

UNIVERSIDAD MAYOR DE SAN ANDRÉS

FACULTAD DE CIENCIAS PURAS Y NATURALES

CARRERA DE INFORMÁTICA



PROYECTO DE GRADO

**“SOFTWARE DE GESTIÓN ACADÉMICA CENTRALIZADA
PARA INSTITUTOS DE FORMACIÓN SUPERIOR
DEL DEPARTAMENTO DE LA PAZ”**

CASO: DIRECCIÓN DEPARTAMENTAL DE EDUCACIÓN LA PAZ

**PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS**

POSTULANTE: YERSON IVÁN SALAZAR RODRIGUEZ

TUTOR: M.Sc. FRANZ CUEVAS QUIROZ

Nuestra Señora de La Paz – Bolivia

2022

HOJA DE CALIFICACIONES

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA

Proyecto de Grado:
“SOFTWARE DE GESTIÓN ACADÉMICA CENTRALIZADA
PARA INSTITUTOS DE FORMACIÓN SUPERIOR
DEL DEPARTAMENTO DE LA PAZ”
CASO: DIRECCIÓN DEPARTAMENTAL DE EDUCACIÓN LA PAZ

Presentado por: Yerson Iván Salazar Rodríguez

Para optar el grado Académico de Licenciado en Informática
Mención Ingeniería de Sistemas Informáticos

Nota Numeral:

Nota Literal:

Ha sido:

Director de la Carrera de Informática: Lic. Hermenegildo Nogales

Tutor: M.Sc. Franz Cuevas Quiroz

Presidente Tribunal: M.Sc. Moisés Martin Silva Choque

Tribunal: M.Sc. Jorge Humberto Terán Pomier

Tribunal: Lic. Juan Gonzalo Contreras Candia

Tribunal: Lic. Reynaldo Javier Zeballos Daza



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS

LICENCIA DE USO

El usuario está autorizado a:

- a) Visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) Copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) Copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

*Este trabajo está dedicado a mis amados padres
Mamita Norberta (†), quiero brindarte esta anhelada alegría
allá en el cielo.*

*Papá José, de ti aprendí que con esfuerzo, sacrificio y honestidad
se cosechan los logros en la vida.*

*Hijito Saúl (†), no alcanzaste ver la luz de la vida,
estarás en mi corazón por siempre.*

AGRADECIMIENTOS

*A mi esposa Carlita,
quien ha estado a mi lado incondicionalmente en todo momento
con su inconmensurable amor y comprensión nunca dudó
que lograré mis objetivos de vida a pesar de las adversidades.*

*A mis hermanos Fidel y Dina,
por ser mi fortaleza y motivación en la consecución de esta meta
y con su invaluable apoyo me han ayudado a superar obstáculos.*

*Al M.Sc. Franz Cuevas Quiroz,
como tutor del presente trabajo y su valiosa orientación académica
en la fundamentación conceptual de esta obra.*

*A las autoridades de la Dirección Departamental de Educación La Paz:
Lic. José Antonio Apala, Subdirector de Educación Superior
Lic. Juan Carlos Copa, Responsable de Formación Técnica y Artística
por **confiar** en mi capacidad y brindarme la **oportunidad** de llevar a cabo
este proyecto en beneficio de la comunidad estudiantil.*

¡Muchas Gracias!

Juán Salazar R.

ivancho.sr@gmail.com

RESUMEN

La Dirección Departamental de Educación La Paz -DDE LP-, a través de la Subdirección de Educación Superior de Formación Profesional, centraliza la información de los institutos de formación superior técnica en lo referido a gestión académica a fin de brindar la emisión de certificaciones y otros trámites a la población estudiantil del departamento. El presente trabajo desarrolla un sistema de gestión académica con el empleo adecuado de los fundamentos de ingeniería de software para beneficio de los operadores académicos.

En la estructuración del proyecto se emplea la metodología ágil XP centrada en potenciar las relaciones interpersonales en el equipo de desarrollo propiciando un buen clima de trabajo y constante retroalimentación con el cliente. También se apoya en la metodología UWE basada en UML y sus modelos de construcción para el desarrollo de aplicaciones web. Para la programación se emplean herramientas de software libre con la plataforma de base de datos MySQL, framework Laravel basado en PHP, VUE basado en JavaScript y la librería Tailwind basada en CSS que combinados ofrecen características de seguridad, operabilidad y experiencia de usuario acorde a las necesidades. Un aspecto de importancia en este trabajo es la generación de reportes estadísticos variados y precisos que sirven como indicadores para la toma de decisiones operativas o estratégicas en el ámbito de gestión académica y políticas educativas de la DDE LP. Finalmente, para la estimación de la calidad de software y resultados, se toma como parámetro inicial la usabilidad con cual se determina el nivel de capacidad del producto para ser entendido, aprendido, usado y resultar atractivo en la experiencia de usuario.

Con estas características el mayor beneficio será para la población estudiantil del departamento de La Paz que acude a esta institución para la tramitación de las distintas certificaciones administrativas.

PALABRAS CLAVE: gestión académica, institutos, formación superior, software

METODOLOGÍA: XP

ABSTRACT

The Education Departmental Directorate La Paz -DDE LP- (*Dirección Departamental de Educación*), through the Higher Education for Vocational Training Subdirectorate (*Subdirección de Educación Superior de Formación Profesional*), centralizes information of the higher technical training institutes regarding academic management in order to provide certifications and other procedures to the student population of the department. The present work develops an academic management system with use of software engineering fundamentals for the benefit of academic operators.

XP agile methodology is used for project structuring, focused on enhancing interpersonal relationships in the development team, promoting the good work environment and constant feedback with client. It also relies on the UML-based UWE methodology and the construction models for web application development. For programming, free software tools are used with the MySQL database platform, Laravel framework on PHP-based, VUE on JavaScript-based and Tailwind library on CSS-based that combined offer security features, operability and user experience according to the requirements. An important aspect in this work is the generation of varied and precise statistical reports that serve as indicators for making operational or strategic decisions in the field of academic management and educational policies of the DDE LP. Finally, for estimation of software quality and results, usability is taken as an initial parameter with which the level of capacity of the product to be understood, learned, used and be attractive in the user experience is determined.

With these characteristics, the greatest benefit will be for the student population of the department of La Paz who comes to this institution for the processing of the different administrative certifications.

KEYWORDS: academic management, institutes, higher education, software.

METHODOLOGY: XP

ÍNDICE

	Pág.
CAPÍTULO I - INTRODUCCIÓN	1
1.1. ANTECEDENTES	4
1.1.1. Institucionales	4
1.1.2. Proyectos Similares	7
1.2. PLANTEAMIENTO DEL PROBLEMA.....	8
1.3. JUSTIFICACIÓN.....	9
1.4. OBJETIVOS.....	10
1.4.1. Objetivo General.....	10
1.4.2. Objetivos Específicos.....	11
1.5. ALCANCES Y LÍMITES	11
1.5.1. Alcances	11
1.5.2. Límites	12
1.6. IMPORTANCIA DEL ESTUDIO.....	13
CAPÍTULO II - MARCO TEÓRICO	14
2.1. AUTOMATIZACIÓN DE LA GESTIÓN ACADÉMICA	14
2.1.1. Gestión Académica	14
2.1.2. Automatización de Procesos	15
2.2. CONCEPTOS SOBRE SISTEMAS	16
2.2.1. Sistema.....	16
2.2.2. Sistema Informático	16
2.2.3. Sistema Web	17
2.3. INGENIERÍA DE SOFTWARE.....	18
2.3.1. Definición.....	18
2.3.2. Ciclo de vida del Software	19
2.3.3. Objetivos de la Ingeniería de Software	20
2.4. METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	20
2.4.1. Metodologías tradicionales	21
2.4.2. Metodologías ágiles	22
2.4.3. Comparación de metodologías.....	24

2.5.	METODOLOGÍA XP	25
2.5.1.	Valores XP	25
2.5.2.	Roles de los integrantes	26
2.5.3.	Etapas del proceso XP	28
2.5.4.	Herramientas en XP	30
2.6.	METODOLOGÍA UWE	32
2.7.	CALIDAD DE SOFTWARE	35
2.8.	HERRAMIENTAS	36
 CAPÍTULO III - MARCO APLICATIVO		37
3.1.	EXPLORACIÓN INICIAL	37
3.2.	APLICACIÓN DE LA METODOLOGÍA	41
3.2.1.	Planeación	41
3.2.2.	Diseño	45
3.2.3.	Construcción	53
3.3.	SEGURIDAD	57
3.3.1.	Nivel de Aplicación	58
3.3.2.	Nivel de Base de Datos	59
3.4.	CALIDAD DE SOFTWARE	60
3.4.1.	Usabilidad	60
 CAPÍTULO IV - CONCLUSIONES Y RECOMENDACIONES		66
4.1.	CONCLUSIONES	66
4.2.	RECOMENDACIONES	67
 BIBLIOGRAFÍA		68

ÍNDICE DE FIGURAS

	Pág.
Figura 1. 1: Estructura organizacional de la DDE LP, aprobado por R.M. 092/2011	6
Figura 2. 1: Diseño responsivo adaptable a varios dispositivos.....	18
Figura 2. 2: Esquema general de metodologías tradicionales.	21
Figura 2. 3: Esquema general de las metodologías ágiles.....	23
Figura 2. 4: Proceso de la programación extrema	28
Figura 2. 5: Esquema y componentes de una Historia de Usuario	30
Figura 2. 6: Conjunto de modelos componentes de la metodología UWE.....	32
Figura 2. 7: Ejemplo de un diagrama especificando al actor y sus casos de uso	32
Figura 2. 8: Ejemplo de diagrama de clases	33
Figura 2. 9: Estereotipos UWE para la estructura de navegación.....	33
Figura 2. 10: Modelo de presentación de una página de inicio	34
Figura 2. 11: Modelo de estructura del proceso.....	34
Figura 3. 1: Interfaz de usuario del Sistema de Institutos	39
Figura 3. 2: Diagrama de casos de uso general	41
Figura 3. 3: Diagrama de casos de uso para el Software de Gestión Académica	47
Figura 3. 4: Diagrama de clases para el Software de Gestión Académica	48
Figura 3. 5: Modelo Relacional para el Software de Gestión Académica	49
Figura 3. 6: Modelo Entidad – Relación para el Software de Gestión Académica	50
Figura 3. 7: Modelo de Navegación	51
Figura 3. 8: Modelo de presentación general.....	52
Figura 3. 9: Modelo de presentación en un dispositivo móvil.....	52
Figura 3. 10: Pantalla de inicio de sesión	53
Figura 3. 11: Interfaz de usuario. ROL ADMINISTRADOR	54
Figura 3. 12: Interfaz de usuario. ROL OPERADOR	54
Figura 3. 13: Interfaz de usuario ROL COLABORADOR.....	55
Figura 3. 14: Interfaz de usuario en dispositivo móvil. ROL OPERADOR	55
Figura 3. 15: Interfaz de usuario en tableta. ROL COLABORADOR.....	56
Figura 3. 16:Formulario para registro de un nuevo instituto	56
Figura 3. 17: Ejemplo de reporte generado	57

ÍNDICE DE TABLAS

	Pág.
Tabla 2. 1: Comparación entre las metodologías ágiles	24
Tabla 2. 2: Patrón para el campo Descripción en una Historia de Usuario	31
Tabla 2. 3: Patrón de una tarjeta CRC	31
Tabla 2. 4: Ejemplo aplicado al patrón.....	31
Tabla 2. 5: Herramientas a emplearse en el desarrollo del proyecto	36
Tabla 3. 1: Resumen de Historias de Usuario	45
Tabla 3. 2: Valoración en la escala de Lickert	61
Tabla 3. 3: Conjunto de preguntas formuladas a los usuarios.....	62
Tabla 3. 4: Resultado obtenido en detalle de las encuestas.....	63
Tabla 3. 5: Resultados de la valoración media de cada usuario	63
Tabla 3. 6: Conformidad media a cada pregunta planteada.....	64

CAPÍTULO I

INTRODUCCIÓN

Para las organizaciones, la información es un recurso esencial utilizado al desempeñar operaciones diarias, en este sentido es importante tenerla disponible en el momento adecuado para que pueda ser analizada por los tomadores de decisiones en el nivel operativo y estratégico. Contar con este conocimiento es fundamental tanto para alcanzar los objetivos y metas corporativas como para elevar el nivel de productividad y calidad de servicio al cliente, sin embargo a pesar de que la información es permanentemente almacenada de manera segura, es aún más importante organizarla y generar resultados estadísticos.

Frecuentemente la información disponible pero no interpretada de manera práctica es una limitante para la acertada toma de decisiones, por esto día a día las empresas apuestan por la implementación de distintas herramientas y soluciones que les ayuden a alcanzar sus metas y simplificar la obtención de resultados confiables para llevar a cabo un análisis válido.

Para este efecto las Tecnologías de Información y Comunicación (TIC) resultan una herramienta muy valiosa que permite recolectar, procesar y almacenar datos que son generados de la misma operación del negocio en el día a día. Así se facilita el acceso a la información y se reduce el margen de error que pudiera existir al realizar el registro de las operaciones, disminuyendo el trabajo redundante y optimizando el flujo de la información. Estas tecnologías han revolucionado la manera en que las empresas pueden operar, comunicarse, planificar y analizar resultados de tal forma que se mejora la eficacia y rentabilidad al momento de brindar sus servicios al público.

El continuo avance de las Tecnologías de la Información y Comunicación obliga a obtener un mayor provecho de las mismas para agilizar procesos monótonos y manuales a objeto de lograr operaciones automatizadas, sistematización de datos y su manejo metódico con la finalidad de facilitar la gestión de dichos procesos para los distintos actores. En tal sentido es imperioso el empleo de sistemas de gestión de información por lo que están disponibles todas las vías de acceso que los recursos tecnológicos ofrecen.

En el ámbito educativo, al margen de utilizar estas herramientas para el proceso enseñanza-aprendizaje, es necesario emplearlas para realizar una adecuada administración académica y tener una mejor gestión de información que tome en cuenta desde las entidades educativas junto al plantel docente y administrativo, además de la población estudiantil junto a su registro de calificaciones y seguimiento académico con el fin de contar con estadísticas que permitan la correcta toma de decisiones de carácter operativo y estratégico por parte de las autoridades educativas.

Con el objetivo de centralizar toda la información relacionada a la gestión académica, el Ministerio de Educación cuenta con el Sistema de Información Educativa S.I.E. que es un compendio de varios sistemas orientados principalmente a la administración del subsistema de educación regular con un registro de unidades educativas y su geolocalización, diplomas de bachiller, datos personales de los estudiantes y estadísticas de indicadores educativos. Este sistema toma en cuenta de manera muy general a los institutos de formación superior técnica y por ende a la población cursante en este ámbito.

La Dirección Departamental de Educación La Paz, a través de la Subdirección de Educación Superior de Formación Profesional, está encargada de centralizar la información de los institutos de formación superior técnica en lo referido a datos de los institutos educativos y de los estudiantes a fin de brindar un adecuado servicio a la población estudiantil del departamento para la emisión de certificaciones de estudios, calificaciones, convalidaciones y otros aspectos relacionados a la validación de sus estudios.

En este sentido, desde la gestión 2013 la institución cuenta con un sistema implementado de manera experimental por lo cual no ha recibido mantenimiento alguno, sin embargo por ser la única herramienta a disposición se ha llegado a depender de ella en gran manera. El sistema ha quedado obsoleto frente a las necesidades requeridas por los usuarios en cuanto a operabilidad y obtención de resultados.

El presente Proyecto de Grado desarrolla un sistema basado en el adecuado empleo de los fundamentos de ingeniería de software e implementación de modernas interfaces de usuario para beneficio de los operadores. Permitirá mejorar procedimientos administrativos, sistematizar la labor del usuario y generar resultados estadísticos útiles que permitan la toma de decisiones en la institución.

Este trabajo está organizado en 4 capítulos:

- El capítulo 1 describe la introducción al caso de estudio donde se parte de la visión general y antecedentes hasta llegar a la identificación del problema para definir objetivos, justificaciones, alcances y límites del proyecto.
- El capítulo 2 presenta el marco teórico para el desarrollo de la metodología adecuada en la aplicación del proyecto desde la recopilación de la información hasta la implementación del producto planteado tomando en cuenta conceptos, metodologías, técnicas y herramientas convenientes en el desarrollo del sistema.
- El capítulo 3 desarrolla el marco aplicativo que explica en detalle el procedimiento a seguir y pone en práctica lo abordado en el capítulo previo hasta la culminación del producto con la obtención y validación de los resultados para realizar un análisis de calidad de software.
- El capítulo 4 expone las conclusiones acordes a los objetivos planteados y lo alcanzado con la aplicación de la metodología empleada y finalmente se sugieren recomendaciones a fin de profundizar los logros conseguidos.

1.1. ANTECEDENTES

En el ámbito de la gestión académica, se han creado soluciones de software de distinta naturaleza y a medida para unidades educativas de formación regular, institutos de formación técnica, carreras profesionales de universidades, con características específicas según el requerimiento de cada institución bajo su propio enfoque.

La Dirección Departamental de Educación La Paz, cuenta con un sistema que no ha recibido mantenimiento desde su implementación en 2013, por lo cual –debido al volumen de datos que maneja– presenta deficiencias de operabilidad y obtención de resultados. En reunión con el personal de la institución se ha visto la necesidad de rediseñar el software, incorporar mejores características funcionales en el registro de datos, búsqueda de información y generación de reportes. El presente proyecto centralizará la información académica de institutos de formación superior en el Departamento de La Paz, bajo los antecedentes que se citan a continuación.

1.1.1. Institucionales

Las Direcciones Departamentales de Educación –DDE– han sido establecidas en el Artículo N° 78 de la Ley Educativa N° 070 Avelino Siñani - Elizardo Pérez del 20 de Diciembre de 2010, dejando de ser SEDUCA's. La Dirección Departamental de Educación La Paz **DDE LP**, se crea por D.S. N° 0813 del 9 de Marzo de 2011 a objeto de reglamentar su estructura, composición y funciones. Se constituye en una entidad pública descentralizada dependiente del Ministerio de Educación como persona jurídica de derecho público, con patrimonio propio y autonomía de gestión administrativa, financiera, legal y técnica. Tiene sede en la ciudad de La Paz y competencia a nivel departamental sobre la administración y gestión de la educación. La Subdirección de Educación Superior dentro de las funciones conferidas como parte de la DDE LP, es responsable de emitir certificaciones administrativas requeridas tanto por la población estudiantil como por los institutos técnicos.

Según lo establecido en el mencionado D.S., el organigrama institucional de la DDE LP establece la organización educativa en tres niveles:

- Subsistema de Educación Regular:
 - ✓ Educación inicial
 - ✓ Educación primaria
 - ✓ Educación secundaria.
- Subsistema de Educación Superior de Formación Profesional:
 - ✓ Educación superior técnica, tecnológica, lingüística, artística
 - ✓ Educación superior universitaria
 - ✓ Formación de maestros
- Subsistema de Educación Alternativa y Especial
 - ✓ Educación de adultos
 - ✓ Educación especial
 - ✓ Post alfabetización

Este proyecto realiza el desarrollo e implementación del sistema centralizado de gestión académica para institutos de formación superior del Departamento de La Paz para su empleo por parte de la Subdirección de Educación Superior de Formación Profesional. Se toma en cuenta una adecuada aplicación de los fundamentos de ingeniería de software que abarca aspectos de metodologías de desarrollo, herramientas de programación actuales, modernas interfaces de usuario y obtención de resultados en forma de reportes estadísticos, todo basado en la reglamentación definida en la R.M. 001/2021 “Normas generales para la gestión institucional académica y administrativa de la formación superior técnica, tecnológica lingüística y artística”.

La figura 1.1 muestra la estructura organizacional de la DDE LP donde se aprecia la ubicación en contexto de la Subdirección de Educación Superior.

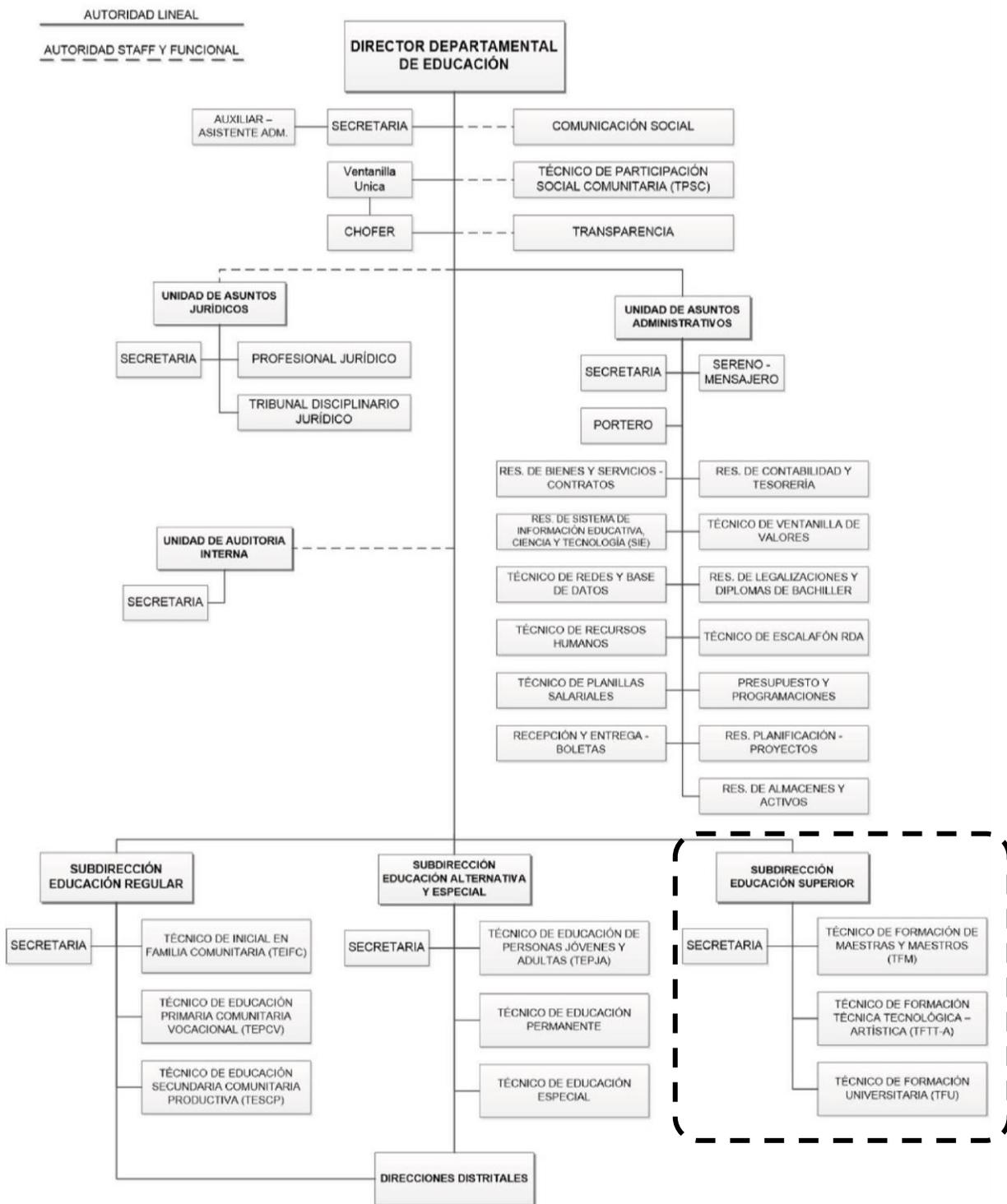


Figura 1. 1: Estructura organizacional de la DDE LP, aprobado por R.M. 092/2011
 Fuente: Unidad de Asuntos Administrativos de la DDE LP

1.1.2. Proyectos Similares

Revisada la documentación, en la modalidad Proyecto de Grado se han desarrollado varios sistemas académicos para unidades educativas de educación regular, y en menor medida para institutos técnicos de educación superior. Los mismos satisfacen requerimientos específicos de las entidades interesadas, entre ellos se citan:

- SISTEMA DE GESTIÓN ACADÉMICA PARA EL INSTITUTO NORMAL SUPERIOR SIMÓN BOLÍVAR, por Diego Omar Chambi Lima, 2007. Tiene como objetivo principal implementar un sistema de gestión académico y seguimiento de adaptaciones curriculares, con el fin de minimizar costos y tiempo en el procesamiento de información institucional. Se utilizó la metodología RUP, desarrollo en PHP y se utilizó el gestor de base de datos MySQL.
- SISTEMA DE GESTIÓN ACADÉMICA PARA EL INSTITUTO TECNOLÓGICO SUPERIOR DE MECAPACA, por Roly Carlos Mamani Mamani, 2013. Cuenta con los habituales módulos de: registro de estudiantes nuevos, matriculación, inscripción, registro y control de notas, seguimiento académico y finalmente emisión de notas. Está desarrollado en lenguaje PHP 5, gestor de base de datos MySQL y plataforma Apache como servidor. Se ha empleado la metodología SCRUM basada en procesos iterativos e incrementales, además se ha realizado el modelado mediante la herramienta UML.
- SISTEMA WEB INTEGRADO DE GESTIÓN ACADÉMICA ADMINISTRATIVA CASO: C.E.C.O.M.P., por Diego Paredes Mendoza, 2015. Desarrolla e implementa un Sistema Web Integrado de Gestión Académica y Administrativa para el instituto CECOMP que coadyuva en los procesos de estas áreas y proporciona mecanismos de gestión de la información en tiempo real. Se utiliza el lenguaje de programación PHP, MySQL como sistema gestor de base de datos y Apache como servidor Web.

Como se observa en los citados títulos, estos sistemas están orientados hacia los requerimientos específicos de cada institución. No se ha planteado solución alguna a nivel Departamento de La Paz, que gestione de manera centralizada los aspectos de administración académica inherentes al ámbito educativo.

1.2. PLANTEAMIENTO DEL PROBLEMA

El Subsistema de Educación Superior está encargado de centralizar la información académica de los institutos técnicos de formación superior públicos, privados y de convenio en lo que respecta a mallas curriculares, pénsum de asignaturas, centralizadores de calificaciones, kárdex de estudiantes, docentes y administrativos, a objeto de emitir las correspondientes certificaciones administrativas, siendo de importancia para la población estudiantil en la continuación de sus trámites académicos ante otras instancias.

En una primera instancia los usuarios más involucrados son los responsables académicos de cada instituto que están encargados de introducir cuidadosamente la información mencionada, posteriormente los técnicos de seguimiento en la DDE LP acceden a esta información para verificar y emitir las certificaciones respectivas. En este sentido, existe un sistema web implementado en la gestión 2013 de manera experimental del cual se han identificado los siguientes problemas.

- Por su carácter experimental e imprecisiones administrativas de carácter contractual, el software no ha recibido mantenimiento por parte de los consultores responsables.
- Las interfaces de usuario han quedado obsoletas en comparación con los actuales paradigmas de usabilidad y otras características requeridas por los operadores.
- Los reportes emitidos son de carácter básico, se aprecia poco manejo de lenguaje de consultas a la base de datos en cuanto al relacionamiento o cruce de información entre tablas.

- Se necesita tener mejor aprovechamiento de toda la información ya contenida en la base de datos para emitir nuevos y variados reportes estadísticos organizados metódicamente que sean de utilidad para todos los operadores bajo sus correspondientes privilegios.
- Por el volumen de datos ya introducido, aproximadamente desde la gestión 2018 el sistema viene presentando deficiencias de rendimiento puntualmente en la validación de datos, generación de reportes tanto en formato como en utilidad, un solo motor de búsqueda con elementos parametrizables de poca utilidad.
- Estos aspectos repercuten en la operabilidad del sistema por parte de los responsables académicos en los institutos y de los técnicos de seguimiento en la DDE LP, lo cual influye negativamente en la obtención de información de manera oportuna.
- En consecuencia, todo lo mencionado genera perjuicio tanto para los institutos como para la población estudiantil que requieren de los distintos trámites de certificación de manera oportuna.

En consideración a lo expuesto, se plantea el problema:

¿Cómo mejorar la gestión académica en las tareas de certificación administrativa para los trámites académicos y obtener información estadística precisa que permita una correcta toma de decisiones en la Subdirección de Educación Superior de la Dirección Departamental de Educación La Paz?

1.3. JUSTIFICACIÓN

La información es un recurso intangible importante que administrado de la manera sistemática agiliza distintos procesos administrativos logrando ahorro de tiempo y por ende reducción de costos operacionales con la obtención de resultados oportunos para una adecuada toma de decisiones operativas y estratégicas.

Los usuarios en una primera instancia son los operadores académicos de los institutos encargados de introducir la información relacionada al seguimiento (datos de estudiantes, calificaciones y otros). En una segunda etapa los técnicos de la DDE LP acceden a esta información para brindar los servicios definidos en la R.M. 01/2021 en cuanto a trámites académico-administrativos. De este modo con la implementación del software el mayor beneficiado será el estudiante, ya que podrá realizar los diferentes trámites de manera rápida y confiable, tendrá la posibilidad de verificar su seguimiento académico oficial mediante consultas a través de la página web institucional.

Al ser una institución pública, en mérito a la Ley N° 164 de 2011 y ratificada con el D.S. N° 3251 de 2017 que reglamentan la implementación de software libre en todos los niveles del Estado, el sistema se desarrollará sobre una plataforma de libre distribución empleando los entornos Laravel y Vue, concebidas sobre los lenguajes PHP y javascript, respectivamente. Como gestor de Base de Datos se empleará MySQL por ser un estándar muy difundido y con amplia aceptación a nivel de seguridad.

En hardware, la institución cuenta con un equipamiento de data-center en su sala de servidores para la ejecución de varios sistemas informáticos con una apreciable capacidad de almacenamiento. Los mismos están debidamente configurados con servicios para accederlos vía web con las correspondientes políticas de seguridad. Por estos motivos no se incurrirán en gastos de implementación en cuanto a equipamiento.

1.4. OBJETIVOS

1.4.1. Objetivo General

Desarrollar e implementar el Software de Gestión Académica Centralizada para institutos de formación superior del Departamento de La Paz, el cual permitirá mejorar la organización y control de los procesos de certificación administrativa que brinda la institución a la población estudiantil.

1.4.2. Objetivos Específicos

- Realizar el modelado de la base de datos para el almacenamiento de información sobre: institutos, estudiantes, docentes, calificaciones, mallas curriculares y datos parametrizables.
- Diseñar e implementar interfaces de usuario que tomen en cuenta procedimientos ágiles para los usuarios en las operaciones CRUD (Create, Read, Update, Delete) de los elementos mencionados.
- Definir la seguridad del software mediante roles y privilegios en niveles de acceso para un super usuario, técnicos de seguimiento en la DDE LP, responsables académicos en los institutos.
- Generar reportes estadísticos bajo distintos criterios requeridos por los técnicos de seguimiento en la DDE LP con fines administrativos para la obtención de información sistematizada y la toma de decisiones.
- Incorporar motores de búsqueda de tiempo real para la ubicación de información en el ámbito de estudiantes e institutos.

1.5. ALCANCES Y LÍMITES

1.5.1. Alcances

El sistema se encuentra organizado en:

- Módulo Estudiantes:
 - ✓ Registro de: kárdex personal, inscripción, calificaciones.
 - ✓ Reportes: récord académico, certificación de calificaciones, historial de estudios, kárdex de documentos presentados.
- Módulo Instituto Educativo:
 - ✓ Registro de: datos del instituto, directivos, carreras impartidas, especialidades, mallas curriculares por carrera, clasificación (público, privado, convenio)

- ✓ Reportes: estadísticas sobre estudiantes: inscritos, abandonos, reincorporación, egresados, titulados. Estos están clasificados por: sexo, rangos de edad, gestión, tipos de institutos, carreras, ubicación geográfica y otros criterios de utilidad requeridos.
- ✓ Un aspecto muy importante es la generación de los centralizadores de calificaciones de cada gestión académica para un seguimiento minucioso y efectivo de los institutos.
- Módulo Plantel Docente:
 - ✓ Registro de: kárdex personal, asignaturas que dicta, historial profesional.
 - ✓ Reportes: acefalías en cargos (para institutos públicos), evaluación de rendimiento.
- Módulo Administrativo
 - ✓ Definición de roles y privilegios para: un super usuario, técnicos de seguimiento en la DDE LP, responsables académicos en los institutos.
 - ✓ Administración de cuentas de usuario bajo los roles mencionados y políticas de seguridad inherentes al sistema.

Finalmente, cada módulo cuenta con un motor de búsqueda definido para su ámbito y con características de búsqueda inteligente mediante la utilidad de autocompletado en el campo de introducción de texto.

1.5.2. Límites

El proyecto está netamente abocado a la gestión académica por ser lo concerniente a la DDE LP. Los aspectos de gestión administrativa y económica los maneja cada institución educativa según sus propios reglamentos.

Por las funciones conferidas en el D.S. 0813 y el reglamento definido en la R.M. 01/2021 este sistema será aplicable solamente al Departamento de La Paz en el Subsistema de Educación Superior de Formación Profesional.

1.6. IMPORTANCIA DEL ESTUDIO

El software de *Gestión Académica para Institutos de Formación Superior del Departamento de La Paz*, brindará un mejor soporte operativo en primera instancia a los directos usuarios que son los responsables académicos de los institutos que gozarán un moderno sistema con interfaces basadas en actuales paradigmas en la experiencia de usuario y características utilitarias como validación de datos, autocompletado de formularios, emisión de reportes relacionados a su rol, motores de búsqueda en su correspondiente ámbito.

En una segunda instancia, los datos correctamente introducidos serán de gran utilidad para los técnicos de seguimiento en la DDE LP que contarán con motores de búsqueda para la ubicación de información precisa sobre los distintos ámbitos del sistema. Además de las operaciones habituales de generación de reportes administrativos, tendrán la facilidad de obtener indicadores educativos plasmados en resultados estadísticos bajo distintos criterios de parametrización personalizable a objeto de apoyar en la toma de decisiones.

Finalmente, con estas características el mayor beneficio será para la población estudiantil que acude a esta institución para la tramitación de las certificaciones administrativas.

CAPÍTULO II

MARCO TEÓRICO

En este capítulo se describen los principales aspectos teóricos para el desarrollo del proyecto con el desglose de las principales definiciones, características y dimensiones que fundamentan la presente investigación y que toman en cuenta conceptos puntuales para el desarrollo del trabajo planteado que se aplicarán durante su ejecución.

2.1. AUTOMATIZACIÓN DE LA GESTIÓN ACADÉMICA

2.1.1. Gestión Académica

El concepto de **gestión** abarca todo el conjunto de trámites que se llevan a cabo para resolver un asunto o concretar un proyecto en cualquier dimensión. La gestión es también el conjunto de acciones y diligencias que permiten la administración de responsabilidades en el ámbito empresarial o personal, (Fernandez, 2003).

La **gestión académica** es el conjunto de actividades relacionadas al tratamiento metódico de la información en una institución educativa en lo referido a: definición de cursos impartidos y sus niveles, pólitos de asignaturas, datos de los estudiantes, matriculación, registro de calificaciones, seguimiento de asistencia, conclusión de estudios (sin dejar de lado casos de abandono, reincorporación, trasposos) hasta la titulación definitiva. La importancia de esta información radica en la obtención de resultados estadísticos que constituyen los indicadores educativos para tener una visión global en la toma de decisiones. La gestión académica es un componente fundamental en las instituciones educativas en complemento al correspondiente proceso de enseñanza-aprendizaje.

Por otra parte, la **calidad de gestión** es un proceso continuo cuyos puntos representa combinaciones de funcionalidad, eficiencia y eficacia altamente correlacionadas a su grado máximo: la excelencia, que supone un óptimo nivel de coherencia entre todos los componentes del sistema, (Delgado, 2006). También se trata de un saber de síntesis que es capaz de ligar conocimiento y acción, ética y eficacia, política y administración de procesos que tienden al mejoramiento continuo de las prácticas administrativas: a la exploración y explotación de todas las posibilidades; y a la innovación permanente como proceso sistemático. (Fernandez, 2003).

2.1.2. Automatización de Procesos

La **automatización de procesos** es la sustitución de tareas tradicionalmente manuales por las mismas realizadas de manera automática por máquinas, robots o cualquier otro tipo de automatismo. Tiene ventajas muy evidentes en los procesos industriales: mejora en costos, servicio y calidad. Además, se producen menos problemas de calidad por realizarse el trabajo de una manera más uniforme debido a las especificaciones dadas al automatismo, (Smith, 2011).

Los principales beneficios de la automatización de procesos son:

- Controlar y dar seguimiento del proceso en todo momento de forma detallada y completa, pudiendo conocer su estatus de forma inmediata.
- Reducir el tiempo del ciclo de proceso, es decir desde que inicia hasta que termina.
- Eliminar el intercambio ampuloso de información a través de correo electrónico, documentos en papel, archivos Word, Excel, PDFs.
- Obtener indicadores de desempeño actualizados en tiempo real.
- Conocer exactamente lo ocurrido en cada paso del proceso, a través del registro o rastro que se genera de cada actividad.
- Identificar tareas redundantes o las que no dan valor para mejorar el proceso.

- Recibir mediante notificaciones automáticas alertas de lo que pasa o deja de pasar en cualquier punto del proceso.
- Facilitar la búsqueda de información que al tenerla centralizada permita conseguir resultados con el mínimo esfuerzo y coste, o que al menos no sean tan variables.

Por lo expuesto, la **automatización de la gestión académica** es el conjunto de actividades relacionadas a la gestión académica apoyadas en las herramientas administrativas y recursos tecnológicos a disposición con el fin de contar con información concreta reflejada en resultados estadísticos de indicadores educativos que permitan una acertada toma de decisiones.

2.2. CONCEPTOS SOBRE SISTEMAS

2.2.1. Sistema

Para citar una definición clásica, **sistema** es un conjunto de elementos que interactúan para realizar funciones específicas, es un todo que se encuentra integrado, compuesto de estructuras diversas, interactuantes y especializadas. Ejecuta una función imposible de realizar por una cualquiera de las partes individuales de tal forma tal que un cambio en un elemento afecta al conjunto de todos ellos. (McLeod, 2000).

2.2.2. Sistema Informático

Por otro lado, un **sistema informático** es un sistema de información que basa su procesamiento en el empleo de la computación como principal herramienta para programar, almacenar programas y datos. Abarca todo el conjunto de funciones interrelacionadas entre el hardware, software y elemento humano. (Blanco, 2008). Si además de la información, es capaz de almacenar y difundir los conocimientos que se generan sobre cierta temática, entonces está en presencia de un **sistema de gestión de información** y conocimientos. Como principal utilidad se emplea esa información en dos actividades fundamentales: la toma de decisiones y el control de procedimientos.

Los componentes de un sistema informático son:

- **Componente Físico:** Formado por los elementos la computadora, sus periféricos y la comunicación entre estos, con el objetivo de proporcionar la capacidad de realizar el proceso del sistema informático con sus respectivos cálculos.
- **Componente Lógico:** Compuesto de los programas ocupados en la realización del sistema, en el que se incluye toda la lógica de programación, para que las personas puedan interactuar con cada uno de las funcionalidades.
- **Componente Humano:** Constituido por los usuarios operadores del sistema y por otro lado por los profesionales que han participado en el desarrollo del mismo como ingenieros, programadores y analistas.

2.2.3. Sistema Web

También denominado *Aplicación Web* o *WebApp*, es una aplicación distribuida a la que el usuario puede acceder desde un navegador web por medio de distintos dispositivos. Las aplicaciones web cada vez son más populares debido al alcance que tiene para llegar al cliente final de forma ligera desde un navegador web independientemente del sistema operativo o del dispositivo. (Padilla, 2014).

Las características principales de una aplicación web son:

- **Compatibilidad multiplataforma.** Una misma versión de la aplicación puede ejecutarse sin problemas en múltiples plataformas como Windows, Linux, Mac.
- **Acceso remoto inmediato.** No necesitan ser descargadas, instaladas ni configuradas. Pueden ser accedidas a través de un navegador web y una URL desde cualquier dispositivo conectado a la red.
- **Menos requerimientos de hardware.** No consume (o consume muy poco) espacio en disco y memoria RAM en comparación con los programas instalados localmente. La mayor parte del trabajo se realiza en el servidor en donde reside la aplicación.

- Seguridad en los datos. Los datos se alojan en servidores con sistemas de almacenamiento altamente fiables y se ven libres de problemas que comúnmente sufren los computadores de usuarios comunes como virus y/o fallas de disco duro.
- Diseño responsivo. Los elementos y contenidos se redimensionan y acomodan a los tamaños de pantalla de dispositivos, permitiendo una correcta visualización y una mejor experiencia de usuario. La figura 2.1 expresa una muestra de ello.

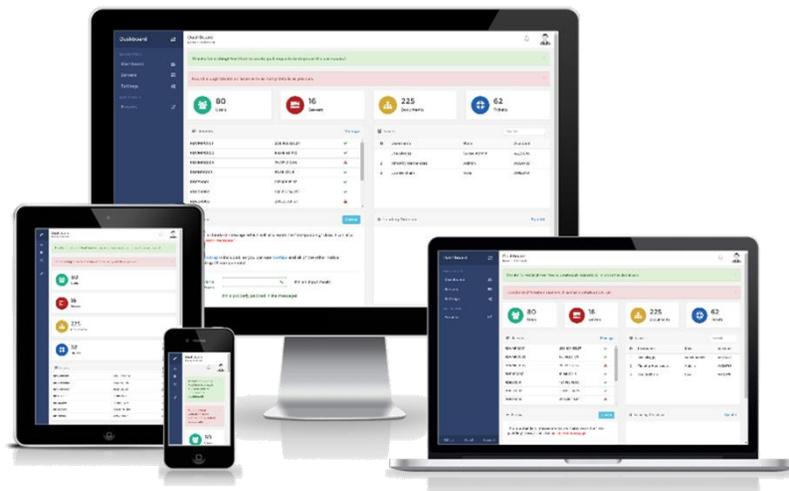


Figura 2. 1: Diseño responsivo adaptable a varios dispositivos
Fuente: <https://www.themelamp.com/best-free-bootstrap-admin-dashboard-templates/>

2.3. INGENIERÍA DE SOFTWARE

2.3.1. Definición

Las definiciones clásicas sostienen que:

Ingeniería de software es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software (Bohem, 1976).

La ingeniería de software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable, que sea fiable y trabaje en máquinas reales (Bauer, 1972).

Un enfoque y análisis más contemporáneo indica que:

La ingeniería de software es una disciplina formada por el conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de programas informáticos –software– brindando soluciones en distintos campos aplicativos. Engloba toda la gestión de un proyecto desde el análisis previo de la situación, el planteamiento del diseño hasta su implementación, pasando por pruebas recurrentes para su correcto funcionamiento. Es una de las ramas de las ciencias de la computación que estudia la creación de software confiable y de calidad, basándose en métodos y técnicas de ingeniería brindando soporte operacional. (Sommerville, 2011).

2.3.2. Ciclo de vida del Software

El desarrollo del software consta de un “ciclo de vida del software” de cuatro etapas:

- Concepción, determina la repercusión del proyecto y diseña el modelo de negocio.
- Elaboración, precisa la planificación y características en la arquitectura del proyecto.
- Construcción, es la elaboración del producto.
- Transición, es la entrega del producto terminado a los usuarios.

La creación del software es un proceso intrínsecamente creativo y la ingeniería del software trata de sistematizar este proceso con el fin de acotar el riesgo de fracaso en la consecución del objetivo, por medio de diversas técnicas que se han demostrado adecuadas sobre la base de la experiencia previa.

La ingeniería de software se puede considerar como la ingeniería aplicada al software, esto es, por medios sistematizados y con herramientas preestablecidas, la aplicación de ellos de la manera más eficiente para la obtención de resultados óptimos; objetivos que siempre busca la ingeniería. No es solo de la resolución de problemas, sino más bien teniendo en cuenta las diferentes soluciones, elegir la más apropiada.

2.3.3. Objetivos de la Ingeniería de Software

Esta disciplina cubre un marco amplio de aplicaciones debiendo entender esto como la posibilidad de plantear varios objetivos a tener en cuenta cuando se quiere implementar una solución de software:

- Diseño de programas informáticos adaptados a las necesidades y exigencias de los clientes.
- Estar presente en todas las fases del ciclo de vida de un producto.
- Contabilizar los costes de un proyecto y evaluar los tiempos de desarrollo.
- Realizar el seguimiento del presupuesto y cumplir los plazos de entrega.
- Liderar equipos de trabajo de desarrollo de software.
- Estructurar la elaboración de evidencias que comprueben el perfecto funcionamiento de los programas.
- Diseñar, construir y administrar bases de datos.
- Incluir procesos de calidad en los sistemas, calculando métricas e indicadores y chequeando la calidad del software producido.
- Estructurar e inspeccionar el trabajo del equipo ya sea el grupo de técnicos de mantenimiento o el grupo de ingenieros de sistemas y redes.

No en todos los casos una ingeniería de software debe enfocarse a todos estos objetivos ya que las empresas que contratan este servicio no siempre requieren el mismo tipo de proyecto.

2.4. METODOLOGÍAS DE DESARROLLO DE SOFTWARE

Las metodologías de desarrollo de software son un conjunto de técnicas y métodos organizativos que se aplican para diseñar soluciones de software informático. El objetivo de las distintas metodologías es el de organizar los equipos de trabajo para que estos desarrollen las funciones de un programa de la mejor manera posible.

Al desarrollar productos o soluciones para un cliente o mercado concreto, es necesario tener en cuenta factores como los costos, la planificación, la dificultad, el equipo de trabajo disponible, los lenguajes utilizados, etc. Estos factores se engloban en una metodología de desarrollo que permite organizar el trabajo de una forma ordenada que permite reducir el nivel de dificultad, organizar las tareas, agilizar el proceso y mejorar el resultado final de las aplicaciones a desarrollar. Bajo este principio, se pueden diferenciar dos grandes grupos de metodologías de desarrollo de software: tradicionales y ágiles. (Santander, 2020)

2.4.1. Metodologías tradicionales

También denominadas pesadas o clásicas. Se caracterizan por definir total y rígidamente los requisitos al inicio de los proyectos de ingeniería de software con ciclos de desarrollo poco flexibles que no permiten realizar cambios. La organización del trabajo de las metodologías tradicionales es lineal, es decir, las etapas se suceden una tras otra y no se puede empezar la siguiente sin terminar la anterior. Tampoco se puede volver hacia atrás una vez se ha cambiado de etapa y no se adaptan nada bien a los cambios. La figura 2.2 muestra un esquema general de las etapas en las metodologías tradicionales.



Figura 2. 2: Esquema general de metodologías tradicionales.

Fuente: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>

Las principales metodologías tradicionales son:

a) Cascada

Las etapas se organizan de arriba a abajo, de ahí el nombre. Se desarrollan las diferentes funciones en etapas diferenciadas y obedeciendo un riguroso orden. Antes de cada etapa se debe revisar el producto para ver si está listo para pasar a la siguiente etapa.

b) Prototipado

Se basa en la construcción de un prototipo de software que se construye rápidamente para que los usuarios puedan probarlo y aportar una retroalimentación, de este modo se puede arreglar lo que está mal e incluir otros requerimientos que puedan surgir.

c) Espiral

Es una combinación de las dos anteriores formado por cuatro etapas: planificación, análisis, desarrollo de prototipo y evaluación del cliente. Están organizadas en forma de espiral: cuanto más cerca del centro se está, más avanzado está el proyecto.

d) Incremental

El software se va construyendo de manera incremental hasta el producto final. En cada etapa se agrega una nueva funcionalidad, lo que permite ver resultados de una forma más rápida en comparación con el modelo en cascada. El software se puede empezar a utilizar incluso antes de que se complete totalmente.

e) RAD

Permite desarrollar software de alta calidad en un corto periodo de tiempo. El código puede contener más errores, y sus funciones son limitadas debido al poco tiempo del que se dispone para desarrollarlas. El objetivo es iterar el menor número posible de veces para conseguir una aplicación de forma rápida.

2.4.2. Metodologías ágiles

Estas metodologías se basan en el modelo incremental, donde en cada ciclo de desarrollo se van agregando nuevas funcionalidades a la aplicación final. Los ciclos son mucho más cortos y rápidos, por lo que se van agregando pequeñas funcionalidades en lugar de grandes cambios. Gradualmente se va construyendo y puliendo el producto final, a la vez que el cliente puede ir aportando nuevos requerimientos o correcciones, ya que puede comprobar cómo avanza el proyecto en tiempo real, tal como se describe en la figura 2.3.



Figura 2. 3: Esquema general de las metodologías ágiles

Fuente: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>

Las principales metodologías ágiles son:

a) Kanban

Metodología de trabajo inventada por la empresa de automóviles Toyota, de hecho en idioma japonés *kanban* significa “*tarjetas visuales*” y es precisamente lo que se maneja en ella. Consiste en dividir las tareas en porciones mínimas y organizarlas en un tablero de trabajo dividido en tareas pendientes, en curso y finalizadas. De esta forma, se crea un flujo de trabajo muy visual basado en tareas prioritarias e incrementando el valor del producto.

b) SCRUM

Es también una metodología incremental que divide los requisitos y tareas de forma similar a Kanban. Se itera sobre bloques de tiempos cortos y fijos (entre dos y cuatro semanas) para conseguir un resultado completo en cada iteración. Las etapas son: planificación de la iteración (planning sprint), ejecución (sprint), reunión diaria (daily meeting) y demostración de resultados (sprint review). Cada iteración por estas etapas se denomina también como sprint.

c) Lean

También conocida como *desarrollo ligero*, es una metodología configurada para que pequeños equipos de desarrollo muy capacitados elaboren cualquier tarea en poco tiempo. Los activos más importantes son las personas y su compromiso, relegando así a un segundo plano el tiempo y los costes. El aprendizaje y potenciar el equipo son fundamentales.

d) Programación Extrema

Es una metodología de desarrollo de software basada en las relaciones interpersonales, que se consideran la clave del éxito. Su principal objetivo es crear un buen ambiente de trabajo en equipo con una constante retroalimentación del cliente. El trabajo se basa en 12 conceptos: diseño sencillo, testeo, refactorización y codificación con estándares, propiedad colectiva del código, programación en parejas, integración continua, entregas semanales e integridad con el cliente, cliente in situ, entregas frecuentes y planificación.

2.4.3. Comparación de metodologías

Entre las metodologías ágiles más empleadas, la tabla 2.2 muestra una comparativa general de sus características.

Tabla 2. 1: Comparación entre las metodologías ágiles
Fuente: (Junquera, 2019)

SCRUM	KANBAN	XP
Divide el proyecto en pequeños bloques (sprints), que se planifican y revisan continuamente.	Utiliza técnicas visuales, permitiendo saber en qué punto se encuentra cada tarea de forma rápida y sencilla.	Divide el proyecto en fases y, en cada una, realiza un ciclo completo de análisis, diseño, desarrollo y pruebas.
Al finalizar un sprint, se obtiene un prototipo del producto para que el cliente pueda empezar a utilizar las características más importantes del mismo.	Los flujos de trabajo en los tableros deben contener tres elementos fundamentales: tareas pendientes, en curso y realizadas.	Los desarrolladores pueden responder rápidamente a los cambios gracias a que existe un flujo de comunicación constante.
Orientada a la administración del proyecto	Orientada a la administración del proyecto	Se centra en la propia creación del producto
No se permiten cambios en los sprints una vez que el planning meeting ha concluido.	Para garantizar un flujo de trabajo sea ágil, no se deben abrir tareas hasta que se hayan cerrado otras.	Se aceptan cambios en las iteraciones, siempre y cuando el equipo no haya empezado ya a trabajar en un requerimiento en particular.
Trata de seguir el orden de prioridades que marca el Product Owner en el Sprint Backlog pero puede cambiarlo si es mejor para el desarrollo de las tareas.	Se basa estrictamente en el cumplimiento de las tareas planteadas en el tablero de trabajo.	El equipo de desarrollo sigue estrictamente el orden de prioridad de las tareas definido por el cliente

Por la naturaleza del presente proyecto, donde el objetivo es entregar un producto y se tiene un plazo de desarrollo relativamente mediano para realizar ajustes en las iteraciones, se ha **escogido la metodología XP** para el desarrollo.

Se tendrá una planificación para la entrega del producto mediante la especificación y estimación de requerimientos entre los usuarios y cada uno de los módulos, con una constante comunicación y retroalimentación en el equipo donde se considera posibilidad de enfrentarse a cambios en donde exista un alto riesgo técnico.

2.5. METODOLOGÍA XP

Surge en el año 1993 formulada por Kent Beck, reconocido programador a nivel mundial con el concepto de crear buenas prácticas y seguirlas constantemente a lo largo del proyecto, bajo la teoría: *“Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar.”* (Beck, 1999).

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo con una permanente intercolaboración, todo ello resumido en **valores** que se enuncian a continuación, (Maida, 2015).

2.5.1. Valores XP

a) Comunicación

Debe ser fluida entre todos los participantes en el proyecto. El entorno tiene que favorecer la comunicación espontánea, ubicando a todos los miembros en un mismo lugar. La comunicación directa nos da mucho más valor que la escrita: podemos observar los gestos del cliente, o la expresión de cansancio de nuestro compañero.

b) Simplicidad

Cuanto más sencilla sea la solución, más fácilmente podremos adaptarla a los cambios. Las complejidades aumentan el coste del cambio y disminuyen la calidad del software, las funcionalidades extras se añadirán más tarde. Sólo se utiliza lo que en ese momento nos da valor, y lo haremos de la forma más sencilla posible.

c) Retroalimentación

El usuario debe utilizar y probar el software desarrollado desde la primera entrega, dando sus impresiones y necesidades no satisfechas, de manera que estas cosas encontradas vuelvan a formar parte de los requisitos del sistema y así poder atacarlas con tiempo.

d) Coraje

Con XP se tocan continuamente cosas que ya funcionan, para mejorarlas o agregar funcionalidad. Es por eso que el coraje es parte del proyecto también, ya que el miedo a tocar o modificar cosas que ya funcionan perfectamente, a veces puede ser difícil.

e) Respeto

Los programadores no pueden realizar cambios que hacen que las pruebas existentes fallen o que demore el trabajo de sus compañeros. Los miembros del equipo respetan el trabajo del resto no haciendo menos a otros, una mejor autoestima en el equipo eleva su ritmo de producción.

2.5.2. Roles de los integrantes

Cada rol tiene unas funciones claras dentro de la metodología. Cada persona del equipo puede ejecutar uno o varios roles, o incluso cambiar de rol durante las diferentes fases del proyecto. (Maida, 2015)

a) Cliente

Escribe las historias de usuario y pruebas funcionales para validar su implementación. Asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.

b) Programador

Produce el código del programa. Estima el tiempo de desarrollo en cada historia de usuario y delimita duraciones. Escribe las pruebas unitarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.

c) Encargado de pruebas (tester)

Ayuda al cliente a escribir las pruebas funcionales de aceptación. Ejecuta estas pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

d) Encargado de seguimiento (tracker)

Proporciona la retroalimentación al equipo. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos.

e) Líder técnico (coach)

Es responsable del proceso global. Debe conocer a fondo la metodología para proveer guías a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

f) Consultor

Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Tiene conocimientos concretos de su área de especialidad. Guía al equipo para resolver un problema específico.

g) Administrador del Proyecto (Project Manager)

Es el máximo responsable del proyecto. Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

2.5.3. Etapas del proceso XP

Esta metodología engloba un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades estructurales: planeación, diseño, codificación y pruebas. La figura 2.4 ilustra el proceso XP y resalta algunas de las ideas y tareas clave que se asocian con cada actividad estructural. (Pressman, 2004)

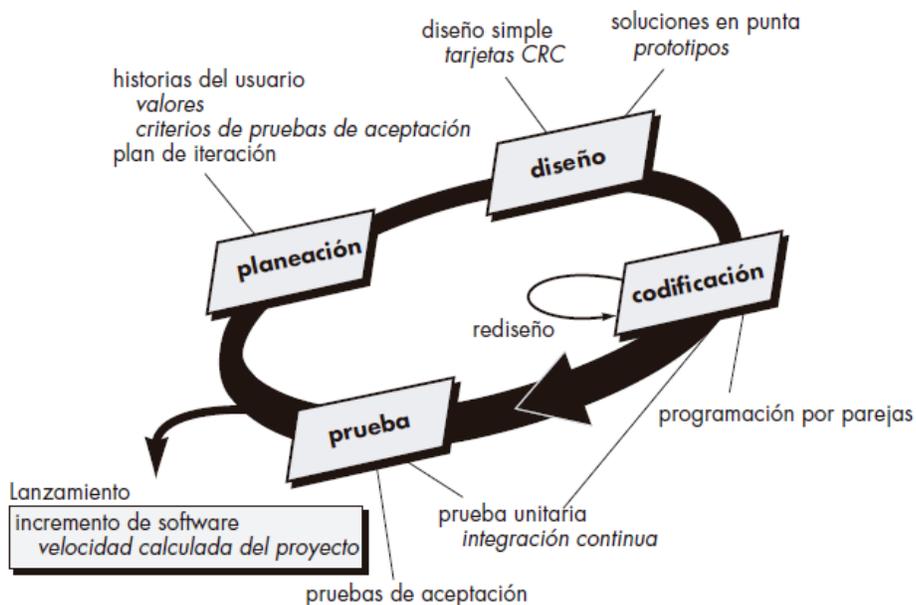


Figura 2. 4: Proceso de la programación extrema
Fuente: (Pressman, 2004)

A continuación, se resumen las actividades que se siguen en la metodología.

a) Planeación

Inicialmente se identifican las **historias de usuario** que son tarjetas donde se detallan funcionalidades específicas del software a desarrollar, describen en no más de 3 líneas necesidades del cliente y cada función o historia se clasifica según una prioridad.

Seguidamente se crea un **plan de entregas**, donde se indiquen las historias de usuario que se crearán para cada versión del programa y las fechas en las que se publicarán.

Se define un **plan de iteraciones** para que desarrolladores y clientes establezcan los tiempos de implementación ideales de las historias de usuario.

b) Diseño

Se deben lograr **diseños simples** y sencillos, procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible e implementable que a la larga costará menos tiempo y esfuerzo desarrollar.

Para un diseño de software orientado a objetos, se crearán **tarjetas CRC** que describen las clases utilizadas en la programación de una historia y que son relevantes para el incremento actual de software.

c) Codificación

En XP la programación se trabaja **en parejas** en frente del mismo computador, incluso se pueden intercambiar las parejas. De esta forma, se asegura realizar un código más universal bajo el principio de *codificación para todos*, con el que cualquier otro programador podría trabajar y entender lo desarrollado.

d) Prueba

Se inician las **pruebas unitarias** que aíslan partes del código (método, procedimiento o función) para comprobar su correcto funcionamiento.

Las **pruebas de integración** implican probar los diferentes módulos del software como grupo con el propósito de validar la interacción entre ellos y corregir las posibles fallas.

Las **pruebas de aceptación**, son especificadas por el cliente y se centran en las características y funcionalidad generales del sistema que son visibles y revisables por el mismo. Estas pruebas se derivan de las historias de usuario planteadas al inicio del proyecto.

Estas dos últimas etapas se ejecutan en varias iteraciones de no más de tres semanas, aconsejablemente, con lo que se conseguirá una entrega incremental de software. Los elementos que se toman en cuenta en cada iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas y tareas no terminadas en la iteración anterior. De este modo, al final de la última iteración el sistema estará listo para entrar en producción y su posterior lanzamiento. (Letieler, 2005)

e) Lanzamiento

En este punto se han probado todas las historias de usuario con éxito donde que han superado las pruebas determinadas por el tester el resto del equipo. Adicionalmente, se complementan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Para culminar, se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. (Mancuzo, 2020)

2.5.4. Herramientas en XP

a) Historias de usuario

Tal como las define Mike Cohn, las historias de usuario son: “descripciones breves y simples de una característica contada desde la perspectiva de la persona que desea la nueva capacidad, generalmente un usuario o cliente del sistema.” (Cohn, 2004)

Las historias de usuario se escriben en tarjetas para poder pegarlas sobre algún tipo de tablero y facilitar el debate sobre cómo organizarlas. Existen varias plantillas sugeridas por distintos autores, la figura 2.5 describe un esquema general y los componentes que lo conforman.

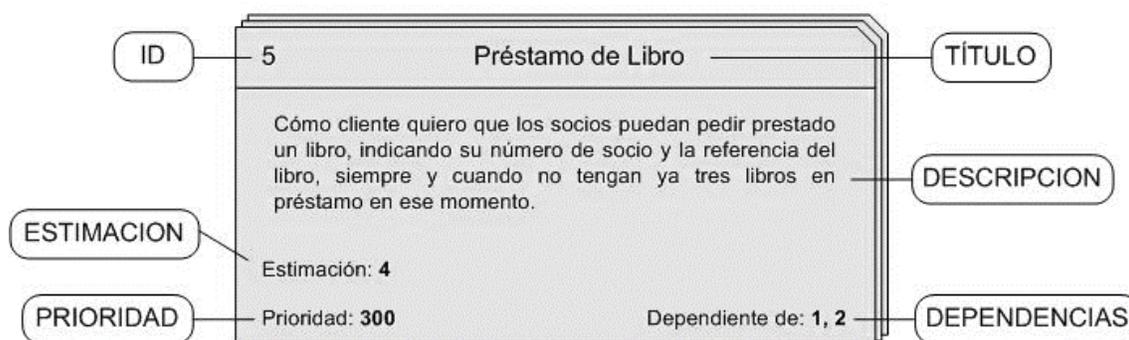


Figura 2. 5: Esquema y componentes de una Historia de Usuario
Fuente: (Vergara, 2021)

En muchos casos sólo se propone utilizar un nombre y una **descripción**, para esta última se sugiere plantearla bajo patrón definido en la tabla 2.3.

Tabla 2. 2: Patrón para el campo Descripción en una Historia de Usuario
Fuente: (García, 2019)

Patrón	Ejemplo
Como <Rol, persona> Persona o rol de usuario que tiene la necesidad	Como usuario de la aplicación móvil del banco
Quiero <objetivo, comportamiento> Lo que se quiere obtener: una característica, funcionalidad, etc,	Quiero visualizar mi posición global en la página de entrada
Para <motivo, razón, valor> Motivo por el que se necesita, valor que se obtiene como resultado	Para saber si dispongo de saldo suficiente nada más acceder sin navegar por menús

b) Tarjetas CRC

Las tarjetas CRC (Clase-Responsabilidad-Colaboración) son una herramienta usada como metodología para el diseño de software orientado a objetos, creada por Kent Beck (Beck, 1999) y Ward Cunningham. La técnica consiste en dibujar una tarjeta por cada clase u objeto, y dividirla en tres zonas. Las tablas 2.4 y 2.5 muestran el patrón a seguir junto a un ejemplo aplicado.

Tabla 2. 3: Patrón de una tarjeta CRC
Fuente: (Viezca, 2011)

Nombre de la clase	
Responsabilidades	Colaboradores

Tabla 2. 4: Ejemplo aplicado al patrón
Fuente: (Viezca, 2011)

CLASE: Ventas	
Mostrar Producto Calcular Precio Ingresar cantidad Calcular impuesto Calcular total a pagar	Producto

2.6. METODOLOGÍA UWE

UWE (UML-based Web Engineering) es “una metodología para el desarrollo de aplicaciones web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen un proceso basado en las técnicas, notación y mecanismos de extensión de UML”, (Rossi & Pastor, 2008). Esta metodología comprende la interrelación entre modelos componentes como describe la figura 2.6.

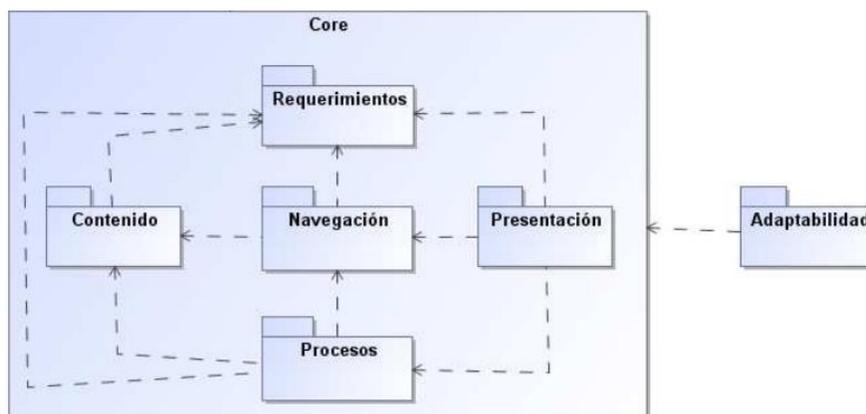


Figura 2. 6: Conjunto de modelos componentes de la metodología UWE
Fuente: (Guerrero & Ucán, 2014)

a) Modelo de requerimientos

La identificación de requisitos es el primer paso en el desarrollo de un sistema web, que involucra el modelado de casos de uso con UML. UWE propone una descripción detallada con las responsabilidades y acciones de las partes interesadas, (Koch, 2000).

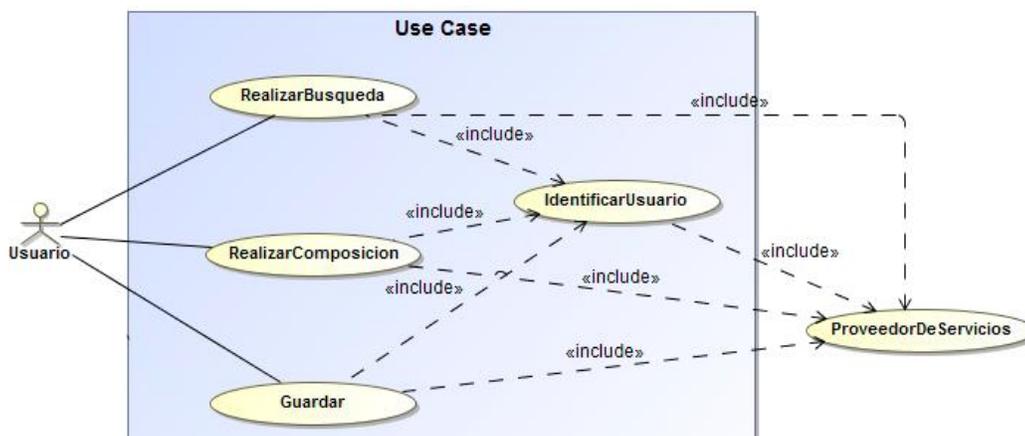


Figura 2. 7: Ejemplo de un diagrama especificando al actor y sus casos de uso
Fuente: (Guerrero & Ucán, 2014)

b) Modelo de Contenido

Proporciona una especificación visual de la información en el dominio relevante para la aplicación Web. Este es un diagrama UML normal de clases, por ello se debe pensar en las clases que son necesarias para el caso de estudio presentado. (Koch, 2000)

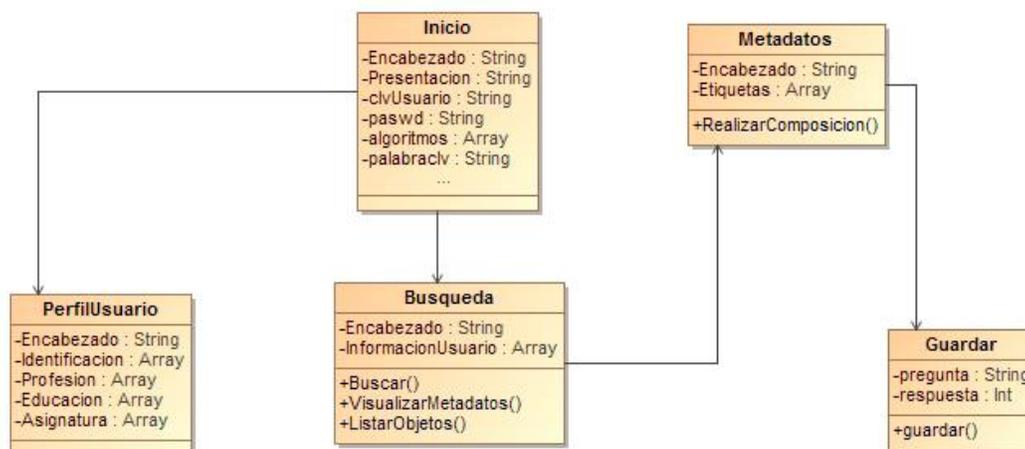


Figura 2. 8: Ejemplo de diagrama de clases
Fuente: (Guerrero & Ucán, 2014)

c) Modelo de navegación

Define cómo están enlazadas las páginas a través de un diagrama de navegación con la especificación de con nodos y enlaces estereotipados. UWE provee diferentes estereotipos para el modelado de navegación. (Koch, 2000).

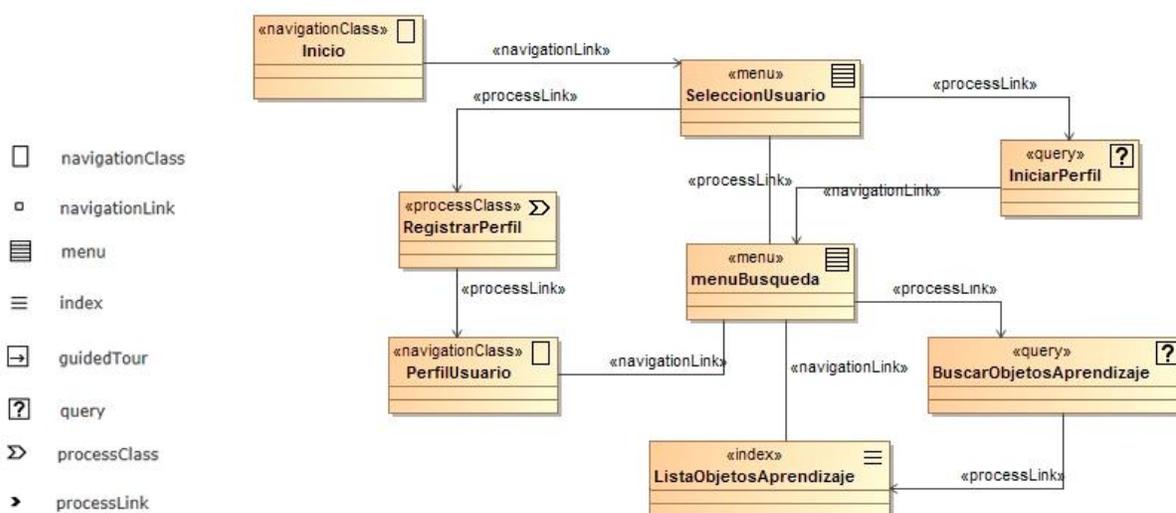


Figura 2. 9: Estereotipos UWE aplicados a la estructura de navegación
Fuente: (Guerrero & Ucán, 2014)

d) Modelo de Presentación

Ofrece una visión abstracta de la interfaz de usuario en la aplicación Web. Se basa en el modelo de navegación y en los aspectos concretos de la interfaz de usuario (IU). Describe su estructura básica: ¿qué elementos de la IU (texto, imágenes, enlaces, formularios) se utilizan para presentar los nodos de navegación? (Koch, 2000)

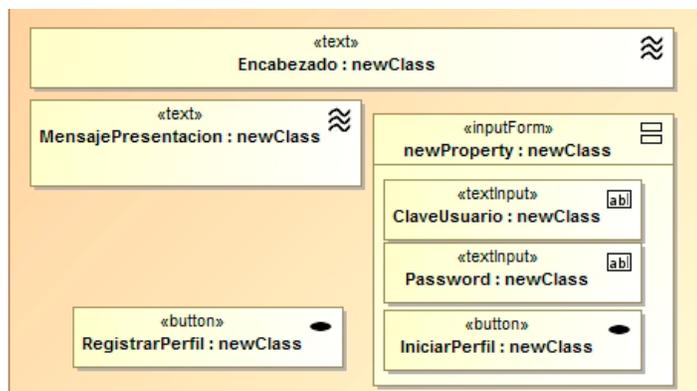


Figura 2. 10: Modelo de presentación de una página de inicio
Fuente: (Guerrero & Ucán, 2014)

e) Modelo de procesos

Describe las entradas y salidas del proceso de negocio, representa el aspecto que tienen las acciones de las clases de proceso. Se manejan dos tipos de modelos:

- Modelo de estructura del proceso: relaciones entre las diferentes clases de proceso.
- Modelo de flujo del proceso: las actividades conectadas con cada *processClass*.

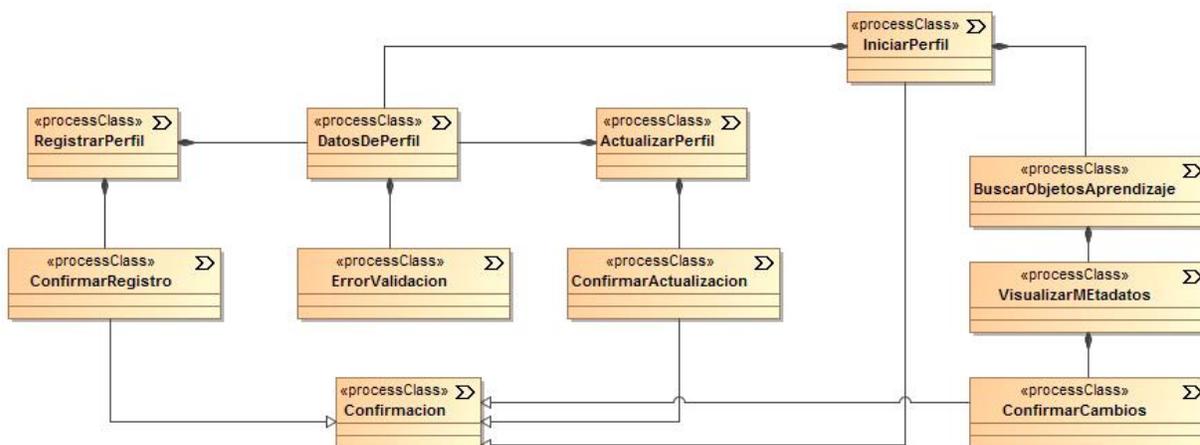


Figura 2. 11: Modelo de estructura del proceso
Fuente: (Guerrero & Ucán, 2014)

2.7. CALIDAD DE SOFTWARE

Según la Organización Internacional de Normalización y la Comisión Electrónica Internacional, (ISO/IEC 9126-1, 2001) la 9126 permite especificar y evaluar la calidad de software desde diferentes criterios, establece un modelo de calidad. Este estándar propone un modelo de calidad que se divide en tres vistas: interior, exterior y en uso.

De acuerdo a la definición en el diccionario del Instituto de Ingenieros Eléctricos y Electrónicos “La calidad de software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”. (IEEE, 1991)

Roger Pressman lo define como “Proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan”, (Pressman, 2004).

El modelo de calidad establece seis atributos importantes:

a) Funcionalidad.

Grado en el que el software satisface las necesidades planteadas según las establecen los siguientes atributos: adaptabilidad, exactitud, interoperabilidad, cumplimiento y seguridad.

b) Usabilidad.

Grado en el que el software es fácil de usar, según lo indican los siguientes subatributos: entendible, aprendible y operable.

c) Confiabilidad.

Cantidad de tiempo que el software se encuentra disponible para su uso, según lo indican los siguientes atributos: madurez, tolerancia a fallas y recuperación.

d) Eficiencia.

Grado en el que el software emplea óptimamente los recursos del sistema, según lo indican los subatributos siguientes: comportamiento del tiempo y de los recursos.

e) Facilidad de recibir mantenimiento.

Facilidad con la que pueden efectuarse reparaciones al software, según lo indican los atributos: analizable, cambiable, estable, susceptible de someterse a pruebas.

f) Portabilidad.

Facilidad con la que el software puede llevarse de un ambiente a otro según lo indican los siguientes atributos: adaptable, instalable, conformidad y sustituible.

2.8. HERRAMIENTAS

Al ser una institución pública, la Ley N° 164 de 2011 y ratificada con el D.S. N° 3251 de 2017 incentiva y reglamenta la implementación de software libre en todos los niveles del Estado. La tabla 2.5 muestra los tipos de herramientas empleadas para el desarrollo del proyecto, junto a la respectiva especificación del nombre y versión de cada una, así como una breve justificación de su uso.

Tabla 2. 5: Herramientas a emplearse en el desarrollo del proyecto

Tipo	Herramienta	Versión
Servidor Web	Apache	2.4
Motor de Base de Datos	MySQL	10.3
Lenguaje de Programación	PHP	7.3.7
Framework Back-End	Laravel	8
Framework Front-End	VUE	3.0.1
Diseño Responsivo	TailWind	1.0
IDE	Visual Studio Code	1.61.1

CAPÍTULO III

MARCO APLICATIVO

A continuación, se exponen los pasos seguidos para el desarrollo del software de gestión académica basado en la metodología XP donde se inicia con la clasificación de los requerimientos iniciales seguido de las fases de planificación, diseño y desarrollo en ciclos iterativos hasta la obtención del producto final. Se tomará como referencia las historias de usuario que representan las necesidades que deben cubrir las funcionalidades de la aplicación, y de este modo satisfacer las exigencias del sistema.

3.1. EXPLORACIÓN INICIAL

En una primera entrevista con el administrador del sistema actual, se ha dado a conocer que el mismo fue implementado de manera experimental en la gestión 2013 bajo las siguientes circunstancias:

- La Subdirección de Educación Superior tiene como una de sus responsabilidades emitir certificaciones oficiales de: estudios, calificaciones, legalizaciones y otros, como parte de los requisitos que los estudiantes deben presentar para proseguir sus trámites de carácter académico ante el Ministerio de Educación u otras instancias que así lo requieran.
- Los institutos de Formación Superior son responsables de entregar los Centralizadores de Calificaciones a la Subdirección de Educación Superior de acuerdo a sus respectivas mallas curriculares y con las debidas observaciones sobre estudiantes que abandonaron estudios, calificaciones convalidadas, incompletas y reprobadas.

- Esta información es entregada periódicamente según cronograma de forma física en libros empastados bajo formato reglamentado y con los respectivos respaldos digitales en planillas Excel.
- Con el tiempo, los archivos de Excel fueron incrementado de volumen rápidamente haciendo dificultosa la búsqueda para la emisión de certificaciones. Hacía falta una centralización general bajo un sistema informático donde se llegó al extremo de que personal de cada instituto se hacía presente en la DDELP para coadyuvar en la elaboración de las mismas.
- En la gestión 2012 se implementó un sistema web para otra área de la DDELP donde se vio una aceptación muy favorable por parte de los usuarios.
- Durante la gestión 2013 --dada la necesidad y premura en el cumplimiento de objetivos institucionales-- se analizó la posibilidad de adaptar el mismo para el tema de registro de calificaciones de institutos y se llegó a implementarlo con una etapa de pruebas satisfactorias con lo que quedó establecido su uso principalmente por la característica de poder accederlo vía web y denominándolo genéricamente como “Sistema de Institutos”.

El *Sistema de Institutos* es una adaptación experimental de otro sistema en la DDELP que no fue concebido específicamente para los requerimientos de la Subdirección de Educación Superior, por lo cual no ha recibido mantenimiento y tampoco se cuenta con la documentación de desarrollo.

Si bien la información está debidamente almacenada, han surgido nuevos requerimientos para sacar provecho de la Base de Datos con características como la emisión de reportes estadísticos de indicadores educativos y la incorporación de motores de búsqueda inteligentes. Adicionalmente es necesario replantear las interfaces de usuario para fines de uso práctico. La figura 3.1 muestra el entorno de usuario de este sistema.

Copyright © 2013 Naji.org | Sitio potenciado con software libre

Figura 3. 1: Interfaz de usuario del Sistema de Institutos
Fuente: ddelapaz.gob.bo/institutos

El Sistema de Institutos cuenta con las siguientes características:

- Registro de datos del instituto
 - ✓ Datos generales, personal docente y administrativo, carreras impartidas, definición de asignaturas.
 - ✓ Reportes: listado de institutos, Kardex de un instituto.
- Registro de datos del estudiante
 - ✓ Datos generales del estudiante, matriculación en carreras.
 - ✓ Reportes: kardex del estudiante, historial académico, certificado de calificaciones.
- Registro de calificaciones
 - ✓ Definidos el instituto, carrera cursada, asignaturas y datos del estudiante, se procede al registro de calificaciones obtenidas.
 - ✓ Reportes: centralizador de calificaciones por gestión académica.

Los actores en el sistema son:

- Técnico de seguimiento. Tiene el rol superior de administrador del sistema
 - ✓ Registra los datos generales de cada instituto
 - ✓ Administra las cuentas de usuario del responsable académico de cada instituto.
 - ✓ Administra las cuentas de usuario de los auxiliares de seguimiento.
 - ✓ Posee permisos de modificación de datos restringidos.
 - ✓ Tiene acceso a todos los módulos del sistema
 - ✓ Puede emitir todos los reportes habidos.
- Responsable académico. Es una persona por cada instituto con acceso vía web. Una vez activada su cuenta de usuario tiene las responsabilidades de:
 - ✓ Registrar los datos de cada estudiante
 - ✓ Definir las carreras que se imparten
 - ✓ Definir las asignaturas por carrera
 - ✓ Realizar la inscripción de estudiantes
 - ✓ Registrar las calificaciones obtenidas por el estudiante
- Auxiliar de Seguimiento. Son los colaboradores inmediatos del técnico de seguimiento, se desempeñan dentro de las instalaciones de la DDE. Mediante solicitudes recibidas administrativamente tienen la función de extraer reportes de:
 - ✓ Certificado de calificaciones de un estudiante al egreso de la carrera cursada.
 - ✓ Centralizador de calificaciones de un instituto organizado por carreras y gestiones.
 - ✓ Historial de reportes emitidos, tanto de certificados individuales como de los centralizadores.
 - ✓ Existe la posibilidad que el estudiante haya cursado más de una carrera, como también que se haya matriculado en más de un instituto.

Finalmente, la figura 3.2 muestra una visión general del funcionamiento actual del Sistema de Institutos expresado en un diagrama de casos de uso definido en el estándar UML.

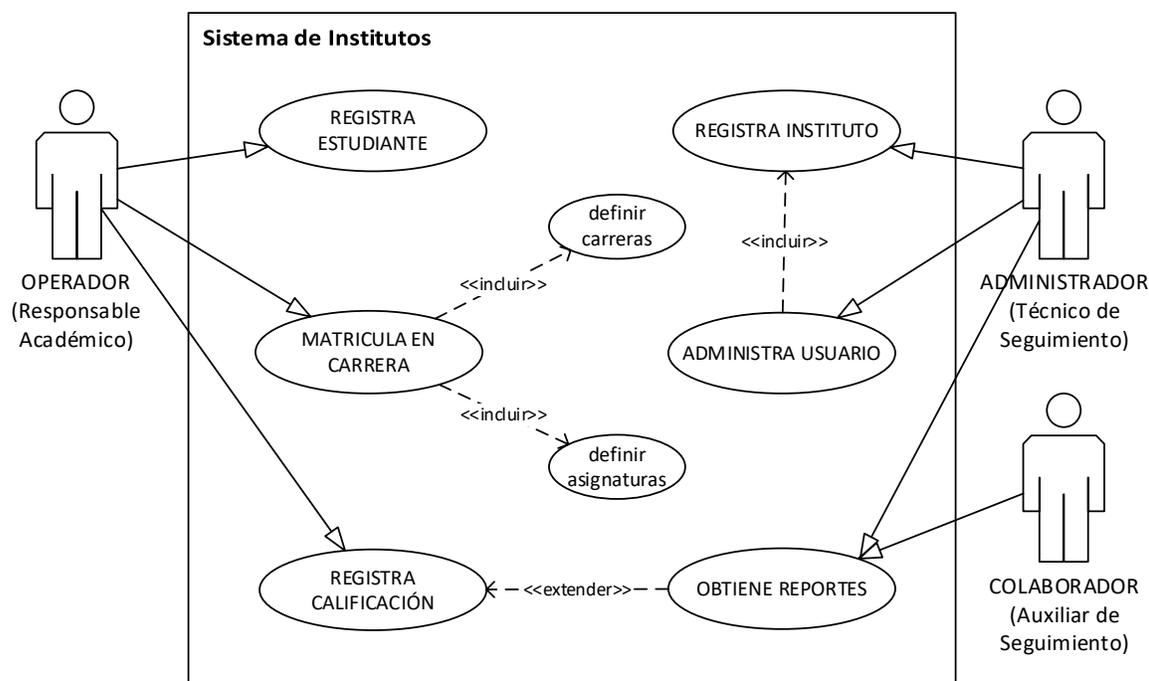


Figura 3. 2: Diagrama de casos de uso general

3.2. APLICACIÓN DE LA METODOLOGÍA

Se empleará la metodología XP (Extreme Programming) para el desarrollo conceptual del proyecto. Adicionalmente, en la etapa de diseño se tendrá como fundamento los lineamientos establecidos en la metodología UWE (UML Web-based Engineering) para el desarrollo de aplicaciones web. Existirá una planificación con tiempos estimados para la entrega del producto con una definición de requerimientos entre los usuarios y los módulos del sistema. Se mantendrá una comunicación fluida entre el equipo de trabajo para mostrar resultados tangibles.

3.2.1. Planeación

Posterior a una primera reunión de exploración con el Técnico de Seguimiento (actual administrador del sistema), se han definido los requerimientos específicos expresados en un conjunto de historias de usuario descritos a continuación.

Historia de Usuario	Número: 01
Título de la Historia: Creación de cuentas	Usuario: Administrador
Descripción: Crear cuentas de usuario en dos categorías: un responsable académico por cada instituto (operador) y varios auxiliares de seguimiento de trámites (colaboradores).	
Validación: Privilegios CRUD. Restitución de contraseñas. Activar/desactivar cuentas.	

Historia de Usuario	Número: 02
Título de la Historia: Registro de institutos	Usuario: Administrador
Descripción: Registrar los datos principales de un instituto: nombre, números de Resolución Ministerial, tipo de servicio: público, privado, convenio. Los datos complementarios serán especificados por los operadores de cada instituto una vez activadas sus cuentas.	
Validación: Privilegios CRUD. Reporte de institutos por tipo de servicio, ubicación geográfica.	

Historia de Usuario	Número: 03
Título de la Historia: Datos complementarios instituto	Usuario: Operador
Descripción: Registrar los datos complementarios del instituto: nombre de la máxima autoridad (Rector/Director), responsable administrativo (Subdirector/Gerente), responsable académico (Subdirector/Coordinador), dirección, teléfonos y otros datos de contacto.	
Validación: Editar datos, sin privilegios de crear o borrar. Reporte Kárdex del Instituto.	

Historia de Usuario	Número: 04
Título de la Historia: Creación de carreras	Usuario: Operador
Descripción: Registrar las carreras que se imparten. Especificar nivel de formación técnico medio o superior. Especificar periodicidad semestral/anual.	
Validación: Privilegios CRUD. Reporte de carreras.	

Historia de Usuario	Número: 05
Título de la Historia: Definición de asignaturas	Usuario: Operador
Descripción: Registrar asignaturas impartidas por cada carrera. Definir codificación reglamentada por la DDE. Especificar nivel y secuencia.	
Validación: Privilegios CRUD. Reporte de asignaturas.	

Historia de Usuario	Número: 06
Título de la Historia: Registro de docentes	Usuario: Operador
Descripción: Registrar datos generales del docente, profesión, especialidades.	
Validación: Privilegios CRU. NO se podrá eliminar un registro ya creado puesto que el docente podría dictar en otro instituto. Motor de búsqueda. Reportes: kárdex personal, listado de docente vigentes y no vigentes. Histórico de asignaturas dictadas.	

Historia de Usuario	Número: 07
Título de la Historia: Registro de estudiantes	Usuario: Operador
Descripción: Registrar datos generales del estudiante.	
Validación: Privilegios CRU. NO se podrá eliminar un registro ya creado puesto que el estudiante podría continuar estudios en otro instituto. Motor de Búsqueda.	

Historia de Usuario	Número: 08
Título de la Historia: Matriculación	Usuario: Operador
Descripción: Inscribir estudiantes a las carreras y asignaturas definidas previamente bajo un cronograma de inicios de curso. En este punto se define la apertura de gestiones académicas.	
Validación: Privilegios CRUD. Reportes: estudiantes por carrera, por gestión, por nivel de estudios.	

Historia de Usuario	Número: 09
Título de la Historia: Registro de calificaciones	Usuario: Operador
<p>Descripción: Registrar las calificaciones por asignatura, por gestión y por nivel de estudios cursados por los estudiantes.</p>	
<p>Validación: Privilegios limitados: se podrá modificar una calificación solamente una vez, NO se podrá eliminar calificaciones. Reportes: récord académico por estudiante, indicadores de efectividad (estudios concluidos, abandonos, traspasos), centralizador de calificaciones del instituto bajo formato y criterios definidos por la DDE.</p>	

Historia de Usuario	Número: 10
Título de la Historia: Reportes estadísticos	Usuario: Colaborador
<p>Descripción: Obtener información estadística general acerca de estudiantes: inscritos, abandonos, reincorporación, egresados, titulados. Deberán estar clasificados por: sexo, rangos de edad, gestión y carrera cursada, tipos de institutos, ubicación geográfica. Estos resultados constituirán la mayor utilidad del sistema pues se interpretarán como indicadores educativos.</p>	
<p>Validación: Junto al usuario Administrador, tendrá privilegio de generar adicionalmente todos los reportes de emitidos por los operadores.</p>	

Historia de Usuario	Número: 10
Título de la Historia: Motores de búsqueda	Usuario: Administrador
<p>Descripción: Acceder a motores que realicen búsquedas por separado en los ámbitos de institutos, estudiantes, docentes, carreras y asignaturas. Los resultados de búsqueda serán enlaces para mostrar información detallada del elemento seleccionado.</p>	
<p>Validación: Deberá existir la característica de texto autocompletable. El usuario Colaborador podrá emplear estos motores de búsqueda.</p>	

Con estos parámetros iniciales definidos en base al sistema actual, la tabla 3.1 centraliza las historias de usuario para plantear un tiempo estimado de desarrollo.

Tabla 3. 1: Resumen de Historias de Usuario

Nro	Nombre de Historia	Estimado	Iteración
01	Creación de cuentas	3 días	1
02	Registro de institutos	5 días	1
03	Datos complementarios instituto	3 días	1
04	Creación de carreras	3 días	2
05	Definición de asignaturas	3 días	2
06	Registro de docentes	3 días	2
07	Registro de estudiantes	10 días	3
08	Matriculación	10 días	3
09	Registro de calificaciones	5 días	3
10	Reportes estadísticos	10 días	4
11	Motores de búsqueda	10 días	5

3.2.2. Diseño

Se trabajará la manera más versátil posible, haciendo lo mínimo imprescindible para que cada proceso planteado cumpla su objetivo. Las funcionalidades extras se añadirán en cada iteración con lo que se logrará un incremento de software.

Para el diseño orientado a objetos, se han creado tarjetas CRC detalladas como sigue.

TARJETA CRC		NRO 1
Clase: ADMINISTRADOR		
Responsabilidades: <ul style="list-style-type: none"> ✓ CRUD de cuentas de usuario. ✓ Definición y asignación de roles. ✓ Reportes de cuentas de usuario. ✓ Creación de institutos (solamente datos principales). ✓ Reportes macro de institutos: tipos, geolocalización, carreras, cantidad de estudiantes. ✓ 	Colaboradores: <ul style="list-style-type: none"> ✓ Operador ✓ Colaborador de seguimiento 	

TARJETA CRC		NRO 2
Clase: OPERADOR		
Responsabilidades: <ul style="list-style-type: none"> ✓ Especificar los datos complementarios del instituto. ✓ Crear las carreras que se impartirán. ✓ Definir la malla curricular: asignaturas por carrera, niveles y secuencia. ✓ Registrar correctamente los datos personales de los estudiantes. ✓ Registrar los datos del plantel docente. ✓ Administrar el proceso de matriculación ✓ Definir apertura y cierre de gestión académica. ✓ Registrar las calificaciones de los estudiantes por cada gestión. 	Colaboradores: <ul style="list-style-type: none"> ✓ Administrador ✓ Colaborador de seguimiento 	

TARJETA CRC		NRO 3
Clase: COLABORADOR DE SEGUIMIENTO		
Responsabilidades: <ul style="list-style-type: none"> ✓ Realizar el seguimiento sobre la correcta introducción de datos por parte de los operadores. ✓ Verificar la consistencia de la información. ✓ Emitir reportes especializados ✓ Generar los centralizadores de calificaciones por gestión. ✓ Generar reportes de certificación de estudios y de calificaciones para su legalización y validez. ✓ Correcciones excepcionales de calificaciones u otros datos previo procedimiento y autorización administrativa. 	Colaboradores: <ul style="list-style-type: none"> ✓ Administrador ✓ Operador 	

En esta etapa se realiza la construcción con la aplicación de las herramientas UWE constituidas por las siguientes fases:

a) Especificación de Requerimientos

Esta fase se apoya en un Diagrama de Caso de Uso que describe el comportamiento de las acciones de cada actor, las funcionalidades del sistema y la interacción entre ellos. Con las tarjetas CRC definidas, se ha modelado el comportamiento del sistema según lo descrito en el diagrama de casos de uso de la figura 3.3

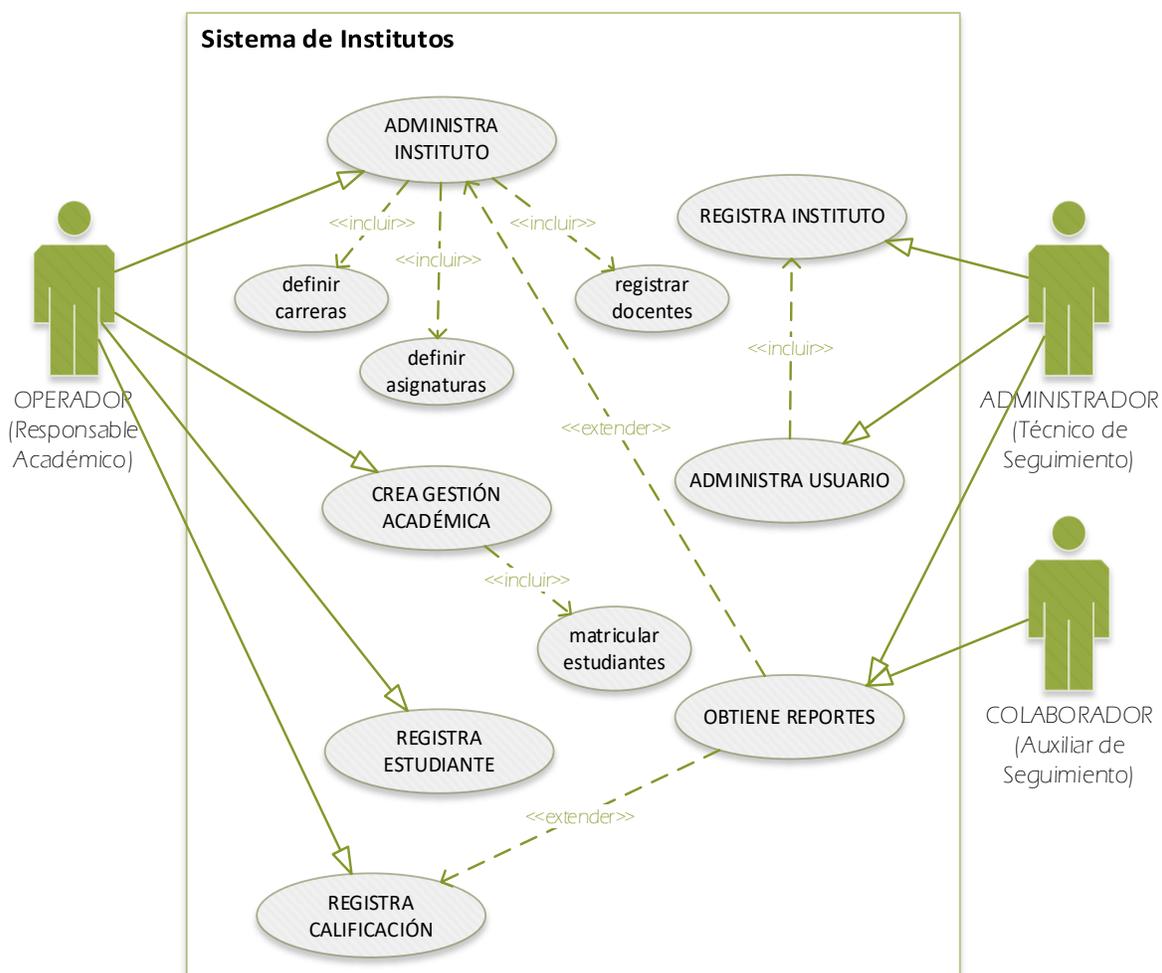


Figura 3. 3: Diagrama de casos de uso para el Software de Gestión Académica

b) Definición de Contenido

Esta etapa muestra el diseño conceptual de la información que se manejará en el sistema y los componentes que se encargaran del funcionamiento y la relación entre ellos apoyado en las herramientas que se describen.

DIAGRAMA DE CLASES. La figura 3.4 expresa la relación entre las clases de objetos, especificando atributos y operaciones junto a sus asociaciones dentro del sistema.

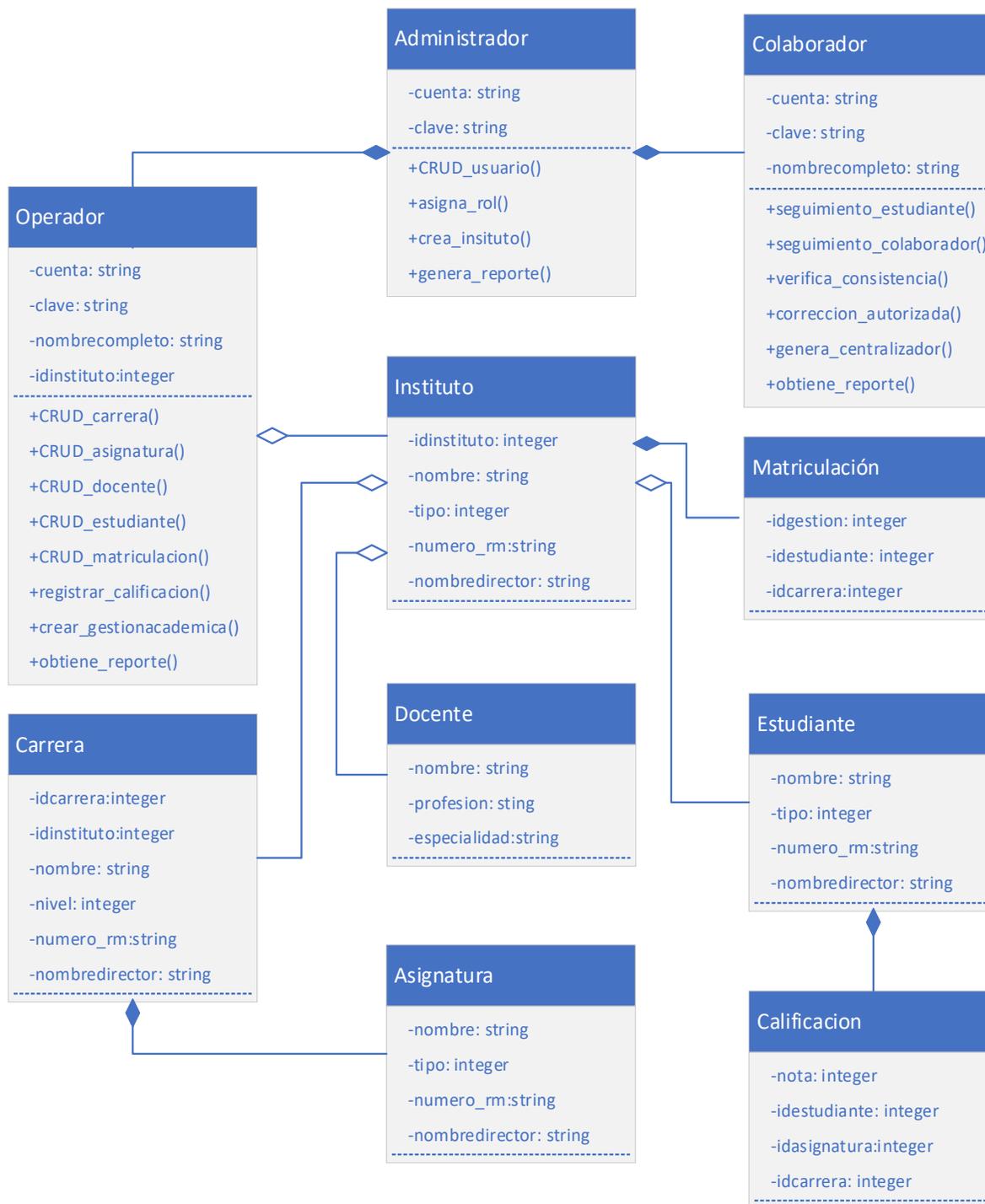


Figura 3. 4: Diagrama de clases para el Software de Gestión Académica

MODELO RELACIONAL. Para la gestión de la Base de Datos, la figura 3.5 muestra el modelado de la relación entre las tablas, especificación de atributos y cardinalidad.

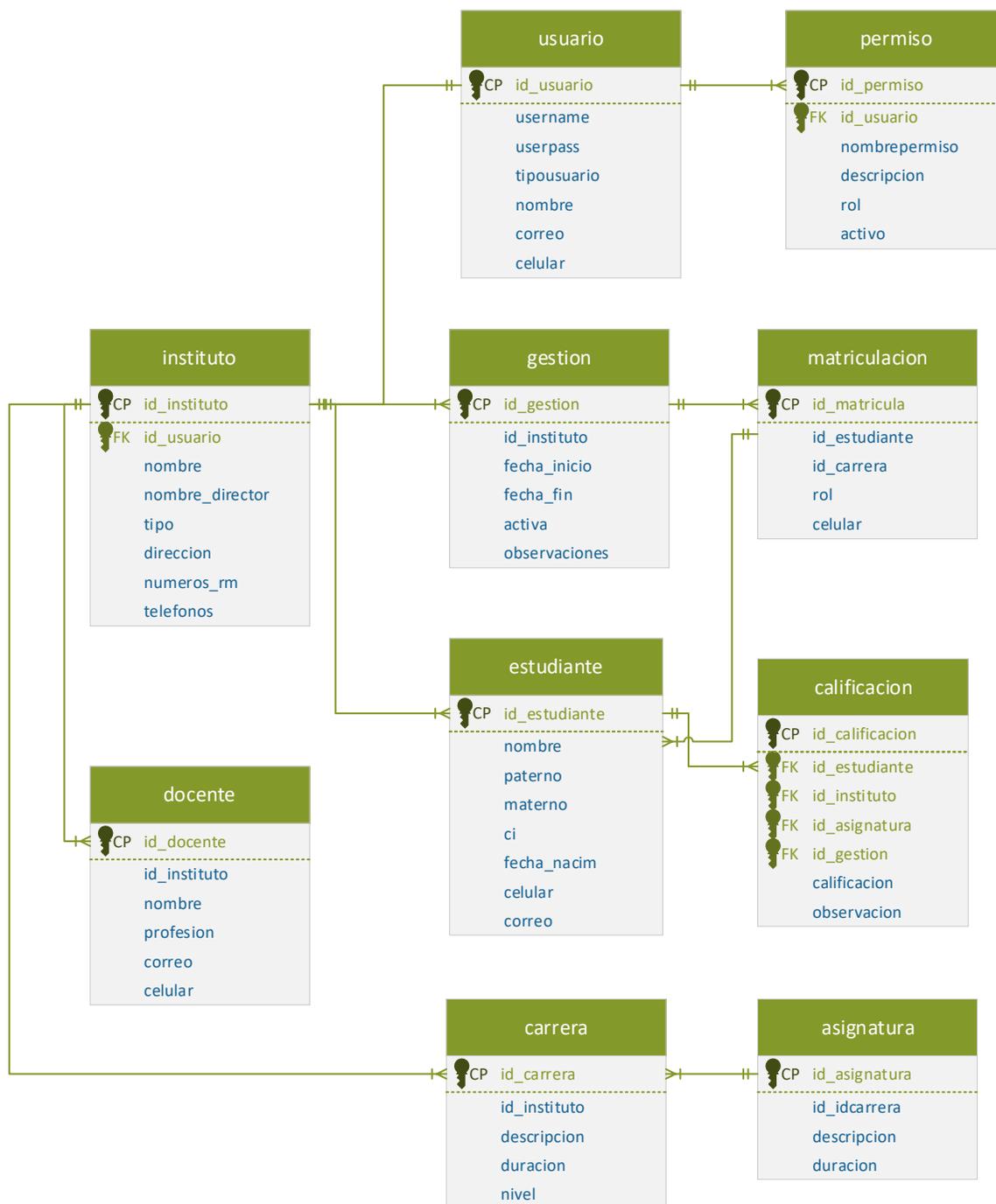


Figura 3. 5: Modelo Relacional para el Software de Gestión Académica

MODELO ENTIDAD RELACIÓN. Facilita la representación de entidades y sus relaciones. La figura 3.6 muestra el modelado propuesto para este proyecto.

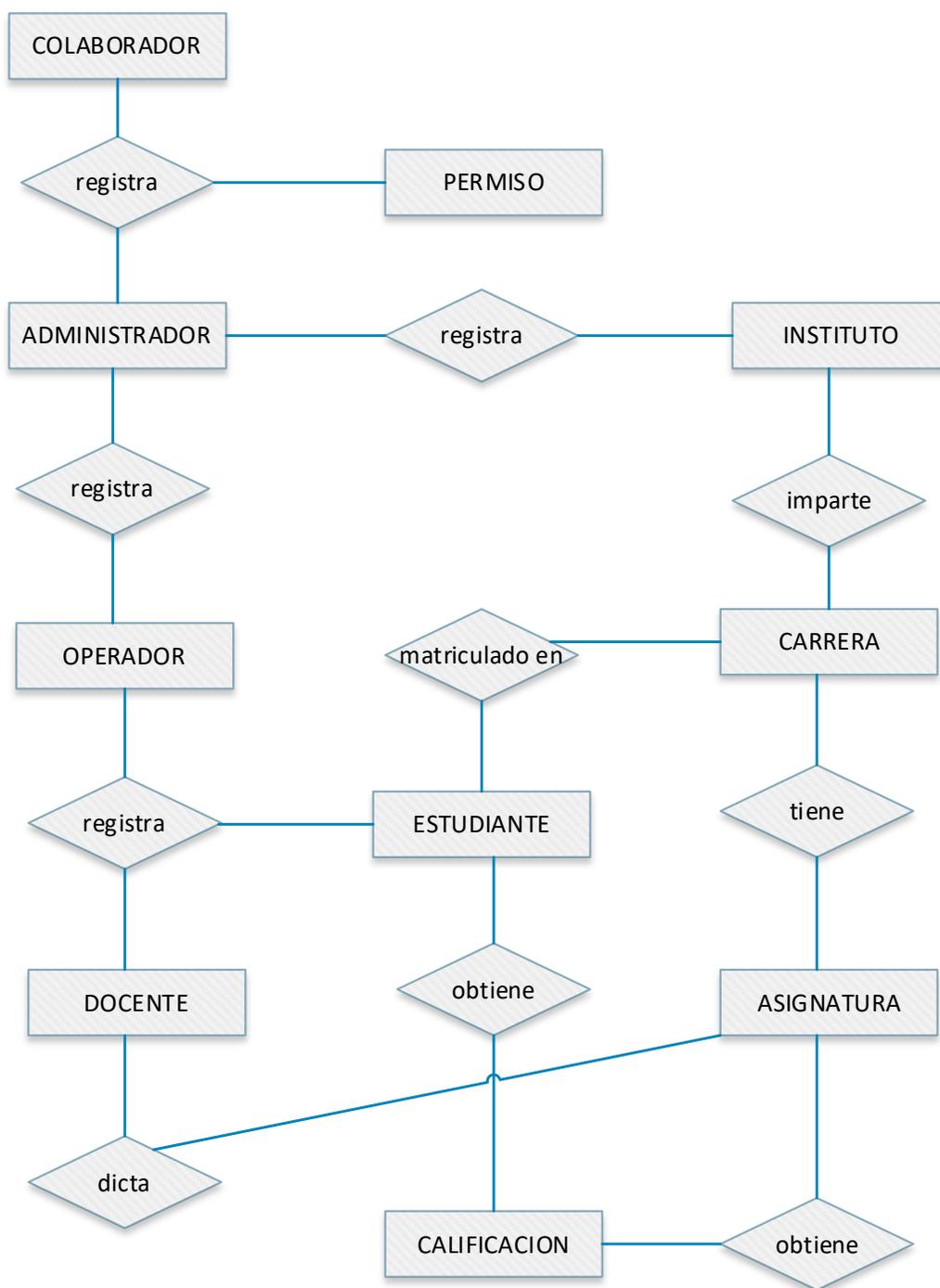


Figura 3. 6: Modelo Entidad – Relación para el Software de Gestión Académica

c) Estructura de Navegación

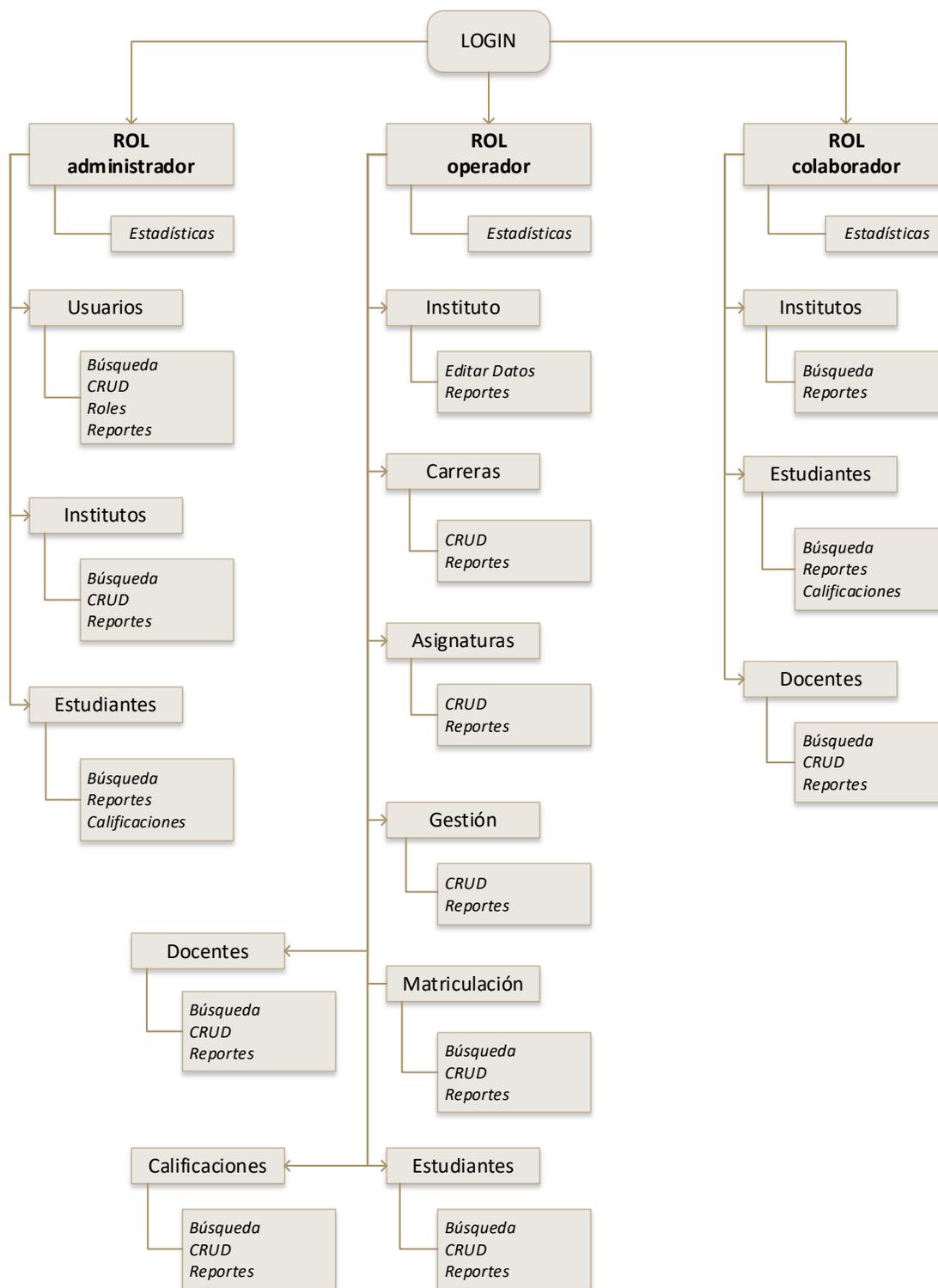


Figura 3. 7: Modelo de Navegación

d) Modelo de Presentación

Esta etapa ofrece una visión abstracta de la interfaz de usuario (IU) basada en el modelo de navegación, tiene la ventaja de ser independiente de las herramientas a utilizar por tratarse de un esquema conceptual.

La figura 3.8 muestra la estructura visual en la maquetación para la interfaz general del usuario. Adicionalmente, dentro de los objetivos se planteó un uso adaptable a dispositivos, es así que la figura 3.9 muestra el esquema de la interfaz para un dispositivo móvil.

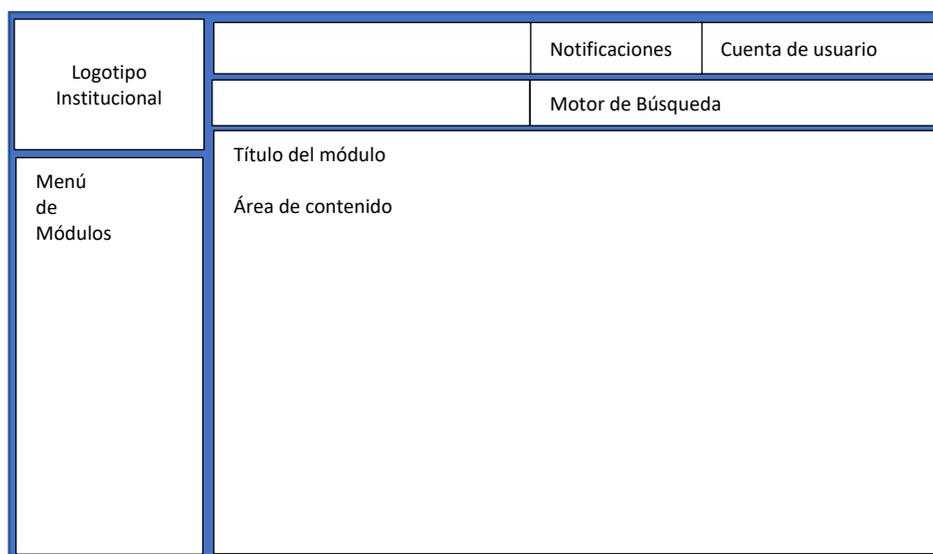


Figura 3. 8: Modelo de presentación general

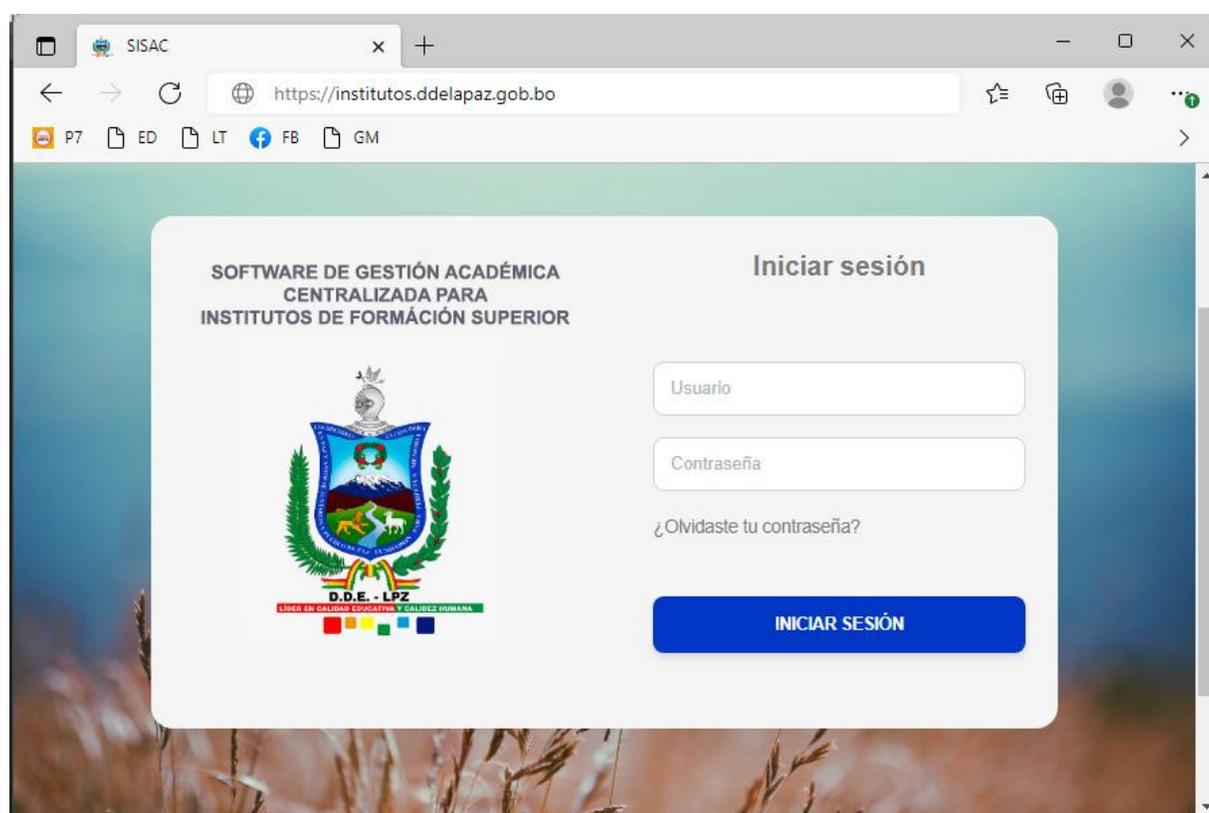


Figura 3. 9: Modelo de presentación en un dispositivo móvil

3.2.3. Construcción

Una vez concretados los aspectos de diseño estructural se procede a la construcción del software partiendo con la codificación según lo planificado en las historias de usuario y lo propuesto en la aplicación de la metodología UWE. Se han cumplido las iteraciones planteadas hasta llegar a los resultados que se exponen a continuación.

La figura 3.10 muestra el inicio de sesión o *login* para los usuarios que previamente han sido definidos de acuerdo a sus roles y controles de seguridad como verificación en dos pasos y definición de contraseñas fuertes.



*Figura 3. 10: Pantalla de inicio de sesión
Fuente: Implementación inicial en ambiente localhost*

Las figuras siguientes muestran el resultado en interfaces de usuario implementadas inicialmente en un ambiente local para ser sometidas a las correspondientes pruebas de aceptación por parte de los usuarios según sus roles definidos.

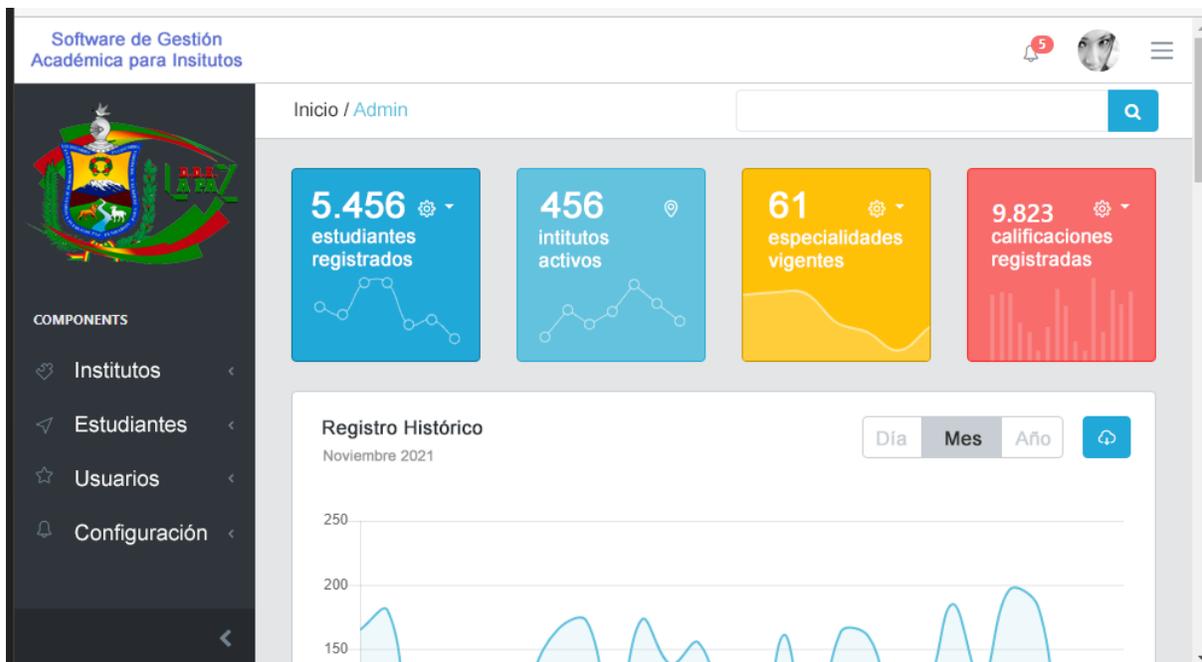


Figura 3. 11: Interfaz de usuario. ROL ADMINISTRADOR
Fuente: Implementación inicial en ambiente localhost

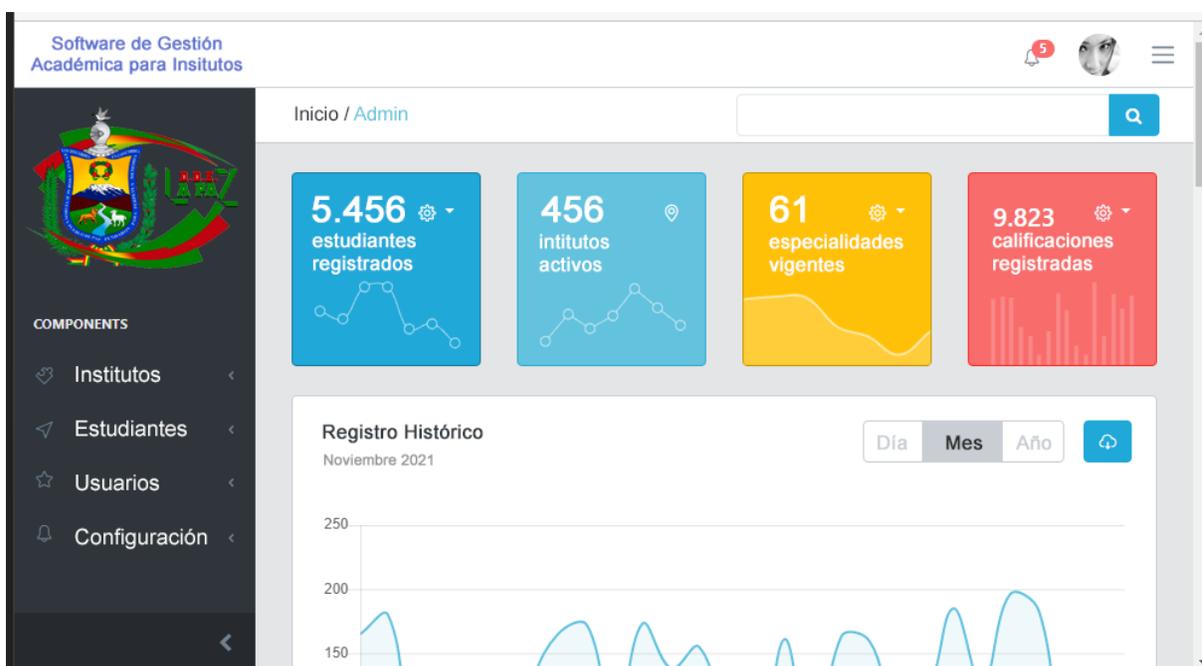


Figura 3. 12: Interfaz de usuario. ROL OPERADOR
Fuente: Implementación inicial en ambiente localhost

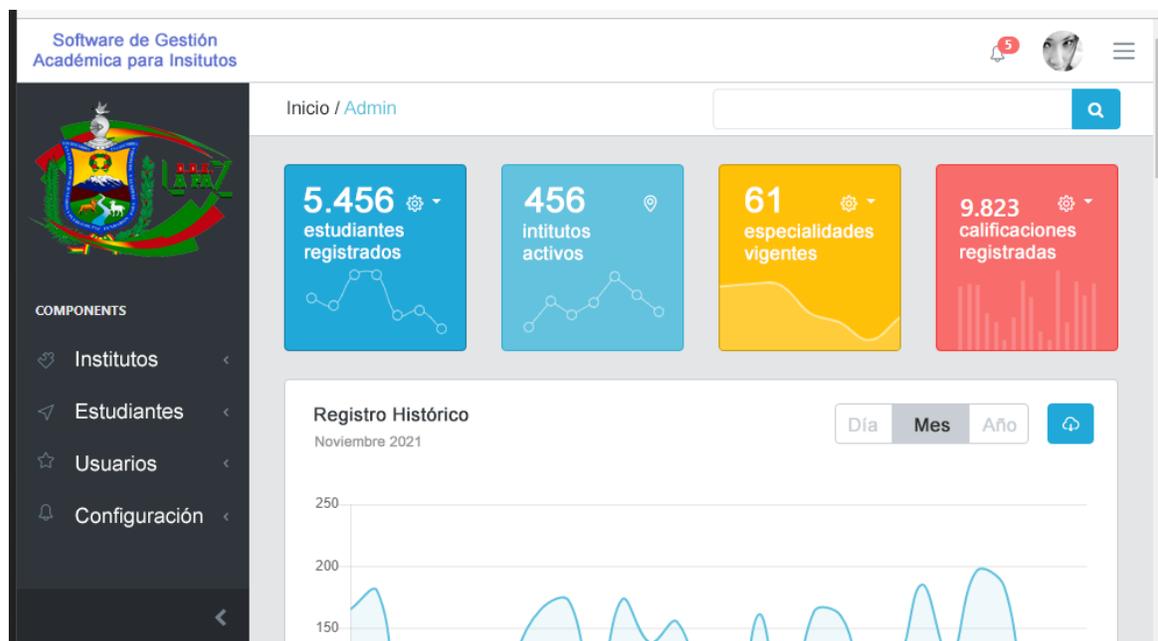


Figura 3. 13: Interfaz de usuario ROL COLABORADOR
Fuente: Implementación inicial en ambiente localhost

Del mismo modo se muestra la pantalla vista en un dispositivo móvil que es una de las características requeridas por el administrador para dar una mayor facilidad de acceso a sus usuarios operadores.



Figura 3. 14: Interfaz de usuario en dispositivo móvil. ROL OPERADOR
Fuente: Implementación inicial en ambiente localhost



Figura 3. 15: Interfaz de usuario en tableta. ROL COLABORADOR
Fuente: Implementación inicial en ambiente localhost

En la entrada de datos, la 3.16 es el formulario de registro para nuevos institutos, el mismo se encuentra debidamente validado.

Módulo Institutos

Registro de Nuevo Instituto

<p>Nombre del Instituto:</p> <input style="width: 90%;" type="text" value="INSTITUTO TÉCNICO COMERCIAL SUPERIOR DE LA NACIÓN TTI"/>	<p>Nombre corto:</p> <input style="width: 90%;" type="text" value="INCOS LA PAZ"/>
<p>Código RIIT:</p> <input style="width: 90%;" type="text" value="80730866"/>	<p>Tipo Instituto:</p> <input type="text" value="Educación Técnica"/>
<div style="text-align: center;"> <p style="color: green; font-weight: bold; margin-top: 5px;">Logo...</p> </div>	<p>Tipo de sede</p> <p><input checked="" type="radio"/> Sede</p> <p><input type="radio"/> Subsede</p>
<p>Departamento:</p> <input type="text" value="La Paz"/>	<p>Dirección:</p> <input type="text" value="Calle Campero N° 94"/>
<p>Provincia:</p> <input type="text" value="Murillo"/>	<p>Zona:</p> <input type="text" value="Central"/>
<p>Municipio:</p> <input type="text" value="La Paz"/>	<p>Teléfono:</p> <input type="text" value="2373296"/>
<p>Código Edificio</p> <input type="text" value="80730598"/>	<p>Celular:</p> <input type="text" value="71256548"/>
<p>Naturaleza:</p> <input type="text" value="Fiscal"/>	<p>Niveles</p> <p><input checked="" type="checkbox"/> Técnico Superior</p> <p><input type="checkbox"/> Técnico Medio</p> <p><input type="checkbox"/> Capacitación</p>
<p>Web:</p> <input type="text" value="www.incoslapaz.org"/>	<p>Correo:</p> <input type="text" value="contacto@incoslapaz.org"/>
<p>Facebook:</p> <input type="text" value="incoslapazoficial"/>	

Figura 3. 16: Formulario para registro de un nuevo instituto
Fuente: Implementación inicial en ambiente localhost

Finalmente, como salida la figura 3.17 es una muestra del formato de reportes generado

MINISTERIO DE EDUCACION VICEMINISTERIO DE EDUCACION SUPERIOR DE FORMACION PROFESIONAL		DIRECCION DEPARTAMENTAL DE EDUCACION LA PAZ - BOLIVIA								
CENTRALIZADOR DE CALIFICACIONES										
UNIDAD: INSTITUTO TÉCNICO BOLIVIANO SUIZO					DISTRITO: LA PAZ					
PLAN: SEMESTRAL			MODALIDAD: Servicios		GESTIÓN: II/2021					
NIVEL: TÉCNICO SUPERIOR		EL NIÑO AUTISTA Y SU CRECIMIENTO PERSONAL	LINGUAJE DE SEÑAS	EL NIÑO ESPECIAL COMO SUJETO DE DERECHOS	AREAS DEL APRENDIZAJE INFANTIL	DESARROLLO DE LAS SOCIEDADES	EL CRECIMIENTO DEL NIÑO ESPECIAL			OBSERVACIONES
CARRERA: EDUCACIÓN ESPECIAL DE APRENDIZAJE										
CURSO: SEGUNDO SEMESTRE										
Nº	NÓMINA									
1	ALTAMIRANO SAENZ, ANA MILENKA	90	94	75	74	72	84			APROBÓ
2	BLANCO GOYTIA, ADELA	76	81	77	82	87	97			APROBÓ
3	CHAVEZ QUISPE, NINOSKA LUZ	90	74	91	97	76	80			APROBÓ
4	ESCOBAR AQUISE, HELEN MARIA	84	85	86	93	71	70			APROBÓ
5	LOPEZ OROS, PAOLA EVELYN	78	88	76	93	75	81			APROBÓ
6	MEJIA MUÑOZ, CARMELA DENISSE	83	88	93	82	77	98			APROBÓ
7	ORO CALLISAYA, MARCO ANTONIO	92	81	93	78	96	93			APROBÓ
8	QUISPE CHURA, BEATRIZ ELENA	94	74	74	95	78	88			APROBÓ
9	RAMOS LOZA, JULIO OMAR	95	90	84	0	0	0			ABANDONÓ
10	TORREZ FLORES, JUAN CLAUDIO	73	82	98	80	93	91			APROBÓ
11	VENTURA COLQUE, NATALY SANDRA	94	73	74	70	94	97			APROBÓ
12	ZAMORA REYES, LORENA EMILY	70	87	86	83	94	75			APROBÓ
13										
14										
15										

Fj-035
20220215 - isalazar

JEFE ACADÉMICO

RECTOR O DIRECTOR

Figura 3. 17: Ejemplo de reporte generado
Fuente: Implementación inicial en ambiente localhost

3.3. SEGURIDAD

La seguridad informática es un aspecto de suma importancia porque existe un permanente riesgo de ataques y robo de información dentro de las empresas o por amenazas del exterior. Sin embargo, a nivel interno también se corren riesgos por usuarios mal intencionados o usuarios que inadvertidamente por falta de conocimiento causan mal funcionamiento del sistema con pérdida de información. Los ataques a software aprovechan cualquier agujero de seguridad informática que es necesario prevenir mediante distintas herramientas a disposición y criterios de seguridad.

En el desarrollo del software objeto del presente trabajo, se han previsto estas contingencias bajo los niveles que se exponen a continuación.

3.3.1. Nivel de Aplicación

Determina características de seguridad dentro de las aplicaciones para evitar vulnerabilidades contra amenazas, tales como la modificación y el acceso no autorizados ante posibles ataques.

a) Validación de acceso

La versión Laravel 8 cuenta con la librería *jeststream* desarrollada sobre PHP y JS, que se encarga de todo el proceso de acceso al sistema desde la pantalla de *login* con las correspondientes validaciones. Cuenta con las funcionalidades de verificación por correo electrónico, autenticación de dos factores, revisión de contraseñas débiles, encriptación AES de 256 bits, administrador de sesiones con límite de tiempo por inactividad, procedimiento de restitución de cuenta.

b) Navegación segura

Los *Cross Site Request Forgery*, son falsificaciones de solicitud entre sitios durante el envío de datos a través de formularios. Laravel incluye tokens CSRF almacenados en la sesión del usuario para la operación en todos los formularios, también hace uso de *middlewares* que son mecanismos utilizados para filtrar las peticiones HTTP en una aplicación, verifica permanentemente que un usuario esté autenticado para hacer uso del sistema, caso contrario será direccionado a la pantalla de *login*.

c) Niveles de privilegio

El sistema cuenta con tres niveles de usuario: administrador, operador y colaborador. De este modo se organiza el acceso a las distintas funcionalidades del software basándose en el tipo de rol que tenga el usuario según sus responsabilidades dentro de la institución con la definición de políticas de restricción y acceso a los procedimientos operativos establecidos en reuniones de coordinación.

d) Inyección JavaScript

Más conocido como XSS (*Cross-Site Scripting*) es un ataque que puede ejecutar código -comúnmente JS- del lado del navegador mediante formulario o en parámetros alterados en la URL. Para contrarrestar esta vulnerabilidad se ha definido adecuadamente una capa *middleware* que hace de filtro entre la interfaz de usuario y las operaciones del sistema con la cual se controla permanentemente los datos recibidos desde el navegador al margen de la validación para la Base de Datos.

e) Protocolo de comunicación

Al existir permanente flujo de datos -en su mayoría confidenciales- a través de campos de formulario y ante la precaución de que estos puedan ser interceptados maliciosamente, se ha configurado el servidor con el protocolo HTTPS para alojar la aplicación mediante un acceso de conexión segura. De este modo se brinda privacidad, seguridad y protección de datos a los usuarios para prevenir ataques como suplantación de identidad y redireccionamientos no percibidos.

3.3.2. Nivel de Base de Datos

La seguridad en Bases de Datos está dada por las medidas, tecnologías y protocolos que una organización implementa para proteger los datos que gestiona, así como para establecer candados que impidan el acceso a dicha información a personas o entidades no autorizadas. (Terrerros, 2021)

a) Inyección SQL

Para este tipo de vulnerabilidad -presente en interfaces con formularios- desde sus primeras versiones Laravel cuenta motor de consultas Eloquent basado en ORM que es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos así se previene estos ataques con la conversión de script SQL a objetos de PHP y viceversa.

b) Copias de seguridad

Se ha planificado generar respaldos de la base de datos semanalmente extraídas en DVD en doble copia debidamente organizadas en orden cronológico. El encargado de esta operación con todos los privilegios de acceso es el administrador del sistema quien podrá eventualmente realizar una restauración de la base de datos en base a este material.

c) Registro de transacciones

Es el denominado *log*, que consiste en una tabla conteniendo todo el detalle de las operaciones realizadas sobre la base de datos especificando información como: usuario, hora de dichas operaciones, tipos de operaciones realizadas. Esto es muy útil en un caso de restitución de la información y realizar un seguimiento histórico de los usuarios que ejecutan las distintas operaciones sobre la base de datos.

3.4. CALIDAD DE SOFTWARE

Se ha tomado la usabilidad como criterio inicial para la evaluar la calidad del proyecto. Los demás aspectos de medición de calidad tendrán un tratamiento más detallado con una observación permanente del rendimiento del software.

3.4.1. Usabilidad

Consiste en un conjunto de atributos que permiten evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el software también visto como la capacidad del software de ser entendido, aprendido y usado de forma fácil y atractiva.

Se valoran los siguientes criterios.

- Entendimiento. Capacidad del producto software que permite al usuario entender si el software es adecuado y como puede ser usado para unas tareas o condiciones de uso particular.
- Aprendizaje. Capacidad del producto software que permite al usuario aprender sobre su aplicación.

- Operabilidad. Capacidad del producto software que permite al usuario operarlo y controlarlo.
- Atracción. Capacidad del producto software para ser atractivo al usuario en cuanto al aspecto visual.
- Cumplimiento de la usabilidad. Capacidad del producto software para adherirse a normas, convenciones, guías de estilo o regulaciones con la usabilidad.

Con todos estos criterios de análisis, para determinar el valor de la usabilidad del sistema se optó por realizar un cuestionario a los usuarios basado en los criterios valorativos de la escala de Likert donde se asigna un valor cuantitativo que permite medir el nivel de conformidad a cada aspecto que se plantea.

La tabla 3.2 describe la escala valorativa que se aplicada en este sondeo con la correspondiente escala valorativa.

Tabla 3. 2: Valoración en la escala de Lickert

Criterio en la escala de Lickert	Escala Valorativa
Totalmente en desacuerdo	1
En desacuerdo	2
Neutral	3
De acuerdo	4
Totalmente de acuerdo	5

Por otro lado, la tabla 3.3 muestra las preguntas formuladas junto a la valoración cuantitativa que deberá decidir cada usuario según su propio criterio de acuerdo s su experiencia en el uso del software.

*Tabla 3. 3: Conjunto de preguntas formuladas a los usuarios
(se ha asignado un valor a la escala de Likert)*

P	Pregunta	1	2	3	4	5
1	Puede ingresar al sistema con facilidad					
2	La interfaz de usuario es intuitiva y sencilla de entender					
3	Los datos obtenidos que vio fueron fáciles de comprender					
4	El sistema responde rápidamente a sus solicitudes requeridas					
5	Los reportes se imprimen de manera correcta en cuanto a formato y presentación					
6	Se reduce el tiempo de trabajo en generar informes específicos					
7	Las estadísticas generadas son acordes a los resultados requeridos					
8	Se adapta a dispositivos móviles					
9	La presentación de datos es coherente al visualizar en dispositivos móviles					
10	Existe una óptima validación al momento de introducir datos					
11	Los motores de búsqueda brindan resultados precisos					
12	Es fácil navegar entre las distintas opciones definidas por la interfaz					

Este cuestionario se ha aplicado en un universo de 9 usuarios entre operadores y colaboradores. La tabla 3.4 expone los resultados centralizados obtenidos de la valoración numérica.

- La cabecera representan los usuarios consultados u1 .. u9
- Cada fila expresa el número de pregunta aplicada p1 .. p12

Tabla 3. 4: Resultado obtenido en detalle de las encuestas
(Aplicadas sobre 9 usuarios para 12 preguntas)

	u1	u2	u3	u4	u5	u6	u7	u8	u9
p1	4	4	4	5	4	5	4	4	5
p2	5	5	4	4	5	5	5	4	4
p3	5	3	5	4	5	4	3	5	4
p4	5	5	5	5	4	4	4	5	5
p5	5	5	5	3	5	5	5	4	5
p6	5	5	4	5	5	5	4	4	4
p7	5	4	4	4	5	4	5	5	4
p8	5	5	5	5	4	5	5	5	4
p9	5	5	5	4	3	4	5	3	5
p10	4	5	3	5	4	5	4	4	4
p11	5	4	5	5	4	5	5	5	4
P12	4	5	5	3	4	4	5	4	5

Se ha determinado la media para cada valoración realizada por el usuario:

$$valoración = \sum_{i=1}^9 \frac{u_i}{n}$$

Dónde:

u_i : valoración de cada usuario

n : cantidad de preguntas

La tabla 3.5 muestra los resultados obtenidos a partir de este cálculo donde la última columna expresa la *valoración media (vm)* general de lo analizado.

Tabla 3. 5: Resultados de la valoración media de cada usuario

	u1	u2	u3	u4	u5	u6	u7	u8	u9	\overline{vm}
Valoración media	4,82	4,55	4,45	4,45	4,36	4,64	4,45	4,36	4,36	4,49

Finalmente, para obtener una escala en términos porcentuales se toma el criterio de que la escala máxima tiene un valor 5 contra el cual se determina la proporción.

$$\frac{4,49}{5} \times 100 = 89.8$$

Por tanto, la usabilidad del sistema tiene un valor de 89.8 por ciento.

Según la ISO 25010, este resultado puede ser interpretado como el nivel de capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario.

Por otra parte, en un enfoque adicional de la información recabada en los cuestionarios, la tabla 3.6 muestra el nivel de *conformidad media (cm)* en una escala de 1 a 5 a cada pregunta planteada bajo el siguiente criterio:

$$\text{conformidad media} = \sum_{i=1}^9 \frac{c_i}{n}$$

Dónde:
 c_i : nivel de conformidad a cada pregunta
 n : cantidad de usuarios

Tabla 3. 6: Conformidad media a cada pregunta planteada
(sobre una escala de 1 a 5 para 12 preguntas en criterio de 9 usuarios)

P	Pregunta	\overline{cm}
1	Puede ingresar al sistema con facilidad	4,3
2	La interfaz de usuario es intuitiva y sencilla de entender	4,6
3	Los datos obtenidos que vio fueron fáciles de comprender	4,2
4	El sistema responde rápidamente a sus solicitudes requeridas	4,7
5	Los reportes se imprimen de manera correcta en cuanto a formato	4,7
6	Se reduce el tiempo de trabajo en generar informes específicos	4,6
7	Las estadísticas generadas son acordes a los resultados requeridos	4,4
8	Se adapta a dispositivos móviles	4,8
9	La presentación de datos es coherente al visualizar en dispositivos móviles	4,3
10	Existe una óptima validación al momento de introducir datos	4,3
11	Los motores de búsqueda brindan resultados precisos	4,7
12	Es fácil navegar entre las distintas opciones definidas por la interfaz	4,4

De los resultados extremos se puede realizar las siguientes observaciones:

P8: Se adapta a dispositivo móviles – Conformidad media: **4.8**

Es el resultado que ha tenido más aceptación lo cual indica que la adaptabilidad es una característica innovadora y útil para los usuarios precisamente porque el sistema antiguo no poseía este comportamiento.

P3: Los datos que vio fueron fáciles de entender – Conformidad media: **4.2**

Tiene el resultado más bajo posiblemente por dificultades en adaptarse para interpretar rápidamente la información mostrada en pantalla debido a que los usuarios se encuentran acostumbrados al formato del anterior sistema. Esto se solucionará paulatinamente con la debida capacitación de este nuevo software.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

El Software de Gestión Académica para Institutos de Formación Superior en el Departamento de La Paz contribuye las operaciones administrativas tanto para los funcionarios de la DDE LP como para los usuarios operadores de cada instituto y que por ende beneficia al público constituido por el universo de estudiantes que requieren certificaciones oficiales.

El diseño del modelado de base de datos propuesto abarca lo requerido para un adecuado almacenamiento de la información con que trabaja la institución que además proporciona una base útil para los motores de búsqueda. En cuanto a fundamentos cumple con principios de mínima redundancia, integridad de datos, soporte de consultas complejas optimizadas y seguridad de acceso.

La interfaz de usuario desarrollada es adaptable a dispositivos tanto en funcionalidad como en diseño por lo cual se cumple con los actuales paradigmas de experiencia de usuario. Tiene como característica una navegabilidad más amigable que permite procedimientos ágiles mediante teclado para el acceso a todas las opciones, ya sea de menú o al momento de introducir datos en formularios en operaciones CRUD.

Se han definido adecuadamente privilegios de uso bajo roles y permisos que garantizan un acceso organizado bajo los correspondientes ámbitos de responsabilidad asignado a cada uno de los usuarios.

Los reportes estadísticos generados en los distintos módulos se organizan bajo varios criterios parametrizables que coadyuban en gran manera a una visión general del comportamiento de la información para la toma de decisiones tanto operativas como estratégicas.

Los motores de búsqueda con la característica de texto autocompletable, ofrecen resultados precisos en la ubicación de información que permiten a través de enlaces explorar en más detalle sobre un elemento seleccionado.

4.2. RECOMENDACIONES

Implementar procedimientos para la inclusión de códigos QR en las distintas certificaciones emitidas con lo cual se tendrá una herramienta de verificación de autenticidad para estos documentos que se encontrarán almacenados en los servidores de la DDE LP.

Integrar un módulo para administrar la vinculación de la información del software implementado a la web institucional desde donde el público podría realizar consultas de interés específico como situación académica personal o de interés general como visualización de las estadísticas sobre indicadores educativos, búsqueda de institutos (privados, convenio, públicos) junto a sus datos de contacto y ubicación, además de las carreras técnicas impartidas, búsqueda de especialidades y dónde se las dictan.

En el contexto tecnológico actual de facilidad de acceso a la información, es posible desarrollar una aplicación móvil orientada al público para que puedan acceder de manera más práctica a toda la información descrita anteriormente.

Incorporar en el módulo administrativo operaciones a nivel sistema donde se gestione aspectos de generación de copias de seguridad y su historial, restauración de la base de datos, visualización de las transacciones.

BIBLIOGRAFÍA

- Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Boston: Addison Wesley.
- Blanco, L. (2008). *EcuRed*. Obtenido de https://www.ecured.cu/Sistema_inform%C3%A1tico
- Bohem, B. W. (1976). *Practical Strategies for Developing Large Software Systems*. California: Addison-Wesley Publishing Company.
- Cohn, M. (2004). *User Stories Applied: For Agile Software Development*. Boston: Addison-Wesley.
- Delgado, L. S. (2006). *Mejoramiento de la Gestion Educativa a travez de Procesos de Calidad debe bajar los indices de desersion en el Colegio de María de Bogota*. Granada: Universidad de Granada.
- Fernandez, E. (2003). *Evaluacion de la Gestion Institucional en Instituciones de Educacion Superior Privadas*. México: Universidad Valle de Mexico.
- García, R. (28 de 06 de 2019). *Muy Agile*. Obtenido de <https://muyagile.com/historias-de-usuario-me-ayudas-a-entenderlo/>
- Guerrero, C. M., & Ucán, J. P. (2014). *UWE en Sistema de Recomendación de Objetos de Aprendizaje*. Mérida: Universidad Autónoma de Yucatán.
- IEEE. (18 de 01 de 1991). *IEEE 610-1990 - IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*. Obtenido de <https://standards.ieee.org/standard/610-1990.html>
- Junquera, A. (24 de 10 de 2019). *Grupo Digital*. Obtenido de <https://www.grupodigital.eu/blog/metodologias-agiles/>
- Koch, N. (2000). *The Authoring Process of the UML-based Web Engineering Approach*. Munich: Institute of Computer Science Ludwig-Maximilians.
- Letieler, P. (15 de 12 de 2005). *Ciencia y Técnica Administrativa*. Obtenido de <https://www.cyta.com.ar/ta0502/v5n2a1.htm>
- Maida, E. G. (2015). *Metodologías de Desarrollo de Software*. Buenos Aires: Facultad de Ingeniería - Pontificia Universidad Católica Argentina Santa Maria de los Buenos Aires.
- Mancuzo, M. (08 de 08 de 2020). *Compara Software*. Obtenido de <https://blog.comparasoftware.com/metodologia-xp/>
- McLeod, R. J. (2000). *Sistemas de Información Gerencial*. Bogotá: Prentica Hall.
- Padilla, J. L. (2014). *Sistema para la inserción laboral y seguimiento de egresados en la facultad de odontología e instituto de investigación y posgrado de la Universidad Central del Ecuador*. Quito. Obtenido de <http://www.dspace.uce.edu.ec/handle/25000/2935>

- Pressman, R. (2004). *Ingeniería de Software. Un enfoque práctico*. México: McGraw Hill.
- Rossi, G., & Pastor, O. (2008). *Web Engineering: Modelling and Implementing Web Applications*. Londres: Springer.
- Rumbaugh, J., Jacobson, I., & Booch, G. (1999). *El lenguaje de modelado unificado. Guía del usuario*. Madrid: Addison Wesley.
- Santander, B. (21 de 12 de 2020). *Becas Santander*. Obtenido de <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>
- Smith, C. (2011). *Control automático de procesos: teoría y práctica*. México: Limusa.
- Sommerville, I. (2011). *Ingeniería de Software*. Mexico: Pearson.
- Terreros, D. (14 de 09 de 2021). *Seguridad de base de datos*. Obtenido de <https://blog.hubspot.es/marketing/que-es-seguridad-base-de-datos>
- Vergara, S. (10 de 08 de 2021). *ITDOH SOLUTIONS*. Obtenido de <https://www.itdo.com/blog/que-son-las-historias-de-usuario/>
- Viezca, J. (29 de 09 de 2011). *Programación Orientada a Objetos y Taller*. Obtenido de <https://jviezca.blogspot.com/2011/09/tarjetas-crc.html>

DOCUMENTACIÓN