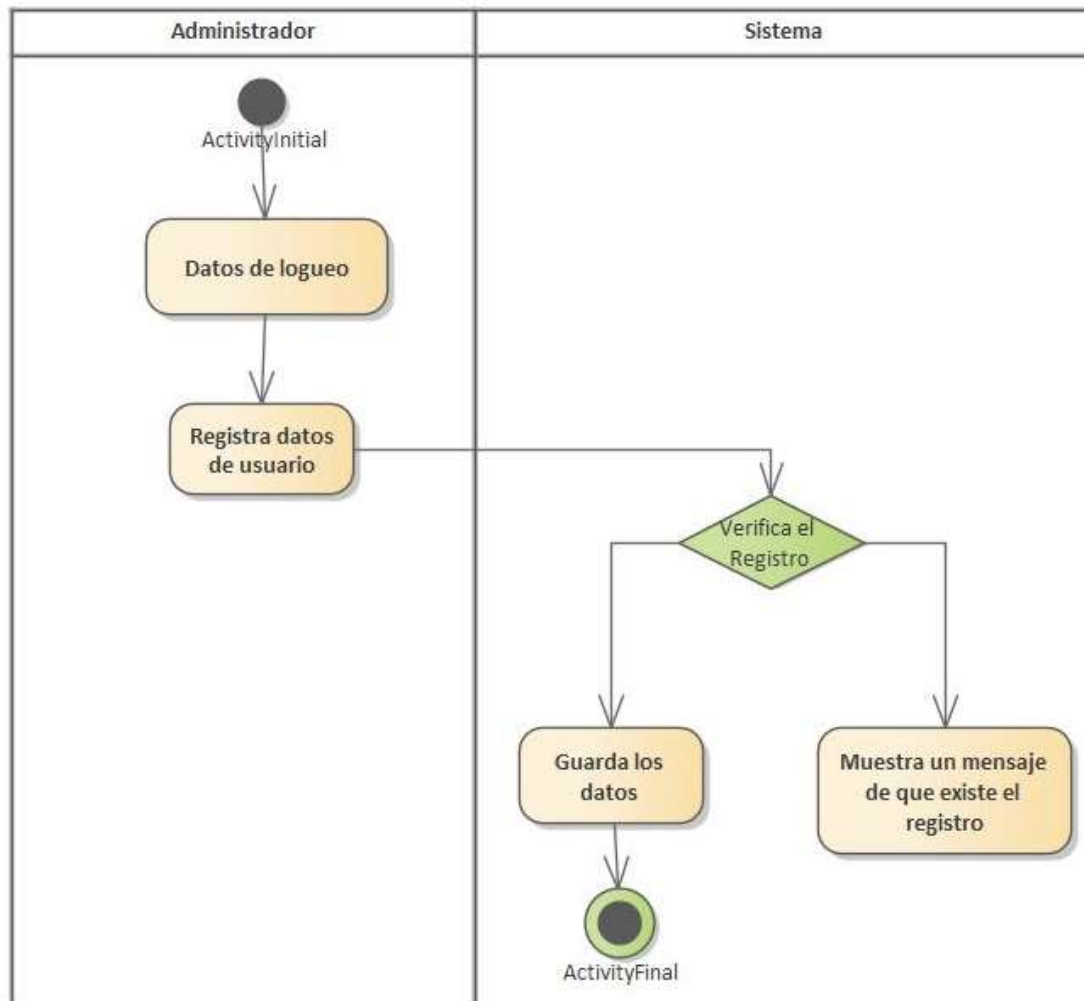
<b>Caso de Uso</b>		<b>Registro de usuario</b>
<b>Actor</b>	Estudiante	
<b>Pre-Condición</b>	El Estudiante debe estar autenticado en el sistema	
<b>Escenarios básicos</b>	<p>En el caso de uso el estudiante requiere consultar las asistencias, inasistencias, licencias, faltas y notas de las tareas completadas.</p> <ol style="list-style-type: none"> <li>1. El Estudiante selecciona la opción para ver asistencia.</li> <li>2. El sistema despliega la interfaz con los detalles de la asistencia.</li> <li>3. El Estudiante selecciona la opción para ver notas.</li> <li>4. El sistema despliega la interfaz con los detalles de la asistencia.</li> </ol>	
<b>ESCENARIOS ITERATIVOS</b>		
<b>Alternativa 1</b>	Si en el punto 4 del escenario básico el sistema encuentra datos duplicado, entonces el sistema despliega un mensaje de alerta	
<b>Post Condición</b>	-	<ul style="list-style-type: none"> <li>- Muestra consultas</li> <li>- Edita perfil de usuario.</li> </ul>

**Fuente:** Elaboración propia

### 3.4.2.11. Diagrama de Actividades

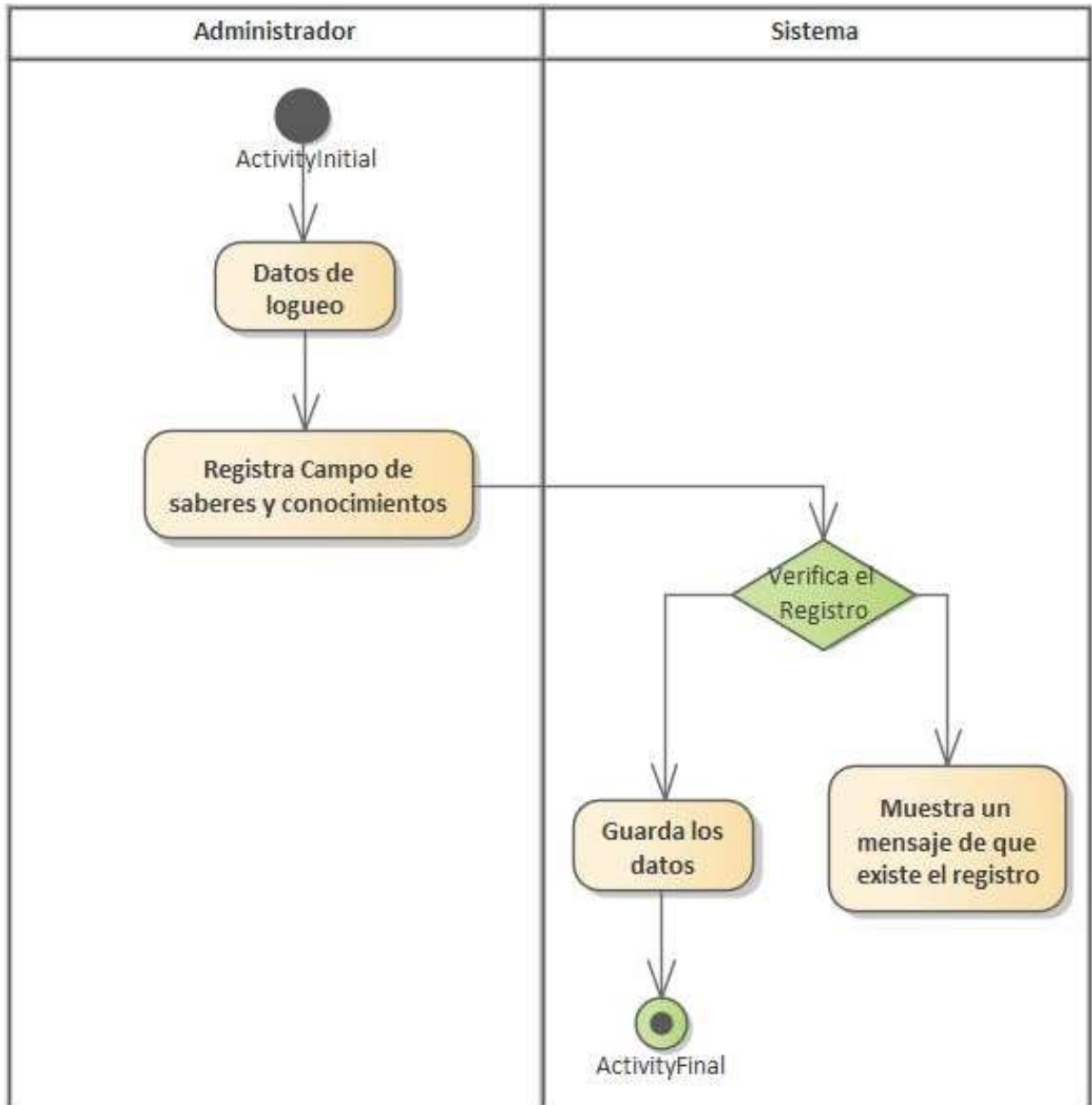
El diagrama de actividades se empleará para definir la secuencia de pasos lógicos que se deben seguir para cumplir con una determinada función en el sistema web.

**Figura 3.10.** Diagrama de actividades – Registro de usuario



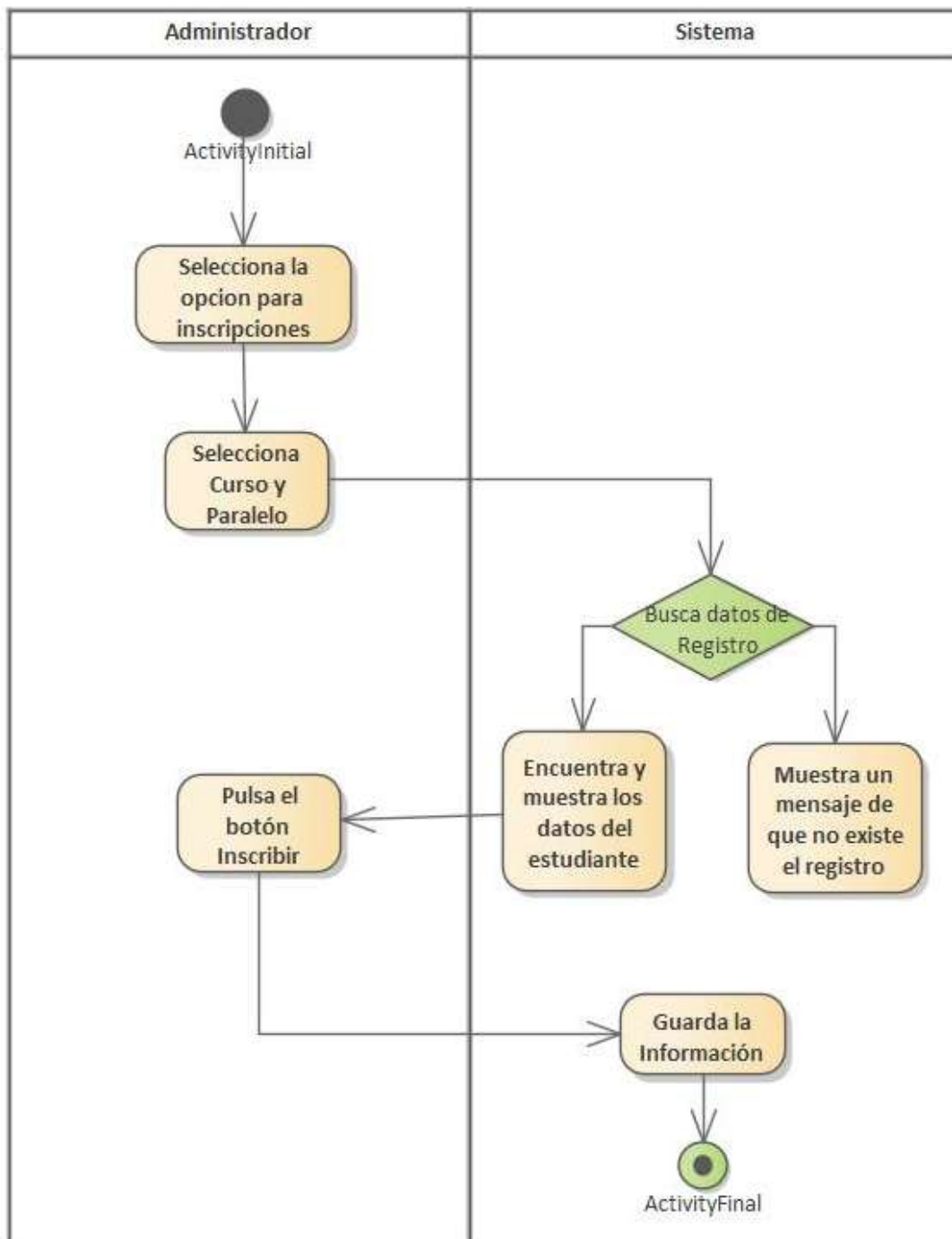
**Fuente:** Elaboración propia

**Figura 3.11.** Diagrama de actividades – Registro de Campo de Saberes y Conocimientos



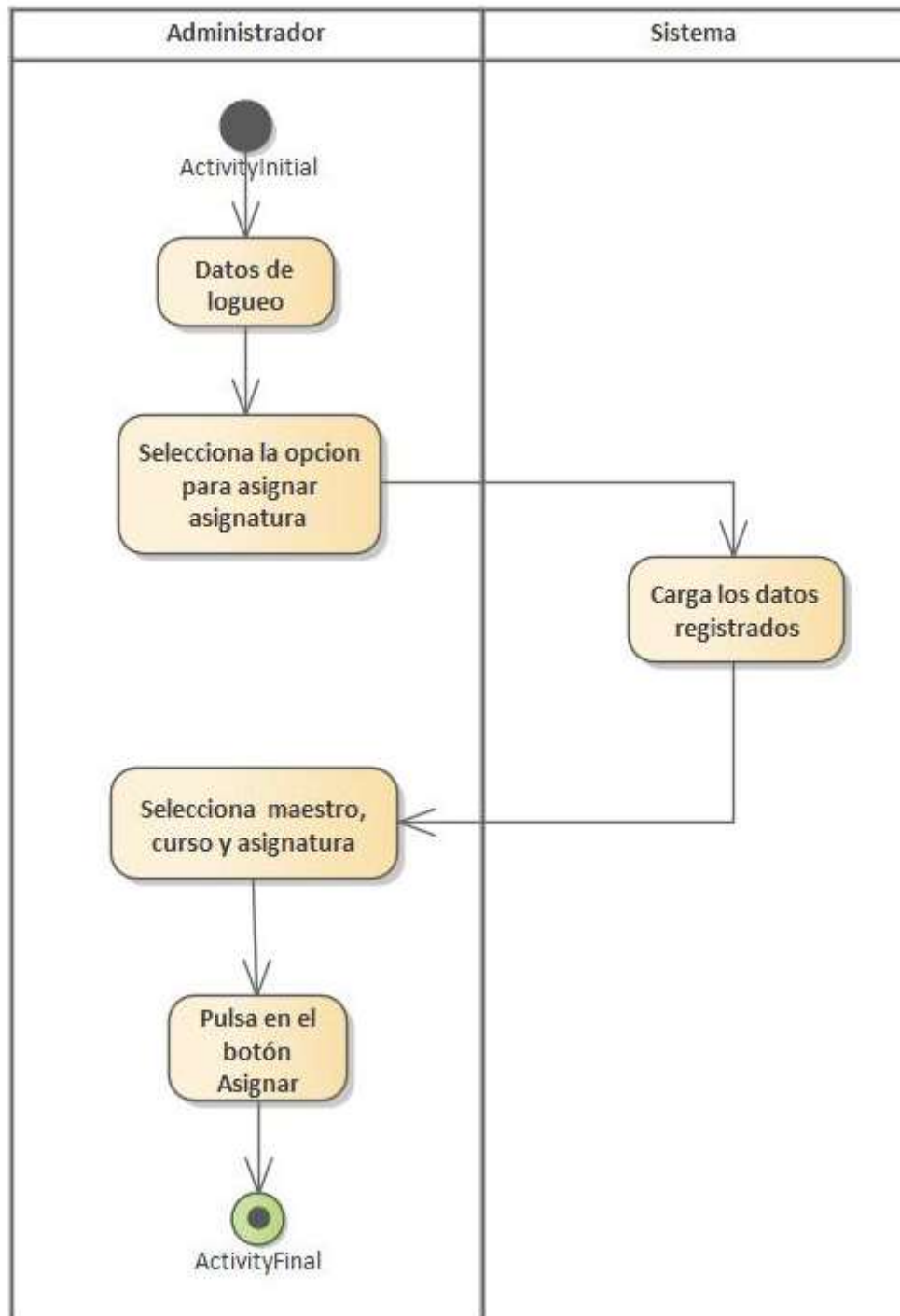
**Fuente:** Elaboración propia

**Figura 3.12.** Diagrama de actividades – Inscripciones



**Fuente:** Elaboración propia

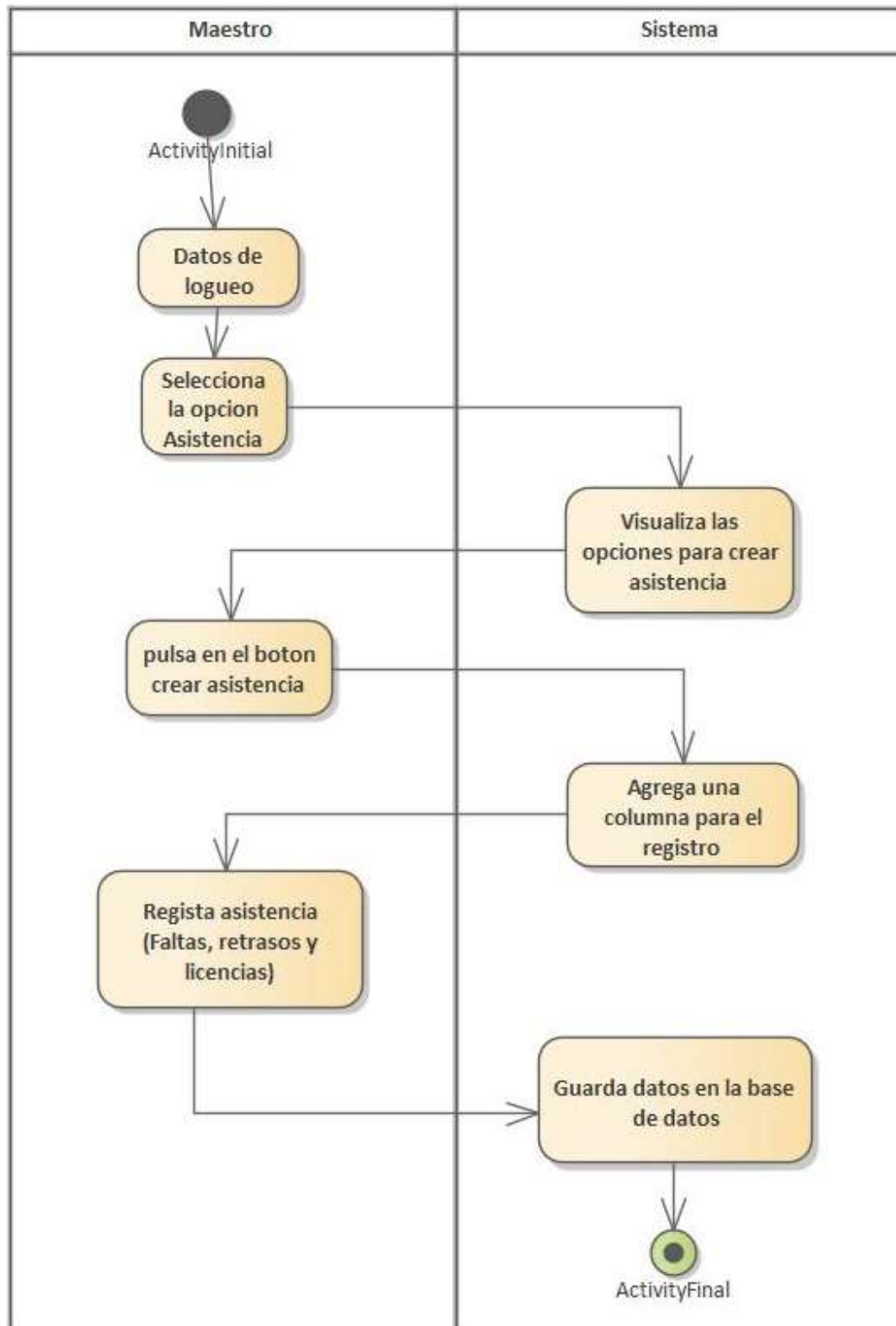
**Figura 3.13.** Diagrama de actividades – Asignar Materia al Maestro



**Fuente:** Elaboración propia

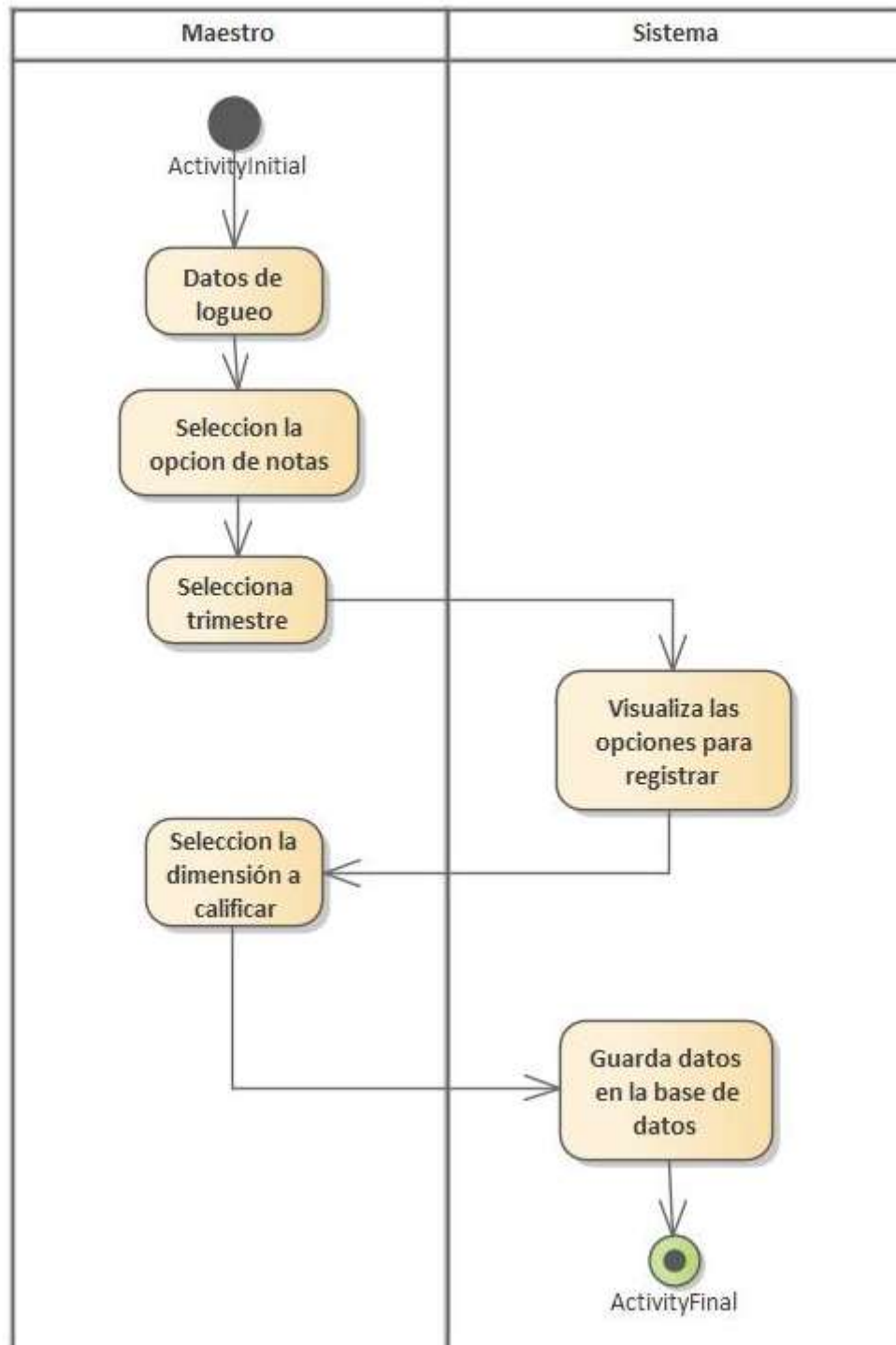


**Figura 3.14.** Diagrama de actividades – Registro de Asistencia



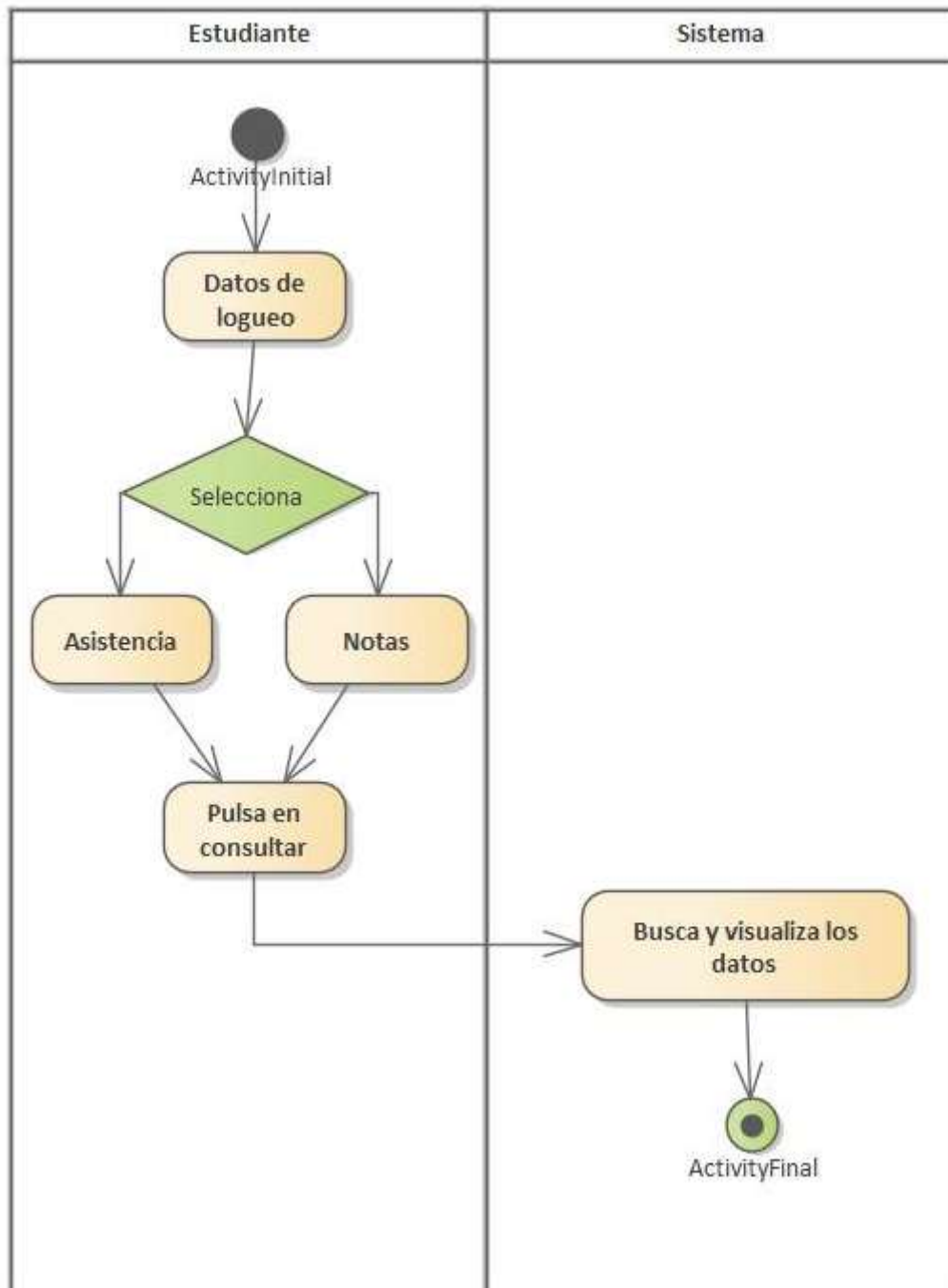
**Fuente:** Elaboración propia

**Figura 3.15.** Diagrama de actividades – Registro de Calificaciones



**Fuente:** Elaboración propia

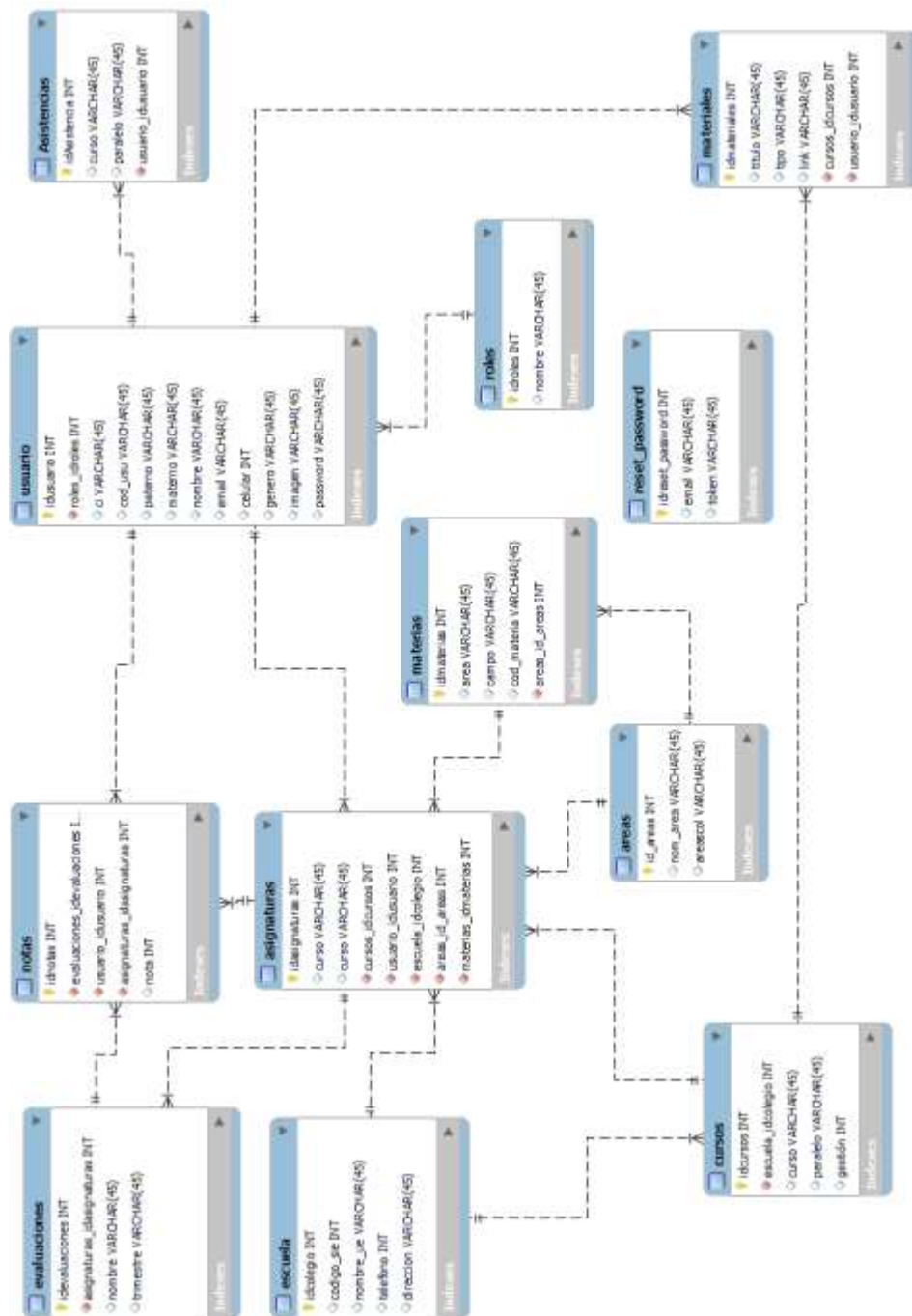
**Figura 3.16.** Diagrama de actividades – Seguimiento del Estudiante



**Fuente:** Elaboración propia

### 3.4.2.12. Diagrama de Clases

Figura 3.17. Diagrama de Clases

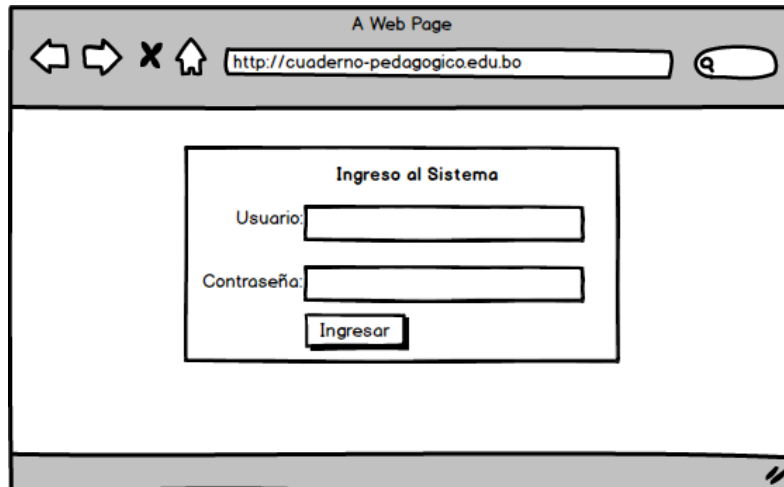


Fuente: Elaboración propia

### 3.4.2.13. Diagrama de Presentación

Los diagramas de presentación, que se muestra a continuación, permiten visualizar el resultado final de las interfaces que tendrá el sistema cumpliendo con los requerimientos planteados.

**Figura 3.18.** Diagrama de presentación – Ingreso al Sistema



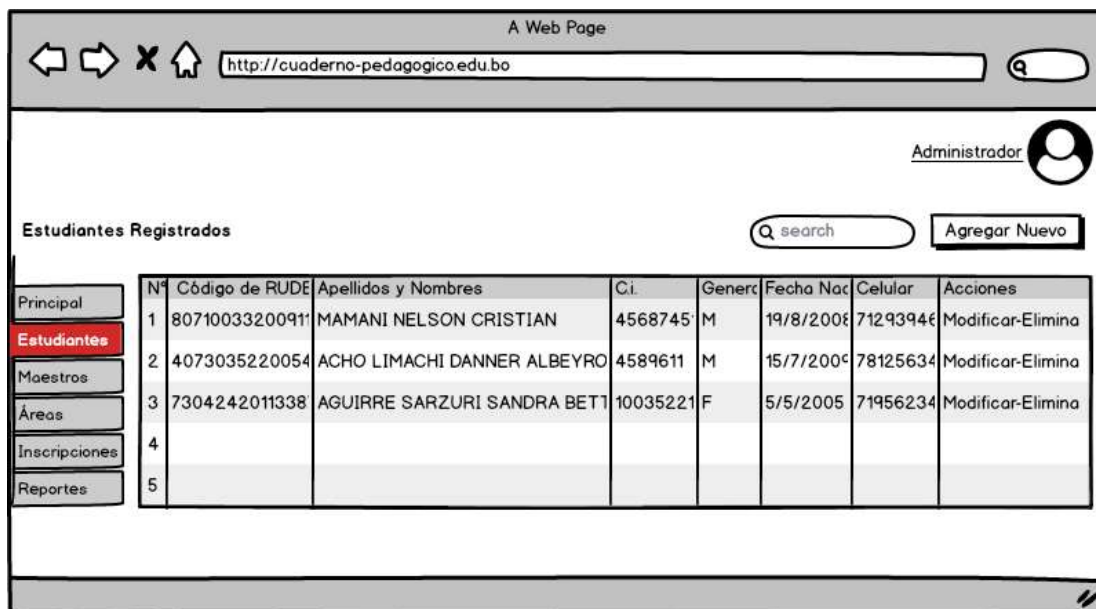
**Fuente:** Elaboración propia

**Figura 3.19.** Diagrama de presentación – Ingreso al Sistema



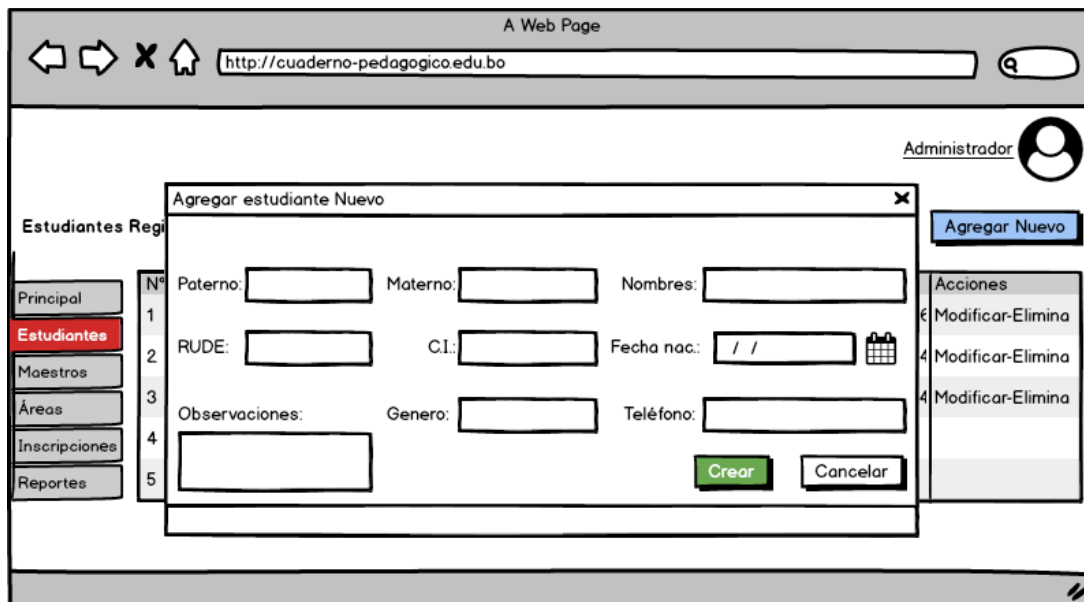
**Fuente:** Elaboración propia

Figura 3.20. Diagrama de presentación –Estudiantes registrados



Fuente: Elaboración propia

Figura 3.21. Diagrama de presentación – Formulario de registro del estudiante



Fuente: Elaboración propia

Figura 3.22. Diagrama de presentación – Formulario de registro de maestros

A Web Page  
http://cuaderno-pedagogico.edu.bo

Administrador

Maestros Registro

Principal  
Estudiantes  
Maestros  
Áreas  
Inscripciones  
Reportes

Agregar Maestro Nuevo

Paterno:  Materno:  Nombres:

RDA:  C:  Fecha nac.:

Especialidad:  Genero:  Teléfono:

Agregar Nuevo

Acciones  
Modificar - Eliminar  
Modificar - Eliminar

Fuente: Elaboración propia

Figura 3.23. Diagrama de presentación – Registro de Área

A Web Page  
http://cuaderno-pedagogico.edu.bo

Administrador

Maestros Registro

Principal  
Estudiantes  
Maestros  
Áreas  
Inscripciones  
Reportes

Registrar área

Campo:

Área:

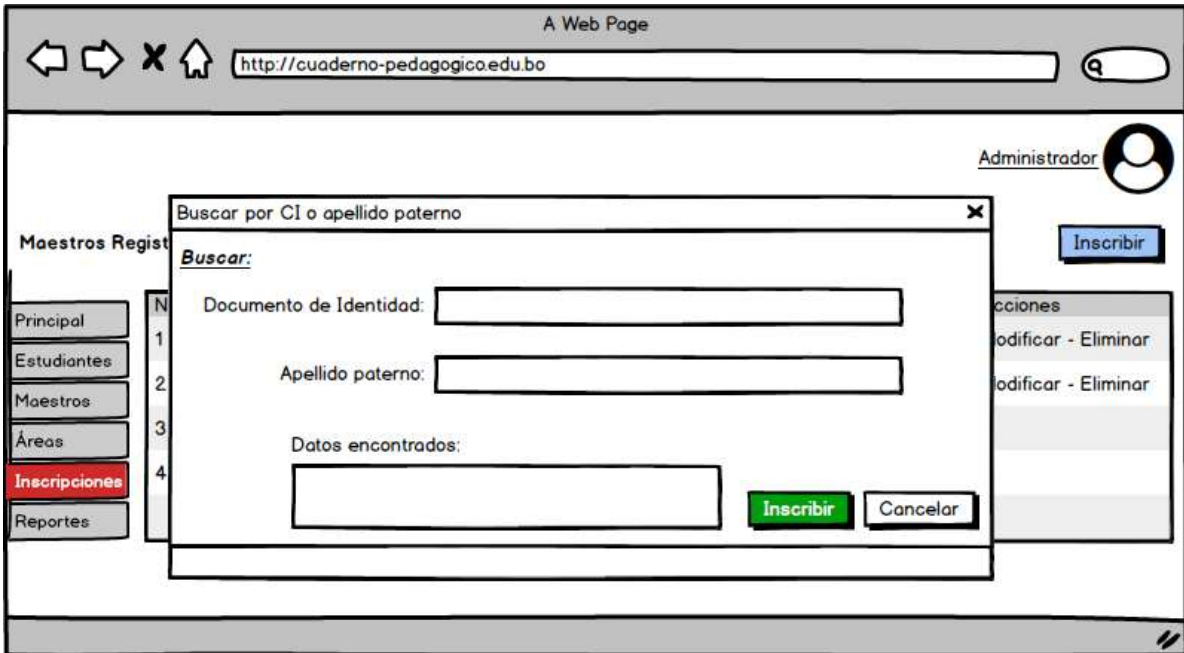
Código de área:

Agregar área

Acciones  
Modificar - Eliminar  
Modificar - Eliminar

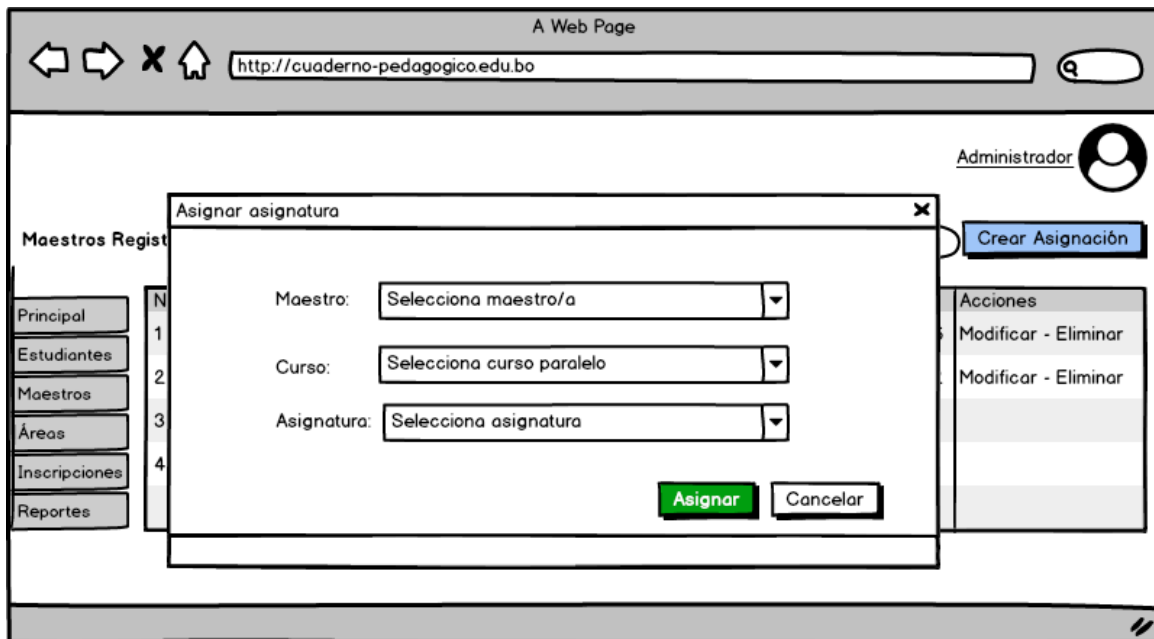
Fuente: Elaboración propia

Figura 3.24. Diagrama de presentación – Inscripción de estudiantes



Fuente: Elaboración propia

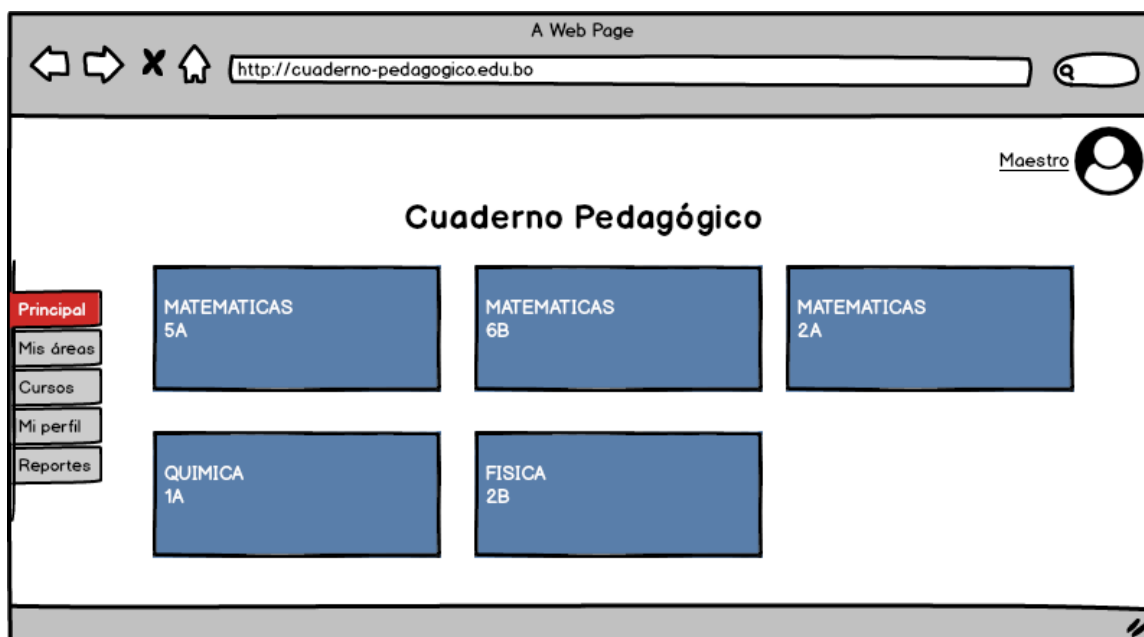
Figura 3.25. Diagrama de presentación – Asignación de Maestros



Fuente: Elaboración propia

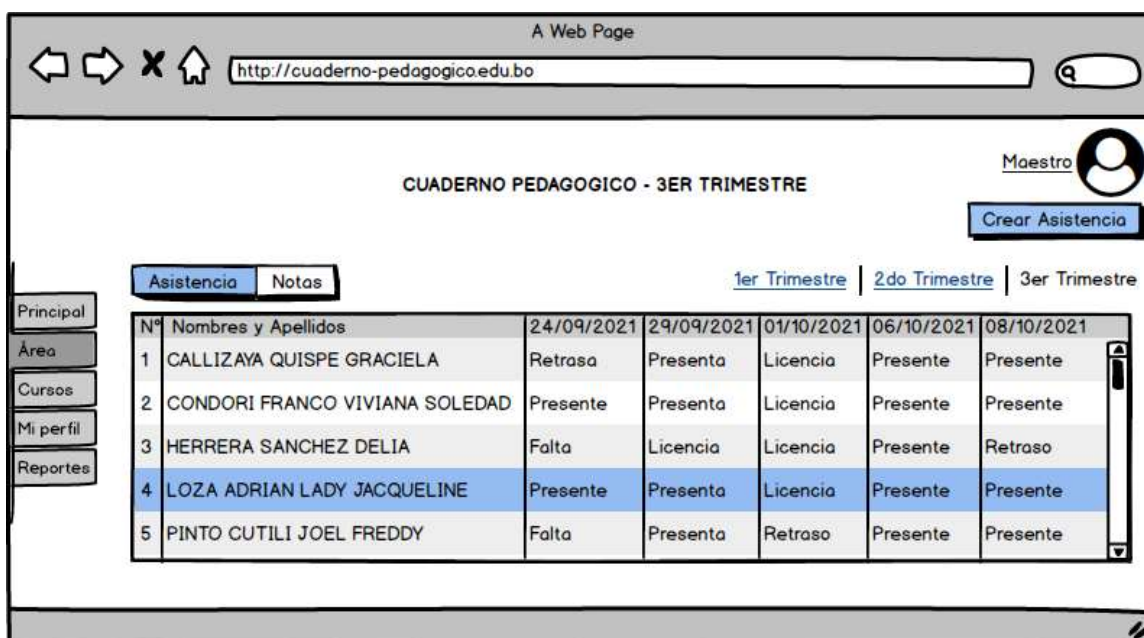


Figura 3.26. Diagrama de presentación – Panel principal del Maestro



Fuente: Elaboración propia

Figura 3.27. Diagrama de presentación – Registro de Asistencia



Fuente: Elaboración propia

Figura 3.28. Diagrama de presentación – Registro de Calificaciones

Principal  
Área  
Cursos  
Mi perfil  
Reportes

Asistencia | **Notas**

1er Trimestre | **2do Trimestre** | 3er Trimestre

N°	Nombres y Apellidos	Participación (Saber)	Cuaderno (Hacer)	Exposición (Saber)	Maqueta (Hacer)	Proyecto (Decidir)
1	CALLIZAYA QUISPE GRACIELA	90	86	100	87	78
2	CONDORI FRANCO VIVIANA SOLEDAD	89	90	100	98	99
3	HERRERA SANCHEZ DELIA	89	87	95	100	99
4	LOZA ADRIAN LADY JACQUELINE	95	96	100	78	95
5	PINTO CUTILI JOEL FREDDY	98	78	69	86	100

Fuente: Elaboración propia

Figura 3.29. Diagrama de presentación – Panel del Estudiante

Principal  
mi Área  
Cursos  
Mi perfil  
Reportes

Asistencia | **Calificaciones**

3er Trimestre

N°	Fecha	Actividades Calificadas	Calificación
1	29/09/2021	Participación	90
2	13/10/2021	Cuaderno	86
3	20/10/2021	Exposición	100
4	22/10/2021	Maqueta	87
5	29/10/2021	Proyecto	78

Fuente: Elaboración propia

### 3.5. POST-GAME

Durante esta etapa post-game, se realizará las actividades de prueba a los módulos desarrollados en sistema de información.

#### 3.5.1. Prueba del Sistema

##### 3.5.1.1. Pruebas de caja blanca

En este apartado se realizaron la comprobación de la interactividad satisfactoria a través de sus interfaces, tanto internas como externas, cubren la funcionalidad establecida y se ajustan a los requisitos no funcionales especificados en las verificaciones correspondientes.

**Tabla 3.23.** Pruebas unitarias

<b>Módulo</b>	<b>Número de pruebas</b>	<b>Resultados</b>
<b>Inicio de sesión</b>	25 usuarios conectados	El sistema responde satisfactoriamente
<b>Registro de Estudiantes</b>	Se realizó el registro de 307 estudiantes	El sistema responde satisfactoriamente
<b>Registro de Maestros</b>	Se realizó el registro de 28 maestros.	El sistema responde satisfactoriamente
<b>Registro de asignaturas</b>	Se realizó el registro de 19 Áreas de saberes y conocimientos.	El sistema responde satisfactoriamente

**Fuente:** Elaboración propia

Evaluación, se ha realizado las pruebas de todos los componentes para encontrar algún problema que puedan surgir en las iteraciones al realizar las pruebas, y no se encontraron fallas en el sistema desarrollado.

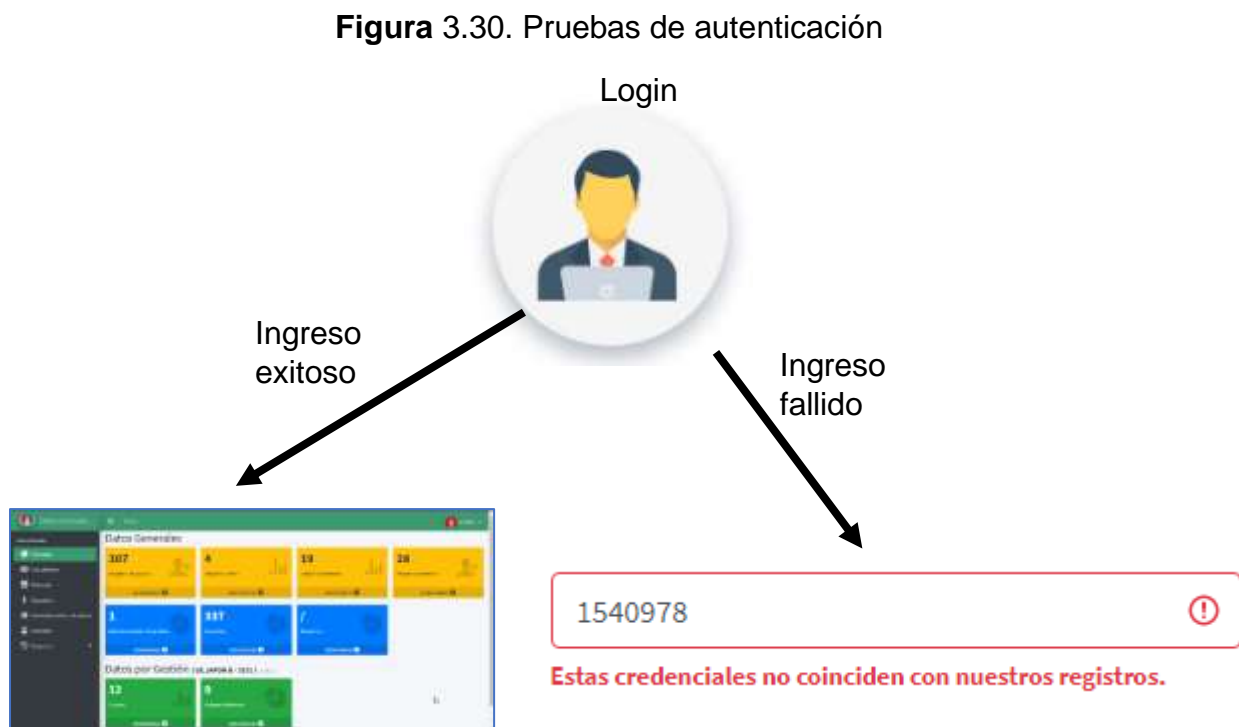
### 3.5.1.2. Pruebas de caja negra

La prueba de caja negra consiste en probar cada una de las funciones del sistema que fue desarrollado. Con este tipo de prueba se debe buscar que las funciones sean operativas, además se debe agotar al sistema de tal manera buscar la mayor cantidad de errores.

Son pruebas sobre la interfaz del software. A continuación, se muestran algunas pruebas relevantes.

#### Pruebas de autenticación

Si el usuario que desea ingresar al sistema es el correcto, desplegará la interfaz gráfica de ingreso, mostrando todos los menús correspondientes al rol, caso contrario se mostrará un mensaje de acceso denegado.

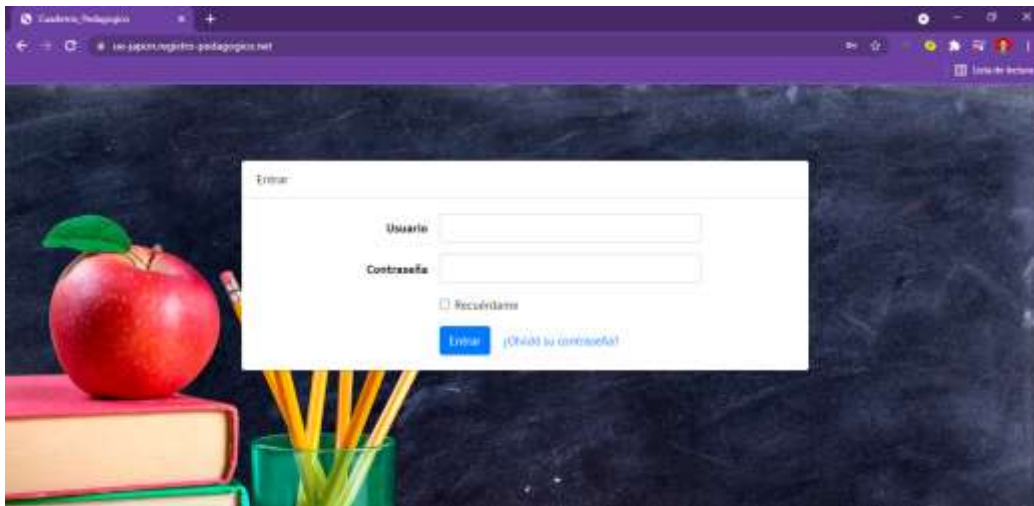


**Fuente:** Elaboración propia

### 3.6. MODELO DE IMPLANTACIÓN DEL PROGRAMA

Los siguientes gráficos modelan los diseños de interfaces de los usuarios a los diferentes módulos, para el ingreso al sistema si la información introducida en la ventana de inicio de sesión es correcta ingresará al sistema caso contrario mostrará error y posteriormente el sistema habilita los menús correspondientes de acuerdo al rol de los usuarios.

**Figura 3.31.** Ingreso al Sistema



**Fuente:** Elaboración propia

En el módulo de administración donde se mostrará todos los módulos para la gestión y administración del Cuaderno Pedagógico Web.

**Figura 3.32.** Panel de Administración



**Fuente:** Elaboración propia

En el menú del estudiante, es el módulo donde gestiona altas, bajas y modificaciones de los estudiantes, organizando de forma ascendente según el apellido paterno. El documento de identidad es asignado por defecto como usuario y contraseña.

**Figura 3.33. Registro de Estudiantes**

N°	Código RUDE	Nombres y Apellidos	Cédula de Identidad	Genero	Fecha de Nacimiento	Celular	Acciones
1	1046000120001A	EVA EVA	13044589	Mujer	18/08/2004		[Edit] [Delete]
2	80710033200011	SALLUCO NELSON CRISTIAN	13277878	Varón	20/12/2002		[Edit] [Delete]
3	407303522000304	ACHO LIMACHI BEYMAR LIMBERTH	13828692	Varón	25/03/2000		[Edit] [Delete]
4	4073035220111809	ACHO LIMACHI DANHER ALBEYRD	13741799	Varón	18/12/2000		[Edit] [Delete]
5	407304212030203	ADRIAN PACHACUTI SANDRA ODALIS	13084122	Mujer	23/03/2000		[Edit] [Delete]
6	4073042420001A	AGUIRRE SARTURI AYDE REBECA	9903889	Mujer	23/03/2003		[Edit] [Delete]

**Fuente:** Elaboración propia

En el menú del maestro, es el módulo donde gestiona altas, bajas y modificaciones de los maestros, organizando de forma ascendente según los apellidos paternos. El documento de identidad es asignado por defecto como usuario y contraseña.

**Figura 3.34. Registro de Estudiantes**

Nombres y Apellidos	C.I.	R.D.A.	Genero	Fecha de nacimiento	Celular	Acciones
AGUILAR APAZA YHONARA	0130007		Mujer	01/01/2021		[Edit] [Delete]
ALIAGA ALCON OMAR	8236376		Varón	01/01/2021		[Edit] [Delete]
ARGANI PATZI JOSE LUIS	9972788		Varón	01/01/2002		[Edit] [Delete]
ARQUIPA CHOQUEHUANCA EDGAR	9483990		Varón	01/01/2021		[Edit] [Delete]
BALTAZAR LOPEZ JACKELINE GLADYS	4258303		Mujer	01/01/2021		[Edit] [Delete]

**Fuente:** Elaboración propia



En el menú de Materias, es el módulo donde gestiona altas, bajas y modificaciones de todas las asignaturas, para su posterior asignación a los maestros.

**Figura 3.35. Registro de Áreas**

#	Campos	Áreas	Código área	Acciones
1	CIENCIA, TECNOLOGÍA Y PRODUCCIÓN	Técnica Tecnológica General	TTG	[Edit] [Delete]
2	CIENCIA, TECNOLOGÍA Y PRODUCCIÓN	Técnica Tecnológica Especializada - Informática	TTE-SI	[Edit] [Delete]
3	CIENCIA, TECNOLOGÍA Y PRODUCCIÓN	MATEMATICAS	MAT	[Edit] [Delete]
4	CIENCIA, TECNOLOGÍA Y PRODUCCIÓN	Técnica Tecnológica Especializada - Mecánica Inst.	TTE-MI	[Edit] [Delete]
5	COMUNIDAD Y SOCIEDAD	ARTES PLASTICAS Y VISUALES	A.PLAST	[Edit] [Delete]
6	COMUNIDAD Y SOCIEDAD	EDUCACION MUSICAL	E.MUS	[Edit] [Delete]
7	COMUNIDAD Y SOCIEDAD	EDUCACION FISICA Y DEPORTES	EFD	[Edit] [Delete]
8	COMUNIDAD Y SOCIEDAD	COMUNICACION Y LENGUAJES	LCO	[Edit] [Delete]
9	COMUNIDAD Y SOCIEDAD	LENGUA EXTRANJERA	LEX	[Edit] [Delete]

**Fuente:** Elaboración propia

En el menú de inscripciones, es el módulo donde gestiona las inscripciones a partir de los datos de los estudiantes registrados en el sistema.

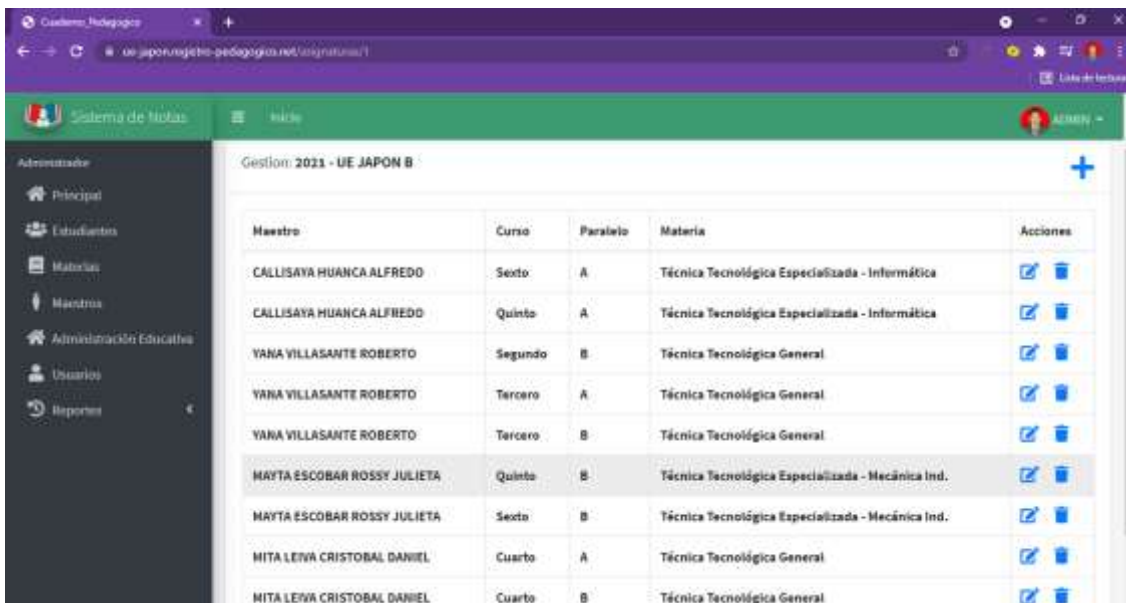
**Figura 3.36. Inscripción de Estudiantes**

Nombre	Numero identificación	Genero	Fecha de Nacimiento	Celular	Acciones
AJACOPA COARITE ROLY ALEXANDER	12833428	Varón	22/07/2008		[X]
ALAMI CONDORI DIANA ROSITA	14928325	Mujer	17/05/2008		[X]
ARGOLLO ZEGARRA ALEXIS FERRANDO	4873052020189620	Varón	18/05/2008		[X]
ARUQUIPA ROJAS JOEL ABRAHAM	15485513	Varón	19/01/2009		[X]
CALA TOLA BLADIMIR JHONNY	13815899	Varón	17/04/2009		[X]
CANAVIRI MAMARI JHOSELIN	15338527	Mujer	22/01/2009		[X]
CANAVIRI FLORES SHADIRA PAOLA	1522771A	Mujer	28/01/2009		[X]
CARLO GOMEZ JHOEL KEVIN	4873035220141701	Varón	13/11/2008		[X]
CARVAJAL VARGAS JHUMTSU ROSALI	14008844	Mujer	19/01/2009		[X]

**Fuente:** Elaboración propia

En el menú de asignar asignatura, es el módulo donde gestiona la asignación de maestros, curso y área de saberes y conocimientos

**Figura 3.37.** Asignación de Maestros



Maestro	Curso	Paralelo	Materia	Acciones
CALLISAYA HUANCA ALFREDDO	Sexto	A	Técnica Tecnológica Especializada - Informática	[Edit] [Delete]
CALLISAYA HUANCA ALFREDDO	Quinto	A	Técnica Tecnológica Especializada - Informática	[Edit] [Delete]
YANA VILLASANTE ROBERTO	Segundo	B	Técnica Tecnológica General	[Edit] [Delete]
YANA VILLASANTE ROBERTO	Tercero	A	Técnica Tecnológica General	[Edit] [Delete]
YANA VILLASANTE ROBERTO	Tercero	B	Técnica Tecnológica General	[Edit] [Delete]
MAYTA ESCOBAR ROSSY JULIETA	Quinto	B	Técnica Tecnológica Especializada - Mecánica Ind.	[Edit] [Delete]
MAYTA ESCOBAR ROSSY JULIETA	Sexto	B	Técnica Tecnológica Especializada - Mecánica Ind.	[Edit] [Delete]
MITA LEIVA CRISTOBAL DANIEL	Cuarto	A	Técnica Tecnológica General	[Edit] [Delete]
MITA LEIVA CRISTOBAL DANIEL	Cuarto	B	Técnica Tecnológica General	[Edit] [Delete]

**Fuente:** Elaboración propia

**Figura 3.38.** Panel principal del Maestro

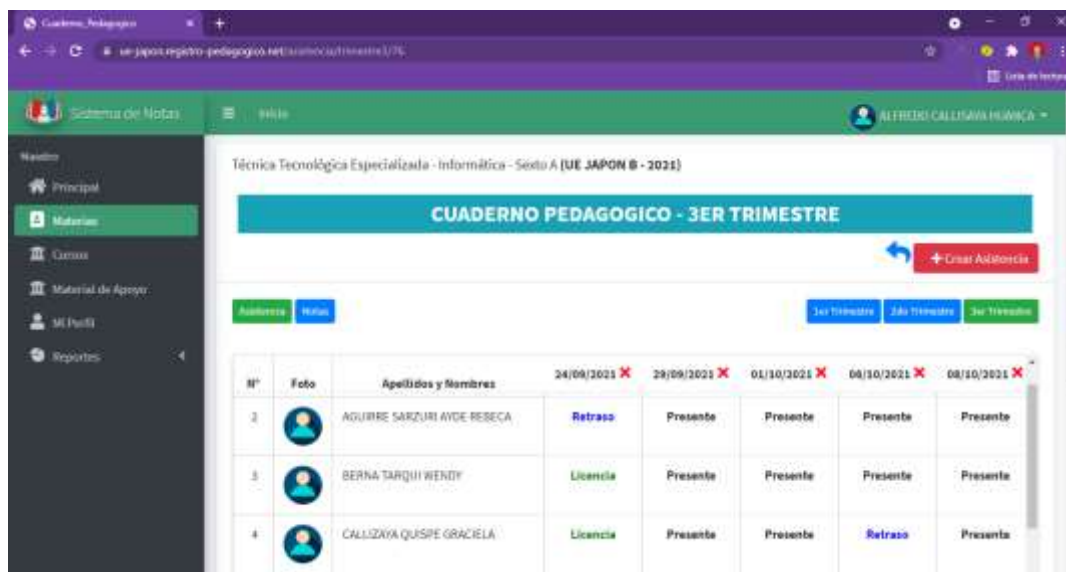


**Fuente:** Elaboración propia



En el módulo de asistencia del panel del maestro gestiona las asistencias, faltas, licencias y retrasos de acuerdo a las fechas registradas.

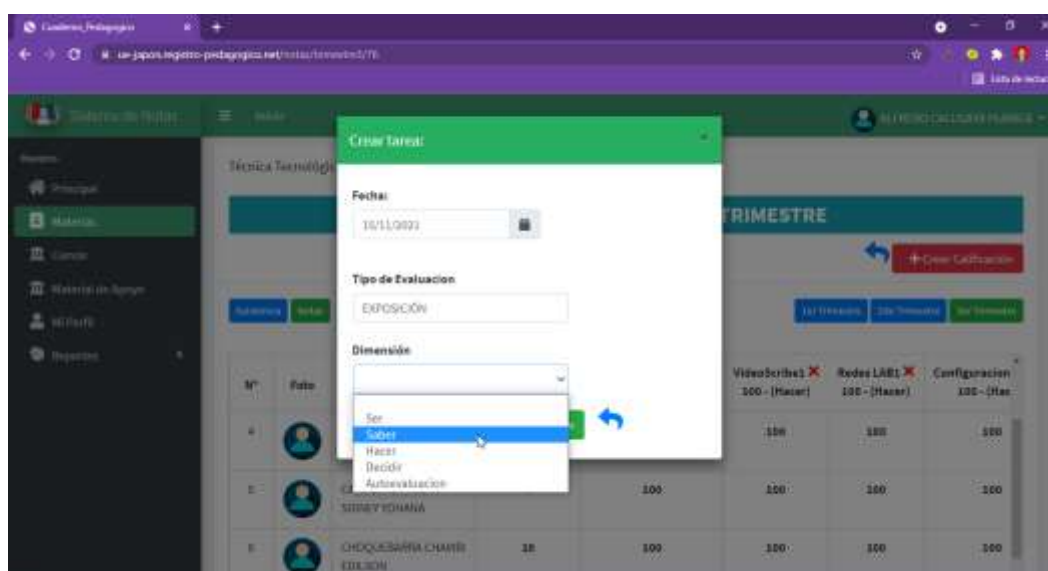
**Figura 3.39. Gestión de Asistencia**



**Fuente:** Elaboración propia

En el módulo de Notas del panel del maestro gestiona las actividades pedagógicas cotidianas bajo las dimensiones del ser, saber, hacer y decidir, según reglamento de evaluación de la gestión educativa 2021.

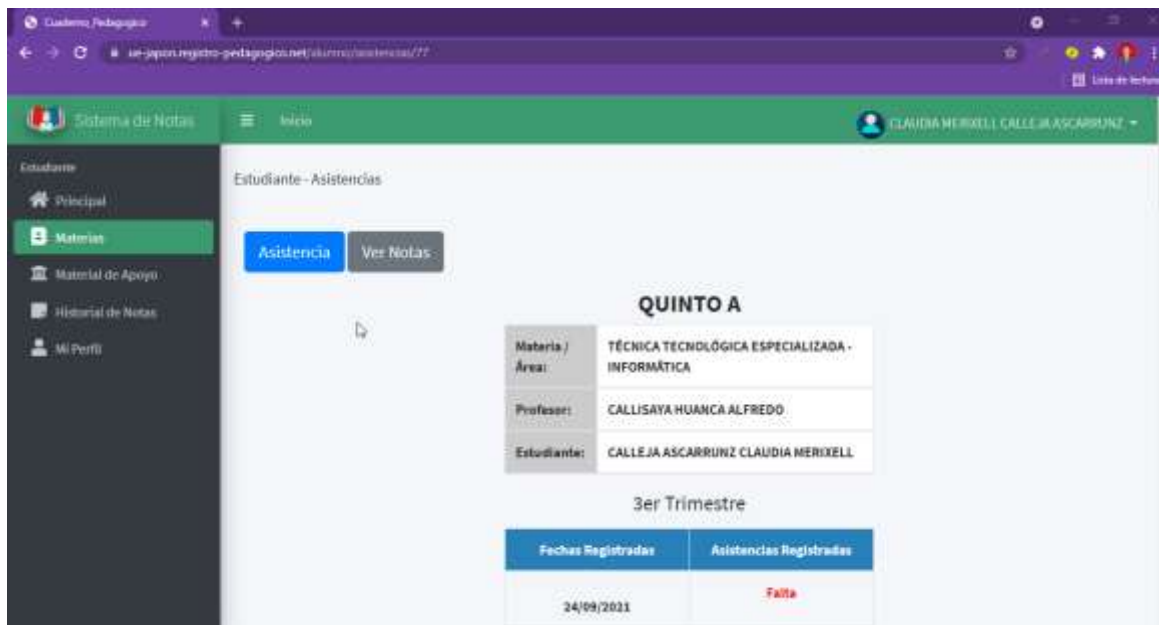
**Figura 3.40. Gestión de Notas**



**Fuente:** Elaboración propia

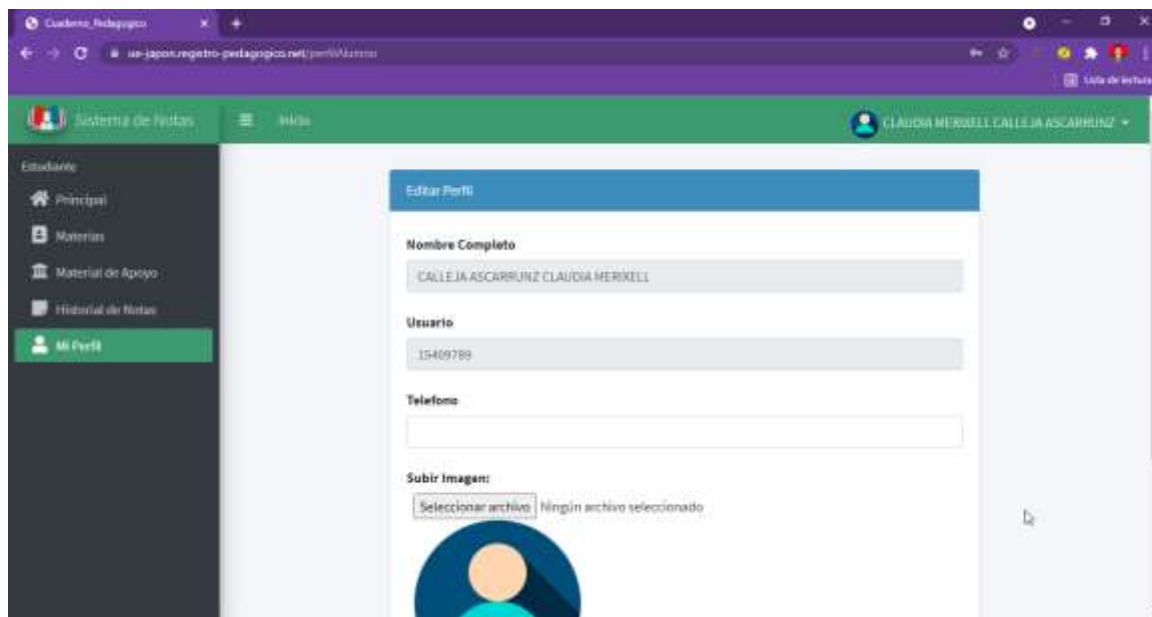
En el panel del estudiante realiza el seguimiento de las actividades registradas por los maestros asistencias y las calificaciones de las actividades cotidianas.

**Figura 3.41. Gestión de Notas**



**Fuente:** Elaboración propia

**Figura 3.42. Perfil del estudiante**



**Fuente:** Elaboración propia

## **CAPÍTULO IV**

---

### **ANÁLISIS DE CALIDAD Y COSTOS**

## CAPITULO IV

### 4. ANÁLISIS DE CALIDAD Y COSTOS

#### 4.1. MÉTRICAS DE CALIDAD

##### 4.1.1. Técnica ISO 9126

El objetivo principal de calidad de software es alcanzar la calidad necesaria para satisfacer las necesidades del cliente. Se evalúan dos ámbitos: el producto final y los procesos.

La calidad según esta norma web ISO 9126 puede ser perdida de acuerdo a los factores:

- Funcionalidad
- Confiabilidad
- Mantenibilidad
- Portabilidad
- Usabilidad

##### 4.1.1.1. Funcionalidad

El Punto Función es una métrica orientada a la función del software y del proceso por el cual se desarrolla. Se centra en la funcionalidad o utilidad del programa, los puntos de función se calculan realizando una serie de actividades, comenzando por determinar los siguientes números:

- **Números de entrada de usuario:** Se cuenta cada entrada del usuario que proporcione al software diferentes datos orientados a la aplicación.
- **Número de salida del usuario:** En este contexto las salidas se refieren a informes, pantalla, mensajes de error, etc.
- **Números de peticiones al usuario:** Una petición está definida como una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida interactiva.

- **Número de archivos:** Se cuenta cada archivo maestro lógico (se refiere a un grupo lógico de datos que puede ser parte de una gran base de datos o un archivo independiente).
- **Número de interfaces externas:** se cuentan todas las interfaces legibles por la máquina, por ejemplo: archivos de datos, en cinta o discos que son utilizados para transmitir información a otro sistema. [Pressman, 2005]

De acuerdo a lo mencionado tenemos los siguientes resultados:

**Tabla 4.1.** Entrada de datos para el Cálculo de Funcionalidad

<b>Entradas de usuario</b>	<b>25</b>
<b>Salidas de usuario</b>	20
<b>Consultas de usuario</b>	15
<b>Número de archivos</b>	10
<b>Interfaces externas</b>	1

**Fuente:** Elaboración propia

Los puntos función se calculan con la ayuda de la anterior tabla considerando los factores de ponderación medio.

**Tabla 4.2.** Cuenta total con fator de ponderación medio

<b>Parámetros de medición</b>	<b>Cuenta</b>	<b>Factor de ponderación medio</b>			<b>Total</b>
<b>Entradas de usuario</b>	25	*	4	=	100
<b>Salidas de usuario</b>	20	*	5	=	100
<b>Consultas de usuario</b>	15	*	4	=	60
<b>Número de archivos</b>	10	*	10	=	100
<b>Interfaces externas</b>	1	*	7	=	7
<b>CUENTA TOTAL</b>					<b>367</b>

**Fuente:** Elaboración propia

La relación que permite calcular los puntos de función es la siguiente:

$$PF = Cuenta\ Total * (Grado\ de\ confiabilidad + tasa\ de\ error * \sum Fi )$$

Donde:

- **PF** = Medida de Funcionalidad.
- **Cuenta Total** = es la suma de valor de las entradas, salidas, peticiones, interfaces externas y archivos.
- **Grado de Confiabilidad** = Es la confiabilidad estimada del sistema.
- **Tasa de error** = Probabilidad subjetiva estimada del dominio de la información, este error estimado es de 1%.
- **Fi** = Son valores de ajuste de complejidad que toman los valores de la tabla 18 y que dan respuesta a las preguntas de la tabla 18.

**Tabla 4.3.** Cuenta total con factor de ponderación medio

ESCALA	IN IMPORTANCIA	INCREMENTAL	MODERADO	MEDIO	SIGNIFICATIVO	ESENCIAL
Factor	0	1	2	3	4	5
¿Requiere el sistema copias de seguridad y de información?						x
¿Existen funciones de procesos distribuidos?			x			
¿Se requiere comunicación de datos?					x	
¿El sistema web se ejecuta en el sistema operativo actual?	x					

¿Se actualiza los archivos de forma interactiva?			x			
¿Es complejo el procesamiento interno del sistema?			x			
¿Se ha diseñado el código para ser reutilizable?						X
¿Requiere el sistema copias de seguridad y recuperación fiable?						X
<b>Total:</b>	<b>25</b>					

**Fuente:** Elaboración propia

A continuación, reemplazamos valores y calculamos el valor de PF:

$$PF = Cuenta\ Total * \left( Grado\ de\ confiabilidad + tasa\ de\ error * \sum Fi \right)$$

$$PF = 367 * (0.40 + 0.01 * 25)$$

$$PF = 238.55$$

Considerando el máximo valor de ajuste de complejidad  $\sum Fi = 40$ .

$$PF = 367 * (0.40 + 0.01 * 40)$$

$$PF = 293.6$$

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

$$Funcionalidad = \frac{238.55}{293.6} \times 100$$

$$Funcionalidad = 81.25$$

Se concluye que la funcionalidad o utilidad del sistema es del 81%. lo que significa que el software se desenvuelve satisfactoriamente

#### 4.1.1.2. Confiabilidad

La confiabilidad es la capacidad del software de mantener su nivel de performance o rendimiento, bajo las condiciones establecidas por un periodo de tiempo.

La confiabilidad del Software se la mide con la siguiente ecuación:

$$R(t) = (\text{Funcionalidad}) * e^{-\lambda t}$$

Dónde:

**R(t)** = Confiabilidad del Sistema

**Funcionalidad** = 0.81

$\lambda$  = 0.01 (es decir 1 error en cada 6 ejecuciones)

t = 12 meses

Hallamos la confiabilidad del sistema:

$$R(12) = 0.81 * e^{-\frac{1}{6} * 12}$$

$$R(t) = 0.11$$

Reemplazando en la fórmula de no hallar una falla se tiene:

$$P(T > t) = 1 - R(t)$$

$$P(T > t) = 1 - 0.11$$

$$P(T > t) = 0.89$$

$$P(T > t) = 0.89 * 100$$

$$P(T > t) = 89\%$$

$$\text{CONFIABILIDAD} = 89\%$$

Se concluye que la confiabilidad del sistema es del 89%



#### 4.1.1.3. Mantenibilidad

Para el cálculo de esta mantenibilidad del sistema, es decir índice de madurez del software (IMS), se establecerá los cambios que ocurrieron con cada versión del producto, para lo cual se determina la siguiente fórmula para hallar en (IMS).

$$IMS = \frac{Mt - (Fc + Fa + Fe)}{Mt}$$

Donde:

**Mt:** número de módulos total de la versión actual.

**Fc:** número de módulos de la versión actual que se cambiaron.

**Fa:** número de módulos de la versión actual que se añadieron.

**Fe:** número de módulos de la versión anterior que se eliminaron en la versión actual.

En el sistema se obtuvieron los siguientes valores como muestra la tabla 19 para la información requerida para el IMS.

**Tabla 4.4.** Información Requerida para el IMS

Información	Valores obtenidos
<b>Mt</b>	8
<b>Fc</b>	1
<b>Fa</b>	0
<b>Fe</b>	0

**Fuente:** Elaboración propia

Calculando el índice de madurez del software sustituyendo los valores:

$$IMS = \frac{8 - (1 + 0 + 0)}{8} = 0.875$$

Para la mejor interpretación del resultado multiplicamos por 100 para obtener en valor porcentual. *Mantenibilidad* = 0.875 \* 100

$$Mantenibilidad = 87.5\%$$

Se concluye que el sistema, tiene una mantenibilidad de 88%, rango satisfactorio.

#### 4.1.1.4. Portabilidad

La portabilidad se refiere al esfuerzo necesario para transferir el sistema de un entorno ya sea de hardware y/o software a otro, es una característica deseable de todo software.

La portabilidad tiene la siguiente fórmula:

$$P = \left[ 1 - \left( \frac{EP}{EI} \right) \right]$$

Dónde:

**P** = Portabilidad

**EP** = Esfuerzo para portar

**EI** = Esfuerzo para implementar

Consideremos que el esfuerzo para portar y para implementar es de 10 y 60 respectivamente de un rango entre el 1-100.

Entonces reemplazando valores a la ecuación tenemos:

$$P = \left[ 1 - \left( \frac{10}{60} \right) \right] = 0.83$$

$$P = 0.83 * 100$$

$$\text{Portabilidad} = 80\%$$

#### 4.1.1.5. Usabilidad

La usabilidad o facilidad de uso (FU), se calcula de la siguiente con la siguiente ecuación:

$$FU = \left[ \frac{\frac{\sum xi}{n} * 100}{n} \right]$$

En la tabla se calcula  $xi$  y  $\sum xi$  utilizando la escala de evaluaciones:

**Tabla 4.5.** Encuestas de usabilidad del sistema

#	Preguntas	Evaluación (xi)
1.	¿El sistema satisface los requerimientos de manejo de información?	5
2.	¿Las salidas del sistema están de acuerdo a sus requerimientos?	4
3.	¿Cómo considera los formularios que elabora el sistema?	4
4.	¿El sistema cumple con los requerimientos?	4
5.	Considera usted que es una herramienta útil para la Unidad Educativa	5
<i>Total</i> $\sum xi$		22

**Fuente:** Elaboración propia

Calculando FU:

$$FU = \left[ \frac{\frac{22}{5} * 100}{5} \right] = 88\%$$

Se concluye que la facilidad de uso es del 88%

#### 4.1.2. Resultados

El factor de calidad total está directamente relacionado con el grado de satisfacción con el usuario que ingresa al sistema denominado Cuaderno Pedagógico.

**Tabla 4.6.** Resultados de usabilidad del sistema

Características	Resultados
<b>Funcionalidad</b>	81%
<b>Confiabilidad</b>	89%
<b>Mantenibilidad</b>	87.5%
<b>Portabilidad</b>	80%
<b>Usabilidad</b>	88%
<b>Evaluación de Calidad Total</b>	<b>85.1%</b>

**Fuente:** Elaboración propia

Finalmente se concluye que el nivel de aceptación está dentro del rango aceptable.

## 4.2. SEGURIDAD DE SOFTWARE

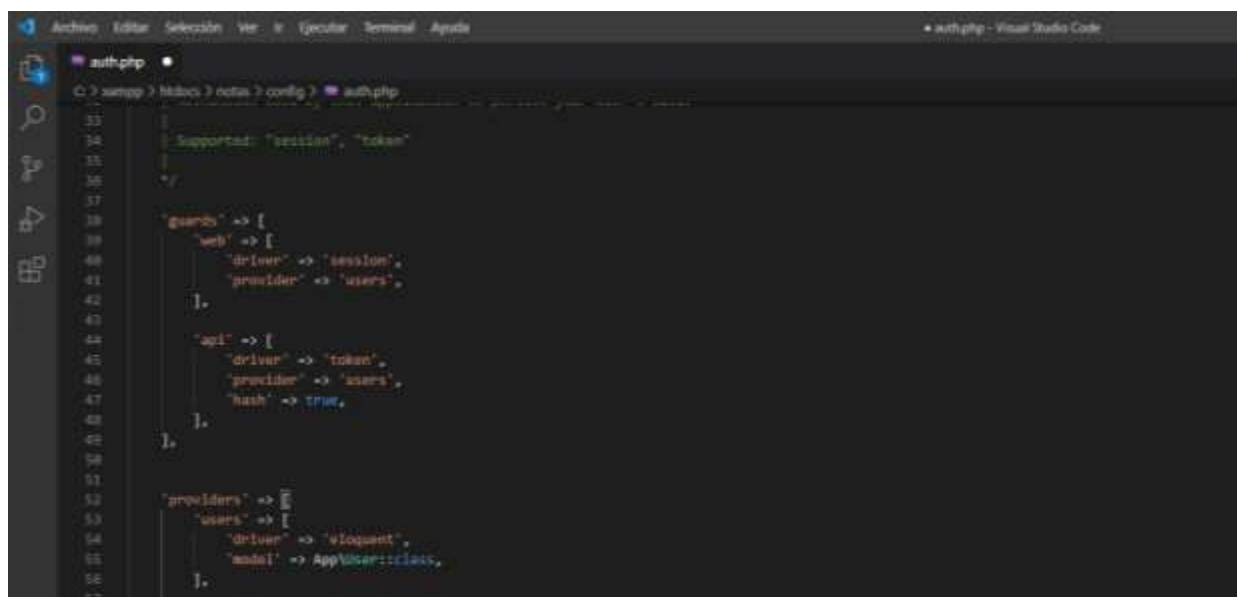
En la documentación oficial de laravel ya nos facilita el tema de seguridad de software y recomienda que se utilice su propia aplicación de testeo de software el cual hace que el trabajo de desarrollo y testeo sea más fácil de manejar.

### 4.2.1. Autenticación

Laravel hace que la implementación de la autenticación sea muy simple. El archivo de configuración de autenticación se encuentra en config/auth.php, que contiene varias opciones bien documentadas para modificar el comportamiento de los servicios de autenticación.

En esencia, las instalaciones de autenticación de Laravel están compuestas por "guardias" y "proveedores". Los guardias definen cómo se autentican los usuarios para cada solicitud. Por ejemplo, Laravel se envía con una sesión protectora que mantiene el estado mediante el almacenamiento de sesiones y las cookies.

**Figura 4.1.** El archivo de configuración de autenticación

The image shows a screenshot of the Visual Studio Code editor displaying the config/auth.php file. The code is written in PHP and defines the authentication configuration. It includes sections for 'supported' authentication methods, 'guards' (web and api), and 'providers' (users). The 'web' guard uses the 'session' driver and 'users' provider. The 'api' guard uses the 'token' driver and 'users' provider, with 'hash' set to true. The 'users' provider uses the 'eloquent' driver and the 'App\User::class' model.

```
33     'supported' => ['session', 'token']
34
35     // ...
36
37
38     'guards' => [
39         'web' => [
40             'driver' => 'session',
41             'provider' => 'users',
42         ],
43
44         'api' => [
45             'driver' => 'token',
46             'provider' => 'users',
47             'hash' => true,
48         ],
49     ],
50
51
52     'providers' => [
53         'users' => [
54             'driver' => 'eloquent',
55             'model' => App\User::class,
56         ],
57     ],
58
59
60
61
62
63
64
65
66
67
```

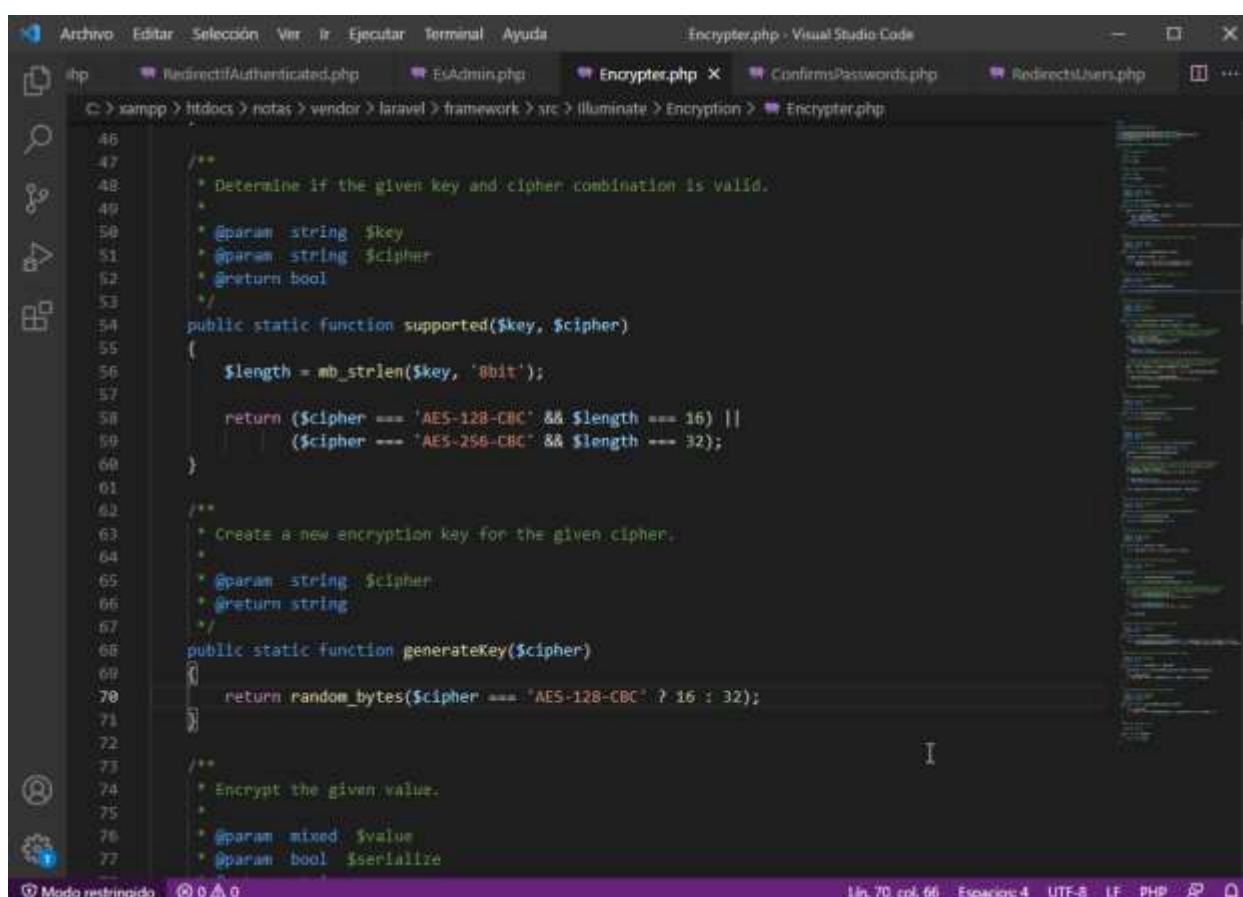
**Fuente:** Elaboración propia

Los proveedores definen cómo se recuperan los usuarios de su almacenamiento persistente. Laravel se envía con soporte para recuperar usuarios usando Eloquent y el generador de consultas de base de datos. Sin embargo, puede definir proveedores adicionales según sea necesario para su aplicación.

#### 4.2.2. Cifrado

El cifrador de Laravel usa OpenSSL para proporcionar cifrado AES-256 y AES-128. Se le recomienda las funciones de cifrado integradas de Laravel. Todos los valores cifrados de Laravel se firman utilizando un código de autenticación de mensajes (MAC) para que su valor subyacente no se pueda modificar una vez cifrados.

Figura 4.2. Cifrado



```
46
47
48  /**
49   * Determine if the given key and cipher combination is valid.
50   *
51   * @param string $key
52   * @param string $cipher
53   * @return bool
54   */
55  public static function supported($key, $cipher)
56  {
57      $length = mb_strlen($key, '8bit');
58
59      return ($cipher === 'AES-128-CBC' && $length === 16) ||
60             ($cipher === 'AES-256-CBC' && $length === 32);
61  }
62
63  /**
64   * Create a new encryption key for the given cipher.
65   *
66   * @param string $cipher
67   * @return string
68   */
69  public static function generateKey($cipher)
70  {
71      return random_bytes($cipher === 'AES-128-CBC' ? 16 : 32);
72  }
73
74  /**
75   * Encrypt the given value.
76   *
77   * @param mixed $value
78   * @param bool $serialize
```

Fuente: Elaboración propia

### 4.2.3. Hash

La Hash fachada de Laravel proporciona hash Bcrypt y Argon2 seguro para almacenar contraseñas de usuario. Si está utilizando las clases integradas LoginControllery RegisterController que se incluyen con su aplicación Laravel, usarán Bcrypt para el registro y la autenticación de forma predeterminada.

**Bcrypt** es una excelente opción para hacer hash de contraseñas porque su "factor de trabajo" es ajustable, lo que significa que el tiempo que se tarda en generar un hash se puede aumentar a medida que aumenta la potencia del hardware.

**Figura 4.2.** Bcrypt

```
'bcrypt' => [  
    'rounds' => env('BCRYPT_ROUNDS', 10),  
],
```

**Fuente:** Elaboración propia

El algoritmo ARGON2, el make método permite administrar el factor de trabajo del algoritmo mediante el memory, timey threadsopciones; sin embargo, los valores predeterminados son aceptables para la mayoría de las aplicaciones.

**Figura 4.3.** Argón

```
'argon' => [  
    'memory' => 1024,  
    'threads' => 2,  
    'time' => 2,  
],
```

**Fuente:** Elaboración propia

### 4.2.4. Restablecimiento de contraseñas

Laravel proporciona métodos convenientes para enviar recordatorios de contraseñas y realizar restablecimientos de contraseñas.

**Figura 4.4.** Reset password

```
class ResetPasswordController extends Controller
{
    use ResetsPasswords;

    @var string

    protected $redirectTo = RouteServiceProvider::HOME;
}
```

**Fuente:** Elaboración propia

### 4.3. COSTO DEL SISTEMA

La técnica de Análisis de Costo y Beneficio, tiene como objetivo fundamental proporcionar una medida de la rentabilidad de un proyecto, mediante la comparación de los costos previstos con los beneficios esperados en la realización del mismo. Esta técnica se debe utilizar al comparar proyectos para la toma de decisiones.

Para calcular el costo del proyecto se lo realizará haciendo uso del modelo COCOMO II. El modelo COCOMO tiene una jerarquía de modelos como ser: básico, intermedio y avanzado, la cual se aplica a tres diferentes tipos de software.

#### 4.3.1. COCOMO II

Para el sistema se utilizó el modelo COCOMO en su nivel básico semi-acoplado.

Aplicando de las fórmulas básicas de esfuerzo, tiempo calendario y personal requerido.

Las 3 ecuaciones de COCOMO básico tiene la siguiente forma:

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

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

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

Donde:

**E:** Esfuerzo requerido por el proyecto, en meses.

**D:** Tiempo requerido por el proyecto, en meses.

**P:** Número de personas requeridas por el proyecto.

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

**KLDC:** Cantidad de líneas de código, en miles.

**Tabla 4.7.** Coeficiente a y c y los exponentes b y d

<b>Proyecto de Software</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
Orgánico	2.4	1.05	2.2	0.38
Semi-Acoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.20	2.5	0.32

**Fuente:** (QSM, 2020).

#### 4.3.2. Tiempo del desarrollo del Software

Para el cálculo del desarrollo del software se tendrá como partida el punto función no ajustado valor ya encontrado en el capítulo anterior:

$$PF = 293.6$$

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



**Tabla 4.8.** Factor LCD/PF de Lenguajes de Programación

LENGUAJE	FACTOR LDC/PFF
Java	53
JavaScript	30
Visual Basic	46
ASP	36
PHP	31
C	150

**Fuente:** (QSM, 2020).

Calculando las líneas de código del sistema utilizando la siguiente ecuación tenemos:

$$LDC = PF * FACTOR \frac{LDC}{PF}$$

$$LDC = 293.6 * 31$$

$$LDC = 9101.6$$

Para convertirlo a Kilo Líneas de Código (KLDC), dividimos LDC entre 1000.

Calculando KLCD tenemos:

$$KLDC = \frac{LDC}{1000}$$

$$KLDC = \frac{9,101.6}{1000} = 9.10$$

A continuación, realizaremos el cálculo del esfuerzo necesario para la programación del sistema. La ecuación que nos ayudará a hallar el esfuerzo es la siguiente:

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

Reemplazando Valores tenemos:

$$E = 2.4 * (9.10)^{1.5}$$

$$E = 24.39 \text{ Personas/mes}$$

Calculando Tiempo requerido D con  $c = 2.2$  y  $d = 0.38$  tenemos:

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

$$T = 2.2 * (24.39)^{0.38}$$

$$T = 7.4 \cong 7 \text{ meses}$$

El proyecto aproximadamente tiene un tiempo de duración para su elaboración de 7 meses.

### 4.3.3. Costo del desarrollo del Software

La siguiente ecuación es para calcular el personal requerido. En este caso el número de programadores para el desarrollo es:

$$P = \frac{E}{T}, \text{ en ersonas}$$

$$P = \frac{24.39}{7.4}, \text{ en ersonas}$$

$$P = 3.29 \cong 3 \text{ programadores}$$

El salario de un programador puede oscilar entre 300 \$, cifra que será tomada en cuenta para la estimación siguiente:

$$\text{Costo del software por persona} = P * \text{salario de un programador}$$

$$\text{Costo del software por persona} = 3 * 300 \$$$

$$\text{Costo del software por persona} = 900 \$$$

Convirtiendo en bolivianos tenemos:

$$\text{Costo total de desarrollo} = \text{Costo del software desarrollado} * 6.97$$

$$\text{Costo total de desarrollo} = 900 * 6.97$$

$$\text{Costo total de desarrollo del software} = 6,273 \text{ Bolivianos}$$

#### 4.3.4. Costo de Elaboración del Proyecto

Los costos de elaboración del proyecto se refieren a los costos del estudio del sistema, en la etapa de análisis y recopilación principalmente, estos costos se representan en la siguiente tabla.

**Tabla 4.9.** Costo de Elaboración del Proyecto

LENGUAJE	FACTOR LDC/PFF
Análisis y diseño de sistema	1100
Material de escritorio	500
Internet	200
Bibliografía	100
Otros	100
<b>TOTAL</b>	<b>2000</b>

**Fuente:** (Elaboración propia)

#### 4.3.5. Costo total de Proyecto

El costo es la sumatoria del costo del software desarrollado, costo de elaboración del proyecto, detallados en la siguiente tabla

**Tabla 4.10.** Costo total del proyecto

DETALLE	IMPORTE (Bs)
Costo del Software Desarrollado	6273
Costo de Elaboración del Proyecto	2000
<b>TOTAL</b>	<b>8273</b>

**Fuente:** (Elaboración propia)

Finalmente se concluye que el costo del sistema web es de 8273 bolivianos.

## **CAPÍTULO V**

---

### **CONCLUSIONES Y RECOMENDACIONES**

## **CAPITULO V**

### **5. CONCLUSIONES Y RECOMENDACIONES**

#### **5.1. CONCLUSIONES**

El objetivo general de este proyecto de grado, fue desarrollar “CUADERNO PEDAGÓGICO WEB COMO HERRAMIENTA DEL MAESTRO PARA EL SEGUIMIENTO DE ACTIVIDADES PEDAGÓGICAS” con la finalidad de innovar procesos educativos de los maestros de la Unidad Educativa Republica del “Japón B”, con la ejecución y puesta en marcha el proyecto web se concluyó satisfactoriamente alcanzando los objetivos específicos propuestos y cumpliendo los diferentes requerimientos de los maestros de la institución educativa como se describe a continuación:

- El objetivo general, se concluyó con el desarrollo e implementación del sistema web del Cuaderno Pedagógico Web de manera satisfactoria tanto para los usuarios del sistema como para la institución.
- El módulo del maestro se implementó de forma satisfactoria para la gestión de asistencias y registro de actividades pedagógicas.
- El cuaderno pedagógico Web genera los reportes de forma trimestral acorde a los requerimientos de la institución educativa.
- El desarrollo del sistema fue desarrollado bajo metodología ágil SCRUM.

#### **5.2. RECOMENDACIONES**

Si bien el sistema cumple con los requerimientos de la institución educativa, se recomienda lo siguiente:

- A los administradores del sistema web, se recomienda realizar backups de la base de datos periódicamente.
- A los maestros que son los actores principales, se recomienda cambiar la contraseña en el primer momento que ingresa al sistema para su propia seguridad.

## BIBLIOGRAFÍA

Joseph, S (1999) *Aprendiendo UML en 24 horas*. (1a ed.) Editorial División computación.

Pressman, R. (2010). *Ingeniería del software. Un enfoque práctico*. (7a ed.) McGraw-Hill.

Toores M. R. (2016) *Diseño web con HTML5 y CSS3*. Editora Macro EIRL

Cobo, Gómez, Pérez y Rocha, (2005) *PHP y MySQL Tecnologías para el desarrollo de aplicaciones web*. Ediciones de santos

Abud, M. (2012). *Calidad en la Industria del Software*. La Norma ISO-9126.

Ríos M. B. (2014) *Diagramas de flujo* (1a ed.) Innograf.

Llanque E.Q.(2004) *Diagramas de flujo estructurados* (1a ed) Imágenes S.R.L.

Eguíluz Pérez, J. (2008). *Introducción a CSS*. Creative Commons

Cahterine M. R.(2009) *Bases de datos*. McGRAW-HILL INTERAMERICANA EDITORES, S.A. de C.V.

Camps, Casillas, Costal, Ginesta, Escofet, Mora, (2005) *Base de Datos*. Eureka Media,SL

Abut F. M. (2012) *Calidad en la Industria del Software*. La Norma ISO-9126

Gómez, A., López, M., Migani, S., y Otazú, A. (2010). *Cocoma - un modelo de estimación de proyectos de software*.

Pressman, R. (2005) Ingeniería del Software, (6a Ed.), McGraw-Hill interamericana editores.

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

Kendall, K. (2005). *Análisis y Diseño de Sistemas*. Earson Education.

<http://www.nacionmulticultural.unam.mx/empresasindigenas/docs/2094.pdf>

Trigás Gallego, M. (2012). *Metodología Scrum*. España: Universitat Oberta de

Catalunya.[http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigas TFC0612memoria.pdf](http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigas_TFC0612memoria.pdf)

Peralta, A. (2003). *Metodología SCRUM*. Uruguay: Universidad ORT.

<https://fi.ort.edu.uy/innovaportal/file/2021/1/scrum.pdf>

Documentación y guía de desarrollo oficial de laravel (2020)

<https://laravel.com/docs/7.x>

Documentación de composer para PHP (2020)

<https://getcomposer.org/doc/>

Documentación de Código de Visual Studio (2020)

<https://code.visualstudio.com/docs>

Paquete de Instalaciones Xampp (2020)

<https://www.apachefriends.org/es/index.html>

## **ANEXOS**

---

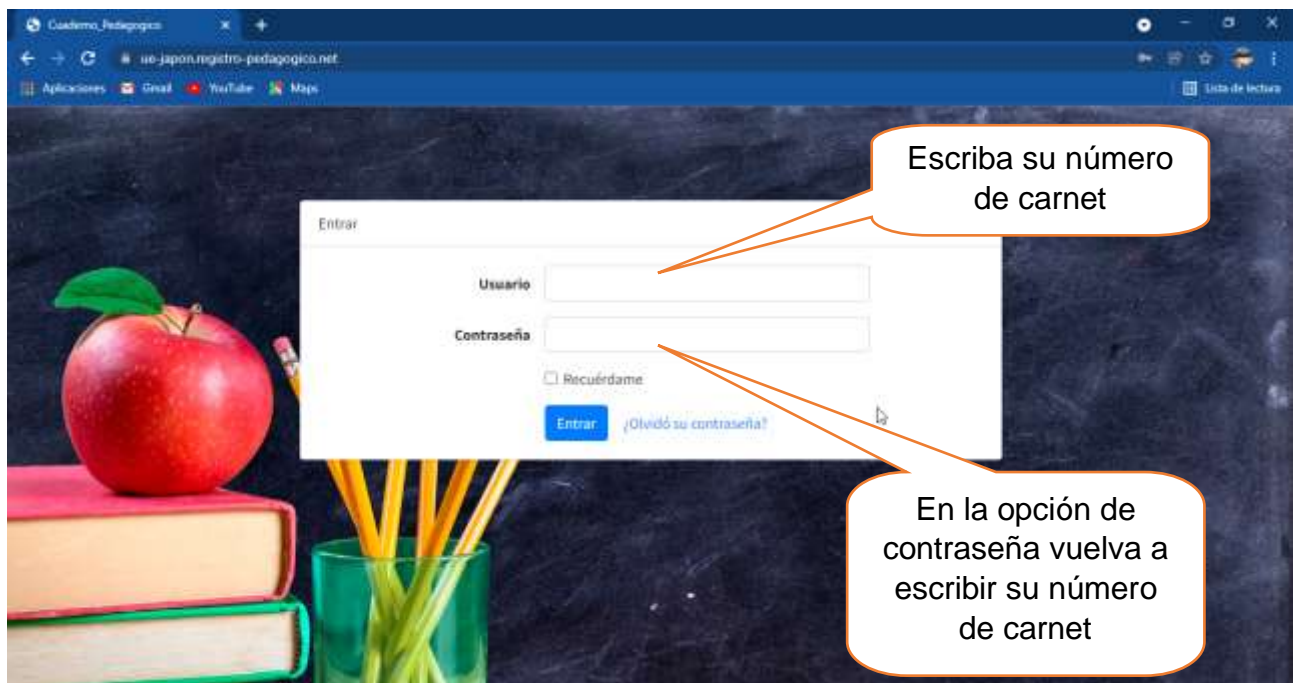


## MANUAL DE USUARIO

### AUTENTICACIÓN

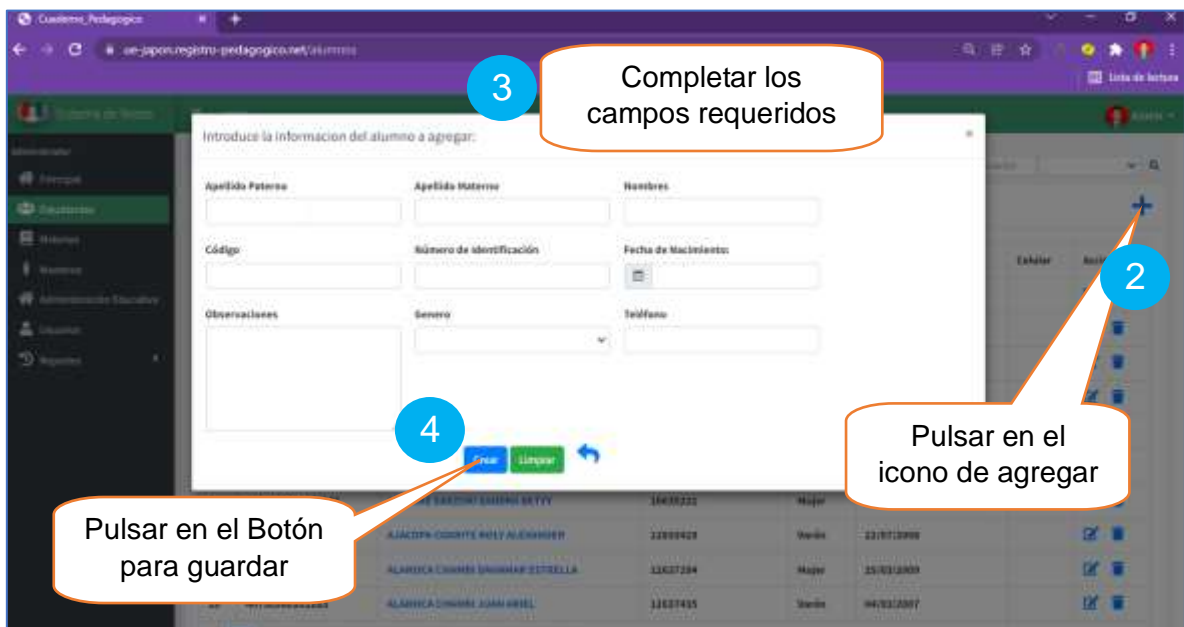
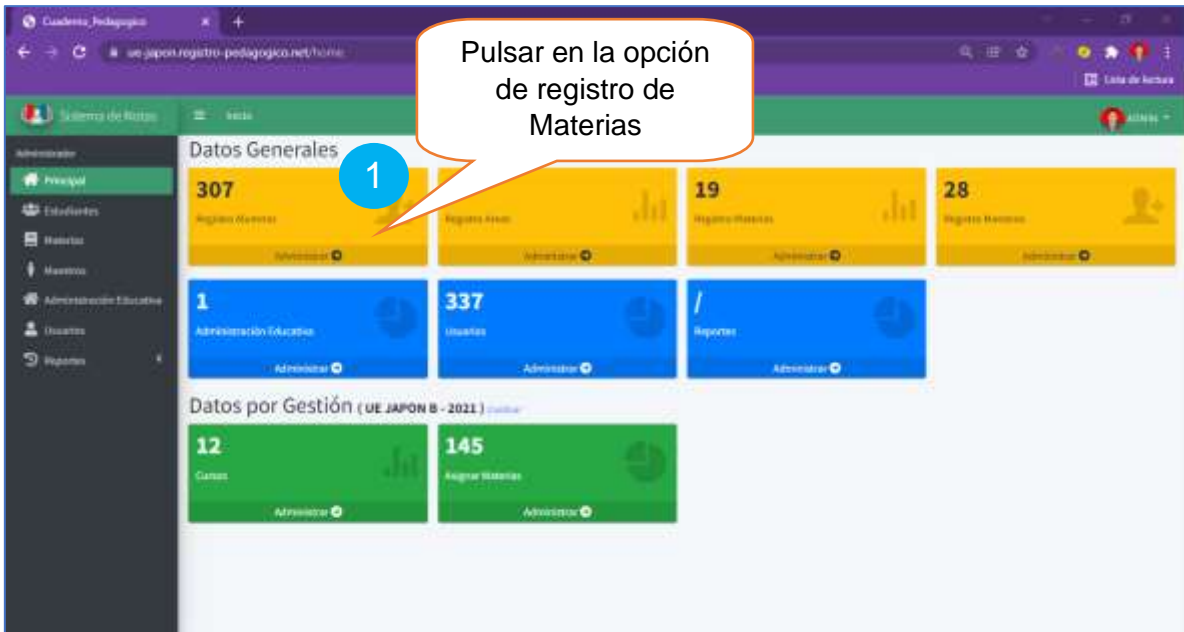
El acceso al sistema denominado Cuaderno Pedagógico Web se lo hace a través de cualquier navegador web mediante la siguiente dirección web.

<https://ue-japon.registro-pedagogico.net/>

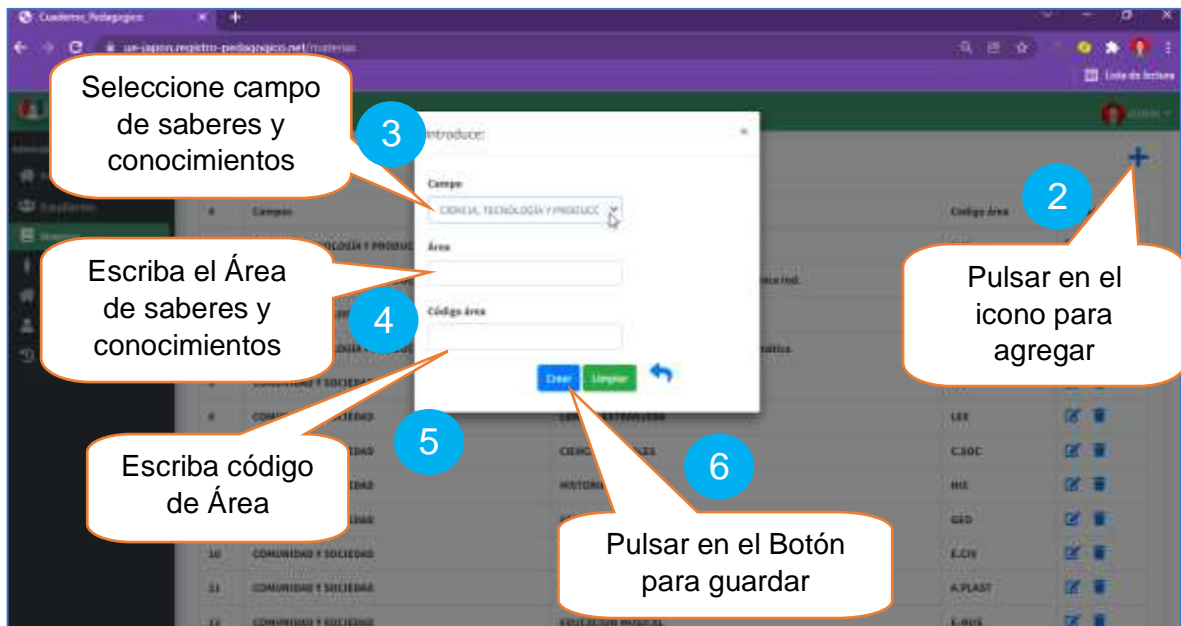
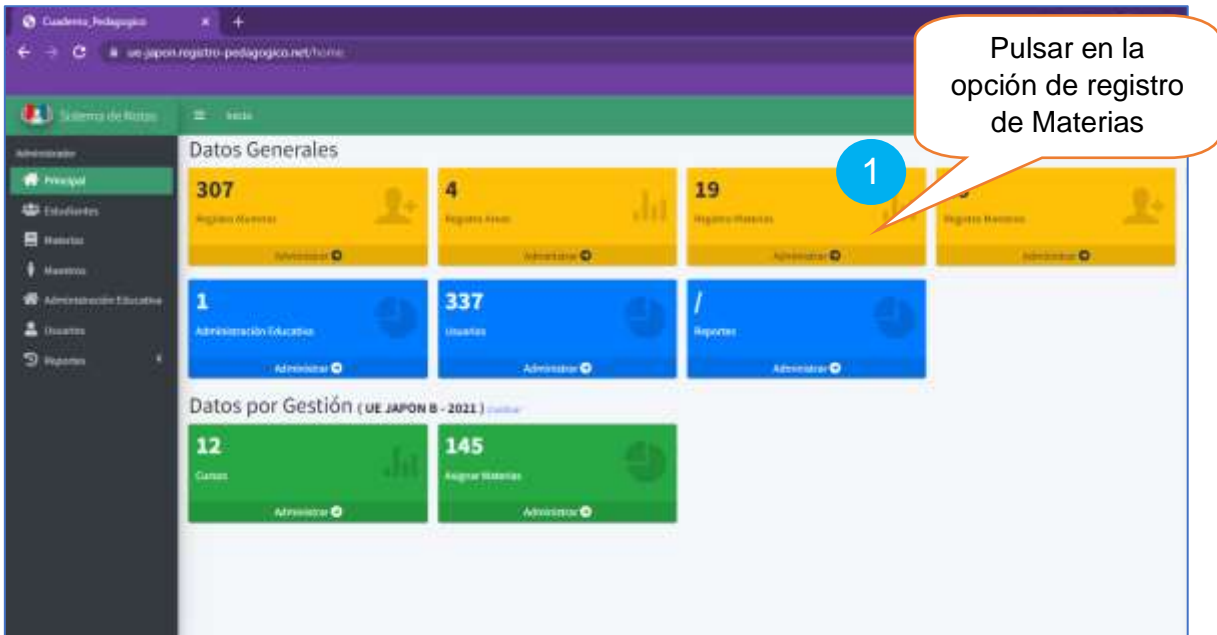


## ADMINISTRACIÓN EN EL PANEL DEL ADMINISTRADOR

### Registro de Estudiantes

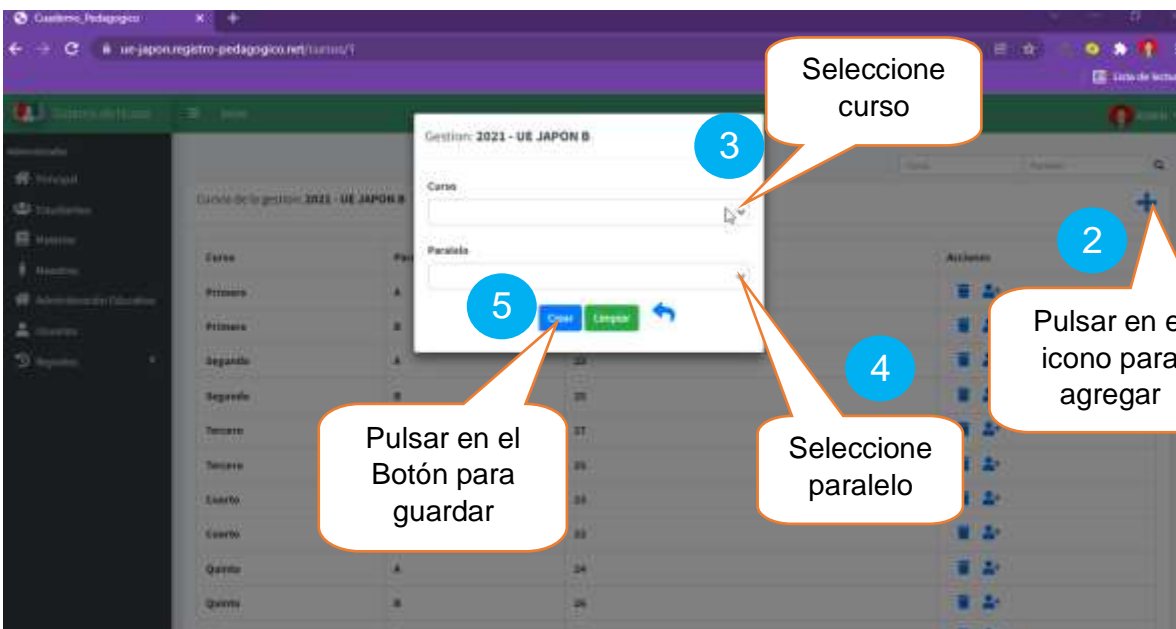
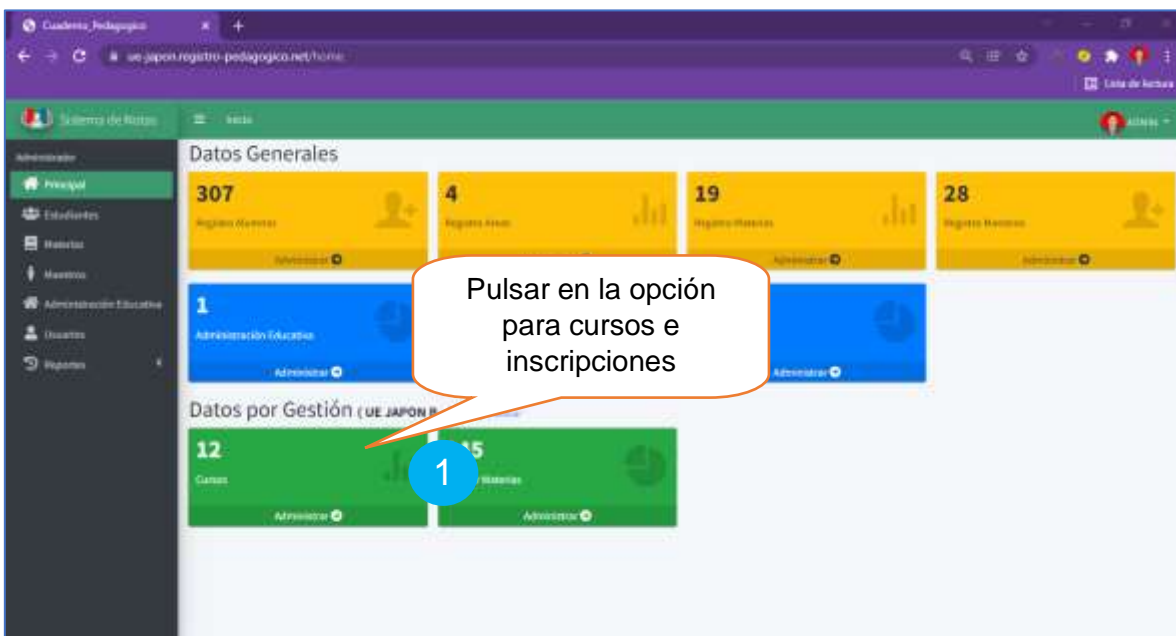


## Registro de Áreas



## Inscripción de estudiantes

### Primero: Crear cursos y paralelos



## Segundo: Inscribir estudiantes en los cursos y paralelos correspondientes

Pulsar en el Icono para inscribir

Curso	Paralelo	Estudiantes inscritos	Acciones
Primero	A	29	[Iconos]
Primero	B	29	[Iconos]
Segundo	A	23	[Iconos]
Segundo	B	25	[Iconos]
Tercero	A	37	[Iconos]
Tercero	B	25	[Iconos]
Cuarto	A	23	[Iconos]
Cuarto	B	22	[Iconos]
Quinto	A	34	[Iconos]

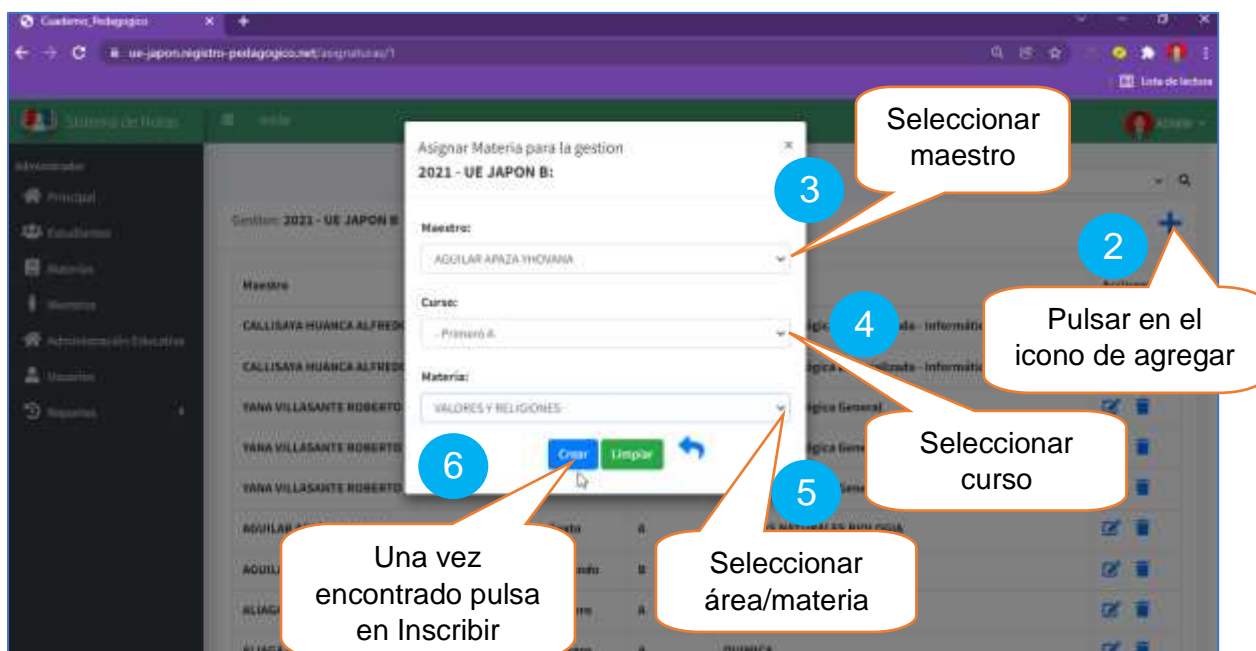
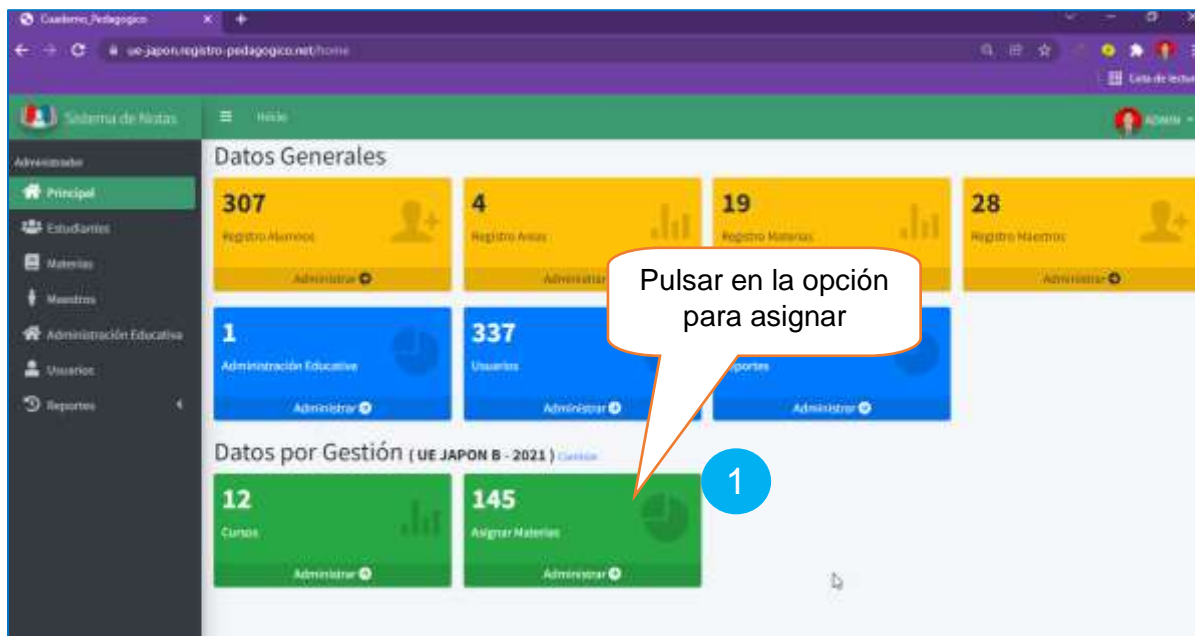
Pulsar en el icono de agregar

Buscar por C.I. o apellido paterno

Una vez encontrado pulsa en Inscribir

Nombre	Fecha de Nacimiento
ABUQUIN ROSAS JOEL ROBERTO	22/07/2000
CALA TOLA BLADIBIR JHONATAN	17/03/2000
CANAVIRI FLORES SHADIRA	18/05/2000
CARRASCO VARGAS JOEL ROBERTO	15/01/2000
CARRASCO VARGAS JOEL ROBERTO	17/04/2000
CARRASCO VARGAS JOEL ROBERTO	28/01/2020
CARRASCO VARGAS JOEL ROBERTO	22/01/2000
CARRASCO VARGAS JOEL ROBERTO	11/11/2000
CARRASCO VARGAS JOEL ROBERTO	18/01/2000
CARRASCO VARGAS JOEL ROBERTO	11/08/2000

## Asignación de maestros



## Reportes por trimestre y área

Pr promedio de Asistencia por Asignaturas

Nombres y Apellidos	CIENCIA, TECNOLOGÍA Y PRODUCCIÓN		COSMOS Y PENSAMIENTO		COMUNIDAD Y SOCIEDAD			
	Técnica Tecnológica General	MATEMÁTICAS	PSICOLOGÍA	VALORES Y RELIGIONES	CIENCIAS SOCIALES	EDUCACIÓN FÍSICA Y DEPORTES	COMUNICACIÓN Y LENGUAJES	LENGUA EXTRANJERA
1. ALONSO CORTE ROY ALEXANDER	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
2. ALAY CONDOIS DIANA ROSITA	50.00%	50.00%	100.00%	50.00%	50.00%	50.00%	50.00%	50.00%
3. ARGOLLO ZESARFA ALEXIS FERNANDO	0.00%	75.00%	75.00%	100.00%	0.00%	100.00%	10.00%	0.00%
4. ARQUILLA ROJAS JOEL ABRAHAM	75.00%	100.00%	100.00%	0.00%	100.00%	75.00%	75.00%	75.00%
5. CALA TOLA BLADIMIR JHONNY	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

## Reporte de Estudiantes destacados

Filtrar alumnos destacados según la siguiente materia: TÉCNICA TECNOLÓGICA GENERAL

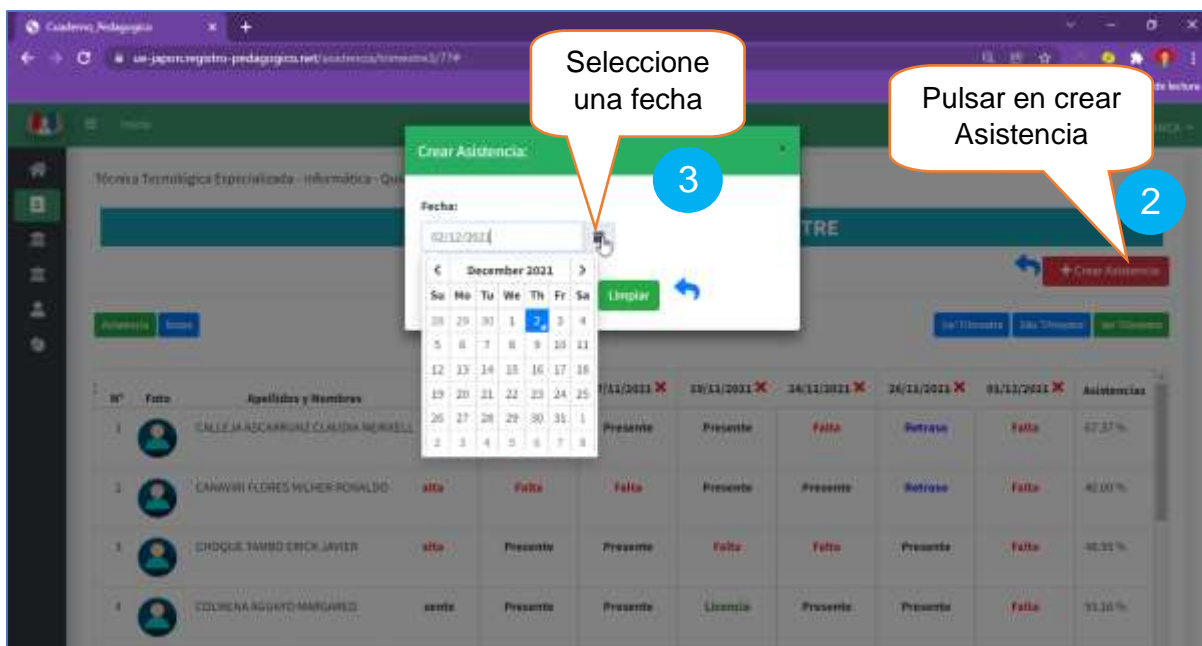
Rango	Nombre	Puntaje
1er Lugar	TAYCARR VARELLA SANDY	50.00 puntos de Promedio
2do Lugar	CARLO GOMEZ JOHEL ROYDI	41.00 puntos de Promedio
3er Lugar		33.00 puntos de Promedio



## ADMINISTRACIÓN EN PANEL DEL MAESTRO



### Registro de Asistencia





## Registro de Calificaciones

1. Selecciona Notas

2. Pulsa el botón para crear

3. Seleccione una fecha

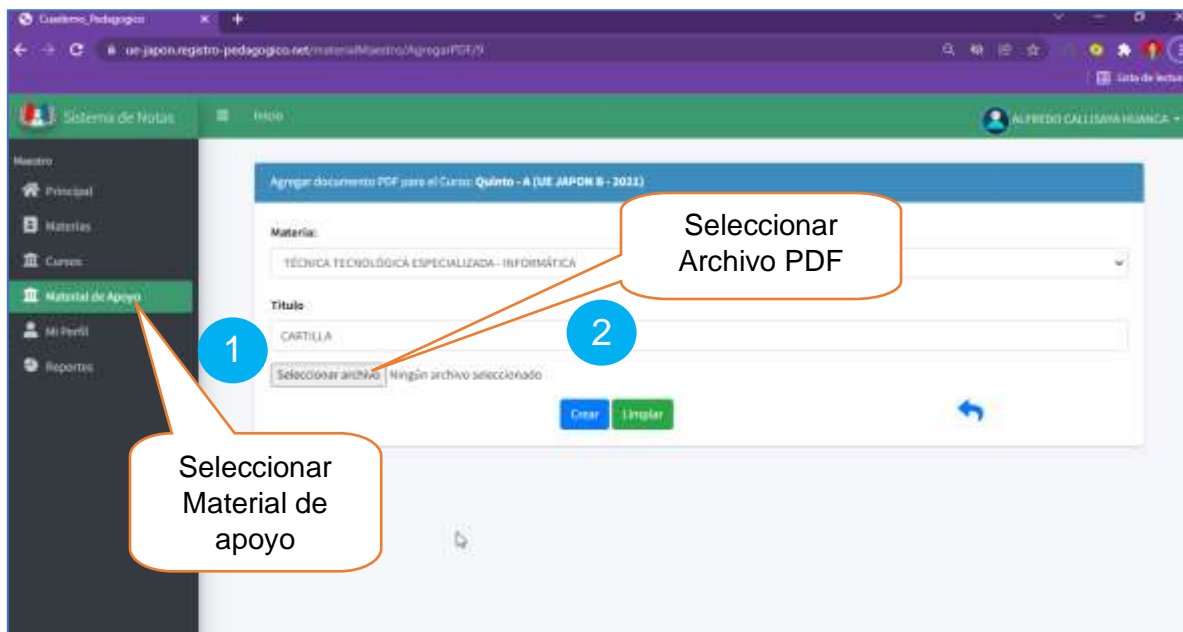
4. Escriba tipo de evaluación

5. Seleccione una dimensión

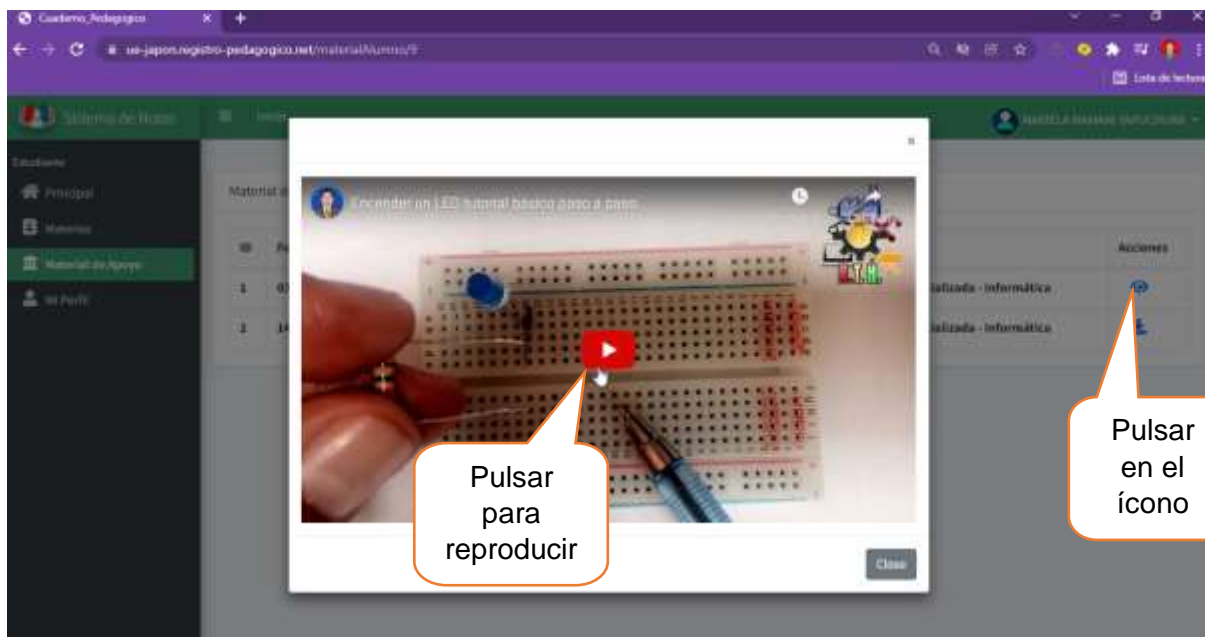
6. Asignar una Calificación

Mº	Foto	Apellidos y Nombres	Exposición	NOTA FINAL Decidir	Asistencia	AutoEvaluación	NOTA FINAL
13		MAMANI JULIAN ALICIA DELIA			95	90	75
14		MAMANI YAPUCHURA MARELA	94	100	94		38
15		MAMANI MIRANDA LUIS MIGUEL	94	100			47
16		MARCA ALBERTO MARY LUIZ	0	0	56	51	25
17		MENDOZA NAVIA SALOME	94	98	72	65	71
18		PAREDES JIMENEZ YUDITH	100	100			66

## Añadir material de apoyo



## Agregar video de YouTube



# Reportes

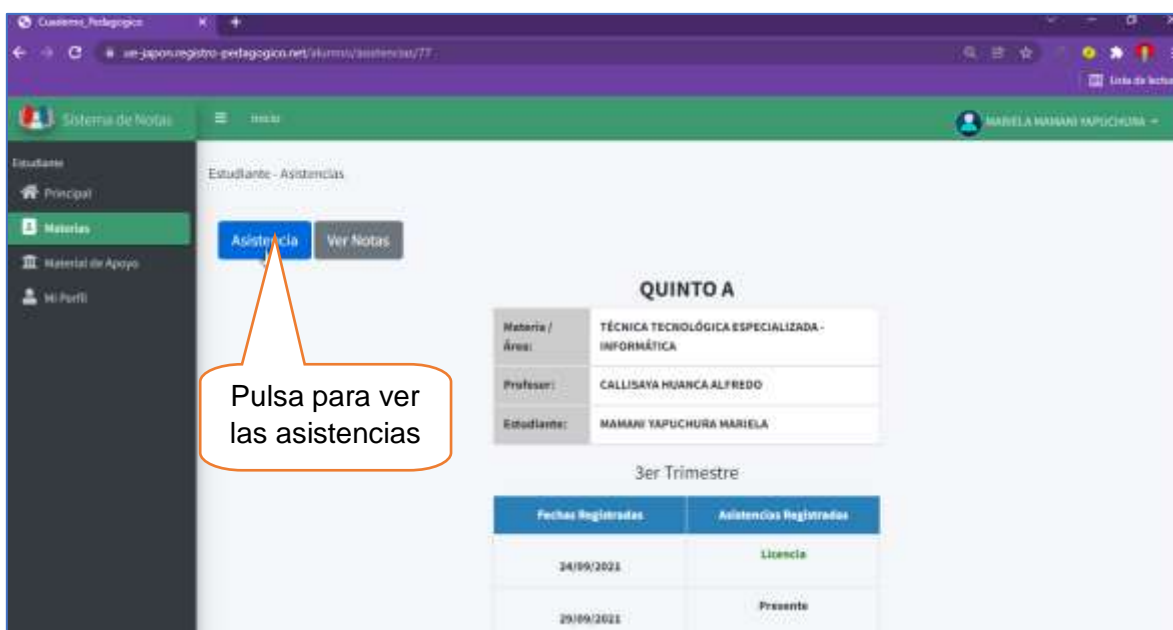
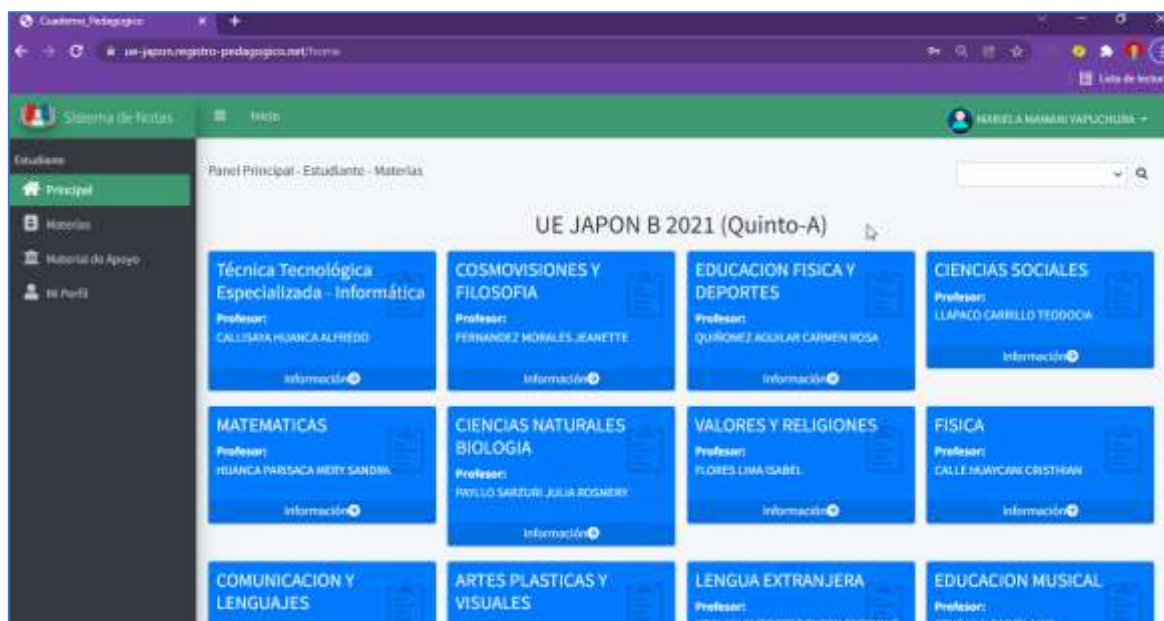
## Reporte trimestral de asistencia

N	Apellidos y nombres	03-11-2021	05-11-2021	10-11-2021	17-11-2021	19-11-2021	24-11-2021	26-11-2021	01-12-2021	Presencias	Licencias	Faltas	Retrasos
		Presente	Presente	Presente	Presente	Presente	Presente	Presente	Presente				
1	SALLUCO NELSON CRISTIAN	Presente	Presente	Presente	Presente	Presente	Presente	Presente	Presente	16	1	0	1
2	AGUIRRE SARZURI AYDE REBECA	Retraso	Falta	Presente	Presente	Presente	Presente	Presente	Presente	15	0	1	3
3	BERNA TARQUI WENDY	Presente	Presente	Presente	Presente	Presente	Presente	Presente	Presente	17	1	0	1
7	CALLIZAYA QUIISPE GRACIELA	Presente	Presente	Presente	Presente	Presente	Presente	Presente	Presente	16	1	0	2
9	CARDENAS FRANCO SIDNEY YOHANA	Presente	Presente	Presente	Presente	Falta	Presente	Presente	Presente	13	0	5	1
11	CHOQUEBARRA CHAMBI EDILSON	Retraso	Presente	Presente	Retraso	Falta	Presente	Presente	Falta	11	0	5	3

## Reporte trimestral de notas

N	Apellidos y nombres	1er Trimestre				2do Trimestre				3er Trimestre				PROMEDIO				
		Hacer	Decidir	Nota Ev	Nota Final	Ser	Saber	Hacer	Decidir	Nota Ev	Nota Final	Ser	Saber		Hacer	Decidir	Nota Ev	Nota Final
1	SALLUCO NELSON CRISTIAN	25	8	9	76	10	32	34	10	10	96	10	35	34	9	10	98.0	99.0
2	AGUIRRE SARZURI AYDE REBECA	30	11	7	81	9	18	30	10	11	71	9	34	29	9	8	80.0	78.0
3	BERNA TARQUI WENDY	31	11	11	83	10	32	34	10	10	96	10	34	34	11	10	96.0	92.0
4	CALLIZAYA QUIISPE GRACIELA	35	10	10	100	10	33	35	10	9	97	10	35	35	8	10	98.0	98.6
5	CARDENAS FRANCO SIDNEY YOHANA	10	4	11	35	1	9	0	0	3	13	7	15	32	8	7	69.0	38.7
6	CHOQUEBARRA CHAMBI EDILSON	25	7	10	77	8	29	22	9	7	66	8	27	27	9	0	71.0	73.5
7	CHURQUI LIMA DAVID SALOMON	35	5	10	52	5	10	16	0	7	38	8	32	33	10	8	91.0	89.7
8	CONDORI BLANCO IBETT ANACELI	20	7	8	66	0	0	0	10	0	10	2	19	19	5	7	53.0	42.1
9	CONDORI FRANCO VIVIANA SOLEDAD	25	10	10	80	10	32	35	10	9	96	10	35	35	11	9	97.0	95.2

## PANEL DEL ESTUDIANTE



**Sistema de Notas**

Inicio

MARIA MAMANI YAPUCHURU

Estudiante

- Principal
- Materias
- Material de Apoyo
- Mi Perfil

Asistencia Ver Notas

**QUINTO A**

Materia / Área: TÉCNICA TECNOLÓGICA ESPECIALIZADA - INFORMÁTICA

Profesor: CALLISAYA HUANCA ALFREDO

Estudiante: MAMANI YAPUCHURA MARIELA

3er Trimestre

Fecha	Actividades Calificadas	%
29/09/2021	Participacion	100
13/10/2021	Lab electronica1	95
20/10/2021	LAB CANVA	100
22/10/2021	LAB2 CANVA	100

Pulsa para ver Actividades Calificadas

**Sistema de Notas**

Inicio

MARIA MAMANI YAPUCHURU

Estudiante

- Principal
- Materias
- Material de Apoyo
- Mi Perfil

Material de Apoyo del curso: Quinto - A (UE JAPON B - 2021) [Cambiar](#)

ID	Fecha	Material de Apoyo	Profesor	Materia	Acciones
1	14/11/2021	ARTICULO	CALLISAYA HUANCA ALFREDO	Técnica Tecnológica Especializada - Informática	<a href="#">Clic para visualizar el Video</a> <a href="#">Clic para descargar el archivo</a>
2	14/11/2021	ARTICULO	CALLISAYA HUANCA ALFREDO	Técnica Tecnológica Especializada - Informática	<a href="#">Clic para visualizar el Video</a> <a href="#">Clic para descargar el archivo</a>

Materiales de apoyo

Clic para visualizar el Video

Clic para descargar el archivo

## AVALES DE CONFORMIDAD

El Alto, noviembre de 2021

Señor:

Lic. David Carlos Mamani Quispe  
DIRECTOR DE CARRERA INGENIERÍA DE SISTEMAS  
UNIVERSIDAD PÚBLICA DE EL ALTO  
Presente. -

**Ref.- AVAL DE CONFORMIDAD TUTOR METODOLÓGICO**

De mi mayor consideración.

Tengo a bien dirigirme a su persona para comunicarle mi conformidad del trabajo final del Proyecto de Grado, titulado **CUADERNO PEDAGÓGICO WEB COMO HERRAMIENTA DEL MAESTRO PARA EL SEGUIMIENTO DE ACTIVIDADES PEDAGÓGICAS**, CASO: "UNIDAD EDUCATIVA TÉCNICO HUMANÍSTICO JAPÓN B", elaborado por el postulante, Univ. ALFREDO CALLISAYA HUANCA con C.I. **6089143** LP, para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido con las consideraciones más distinguidas.

Atentamente



Ing. Dionicio Henry Pacheco Ríos  
TUTOR METODOLÓGICO



El Alto, noviembre de 2021

Señor:

Lic. David Carlos Mamani Quispe  
DIRECTOR DE CARRERA INGENIERÍA DE SISTEMAS  
UNIVERSIDAD PÚBLICA DE EL ALTO

Presente. -

**Ref.- AVAL DE CONFORMIDAD TUTOR ESPECIALISTA**

De mi mayor consideración.

Tengo a bien dirigirme a su persona para comunicarle mi conformidad del trabajo final del Proyecto de Grado, titulado **CUADERNO PEDAGÓGICO WEB COMO HERRAMIENTA DEL MAESTRO PARA EL SEGUIMIENTO DE ACTIVIDADES PEDAGÓGICAS**, CASO: "UNIDAD EDUCATIVA TÉCNICO HUMANÍSTICO JAPÓN B". elaborado por el postulante, Univ. ALFREDO CALLISAYA HUANCA con C.I. **6089143** LP, para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido con las consideraciones más distinguidas.

Atentamente



Lic. Santos Zenón Quispe Apaza  
TUTOR ESPECIALISTA

El Alto, noviembre de 2021

Señor:

Lic. David Carlos Mamani Quispe  
DIRECTOR DE CARRERA INGENIERÍA DE SISTEMAS  
UNIVERSIDAD PÚBLICA DE EL ALTO  
Presente. -

**Ref.- AVAL DE CONFORMIDAD TUTOR REVISOR**

De mi mayor consideración,

Tengo a bien dirigirme a su persona para comunicarle mi conformidad del trabajo final del Proyecto de Grado, titulado **CUADERNO PEDAGÓGICO WEB COMO HERRAMIENTA DEL MAESTRO PARA EL SEGUIMIENTO DE ACTIVIDADES PEDAGÓGICAS, CASO: "UNIDAD EDUCATIVA TÉCNICO HUMANÍSTICO JAPÓN B"**, elaborado por el postulante, **Univ. ALFREDO CALLISAYA HUANCA** con C.I. **6089143** LP, para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.

Atentamente



Ing. Fanny Helen Pérez Mamani  
TUTOR REVISOR





UNIDAD EDUCATIVA  
TÉCNICO HUMANÍSTICO  
"REPÚBLICA DEL JAPÓN B"  
COL. DE. 40730638  
Nivel Secundario Comunitario Productivo  
Z/ Villa Mercedes "A" c/ Marcelo Quiroga s/n



**EL SUSCRITO DIRECTOR DE LA UNIDAD EDUCATIVA  
TÉCNICO HUMANÍSTICO REPUBLICA DEL JAPÓN B  
DEPENDIENTE DE LA DIRECCIÓN DISTRITAL DE  
EDUCACIÓN DE EL ALTO 3, EN USO DE SUS  
ATRIBUCIONES.**

**CERTIFICA:**

Qué, el Sr. **ALFREDO CALLISAYA HUANCA** con C.I. 6089143 L.P. y Registro Universitario N° 200363, estudiante de la Carrera de INGENIERIA DE SISTEMAS, formalizo la implementación del sistema en la Unidad Educativa Técnico Humanístico Republica del Japón "B".

El Proyecto de titulado: **CUADERNO PEDAGÓGICO WEB COMO HERRAMIENTA DEL MAESTRO PARA EL SEGUIMIENTO DE ACTIVIDADES PEDAGÓGICAS,** Caso: Unidad Educativa Técnico Humanístico Republica del Japón "B".

Es cuanto certifico en honor a la verdad para fines que convenga al interesado.

El Alto, 24 de Noviembre de 2021

