

UNIVERSIDAD PÚBLICA DE EL ALTO
CARRERA DE INGENIERIA DE SISTEMAS



Proyecto De Grado

SISTEMA WEB PARA EL REGISTRO Y SEGUIMIENTO ACADEMICO
CASO: ACADEMIA DE MUSICA “MOZART”

Para optar al título de Licenciatura en Ingeniería de Sistemas

Mención: Informática Y Comunicaciones

Postulante: Univ. Martha Mita Choque

Tutor Metodológico: Ing. Marisol Arguedas Balladares

Tutor Especialista: Ing. Santos Aurelio Limachi Huanca

Tutor Revisor Lic. Adrian Eusebio Quisbert Vilela

EL ALTO – BOLIVIA

2020

Dedicatoria

A Dios por sobre todas las cosas,

A mi Padre y Hermanos con todo mi amor y cariño y

A toda mi familia por el apoyo brindado

Gracias.

Agradecimientos

En primer lugar agradezco a Dios por darme la fortaleza
para llegar al final de esta etapa de mi vida.

A mi familia de la Universidad Pública de El Alto gracias por los
momentos compartidos que ayudaron a mi formación personal.

Finalmente agradecer a mis amigos que son parte de la comunidad
universitaria

GRACIAS.

Resumen

El presente proyecto tiene por objetivo desarrollar un Sistema Web para el Registro y Seguimiento Académico de la Academia de Música Mozart.

En la actualidad el desarrollo de las tecnologías y el auge de las comunicaciones hacen que día a día las empresas, industrias e instituciones, se vean en la necesidad de automatizar los procesos de modo que esto les permita trabajar de forma más efectiva.

Es por tal razón que el sistema desarrollado ha sido concebido con la idea de mejorar los procesos desarrollados en la academia de música en cuanto al registro y Seguimiento Académico, lo cual proveerá un mayor índice de rendimiento con tiempos de ejecución mínimos en los procesos de inscripción del estudiante, manejo y accesibilidad a la información de manera efectiva.

Para el desarrollo del presente proyecto, se utilizó la metodología UWE el cual es herramienta más utilizada para el análisis, implementación y documentación de sistemas.

El resultado del presente proyecto es un sistema web, desarrollado en lenguaje de programación PHP con un motor de base de datos María DB, permitiendo a la Academia de Música Mozart el Seguimiento Académico de estudiantes que se registren a una especialidad de uno o varios instrumentos musicales para así mejorar su imagen como institución.

El del Sistema Web permitirá registrar los principales eventos que suceden en la actualidad, es necesario e indispensable ya que así se convierte en el punto esencial de todas las actividades de la academia.

Indicé General

1. Marco Preliminar	1
1.1. Introducción	1
1.2. Antecedentes	2
1.2.1. Antecedentes de la Institución	2
1.2.1.1. Misión.....	2
1.2.1.2. Visión	2
1.2.2. Antecedentes de Proyectos Similares.....	3
1.2.2.1. Antecedentes Internacionales.....	3
1.2.2.2. Antecedentes Nacionales	3
1.3. Planteamiento de Problema	4
1.3.1. Problema Principal	4
1.3.2. Problemas secundarios.....	4
1.4. Objetivos	5
1.4.1. Objetivos General.....	5
1.4.2. Objetivos Específicos	5
1.5. Justificación	6
1.5.1. Justificación Técnica	6
1.5.2. Justificación económica	6
1.5.3. Justificación social.....	7
1.6. Metodología	7
1.6.1. Técnica de Recopilación de Datos.....	7
1.6.2. Metodología de Desarrollo de Trabajo de Grado	8

1.7. Herramientas	9
1.7.1. JQuery.....	9
1.7.2. PHP 5.4.....	9
1.7.3. Codeigniter.....	10
1.7.4. Css3.....	10
1.7.5. Bootsrap.....	11
1.7.6. Servidor Apache.....	11
1.7.7. Gestor de base de datos MariaDB.....	12
1.7.8. Métricas De Calidad De Software.....	13
1.7.9. Métodos De Estimación De Costos.....	13
1.8. Límites Y Alcances	14
1.8.1. Limites.....	14
1.8.2. Alcances.....	14
1.9. Aportes	15
2. Marco Teórico	16
2.1. Introducción	16
2.2. Conceptos	16
2.2.1. Sistema.....	16
2.2.1.1. Sistemas Conceptuales O Abstractos.....	17
2.2.1.2. Sistemas Reales O Materiales.....	17
2.2.2. Sistema Web.....	17
2.2.3. Registro.....	18
2.2.4. Seguimiento Académico.....	19

2.3. Ingeniería De Requerimientos	20
2.3.1. Tareas De Análisis De Requerimientos.....	20
2.3.2. Funciones Y Habilidades Del Analista.....	22
2.4. Metodología.....	22
2.4.1. Metodología UWE	22
2.4.2. Características De UWE	23
2.4.3. Modelos De UWE.....	24
2.4.4. Fases De UWE.....	29
2.5. Herramientas de Desarrollo	30
2.5.1. Modelo Vista Controlador.....	30
2.5.1.1. Características	31
2.5.2. Gestor de Base De Datos MaríaDB	31
2.5.2.1. Utilidad del Sistema	32
2.5.2.2. Utilidad del Sistema	33
2.5.2.3. Prestaciones	34
2.5.2.4. Testeo.....	34
2.6. Lenguaje De Programación Php.....	35
2.7. Métricas De Calidad.....	36
2.7.1. Introducción A la Norma Iso/lec 9126	37
2.7.2. Características De La Norma Iso/lec 9126.....	38
2.8. Seguridad	49
2.8.1. Tipos De Software De Seguridad.....	50
2.8.2. Cómo Garantizar La Seguridad Del Software	50

2.9. Costo Del Proyecto	51
2.9.1. Modelo Cocomo	51
2.9.1.1. Modelos De Estimación	52
3. Marco Aplicativo	59
3.1. Introducción	59
3.2. Diagnóstico De La Situación Actual	59
3.2.1. Descripción De Funciones	59
3.2.2. Proceso De Inscripción	60
3.3. Ingeniería de Requerimientos	62
3.4. Aplicación de la Metodología UWE	63
3.4.1. Fase Captura, Análisis y Especificación de Requisitos	63
3.4.2. Definición De Actores.....	63
3.4.3. Requerimientos Funcionales.....	63
3.4.4. Definición De Procesos	64
3.4.5. Modelo De Casos De Uso	66
3.5. Fase Diseño de Sistema	83
3.5.1. Modelo De Contenido	83
3.5.2. Modelo De Navegación	84
3.5.3. Modelo De Presentación	86
3.5.4. Captura De Ventanas Del Sistema.....	88
3.6. Fase de Codificación del Sistema	96
3.6.1. Actividades.....	97
4. Calidad y Seguridad Del Software	98

4.1.	Pruebas De Calidad	98
4.1.1.	Funcionalidad	98
4.1.2.	Confiabilidad	103
4.1.3.	Usabilidad	105
4.1.4.	Eficiencia	106
4.1.5.	Mantenibilidad	107
4.1.6.	Portabilidad	109
4.2.	Seguridad de Software	111
4.2.1.	Pruebas de Caja Negra	111
4.2.2.	Pruebas de Caja Blanca	115
4.2.3.	Seguridad de base de Datos	118
5.	Costos y Beneficios	120
5.1.	Costos	120
5.2.	Beneficios	126
6.	Conclusiones Y Recomendaciones	127
6.2.	Conclusiones	127
6.3.	Recomendaciones	128
7.	Bibliografía	129
Anexo	133

Indicé de Tablas

Tabla 2.1: Dominios de información de puntos de función	38
Tabla 2.2: Factores de ponderación	39
Tabla 2.3: Valores de ajuste de la complejidad.....	40
Tabla 2.4: Métrica de adecuación.....	42
Tabla 2.5: Métrica de madurez	43
Tabla 2.6: Métrica de entendibilidad	44
Tabla 2.7: Métrica de comportamiento en el tiempo	45
Tabla 2.8: Métrica de cambiabilidad	47
Tabla 2.9: Métrica de conformidad de transportabilidad	48
Tabla 2.10: constantes modo básico	53
Tabla 2.11: constantes modo intermedio	54
Tabla 2.12: variables factor de ajustes del esfuerzo	56
Tabla 3.1: Especificación de Caso De Uso Secretaria.....	67
Tabla 3.2: Especificación de casos de uso Registrar Persona	67
Tabla 3.3: Especificación de Caso De Uso Asignar Instrumentos	68
Tabla 3.4: Especificación De Caso Uso Ver Convocatoria	68
Tabla 3.5: Especificación de Caso De Uso Generar Reportes	69
Tabla 3.6: Especificación de Caso De Uso Estudiante	70
Tabla 3.7: Especificación De Caso De Uso Docente	70
Tabla 3.8: Especificación de Caso De Uso Ingresar Al Sistema.....	71
Tabla 3.9: Especificación de Caso De Uso Ver Mi Perfil	72

Tabla 3.10: Especificación de Caso De Uso Ver Mis Cursos	73
Tabla 3.11: Especificación de Caso De Uso Ver Lista De Estudiantes.....	73
Tabla 3.12: Especificación de Caso De Uso Ver Acta De Calificaciones.....	74
Tabla 3.13: Especificación de Caso De Uso Asignar Nota	74
Tabla 3.14: Especificación de Caso De Uso Ingresar al Sistema	76
Tabla 3.15: Especificación de Caso De Uso Ver Convocatorias	76
Tabla 3.16: Especificación de Caso De Uso Ver Docentes	77
Tabla 3.17: Especificación de Caso De Uso Ingresar al Sistema	78
Tabla 3.18: Especificación de Caso De Uso Administrar Usuarios.....	79
Tabla 3.19: Especificación de Caso De Uso Administrar Cursos.....	79
Tabla 3.20: Especificación de Caso De Uso Administrar Instrumentos	80
Tabla 3.21: Especificación de Caso De Uso Administrar Convocatoria.....	81
Tabla 3.22: Especificación de Caso De Uso Administrar Inscripciones	81
Tabla 3.23: Especificación de Caso De Uso Generar Reportes	82
Tabla 4.1: Parámetros de medición	99
Tabla 4.2: Calculo de Punto de función	99
Tabla 4.3: Valores de ajuste de complejidad	100
Tabla 4.4: Métrica de adecuación.....	102
Tabla 4.5: Métrica de madurez	104
Tabla 4.6: Métrica de entendibilidad	105
Tabla 4.7: Métrica de comportamiento en el tiempo	106
Tabla 4.8: Métrica de cambiabilidad	108
Tabla 4.9: Métrica de conformidad de transportabilidad	110

Tabla 4.10: Resultados de la Norma ISO-9126	111
Tabla 4.11: Valores Límite de Inicio de Sesión	111
Tabla 4.12: Descripción de Pruebas de la caja negra Inicio de Sesión	112
Tabla 4.13: Valores Límite de Registro de Estudiantes	113
Tabla 4.14: Descripción de Pruebas caja negra Registro de Datos de Estudiante	114
Tabla 4.15: Descripción de Pruebas de caja Blanca mantenimiento al software.	116
Tabla 4.16: Pruebas de caja Blanca Estructura de condición fuera de estándar.	117
Tabla 4.17: seguridad de base de datos	118
Tabla 5.1: variables factor de ajustes del esfuerzo	120
Tabla 5.2: Constantes de modelo COCOMO	122
Tabla 5.3: Costos de Alojamiento Web de sysoft-Bo	124
Tabla 5.4: Costos de alquiler de Alojamiento Web	125
Tabla 5.5: Costos totales	126

Índice de Figuras

Figura 2.1: Representación de Casos de Uso	25
Figura 2.2: Representación de Diagrama de Actividades	25
Figura 2.3: Representación de una clase	26
Figura 2.4: Representación de una clase de navegación	27
Figura 2.5: Clase Índice, Consulta y Menú	28
Figura 2.6: Clase de Presentación.....	28
Figura 2.7: Modelo, Vista, Controlador	30
Figura 2.8: Características de la norma ISO 9126	37
Figura 2.9: Subcaracterísticas de la norma ISO 9126.....	37
Figura 3.1: Situación Actual	62
Figura 3.2: Caso de Uso Secretaria.....	66
Figura 3.3: Caso de Uso Docente.....	71
Figura 3.4: Caso de Uso Estudiante	75
Figura 3.5: Caso de Uso Director.....	78
Figura 3.6: Modelo de contenido.....	84
Figura 3.7: Modelo de Navegación Usuario Secretaria.....	85
Figura 3.8: Modelo de Navegación Usuario Docente.....	85
Figura 3.9: Modelo de Navegación Usuario Administrador (Director)	86
Figura 3.10: Modelo de Presentación Usuario Secretaria.....	86
Figura 3.11: Modelo de Presentación Usuario Docente.....	87
Figura 3.12: Modelo de Presentación Usuario Administrador	87

Figura 3.13: Ventana principal de la página web Academia de Música Mozart	88
Figura 3.14: Ventana principal de las convocatorias.....	88
Figura 3.15: Ventana principal de Información de la Academia de Música Mozart	89
Figura 3.16: Ingreso Al Sistema Inicio de Sesión.....	90
Figura 3.17: Perfil De Usuario administrador	91
Figura 3.18: Asignación de Usuarios y Grupos.....	91
Figura 3.19: Registro de Estudiantes y docentes.....	92
Figura 3.20: Inscripción del estudiante nuevo o antiguo	92
Figura 3.21: Asignar Cursos a Docentes	93
Figura 3.22: Asignar convocatorias.....	93
Figura 3.23: Asignación de Instrumentos.....	94
Figura 3.24: Registro de Pagos	94
Figura 3.25: Acta de Notas de los Estudiantes	95
Figura 3.26: Perfil de Docente	95
Figura 3.27: Control de Asistencia	96
Figura 4.1: pruebas de caja negra de Inicio de Sesión	112
Figura 4.2: Pruebas de caja negra Registro de Estudiantes.....	114
Figura 4.3: Falta de comentarios realizar Mantenimiento al Software	115
Figura 4.3: Prueba de caja blanca Estructura de condición fuera de estándar	116

1. Marco Preliminar

1.1. Introducción

“La tecnología es el conocimiento y la utilización de herramientas, técnicas y Sistemas con el fin de servir a un propósito más grande como la resolución de problemas o hacer la vida más fácil y mejor. Su importancia para los seres humanos es enorme porque les ha ayudado a adaptarse a su ambiente”. (Leon, 2016)

“Las nuevas tecnologías ponen las fuentes de aprendizaje a disposición de los estudiantes especialmente entre los más maduros, quienes usan la tecnología para dar forma y descubrir su propio aprendizaje”. (Cobo, 2011)

Actualmente la Academia de Música Mozart cuenta con registro de estudiantes, seguimiento académica de estudiantes de forma manual, realizadas en libros de actas y con respaldos correspondientes, esto genera perdida de información y confusión de datos e incumple en el cumplimiento de su misión y visión.

Las instituciones públicas o privadas tienden a utilizar la tecnología como herramienta indispensable para el desarrollo de proyectos al servicio de la comunidad para esto es necesario adoptar tecnologías de la información en la Academia de Música “Mozart” logrando así la optimización de sus recursos y agilizando los procesos mediante el Sistema Web para Registro y Seguimiento Académico de estudiantes .

Para el desarrollo del software se hará uso de la metodología UWE es una metodología que permite modelar de mejor manera una aplicación Web. Teniendo como lenguaje de programación PHP, y gestor de base de datos Maria DB. Juntamente con las herramientas de desarrollo html5, Css3, JavaScript , bootstrap , codelgniter, jquery.

1.2. Antecedentes

1.2.1. Antecedentes de la Institución

La Academia de Música Mozart es una institución de educación para todas aquellas personas que deseen aprender todo sobre los instrumentos y técnicas de Música que se encuentra ubicada en la ciudad de El Alto av. Junín de Zona Villa Adela, las convocatorias lanzadas son para diferentes cursos y niveles que son: Básico, intermedio y avanzada.

1.2.1.1. Misión

Su Misión, formar niños, jóvenes y adultos líderes en valores artísticos tecnológicos por niveles de aprendizaje aplicando técnicas de enseñanza con métodos educativos más recientes e innovadoras a través de espacios de practica instrumental, teoría y estudios tecnológicos para reforzar las habilidades y destrezas creativas e interpretativas en beneficio de los estudiantes y la comunidad.

1.2.1.2. Visión

Su Visión ser una de las mejores instituciones educativas de arte y tecnología en el país con un modelo educativo integral a nivel profesional en sus diferentes áreas a través de nuevas infraestructuras y con amplio equipamiento de última generación para formar músicos profesionales capaces y líderes dentro de la industria de la música predispuestos a competir a nivel nacional e internacional.

1.2.2. Antecedentes de Proyectos Similares

1.2.2.1. Antecedentes Internacionales

En 2015 se realizó el trabajo de tesis para la obtención del título “Ingeniero de Sistemas Informáticos” en El Salvador.

Roberto Avelar García, Eric Marvin Guerrero, Carmen Mariela Reyes De Márquez(2015) Sistema Informático Con Interfaz Web Para El Registro Académico, Recurso Humano, Control Bibliotecario Y Bono Escolar, Del Centro Escolar Cantón El Espino Abajo De Zacatecoluca, Departamento De La Paz (Universidad De El Salvador Facultad Multidisciplinaria Paracentral Departamento De Informática, 2015).

En 2014 se realizó el trabajo de proyecto de grado para la obtención del título “ingeniería de Sistemas” en Ecuador.

Carmen Bastidas (2014) Proyecto De Desarrollo E Implementación De Un Sistema Automatizado De Registro De Alumnos, Profesores Y Personal Administrativo De La Unidad Educativa Darío Egos Grijalva De La Ciudad De San Gabriel, (Universidad Católica Del Ecuador Escuela De Ingeniería, 2014).

1.2.2.2. Antecedentes Nacionales

En 2017 se realizó el trabajo de proyecto de grado para la obtención del título “Licenciatura en Informática” en La Paz Bolivia.

Oliver Alarcón Arroyo (2017) “Sistema Web Para El Control Y Seguimiento De Kárdex Administrativo Caso: Postgrado En Informática”, (Facultad De Ciencias Puras Informática, Universidad Mayor De San Andrés, 2017).

En 2016 se realizó el trabajo de proyecto de grado para la obtención del título “Licenciatura en Ingeniería de Sistemas” en La Paz El Alto Bolivia.

Iván Alfredo Mamani Ochoa (2016) “Sistema De Información Académica Caso: Carrera De Medicina-Upea”, (Universidad Pública De El Alto, 2016).

1.3. Planteamiento de Problema

Actualmente la Academia de Música Mozart realiza el procesamiento de la información como ser el Registro y Seguimiento académico de los estudiantes de forma manual, por lo tanto la falta de un Sistema automatizado hace que exista errores, redundancia de datos, perdida de información y sobre todo influye en la calidad de atención a los usuarios ocasionando deserción. Bajo este contexto se presenta el siguiente problema

1.3.1. Problema Principal

Actualmente la Academia de música Mozart realiza sus procesos de registro de estudiantes y seguimiento académico de forma manual lo cual ocasiona dificultad en la toma de decisiones y el crecimiento institucional concluyendo con pérdidas económicas.

1.3.2. Problemas secundarios

- Registro de datos de los estudiante realizadas en libros de actas y cuadernos de notas de forma manual.
- Existe la dificultad en la coordinación de inscripción y asignación de Cursos a cada estudiante.

- Existe la dificultad en el Registro de notas por el docente de turno en las aulas de especialidad debido a la confusión de Registro de estudiantes existentes por especialidad de instrumento.
- La información que se necesita de cada proceso de registro de estudiantes no son oportunos para generar informes confiables.
- La información del proceso de avance del estudiante no es oportuna para brindar un informe en tiempo real

1.4. Objetivos

1.4.1. Objetivos General

Desarrollar Sistema Web para el Registro y Seguimiento académico de la Academia de Música “Mozart”, que genere Información de forma eficiente, oportuna y que coadyuve a la correcta toma de decisiones y por ende al crecimiento de la institución.

1.4.2. Objetivos Específicos

- Diseñar Sistema Web para el Registro y Seguimiento Académico de los estudiantes, con interfaces amigables que permitan una interacción entre el usuario y Sistema.
- Registrar y asignar de manera oportuna los cursos correspondientes del estudiante.
- Automatizar la asignación de notas por el docente de turno al estudiante por especialidad a la que corresponde.
- Generar informes confiables y oportunos de los estudiantes inscritos y docentes registrados y asignados a un curso.

- Generar un informe del total de estudiantes inscritos a los cursos de instrumentos musicales por modalidad de convocatoria.

1.5. Justificación

1.5.1. Justificación Técnica

Las técnicas que se utilizaran para desarrollar el Sistema web, son las herramientas de software (Maria DB, Php, Html5,Css3, jquery, javaScript, Bootstrap, Codeinaiter) las cuales son de gran utilidad.

El Sistema Web para el Registro y Seguimiento Académico será de gran ayuda para la Academia ya que acortara los tiempos de procesos de Registros de Estudiantes.

El Sistema desarrollado es multiplataforma puede funcionar tanto en GNU/Linux o Windows; escalable con capacidad de crecer en el tiempo.

La Academia cuenta con el hardware y software que se requiere para el funcionamiento del Sistema.

1.5.2. Justificación económica

El Sistema Web para el Registro y Seguimiento Académico no implica riesgos ni perdidas económicamente este proyecto se justifica en su construcción ya que no requiere de sumas económicas elevadas porque se usara herramientas libres y su utilidad y beneficio aportara directamente a la institución. Maximizando los Ingresos económicos por la atención de la calidad que se brindara a los estudiantes por contar con información confiable.

1.5.3. Justificación social

El Sistema Web realizara el Registro y Seguimiento Académico de estudiantes que quieran tomar un curso de especialización en uno o varios instrumentos, aportara directamente a la Academia para su uso como herramienta de trabajo para el área administrativa para optimizar sus procesos de manejo de información en la Academia de música Mozart.

El Sistema web también beneficiara a los usuarios (Padres de Familia, Docentes y Estudiantes) que visiten la página web ya que podrán ver algunos de los materiales que se usan para los cursos de instrumentos podrán ver la información de las convocatorias de los cursos que se están llevando a cabo con cada estudiante y los docentes de turno.

1.6. Metodología

Las metodologías y técnicas que se utilizaran para el desarrollo del presente proyecto son los siguientes:

1.6.1. Técnica de Recopilación de Datos

En el desarrollo del presente proyecto, se utiliza las técnicas de: observación, entrevista, cuestionarios y encuestas a los usuarios finales lo que nos permite de forma directa, obtener la información necesaria concerniente al manejo de información de estudiantes en la Academia de música

1.6.2. Metodología de Desarrollo de Trabajo de Grado

Para el desarrollo del software se hará uso de la metodología UWE (UML-Based Web Engineering, en español Ingeniería Web Basada en UML) es una metodología que permite modelar de mejor manera una aplicación Web, para proceso de creación de aplicaciones esta detallada, con una gran cantidad de definiciones, en el proceso de diseño lista que debe utilizarse. Procede de manera iterativa e incremental, coincidiendo con UML, incluyendo flujos de trabajo y puntos de control.

El principal objetivo del enfoque UWE es proporcionar: un lenguaje de modelado específico del dominio basado en UML; una metodología dirigida por modelos; herramientas de soporte para el diseño sistemático; y herramientas de soporte para la generación semi-automática de Aplicaciones Web.

Modelo de aplicación web según el metodología UWE.

Modelo de Casos de Uso: se modela requisitos funcionales de la aplicación Web para ver como interactúa cada uno de ellos.

Modelo Conceptual: Materializa en un modelo de dominio, considerando los requisitos reflejados en los casos de uso.

Modelo Navegación: Especifica el entorno en la cual se realizará el aspecto de navegación de la aplicación Web.

Modelo de presentación: Representa las vistas del interfaz del usuario mediante modelos estándares de interacción UML.

En cuanto a los requisitos, UWE los clasifica dependiendo del carácter de cada uno. Además distingue entre las fases de captura definición y validación de requisitos (Engineering, 2012).

1.7. Herramientas

Para la elaboración del proyecto se necesitara de las siguientes herramientas que ayudara en el desarrollo del mismo:

1.7.1. JQuery

Es un Framework JavaScript, nos ofrece una infraestructura con la que tendremos mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Por ejemplo, con JQuery obtendremos ayuda en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax, etc. Cuando programemos JavaScript con JQuery tendremos a nuestra disposición una interfaz para programación que nos permitirá hacer cosas con el navegador que estemos seguros que funcionarán para todos nuestros visitantes. Simplemente debemos conocer las librerías del Framework y programar utilizando las clases y métodos para la consecución de nuestros objetivos. Según (Miguel Angel, 2012).

1.7.2. PHP 5.4

Es un lenguaje de programación interpretado de alto nivel al lado del servidor para internet, muy similar en su sintaxis al lenguaje C, con algunas diferencias, no compila como al igual que C, ya que es un intérprete, por lo tanto cada vez que se debe ejecutar un programa, lo interpreta verificando toda su sintaxis.

PHP nos brinda la posibilidad de realizar tareas de forma automatizadas, mejorando la productividad de nuestro sitio web y dando la posibilidad de añadir

gran cantidad de funcionalidades que con HTML no podemos hacerlo, ya que HTML no es un lenguaje de programación. Según (Alvarez, López, & Gutierrez, 2013).

1.7.3. Codeigniter

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

El diseño orientado al rendimiento de este framework de desarrollo web se revela en su parca arquitectura, pues se basa en el patrón Modelo-Vista-Controlador (MVC). El principio fundamental que sustenta a la arquitectura de desarrollo MVC es la estricta separación entre el código y la presentación, gracias a una estructura modular de software y a la externalización del código PHP. Según (pineda, 2016).

1.7.4. Css3

El lenguaje CSS permite presentar, de manera estructurada, un documento que fue escrito en un lenguaje de marcado. Se usa especialmente en el diseño visual de un sitio web cuando las páginas se hallan escritas en XML o HTML.

El diseño del CSS posibilita establecer una separación entre el contenido y la forma de presentación del documento (dada por las fuentes, los colores y las capas empleadas). Así se puede lograr que muchos documentos HTML compartan la apariencia, utilizando una única hoja de estilo

para todos (que se especifica en un archivo .css). Gracias a esta particularidad, se evita tener que repetir el código en la estructura. Según (Porto, 2019).

1.7.5. Bootstrap

Es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “responsive design” o diseño adaptativo.

El beneficio de usar responsive design en un sitio web, es principalmente que el sitio web se adapta automáticamente al dispositivo desde donde se acceda. Lo que se usa con más frecuencia, y que a mi opinión personal me gusta más, es el uso de media queries, que es un módulo de CSS3 que permite la representación de contenido para adaptarse a condiciones como la resolución de la pantalla y si trabajás las dimensiones de tu contenido en porcentajes, puedes tener una web muy fluida capaz de adaptarse a casi cualquier tamaño de forma automática. Según (Solis, 2014).

1.7.6. Servidor Apache

Apache es el Servidor Web hecho por excelencia por su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. La licencia es una descendiente de la licencias BSD (Berkeley Software Distribution), no es GPL (General Public License). Esta

licencia permite hacer lo que quieras con el código fuente siempre que les reconozcas su trabajo.

Apache es una tecnología gratuita de código fuente abierta. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto.

Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor las características de Apache se pueden extender hasta donde nuestra imaginación y conocimientos lleguen. Según (Cooper, 2004).

1.7.7. Gestor de base de datos MariaDB

MariaDB es un Sistema de gestión de bases de datos. Se deriva de MySQL, una de las base de datos más importantes que ha existido en el mercado, utilizada para manejar grandes cantidades de información.

Para que se tenga una idea de la enorme capacidad para mover grandes cantidades de información, MySQL ha sido la base de datos utilizada por proyectos de internet de la índole de Facebook, Twitter y Wikipedia.

La simplicidad de la sintaxis permite crear bases de datos simples o complejos con mucha facilidad; es compatible con múltiples plataformas informáticas y está provista de una infinidad de aplicaciones que permiten acceder rápidamente a las sentencias de la gestión de base de datos.

Además, permite a los desarrolladores y diseñadores realizar cambios en los sitios web con sólo cambiar un archivo, (sin necesidad de modificar todo el código web) para que se ejecuten en toda la estructura de datos que se comparte en la red. Según (Inco, 2018).

1.7.8. Métricas De Calidad De Software

Se evaluará la calidad del software usando la ISO 9126.

ISO 9126 es un estándar internacional para la evaluación de la calidad del software. Está reemplazado por el proyecto SQuaRE, ISO 25000:2005, el cual sigue los mismos conceptos.

El estándar está dividido en cuatro partes las cuales dirigen, realidad, métricas externas, métricas internas y calidad en las métricas de uso y expendido. El modelo de calidad establecido en la primera parte del estándar, ISO 9126-1, clasifica la calidad del software en un conjunto estructurado de características y subcaracterísticas de la siguiente manera:

Funcionalidad: Adecuación, Exactitud, Interoperatividad y Seguridad

Confiabilidad: Madurez, Tolerancia a fallas y Recuperabilidad

Usabilidad: Entendibilidad, capacidad de aprendizaje y operatividad

Eficiencia: Comportamiento en tiempos y Comportamiento de recursos

Mantenibilidad: Analizabilidad, Modificabilidad, Estabilidad y Capacidad de prueba

Portabilidad: Adaptabilidad, Instalabilidad y Remplazabilidad. Según. (Prieto, 2017)

1.7.9. Métodos De Estimación De Costos

Para la estimación de costos del software se aplicará el Modelo COCOMO.

El modelo COCOMO original se convirtió en uno de los modelos de estimación más ampliamente utilizados y estudiados en el mundo. Pressman R. S. (2010).

Está compuesto por tres modelos que corresponden a distintos niveles de detalle y precisión. Mencionados en orden creciente son: Modelo Básico, Intermedio y Detallado.

La estimación es más precisa a medida que se toman en cuenta mayor cantidad de factores que influyen en el desarrollo de un producto de software. Según. (Cocomo, 2017)

1.8. Límites Y Alcances

1.8.1. Limites

El Sistema web para el Registro y Seguimiento académico de estudiantes de la Academia de música Mozart está elaborado exclusivamente para la institución educativa con el fin de optimizar sus procesos de manejo de información.

- ❖ No esta enlazado al Ministerio de Educación
- ❖ No controla el Pago de Mensualidades
- ❖ No realiza Backups en forma Automática

1.8.2. Alcances

El Sistema web para el Registro y Seguimiento Académico de estudiantes tendrán los siguientes alcances:

- Módulo de interfaz dinámica y presentación de la Academia de música Mozart
- Módulo de Grupos y usuarios.
- Módulo de Registro e inscripción de estudiantes que quieran cursar una especialización de uno o varios instrumentos.

- Módulo de Registro a Docentes para los cursos de especialización de uno o varios instrumentos Musicales.
- Módulo de asignar Convocatoria.
- Módulo de Asignar Instrumentos.
- Módulo de reportes de Registro e inscripción de cada estudiante.
- Módulo de Reportes de Pagos Académicos.

1.9. Aportes

El Sistema Web para Registro y Seguimiento académico de estudiantes es un importante aporte para la Academia de música Mozart, ya que beneficia directamente a la institución a acortar sus tareas de forma automatizada en cuanto a la inscripción y asignación de convocatorias a los cursos de especialidad de uno o varios instrumentos Musicales.

2. Marco Teórico

2.1. Introducción

En este capítulo se conocerá las definiciones y conceptos fundamentales acerca del Sistema Web para el Registro y Seguimiento Académico de estudiantes de la Academia de Música “Mozart”.

Se llegara a conocer las herramientas, que serán clave en el desarrollo del Sistema, la conexión con una base de datos que contiene la información necesaria para que los usuarios puedan realizar sus actividades en el Sistema.

La información de este capítulo ayudara al desarrollo del Sistema en los requerimientos para desarrollar el Sistema web de información con los siguientes pasos análisis, diseño, implantación, pruebas y análisis de costos que serán necesarios para la mejor comprensión en el proceso de elaboración del Sistema web.

2.2. Conceptos

2.2.1. Sistema

Un Sistema es un conjunto de elementos relacionados entre sí que funciona como un todo.

Si bien cada uno de los elementos de un Sistema puede funcionar de manera independiente, siempre formará parte de una estructura mayor. Del mismo modo, un Sistema puede ser, a su vez, un componente de otro Sistema.

La palabra Sistema procede del latín *systema*, y este del griego *σύστημα* (*systema*), identificado en español como “unión de cosas de manera organizada”. De esta palabra se derivan otras como antiSistema o ecoSistema.

De igual forma, existe una corriente de pensamiento filosófico llamada sistemismo, creada por el epistemólogo argentino Mario Bunge, que propone que todo lo que existe es un Sistema o un componente de un Sistema más complejo.

Existen dos grandes tipos de Sistemas:

2.2.1.1. Sistemas Conceptuales O Abstractos

Un Sistema conceptual son todas las ideas, conceptos, signos, hipótesis, teorías o símbolos que se utilizan para crear un constructor, es decir, una entidad hipotética. Un ejemplo de Sistema conceptual es la matemática, que a su vez está formada por varios componentes abstractos (álgebra, cálculo, etc.).

2.2.1.2. Sistemas Reales O Materiales

Son estructuras compuestas por elementos tangibles, sean de origen natural o artificial. Ejemplos de Sistemas reales son el cuerpo humano o el hardware de una computadora. Según (Raffino, Sistema, 11)

2.2.2. Sistema Web

Los Sistemas Web o también conocido como aplicaciones Web son aquellos que están creados e instalados no sobre una plataforma o Sistemas operativos (Windows, Linux). Sino que se alojan en un servidor en Internet o sobre una intranet (red local). Su aspecto es muy similar a páginas Web que vemos normalmente, pero en realidad los 'Sistemas Web' tienen funcionalidades muy potentes que brindan respuestas a casos particulares.

Los Sistemas Web se pueden utilizar en cualquier navegador Web (chrome, firefox, Internet Explorer, etc) sin importar el Sistema operativo. Para utilizar las aplicaciones Web no es necesario instalarlas en cada computadora ya que los usuarios se conectan a un servidor donde se aloja el Sistema.

Aplicaciones Web.- Las trabajan con bases de datos que permiten procesar y mostrar información de forma dinámica para el usuario.

Los Sistemas desarrollados en plataformas Web, tienen marcadas diferencias con otros tipos de Sistemas, lo que lo hacen muy beneficioso tanto para las empresas que lo utilizan, como para los usuarios que operan en el Sistema.

Este tipo de diferencias se ven reflejada en los costos, en la rapidez de obtención de la información, en la optimización de las tareas por parte de los usuarios y en alcanzar una gestión estable. Según (Baez, 2012)

2.2.3. Registro

El término Registro puede referirse a un gran número de circunstancias que tienen en común el hecho de dejar establecido un determinado fenómeno con sus características específicas para que haya conocimiento al respecto por parte de terceros o por un control. Para la Tecnología de información. Existen diferentes tipos de Registros, pero en todos los casos se hace referencia al concepto de almacenamiento de datos o información sobre el estado, proceso o uso de la computadora.

Un área donde este tipo de situación suele ser recurrente es en entidades públicas, que generalmente necesitan tomar referencias de la población de manera continua para lograr una administración más eficiente. Con el desarrollo de la informática, este tipo de procedimiento, sin duda, se ha simplificado en gran medida.

Un Registro de Sistema se convierte en una base de datos para almacenar la configuración, las opciones y los comandos del Sistema operativo. En general, estos Registros se usan en Sistemas Microsoft Windows. Un Registro del Sistema puede contener información y configuraciones de hardware y software utilizadas, preferencias del usuario, asociaciones de archivos y archivos, usos, cambios y modificaciones del Sistema, y más. Estos Registros se guardan en el Sistema con nombres como «User.dat» o «System.dat» y el usuario puede recuperarlos para transportarlos a otro Sistema. Según (Redaccion, Definicion de Registro, 2019)

2.2.4. Seguimiento Académico

Seguimiento.- La palabra Seguimiento es la acción y efecto de seguir o seguirse, en el contexto popular suele usarse como sinónimo de persecución, observación o vigilancia. Siendo este mismo usado principalmente en el contexto de investigaciones policiales, detectivescas, jurídicas, medicas, científicas, estadística, entre otras; para observar y analizar la evolución un determinado caso. Aunque el término puede aplicarse a cualquier investigación, proceso o proyecto con observación constante. Según (Redaccion, Definicion de Seguimiento , 2019)

Académico.- es aquel que es utilizado para denominar no sólo a individuos sino también a entidades, objetos o proyectos que se relacionan con niveles superiores de educación. La variedad de los significados del concepto de académico permite que este sea utilizado no sólo para aquellos que realizan investigaciones o trabajan como tales, sino también para individuos que cursan estudios correspondientes al nivel superior. Segun (Bembibre, 2009)

2.3. Ingeniería De Requerimientos

Según Michael Arias Chaves, la ingeniería de requerimientos es: “El proceso de recopilar, analizar y verificar las necesidades del cliente o usuario para un Sistema es llamado ingeniería de requerimientos. La meta de la ingeniería de requerimientos (IR) es entregar una especificación de requisitos de software correcta y completa”. (Chaves, 2007), 4.

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

“La especificación del software o la ingeniería de requerimientos consisten en el proceso de comprender y definir qué servicios se requieren del Sistema, así como la identificación de las restricciones sobre la operación y el desarrollo del Sistema. La ingeniería de requerimientos es una etapa particularmente crítica del proceso de software, ya que los errores en esta etapa conducen de manera inevitable a problemas posteriores tanto en el diseño como en la implementación del Sistema. Sommerville, (2011), 36.

2.3.1. Tareas De Análisis De Requerimientos

El análisis de requerimiento del software se puede subdividir en cinco áreas de esfuerzo:

- a) Reconocimiento del problema.
- b) Evaluación y síntesis.
- c) Modelado.
- d) Especificación.
- e) Revisión.

Todos los métodos de análisis se relacionan por un conjunto de principios operativos:

- a) Debe representarse y entenderse el dominio de la información de un problema.
- b) Deben definirse las funciones que debe realizar el software.
- c) Debe representarse el comportamiento del software (como consecuencia de acontecimientos externos),
- d) Deben dividirse los modelos que representan información, función y comportamiento de manera que se descubran los detalles por capas (o jerárquicamente).
- e) El proceso de análisis debería ir desde la información esencial hasta el detalle de la implementación.

Además de los principios operativos mencionados anteriormente, se sugiere un conjunto de principios directrices para la ingeniería de requerimientos:

- a) Entender el problema antes de empezar a crear el modelo de análisis.
- b) Desarrollar prototipos que permitan al usuario entender cómo será la interacción hombre-máquina.
- c) Registrar el orden y la razón de cada requerimiento,
- d) Usar múltiples planteamientos de requerimientos.
- e) Priorizar los requerimientos.
- f) Trabajar para eliminar la ambigüedad.

2.3.2. Funciones Y Habilidades Del Analista

La función principal de un analista del software o ingeniero de requerimientos es llevar a cabo las actividades necesarias para cumplir con las cinco áreas de esfuerzo descritas en la sección anterior. Para lo cual hace uso de las siguientes técnicas:

- a) Entrevistas
- b) Talleres
- c) Observación
- d) Encuestas
- e) Revisión documental
- f) Uso de especificaciones formales para requerimientos (formatos estándar de documentos, UML, etc.)

El ingeniero de requisitos debe poseer habilidades particulares para facilitar la comunicación con el cliente y ganar su confianza.

2.4. Metodología

2.4.1. Metodología UWE

UWE UML (UML-Based Web Engineering – Ingeniería Web Basada en UML) es una propuesta metodológica basada en el Proceso Unificado (Jacobson & Booch & Rumbaugh, 1999) y UML para el desarrollo de aplicaciones web (Hennicker & Koch, 2000, Koch, 2001). UWE cubre todo el ciclo de vida de este tipo de aplicaciones, centrandó además su atención en aplicaciones personalizadas. Para este trabajo, nos interesa principalmente analizar la propuesta de captura de requisitos de UWE. Esta metodología distingue entre la tarea de elicitar requisitos, definir y validar los requisitos. El

resultado final de la captura de requisitos en UWE es un modelo de casos de uso acompañado de documentación que describe los usuarios del Sistema, las reglas de adaptación, los casos de uso y la interfaz.

UWE clasifica los requisitos en dos grandes grupos: funcionales y no funcionales. Los requisitos funcionales tratados por UWE son:

- Requisitos relacionados con el contenido.
- Requisitos relacionados con la estructura.
- Requisitos relacionados con la presentación.
- Requisitos relacionados con la adaptación.
- Requisitos relacionados con los usuarios.

Además, UWE propone como técnicas apropiadas para la captura de los requisitos de un Sistema web las entrevistas, los cuestionarios y los checklists y los casos de uso, los escenarios y el glosario para la definición de requisitos. Para la validación propone walk-throughs (“recorrer” el Sistema), auditorías y prototipos. (Escalona & Koch, 2002: 14)

2.4.2. Características De UWE

La metodología UWE define vistas especiales representadas gráficamente por diagramas en UML, tales como el modelo de navegación y el modelo de presentación.

Los diagramas se pueden adaptar como mecanismos de extensión basados en estereotipos que proporciona UML. Estos mecanismos de extensión son los que UWE utiliza para definir estereotipos que son los que finalmente se utilizarán en las vistas especiales para el modelado de aplicaciones Web. De esta manera se

obtiene una notación UML adecuada para un dominio específico a la que se conoce como perfil UML.

Un perfil UML consiste en una jerarquía de estereotipos y un conjunto de restricciones. Los estereotipos son utilizados para representar instancias de las clases. La ventaja de utilizar los perfiles de UML es que casi todas las herramientas case de UML se reconocen. Los modelos deben ser fácilmente adaptables al cambio en cualquier etapa del desarrollo.

2.4.3. Modelos De UWE

Con respecto al proceso de creación de la aplicación, UWE se vale mediante el uso de metodologías estándares reconocidas como UML Principalmente y también del lenguaje de especificación de restricciones asociado a OCL (lenguaje de restricciones para objetos).

Para recolectar los requerimientos de las aplicaciones web, esta metodología propone una ampliación en el proceso de creación, la cual se divide en las siguientes cuatro actividades:

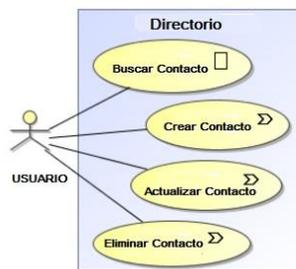
Análisis de requisitos: plasma los requerimientos funcionales de la aplicación web, mediante modelos de caso de uso.

Para describir los requerimientos funcionales de una aplicación se puede usar un modelo de casos de uso, este modelo describe una parte del comportamiento de la aplicación.

Los casos de uso no son propiamente un caso de análisis, se limitan a describir procesos de dominio que pueden expresarse en forma narrativa en un formato estructurado de prosa y pueden ser eficaces en un proyecto de tecnología. No obstante, constituyen un paso preliminar muy útil porque describen las especificaciones de un Sistema. (Jacobson, 2000).

Los modelos de casos de uso están conformados por dos elementos de modelado principales, que son los casos de uso y los actores. Un caso de uso es la unidad coherente de funcionalidad provista de aplicaciones que interactúan con uno o más actores externos de la aplicación, un actor es el rol que un usuario puede desempeñar con respecto a un Sistema o entidad, tales como un Sistema o una base de datos, además existen relaciones entre estos elementos como asociaciones entre actores y casos de uso.

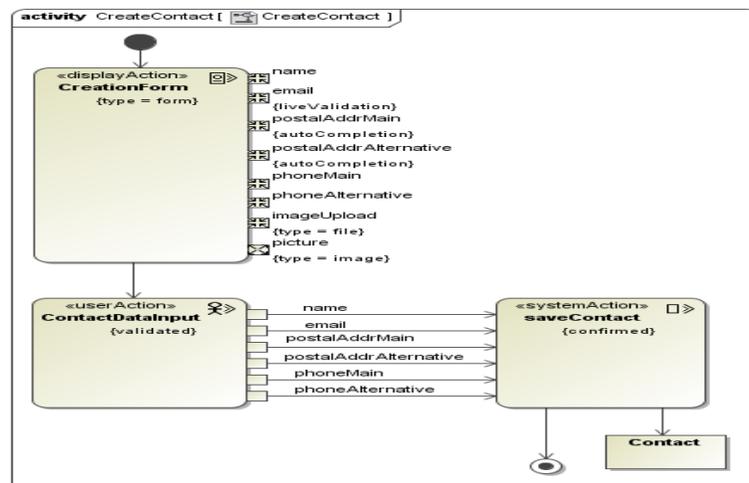
Figura 2.1: Representación de Casos de Uso



Fuente y Elaboración: (Koch, 2000)

Los Diagramas de actividades describen el flujo de los procesos que se realizan en un caso de uso.

Figura 2.2: Representación de Diagrama de Actividades

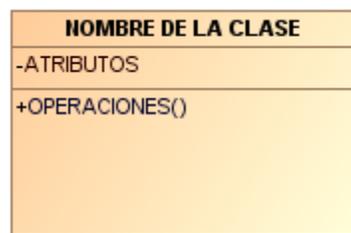


Fuente y Elaboración: (Koch, 2000)

Diseño conceptual: se define un modelo de dominio, considerando los requisitos plasmados en los casos de uso, el diagrama de clases representará los conceptos con un gran porcentaje de detalle.

El diseño conceptual está basado en el análisis de requerimientos del paso anterior, incluye a los objetivos involucrados en la interacción entre el usuario y la aplicación especificados en los casos de uso. Apunta a la construcción de modelos de clase con estos objetos que intentan ignorar tanto como sea posible los caminos de navegación y los pasos de presentación. (Koch, 2000)

Figura 2.3: Representación de una clase



Fuente: (Koch, 2000)

Diseño navegacional: consta de la construcción de dos modelos de navegación el modelo de espacio de navegación y el modelo de estructura de navegación.

Modelo de espacio navegacional: especifica objetos que pueden ser visitados a través de la aplicación.

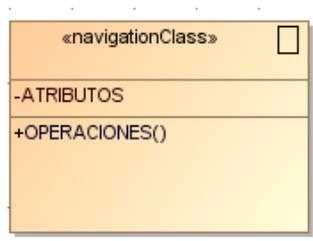
El modelo de espacio de navegación es construido con las clases de navegación y las asociaciones de navegación y están representadas gráficamente por un diagrama de clases de UML. (Koch, 2000)

La clase de navegación modela una clase cuyas instancias son visitadas por los usuarios durante la navegación. Se les asigna el nombre que se diera a las

correspondientes clases conceptuales. Sin embargo, se diferencia de esta por el estereotipo <<navigation class>>. Además, una clase de navegación puede contener atributos de otras clases del modelo conceptual, siempre que la clase de navegación tenga una asociación con la clase de la que se presenta el o los atributos. Para diferenciar dichos atributos se coloca una barra inclinada a la derecha antes del nombre.

La navegación directa es representada por asociaciones en el modelo de espacio de navegación que provienen de la clase de navegación de origen. Por lo tanto, sus semánticas son diferentes de las asociaciones usadas en el modelo conceptual. Estas asociaciones son interpretadas como el enlace o vínculo entre la clase de navegación inicial y la clase de navegación final.

Figura 2.4: Representación de una clase de navegación



Fuente: (Koch, 2000)

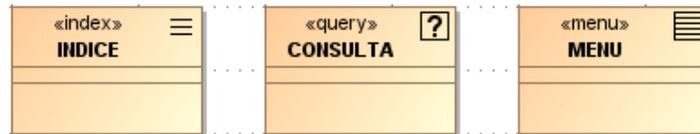
Modelo de estructura de navegación: amplía el modelo con un conjunto de estructuras de acceso necesarias para la navegación.

Describe como la navegación es soportada por elementos de acceso tales como índices, vistas guiadas, preguntas y menús. El resultado es un diagrama de clases UML construido con estereotipos los cuales están definidos según mecanismos de extensión UML.

Las primitivas de acceso son nodos de navegación adicionales requeridas para acceder a los objetos de navegación. Las primitivas de acceso son definidas

como estereotipos UML que son las siguientes: índices, vistas guiadas, consultas y menús. Los índices nos permiten el acceso directo a las instancias de la clase índice y utiliza el estereotipo <<index>> con su icono correspondiente.

Figura 2.5: Clase Índice, Consulta y Menú



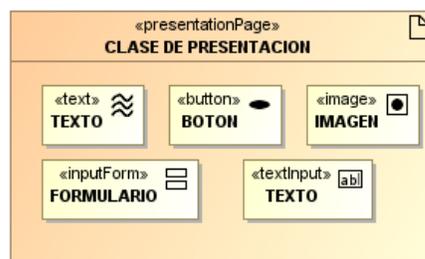
Fuente: (Koch, 2000)

Diseño de presentación: permite la especificación lógica de la aplicación web, basada sobre el modelo lógico, una representación física puede ser construida, dentro de este modelo se distinguen dos diferentes vistas:

Estructura de vista: muestra la estructura del espacio de presentación.

Interfaz de usuario: vista que presenta detalles acerca de los elementos de la interfaz de usuario dentro de las páginas

Figura 2.6: Clase de Presentación



Fuente: (Koch, 2000)

2.4.4. Fases De UWE

UWE cubre todo el ciclo de vida de aplicaciones web centrado en aplicaciones personalizadas y centralizadas.

Las fases de UWE son:

Captura, análisis y especificación de requisitos: en esta fase se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web. Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las necesidades de interfaz de usuario, así como requisitos adicionales. Centra el trabajo en el estudio de los casos de uso, la generación de glosarios y el prototipo de interfaz de usuario.

Diseño del Sistema: se basa en la especificación de requisitos del producto por el análisis de requerimientos, el diseño se define como estos requisitos se cumplirán y la estructura que debe darse a la aplicación web.

Codificación del software: en esta etapa se realizan las tareas de programación que es esencialmente llevar todo lo diseñado anteriormente a código fuente en el lenguaje de programación elegido.

Pruebas: las pruebas se realizan para verificar el correcto funcionamiento de las secciones de código.

Implementación: es el proceso en el cual el Sistema desarrollado es llevado al usuario por medio de la instalación de este en un servidor ya configurado para el hospedaje del Sistema.

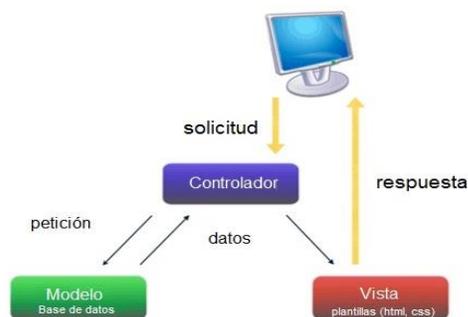
Mantenimiento: es el proceso de control, mejora y optimización del software desarrollado, también incluye la depuración de errores y defectos que pueden haberse filtrado de la fase de pruebas de control. (Galeano, 2012)

2.5. Herramientas de Desarrollo

2.5.1. Modelo Vista Controlador

Modelo Vista Controlador (MVC), es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

Figura 2.7: Modelo, Vista, Controlador



Fuente: (Smalltalk-88- MVC)

2.5.1.1. Características

De manera genérica, los componentes de MVC se podrían definir como sigue:

El Modelo: Es la representación de la información con la cual el Sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la vista aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al modelo a través del controlador.

El Controlador: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al modelo cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un Registro en una base de datos). También puede enviar comandos a su vista asociada si se solicita un cambio en la forma en que se presenta de modelo (por ejemplo, desplazamiento o scroll por un documento o por los diferentes Registros de una base de datos), por tanto se podría decir que el controlador hace de intermediario entre la vista y el modelo.

La Vista: Presenta el modelo (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho modelo la información que debe representar como salida. (Alvarez M. A., Que es MVC, 2014).

2.5.2. Gestor de Base De Datos MaríaDB

MariaDB es un Sistema de gestión de bases de datos. Se deriva de MySQL, una de las base de datos más importantes que ha existido en el mercado, utilizada para manejar grandes cantidades de información.

Para que se tenga una idea de la enorme capacidad para mover grandes cantidades de información, MySQL ha sido la base de datos utilizada por proyectos de internet de la índole de Facebook, Twitter y Wikipedia.

La simplicidad de la sintaxis permite crear bases de datos simples o complejos con mucha facilidad; es compatible con múltiples plataformas informáticas y está provista de una infinidad de aplicaciones que permiten acceder rápidamente a las sentencias de la gestión de base de datos.

Además, permite a los desarrolladores y diseñadores realizar cambios en los sitios web con sólo cambiar un archivo, (sin necesidad de modificar todo el código web) para que se ejecuten en toda la estructura de datos que se comparte en la red.

2.5.2.1. Utilidad del Sistema

La utilidad empresarial proviene de la capacidad del Sistema de gestión para manejar información relacional (temas o propósitos relacionados entre sí) multiusuario (diversos usuarios utilizando el Sistema simultáneamente) y multihilo (desde diversos procesadores).

Es un Sistema que permite, por ejemplo, llevar los Registros de los empleados, las listas de posibles clientes y proveedores, en una base de datos rápida, segura y potente.

Ahora bien: MariaDB es un sustituto de MySQL que incorpora las funcionalidades propias de MySQL e incluye otras mejoras, como la incorporación de nuevos motores de almacenamiento mucho más eficientes:

- Aria y XtraDB, desarrollados para ser los sustitutos de MyISAM e InnoDB respectivamente. Permiten ejecutar consultas más complejas y almacenarlas en caché y no en disco duro.
- FederatedX, para reemplazar a Federated.

- Oqgraph¹, para que el Sistema de base de datos soporte el uso de jerarquías de estructuras y graphs complejos.
- SphinxSE , para hacer búsquedas de texto bajo Sphinx.
- Cassandra Storage Engine, para acceder a un clúster de datos. Este motor se debe activar por separado, porque no viene instalado por defecto.

Además de los nuevos motores de almacenamiento mencionados, MariaDB incorpora otras mejoras de rendimiento y versiones de seguridad más rápidas y transparentes.

De la misma forma que ha ocurrido con MySQL, MariaDB es de código libre y está teniendo un formidable soporte de la comunidad de desarrolladores, aunque también cuenta con el soporte de Oracle.

La migración de MySQL a MariaDB es relativamente fácil y tiene la ventaja adicional de que MariaDB es compatible con todos los scripts PHP, al menos con WordPress, XenForo, phpBB, MyBB, SMF, Drupal, Vbulletin.

2.5.2.2. Utilidad del Sistema

La versión de desarrollo de MariaDB es la 10.0. Está construida sobre la versión 5.5, con algunas características de MySQL 5.6 y otras prestaciones nuevas no encontradas en ninguna otra versión anterior.

- ✚ Facilidad de uso
- ✚ Proporciona estadísticas de índices y tabla, para lo que añade nuevas tablas en information_schema y nuevas opciones a los comandos flush y show para identificar la causa en la carga del sgbd.

¹ **OQGRAPH** .- Graph es un programa diseñado para representar gráficamente funciones matemáticas en un Sistema de coordenadas

- ✚ Los comandos alter table y load data infile dejan de ser opacos e informan del progreso.
- ✚ La precisión para tipo de datos time, datetime, y timestamp ampliada al microsegundo.
- ✚ Introducidas características estilo NoSQL, como HandlerSocket que proporciona acceso directo a tablas InnoDB saltándose la capa SQL.
- ✚ Columnas dinámicas, que proporcionan al usuario columnas virtuales en las tablas.
- ✚ Las subqueries funcionan correctamente.

2.5.2.3. Prestaciones

- ✚ El optimizador de MariaDB que se encuentra en el núcleo de cualquier SGBD- funciona claramente más rápido con cargas complejas.
- ✚ En la replicación se han introducido sustanciosas mejoras, por ejemplo el “group commit for the binary log” que acelera la replicación hasta el doble.
- ✚ Eliminación de tablas. El acceso a tablas a través de views acelera el acceso.

2.5.2.4. Testeo

- ✚ Más juegos de test en la distribución.
- ✚ Parches para los tests.
- ✚ Distintas combinaciones de configuración y Sistema operativo para los tests.
- ✚ Eliminación de tests innecesarios, como "no testar la característica X si no la he incluido en mi ejecutable".
- ✚ Menos errores y alertas

- ✚ Los juegos de testeo han permitido reducir los errores sin introducir nuevos.
- ✚ Las alertas de compilación están relacionadas, y los desarrolladores las han intentado reducir. (Wikipedia, 2020)

2.6. Lenguaje De Programación Php

Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los Sistemas operativos y plataformas sin ningún costo.

PHP se considera uno de los lenguajes más flexibles, potentes y de alto rendimiento conocidos hasta el día de hoy, lo que ha atraído el interés de múltiples sitios con gran demanda de tráfico, como Facebook, para optar por el mismo como tecnología de servidor.

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

PHP es un acrónimo recursivo que significa PHP Pre Hypertext - processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdorf; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre. (Alvarez M. A., que es Php, 2001)

2.7. Métricas De Calidad

La calidad del software es una preocupación a la que se dedican muchos esfuerzos. Sin embargo, el software casi nunca es perfecto. Todo proyecto tiene como objetivo producir software de la mejor calidad posible, que cumpla, y si puede supere las expectativas de los usuarios.

- Es la aptitud de un producto o servicio para satisfacer las necesidades del usuario.
- Es la cualidad de todos los productos, no solamente de equipos sino también de programas.

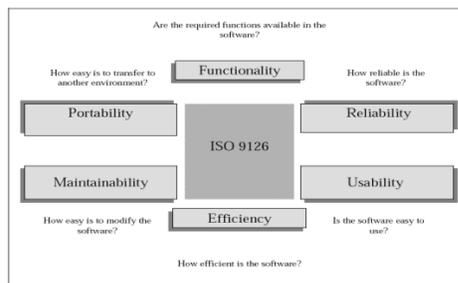
En el desarrollo de software, la calidad de diseño acompaña a la calidad de los requisitos, especificaciones y diseño del Sistema. La calidad de concordancia es un aspecto centrado principalmente en la implementación; Si la implementación sigue al diseño, y el Sistema resultante cumple con los objetivos de requisitos y de rendimiento, la calidad de concordancia es alta. (Pressman, 2010), define la calidad como la concordancia con los requisitos funcionales y de rendimiento explícitamente establecido, los estándares de desarrolló explícitamente documentados y las características implícitas que se esperan de todo software desarrollado profesionalmente. (PRESSMAN, 2002).

2.7.1. Introducción A la Norma Iso/lec 9126

La norma ISO/IEC 9126 es un modelo de calidad estándar para productos de software, donde se describen las diferentes características y subcaracterísticas que debe cumplir un Sistema de software para que pueda ser considerado como un Sistema de calidad. Además, este modelo también define una serie de métricas y se divide en dos partes:

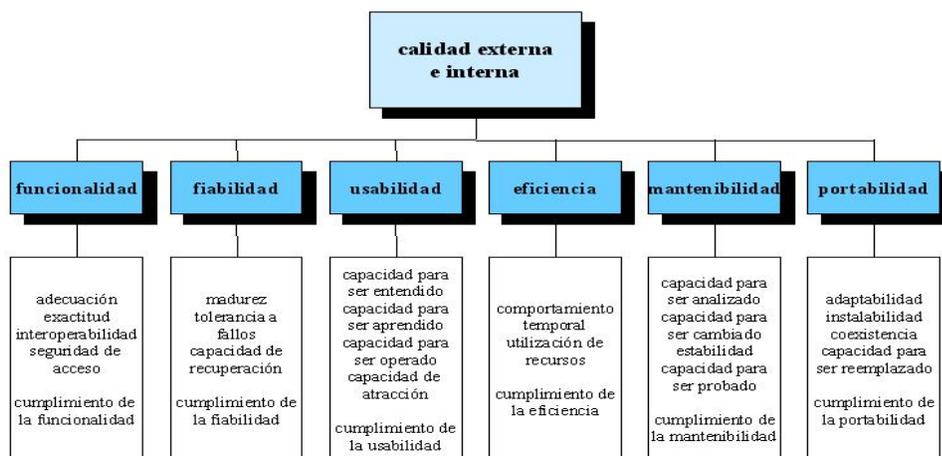
- Calidad externa e interna del producto de software-
- Calidad de uso del producto.

Figura 2.8: Características de la norma ISO 9126



Fuente: [http://mena.com.mx/gonzalo/maestria/calidad/presenta/iso_9126-3/]

Figura 2.9: Subcaracterísticas de la norma ISO 9126



Fuente: [http://mena.com.mx/gonzalo/maestria/calidad/presenta/iso_9126-3/]

2.7.2. Características De La Norma Iso/lec 9126

Funcionalidad: se refiere a un conjunto de funciones y propiedades que tratan de satisfacer las necesidades. Sus atributos den adecuación, exactitud, interoperabilidad y seguridad

Los puntos de función se describen como medidas básicas desde donde se calculan métricas de productividad, estos se utilizan de las siguientes dos formas:

- Como una variable de estimación que se utiliza para dimensionar cada elemento del software
- Como métricas de líneas base recopilada de proyectos anteriores y utilizados junto con variables de estimación para desarrollar proyecciones de costo y esfuerzo.

Para estimaciones de PF la descomposición funciona de la siguiente manera:

Tabla 2.1: Dominios de información de puntos de función

Dominio de información	Descripción
Número de entradas del usuario	Se encuentra cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada
Número de salidas del usuario	Se cuenta cada salida que proporciona al usuario información orientada a la aplicación, en este contexto la salida se refiere a informes, pantallas, mensajes de error y demás. Los elementos de datos particulares

	dentro de un informe no se encuentran de forma separada.
Número de peticiones al usuario	Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado.
Numero de archivos	Se cuenta cada archivo maestro lógico (esto es un grupo lógico de datos que se puede ser una parte de una gran base de datos o un archivo independiente)
Numero de interfaces externas	Se cuenta todas las interfaces legibles por la maquina (por ejemplo archivos de datos de disco), que se utilizan para transmitir información a otros Sistemas.

Fuente: [Pressman, 2002]

Los puntos de función se calculan completando la siguiente tabla:

Tabla 2.2: Factores de ponderación

Parámetros de medición	Cuenta	Factor de ponderación			Resultado
		Simple	Medio	Complejo	
Número de entradas de usuario	N_1	3	4	6	$N_1 \cdot \text{factor}$
Número de salidas de	N_2	4	5	7	$N_2 \cdot \text{factor}$

usuario					
Número de peticiones de usuario	N ₃	3	4	6	N ₃ *factor
Numero de archivos	N ₄	7	10	15	N ₄ *factor
Numero de interfaces externas	N ₅	5	7	10	N ₅ *factor
Cuenta total					$\sum(N_i*\text{factor})$

Fuente: [Pressman, 2002]

Para calcular puntos de función (PF) se utiliza la siguiente relación:

$$PF = \text{cuenta} - \text{total} * [0,65 + 0,01 * \sum (F_i)]$$

En donde cuenta – total es la suma de todas las entradas de los factores de ponderación obtenidas en la tabla 2.2.

F_i (i = 1 a 14), son valores de ajuste de complejidad según las respuestas a las siguientes preguntas:

Tabla 2.3: Valores de ajuste de la complejidad

Importancia	0%	20%	40%	60%	80%	100%
Escala	No	Incidental	Moderado	Medio	Significativ	Esencial
Factor	0	1	2	3	4	5
¿Requiere el Sistema copias de seguridad y						

recuperación fiables?						
¿Se requiere comunicación de datos?						
¿Existen funciones de procesamiento distribuido?						
¿Es crítico el rendimiento?						
¿Se ejecuta el Sistema en un entorno operativo existente y fuertemente utilizado?						
¿Requiere el Sistema entrada de datos interactiva?						
¿Requiere el Sistema entrada de datos interactivos que las transacciones de entrada se lleven a cabo sobre múltiples entradas u operaciones?						
¿Se actualizan los archivos maestros de forma interactiva?						
¿Son complejas las entradas, las salidas, los archivos o peticiones?						
¿Es complejo el procesamiento interno?						
¿Se ha diseñado código para ser reutilizable?						
¿Están incluidas en el diseño la conversión y la instalación?						
¿Se ha diseñado el Sistema para soportar múltiples instalaciones en diferentes organizaciones?						
¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?						

Fuente: [Pressman, 2002]

También la métrica de adecuación según la siguiente tabla:

Tabla 2.4: Métrica de adecuación

Nombre:	Completitud de implementación funcional
Propósito:	Cómo de completa es la implementación funcional.
Método de aplicación:	Contar las funciones faltantes detectadas en la evaluación y comparar con el número de funciones descritas en la especificación de requisitos.
Medición, fórmula:	$X = 1 - A/B$ A = número de funciones faltantes B = número de funciones descritas en la especificación de requisitos
Interpretación:	$0 \leq X \leq 1$ Entre más cercano a 1, más completa.
Tipo de escala:	Absoluta
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	Especificación de requisitos Diseño Código fuente Informe de revisión

Fuente: [Pressman, 2002]

Fiabilidad: se refiere a un conjunto de atributos que miden la capacidad que tiene el software para mantener un nivel de rendimiento óptimo, bajo determinadas condiciones y durante un periodo de tiempo determinado. Sus atributos son madurez, tolerancia a fallos y la capacidad de recuperación ante un fallo.

Para que un Sistema sea fiable, se debe garantizar un nivel de seguridad. La seguridad se subdivide a su vez en confidencialidad, autenticación, control de acceso, integridad de los datos y responsabilidad de los usuarios. Para garantizarla se ofrecen distintos mecanismos como certificados digitales y sockets (SSL) y hace un tratamiento adecuado de la información personal y privada de los usuarios.

La confiabilidad de un Sistema se calcula mediante la siguiente relación

Probabilidad de hallar una falla: $P(T \leq t) = F(t)$
 Probabilidad de no hallar una falla: $P(T > t) = 1 - F(t)$
 Con: $F(t) = F_c * (e^{-\lambda/7 * 12})$

Dónde:

$F_c = 0,87$: funcionalidad del Sistema

$\lambda = 1$: tasa de fallos dentro de un mes

También utilizando la métrica de madurez de la siguiente tabla:

Tabla 2.5: Métrica de madurez

Nombre:	Suficiencia de las pruebas
Propósito:	Cuántos de los casos de prueba necesarios están cubiertos por el plan de pruebas.
Método de aplicación:	Contar las pruebas planeadas y comparar con el número de pruebas requeridas para obtener una cobertura adecuada.
Medición, fórmula:	$X = A/B$ $A =$ número de casos de prueba en el plan $B =$ número de casos de prueba requeridos
Interpretación:	$0 \leq X$ Entre X sea mayor, mejor la suficiencia.
Tipo de	Absoluta

escala:	
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	A proviene del plan de pruebas B proviene de la especificación de requisitos
ISO/IEC 12207 SLCP:	Aseguramiento de Calidad Resolución de problemas Verificación
Audiencia:	Desarrolladores Mantenedores

Fuente: [http://mena.com.mx/gonzalo/maestria/calidad/iso_91263/]

Usabilidad: se refiere a un conjunto de atributos que miden el esfuerzo cognitivo necesario que deben realizar los usuarios para utilizar el Sistema de software. Sus atributos son comprensión, curva de aprendizaje y operatividad.

Utilizando la métrica de entendibilidad según la siguiente tabla:

Tabla 2.6: Métrica de entendibilidad

Nombre:	Funciones evidentes
Propósito:	Qué proporción de las funciones del Sistema son evidentes al usuario.
Método de aplicación:	Contar las funciones evidentes al usuario y comparar con el número total de funciones.
Medición, fórmula:	$X = A/B$ A = número de funciones (o tipos de funciones) evidentes al usuario B = total de funciones (o tipos de funciones)

Interpretación:	0 <= X <= 1 Entre más cercano a 1, mejor.
Tipo de escala:	Absoluta
Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Especificación de requisitos Diseño Informe de revisión
ISO/IEC 12207 SLCP:	Verificación Revisión conjunta

Fuente: [http://mena.com.mx/gonzalo/maestria/calidad/iso_9126-3/]

Eficiencia: se refiere a un conjuntos de atributos que miden la relación entre el rendimiento del software y la cantidad de recursos utilizados, dad una situación determinada. Sus atributos son tiempo de respuesta y recursos utilizados

La eficiencia se entiende como la capacidad del Sistema para proporcionar tiempos de respuesta, tiempos de proceso y potencia apropiados bajo condiciones determinadas.

Utilizando la métrica de comportamiento en el tiempo según la siguiente tabla:

Tabla 2.7: Métrica de comportamiento en el tiempo

Nombre:	Tiempo de respuesta
Propósito:	Cuál es el tiempo estimado para completar una tarea.
Método de aplicación:	Evaluar la eficiencia de las llamadas al SO y a la aplicación. Estimar el tiempo de respuesta basado en ello. Puede medirse: Todo o partes de las especificaciones de diseño.

	Probar la ruta completa de una transacción. Probar módulos o partes completas del producto. Producto completo durante la fase de pruebas.
Medición, fórmula:	X = tiempo (calculado o simulado)
Interpretación:	Entre más corto, mejor.
Tipo de escala:	Proporción
Tipo de medida:	X = time
Fuente de medición:	Sistema operativo conocido Tiempo estimado en llamadas al Sistema
ISO/IEC 12207 SLCP:	Verificación Revisión conjunta
Audiencia:	Desarrolladores Requeridores

Fuente: [http://mena.com.mx/gonzalo/maestria/calidad/iso_9126-3/]

Mantenibilidad: se refiere a un conjunto de atributos relacionados con el esfuerzo necesario para realizar determinadas modificaciones en el producto. Sus atributos son la capacidad de ser analizado, capacidad para ser modificado, estabilidad y capacidad para ser probado.

El estándar IEEE 982.1 sugiere un índice de madurez del software (IMS) que proporciona una indicación de la estabilidad del producto de software, se determina con la siguiente relación

$$IMS = [M_T - (F_c + F_a + F_d)] / M_T$$

Dónde:

M_T = número de módulos en la versión actual.

F_c = número de módulos en la versión actual que se han cambiado.

F_a = número de módulos en la versión actual que se han añadido.

F_d = número de módulos en la versión anterior que se han borrado en la versión actual.

A medida que el IMS se aproxima a 1.0 se logra una madurez estable.

Utilizando la métrica de cambiabilidad según la siguiente tabla:

Tabla 2.8: Métrica de cambiabilidad

Nombre:	Registrabilidad de cambios
Propósito:	¿Se registran adecuadamente los cambios a la especificación y a los módulos con comentarios en el código?
Método de aplicación:	Registrar la proporción de información sobre cambios a los módulos
Medición, fórmula:	$X = A/B$ A = número de cambios a funciones o módulos que tienen comentarios confirmados B = total de funciones o módulos modificados
Interpretación:	$0 \leq X \leq 1$ Entre más cercano a 1, más registrable. 0 indica un control de cambios deficiente o pocos cambios y alta estabilidad.
Tipo de escala:	Absoluta
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de	Sistema de control de configuraciones

medición:	Bitácora de versiones Especificaciones
ISO/IEC 12207 SLCP:	Verificación Revisión conjunta
Audiencia:	Desarrolladores Mantenedores Requeridores

Fuente: [http://mena.com.mx/gonzalo/maestria/calidad/iso_9126-3/]

Portabilidad: son atributos con la capacidad del software de ser transferido de un entorno a otro. Sus atributos son adaptabilidad, capacidad de instalación, coexistencia y capacidad de reemplazamiento (Prieto, 2017)

Utilizando la métrica de conformidad de transportabilidad según la siguiente tabla:

Tabla 2.9: Métrica de conformidad de transportabilidad

Nombre:	Conformidad de transportabilidad
Propósito:	Cómo de transportable es el producto según las regulaciones, estándares y convenciones aplicables.
Método de aplicación:	Contar los artículos encontrados con conformidad y comparar con el número de artículos en la especificación que requieren conformidad.
Medición, fórmula:	$X = A/B$ A = número de artículos implementados de conformidad B = total de artículos que requieren conformidad
Interpretación:	$0 \leq X \leq 1$ Entre más cercano a 1, más completa.
Tipo de escala:	Absoluta

Tipo de medida:	X = count/count A = count B = count
Fuente de medición:	Especificación de conformidad y estándares, convenciones y regulaciones relacionados. Diseño Código fuente Informe de revisión
ISO/IEC 12207 SLCP:	Verificación Revisión conjunta
Audiencia:	Requeridores Desarrolladores

Fuente: [http://mena.com.mx/gonzalo/maestria/calidad/iso_9126-3/]

2.8. Seguridad

La seguridad es una disciplina que se encarga de proteger la integridad y la privacidad de la información almacenada en un sistema.

“La seguridad del software se relaciona por completo con la calidad . Debe pensarse en seguridad, confiabilidad, disponibilidad y dependencia, en la fase inicial, en la de diseño, en la de arquitectura, pruebas y codificación, durante todo el ciclo de vida del software”. Según (McGraw, 2016)

En pocas palabras, el software que no tiene alta calidad es fácil de penetrar por parte de intrusos y en consecuencia, el software de mala calidad aumenta indirectamente el riesgo de la seguridad, con todos los costos y problemas que eso conlleva.

También plantea que la base de los problemas de seguridad son la conectividad, la complejidad y la extensibilidad de los sistemas actuales y su defunción está dada bajo 2 conceptos orientados dentro los objetivos.

1. La seguridad de un producto desarrollado se orienta a la búsqueda de que dicho producto continúe funcionando correctamente ante ataques maliciosos.
2. La seguridad del Software en construcción se orienta a la resistencia proactiva de posibles ataques

2.8.1. Tipos De Software De Seguridad

Sin duda, el tipo de software de seguridad más conocido son los programas antivirus, que se encargan de detectar y eliminar virus informáticos. Un buen programa antivirus dispone de un archivo de firmas de virus que se actualiza automáticamente y detecta virus nuevos. Este tipo de actualización se realiza periódicamente, varias veces al día. El software de seguridad suele venderse en las denominadas suites. Son paquetes compuestos de:

- Programa antivirus.
- Cortafuegos.
- Filtro anti spam.
- Software para filtrar contenidos.
- Software contra publicidad no deseada.
- Control de sitios web.

2.8.2. Cómo Garantizar La Seguridad Del Software

En primer lugar, si se piensa en materia de seguridad del software, un aspecto básico y fundamental es evitar las licencias piratas. Se estima que más del 60 por ciento de las computadoras con programas ilegales presenta algún tipo de actividad relacionada con instalaciones maliciosas. Es importante, además, tener en cuenta la protección de derechos de autor de cada programa para evitar un uso inadecuado.

De cualquier forma, debido a que los sistemas informáticos constituyen un bien de la empresa, siempre hay que tener en la mira mecanismos para protegerlos.

Los parches de seguridad son una solución ya que resultan efectivos y se adaptan a vulnerabilidades específicas.

Si la empresa desarrolla su propio programa, un aspecto clave son las pruebas, esto quiere decir realizar ensayos de funcionamiento en entornos controlados con el objetivo de detectar posibles fallas. Si dichos experimentos arrojan errores, entonces hay un margen para evitar que el producto defectuoso llegue al cliente o a la vida cotidiana de la empresa. Según (Editorial, 2019)

2.9. Costo Del Proyecto

Existe una gran variedad de métricas para determinar el costo de un proyecto no solo de costo sino también para determinar el esfuerzo de un proyecto, el alcance del mismo y la productividad de sus programadores.

El manejador del costo principal para un proyecto de desarrollo es sin duda el tamaño del producto. La medida del tamaño debe ser que este en relación directa con el esfuerzo de desarrollo, por lo que las métricas de tamaño tratan de considerar los aspectos que influyen en el costo, como tecnología, tipos de recursos y complejidad. Existen técnicas para la estimación de costos para ello se requiere de un buen acceso a la información y experiencia.

2.9.1. Modelo Cocomo

El modelo constructivo de costos cócono es un modelo desarrollado por Barry M Boehm, se engloba en el grupo de los modelos algorítmicos que tratan de establecer una relación matemática la cual permite estimar el esfuerzo y tiempo requerido para desarrollar un producto.

Pertenece a la categoría de modelos de subestimaciones basados en estimaciones matemáticas. Está orientado a la magnitud del producto final, midiendo el "tamaño" del proyecto, en líneas de código principalmente.

Cocomo define tres modos de desarrollo o tipos de proyecto:

- Orgánico: proyectos relativamente sencillos. Se tiene experiencia en proyectos similares y se encuentra un entorno estable.
- Semi-acoplado o semi encajado: proyectos intermedios en complejidad y tamaño. La experiencia en este tipo de trabajos es variable y las restricciones intermedias.
- Empotrado o modo rígido: proyectos bastante complejos, en los que apenas se tiene experiencia, se trabaja con requisitos muy restrictivos y de gran vitalidad.

2.9.1.1. Modelos De Estimación

Las ecuaciones que se utilizan son:

- $E = a(KI)^b * m(X)$ persona-mes
 - $Tdev = c(E)^d$ en meses
 - $P = E/Tdev$, en personas

Dónde:

E es el esfuerzo requerido por el proyecto, en persona-mes.

Tdev es el tiempo requerido por el proyecto, en meses.

P es el número de personas requerido por el proyecto.

a, b, c y d son constantes con valores definidos en una tabla, según cada submodelo.

KI es la cantidad de líneas de código, en miles.

m(X) Es un multiplicador que depende de 15 atributos.

- **Modelo Básico**

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

Tabla 2.10: constantes modo básico

MODO	a	b	c	d
Orgánico	2.40	1.05	2.50	0.38
Semilibre	3.00	1.12	2.50	0.35
Rígido	3.60	1.20	2.50	0.32

Fuente: [es.wikipedia.org]

Estos valores son para las fórmulas:

- Personas necesarias por mes para llevar adelante el proyecto (MM) = $a \cdot (KI)^b$
- Tiempo de desarrollo del proyecto (TDEV) = $c \cdot (MM)^d$
- Personas necesarias para realizar el proyecto (CosteH) = $MM/TDEV$
- Costo total del proyecto (CosteM) = CosteH * Salario medio entre los programadores y analistas.

Se puede observar que a medida que aumenta la complejidad del proyecto (modo), las constantes aumentan de 2.4 a 3.6, que corresponde a un incremento del esfuerzo del personal. Hay que utilizar con mucho cuidado el modelo básico puesto que se obvian muchas características del entorno

- **Modelo Intermedio**

Este añade al modelo básico quince modificadores opcionales para tener en cuenta en el entorno de trabajo, incrementando así la precisión de la estimación.

Para este ajuste, al resultado de la fórmula general se lo multiplica por el coeficiente surgido de aplicar los atributos que se decidan utilizar.

Los valores de las constantes a reemplazar en la fórmula son:

Tabla 2.11: constantes modo intermedio

MODO	a	b
Orgánico	3.20	1.05
Semilibre	3.00	1.12
Rígido	2.80	1.20

Fuente: [es.wikipedia.org]

Se puede observar que los exponentes son los mismos que los del modelo básico, confirmando el papel que representa el tamaño; mientras que los coeficientes de los modos orgánico y rígido han cambiado, para mantener el equilibrio alrededor del semilibre con respecto al efecto multiplicador de los atributos de coste.

- Atributos

Cada atributo se cuantifica para un entorno de proyecto. La escala es muy baja - bajo - nominal - alto - muy alto - extremadamente alto. Dependiendo de la calificación de cada atributo, se asigna un valor para usar de multiplicador en la fórmula (por ejemplo, si para un proyecto el atributo DATA es calificado como muy alto, el resultado de la fórmula debe ser multiplicado por 1000).

El significado de los atributos es el siguiente, según su tipo:

- De software
 - Rely: garantía de funcionamiento requerida al software. Indica las posibles consecuencias para el usuario en el caso que existan defectos en el producto. Va desde la sola inconveniencia de corregir un fallo (muy bajo) hasta la posible pérdida de vidas humanas (extremadamente alto, software de alta criticidad).
 - Data: tamaño de la base de datos en relación con el tamaño del programa. El valor del modificador se define por la relación: d/k , donde D corresponde al tamaño de la base de datos en bytes y K es el tamaño del programa en cantidad de líneas de código.
 - Cplx: representa la complejidad del producto.
- De hardware
 - Time: limitaciones en el porcentaje del uso de la CPU.
 - Stor: limitaciones en el porcentaje del uso de la memoria.
 - Virt: volatilidad de la máquina virtual.
 - Turn: tiempo de respuesta requerido.
- De personal
 - Acap: calificación de los analistas.
 - Aexp: experiencia del personal en aplicaciones similares.

- Pcap: calificación de los programadores.
 - Vexp: experiencia del personal en la máquina virtual.
 - Lexp: experiencia en el lenguaje de programación a usar.
- De proyecto
 - Modp: uso de prácticas modernas de programación.
 - Tool: uso de herramientas de desarrollo de software.
 - Sced: limitaciones en el cumplimiento de la planificación.

El valor de cada atributo, de acuerdo a su calificación:

Tabla 2.12: variables factor de ajustes del esfuerzo

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de software						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de hardware						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	

Tiempo de respuesta		0,87	1,00	1,07	1,15	
Atributos de personal						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
Atributos del proyecto						
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82	
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,22	1,08	1,00	1,04	1,10	

Fuente: [es.wikipedia.org]

- **Modelo Detallado**

Presenta principalmente dos mejoras respecto al anterior:

- Los factores correspondientes a los atributos son sensibles o dependientes de la fase sobre la que se realizan las estimaciones. Aspectos tales como la experiencia en la aplicación, utilización de herramientas de software, etc., tienen mayor influencia en unas fases que en otras, y además van variando de una etapa a otra.

- Establece una jerarquía de tres niveles de productos, de forma que los aspectos que representan gran variación a bajo nivel, se consideran a nivel módulo, los que representan pocas variaciones, a nivel de subSistema; y los restantes son considerados a nivel Sistema. (Cocomo, 2017)

3. Marco Aplicativo

3.1. Introducción

En este punto se explicara de forma clara y concisa los aspectos relacionados con las características, organización y descripción de las funciones y los diferentes procesos que existen dentro de la entidad.

También se aplicara la metodología de desarrollo y las diferentes tecnologías aplicadas para el proyecto.

3.2. Diagnóstico De La Situación Actual

Para realizar las fases de modelado se debe revisar la situación actual de la Academia de música Mozart en cuanto a las actividades realizadas al proceso de Registro y Seguimiento académico de estudiantes que toman cursos de uno o más instrumentos.

3.2.1. Descripción De Funciones

- Estudiante: su función con respecto al proceso de inscripción es la de inscribirse a la especialidad de instrumento que desea aprender, solicita información referente a sus notas.
- Docente: su función con respecto al proceso de inscripción es la de verificar cuantos estudiantes hay en el curso que impartirá de acuerdo a la especialidad en instrumento y de brindar la nota que le corresponde a cada estudiante.

- **Secretaria:** su función es colaborar en el área administrativa y académica, es la que realiza la función de inscripción de los estudiantes, registra los contratos de los docentes de acuerdo a la especialidad al tipo de instrumento de enseñanza, también realiza los reportes e informes de todos los Registros y estudiantes inscritos.
- **Director:** su función es el de administrar la Academia revisa reportes de estudiantes inscritos, docentes contratados, lanza convocatorias revisar la parte contable.

3.2.2. Proceso De Inscripción

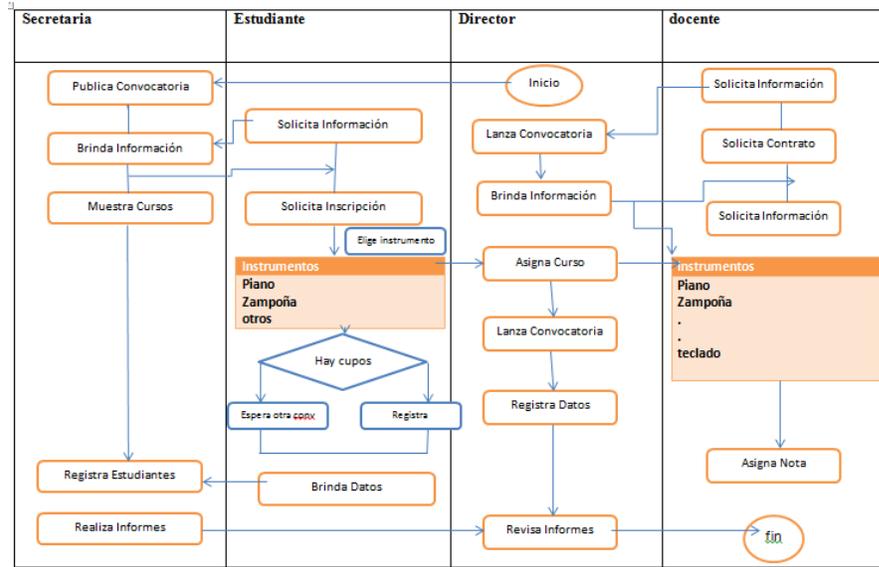
- **Estudiante**
 - Solicita información de cursos activos, instrumentos activos, tiempo de duración, horarios disponibles y costos.
 - Solicita inscripción a uno o más cursos de instrumentos elegidos.
 - Solicita asignación de horario correspondiente a los días elegidos.
 - Solicita materiales y libros de apoyo a los cursos que se inscriben de acuerdo al tipo de instrumento que cursara.
 - Solicita certificado de culminación de curso de un determinado instrumento.
- **Docente**
 - Solicita información de los cursos, horarios, instrumentos.
 - Solicita contrato para el curso que dará de acuerdo a su especialidad con un determinado instrumento.
 - Solicita lista de estudiantes de su curso a asistir.
 - Solicita los horarios y días que correspondan a su especialidad.
 - Llena el acta de notas de su curso asignado.

- Secretaria
 - Encargado de la inscripción de los estudiantes.
 - Encargado de la asignación de horarios a los cursos en diferentes especialidad de instrumento Musical.
 - Encargado del cobro del total o en planes del curso.
 - Registra asignación de materias con sus respectivos docentes y horarios.
 - Registro y actualización de información de estudiantes y docentes.

- Director
 - Encargado de sacar y lanzar convocatoria de cursos disponibles de los diferentes instrumentos.
 - Encargado de revisar y aprobar los contratos con los docentes de la Academia
 - Encargado de la parte contable de la Academia como ser los costos de los cursos y materiales de apoyo.
 - Encargado de revisar reportes e informes de los Registro de los estudiantes.

En la siguiente Figura se muestra la situación actual:

Figura 3.1: Situación Actual



Fuente: [Elaboración propia]

3.3. Ingeniería de Requerimientos

Esta parte es fundamental para el desarrollo del Sistema, en este sentido se describe lo siguiente:

- **Entrevista:** Se realizaron entrevistas con el director y Secretaria de la Academia llegando a la conclusión de que el Registro y Seguimiento académico de estudiante no coadyuva a la toma de decisiones.
- **Observación:** La Academia de música Mozart tiene varios cursos de instrumento Musical en diferentes horarios y días a elección de los estudiantes lo cual genera perdida de información.
- **Documentación:** Se tuvo acceso a la documentación para la elaboración del Sistema y sus pruebas. Documentos como ser los formatos establecidos para la boleta de inscripción, historial académico, lista de

cursos por instrumento, las diferentes listas de estudiantes, horarios, listas de docentes, actas de calificaciones, recibos de pagos de los cursos y también la información de los estudiantes.

3.4. Aplicación de la Metodología UWE

3.4.1. Fase Captura, Análisis y Especificación de Requisitos

En esta fase se adquieren, reúnen especifican las características funcionales y no funcionales que debería cumplir la aplicación web.

3.4.2. Definición De Actores

- Secretaria: Encargada del Registro de estudiantes y docentes e instrumentos de la Academia de música Mozart.
- Estudiante: Solicita información, inscripción, para cursar una especialización de instrumento elegido.
- Docente: Solicita información de los cursos, solicita contrato, realiza la actualización de notas en el acta de calificaciones y recibe información de estudiantes asignados a su materia de los cursos que impartirá.
- Administrador: Lanza convocatoria revisa los reportes de cantidad de estudiantes inscritos por cursos, niveles, gestiones, tipo de instrumentos.

3.4.3. Requerimientos Funcionales

- Control de usuarios Administrador, Secretaria y Docentes.

- Registro de inscripción de estudiantes.
- Registro de Docentes.
- Registro de Instrumento Musical.
- Asignación de Instrumento Musical.
- Asignación de Docentes.
- Reportes de estudiantes inscritos.
- Reporte de acta calificaciones.
- Reporte de tipo de pago.

3.4.4. Definición De Procesos

- Secretaria
 - Ingreso al Sistema: Podrá ingresar al Sistema con un usuario y contraseña asignado por el director de la Academia.
 - Registro e Inscripción: Podrá realizar el Registro de estudiantes y docentes nuevos, actualización, eliminación. También se realiza la inscripción de estudiantes y docentes antiguos para luego ser asignados a un curso.
 - Asignación de cursos: realiza la asignación de un curso de instrumento a estudiantes y docentes de acuerdo al horario elegido por el estudiante.
 - Asignación de Instrumentos: realiza la asignación de un instrumento a estudiantes y docentes de acuerdo al horario que corresponde a cada estudiante y docente.
 - Generar reportes: podrá generar reportes como ser; Registro de estudiantes inscritos, Registro de docentes activos, Registro de instrumentos activos, Registro de tipo de pago realizado por el estudiante.

- Docente
 - Ingresar al Sistema: podrá ingresar al Sistema con un usuario y contraseña.
 - Ver Perfil: podrá ver su perfil, subir su foto.
 - Ver Curso: podrá ver los cursos asignados a su persona.
 - Ver Lista de Estudiantes: podrá ver los estudiantes registrado en su curso.
 - Ver Acta de Calificaciones: podrá asignar nota al estudiante que este cursando con su persona.

- Estudiante
 - Ingresar Al sistema: El estudiante podrá ver la página web desde cualquier navegador.
 - Ver Convocatoria: podrá ver las convocatorias lanzadas y activas.
 - Ver Docentes: el estudiante podrá visualizar los docentes asignados a los diferentes cursos

- Administrador (Director)
 - Ingresar Al Sistema: El Administrador podrá ingresar al Sistema con un usuario y contraseña.
 - Administración de usuarios: El Administrador podrá realizar el Registro de nuevos usuarios, actualización de datos de usuario, eliminar usuarios.
 - Administración de Instrumentos: El Administrador podrá registrar los instrumentos y asignarlos a un curso.
 - Administración de Convocatoria: podrá lanzar las convocatorias para los distintos cursos de instrumentos activos.
 - Administración de Inscripciones: podrá ver y hacer Seguimiento al estudiante inscrito en un curso de la Academia.

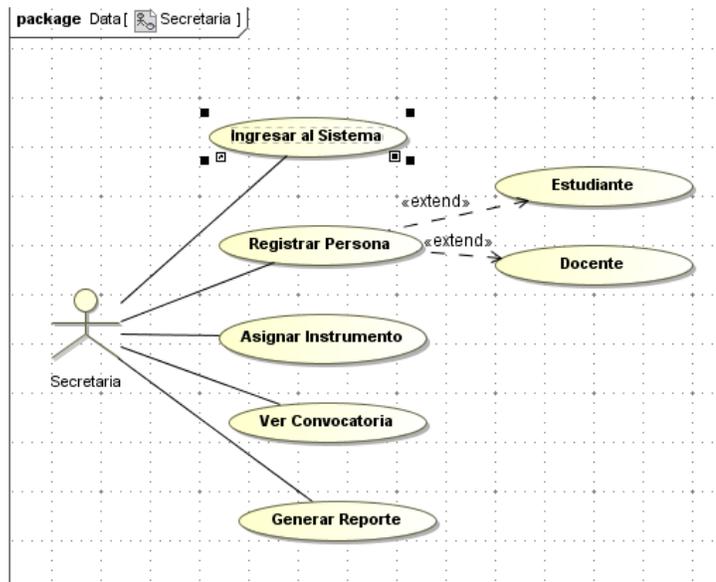
- Generar reportes: El Administrador podrá generar reportes como ser; Registro de estudiantes inscritos, Registro de docentes activos, Registro de instrumentos activos, Registro de tipo de pago realizado por el estudiante.

3.4.5. Modelo De Casos De Uso

En este punto se plasma el análisis de requerimientos del Sistema mediante el diseño de casos de uso, que describe el comportamiento del Sistema frente a las acciones de los actores del mismo, funcionamiento del Sistema y además elementos que permiten la abstracción del problema.

A continuación, se realiza el modelamiento donde se puede apreciar cómo interactúan los actores sobre los casos de uso.

Figura 3.2: Caso de Uso Secretaria



Fuente: [Elaboración propia]

Tabla 3.1: Especificación de Caso De Uso Secretaria

Caso de uso	Ingresar al Sistema
Objetivo	Describe el proceso de ingresar al Sistema web
Precondiciones	Ninguna
Actores	Secretaria
Secuencia	Ingresar al Sistema web Inserta nombre de usuario, contraseña e ingresa El Sistema comprueba al usuario y accede al Sistema
Secuencia Alternativa	Ingresar al Sistema web Inserta nombre de usuario, contraseña, tipo de usuario e ingresa El Sistema comprueba al usuario y este no accede, el Sistema vuelve a la raíz

Fuente: [Elaboración propia]

Tabla 3.2: Especificación de casos de uso Registrar Persona

Caso de uso	Registrar Persona
Objetivo	Describe el proceso para el Registro de docente y estudiante
Precondiciones	La Secretaria debe ingresar al Sistema con usuario y contraseña
Actores	Secretaria
Secuencia	Selecciona la opción Registro el Sistema pide datos para llenar ya sea a estudiante o docente
Secuencia	Ninguna

Alternativa	
--------------------	--

Fuente: [Elaboración propia]

Tabla 3.3: Especificación de Caso De Uso Asignar Instrumentos

Caso de uso	Asignar Instrumentos
Objetivo	Describe los instrumentos registrados para asignar a un estudiante o docente
Precondiciones	La Secretaria debe ingresar al Sistema con usuario y contraseña
Actores	Secretaria
Secuencia	<p>Selecciona la opción asignar Instrumento</p> <p>El Sistema muestra un formulario para asignar instrumento a un estudiante</p> <p>El usuario selecciona el tipo de instrumento que eligió el estudiante</p> <p>El Sistema muestra el instrumento elegido</p>
Secuencia Alternativa	<p>Selecciona la opción instrumentos</p> <p>El Sistema muestra un formulario para la selección de instrumentos</p> <p>La Secretaria selecciona el instrumento elegido por estudiante</p> <p>El Sistema muestra el instrumento elegido</p>

Fuente: [Elaboración propia]

Tabla 3.4: Especificación De Caso Uso Ver Convocatoria

Caso de uso	Ver Convocatoria
--------------------	-------------------------

Objetivo	Describe el proceso de Visualizar los cursos lanzados en convocatoria
Precondiciones	La Secretaria debe ingresar al Sistema con usuario y contraseña
Actores	Secretaria
Secuencia	Selecciona la Asignar Convocatoria El Sistema muestra las convocatorias existentes
Secuencia Alternativa	Ninguna

Fuente: [Elaboración propia]

Tabla 3.5: Especificación de Caso De Uso Generar Reportes

Caso de uso	Generar Reportes
Objetivo	Genera reportes como ser boletas de inscripción, Registros de instrumento asignación de cursos y horarios
Precondiciones	La Secretaria debe ingresar al Sistema con usuario y contraseña
Actores	Secretaria
Secuencia	Selecciona la opción Reportes y muestra un listado del tipo de reporte a generar para luego imprimir
Secuencia Alternativa	Ninguna

Fuente: [Elaboración propia]

Tabla 3.6: Especificación de Caso De Uso Estudiante

Caso de uso	Estudiante
Objetivo	Describe el proceso de Registro a los estudiantes
Precondiciones	La Secretaria debe ingresar al Sistema con usuario y contraseña
Actores	Secretaria
Secuencia	<p>Selecciona la opción Inscripción</p> <p>El Sistema muestra una ventana para el Registro de datos del estudiante.</p> <p>Se busca al estudiante y se le asigna una matricula</p> <p>Se busca al estudiante se le asigna un horario y un curso</p>
Secuencia Alternativa	<p>Selecciona la opción inscripción</p> <p>El Sistema muestra los estudiantes registrados y que el periodo de avance</p>

Fuente: [Elaboración propia]

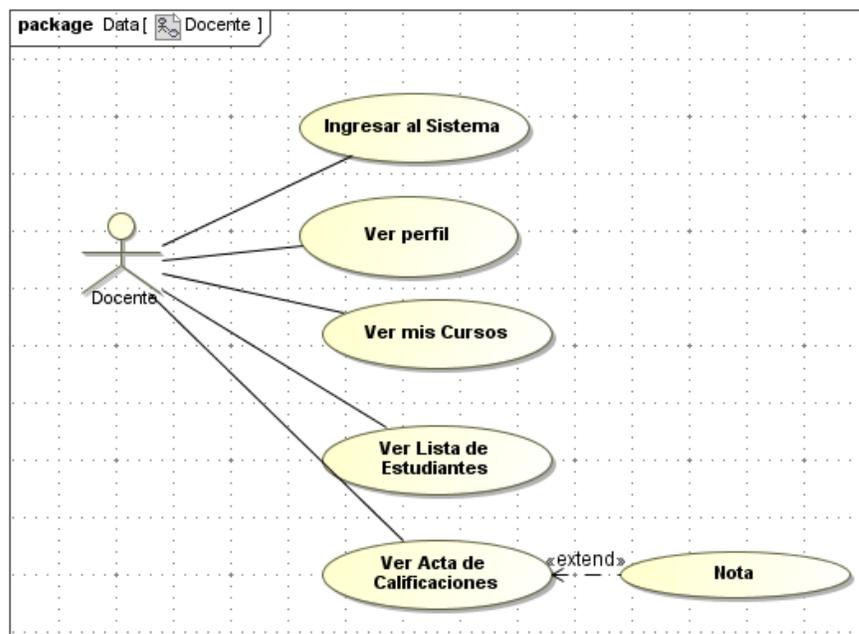
Tabla 3.7: Especificación De Caso De Uso Docente

Caso de uso	Docente
Objetivo	Describe el proceso de Registro de curso de Instrumentos
Precondiciones	La Secretaria debe ingresar al Sistema con usuario y contraseña
Actores	Secretaria

Secuencia	Selecciona la opción Registro e inscripción El Sistema inscribe al estudiante al curso seleccionado
Secuencia Alternativa	Ninguna

Fuente: [Elaboración propia]

Figura 3.3: Caso de Uso Docente



Fuente: [Elaboración propia]

Tabla 3.8: Especificación de Caso De Uso Ingresar Al Sistema

Caso de uso	Ingresar al Sistema
Objetivo	Describe el proceso de ingresar al Sistema web
Precondiciones	Ninguna

Actores	Docente
Secuencia	<ul style="list-style-type: none"> • Ingresa al Sistema web • Inserta nombre de usuario, contraseña, e ingresa • El Sistema comprueba al usuario y accede al Sistema
Secuencia Alternativa	<ul style="list-style-type: none"> • Ingresa al Sistema web • Inserta nombre de usuario, contraseña, e ingresa • El Sistema comprueba al usuario y este no accede, el Sistema vuelve a la raíz

Fuente: [Elaboración propia]

Tabla 3.9: Especificación de Caso De Uso Ver Mi Perfil

Caso de uso	Ver mi perfil
Objetivo	Describe el proceso para revisar los datos registrados del docente que ingreso al Sistema
Precondiciones	El docente debe ingresar al Sistema con usuario y contraseña
Actores	Docente
Secuencia	<ul style="list-style-type: none"> • Selecciona la opción mi perfil • El Sistema muestra los datos del usuario
Secuencia Alternativa	Ninguna

Fuente: [Elaboración propia]

Tabla 3.10: Especificación de Caso De Uso Ver Mis Cursos

Caso de uso	Ver mis materias
Objetivo	Describe los la información del docente y sus Cursos Asignados de acuerdo a su especialización en un instrumento
Precondiciones	El docente debe ingresar al Sistema con usuario y contraseña
Actores	Docente
Secuencia	<ul style="list-style-type: none"> • Selecciona la opción mis cursos • El Sistema muestra las materias que el docente impartirá en la gestión académica
Secuencia Alternativa	Ninguna

Fuente: [Elaboración propia]

Tabla 3.11: Especificación de Caso De Uso Ver Lista De Estudiantes

Caso de uso	Ver lista de estudiantes
Objetivo	Muestra una lista de estudiantes de los cursos de los docentes
Precondiciones	El docente debe ingresar al Sistema con usuario y contraseña
Actores	Docente
Secuencia	<ul style="list-style-type: none"> • Selecciona la opción estudiantes • El Sistema muestra una lista de los estudiantes

Secuencia Alternativa	Ninguna
------------------------------	---------

Fuente: [Elaboración propia]

Tabla 3.12: Especificación de Caso De Uso Ver Acta De Calificaciones

Caso de uso	Ver acta de calificaciones
Objetivo	Muestra una lista de estudiantes y sus notas para ser actualizadas
Precondiciones	El docente debe ingresar al Sistema con usuario y contraseña
Actores	Docente
Secuencia	<ul style="list-style-type: none"> • Selecciona la opción acta de calificaciones • El Sistema muestra una lista de los estudiantes y sus notas
Secuencia Alternativa	Ninguna

Fuente: [Elaboración propia]

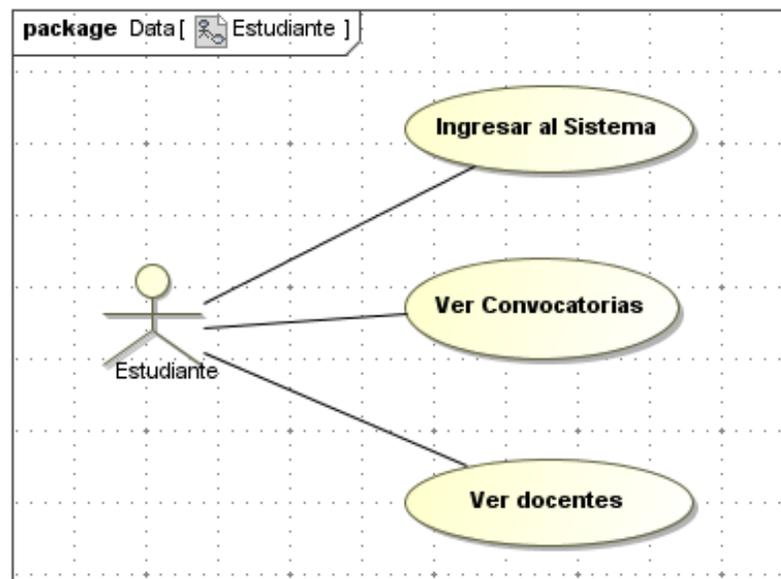
Tabla 3.13: Especificación de Caso De Uso Asignar Nota

Caso de uso	Asignar nota
Objetivo	Realiza la operación de Asignar la nota de los estudiantes
Precondiciones	El docente debe ingresar al Sistema con usuario y

	contraseña
Actores	Docente
Secuencia	<ul style="list-style-type: none"> • Selecciona la opción editar • El Sistema muestra el formulario para la edición de la nota • El docente ingresa la nota • El Sistema actualiza la nota del estudiante
Secuencia Alternativa	<ul style="list-style-type: none"> • Selecciona la opción editar • El Sistema muestra el formulario para la edición de la nota • El docente cancela la operación • El Sistema muestra una lista de estudiantes

Fuente: [Elaboración propia]

Figura 3.4: Caso de Uso Estudiante



Fuente: [Elaboración propia]

Tabla 3.14: Especificación de Caso De Uso Ingresar al Sistema

Caso de uso	Ingresar al Sistema
Objetivo	Describe el proceso de ingresar al Sistema web
Precondiciones	Ninguna
Actores	Estudiante
Secuencia	Ingresar al Sistema web desde cualquier navegador Se visualiza el portal del Sistema mostrando el contenido fotos, convocatorias
Secuencia Alternativa	Ninguno

Fuente: [Elaboración propia]

Tabla 3.15: Especificación de Caso De Uso Ver Convocatorias

Caso de uso	Ver Convocatorias
Objetivo	Describe la información de las convocatorias de los cursos de instrumentos activos
Precondiciones	El estudiante debe ingresar al Sistema web desde cualquier navegador
Actores	Estudiante
Secuencia	Visualiza los cursos y convocatorias que están en el Sistema web El Sistema muestra los cursos y convocatorias que estén

	activos
Secuencia Alternativa	Ninguna

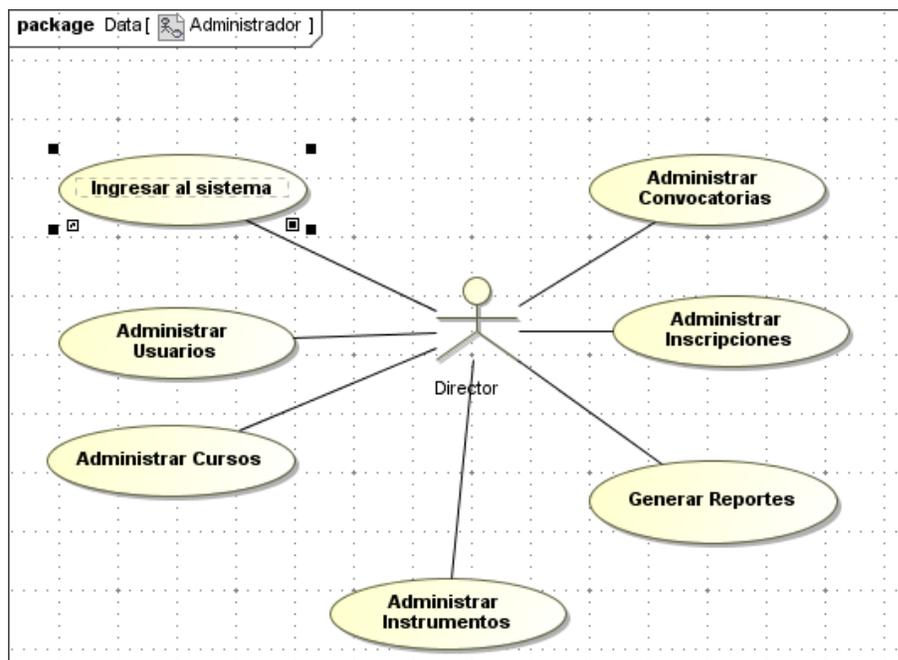
Fuente: [Elaboración propia]

Tabla 3.16: Especificación de Caso De Uso Ver Docentes

Caso de uso	Ver Docentes
Objetivo	Describe la información de los docentes que están asignados para los diferentes cursos
Precondiciones	El estudiante debe ingresar al Sistema desde cualquier navegador
Actores	Estudiante
Secuencia	Visualiza en el sistema web los nombres de los docentes que impartirán los cursos
Secuencia Alternativa	Ninguna

Fuente: [Elaboración propia]

Figura 3.5: Caso de Uso Director



Fuente: [Elaboración propia]

Tabla 3.17: Especificación de Caso De Uso Ingresar al Sistema

Caso de uso	Ingresar al Sistema
Objetivo	Describe el proceso de ingresar al Sistema web
Precondiciones	Ninguna
Actores	Administrador
Secuencia	<ul style="list-style-type: none"> • Ingresa al Sistema Web • Inserta nombre de usuario, contraseña, e ingresa • El Sistema comprueba al usuario y accede al Sistema
Secuencia	<ul style="list-style-type: none"> • Ingresa al Sistema

Alternativa	<ul style="list-style-type: none"> • Inserta nombre de usuario, contraseña e ingresa • El Sistema comprueba al usuario y este no accede, el Sistema vuela a la raíz
--------------------	---

Fuente: [Elaboración propia]

Tabla 3.18: Especificación de Caso De Uso Administrar Usuarios

Caso de uso	Administrar usuarios
Objetivo	Describe la administración de los usuarios docentes, estudiantes, Secretaria como nuevo, borrar, editar, imprimir
Precondiciones	Director debe ingresar al Sistema con usuario y contraseña
Actores	Administrador
Secuencia	<ul style="list-style-type: none"> • Selecciona la opción Grupos y Usuarios • Muestra una lista de los usuarios activos • El usuario tiene las diferentes opciones administrar a los usuarios de acuerdo a su condición
Secuencia Alternativa	Ninguna

Fuente: [Elaboración propia]

Tabla 3.19: Especificación de Caso De Uso Administrar Cursos

Caso de uso	Administrar cursos
Objetivo	Describe la administración de los cursos de instrumentos activos como nuevo, borrar, editar, Actualizar

Precondiciones	El Administrador debe ingresar al Sistema con usuario y contraseña
Actores	Administrador
Secuencia	<ul style="list-style-type: none"> • Selecciona la opción cursos • El Sistema muestra una lista de cursos • El usuario tiene las diferentes opciones para su administración
Secuencia Alternativa	Ninguna

Fuente: [Elaboración propia]

Tabla 3.20: Especificación de Caso De Uso Administrar Instrumentos

Caso de uso	Administrar Instrumentos
Objetivo	Describe la administración de los instrumentos para la gestión académica como nuevo, borrar, editar, Actualizar
Precondiciones	El Administrador debe ingresar al Sistema con usuario y contraseña
Actores	Administrador
Secuencia	<ul style="list-style-type: none"> • Selecciona la opción Instrumentos • El Sistema muestra una lista de instrumentos • El usuario tiene las diferentes opciones para su administración
Secuencia	Ninguna

Alternativa	
--------------------	--

Fuente: [Elaboración propia]

Tabla 3.21: Especificación de Caso De Uso Administrar Convocatoria

Caso de uso	Administrar Convocatoria
Objetivo	Describe la administración de las convocatorias que se lanzaran de los distintos cursos d instrumentos Musicales que dispone la Academia de música Mozart como nuevo, borrar, editar, Actualizar
Precondiciones	El Administrador debe ingresar al Sistema con usuario y contraseña
Actores	Administrador
Secuencia	<ul style="list-style-type: none"> • Selecciona la opción Convocatorias • Registra el instrumento para la convocatoria • El usuario tiene las diferentes opciones para su administración
Secuencia Alternativa	Ninguna

Fuente: [Elaboración propia]

Tabla 3.22: Especificación de Caso De Uso Administrar Inscripciones

Caso de uso	Administrar inscripciones
Objetivo	Describe la administración de los Registros de inscripciones

	como ser: nuevo, borrar, editar, Actualizar
Precondiciones	El Administrador debe ingresar al Sistema con usuario y contraseña
Actores	Administrador
Secuencia	<ul style="list-style-type: none"> • Selecciona la opción de los Registros ya sea de docentes o estudiantes • Muestra una lista de las los inscritos y el curso donde están inscritos • El usuario tiene las diferentes opciones para su administración
Secuencia Alternativa	Ninguna

Fuente: [Elaboración propia]

Tabla 3.23: Especificación de Caso De Uso Generar Reportes

Caso de uso	Generar reportes
Objetivo	Muestra la forma en la cual se quieren generar reportes con diferentes criterios
Precondiciones	El Administrador debe ingresar al Sistema con usuario y contraseña
Actores	Administrador
Secuencia	<ul style="list-style-type: none"> • Selecciona la opción reportes • El Sistema muestra un diferentes formularios para la generación de estos reportes

	<ul style="list-style-type: none"> • El usuario ingresa el criterio para generar el reporte • El Sistema muestra el reporte
Secuencia Alternativa	Ninguna

Fuente: [Elaboración propia]

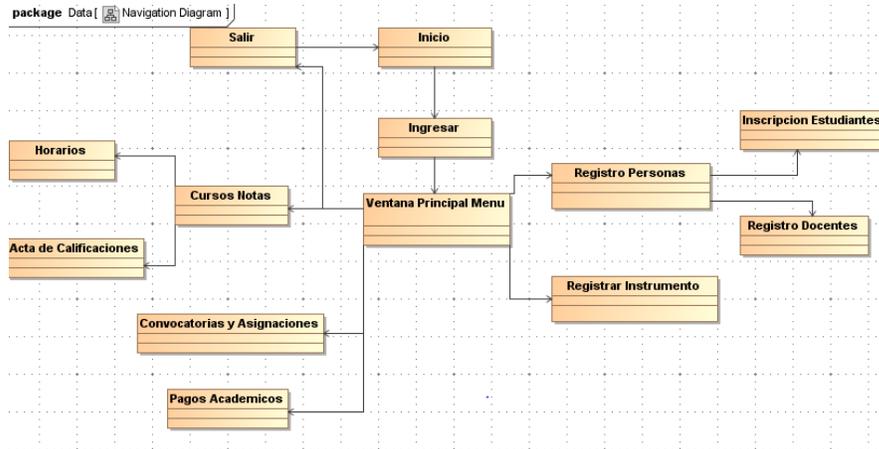
3.5. Fase Diseño de Sistema

3.5.1. Modelo De Contenido

El diagrama de contenido tiene por propósito mostrar las relaciones entre las entidades y la estructura de los datos que se encuentran alojadas en el Sistema el modelo de contenido contiene la información relevante almacenada en el Sistema como se estructura y como se relaciona.

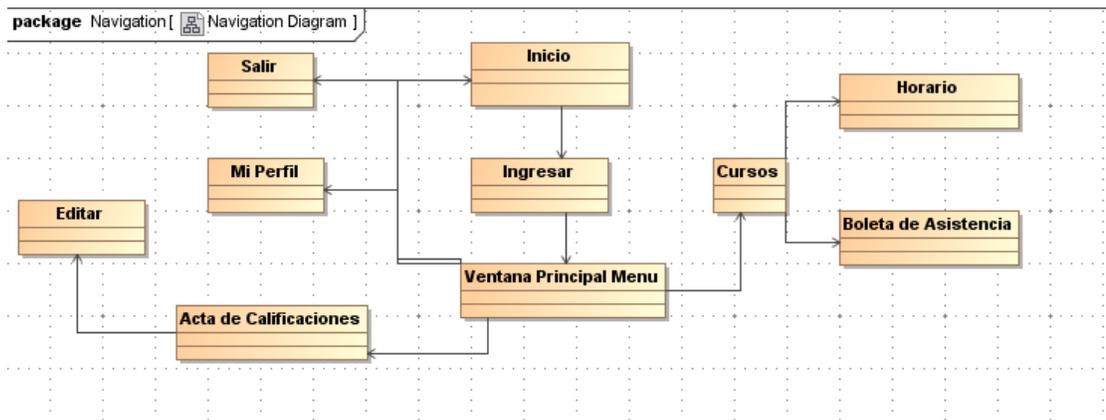
Esto se representa mediante un diagrama de clases UML como se muestra en la siguiente figura.

Figura 3.7: Modelo de Navegación Usuario Secretaria



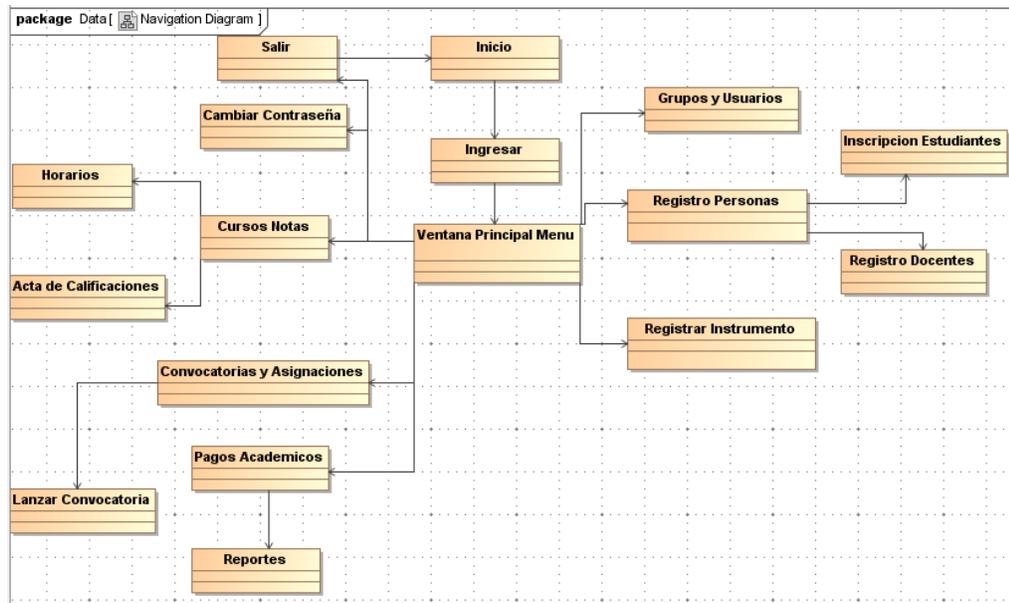
Fuente: [Elaboración propia]

Figura 3.8: Modelo de Navegación Usuario Docente



Fuente: [Elaboración propia]

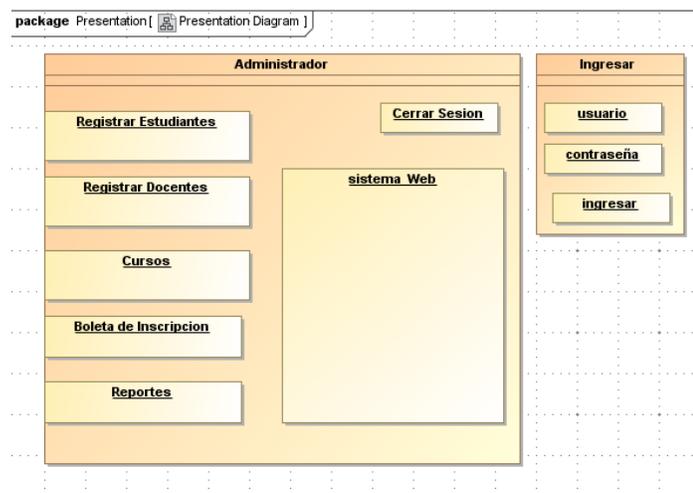
Figura 3.9: Modelo de Navegación Usuario Administrador (Director)



Fuente: [Elaboración propia]

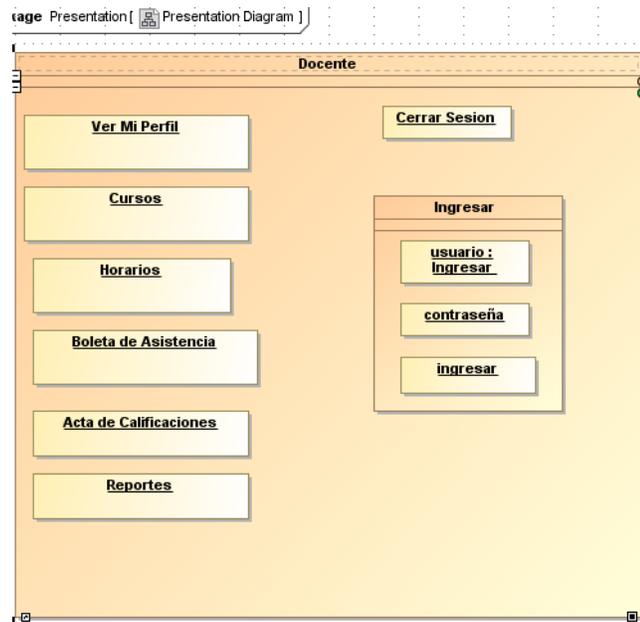
3.5.3. Modelo De Presentación

Figura 3.10: Modelo de Presentación Usuario Secretaria



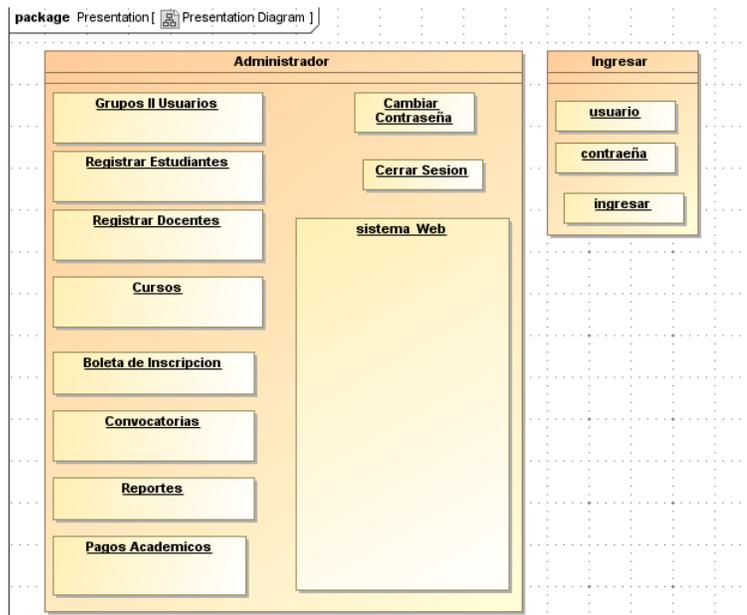
Fuente: [Elaboración propia]

Figura 3.11: Modelo de Presentación Usuario Docente



Fuente: [Elaboración propia]

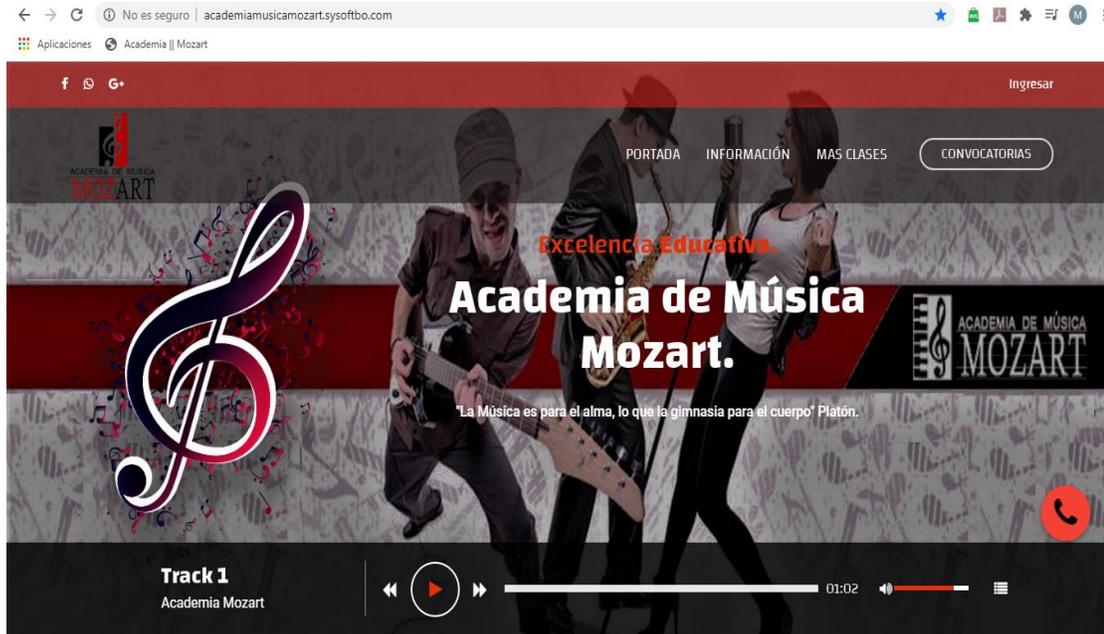
Figura 3.12: Modelo de Presentación Usuario Administrador



Fuente: [Elaboración propia]

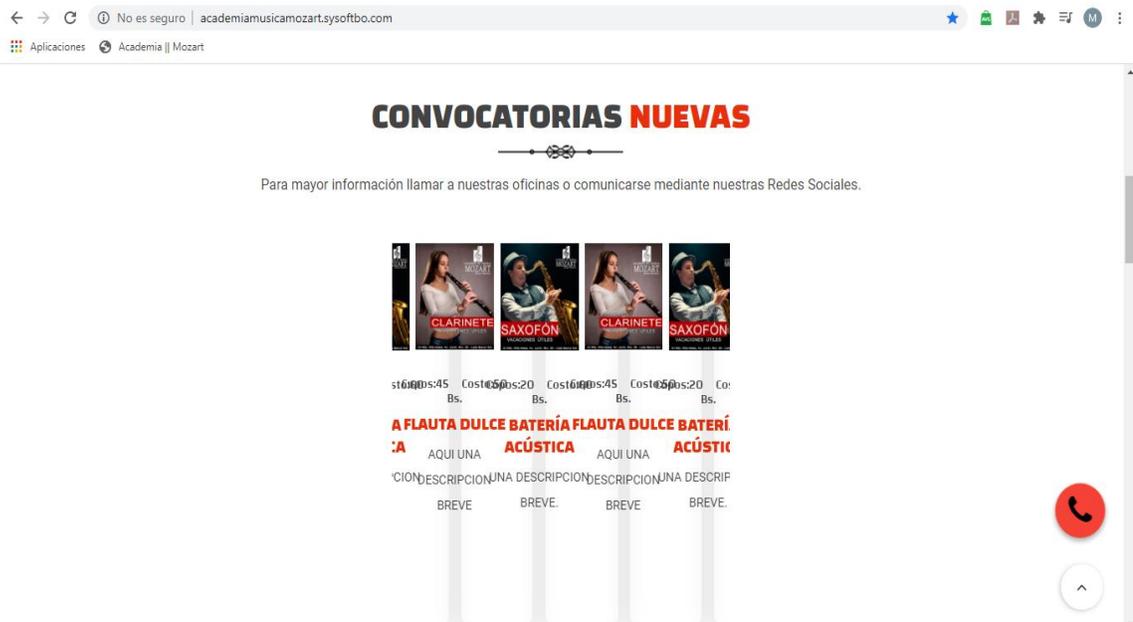
3.5.4. Captura De Ventanas Del Sistema

Figura 3.13: Ventana principal de la página web Academia de Música Mozart



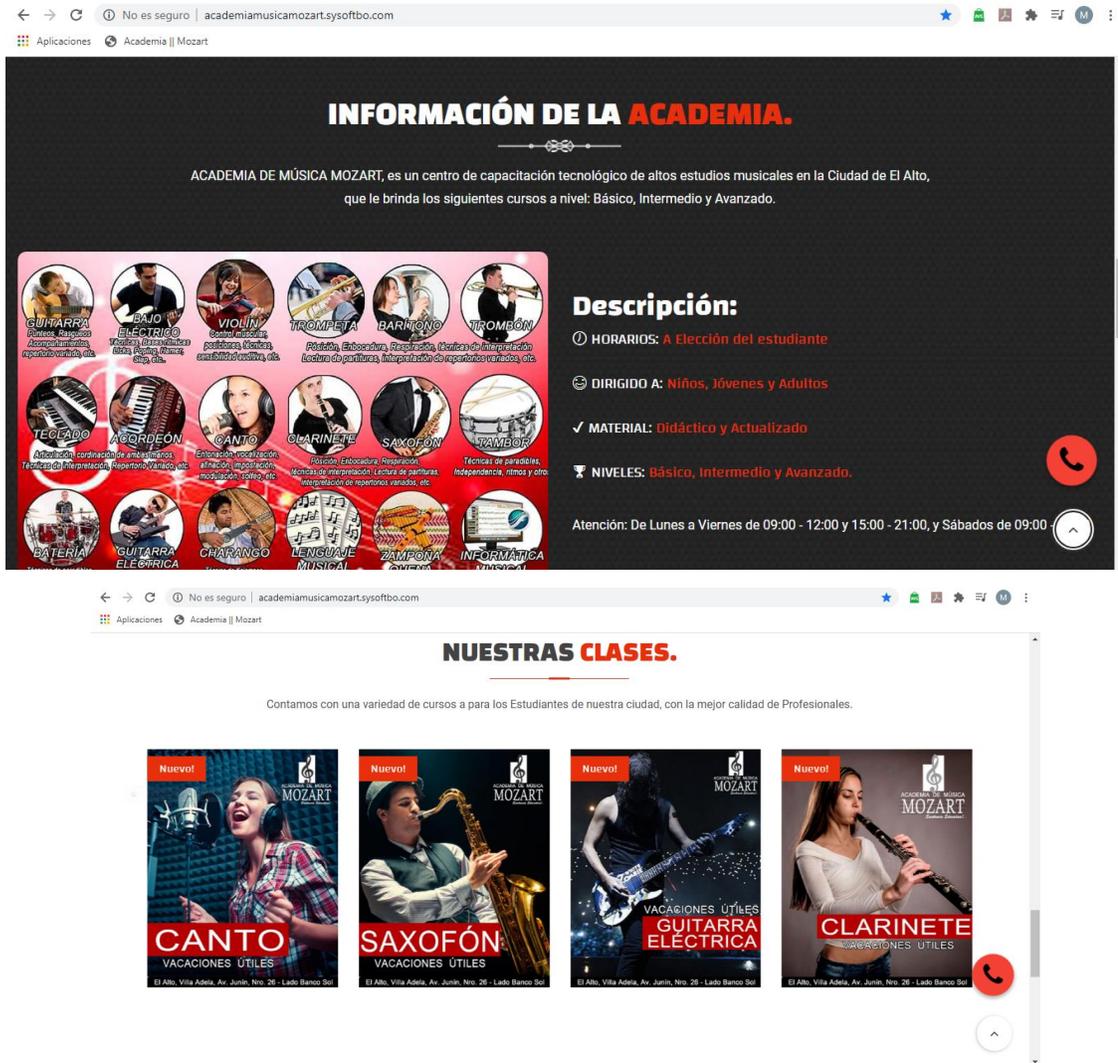
Fuente: [Elaboración propia]

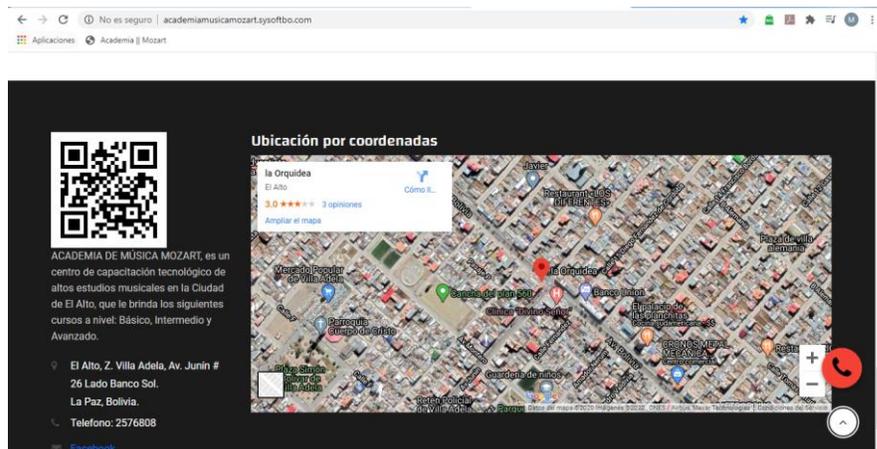
Figura 3.14: Ventana principal de las convocatorias



Fuente: [Elaboración propia]

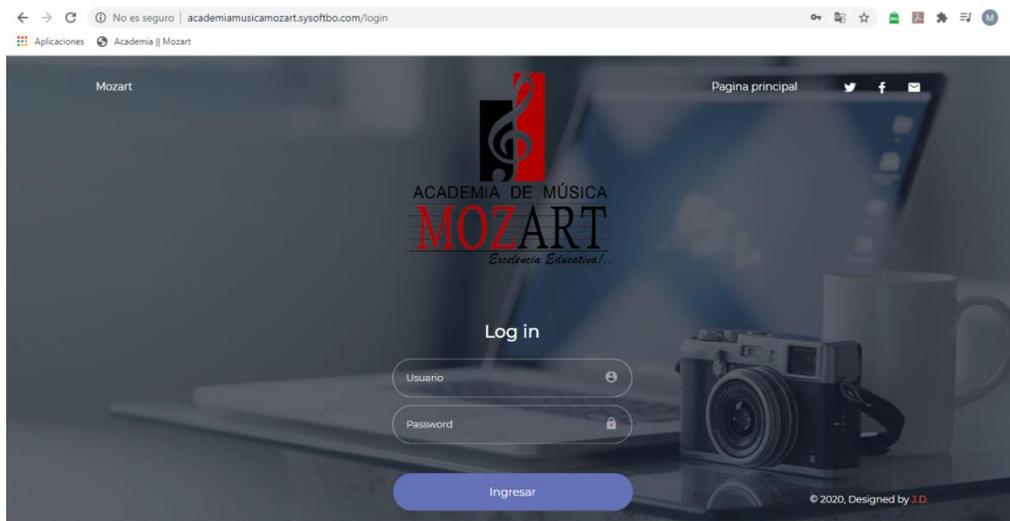
Figura 3.15: Ventana principal de Información de la Academia de Música Mozart





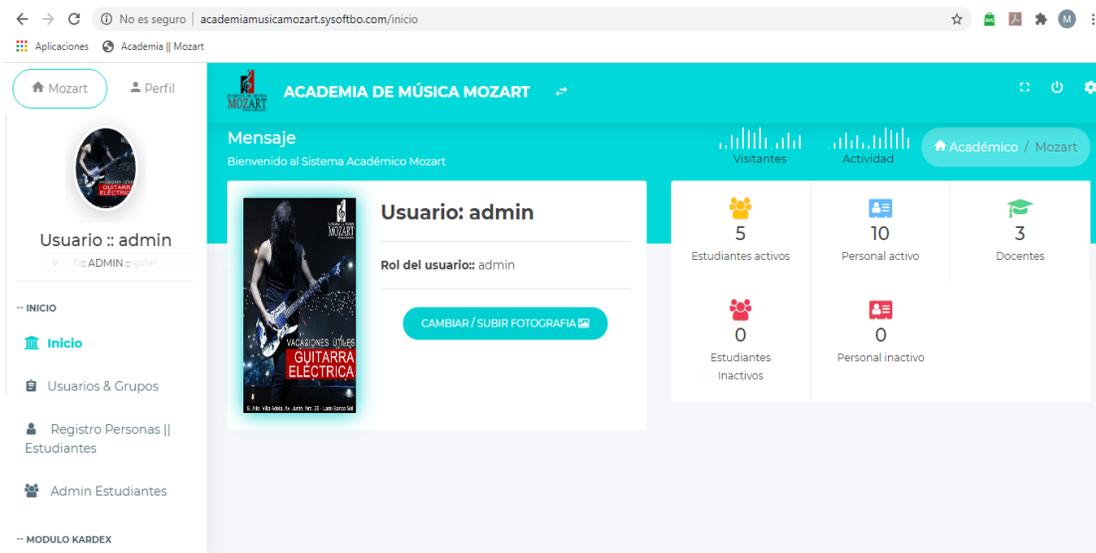
Fuente: [Elaboración propia]

Figura 3.16: Ingreso Al Sistema Inicio de Sesión



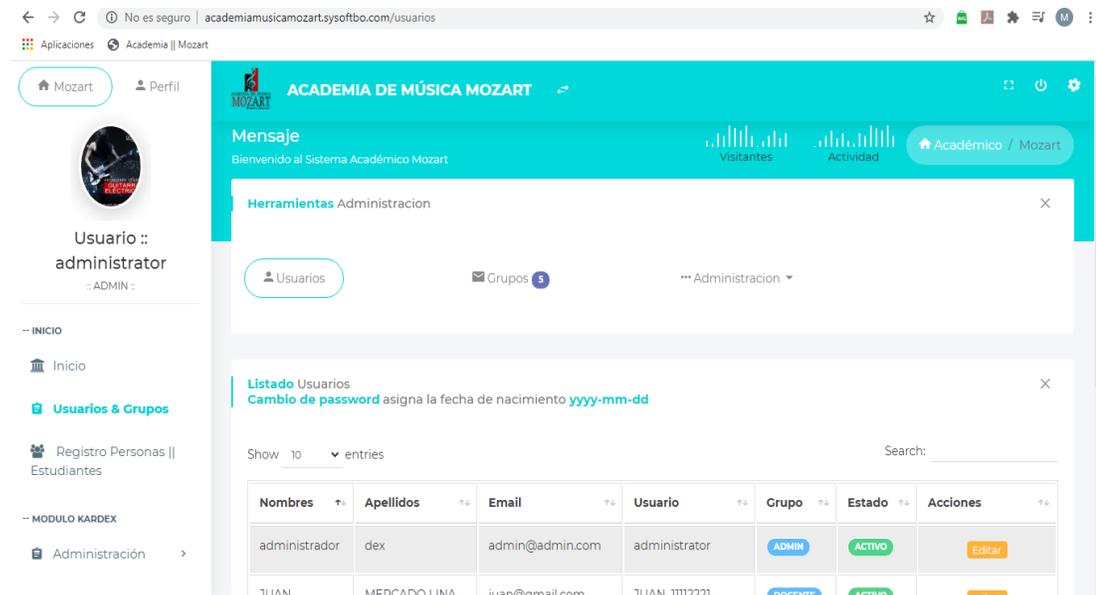
Fuente: [Elaboración propia]

Figura 3.17: Perfil De Usuario administrador



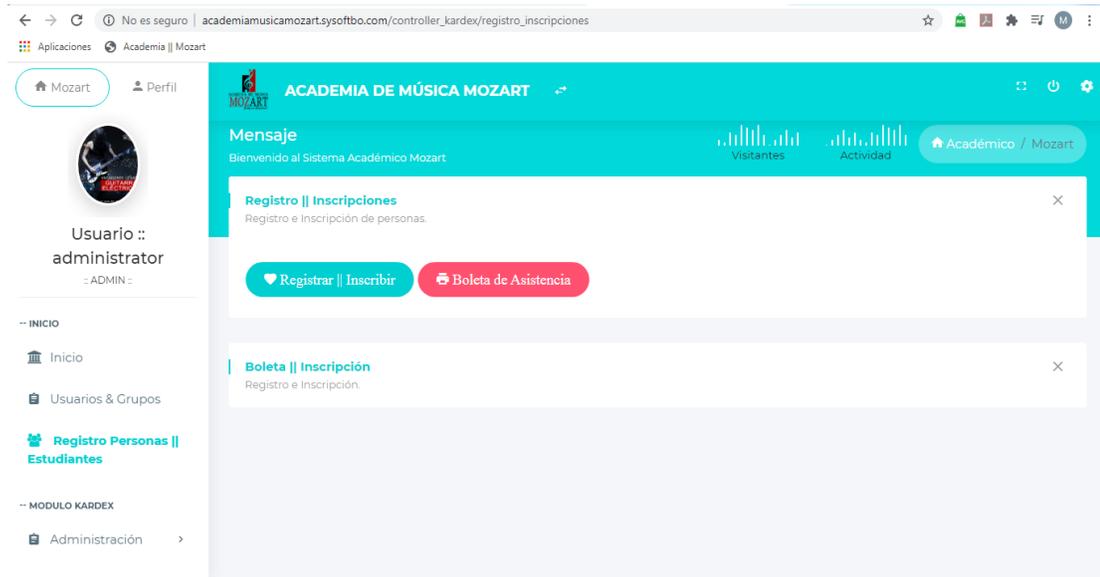
Fuente: [Elaboración propia]

Figura 3.18: Asignación de Usuarios y Grupos



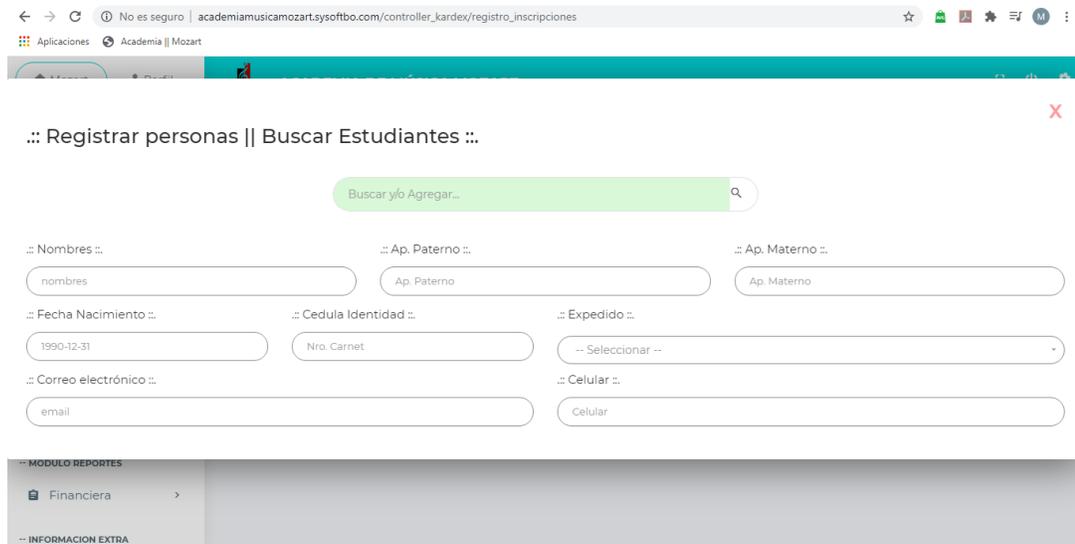
Fuente: [Elaboración propia]

Figura 3.19: Registro de Estudiantes y docentes



Fuente: [Elaboración propia]

Figura 3.20: Inscripción del estudiante nuevo o antiguo



Fuente: [Elaboración propia]

Figura 3.21: Asignar Cursos a Docentes

The screenshot shows the 'ACADEMIA DE MÚSICA MOZART' interface. On the left is a sidebar with user information and navigation options. The main content area displays a table of courses. The table has columns for 'Acciones', 'Cursos', 'Descripción', 'Categoría', and 'Opciones'. The first two rows are visible, both for 'BATERÍA ACÚSTICA' and 'GUITARRA ACÚSTICA'. Each row includes details like 'Sigla', 'Nivel', 'Modulo', 'Inicio Clases', 'Conclusion Clases', 'Horario', 'Cupos', 'Categoría', 'Precio', 'Gestión', 'Docente', and 'C.I.'. An 'ASIGNAR DOC' button is present in the 'Opciones' column of each row.

Acciones	Cursos	Descripción	Categoría	Opciones
	BATERÍA ACÚSTICA Sigla : BAA-MOZ-108 Nivel : BÁSICO Modulo : I	UNA DESCRIPCION BREV... Inicio Clases : 2020-08-01 Conclusion Clases : 2020-08-30 Horario : 11:00 am Cupos : 20	NORMAL Precio: Bs 60 Gestión: 2020 Docente: MARCELO TICONA MAMANI C.I: 11112222 - PO	 estado ACTIVO ASIGNAR DOC
	GUITARRA ACÚSTICA Sigla : GUA-MOZ-104 Nivel : BÁSICO Modulo : I	ESTE CURSO LANZO CON... Inicio Clases : 2020-05-30 Conclusion Clases : 2020-08-30 Horario : 02:00 pm Cupos : 50	NORMAL Precio: Bs 35 Gestión: 2020 Docente: MARCELO TICONA MAMANI C.I: 11112222 - PO	 estado ACTIVO ASIGNAR DOC

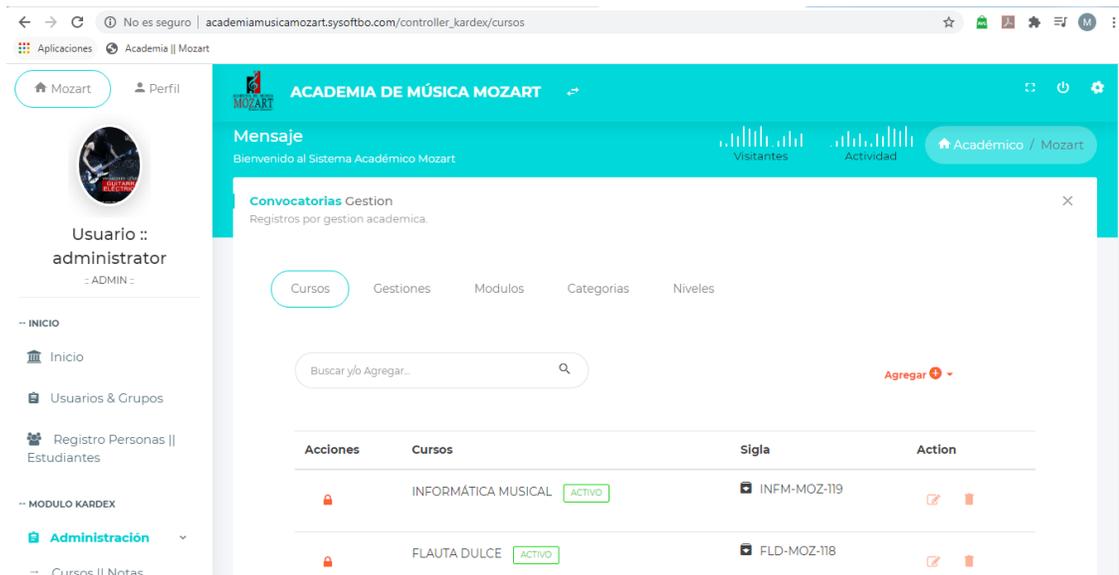
Fuente: [Elaboración propia]

Figura 3.22: Asignar convocatorias

The screenshot shows the 'ACADEMIA DE MÚSICA MOZART' interface with a 'Convocatorias Gestión' modal window open. The modal has a title bar and a close button. Below the title bar is a message: 'ASIGNAR DOCENTES LUEGO DE AGREGAR EL CURSO.' and an 'Agregar' button. The main content of the modal is a table of courses, identical to the one in Figure 3.21. The table has columns for 'Acciones', 'Cursos', 'Descripción', 'Categoría', and 'Opciones'. The first two rows are visible, both for 'BATERÍA ACÚSTICA' and 'GUITARRA ACÚSTICA'. Each row includes details like 'Sigla', 'Nivel', 'Modulo', 'Inicio Clases', 'Conclusion Clases', 'Horario', 'Cupos', 'Categoría', 'Precio', 'Gestión', 'Docente', and 'C.I.'. An 'ASIGNAR DOC' button is present in the 'Opciones' column of each row.

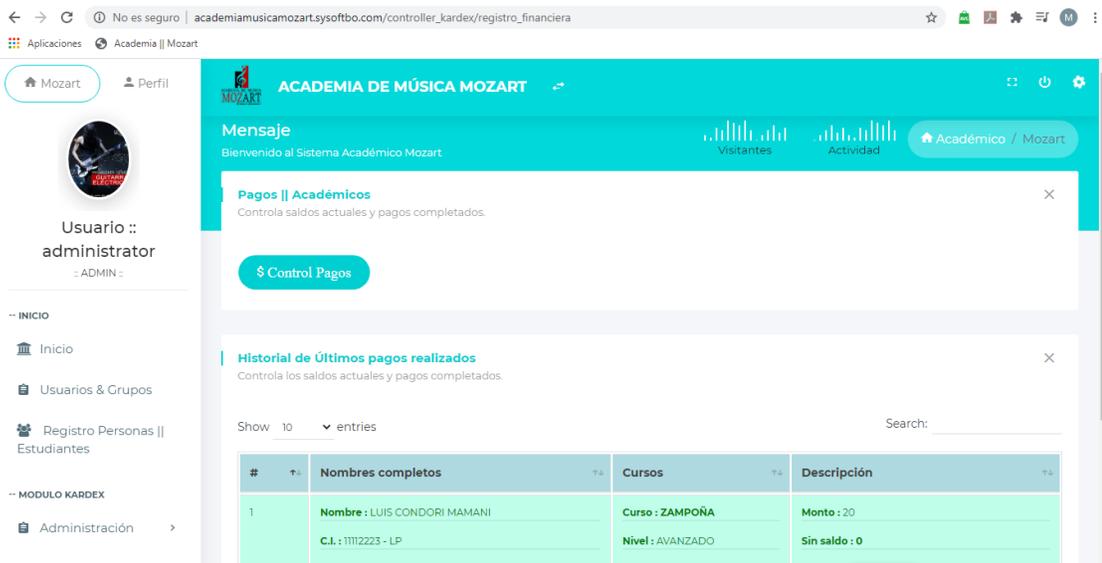
Fuente: [Elaboración propia]

Figura 3.23: Asignación de Instrumentos



Fuente: [Elaboración propia]

Figura 3.24: Registro de Pagos



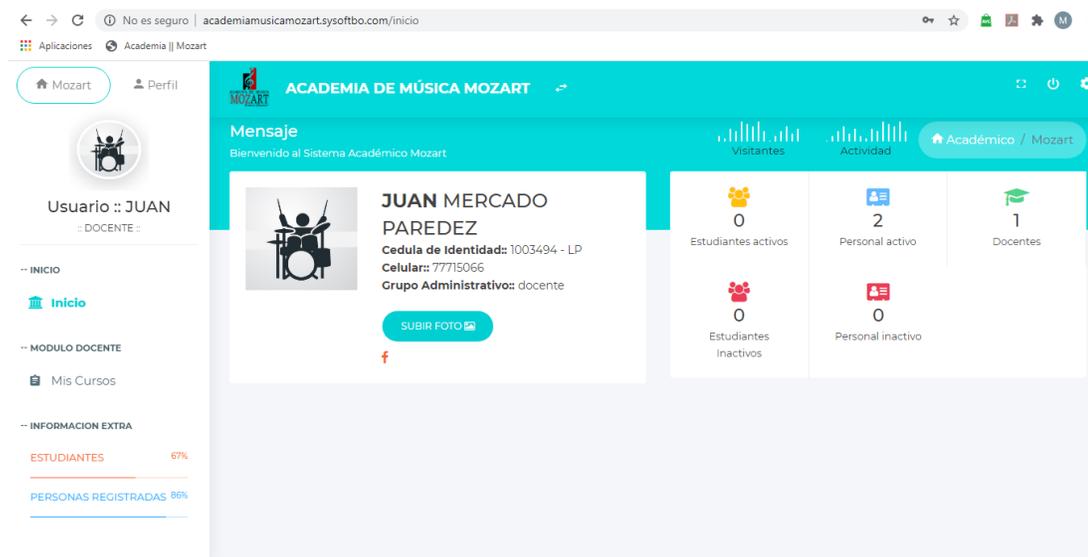
Fuente: [Elaboración propia]

Figura 3.25: Acta de Notas de los Estudiantes



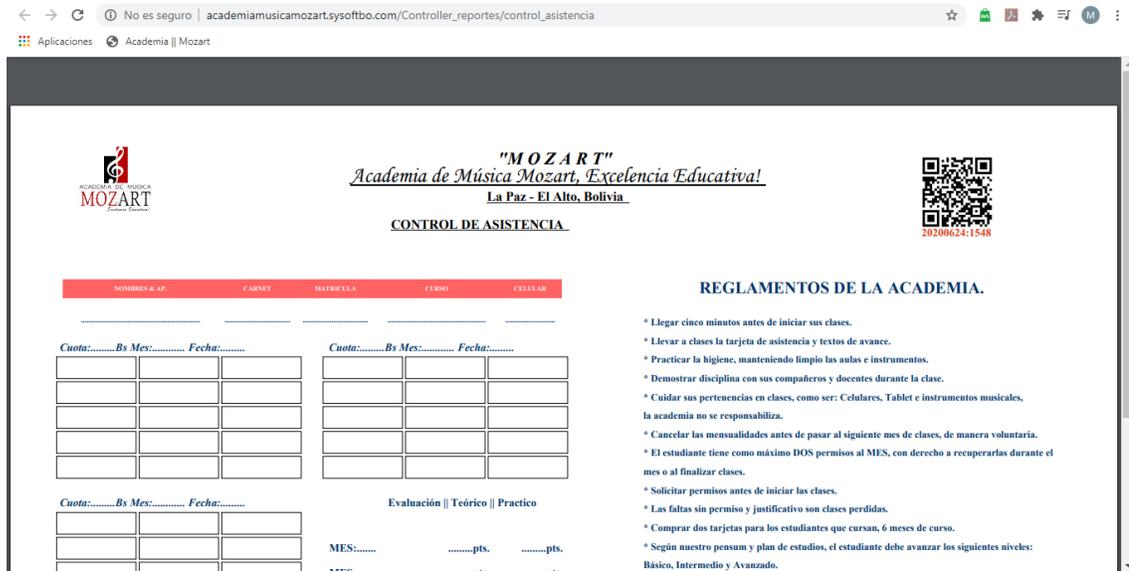
Fuente: [Elaboración propia]

Figura 3.26: Perfil de Docente



Fuente: [Elaboración propia]

Figura 3.27: Control de Asistencia



Fuente: [Elaboración propia]

3.6. Fase de Codificación del Sistema

Como se trata de un Sistema elaborado con tecnología web está disponible para los usuarios en multiplataforma ya sea en Sistema operativo Windows, Linux, MacOS u otros, siempre que cuenten con un explorador de páginas web como Explorer, Mozilla, Chrome, etc.

Las herramientas utilizadas para poner en funcionamiento el Sistema se detallan a continuación:

- Sistema Operativo: el Sistema operativo del servidor que será host del Sistema es Linux
- Servidor Web: el software de servidor web utilizado para el Sistema es apache que viene incluido en el paquete de instalación XAMPP

- Gestor de base de datos: el gestor de base de datos que contiene la información esencial del Sistema es Maria Db.
- Herramientas de programación: para el desarrollo del Sistema académico de inscripción se utilizó el lenguaje de programación php, utilizando además codeigniter que es un Framework muy completo y con la estructura MVC (modelo-vista-controlador).
- Herramientas para el diseño: se utilizó para el análisis y diseño del Sistema MagicDraw que posee una extensión de la metodología UWE.

3.6.1. Actividades

1. Instalar y configurar el Sistema operativo para este caso ya se tiene el Sistema operativo instalado y configurado en el servidor.
2. Instalar y configurar apache, el servidor con el que se cuenta tiene instalado el Sistema operativo Ubuntu con servidor web apache ya configurado.
3. Instalar y configurar MariaDb gestor de base de datos, el servidor con el que se cuenta tiene instalado y configurado Mysql.
4. Cargar el Sistema, se cargó el Sistema en el servidor sin ninguna complicación.
5. Se creó los usuarios de acuerdo a sus roles dentro de la institución.
6. Se informa al Director de la Academia de Música Mozart y a los Docentes acerca del funcionamiento del Sistema entregando un manual de usuario.

4. Calidad y Seguridad Del Software

La calidad del software es la eficiencia y producción de su rendimiento y funcionamiento del equipo, es el estado de un producto o servicio para satisfacer la necesidad del usuario, también cualidad de todos los productos, no solamente de equipos sino también de programas entre otros también podemos decir que la calidad del software debe tener un buen análisis para observar que todo funcione en concordancia.

Es necesario evaluar la calidad del software de esta manera se detecta los problemas que pudiera llegar a tener en su desarrollo antes de ser implantado y así lograr un mejor producto cumpliendo con los objetivos.

4.1. Pruebas De Calidad

Se aplicara la norma ISO 9126 un estándar internacional para la evaluación del software, que clasifica la calidad del software en funcionalidad, fiabilidad, usabilidad, eficiencia, mantenimiento y portabilidad.

4.1.1. Funcionalidad

Para cumplir con la funcionalidad primero hallamos el punto de función, el punto de función se calcula realizando una serie de actividades comenzando por determinar los siguientes valores:

- Número de entradas del usuario
- Número de salidas del usuario

- Número de consultas de usuario
- Número de archivos
- Número de interfaces externas

Aplicando esto al proyecto se tiene los siguientes datos en la tabla:

Tabla 4.1: Parámetros de medición

Número de entradas del usuario	25
Número de salidas del usuario	20
Número de consultas de usuario	27
Número de archivos	15
Número de interfaces externas	0

Fuente: [Elaboración propia]

Para calcular los puntos de función se tiene:

Tabla 4.2: Calculo de Punto de función

Parámetros de medición	Cuenta	Medio	Resultado
Número de entradas del usuario	25	4	100
Número de salidas del usuario	15	5	75
Número de consultas de usuario	27	4	108
Número de archivos	15	10	150
Número de interfaces externas	0	7	0
Cuenta Total			433

Fuente: [Elaboración propia]

La relación que nos permite calcular el punto de función es la siguiente

$$PF = \text{Cuenta total} * (\text{Grado de confiabilidad} + \text{Tasa de error } \sum Fi)$$

Dónde:

PF: medida de funciones

Cuenta Total: es la sumatoria del producto del factor de ponderación y valores de los parámetros.

Grado de confiabilidad: es la confiabilidad estimada del Sistema.

Tasa de error: probabilidad subjetiva estimada del dominio de la información.

Fi: sin valores de ajuste de complejidad.

Tabla 4.3: Valores de ajuste de complejidad

Importancia	0%	20%	40%	60%	80%	100%	Valor obtenido
Escala	No influencia	Incidenti	Moderad	Medio	Significati	Esencial	
Factor	0	1	2	3	4	5	
¿Requiere el Sistema copias de seguridad y recuperación fiables?					X		4
¿Se requiere comunicación de datos?					X		4
¿Existen funciones de procesamiento distribuido?					X		4
¿Es crítico el rendimiento?			X				2
¿Se ejecuta el Sistema en un entorno operativo existente y fuertemente utilizado?					X		4

¿Requiere el Sistema entrada de datos interactiva?					X		4
¿Requiere el Sistema entrada de dato interactivo que las transacciones de entrada se lleven a cabo sobre múltiples entradas u operaciones?				X			3
¿Se actualizan los archivos maestros de forma interactiva?				X			3
¿Son complejas las entradas, las salidas, los archivos o peticiones?				X			3
¿Es complejo el procesamiento interno?				X			3
¿Se ha diseñado código para ser reutilizable?				X			3
¿Están incluidas en el diseño la conversión y la instalación?			X				2
¿Se ha diseñado el Sistema para soportar múltiples instalaciones en diferentes organizaciones?			X				2
¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?					X		4
Total ΣFi							45

Fuente: [Elaboración propia]

Con la obtención de los datos anteriores y considerando un grado de confiabilidad mínimo, calculamos:

$$PF_{\text{real}} = \text{Cuenta total} (0,65 + 0,01 * \Sigma Fi)$$

$$PF_{\text{real}} = 433 * (0,65 + 0,01 * 45)$$

$$PF_{\text{real}} = 476,3$$

Si consideramos el máximo valor de ajuste de complejidad $\sum Fi = 70$ tenemos:

$$PF_{\text{esperada}} = \text{Cuenta total } (0,65 + 0,01 * \sum Fi)$$

$$PF_{\text{esperada}} = 433 * (0,65 + 0,01 * 70)$$

$$PF_{\text{esperada}} = 584,55$$

La relación obtenida entre ambos es la funcionalidad:

$$\%PF = Pf_{\text{real}} / PF_{\text{esperada}}$$

$$\%PF = 476,3 / 584,55$$

$$\%PF = 0,81$$

Por lo que se concluye que la funcionalidad del Sistema es de 81%

Calculamos la métrica de adecuación:

Tabla 4.4: Métrica de adecuación

Nombre	Complejidad de implementación funcional
Propósito	Verificar que tan completa es la implementación funcional
Método de aplicación	Comparar funciones faltantes y funciones descritas en la especificación de requisitos
Formula	$x = 1 - a/b$ a: número de funciones faltantes. b: número de funciones descritas.
Interpretación	$0 \leq x \leq 1$

	Entre más cerca de 1 más completa
Aplicación	$x = 1 - a/b$ $a = 0$ $b = 8$ $x = 1 - 0/8$ $x = 1$
Interpretación	$x = 1$, implica que cumple con la métrica de adecuadad.

Fuente: [Elaboración propia]

4.1.2. Confiabilidad

Para determinar la confiabilidad de un Sistema especificamos desde el instante que empieza a funcionar es decir $t_0 = 0$, a partir de este momento se realiza las observaciones pertinentes. En son de encontrar una falla en el Sistema considerando el tiempo de falla como t_1 , como intervalo entre ambos tiempos es una variable continua se vio la necesidad del uso de una función continua, que nos da la confiabilidad en términos probabilísticos.

$$P(T \leq t) = F(t) \quad \text{Probabilidad de fallos}$$

$$P(T > t) = 1 - F(t) \quad \text{Probabilidad de éxito}$$

Para el cálculo de las probabilidades se tomó la distribución exponencial, por la existencia de intervalos continuos.

$$F(t) = PF e^{-\lambda * t}$$

Para calcular el índice de error tomamos 8 ejecuciones en una semana durante un mes y medio, y reemplazando tenemos:

$$F(t) = 0,81 e^{-[(-1/5)*8]}$$

$$F(t) = 0,12$$

Reemplazando en las fórmulas de probabilidades:

$$P(T \leq t) = F(t)$$

$$P(T \leq t) = 0,12 \quad \text{Probabilidad de fallos}$$

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

$$P(T > t) = 1 - 0,12$$

$$P(T > t) = 0,88 \quad \text{Probabilidad de éxito}$$

Siendo la probabilidad de fallo del 24% y la probabilidad de éxito de un 88%.

Calculamos la métrica de madurez:

Tabla 4.5: Métrica de madurez

Nombre	Suficiencia de pruebas
Propósito	Cuantos de los casos de prueba necesarios están cubiertos por el plan de pruebas
Método de aplicación	Contar las pruebas planeadas y comparar con el número de pruebas requeridas para obtener una cobertura adecuada.

Formula	$X = A/B$ A = número de casos de prueba en el plan B = número de casos de prueba requeridos
Interpretación	$0 \leq X$ Entre X sea mayor, mejor la suficiencia.
Aplicación	$X = A/B$ A = 20 B = 25 $X = 20/25$ $X = 0.8$
Interpretación	X = 0.80 , implica que el número de pruebas es aceptable tal que sobrepasa el 50% pero se recomienda el uso total de las pruebas

Fuente: [Elaboración propia]

4.1.3. Usabilidad

Tabla 4.6: Métrica de entendibilidad

Nombre	Funciones evidentes
Propósito	Qué proporción de las funciones del Sistemas son evidentes al usuario.
Método de	Contar las funciones evidentes al usuario y comparar con

aplicación	el número total de funciones.
Formula	$X = A/B$ A = número de funciones (o tipos de funciones) evidentes al usuario B = total de funciones (o tipos de funciones)
Interpretación	$0 \leq X \leq 1$ Entre más cercano a 1, mejor.
Aplicación	$X = A/B$ A = 28 B = 30 $X = 28/30$ $X = 0.93$
Interpretación	X = 0.93 implica que el 93% de las funciones son entendibles por el usuario

Fuente: [Elaboración propia]

4.1.4. Eficiencia

Tabla 4.7: Métrica de comportamiento en el tiempo

Nombre	Tiempo de respuesta
Propósito	Cuál es el tiempo estimado para completar una tarea.
Método de aplicación	Evaluar la eficiencia de las llamadas al SO y a la aplicación. Estimar el tiempo de respuesta basado en ello. Puede medirse:

	<p>Todo o partes de las especificaciones de diseño.</p> <p>Probar la ruta completa de una transacción.</p> <p>Probar módulos o partes completas del producto.</p> <p>Producto completo durante la fase de pruebas.</p>
Formula	X = tiempo (calculado o simulado)
Interpretación	Entre más corto, mejor.
Aplicación	X = 1.2 s
Interpretación	X = 1.2 s implica que el Sistema es ágil en la respuesta a solicitudes

Fuente: [Elaboración propia]

4.1.5. Mantenibilidad

Para verificar la estabilidad del Sistema es decir el índice de madurez del software (IMS), se probó con los cambios que ocurrieron en el desarrollo del software. Para ello tenemos la siguiente formula:

$$IMS = [MT - (Fc + Fa + Fe)]/MT$$

Dónde:

MT: número de módulos en la versión actual.

Fc: número de módulos en la versión actual que se han cambiado.

Fa: número de módulos en la versión actual que se han añadido.

Fd: número de módulos en la versión anterior que se han borrado en la versión actual.

Una vez realizada una revisión del Sistema tenemos lo siguiente:

$$MT = 5 \quad Fc = 0 \quad Fa = 0 \quad Fd = 0,5$$

Reemplazando estos datos en la formula tenemos:

$$IMS = [MT - (Fc + Fa + Fd)]/MT$$

$$IMS = [5 - (0 + 0 + 0,5)]/5$$

$$IMS = 9/10$$

$$IMS = 0,9$$

Tomando en cuenta las ponderaciones se concluye que el 90% está en el intervalo con calificación de buena que implica aceptablemente estable.

Calculando la métrica de cambiabilidad:

Tabla 4.8: Métrica de cambiabilidad

Nombre	Registrabilidad de cambios
Propósito	¿Se registran adecuadamente los cambios a la especificación y a los módulos con comentarios en el código?
Método de aplicación	Registrar la proporción de información sobre cambios a los módulos
Formula	$X = A/B$ A = número de cambios a funciones o módulos que tienen comentarios confirmados

	B = total de funciones o módulos modificados
Interpretación	$0 \leq X \leq 1$ Entre más cercano a 1, más registrable. 0 indica un control de cambios deficiente o pocos cambios y alta estabilidad.
Aplicación	$X = A/B$ A = 4 B = 5 $X = 4/5$ $X = 0.8$
Interpretación	Implica que gran parte del código se encuentra con comentario explicativos

Fuente: [Elaboración propia]

4.1.6. Portabilidad

Tratándose de un Sistema con tecnología web este estará al alcance de cualquier usuario con acceso a internet, pero también se toma en cuenta que el Sistema pueda ser fácilmente implementado en una institución.

El Sistema desarrollado como se trata de un Sistema con tecnología web es fácilmente implementado en un cualquier plataforma con servidor web y gestor de base de datos Mysql y puede ser ejecutado en cualquier computadora con acceso a internet con cualquier navegador web como ser Explorer, Firefox, Opera, Google Chrome, etc.

Tabla 4.9: Métrica de conformidad de transportabilidad

Nombre:	Conformidad de transportabilidad
Propósito:	¿Es transportable los sistemas web según las regulaciones, estándares y convenciones?
Método de aplicación:	Registrar artículos encontrados con conformidad y comparar con el número de artículos no encontrados.
Medición, fórmula:	$X = A/B$ A = número de artículos implementados de conformidad B = total de artículos que requieren conformidad
Interpretación:	$0 \leq X \leq 1$ Entre más cercano a 1, más completa.
Tipo de escala:	Absoluta
Tipo de medida:	$X = 10/11$ A = 10 B = 11
Fuente de medición:	Especificación de conformidad y estándares, convenciones y regulaciones relacionados. Diseño Código fuente Informe de revisión
ISO/IEC 12207 SLCP:	$X = 0.90$
Audiencia:	Indica que el 90 % del sistema esta confiable y apto para ser usado.

Fuente: [Elaboración Propia]

Tabla 4.10: Resultados de la Norma ISO-9126

Características	Resultados %
Funcionalidad	81
Usabilidad	93
Confiabilidad	88%
Mantenibilidad	90
Portabilidad	90
Evaluación de la Calidad Final	89 %

Fuente: [Elaboración Propia]

4.2. Seguridad de Software

La seguridad de software consiste en identificar que partes del sistema son vulnerables y establecer medidas que minimicen el riesgo.

4.2.1. Pruebas de Caja Negra

Es aquel elemento que es Estudiado desde el punto de vista de las entradas que recibe o las salidas o respuesta que produce sin tener en cuenta su funcionamiento interno.

Tabla 4.11: Valores Límite de Inicio de Sesión

Petición Datos de Entrada	Entrada de Datos Valida	Entrada de Datos no Valida	Resultado
----------------------------------	--------------------------------	-----------------------------------	------------------

Usuario- password	Cadena de texto	Caracteres Especiales, Espacios en blanco	Ingresa al sistema Error
Tipo de flujo de datos			
La aplicación a la cual se accede			
La estructura de datos que viaja con el flujo:			
Usuario y password			
Descripción:			
En el momento que el usuario ingresa nombre de usuario y contraseña de usuario el sistema lo valida y permite su ingreso a la aplicación.			

Fuente: [Elaboración propia]

Figura 4.1: pruebas de caja negra de Inicio de Sesión



Fuente: [Elaboración propia]

Tabla 4.12: Descripción de Pruebas de la caja negra Inicio de Sesión

Petición Datos de Entrada	Entrada de Datos	Resultado
	Valida	

Usuario- password	Administrador Mozart012***	Ingresa al sistema
Tipo de flujo de datos		
Archivo Interno	pantalla	Informe Formulario
La estructura de datos que viaja con el flujo:		
Usuario y password		
Descripción:		
El sistema valida que no ingresen espacios en blanco, al introducir datos validos el sistema concede el ingreso.		

Fuente: [Elaboración propia]

Tabla 4.13: Valores Límite de Registro de Estudiantes

Entrada de Datos Valida	Entrada de Datos no Valida	Resultado
Cadena de texto	Celdas en vacias	Registro de Datos de Estudiantes
Tipo de Flujo de Datos		
La aplicación a la cual se accede		
La estructura de datos que viaja con el flujo:		
Campos de Datos de Registro		
Descripción:		

En el momento que el usuario ingresa los datos del estudiante el sistema valida que no hayan celdas vacías

Fuente: [Elaboración propia]

Figura 4.2: Pruebas de caja negra Registro de Estudiantes

X

:: Registrar personas || Buscar Estudiantes ::

Buscar y/o Agregar... 🔍

:: Nombres ::
 :: Ap. Paterno ::
 :: Ap. Materno ::

:: Fecha Nacimiento ::
 :: Cedula Identidad ::
 :: Expedido ::

:: Correo electrónico ::
 :: Celular ::

Fuente: [Elaboración propia]

Tabla 4.14: Descripción de Pruebas caja negra Registro de Datos de Estudiante

Entrada de Datos Valida	Entrada de Datos no Valida	Resultado
Nombres	Juan	Registro de Datos de Estudiantes Con éxito
Apellido Paterno	Perez	
Apellido Materno	Perez	
Cedula de Identidad	8888888	
Expedido	La Paz	
Fecha de Nacimiento	28-10-1990	

Correo	juanperezperez@gmail.com		
Celular	77777777		
Tipo de Flujo de Datos			
Archivo	pantalla	Informe	Formulario
Interno			
La estructura de datos que viaja con el flujo:			
Nombres, Apellido Paterno, Apellido Materno, Cedula de Identidad, Expedido, Fecha de Nacimiento, Correo, Celular			
Descripción:			
El usuario selecciona registrar estudiantes el sistema verifica que no hayan celdas vacías de haber celdas vacíos el sistema lanza un mensaje de alerta indicando que se debe registrar el campo vacío.			

Fuente: [Elaboración propia]

4.2.2. Pruebas de Caja Blanca

Las pruebas de caja blanca se centran en los detalles procedimentales del software por los que su diseño está fuertemente ligado al código fuente el testeador escoge distintos valores de entrada para examinar cada uno de los posibles flujos de ejecución del programa y cerciorarse de que se devuelven los valores de salida adecuados.

Figura 4.3: Falta de comentarios realizar Mantenimiento al Software

```

public function __construct(){
    parent::__construct();
    //Do your magic here
    $this->load->model('Model_administrator');
    $this->load->library('ion_auth');
    if (!$this->ion_auth->logged_in() && !$this->ion_auth->is_admin()) {
        redirect('login', 'refresh');
    }
    $dato = $this->ion_auth->user()->row();
    $this->id_persona=$dato->id_academico_persona;
    $this->id_usuario=$dato->id;
}

```

Fuente: [Elaboración propia]

Tabla 4.15: Descripción de Pruebas de caja Blanca mantenimiento al software

Datos de Entrada	Resultado
Código fuente	Código fuente sin documentación
Tipo de flujo de datos Archivo pantalla Informe Formulario Interno	
La estructura de datos que viaja con el flujo: <p style="text-align: center;">Código fuente</p>	
Comentarios : Las instrucciones e instancias no están comentadas por lo cual el mantenimiento del código es más complicado de realizar.	

Fuente: [Elaboración propia]

Figura 4.3: Prueba de caja blanca Estructura de condición fuera de estándar

```

public function eliminar_convocatoria(){
    $id_convocatoria=$this->input->post("idConv");
    $this->Model_kardex->eliminar_convocatoria($id_convocatoria);
}
public function get_matriculas(){
    $out = array();
    $dato = $this->Model_kardex->get_matriculas();
    foreach ($dato as $val) {
        $out['academico_estudiante_matricula'] = $val->academico_estudiante_matricula;
    }
    if ($out) {
        echo json_encode($out);
        exit;
    }else{
        echo json_encode([]);
        exit;
    }
}
}

```

Fuente: [Elaboración propia]

Tabla 4.16: Pruebas de caja Blanca Estructura de condición fuera de estándar

<p>Datos de Entrada</p> <p>Código fuente=Estructura condicional</p>	<p>Resultado</p> <p>La estructura condicional fuera del estándar</p>
<p>Tipo de flujo de datos</p> <p>Archivo pantalla Informe Formulario</p> <p>Interno</p>	
<p>La estructura de datos que viaja con el flujo:</p> <p>Código fuente= Estructura condicional</p>	
<p>Comentarios :</p> <p>La estructura condicional no se rige al estándar</p>	

Fuente: [Elaboración propia]

4.2.3. Seguridad de base de Datos

Las principales características de la seguridad en una base de datos son:

- ❖ La confidencialidad de la información
- ❖ La integridad de la información
- ❖ La disponibilidad de la información

Hay dos tipos de usuarios:

- ❖ Usuario con derecho a crear, borrar y modificar objetos y que además puede conceder privilegios a otros usuarios sobre los objetos que ha creado.
- ❖ Usuario con derecho a consultar, o actualizar, y sin derecho a crear o borrar objetos. privilegios sobre los objetos, añadir nuevos campos, indexar, alterar la estructura de los objetos, etc.

Tabla 4.17: seguridad de base de datos

Tipos de Usuario	Accesos Permitidos
Administrador Secretaria Docente	el sistema debe identificar y autenticar a los usuarios utilizando alguno de las siguientes formas: <ul style="list-style-type: none">• Nombre y contraseña• módulos habilitados para cada usuario
Tareas que realiza cada Usuario Usa la B.D. Consulta datos Actualizar datos	

Elimina datos
Descripción <ul style="list-style-type: none">• Se debe ingresar con un nombre y contraseña• El sistema permite el acceso• Se puede realizar las diferentes tareas(consultar, modificar, añadir)

Fuente: [Elaboración propia]

5. Costos y Beneficios

5.1. Costos

El modelo COCOMO es un modelo de estimación de costes de software, orientado a la magnitud del producto final midiendo el tamaño del proyecto. Este modelo ayuda a estimar esfuerzo, tiempo, gente y costos del proyecto en este caso se utilizara el modelo de aplicación intermedio.

Teniendo hallado antes el valor de punto de función $PF=476.3$ reemplazamos en la siguiente formula:

$$KLDC = (PF * \text{Factor LDC})/1000$$

$$KLDC = (476.3 * 29)/1000$$

$$KLDC = 13812.7 / 1000$$

$$KLDC = 13,8127$$

Hallamos el valor de FAE (Factor de ajuste del esfuerzo) en base a la siguiente tabla:

Tabla 5.1: variables factor de ajustes del esfuerzo

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto

Atributos de software						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de hardware						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
Atributos de personal						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
Atributos del proyecto						
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82	
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,22	1,08	1,00	1,04	1,10	

Fuente: [Elaboración propia]

FAE = 1,15 * 1,08 * 1,00 * 1,00 * 1,00 * 1,00 * 1,07 * 0,86 * 0,91 * 0,86 * 1,00 * 0,95 * 0,91 * 0,91 * 1.08

$$FAE = 0,7599$$

Hallamos la variable del esfuerzo según la siguiente tabla:

Tabla 5.2: Constantes de modelo COCOMO

MODO	a	b	c	D
Orgánico	2.40	1.05	2.50	0.38
Semilibre	3.00	1.12	2.50	0.35
Rígido	3.60	1.20	2.50	0.32

Fuente: [Elaboración propia]

$$E = a * KLDC^b * FAE$$

$$E = 2,40 * 13,8127^{1,05} * 0,7599$$

$$E = 28,72$$

$$E = 28 \text{ personas} * \text{mes}$$

Hallamos tiempo de desarrollo según la siguiente formula:

$$T = c * (E)^d$$

$$T = 2,50 * (28,72)^{0,38}$$

$$T = 8,95$$

$$T = 8 \text{ meses}$$

Hallamos la productividad según la siguiente formula:

$$PR = LDC / E$$

$$PR = 13812,7 / 28,72$$

$$PR = 480,94$$

Hallamos número de personas para trabajar en el proyecto:

$$P = E / T$$

$$P = 28,72 / 8,95$$

$$P = 3,20$$

$$P = 3 \text{ personas}$$

Implica que es necesario 3 personas trabajando en el proyecto durante un periodo de 8 meses.

Calculamos el salario para los programadores en el tiempo ya antes calculado según la siguiente fórmula:

$$C_{sof} = (P * S_{pro}) * T$$

Dónde:

C_{sof}: Costo del software desarrollado

P: número de personas o programadores trabajando

S_{pro}: salario promedio de un programador

T: tiempo de desarrollo

Aplicamos los datos antes hallados en la fórmula tomando en cuenta que el salario de un programador es de un valor promedio de 300 dólares:

$$C_{sof} = (3 * 300) * 8$$

$$C_{sof} = 7200$$

Implica que la estimación del costo de desarrollo será de 7200 con 3 programadores en un tiempo estimado de 8 meses.

El costo de las licencias para el desarrollo es de 0 puesto que el Sistema será desarrollado sobre la plataforma Ubuntu que tiene una licencia sin costo también se usara PHP como lenguaje de programación que también tiene una licencia sin costo y otras herramientas de distribución libre, entonces se concluye que en cuanto a software el costo es de cero.

Las características de los servicios para la implantación, alojamiento del Sistema de la empresa Sysoft-Bo es la siguiente:

Tabla 5.3: Costos de Alojamiento Web de sysoft-Bo

Plan Básico 400 Bs Anual						
Espacio en Disco	transferencia	Cuentas Email	Base de Datos Mysql	Cuentas Ftp	Dominios Apuntados	Subdominios
1000 MB	1000 MB	10	2	2	1(.COM, .NET, .ORG)	1
Plan Intermedio 700 Bs Anual						
Espacio en Disco	transferencia	Cuentas Email	Base de Datos Mysql	Cuentas Ftp	Dominios Apuntados	Subdominios

3000 MB	3000 MB	20	5	3	1(.COM, .NET, .ORG)	6
Plan Avanzado 1000 Bs Anual						
Espacio en Disco	transfere ncia	Cuentas Email	Base de Datos Mysql	Cuent as Ftp	Dominios Apuntados	Subdo minios
Ilimitado	Ilimitado	Ilimitado	Ilimitado	Ilimitad o	1(.COM, .NET, .ORG)	Ilimitad o

Fuente: [Elaboración propia]

El sistema Web se implantara y alojara con estos costos:

Tabla 5.4: Costos de alquiler de Alojamiento Web

Plan Avanzado 1000 Bs Anual						
Espacio en Disco	transferenc ia	Cuentas Email	Base de Datos Mysql	Cuenta s Ftp	Dominios Apuntado s	Subdo minios
Ilimitado	Ilimitado	Ilimitado	Ilimitado	Ilimitado	1(.COM, .NET, .ORG)	Ilimitad o

Fuente: [Elaboración propia]

Por lo tanto el costo total del proyecto se detalla en la siguiente tabla:

Tabla 5.5: Costos totales

DESCRIPCION	CANTIDAD	COSTO MENSUAL \$us	MESES	TOTAL
Desarrolladores	3	300	8	7200\$
Software	1	0	-	0
Alojamiento web	1	12,3	-	12,3\$
TOTAL COSTOS				7212.3 \$us

Fuente: [Elaboración propia]

5.2. Beneficios

Para la realización del Sistema Web de Registro y Seguimiento Académico se utilizaron herramientas libres con código abierto lo cual no implica ningún costo.

Teniendo a si los siguientes beneficios para la academia de música “Mozart”

Ahorro de costes de hardware y software (solo se necesita un computador)

Fácil de usar

Facilita el trabajo de Inscripción de estudiantes

Facilita el trabajo a distancia

Es escalable y de rápida actualización

Provoca menos errores y problema de redundancia de datos

Los datos son más Seguros

6. Conclusiones Y Recomendaciones

Por tanto se concluye con el Sistema Web para el Registro y Seguimiento Académico, para la Academia de Música “Mozart” es un aporte bastante importante tomando en cuenta que anteriormente se realizaba de forma manual.

6.2. Conclusiones

Después de haber desarrollado el sistema, se concluyó con los objetivos planteados en el presente proyecto, desarrollando un Sistema Web para el Registro y Seguimiento Académico. Que coadyuvara en la toma de decisiones y ayudara al crecimiento de la Academia de Música Mozart.

Aplicando con éxito las normas de calidad y las herramientas de programación para que tenga alta usabilidad, funcionalidad y fiabilidad.

Una vez concluido el presente proyecto, se llegaron a las siguientes conclusiones:

- Se Diseñó el Sistema Web para el Registro y Seguimiento académico de estudiantes con interfaces amigables entre usuario y sistema.
- Registra y Asigna a estudiantes de manera oportuna los cursos correspondientes
- Automatiza la asignación de notas el docente de turno ingresando a su perfil y en actas de calificaciones anota la nota que le corresponde a cada estudiante.
- Genera informes confiables y oportunos de los estudiantes registrados.
- Genera informes del total de estudiantes inscritos por cada curso.

En forma general se concluye que todos los objetivos planteados fueron alcanzados desarrollando el sistema Web para el Registro y Seguimiento Académico. Lo cual permite disminuir los problemas frecuentes que se presentan en la parte de administración de la Academia de Música Mozart, como la duplicidad, pérdida, deterioro de la información u otros.

6.3. Recomendaciones

Hoy en día es recomendable utilizar la metodología orientada a la WEB en los procesos de investigación, las recomendaciones que se deben considerar para el sistema son:

- En primer lugar, capacitar a los usuarios administradores que ingresen al sistema para un buen manejo.
- Que los usuarios mantengan su contraseña y nombre de usuario como privacidad
- Ampliar el sistema si así lo requiere la Academia de Música Mozart.
- Realizar siempre copias de seguridad periódicamente en la base de datos y obtener un respaldo en un proceso que así lo requieran.
- Realizar evaluaciones periódicas del sistema y de la información para determinar las nuevas necesidades
- Realizar siempre el mantenimiento respectivo

7. Bibliografía

- Alvarez, M. A. (09 de Mayo de 2001). *que es Php*. Recuperado el 28 de Mayo de 2020, de <https://desarrolloweb.com/articulos/392.php>
- Alvarez, M. A. (2 de Enero de 2014). *Que es MVC*. Recuperado el 27 de Mayo de 2020, de <https://desarrolloweb.com/articulos/que-es-mvc.html>
- Alvarez, M., López, D., & Gutierrez, M. (17 de 01 de 2013). *Taller de PHP*. Recuperado el 24 de febrero de 2020, de <http://www.desarrolloweb.com/manuales/6/>
- Baez, S. (20 de Octubre de 2012). *Sistema Web*. Recuperado el 27 de Mayo de 2020, de <http://knowdo.org/knowledge/39-sistemas-web>
- Bembibre, C. (1 de Abril de 2009). *Definicion de Academico*. Recuperado el 27 de Mayo de 2020, de <https://www.definicionabc.com/social/academico.php>
- Cobo, C. (2011). *La Tecnologia*.
- Cocomo. (11 de Diciembre de 2017). *Cocomo*. Recuperado el 27 de mayo de 2020, de Center For System and Software Engineering: <https://es.wikipedia.org/wiki/COCOMO>
- Cooper, R. (2004). *Apache Práctico* (1ra Edicion ed.). Madrid: Anaya Multimedia. Recuperado el 15 de febrero de 2020, de http://linux.ciberaula.com/articulo/linux_apache_intro
- CUATRORIOS. (2011). *Norma ISO-9126 para análisis de software*. Recuperado el 10 de febrero de 2020, de http://www.cuatrorios.org/index.php?option=com_content&view=article&id=163:norma-iso-9126-para-an%C3%A1lisis-de-software&catid=39:blogsfeeds

- EcuRed. (2007). *Características de MagicDraw*. Recuperado el 12 de febrero de 2020, de <http://www.ecured.cu/index.php/MagicDraw>
- Editorial, E. (24 de Abril de 2019). *Reporte digital*. Recuperado el 15 de Junio de 2020, de <https://reportedigital.com/seguridad/seguridad-del-software/>
- Engineering, I. f. (07 de 09 de 2012). *UWE - UML - BASED WEB ENGINEERING*. Recuperado el 18 de febrero de 2020, de <http://uwe.pst.ifi.lmu.de/>
- Galeano, L. (03 de Noviembre de 2012). *Metodologia Uwe*. Recuperado el 28 de Mayo de 2020, de <https://elproyectedeluisgaliano.blogspot.com/2012/11/metodologia-uwe-aplicada-mi-solucion.html>
- Inco. (2 de Febrero de 2018). *que es MariaDB*. Recuperado el 28 de Mayo de 2020, de <https://www.incosa.com.uy/blog/que-es-mariadb/>
- Koch, N. (Enero de 06 de 2000). *Software Engieenerig for Adaptative Hypermedia System for Reference Modeling techniques an developement Process*. Recuperado el 27 de Mayo de 2020, de Calidad del Producto y Proceso Software: https://books.google.com.bo/books?id=MY0zoXYFVd8C&pg=PA598&lpg=PA598&dq=ingenieria+de+software+koch&source=bl&ots=VgrifjPDMH&sig=ACfU3U0bCwiTTkFttc02uT0k0av5_dbfK
- Leon, D. E. (2016). *La Teconologia*. Ecuador.
- McGraw, G. (29 de Mayo de 2016). *Ingenieria de Software*. Recuperado el 15 de Junio de 2020, de <https://www.garymcgraw.com/>
- Miguel Angel, A. (19 de Septiembre de 2012). *Manual de JQuery*. Recuperado el 05 de febrero de 2020, de <http://www.desarrolloweb.com/manuales/manual-jquery.html>

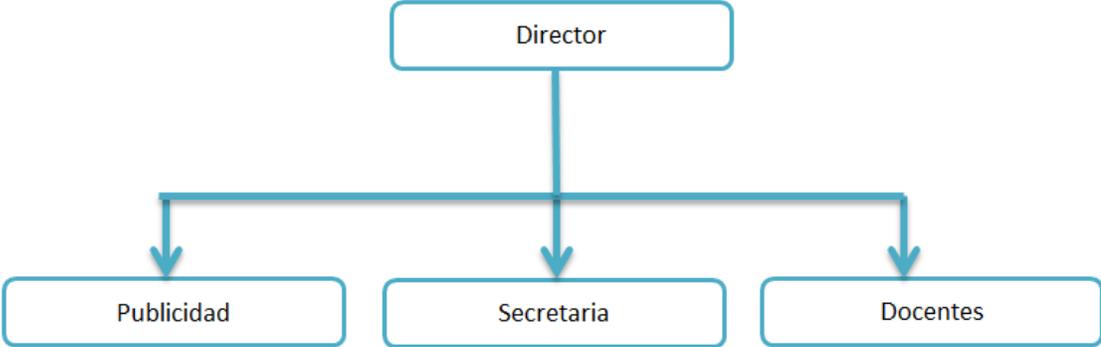
- pineda, j. m. (03 de noviembre de 2016). *definicion CodeIgniter*. Recuperado el 10 de marzo de 2020, de Qué es CodeIgniter y cuáles son algunas de sus ventajas: <https://www.coriaweb.hosting/codeigniter-cuales-algunas-ventajas/>
- Porto, J. P. (21 de diciembre de 2019). *definicion de css*. Recuperado el 12 de marzo de 2020, de definicion de css: <https://definicion.de/css/>
- Pressman, R. S. (2010). *Ingeniería del software un Enfoque Practico* (7ma. ed.). Mexico: Mc Graw Hill. Recuperado el 23 de febrero de 2020
- Prieto, R. M. (24 de Julio de 2017). *Modelos y Estándares de Calidad Aplicados al Sistema de Informacion*. Recuperado el 27 de mayo de 2020, de <https://unidad4rociomp.blogspot.com/2017/07/46.html>
- Raffino, M. E. (2019 de Diciembre de 11). *Sistema*. Recuperado el 27 de Mayo de 2020, de <https://concepto.de/sistema/>
- Raffino, M. E. (14 de Febrero de 2020). *Base de Datos*. Recuperado el 27 de Mayo de 2020, de <https://concepto.de/base-de-datos/>
- Redaccion. (19 de Julio de 2019). *Definicion de Seguimiento* . Recuperado el 27 de Mayo de 2020, de <https://conceptodefinicion.de/seguimiento/>.
- Redaccion. (14 de Noviembre de 2019). *Definicion de Registro*. Recuperado el 29 de Mayo de 2020, de <https://conceptodefinicion.de/registro/>.
- Sarmiento, M. (28 de Junio de 2017). *Normalizacion de la Base de Datos*. Recuperado el 28 de Mayo de 2020, de <http://www.marcossarmiento.com/2017/06/28/normalizacion-de-base-de-datos/>
- Solis, J. (26 de septiembre de 2014). *¿Qué es Bootstrap y cómo funciona en el diseño web?* Recuperado el 02 de abril de 2020, de ¿Qué es Bootstrap y cómo funciona en el diseño web?:

<https://www.arweb.com/blog/%C2%BFque-es-bootstrap-y-como-funciona-en-el-diseno-web/>

Wikipedia. (03 de Mayo de 2020). *Maria DB*. Recuperado el 28 de Mayo de 2020, de <https://es.wikipedia.org/wiki/MariaDB>

Anexo

Anexo A: Organigrama de la academia de Música “Mozart”



Fuente: [Elaboración Propia]

