

**UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA**

*La Universidad Católica de Loja*

**FACULTAD DE INGENIERÍAS Y ARQUITECTURA**

**CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN**

**Diseño e implementación de un sistema de**

**seguimiento de tratamientos fisioterapéuticos. Caso de estudio: GYMES**

Trabajo de integración curricular previo a la obtención del título de:

**Ingeniero en Tecnologías de la Información**

**Autor:** Pasquel Chamorro, Pablo Andrés

**Director:** Cadme Samaniego, Irma Elizabeth

CENTRO UNIVERSITARIO QUITO

2023

# Aprobación del director del Trabajo de Integración Curricular

Loja, día de mes de año

Título académico completo, (no colocar siglas)

Nombres y Apellidos completos del director de la carrera

**Director de la carrera de Xxxxxxxxxx**

Ciudad.-

De mi consideración:

Me permito comunicar que, en calidad de director del presente Trabajo de Integración Curricular denominado: (nombre del trabajo) realizado por Nombres y Apellidos completos del autor o autores (as) ha sido orientado y revisado durante su ejecución, así mismo ha sido verificado a través de la herramienta de similitud académica institucional, y cuenta con un porcentaje de coincidencia aceptable. En virtud de ello, y por considerar que el mismo cumple con todos los parámetros establecidos por la Universidad, doy mi aprobación a fin de continuar con el proceso académico correspondiente.

Particular que comunico para los fines pertinentes.

Atentamente,

Director: Nombres y Apellidos completos del Director del Trabajo de Integración Curricular y título académico.

C.I.:

Correo electrónico:

# Declaración de **autoría** y cesión de derechos

Yo, Pablo Andrés Pasquel Chamorro, declaro y acepto en forma expresa lo siguiente:

Ser autor (a) del Trabajo de Integración Curricular denominado: …………………, de la carrera de…………….…., específicamente de los contenidos comprendidos en: (se debe colocar los nombres de los capítulos elaborados en el Trabajo de Integración Curricular), siendo (nombres y apellidos completos), director (a) del presente trabajo; también declaro que la presente investigación no vulnera derechos de terceros ni utiliza fraudulentamente obras preexistentes. Además, ratifico que las ideas, criterios, opiniones, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad. Eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones judiciales o administrativas, en relación a la propiedad intelectual de este trabajo.

Que la presente obra, producto de mis actividades académicas y de investigación, forma parte del patrimonio de la Universidad Técnica Particular de Loja, de conformidad con el artículo 20, literal j), de la Ley Orgánica de Educación Superior; y, artículo 91 del Estatuto Orgánico de la UTPL, que establece: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado que se realicen a través, o con el apoyo financiero, académico o institucional (operativo) de la Universidad”, en tal virtud, cedo a favor de la Universidad Técnica Particular de Loja la titularidad de los derechos patrimoniales que me corresponden en calidad de autor/a, de forma incondicional, completa, exclusiva y por todo el tiempo de su vigencia.

La Universidad Técnica Particular de Loja queda facultada para ingresar el presente trabajo al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública, en cumplimiento del artículo 144 de la Ley Orgánica de Educación Superior.

.................................................................

Autor: Nombres y Apellidos completos del autor

C.I.:

Correo electrónico:

# Dedicatoria

# Agradecimiento

# Índice de contenido

[Aprobación del director del Trabajo de Integración Curricular II](#_Toc126683494)

[Declaración de autoría y cesión de derechos III](#_Toc126683495)

[Dedicatoria VI](#_Toc126683496)

[Agradecimiento VII](#_Toc126683497)

[Índice de contenido VIII](#_Toc126683498)

[Resumen 1](#_Toc126683499)

[Abstract 2](#_Toc126683500)

[Introducción 3](#_Toc126683501)

[Contexto 3](#_Toc126683502)

[Problemática 3](#_Toc126683503)

[Capítulo uno 7](#_Toc126683504)

[Introducción 7](#_Toc126683505)

[1.1. Problemática 7](#_Toc126683506)

[1.2. Justificación 7](#_Toc126683507)

[1.3. Fases del proceso de desarrollo de software 8](#_Toc126683508)

[Capítulo dos 9](#_Toc126683509)

[Proceso de desarrollo de Software 9](#_Toc126683510)

[2.1. Fases del proceso de desarrollo de software 9](#_Toc126683511)

[2.1.1. Especificación de software 10](#_Toc126683512)

[Capítulo tres 20](#_Toc126683513)

[Nombre del capítulo 20](#_Toc126683514)

[Conclusiones 21](#_Toc126683515)

[Recomendaciones 22](#_Toc126683516)

[Referencias 23](#_Toc126683517)

[Apéndice 27](#_Toc126683518)

**Índice de tablas**

**Índice de figuras**

# Resumen

# Abstract

# Introducción

# Capítulo uno

# Generalidades

## Problemática

GYMES es un centro de rehabilitación médica especializada en fisioterapia que se encuentra en la ciudad de Quito. Este centro ofrece sus servicios a la comunidad desde hace 5 años, y gracias a la labor de sus profesionales y los buenos resultados que han tenidos los pacientes en sus tratamientos, ha sido posible que este centro médico tenga mayor acogida por parte de la comunidad razón por la cual esta se ve en la necesidad de mejorar sus servicios para atender a una mayor cantidad de pacientes.

Actualmente uno de sus principales problemas es la forma de gestionar el registro de información médica de los pacientes ya que lo realiza de forma manual en formatos elaborados por el personal médico y se lo archiva en las oficinas del centro, lo cual conlleva la utilización de mucho tiempo en la elaboración, búsqueda y archivo de los registros y también en el uso de espacio físico en el lugar.

## Justificación

El presente trabajo tiene como finalidad el diseño y la implementación de un sistema que automatice la forma en que se llevan los registros de las sesiones de fisioterapia para los pacientes que acuden al centro de rehabilitación física GYMES.

Basándonos en el contexto planteado en la problemática, se propone implementar un sistema web que permita realizar un seguimiento sobre los tratamientos ofrecidos, de esta forma el centro podrá contar con un registro automatizado el cual brindará la información sobre las sesiones a las cuales el paciente ha asistido, realizar un acompañamiento sobre su evolución y llegar a la conclusión si una terapia ha sido efectiva.

* 1. **Objetivos**
     1. **Objetivo General**
* Diseñar e implementar un sistema de seguimiento de tratamientos fisioterapéuticos para el Centro Médico Gymes
  + 1. **Objetivos Específicos**
* Realizar una revisión bibliográfica sobre el tema a desarrollar mediante una revisión sistemática de literatura.
* Recolectar los requisitos de usuario mediante técnicas de recolección de información para su posterior análisis y clasificación.
* Diseñar la solución mediante la selección de la arquitectura, la creación de prototipos y el diseño de la experiencia de usuario para la fase de implementación.
* Desarrollar la solución mediante la codificación de los componentes a partir de los diseños obtenidos previamente.
* Verificar la solución mediante pruebas del sistema en el lugar de trabajo para comprobar su efectividad.
  1. **Componentes**

1. Para el proceso de desarrollo de la aplicación web, se aplicarán elementos de la metodología UWE (UML-based Web Engineering) la cual nos provee pautas para la captura de requisitos y el diseño de la solución. El front-end hará uso de tecnologías como HTML, CSS, Javascript y el framework Bootstrap para asegurar un diseño responsivo y adaptable a los diferentes tipos de dispositivos, el back-end se hará uso de tecnologías como PHP y MySQL como base de datos de la aplicación.
   1. **Metodología**

Se utilizarán elementos del desarrollo de aplicaciones ágil, elementos de UWE (UML-Based Web Engineering) para el análisis y diseño de la solución y una arquitectura en capas con un estilo Modelo-Vista-Controlador para separar la lógica de cada componente. Se utilizarán herramientas DevOps como el control de versiones mediante GitHub, herramientas de colaboración como Microsoft Teams y herramientas de planificación como Microsoft Project o Jira.

* 1. **Estructura del documento**

El capítulo uno abordará el marco teórico el cual explicará sobre las fases del proceso de desarrollo de software. En el capítulo dos Se definirá la metodología de desarrollo a usar durante la elaboración del proyecto.

# Capítulo dos

# Proceso de desarrollo de Software

## Fases del proceso de desarrollo de software

El ciclo de vida del desarrollo de software es un enfoque sistemático usado para desarrollar sistemas de software de alta calidad, eficiencia y que cumplan las expectativas del usuario (Govil & Sharma, 2022). Según McGilvray (2021), el ciclo de vida del desarrollo de software (SDLC) define el enfoque y las fases para desarrollar soluciones de software, es decir, contiene todas las actividades, metodologías, modelos, procedimientos, etc. Que permiten el desarrollo de un software.

La siguiente figura presenta en forma secuencial las fases típicas que presenta un proceso de desarrollo de software a través de su SDLC:

**Figura 1**

*Fases del ciclo de vida de desarrollo del software*

*Nota*:

A continuación, se procede a dar una breve explicación de cada fase.

**Especificación:** Se procede a identificar, evaluar e investigar detalladamente las necesidades y metas del usuario. Una vez identificados se establecen objetivos, prioridades, se planifican las actividades y se refinan los planes (McGilvray, 2021).

**Diseño:** Se definen posibles soluciones y se selecciona la mejor opción basado en dependencias y restricciones. Se determina como cumplir los requisitos funcionales y no funcionales y se ajustan los planes conforme a lo obtenido (McGilvray, 2021).

**Desarrollo:** También llamado construcción del software, las actividades de esta fase comprenden la programación de líneas de código y su ejecución, la programación de la base de datos y la creación de interfaces de usuario (Mendoza et al., 2020).

**Validación:** En esta fase se determina si los requerimientos han sido satisfechos. Se deben hacer varias pruebas de unidad y del sistema en general, con el objetivo de asegurar que todos los elementos del sistema funcionen como se espera (Rahmani & Hikmawati, 2020).

**Evolución:** Es la fase donde se mejora la solución y se capacita a los usuarios en el entorno operacional. En este punto es donde aparecen los problemas los cuales deben ser resueltos mediante actualizaciones rutinarias (Dwivedi et al., 2022).

## Especificación de software

La especificación de requisitos se la realiza mediante la ingeniería de requisitos la cual “se define como una aplicación disciplinada de principios científicos y técnicas para desarrollar, comunicar y gestionar requerimientos” (Medina Cruz et al., 2019), mediante la cual se establecen los servicios y límites del sistema conforme a su operación y desarrollo.

Algo que resaltar es la diferencia que existe entre requisito y requerimiento. (Peláez Valencia et al., 2019) menciona que el requerimiento determina las necesidades del cliente con un bien en específico, mientras que el requisito se basa en las necesidades del cliente respecto a las funcionalidades de operación del producto.

También menciona que “el origen de este problema es debido a la mala traducción del idioma inglés ya que la palabra requirement (requisito en inglés) puede fácilmente ser mal traducida como requerimiento (request en inglés).”(Peláez Valencia et al., 2019).

Un documento de especificación de software por sus siglas en inglés SRS es el artefacto más significativo en el proceso de desarrollo de software ya que influye directamente en éxito del proyecto. Contiene todos los requisitos funcionales RF y los requisitos no funcionales RNF(Asif et al., 2019).

* + - 1. **Requisitos Funcionales**

**“**Los requisitos funcionales describen la interacción entre el sistema y su ambiente, así como la forma en la que el sistema debe comportarse ante determinado estímulo.**”** (Elías Chanchí et al., 2019)**.** Estos requisitos describen de manera precisa lo que el sistema debe hacer y las interacciones que se prevé que el usuario tendrá con el software.

Se presenta una tabla con el dominio de requisitos funcionales comunes en el proyecto:

Tabla 1

*Dominio de requisitos funcionales*

|  |  |
| --- | --- |
| ID | Requisitos comunes |
| RF-01 | Autenticación |
| RF-02 | Autorización |
| RF-03 | Operaciones CRUD (Create, Read, Update, Delete) |
| RF-04 | Reglas de negocio |
| RF-05 | Captura de datos |
| RF-06 | Requisitos de búsqueda |
| RF-07 | Interfaces externas |

Nota: Adaptado de (Sarria Mediaceja, 2019)

* + - 1. **Requisitos no funcionales**

Estos requisitos hacen referencia a la calidad del producto de software, algunos de estos requisitos pueden ser: rendimiento, fiabilidad, exactitud, seguridad, usabilidad, etc. También describen un problema en cuanto a la restricción de funciones o prestaciones ofrecidas por el sistema (Elías Chanchí et al., 2019).

Se presenta una tabla con el dominio de requisitos funcionales comunes en el proyecto:

Tabla 2

*Dominio de requisitos no funcionales*

|  |  |
| --- | --- |
| ID | Requisitos comunes |
| RNF-01 | Seguridad |
| RNF-02 | Usabilidad |
| RNF-03 | Portabilidad |
| RNF-04 | Accesibilidad |
| RNF-05 | Adaptabilidad |
| RNF-06 | Rendimiento |
| RNF-07 | Regulaciones |

Nota: Adaptado de (Sarria Mediaceja, 2019)

* + 1. **Diseño**

La fase de diseño se basa en la especificación de los requisitos obtenidios previamente, el diseño define cómo estos requisitos se cumplirán así como también la estructura que se le debe dar a la aplicación web (Blas Cerda, 2020).

UWE (UML Web Engineering) propone la elaboración de cinco modelos que utilizan la notación estándar basada en UML (Unified Modeling Language), estos son: Diagrama de casos de uso, Diagrama de contenido, Diagrama de navegación, Diagrama de presentación y diagrama de proceso (Chán et al., 2021).

La metodología UWE provee gran manejabilidad de requerimientos debido a su inclusión en etapas tempranas del proceso de desarrollo. Toma en consideración tres dimensiones escenciales para el diseño, incluyendo la lógica del negocio (Molina-Ríos & Pedreira-Souto, 2020):

**Figura 2**

*Dimensiones en la metodología UWE*

Nota: Adaptado de (Molina-Ríos & Pedreira-Souto, 2020)

* + 1. **Desarrollo**

En esta etapa, los requisitos se convierten en código siguiendo los lineamientos de diseño definidos en la etapa anterior y se acompaña de pruebas unitarias para verificar la especificación de cada una (Solano-Fernández et al., 2020).

* + 1. **Validación**

La validación es una parte esencial de cualquier desarrollo de software. Cada nueva versión debe ser testeada para comprobar su funcionalidad, usabilidad, seguridad, etc. Se pueden incorporar metodologías de prueba de software con el fin de corregir errores y asegurar que el producto tiene calidad. Hay diferentes enfoques de pruebas como pueden ser: pruebas del sistema, pruebas orientadas a objetos, pruebas de módulos, pruebas de aceptación, etc. Además estas pueden ser estáticas o dinámicas (Alenezi & Almuairfi, 2019).

Las pruebas estáticas son aquellas que se las realiza sin ejecutar el programa y son llamadas “actividades de verificación”, son basadas en la revisión de código. Mientras que en las pruebas dinámicas el programa se prueba en tiempo de ejecución con sus entradas y salidas para comprobar su correcto funcionamiento (Umar, 2020).

A continuación se presenta un diagrama que presenta las categorías de pruebas del software:

**Figura 3**

***Categorías de pruebas de software***

Nota: Adaptado de (Umar, 2020)

* + 1. **Evolución**

En esta etapa se resuelven los problemas encontrados en la etapa de pruebas y se realiza cualquier cambio en el software existente con el fin de adaptar los requisitos de usuario y de mercado. Esta etapa también es considerada el mantenimiento del software (Alfawair, 2022).

* 1. **Metodologías de desarrollo**

“Una metodología es una colección estructurada de procedimientos que ayudan a los desarrolladores de software en sus proyectos, ofreciendo una guía para la toma de decisiones, así como para planificarlo, gestionarlo, controlarlo y evaluarlo” (Milena Velásquez Restrepo et al., 2019).

Las metodologías de desarrollo de software se clasifican en traicionales y ágiles, a continuación se presenta una tabla comparativa con características.

* + 1. **Metodologías ágiles vs tradicionales**

**Tabla 3**

*Metodologías tradicionales vs ágiles*

|  |  |
| --- | --- |
| **Metología Tradicional** | **Metodología Ágil** |
| **Enfoque predictivo**: Este enfoque planea el proyecto entero en un período extendido de tiempo. Este plan es la base para la construcción del sistema, describe sus funciones, los roles de cada miembro, el costo y el tiempo de desarrollo. | **Enfoque Adaptativo:** Este método está diseñado para aceptar los cambios en el desarrollo ya que las metodologías ágiles permiten los cambios en cualquier etapa del desarrollo porque se piensa que los cambios en los requisitos son el mejor objetivo para desarrollar un producto que satisfaga al usuario |
| **Documentación Comprehensiva:** En esta metodología, la documentación es clave para el desarrollo. Se asume que una documentación apropiada debe incluir: requisitos de usuario, requisitos de sistema, y la información necesaria para la codificación. | **Se equilibra flexibilidad y planeación:** A pesar de que la planificación es el factor más importante, planear para todo el proyecto en un solo momento no es el enfoque correcto, especialmente para proyectos que tienen a cambiar en el tiempo. La preparación para cortos períodos de tiempo es una estrategia válida o planear en una forma en la que rápidamente se puedan cambiar de decisiones o revertir las decisiones tomadas de acuerdo a la situación. |
| **Orientada a procesos:** Hay un proceso distinto para hacer todo el trabajo en el sistema, pero no está definido quien usará ya que cualquiera puede usarlo. Este proceso describe administradores, desarrolladores, analistas, testers, etc. Pero el procedimiento para estas tareas no es claro | **Orientada a personas:** Esta metodología considera a la gente de la organización como la mejor parte del desarrollo antes que cualquier proceso. Esta gente involucra: directores, desarrolladores, diseñadores, testers con habilidades altas. Si esta gente posee mencionadas habilidades se pueden adaptar rápidamente a los cambios e implementar cualquier proceso de desarrollo. |

Nota: Adaptado de (Shaikh & Abro, 2019)

* + - 1. **Ventajas y desventajas de metodologías ágiles vs tradicionales**

**Tabla 4**

*Ventajas y desventajas de metodologías tradicionales y ágiles*

|  |  |  |
| --- | --- | --- |
| **Metodología** | **Ventajas** | **Desventajas** |
| **Tradicional** | Orientada a proyectos de cualquier tamaño  Entregables específicos  Organización detallada  Adecuado para proyectos cortos y de baja complejidad | Se deben tener todos los requisitos desde el principio  Rígido y poco flexible al cambio  El producto se prueba al final  Metodología mas lenta y costosa, puede no satisfacer completamente las necesidades del cliente |
| **Ágil** | Metodología adecuada en entornos cambiantes  Mejora de software continua  Entregables funcionales en períodos cortos  Adecuado para proyectos tecnológicos y de innovación | Puede crear ambigüedad en el proceso si no se definen bien el alcance, las metas y los objetivos específicos.  Es indispensable contar con un equipo de gran experiencia o capacidad.  Esta metodología involucra un mayor esfuerzo por parte del cliente ya que este es el que da los requerimientos que pueden cambiar con el tiempo  Es complejo de implementar, se necesita la planificación a detalle de épicas, historias de usuario, cronogramas, roles, etc. Por lo que no es adecuado para grupos junior o que estén en formación. |

Nota: Adaptado de: (SAC, 2022), (Drew, 2019)

* + 1. **Metodología UWE**

UWE (UML-based Web Engineering) permite definir, modelar y documentar una aplicación web de una manera estandarizada a través del uso de UML (Unified Modeling Language) (Altamirano et al., 2019). En su implementación se contemplan 5 etapas las cuales se detallan a continuación:

**Análisis de requisitos:** Se recojen los requisitos funcionales a través de modelos de casos de uso

**Modelo de contenido:** Define el dominio de la aplicación mediante un diagrama de clases.

**Modelo de navegación:** Representa la navegación dentro de la aplicación mediante objetos y estructuras como son índices, menús y consultas.

**Modelo de representación:** Describe la estructura de la interacción del usuario con la aplicación web a través de las interfaces gráficas.

**Modelo de proceso:** Se describe los procesos principales que tiene la organización (Altamirano et al., 2019).

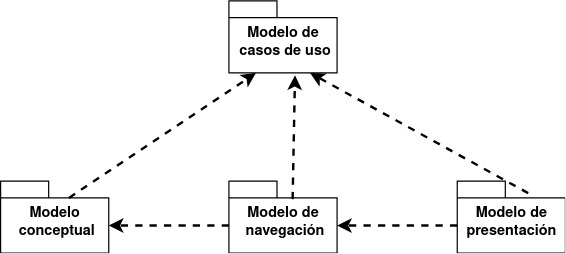
Es indispensable contar con una buena clasificación de requisitos de software. De esta manera podemos cubrir los aspectos mas relevantes del proyecto. A más de los requisitos funcionales, no funcionales y reglas de negocio mencionados anteriormente tenemos los requisitos de almacenamiento que define la información que se almacena en el sistema y como se relaciona entre si (Altamirano et al., 2019).

* + 1. **Proceso de la metodología UWE**

UWE nos provee de modelos para describir una aplicación web desde puntos de vista abstractos, cada uno de ellos se representa como paquetes UML. Sus cuatro procesos de diseño principales son: análisis de requisitos, diseño conceptual, diseño de navegación y diseño de presentación (Altamirano Rodríguez et al., 2019).

**Figura 4**

*Procesos de diseño UWE*



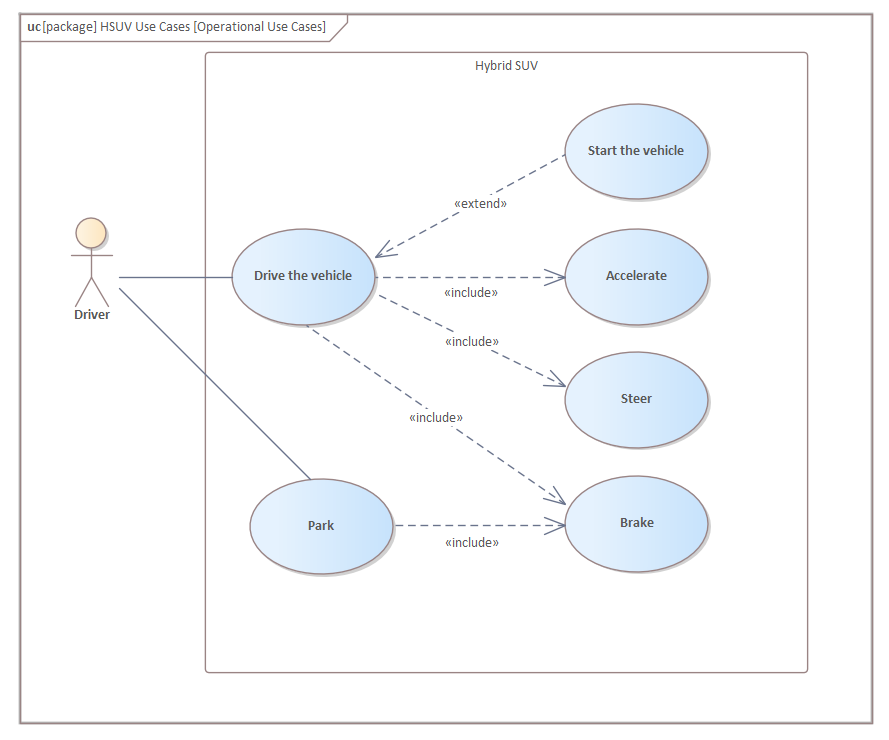
Nota: Adaptado de (Mubin et al., 2016)

* + 1. **Fases de UWE**

La primera fase es el la identificación de requisitos los cuales se especifican mediante el modelado de requerimientos funcionales, esto se logra modelando los casos de uso con un diagrama UML

**Figura 4**

*Diagrama UML de casos de uso (Pasar a arquitectura este diagrama)*

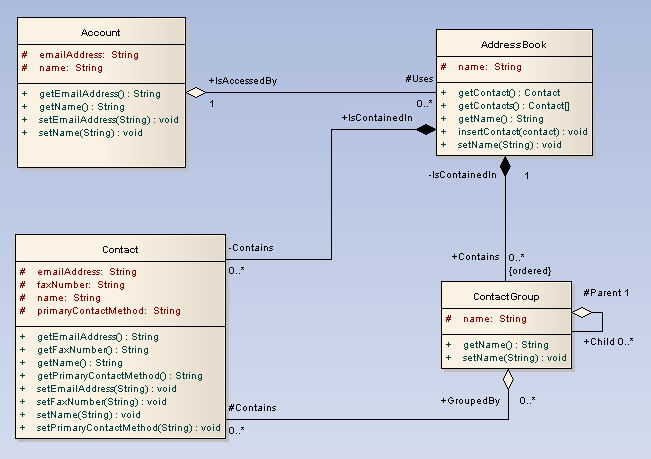


Nota: Obtenido de (Sparx Systems, s/f-b)

En la fase dos se elabora el modelo de contenido que proporciona una especificación visual mediante un diagrama UML de clases.

**Figura 5**

*Diagrama de clases UML*

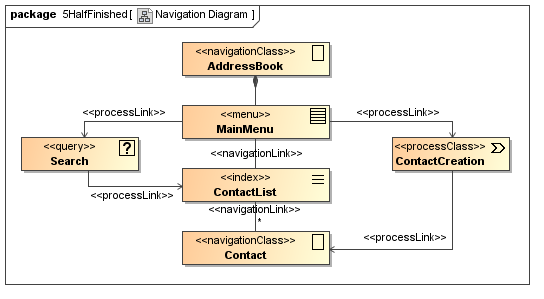


Nota: Obtenido de (Sparx Systems, s/f-a)

En la fase tres se elabora un diagrama de navegación cuyos elementos son los nodos y enlaces. Este diagrama viene dado por el análisis de requisitos y el modelo de contenido.

**Figura 6**

*Diagrama de navegación*

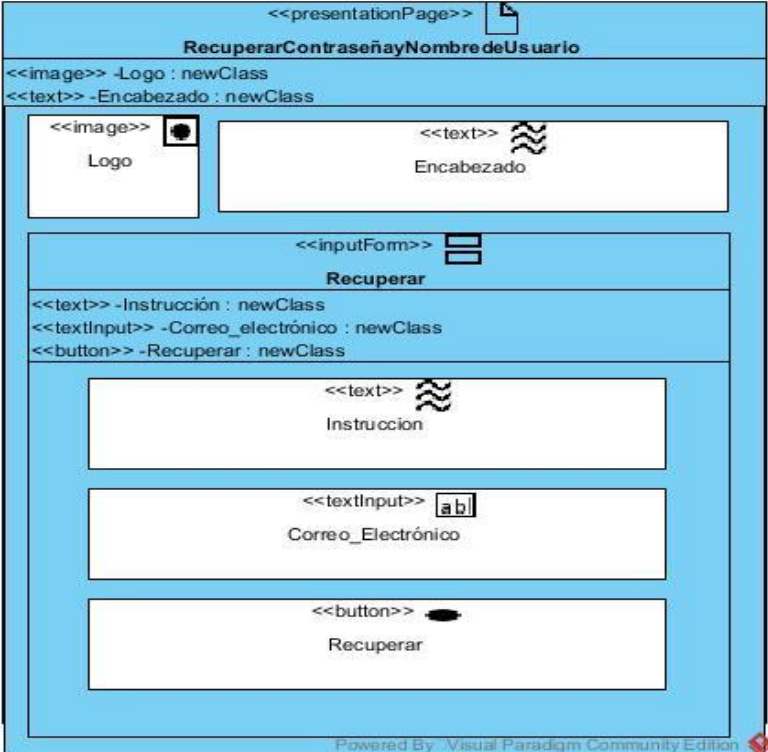


Nota: Obtenido de (Sparx Systems, s/f-c)

La fase cuatro se diseña un modelo de presentación que describe los elementos de la interfaz de usuario. Su diseño se basa en el modelo de navegación y el diseño UI (interfaz de usuario)

**Figura 6**

*Diagrama UML de presentación*



Nota: Obtenido de (Altamirano Rodríguez et al., 2019)

Finalmente, la fase cinco consta del modelo de procesos en el cual se representa a través de los diagramas de actividades las acciones, actores responsables de cada actividad y elementos de control de flujo (Altamirano Rodríguez et al., 2019).

* 1. **Arquitectura de Software**
     1. **Conceptos generales**

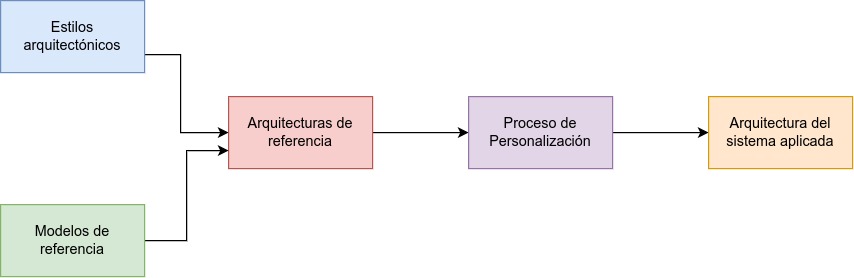
La arquitectura de software puede ser definida como el conjunto de estructuras que se necesitan para diseñar el sistema, las cuales componen los elementos de software, las relaciones entre estos, y las propiedades de ambos (Cruz-Benito et al., 2019).

Una arquitectura puede ser considerada como un grupo de elementos arquitectónicos los cuales pueden tener clases y paquetes UML los cuales están interconectados de una forma determinada (Capilla, 2022).

Existen arquitecturas de referencia que son diseños de software de alto nivel y describen las partes fundamentales de un conjunto de sistemas en un dominio particular. La construcción de arquitecturas de software se basan en los estilos arquitectónicos y modelos de referencia(Capilla, 2022).

**Figura 7**

*Composición de una arquitectura de referencia*



Nota: Autoría propia

Un patrón de arquitectura de software son aquellas reglas que determinan el contexto bajo el cual se llevará a cabo el desarrollo, cuya finalidad es la obtención de las características esperadas del software en cuestión. Cada patrón define un enfoque específico para el manejo de una característica en concreto (L.F. Santiago González et al., 2021). Son soluciones predeterminadas para un problema recurrente en un contexto determinado, se puede afirmar que son reutilizables ya que han sido probadas con anterioridad.

Los principios SOLID son un acrónimo introducido por Robert Cecile Martin y representan los cinco principios básicos del desarrollo orientado a objetos. Estos principios ayudan a los programadores a crear programas legibles y sin dificultad en su mantenimiento (AtSistemas, 2021).

Los principios SOLID son los siguientes:

**S: Single responsibility principle (Principio de responsabilidad única).** Este principio sostiene que una clase, microservicio o componente tenga asignada la responsabilidad de una única cosa. Esto es así porque si una clase tiene varias responsabilidades, al cambiar alguna de ellas afectaría a las demás. Se debe separar las responsabilidades de cada clase para que el programa sea más cohesivo y encapsulado (AtSistemas, 2021).

**O: Open/closed principle (Principio de abierto/cerrado).** Las clases que se usen deben estar abiertas para extenderse y cerradas para hacer cualquier modificación, esto nos permite que el desarrollo sea escalable (AtSistemas, 2021).

**L: Liskov substitution principle (Principio de sustitución de Liskov).** Este principio dice que una subclase debe ser substituible por una superclase. Baniassad (2018) menciona que existen dos restricciones sobre las precondiciones y postcondiciones: una subclase sólo puede ser sustituída por su superclase si sus condiciones previas no se fortalecen, y las condiciones posteriores de sus operaciones no se debilitan.

**I: Interface segregation principle (Principio de segregación de la interfaz).**  Los clientes no deben verse forzados a depender de interfaces que no usan ya que pueden verse afectados por cambios requeridos por otros clientes, esto permite evitar problemas de dependencias y damos cumplimiento al principio de responsabilidad única (AtSistemas, 2021).

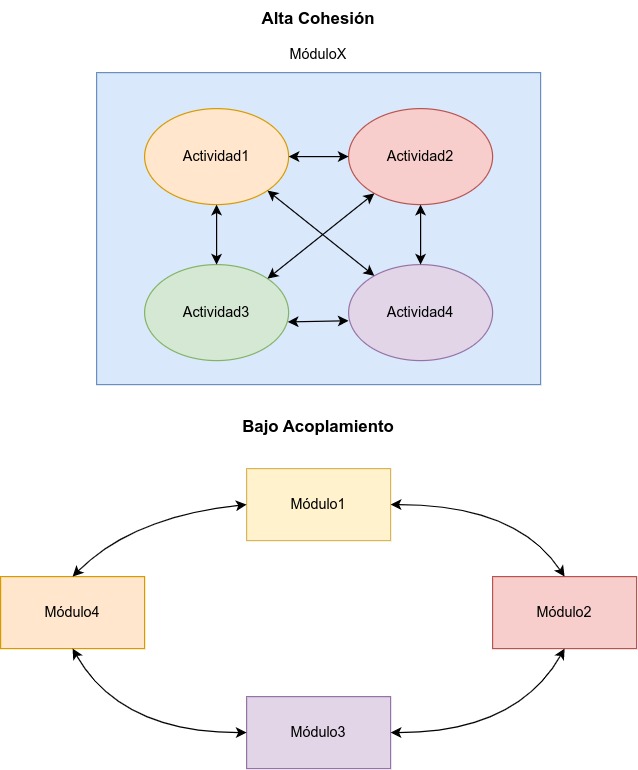
**D: Dependency inversión principle (Principio de inversión de dependencia).** Este principio nos permite reducir las dependencias entre los módulos de código para alcanzar un bajo acoplamiento de clases. Los módulos de alto nivel no deberían depender de los de bajo nivel sino que ambos deberían depender de abstracciones(AtSistemas, 2021).

La modularidad es la propiedad del software que permite dividir una aplicación en partes más pequeñas (módulos), cada una de las cuales debe ser independiente de los demás módulos y de la aplicación en sí (Barrera, 2020).

Hay ciertos aspectos que se deben tomar en cuenta durante la modularización de software. La cohesión hace referencia al grado en el que un módulo se dedica a realizar un solo tipo de tareas, y el acoplamiento es el grado en el que un módulo necesita comunicarse con otros módulos para realizar las tareas. Para lograr la modularización efectiva se necesita de una alta cohesión y un bajo acoplamiento (Gómez et al., 2019).

**Figura 8**

*Alta cohesión y bajo acoplamiento*



Nota: Autoría propia

* + 1. **Proceso de definición y diseño de la arquitectura de software**

**“**la arquitectura de un producto de software comprende los componentes del software, las propiedades visibles de dichos componentes y las relaciones que existen entre ellos**.”** (Blas et al., 2020).

El diseño de la arquitectura es una proceso en el que se establece la organización de los componentes del sistema para que satisfaga los requerimientos funcionales y no funcionales. Es necesario seleccionar los principios de trabajo adecuados para lograr un balance sobre las propiedades del software bajo diseño (Blas et al., 2020).

* + 1. **Arquitectura Lógica vs Arquitectura Física**

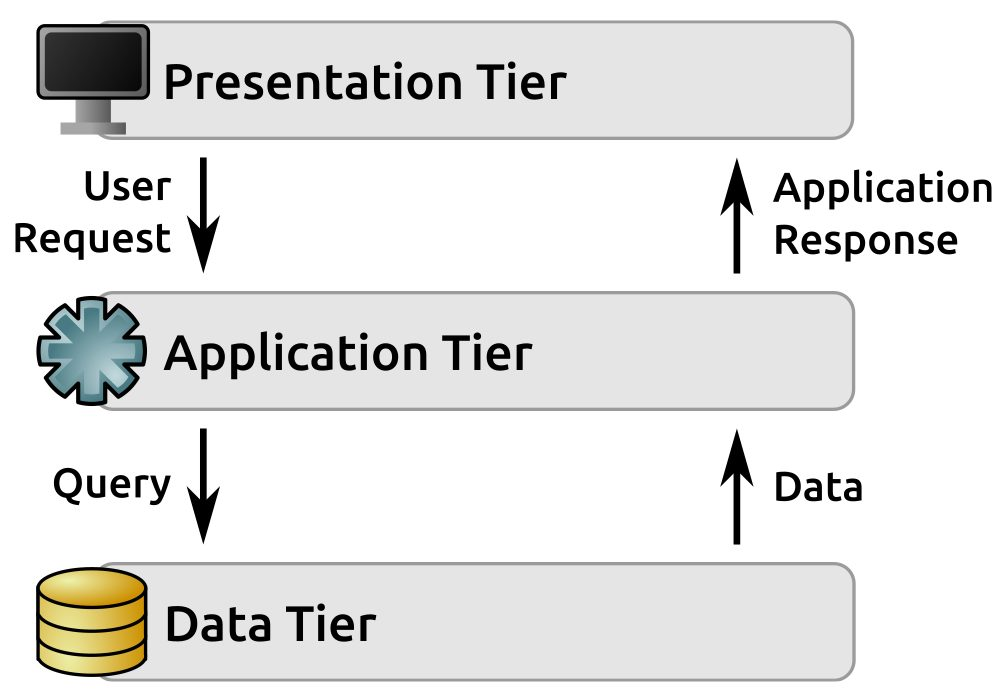
La arquitectura lógica es considerada un tipo de diseño estructural que es usado para proveer una descripción detallada del sistema sin definir el medio o la tecnología usada. Incluye toda la información en detalle sin usar terminología técnica en exceso (Pooja Gupta, 2022).

La arquitectura lógica comprende el diseño en general del producto de software, sus módulos, granularidad, responsabilidades, reuso, etc. Su propósito es guiar en la investigación y desarrollo del software (Alibaba, 2021).

El propósito principal de la arquitectura lógica es planear el desarrollos del sistema y usar como medio de comunicación entre los diseñadores, desarrolladores y el usuario( Gupta, 2022).

**Figura 9**

*Arquitectura lógica de software separada en tres capas*

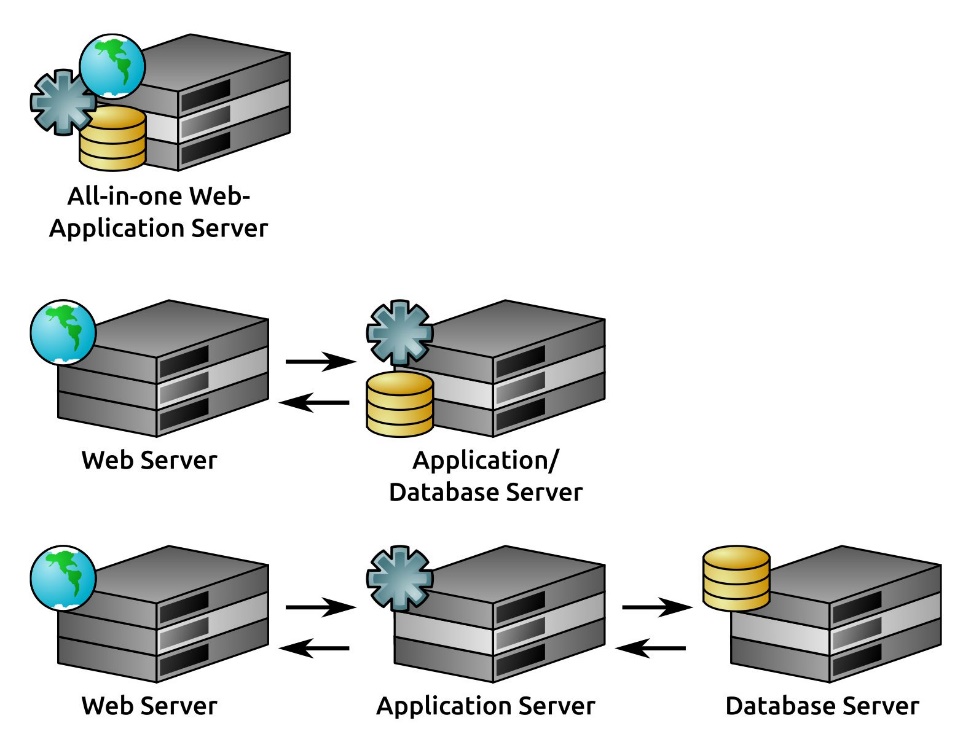


Nota: Obtenido de (TechTarget, 2018)

Mientras que la arquitectura lógica identifica los elementos funcionales del sistema, la arquitectura física especifica los dispositivos en donde estos elementos funcionarán, determina en qué computadores o servidores correrán los componentes de software (PacktPublishing, 2018).

**Figura 10**

*Arquitectura Física separada en una, dos y tres capas*



Nota: Obtenido de (PacktPublishing, 2018)

* + 1. **Descripción del patrón de software seleccionado**

Los patrones de software tienen como objetivo compartir una solución probada a un problema de diseño en particular de una manera estandarizada y fácilmente reusable, los patrones deben ser capaces de proporcionar las siguientes piezas de información: nombre, contexto, problema, solución y consecuencias. (Guamán, 2021).

El modelo definido para estre trabajo es Modelo-Vista-Controlador (MVC), creado en 1979 por Trygve Reenskaug, es un patrón arquitectónico que se basa en la separación del código en tres capas diferentes, autónomas, y cada una con una responsabilidad definida (Guamán, 2021).

MVC propone la construcción de tres componentes distintos: modelo, vista, controlador. Por un lado, define componentes para la representación de la información y por otro lado para la interacción del usuario. De esta manera se permite una fácil reutilización de código y separación de conceptos mediante el cual se separa el modelo (lógica del negocio), las vistas (la presentación) y controlador (capa de datos) (García Sosa, 2021).

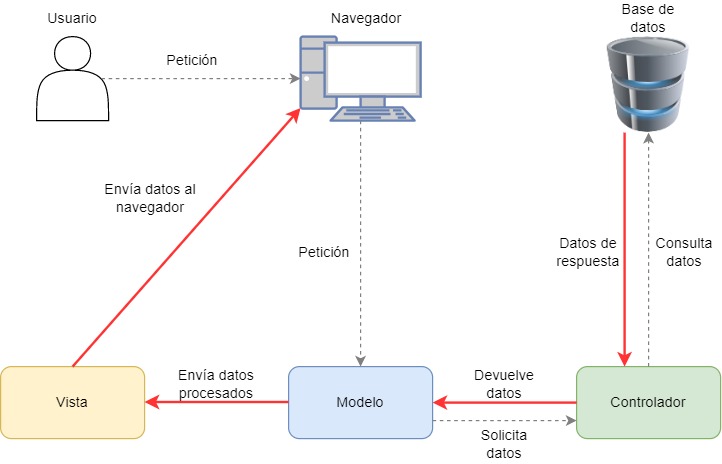
**Modelo:** Es el que se encarga de ser el proveedor de recursos tanto para la vista como el controlador. Esta es la representación de los datos y reglas de negocio, maneja un registro de las vistas y de los controladores que existe en el sistema.

**Vista:** Permite mostrar la información del modelo en un formato que permita la interacción, en nuestro caso es la parte front-end del desarrollo.

**Controlador:** Interpreta las entradas a nivel de hardware, informando al modelo y/o la vista. Responde a los eventos provocados por el usuario como por ejemplo: se de clic, se digite un texto, se marque una casilla, etc. Esto implica cambios en el modelo y la vista (García Sosa, 2021).

**Figura 11**

*Patrón arquitectónico modelo-vista-controlador en un entorno web*



Nota: Autoría propia

(info sobre mvc) <https://www.youtube.com/watch?v=a3nE8neOwHM&t=12s>

(para tener mas claro como hacer la pagina <https://www.youtube.com/watch?v=ExAs5ySWqGg>)

* 1. **Herramientas y tecnologías del desarrollo web**
     1. **Tecnologías a nivel de Back-end**

Las tecnologías back-end se enfocan en los aspectos del lado del servidor de la aplicación web. El código back-end permite la comunicación entre el navegador web y las bases de datos. El desarrollo back-end conjuntamente con el desarrollo front-end nos permite entregar a los usuarios una experiencia útil e interactiva (Dutonde et al., 2022).

**PHP** es un lenguaje de scripting código abierto que se ejecuta del lado del servidor el cual nos será útil para realizar operaciones CRUD (Create, Read, Update, Delete) en la base de datos (PHP, s/f).

* + 1. **Tecnologías a nivel de front-end**

**Hypertext Markup Language** (HTML) Es el lenguaje de etiquetado estándar el cual está diseñado para mostrarse en un navegador web. El uso de este lenguaje nos servirá como los bloques de construcción de nuestra página web. En nuestro caso usaremos HTML5 el cual ofrece mejores servicios semánticos y multimedia que en las versiones pasadas (Tabarés, 2021).

**Cascade Style Sheet** (CSS) Nos permiten aplicar estilos (colores, formas, imágenes, etc.) a uno o varias páginas diseñadas previamente de forma automática y masiva. Se le denomina estilos en cascada porque se lee, procesa y aplica el código desde arriba hacia abajo, y las cuales siguen una serie de normas para resolver problema de ambigüedad. Enlazaremos el archivo CSS a los diferentes archivos HTML que tengamos (Román, s/f).

**Javascript** es un lenguaje de programación que permite implementar funciones complejas en páginas web como es el contenido de actualización dinámica, control de multimedia, animación de imágenes, etc. El código dinámico se refiere a la capacidad de actualizar la visualización de una página web para mostrar diferentes cosas en diferentes circunstancias. El código de lado del servidor como es PHP en nuestro proyecto genera dinámicamente nuevo contenido en el servidor como extraer los datos de una base de datos, mientras que JavaScript de lado del cliente genera dinámicamente nuevo contenido dentro del navegador del cliente (Mozilla, 2023).

**Bootstrap** es un framework para la creación de sitios web. Nos permite un desarrollo responsivo y adaptable a los dispositivos móviles ya que detecta el tamaño y orientación de pantalla y adapta automáticamente la página (Zola, 2022).

* + 1. **Gestor de base de datos**

El sistema de administración de base de datos **MySQL** es un proyecto de código abierto que es desarrollado y distribuido por la corporación Oracle. Esta base de datos es relacional, es decir, los datos se guardan en tablas, vistas, filas y columnas. Esto nos permite crear relaciones entre los campos de datos como uno a uno, uno a muchos, campo único, requerido u opcional, y punteros entre las tablas. De esta manera se evita la inconsistencia, duplicidad, orfandad, desactualización o desaparición de los datos (MySQL, s/f).

* + 1. **Entorno de desarrollo**

**Visual Studio Code** es un editor de código fuente independiente que se ejecuta en Windows, macOS y Linux. La elección principal para desarrolladores web y JavaScript, con extensiones para admitir casi cualquier lenguaje de programación. Nos provee de extensiones para las diferentes tecnologías usadas en el desarrollo del proyecto como intellisense, color en la sintaxis del código, autocompletado, etc (Microsoft, 2023).

* + 1. **Herramientas adicionales para el desarrollo**

**Git** es una plataforma de código abierto para el control de versiones distribuido creada por Linus Torvalds en el año 2005. Git también facilit el registro y comparación de versiones de un archivo, nos permite saber qué cambió, quién lo cambió o quién ha iniciado una propuesta (Castellanos, 2021).

**GitHub** is una plataforma basada en la web para el control versiones y colaboración donde los usuarios pueden alojar sus repositorios Git. Es una plataforma que puede mantener repositorios de código en la nube para que varios desarrolladores puedan trabajar en un solo proyecto y ver ediciones en tiempo real (Castellanos, 2021).

* 1. **Trabajos relacionados**

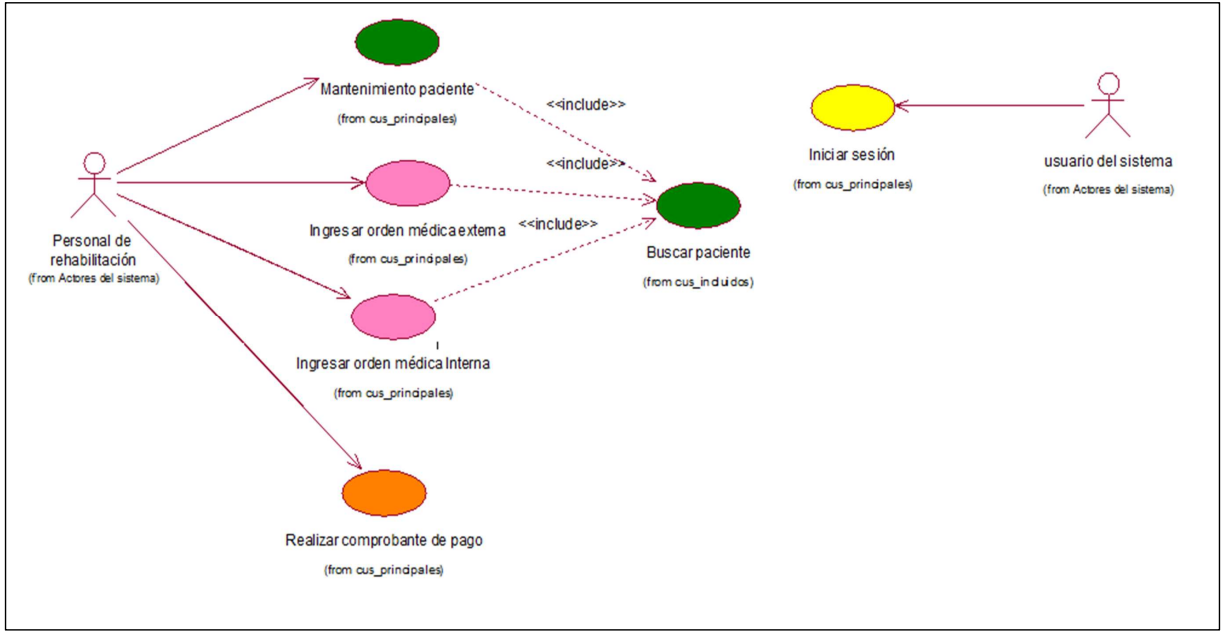
En la tesis de Alex Briones de la Universidad Peruana de las Américas llamada “Implementación de un Sistema Web para la Gestión Administrativa en el Centro de Terapia Física y Rehabilitación de la Clínica Good Hope, Miraflores 2018”, se menciona que el centro no contaba con un sistema que pueda optimizar la gestión de los procesos de rehabilitación. Parte de la problemática era que muchos procesos, como la elaboración de la cita médica, la elaboración de los cronogramas y los expedientes físicos se elaboraban de forma normal, por lo que esto ofrecía poca seguridad ya que se podía extraviar la información confidencial del paciente.

El desarrollo usó lenguaje UML para la maquetación del proceso administrativo de la gestión de rehabilitación, diagrama de casos de uso del sistema y modelo de análisis del sistema, usó Java Server Page y la base de datos Oracle.

El diagrama de casos de uso separa al usuario del sistema el cual es el usuario que puede ingresar al sistema mediante nombre de usuario y contraseña, y el personal de rehabilitación el cual usa los módulos del que consta el sistema para poder administrar las órdenes médicas y la búsqueda de los pacientes (Briones, 2018):

**Figura 12**

*Modelo de caso de uso sistema para la clínica Good Hope*

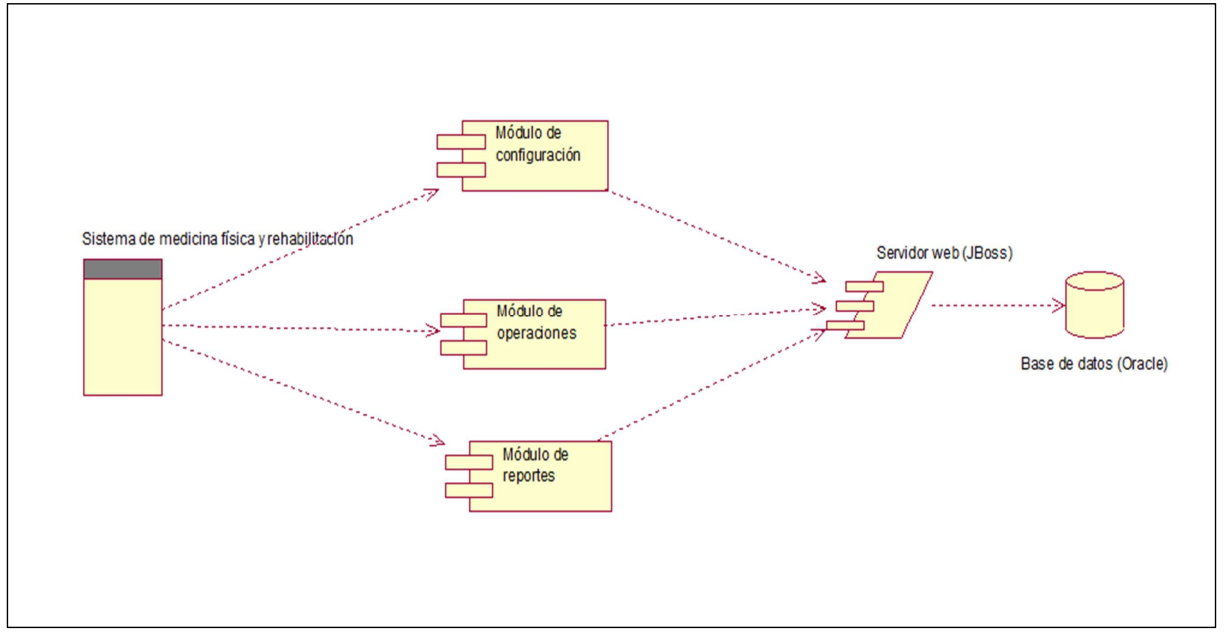


Nota: Obtenido de: (Briones, 2018)

El sistema engloba a nivel integral la administración de la clínica, este sistema no solamente se encarga de la administración de las citas médicas, sino también de los comprobantes de pago y los cronogramas de las citas para evitar que éstas se crucen. Los componentes del sistema se basan en tres módulos: configuración, operaciones, reportes.

**Figura 13**

*Diagrama de componentes sistema de la clínica Good Hope*



Nota: Obtenido de (Briones, 2018)

El back-end de este desarrollo web es diferente al de este proyecto, para la comunicación de la base de datos Briones usa JBoss debido a que el desarrollo de su proyecto está pensado en el ambiente Java, en el caso de el desarrollo para la empresa GYMES se usará PHP para la comunicación con la base de datos MySQL.

En lo que respecta a interfaces de usuario, la orden de servicio de medicina y rehabilitación es en donde se guardan los datos de la cita, consta de los datos personales del paciente, la historia clínica, el tipo de tratamiento al cual se va a someter y observaciones generales. Esta pantalla es parecida a la forma de administración de citas que se usa en el centro medico GYMES:

**Figura 14**

*Interfaz web orden médica rehabilitación*



Nota: Obtenido de (Briones, 2018)

En el trabajo “Desarrollo de un Sistema Web para la gestión del historial clínico y terapias para el Instituto de Medicina del Deporte y Ortopedia, IMDO” se propuso el diseño de un aplicativo web de gestión de historial clínico y terapias. El sistema consta de seis módulos: configuración, registro, seguridad, servicios, consultas y bioseguridad.

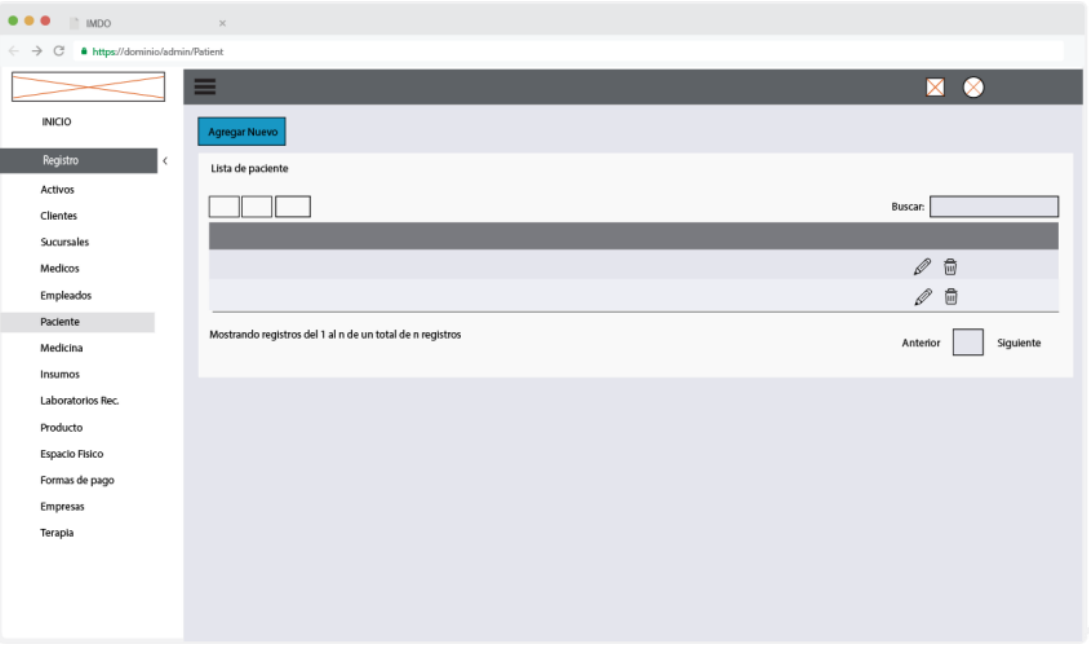
La aplicación cuenta con una pantalla de login la cual redirige a la pantalla principal, en esta, se cuenta con varias opciones para agregar datos no solo del cliente, sino que también del tipo de tratamiento ofertado, sucursales, tipo de empleado, especialidad, e incluso hábitos del paciente.

El programa abarca muchas de las funciones que cuenta la empresa, pretende ser una solución integral para el instituto y sus sucursales. Este sistema es altamente personalizable y no solamente se limita al servicio de fisioterapia, sino contiene elementos de triaje como toma de signos vitales y venta de productos. Su base de datos es amplia y consta de mas de 40 tablas, utiliza el framework Laravel y una base de datos PostgreSQL(Piñero Villena & Reyes Reyes, 2021).

Para destacar la lista de perfil de los pacientes consta de una barra de búsqueda, en la cuales se dispone de los botones ver detalle y editar el perfil del paciente:

**Figura 15**

*Ventana de lista de perfil de pacientes sistema IMDO*



Nota: Obtenido de (Piñero Villena & Reyes Reyes, 2021)

En el Trabajo de Titulación de Alvarado y López (2022) titulado “Implementación de un sistema web para la gestión administrativa y un aplicativo móvil informativo para el centro de rehabilitación FisiosaludXP”, se aborda el problema de un creciente número de pacientes y la necesidad de mejorar los tiempo de respuesta en los procesos administrativos que son llevados de forma manual.

En este trabajo se proponen varios módulos:

Administrativo, cuyo objetivo es administrar personal, administrar pacientes y administrar las citas programadas.

Gestión, el cual contiene agenda de citas, jornada de trabajo de los médicos y estado de pacientes.

Seguridad, para definir el nivel de privilegio de los usuarios.

Inventario, el cual registra los equipos técnicos e insumos.

Reportes, proporciona información sobre los módulos que componen al sistema.

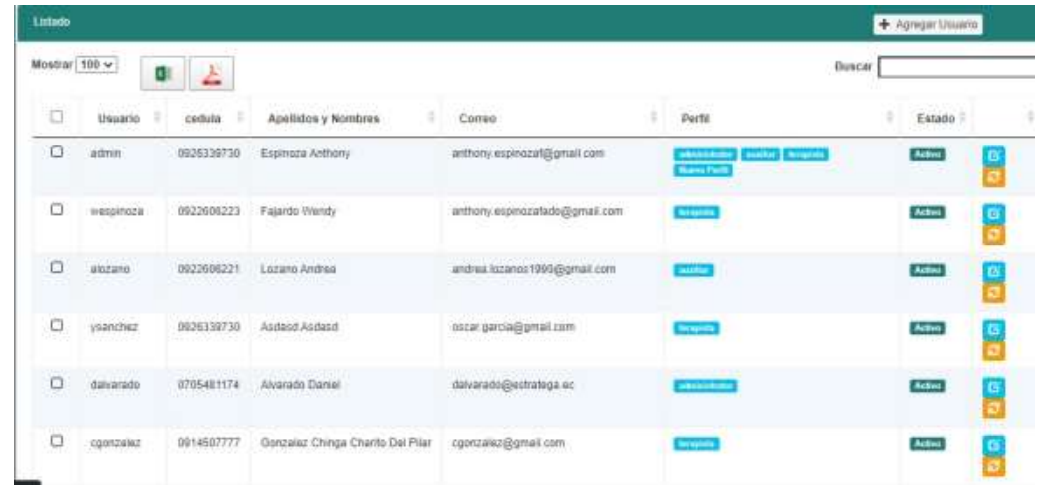
Este trabajo también consta de un aplicativo móvil diseñado para Android el cual muestra información sobre datos personales de los pacientes, citas programadas e información de la empresa.

Las tecnologías usadas son parecidas a las que usaremos en este trabajo, cabe destacar que se usa Java como lenguaje nativo para la aplicación Android.

En los anexos constan los casos de uso detallados para cada componente del sistema, así como sus diagramas de secuencia y diagrama de clases del sistema en general. En el apartado de interfaz gráfica se presenta una barra de búsqueda con la tabla de datos personales de cada usuario. En la parte derecha se cuenta con botones para actualizar o eliminar usuarios, así como una barra de estado:

**Figura 16**

*Pantalla usuarios del sistema*



Nota: Obtenido de (Alvarado Tintin & López Ibarra, 2022)

En el proyecto de grado “Sistema Web para la administración y seguimiento del tratamiento de fisioterapia y kinesiología integral. Caso: PyhisioActive” de Josías Calderón (2020), se intenta solucionar problemas como la duplicidad de citas o la no disponibilidad de profesionales para la atención a pacientes que solicitaron el servicio previamente.

Mediante el sistema web desarrollado, permite la automatización de los procesos de historias clínicas, agendamiento de citas, registro de tratamientos y evolución del paciente. El sistema cuenta con varios casos de uso para la administración de usuarios, sitio web, pacientes y tratamientos.

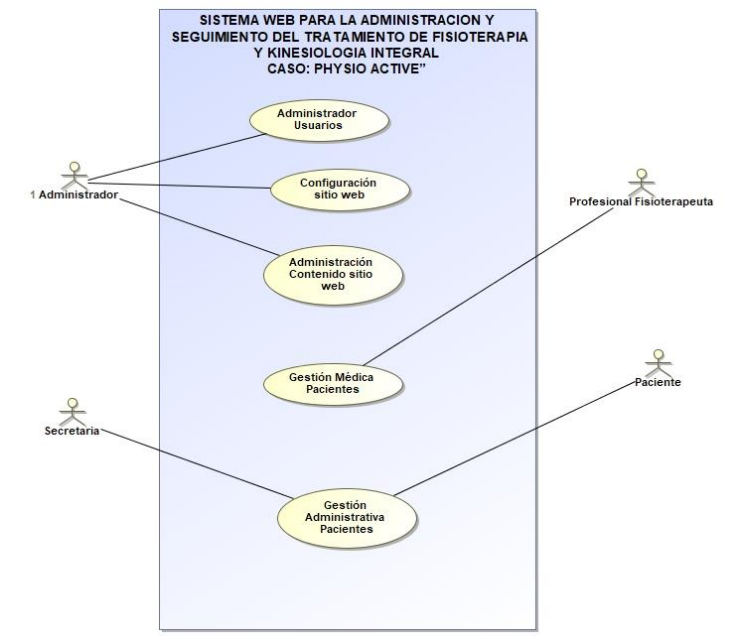
Utiliza la metodología UWE junto a un Proceso Unificado Ágil (AUP), lo que le permite realizar entregas pequeñas y funcionales de software, mejorando la interacción entre desarrolladores y clientes.

Su trabajo está bien delimitado y se compone de los siguientes módulos: Agenda, Gestión de pacientes, historia clínica, reportes, gestión de usuarios, control de pagos, marketing, administrador de sistema web, multimedia, variables y blog.

Lo destacable es su diagrama de casos de uso del sistema general donde se pueden apreciar cuatro actores: Administrador, Profesional Fisoterapeuta, Paciente, Secretaria:

**Figura 17**

*Diagrama de casos de uso sistema PhyisioActive*



Obtenido de: (Calderón, 2020)

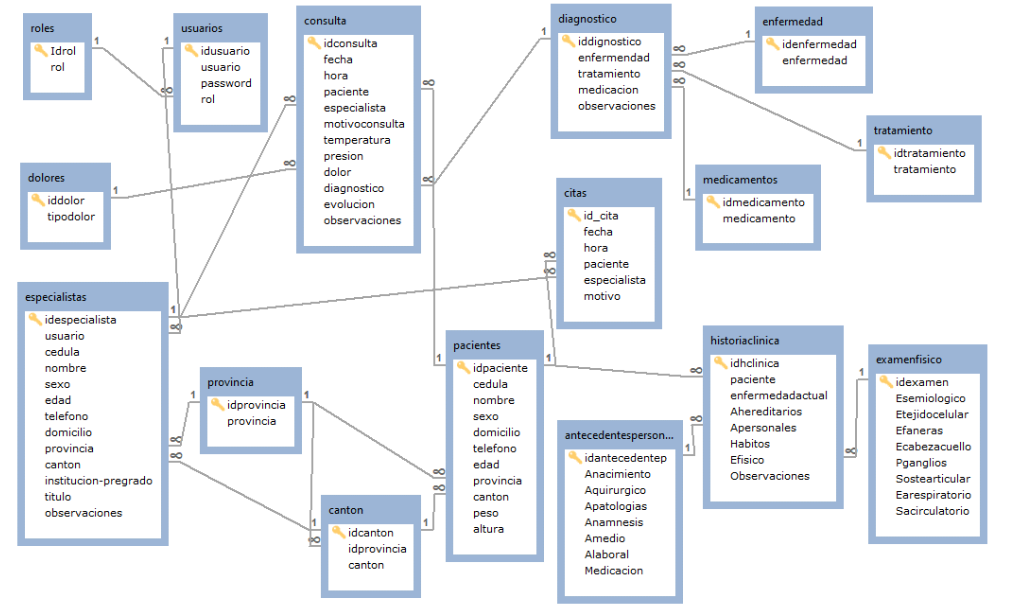
En el proyecto de examen complexivo llamado “Análisis del diseño de un sistema informático para la gestión administrativa en el consultorio de fisioterapia y rehabilitación AMP de la ciudad de Babahoyo” de la estudiante Dayana Ortiz (2022), se tiene el fin de mejorar la gestión administrativa del centro. El aporte es que se podrá realizar los procesos en menor tiempo mediante un sistema que asegure la confiabilidad de información y el fácil manejo de la gestión administrativa.

El proyecto usa metodología UML y utiliza diagramas de caso de uso y diagrama de clases. Hace un análisis de factibilidad técnica para determinar si el consultorio cuenta con la infraestructura tecnológica y de comunicaciones llegando a la conclusión que lo mas costoso en el despliegue del sistema es una computadora portátil con 8Gb de memoria RAM y 128Gb de almacenamiento SSD, ya que las tecnologías que usó para la programación son de código abierto. El sistema no prevee un despliegue en un web hosting o VPS y no responde a inquietudes como qué pasaría si la laptop usada sufriera daños de hardware o software ya que no está estipulado en el alcance del proyecto. Hace uso de tecnologías como Visual Basic, .Net, C# y la base de datos MySQL.

Lo destacable del proyecto es su diseño de base de datos, este incluye tablas como historia clínica, antecedentes personales, medicamentos,etc. Para construir una mejor historia clínica del paciente.

**Figura 18**

*Modelo de base de datos sistema consultorio AMP*



Nota: Obtenido de (Ortiz Sánchez, 2022)

* 1. **Tabla comparativa trabajos relacionados**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Trabajo** | **Problemática** | **Objetivos** | **Alcance** | **Metodología** | **Tecnologías** | **Conclusiones** |
| Implementación de un Sistema Web para la Gestión Administrativa en el Centro de Terapia Física y Rehabilitación de la Clínica Good Hope, Miraflores 2018. | ¿De qué manera la implementación de un Sistema Web optimiza la gestión del proceso de  rehabilitación de la Clínica Good Hope? | Desarrollar la Implementación de un Sistema Web para optimizar la Gestión del proceso de emisión de órdenes médicas, cronograma de citas, control de sesiones y pagos de los pacientes de la Clínica Good Hope | Implementación de un sistema web para optimizar la gestión del proceso de emisión de órdenes médicas de rehabilitación para los pacientes de la CGH.  Implementación de un sistema web para optimizar la Gestión del proceso de cronograma de citas de los pacientes de la CGH.  Implementación de un sistema web para optimizar la gestión del proceso de control de sesiones y pagos de los pacientes de la CGH. | **Metodología de desarrollo:** se aplicará la metodología RUP para la ilustración de los procesos en acción y el lenguaje de programación Java Server Page (JSP), se dividirá en las siguientes fases: Inicio, Elaboración, Construcción y transición.  **Metodología de Investigación:** Se realiza un enfoque cuantitativo de investigación, los planteamientos son delimitados desde el inicio del estudio. La hipótesis a probar es que “La implementación de un Sistema Web optimiza la Gestión del Proceso de Rehabilitación de la Clínica Good Hope.” | Java Server Page (JSP)  Base de datos Oracle | Se demuestra que el sistema mejora de una manera significativa a todos los problemas que se encontraban, y mejora de manera significativa los procesos que se realizaban de manera manual.  El sistema pretende ser una solución integral, por lo que la capacitación al personal en el uso es imprescindible. |
| Desarrollo de un Sistema Web para la gestión del historial clínico y terapias para el Instituto de Medicina del Deporte y Ortopedia, IMDO | Gestión de turnos manejado por la recepcionista a través de Google Calendar, las fichas médicas, recetas y consultas eran registraras a través de formularios físicos. | Desarrollar un sistema web para la gestión del historial clínico y terapias para el instituto médico del deporte y ortopedia IMDO | Consta de seis módulos: Configuración, Registro, Seguridad, Servicios, Consultas y Bioseguridad. Pretende ser una solución integral para la empresa y sus sucursales. | **Metodología de desarrollo:** Utiliza el desarrollo en cascada el cual es un procedimiento lineal que se caracteriza por dividir los procesos de desarrollo en fases sucesivas del proyecto.  **Metodología de investigación:** Se utilizaron métodos cuantitativos y cualitativos como la observación, la entrevista, etc. Aunque no deja constancia si el prototipo satisfizo las necesidades de los usuarios y del centro. Deja en recomendaciones que se requiere la capacitación en el uso del sistema por parte de los empleados. | Servidor HTTP apache  PostgreSQL  PHP, Laravel  RESTFUL  Bootstrap  jQuery  Javascript | Debido a que no hay constancia de uso y exámenes de usabilidad por parte de los usuarios, no se puede llegar a la conclusión si las necesidades de los usuarios fueron satisfechas. El autor en sus conclusiones describe lo que hace la aplicación mas no la conformidad del usuario. |
| Implementación de un sistema web para la gestión administrativa y un aplicativo móvil informativo para el centro de rehabilitación FisiosaludXP. | Los registros de mas de 6000 atenciones de pacientes eran llevados en cartillas físicas y mediante hojas de cálculo. Muchas veces las fichas son extraviadas y no se cuenta con ningún respaldo. La pregunta que responde el trabajo es: ¿Al implementar un sistema que procesos administrativos se van a automatizar para agilizar el tiempo de respuesta y reducir costos en el centro de rehabilitación física FISIOSALUDXP? | Implementar un sistema web administrativo y un aplicativo móvil informativo, mediante códigos de programación web y móvil de software libre para la gestión operativa-administrativa de la empresa FISIOSALUDXP. | El sistema cuenta con los siguientes módulos: Administrativo: Administra personal, pacientes y citas.  Gestión: Agenda, jornada de trabajo, información pacientes.  Seguridad: perfil de usuarios y privilegios.  Inventario: Registra equipos técnicos e insumos.  Reportes: información sobre los módulos. | **Metodología de desarrollo:** Metodología RUP, consiste en asignar actividades a los integrantes del proyecto con el propósito de construir un software de calidad.  **Metodología de Investigación:** Investigación no experimental, ya que se basa en la observación y el análisis sin que el investigador altere el objeto de investigación. | JavaScript  MySQL  XAMPP  jQuery  Bootstrap  Android Studio | Se concluye que se logró resolver los inconvenientes que presentaban los procesos de gestión administrativa. Se realizan sugerencias para mejorar la atención al cliente como opción de pago con tarjeta, incorporación de gestión de informes, mejoramiento de aplicación móvil y agregar un módulo de chat para mantener el contacto con los clientes. |
| Sistema Web para la Administración y Seguimiento del Tratamiento de Fisioterapia y Kinesiología Integral Caso: PhysioActive | Se ocasiona duplicidad de citas debido a que no se cuenta con un agendamiento adecuado. Los registros se los realiza de forma manual por lo que el seguimiento a los pacientes se torna complicado. La información sobre pagos también se encuentra dispersa. | Desarrollar un “Sistema Web para la administración y seguimiento del tratamiento de Fisioterapia y Kinesiología Integral CASO: Physio Active.” que permita la automatización de los procesos descritos anteriormente. | El Sistema Web no emitirá facturación electrónica.  El Sistema Web no registrará firma digital de los pacientes.  El Sistema Web no realizara gestion de contabilidad.  El Sistema Web no contempla módulos de inventarios y activos fijos.  El Sistema Web únicamente constará de los módulos descritos anteriormente. | **Metodología de desarrollo:** Metodología Ágil, Proceso Unificado Ágil (AUP).  Modelos de UWE para el análisis y diseño de la solución.  **Metodología de investigación:** Investigación descriptiva y aplicada. | Go  Vue.js  PostgreSQL  Webpack | Los objetivos específicos han sido satisfechos con la implantación del sistema web. La metodología ágil AUP permitió desarrollar el software de una manera simple y fácil de entender. |
| Análisis del diseño de un sistema informático para la gestión administrativa en el consultorio de fisioterapia y rehabilitación AMP de la ciudad de Babahoyo. | Las actividades en el consultorio se planean y registran a través de una lista en una libreta, desencadenando problemas como servicios ineficientes; consultas lentas; pérdida, desactualización y repetición de la información, además del crecimiento limitado; desperdicio de recursos y acumulación de papel, volviendo poco sostenible el cuidado del medio ambiente; | Realizar un análisis de diseño de un sistema informático para mejorar la gestión administrativa en el consultorio de fisioterapia y rehabilitación AMP de la ciudad de Babahoyo. | El trabajo diseña el prototipo del sistema informático y presenta la factibilidad de implementación del mismo, mas no lo desarrolla y despliega. | **Metodología de desarrollo:** Utiliza la metodología UML para el diseño de componentes.  **Metodología de investigación:** Cualitativas como la entrevista y el cuestionario | C#  MySQL | En base al análisis de factibilidad presentado se puede concluir que la viabilidad del sistema es positiva.  Conociendo la aportación y el beneficio que brindará la solución, se espera, que el dueño del consultorio busque el financiamiento necesario para la puesta en marcha del sistema informático. |

# Capítulo Tres

# Aplicación de los fundamentos teóricos y metodología UWE

**Análisis de Requerimientos**

**Requerimientos funcionales**

|  |  |
| --- | --- |
| **Número** | **Requerimiento** |
| RF-01 | Mantener el registro de los datos personales del paciente |
| RF-02 | Mantener los datos médicos de los pacientes como forma de historial clínico |
| RF-03 | Mantener un registro de la alimentación recomendada a los pacientes |
| RF-04 | Mantener un registro de medidas y peso de las diferentes áreas del cuerpo |
| RF-05 | Mantener una constancia del tratamiento a realizar |
| RF-06 | Implementar un certificado SSL para garantizar la autenticidad del sitio web |
| RF-07 | Implementar un login para el inicio de sesión de usuario sea este: administrador, doctor o paciente |
| RF-08 |  |
|  |  |

**Requerimientos no funcionales**

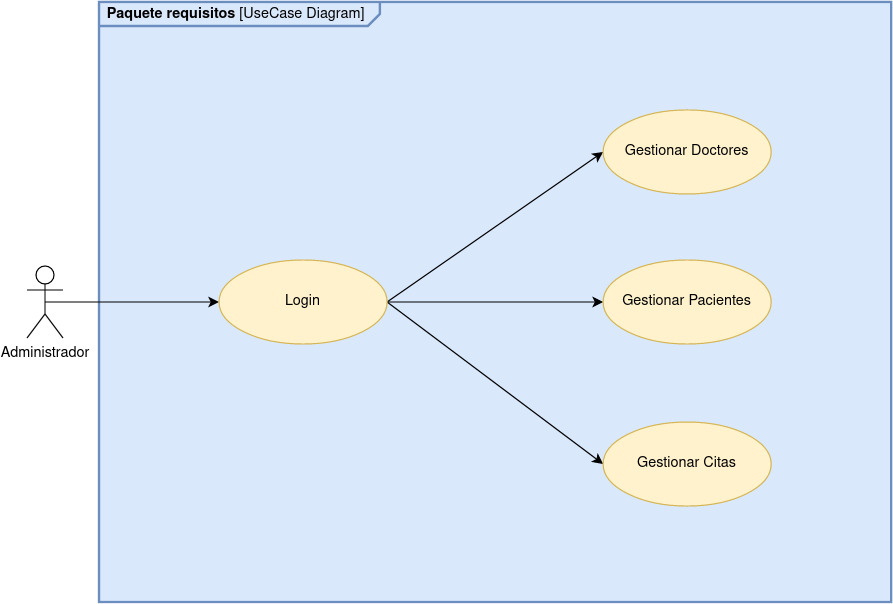
|  |  |
| --- | --- |
| **Número** | **Requerimiento** |
| RNF-01 | La interfaz del sistema debe presentar un diseño responsivo, debe ser compatible con los diferentes tamaños de pantalla |
| RNF-02 | El paciente debe asegurarse de que es seguro entrar en el sitio web |
| RNF-03 | La interfaz debe contar con un diseño agradable y entendible tomando en cuenta principios de usabilidad y experiencia de usuario |
| RNF-04 |  |
|  |  |
|  |  |

**Modelado de Requerimientos**

La primera etapa para desarrollar un sistema web es la identificación de requisitos. UWE provee de dos niveles de granularidad, el primero es una descripción breve de las funcionalidades el cual se modela a través de un diagrama de casos de uso UML.

**Figura 19**

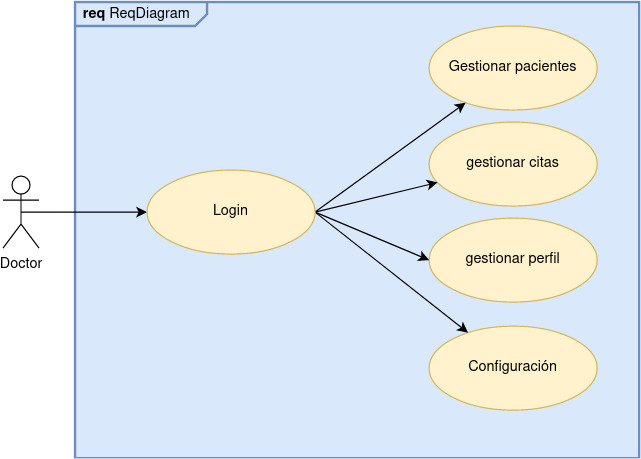
*Diagrama de casos de uso administrador*



Nota: Autoría propia

**Figura 20**

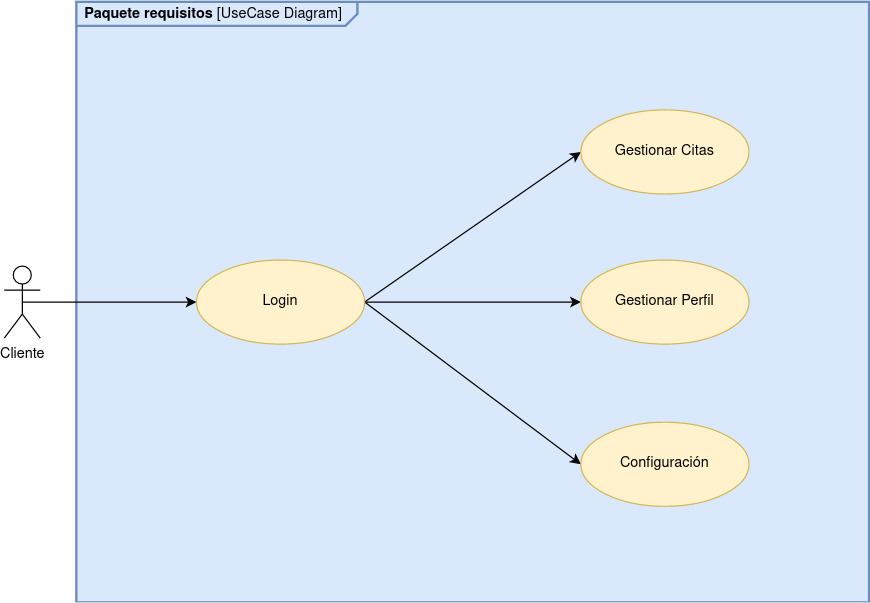
*Diagrama de casos de uso Doctor*



Nota: Autoría Propia

**Figura 21**

*Caso de uso paciente*

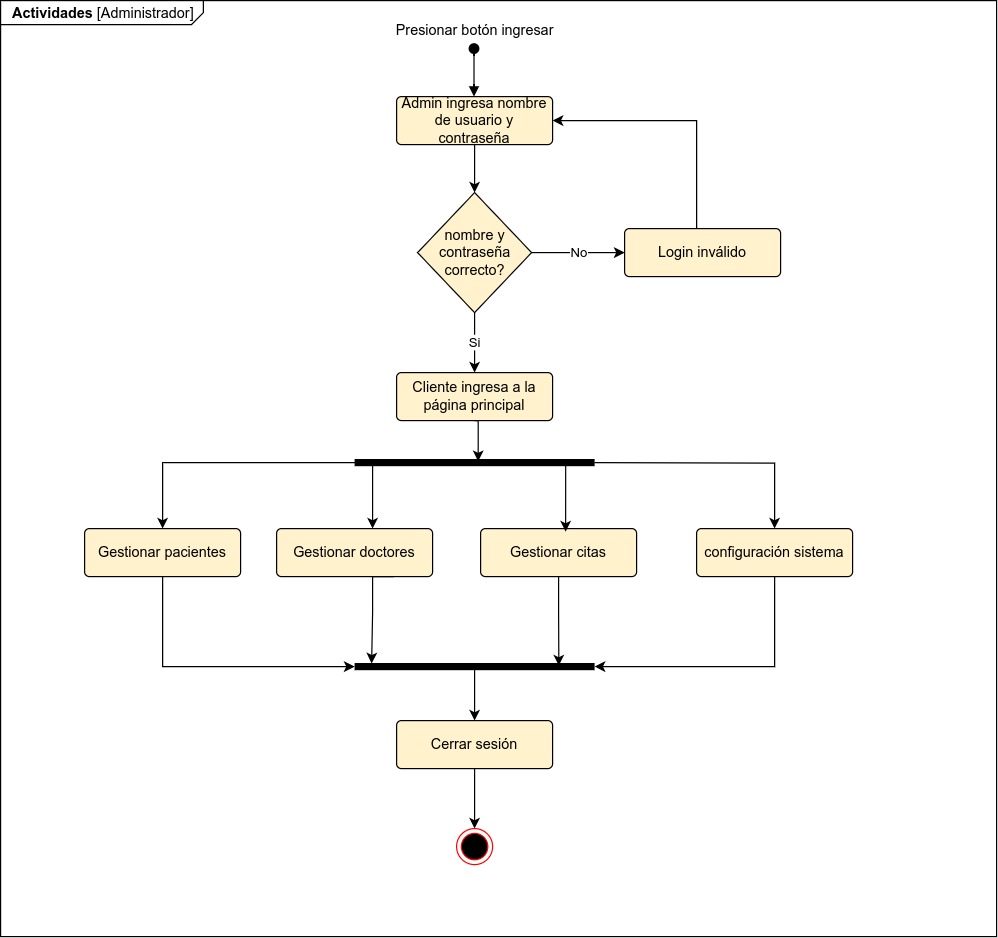


Nota: Autoría propia.

El diagrama de actividades es un tipo de diagrama que se utiliza para modelar el flujo de trabajo o el comportamiento de un sistema o proceso de negocio. Es útil para representar procesos, procedimientos y algoritmos complejos en una forma visual fácil de entender, representa un flujo de trabajo que va desde un punto inicial hasta un punto final.

**Figura 22**

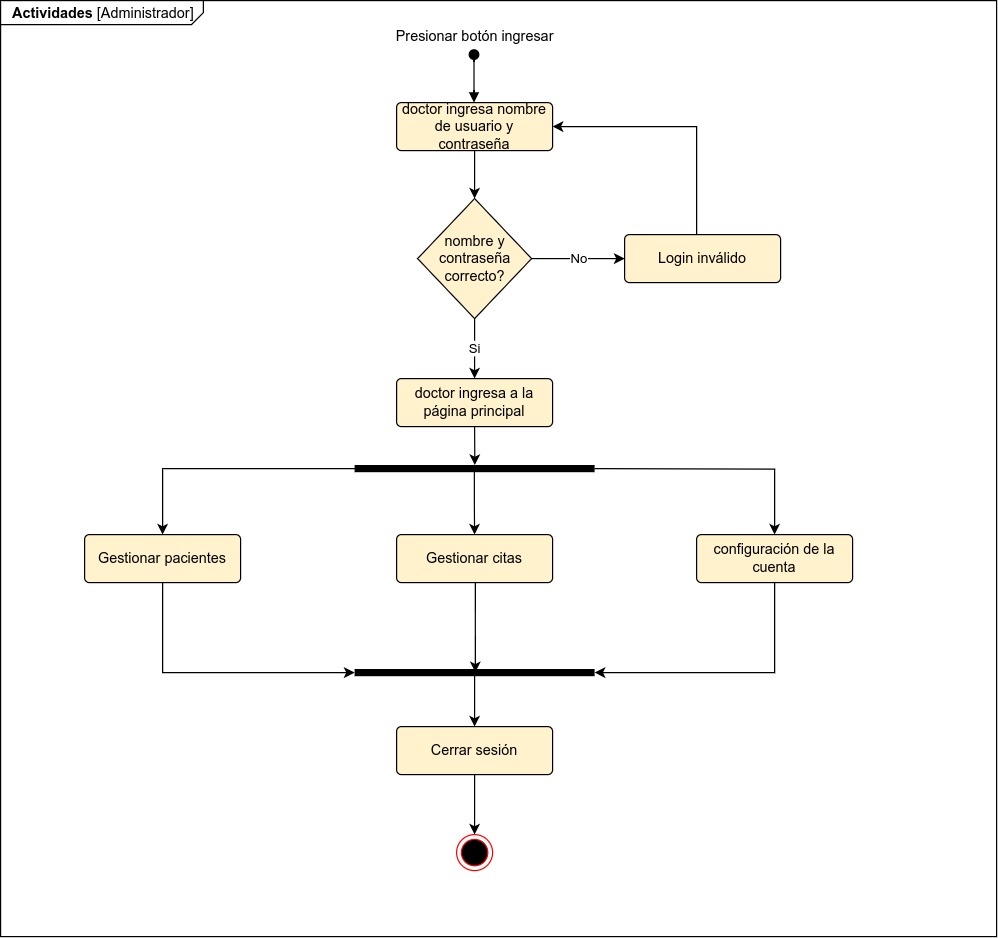
*Diagrama de actividades administrador*

****

Nota: Autoría Propia

**Figura 23**

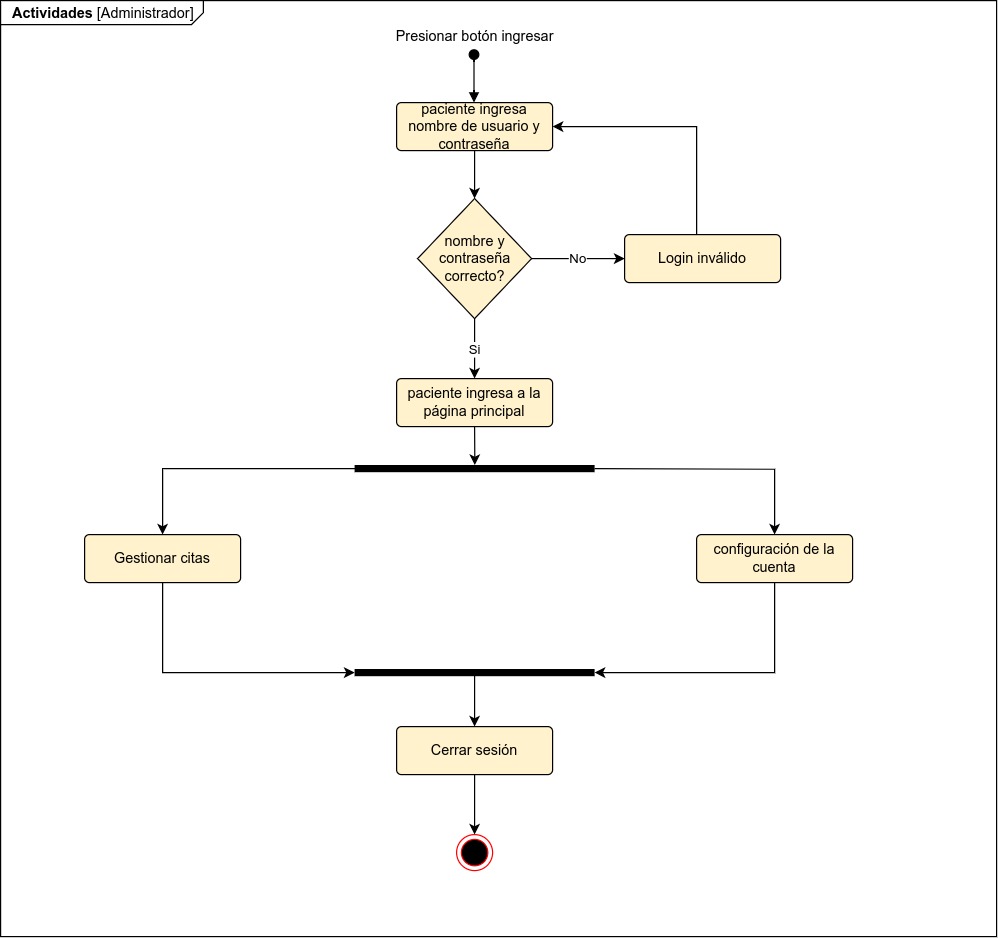
*Diagrama de actividades doctor*



Nota: Autoría propia.

**Figura 24**

*Diagrama de actividades paciente*



Nota: Autoría propia.

# Conclusiones

# Recomendaciones

# Referencias

Alenezi, M., & Almuairfi, S. (2019). Security risks in the software development lifecycle. International Journal of Recent Technology and Engineering, 8(3), 7048–7055. https://doi.org/10.35940/IJRTE.C5374.098319

Alfawair, M. (2022). INTERNET-OF-THINGS: A SYSTEM DEVELOPMENT LIFE CYCLE (SDLC). *Journal of Theoretical and Applied Information Technology*, *31*(6). www.jatit.org

Alibaba. (2021, enero 4). *Application Logical Architecture: Definition and Derivation - Alibaba Cloud Community*. https://www.alibabacloud.com/blog/application-logical-architecture-definition-and-derivation\_597098

Altamirano Rodríguez, M. de L., Amador Angón, L., Andrade Cortés, M., Aquino Muñiz, D., Arriola Díaz, L., Benítez Moreno, G. E., Blanno Castro, A., Cabrera Cabello, E., Cabrera Hernández, N. L., Contreras Ruiz, J., Durán Cruzado, J., Enríquez Corona, R., Estévez Dorantes, T. L., Galán García, J. J., Galán Montero, A. D. S., Galán Montero, J. A., Galán Montero, S., García Santos, C., Gómez Nieto, J. P., … Venegas García, J. A. (2019). *COMPRENDER EL ENTORNO. CAMINOS HACIA EL CONOCIMIENTO* (S. C. Instituto de Estudios Superiores del Valle de Orizaba, Ed.; 1a ed.). http://www2.univo.edu.mx/web/ForoUniVO/Home/Memorias/2019.pdf#page=27

Alvarado Tintin, E. D., & López Ibarra, F. J. (2022). *IMPLEMENTACIÓN DE UN SISTEMA WEB PARA LA GESTIÓN ADMINISTRATIVA Y UN APLICATIVO MÓVIL INFORMATIVO PARA EL CENTRO DE REHABILITACIÓN FISIOSALUDXP* [Universidad Agraria del Ecuador]. https://cia.uagraria.edu.ec/Archivos/LOPEZ%20IBARRA%20FRANCISCO%20JAVIER.pdf

Asif, M., Ali, I., Malik, M. S. A., Chaudary, M. H., Tayyaba, S., & Mahmood, M. T. (2019). Annotation of Software Requirements Specification (SRS), Extractions of Nonfunctional Requirements, and Measurement of Their Tradeoff. *IEEE Access*, *7*, 36164–36176. https://doi.org/10.1109/ACCESS.2019.2903133

AtSistemas. (2021, septiembre 21). *Los principios SOLID en el diseño de aplicaciones de software | Cuti*. https://cuti.org.uy/blog/de-socios-blog/los-principios-solid-en-el-diseno-de-aplicaciones-de-software/

Barrera, H. (2020). *Evaluación Experimental para Determinar la Relación entre las Métricas de Cohesión y Acoplamiento en Arquitecturas Orientadas a Objetos*. https://rinacional.tecnm.mx/jspui/handle/TecNM/4152

Blas Cerda, E. C. (2020). Implementación de un sistema web, aplicando la metodología UWE, para dar soporte a la gestión contable en megaconcreto ingeniería y construcción S.A.C., 2018. *Universidad Nacional Santiago Antúnez de Mayolo*. http://repositorio.unasam.edu.pe/handle/UNASAM/4349

Blas, M. J., Leone, H. P., & Gonnet, S. M. (2020). Modelado y verificación de patrones de diseño de arquitectura de software para entornos de computación en la nube. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, *35*, 1–17. https://doi.org/10.17013/RISTI.35.1-17

Briones, A. (2018). Implementación de Un Sistema Web Para La Gestión Administrativa EN EL CENTRO DE TERAPIA FÍSICA Y REHABILITACIÓN DE LA CLÍNICA GOOD HOPE, MIRAFLORES 2018. *Universidad Peruana de las Américas*. http://repositorio.ulasamericas.edu.pe/xmlui/handle/upa/376

Calderón, J. (2020). *SISTEMA WEB PARA LA ADMINISTRACIÓN Y SEGUIMIENTO DEL TRATAMIENTO DE FISIOTERAPIA Y KINESIOLOGIA INTEGRAL* [UNIVERSIDAD PÚBLICA DE EL ALTO]. http://repositorio.upea.bo/handle/123456789/166

Capilla, R. (2022). *Concepto de Arquitectura Software y Principios de Diseño*. https://burjcdigital.urjc.es/bitstream/handle/10115/20301/DAS-Temas.pdf?sequence=1&isAllowed=y

Castellanos, E. (2021, febrero 14). *Git vs GitHub – ¿Qué es el Control de Versiones y Cómo Funciona?* https://www.freecodecamp.org/espanol/news/git-vs-github-what-is-version-control-and-how-does-it-work/

Chán, D. I. C., Vera, R. A. A., Pech, J. P. U., Mendoza, J. C. D., & Tejero, J. A. S. (2021). Diseño de un prototipo para configuración de entornos virtuales de aprendizaje basados en gamificación utilizando UWE. *ReCIBE, Revista electrónica de Computación, Informática, Biomédica y Electrónica*, *10*(1), C1-16. https://doi.org/10.32870/RECIBE.V10I1.217

Cruz-Benito, J., García-Peñalvo, F. J., & Therón, R. (2019). Analyzing the software architectures supporting HCI/HMI processes through a systematic review of the literature. *Telematics and Informatics*, *38*, 118–132. https://doi.org/10.1016/J.TELE.2018.09.006

Drew. (2019, diciembre 3). *Ventajas y desventajas de la metodología Scrum*. http://blog.wearedrew.co/productividad/-ventajas-y-desventajas-de-la-metodologia-scrum

Dutonde, P. D., Mamidwar, S. S., Sunil Korvate, M., Bafna, S., & Shirbhate, D. D. (2022). *Website Developmemt Technologies: A Review*. *10*. https://doi.org/10.22214/ijraset.2022.39839

Dwivedi, N., Katiyar, D., & Goel, G. (2022). A Comparative Study of Various Software Development Life Cycle (SDLC) Models. *International Journal of Research in Engineering, Science and Management*, *5*(3), 141–144. http://journals.resaim.com/ijresm/article/view/1881

Elías Chanchí, G. G., Clara Gómez, M. A., & Yesid Campo, W. M. (2019). *Propuesta de un videojuego educativo para la enseñanza-aprendizaje de la clasificación de requisitos en ingeniería de software*.

Elisa Baniassad. (2018). *Making the Liskov Substitution Principle Happy and Sad*. https://doi.org/10.1145/3183377.3183380

García Sosa, S. A. (2021). SISTEMA WEB BASADO EN EL MODELO VISTA CONTROLADOR PARA OPTIMIZAR LA GESTIÓN DE TRÁMITE DOCUMENTARIO EN LA UNIDAD DE GESTIÓN EDUCATIVA LOCAL DE MARISCAL RAMÓN CASTILLA – 2019. *Universidad Científica del Perú*. http://repositorio.ucp.edu.pe/handle/UCP/1424

Gómez Fuentes, M. del C., Cervantes Ojeda, J., & Gonzáles Perez, P. P. (2019). *Fundamentos de ingeniería de software*. http://ilitia.cua.uam.mx:8080/jspui/handle/123456789/1000

Govil, N., & Sharma, A. (2022). Validation of agile methodology as ideal software development process using Fuzzy-TOPSIS method. *Advances in Engineering Software*, *168*, 103125. https://doi.org/10.1016/J.ADVENGSOFT.2022.103125

Guamán Coronel, D. A. (2021). *Arquitectura de Software*. Ediloja Cía. Ltda.

L.F. Santiago González, Mendoza, J. C. H., Luna, Y. M., Rodríguez, A. J. R., Requena, D. T. V., Rodríguez, J. L. M., & Navejar, J. G. M. (2021). Aplicación Web Basada en el Patrón de Arquitectura de Software Modelo-Vista- Controlador (MVC) para Incrementar el Desempeño Académico en la Asignatura de Matemáticas Básicas. *Tecnología Educativa Revista CONAIC*, *8*(1), 7–21. https://doi.org/10.32671/TERC.V8I1.187

McGilvray, D. (2021). Executing Data Quality Projects: Ten Steps to Quality Data and Trusted Information (TM). En *Executing Data Quality Projects: Ten Steps to Quality Data and Trusted Information (TM)*. Elsevier. https://doi.org/10.1016/B978-0-12-818015-0.09991-6

Medina Cruz, J., Pineda Ballesteros, E., Téllez Acuña, F. R., Medina Cruz, J., Pineda Ballesteros, E., & Téllez Acuña, F. R. (2019). Requerimientos de software: prototipado, software heredado y análisis de documentos. *Ingeniería y Desarrollo*, *37*(2), 327–345. https://doi.org/10.14482/INDE.37.2.1053

Mendoza, M. D., Trisna, T., & Putri, A. (2020). Payroll System Design With SDLC (System Development Life Cycle) Approac. *Jurnal Mantik*, *4*(1), 27–32. https://doi.org/10.18502/KSS.V3I10.3406

Microsoft. (2023). *Visual Studio: IDE y Editor de código para desarrolladores de software y Teams*. https://visualstudio.microsoft.com/es/

Milena Velásquez Restrepo, S., David Vahos-Montoya, J., Ester Gómez-Adasme, M., Alexandra Pino -Martínez, A., Juliett Restrepo-Zapata, E., & Londoño-Marín, S. (2019). Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software. *Revista CINTEX*, *24*(2), 13–23. https://doi.org/10.33131/24222208.334

Molina-Ríos, J., & Pedreira-Souto, N. (2020). Comparison of development methodologies in web applications. *Information and Software Technology*, *119*, 106238. https://doi.org/10.1016/J.INFSOF.2019.106238

Mozilla. (2023, marzo 9). *¿Qué es JavaScript? - Aprende sobre desarrollo web | MDN*. https://developer.mozilla.org/es/docs/Learn/JavaScript/First\_steps/What\_is\_JavaScript

Mubin, S. A., Jantan, A. H., Abdullah, R., & Kamaruddin, A. (2016). UML stereotypes for the development of process interaction-driven web applications. *Proceedings of CHIuXiD 2016, the 2nd International Human Computer Interaction and User Experience Conference in Indonesia: Bridging the Gaps in the HCI and UX World*, 81–88. https://doi.org/10.1145/2898459.2898472

*MySQL :: MySQL 8.0 Reference Manual :: 1.2.1 What is MySQL?* (s/f). Recuperado el 5 de marzo de 2023, de https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html

Ortiz Sánchez, D. L. (2022). *Análisis del diseño de un sistema informático para la gestión administrativa en el consultorio de fisioterapia y rehabilitación AMP de la ciudad de Babahoyo.* http://dspace.utb.edu.ec/handle/49000/12674

PacktPublishing. (2018, mayo 12). *System Modeling: Understanding Logical and Physical Architecture - DataScienceCentral.com*. https://www.datasciencecentral.com/system-modeling-understanding-logical-and-physical-architecture/

Peláez Valencia, L. E., Toro Lazo, A., Arias Vargas, J. L., & Rodríguez Franco, D. E. (2019). Ingeniería de Software: el aseguramiento de la calidad de los requisitos en la industria del software en el eje cafetero colombiano. *INGE CUC, ISSN 0122-6517, ISSN-e 2382-4700, Vol. 15, No. 2 (July-December), 2019, págs. 110-122*, *15*(2), 110–122. https://doi.org/10.17981/ingecuc.15.2.2019.11

*PHP: What is PHP? - Manual*. (s/f). Recuperado el 5 de marzo de 2023, de https://www.php.net/manual/en/intro-whatis.php

Piñero Villena, J. J., & Reyes Reyes, J. A. (2021). *Desarrollo de un Sistema Web para la gestión del historial clínico y terapias para el Instituto de Medicina del Deporte y Ortopedia, IMDO*. http://dspace.ups.edu.ec/handle/123456789/19755

Pooja Gupta. (2022). *Logical Architecture | An Overview on Components of Logical Architecture*. https://www.educba.com/logical-architecture/

Rahmani, H. F., & Hikmawati, E. (2020). Combining SDLC Method and ITIL Framework by Involving Auditors. *Jurnal AKSI (Akuntansi dan Sistem Informasi)*, *5*(1), 6–12. https://doi.org/10.32486/AKSI.V5I1.32

Román, J. (s/f). *¿Qué es CSS? - CSS en español - Lenguaje CSS*. Recuperado el 9 de marzo de 2023, de https://lenguajecss.com/css/introduccion/que-es-css/

SAC. (2022, noviembre 18). *Metodología tradicional(cascada) vs. Metodología Ágil(Scrum) – Sac*. https://sacpma.com/metodologia-tradicional-vs-metodologia-agil-2/

Sarria Mediaceja, A. (2019). *Requerimientos de seguridad de autenticación y generación de notificaciones en el Sistema Informático XABAL Arkheia 4.0*. https://repositorio.uci.cu/jspui/handle/123456789/10264

Shaikh, S., & Abro, S. (2019). Comparison of Traditional & Agile Software Development Methodology: A Short Survey. *International Journal of Software Engineering and Computer Systems*, *5*(2), 1–14. https://doi.org/10.15282/ijsecs.5.2.2019.1.0057

Solano-Fernández, E., Porras-Alfaro, D., -Fernández, S., Porras-Alfaro, E. ;, & El, D. (2020). El modelo iterativo e incremental para el desarrollo de la aplicación de realidad aumentada Amón\_RA. *Tecnología en Marcha, ISSN 0379-3962, ISSN-e 2215-3241, Vol. 33, No. Extra 8, 2020 (Ejemplar dedicado a: Escuela de Arquitectura y Urbanismo), págs. 165-177*, *33*(8), 165–177. https://doi.org/10.18845/tm.v33i8.5518

Sparx Systems. (s/f-a). *Class Diagram - UML 2 Tutorial | Sparx Systems*. Recuperado el 8 de febrero de 2023, de https://sparxsystems.com/resources/tutorials/uml2/class-diagram.html

Sparx Systems. (s/f-b). *Use Case Diagram | Enterprise Architect User Guide*. Recuperado el 8 de febrero de 2023, de https://sparxsystems.com/enterprise\_architect\_user\_guide/15.2/guidebooks/tech\_sysml\_usecase\_diagram.html

Sparx Systems. (s/f-c). *UWE - Tutorial - Navigation Model (Español)*. Recuperado el 9 de febrero de 2023, de https://uwe.pst.ifi.lmu.de/teachingTutorialNavigationSpanish.html

Tabarés, R. (2021). HTML5 and the evolution of HTML; tracing the origins of digital platforms. *Technology in Society*, *65*, 101529. https://doi.org/10.1016/J.TECHSOC.2021.101529

Umar, M. A. (2020, junio 29). *Comprehensive study of software testing: Categories, levels, techniques, and types*. https://www.researchgate.net/publication/342538504\_Comprehensive\_study\_of\_software\_testing\_Categories\_levels\_techniques\_and\_types

Zola, A. (2022, agosto). *What is a Bootstrap and how does it work?* https://www.techtarget.com/whatis/definition/bootstrap

# Apéndice