



UNIMEDIC v0.1

ASISTENTE VIRTUAL MÉDICO

---

## Plan de Desarrollo de Software

---

# Índice

<b>1. Introducción</b>	<b>2</b>
1.1. Propósito . . . . .	2
1.2. Alcance . . . . .	2
1.3. Resumen . . . . .	2
<b>2. Vista General del Proyecto</b>	<b>2</b>
2.1. Propósito, Alcance y Objetivos . . . . .	2
2.2. Suposiciones y Restricciones . . . . .	3
2.3. Entregables del proyecto . . . . .	3
2.3.1. Modelo de negocio . . . . .	3
2.3.2. Visión . . . . .	3
2.3.3. Glosario . . . . .	3
2.3.4. Modelo de casos de uso . . . . .	3
2.3.5. Especificaciones de requisitos de software (con casos de uso) . . . . .	4
2.3.6. Especificaciones de caso de uso con prototipo . . . . .	4
2.3.7. Diagrama de robustez . . . . .	4
2.3.8. Diagrama de secuencia . . . . .	4
2.3.9. Arquitectura C4 . . . . .	4
2.3.10. Diagrama de clases . . . . .	4
2.3.11. Diagrama de colaboración . . . . .	4
2.3.12. Diagrama de componentes . . . . .	5
2.3.13. Diagrama de despliegue . . . . .	5
2.3.14. Diagrama de estados . . . . .	5
2.3.15. Demo . . . . .	5
2.4. Evaluación del Plan de Desarrollo del Software . . . . .	5
<b>3. Organización del Proyecto</b>	<b>5</b>
3.1. Participantes del proyecto . . . . .	5
3.2. Interfaces Externas . . . . .	6
3.3. Roles y Responsabilidades . . . . .	7
<b>4. Gestión de proceso</b>	<b>7</b>
4.1. Plan de Proyecto . . . . .	7
4.1.1. Plan de las Fases . . . . .	7
4.1.2. Calendario del proyecto . . . . .	8

## 1. Introducción

Este Plan de Desarrollo del Software es una versión preliminar preparada para ser incluida en la respuesta al Proyecto de Modelamiento de software. Este proyecto tiene como finalidad diseñar un aplicativo web y móvil de un asistente medico virtual que permitirá realizar consultas y obtener un diagnóstico sin la necesidad de formar grandes colas para conseguir una cita y así evitar la propagación del covid-19.

### 1.1. Propósito

Establece crear un aplicativo web y móvil para facilitar la comprensión y el manejo del asistente medico uniMedic.

Establecer un modelo que facilite al usuario ingresar las consultas e imágenes médicas respectivas, también para que le sea más realista la interacción con el chatBot.

### 1.2. Alcance

Desarrollar una interfaz acuerdo a la necesidad dada en la planeación del problema.

### 1.3. Resumen

Después de esta introducción, el resto del documento está organizado en las siguientes secciones:

- Vista General del Proyecto – proporciona una descripción del propósito, alcance y objetivos del proyecto, estableciendo los *artefactos* que serán brindados y utilizados.
- Organización del Proyecto – describe la estructura organizacional del equipo de desarrollo.
- Gestión de Procesos – explica los costos y planificación estimada, define las fases e hitos del proyecto y describe cómo se realizará su seguimiento.
- Planes y Guías de Aplicación – proporciona una vista global del proceso de desarrollo de software incluyendo métodos, herramientas y técnicas que serán utilizadas.

## 2. Vista General del Proyecto

### 2.1. Propósito, Alcance y Objetivos

Establece medios suficientes para un tratamiento eficaz de la misma forma que si fueras a un hospital con un médico especializado. La aplicación hace uso de modelos de aprendizaje automático y traerá los siguientes beneficios:

- No hay costo alguno en las consultas y en el diagnóstico
- Evita realizar largas colas en hospitales.
- Permite dar un diagnóstico confiable para una rápida recuperación del paciente.
- Permite realizar consultas de tomografías y radiografías pulmonares gratuitamente.

## **2.2. Suposiciones y Restricciones**

- Soporte de varios idiomas
- Alto tiempo de inferencia, algunos modelos sufren de retardo o demora al momento de hacer una predicción o inferencia.
- Etiquetado médico (tipo de imagen que se está introduciendo en la aplicación).
- Falta de modelos, el catálogo de modelos de IA disponibles no cubre la totalidad de problemas que el paciente pueda presentar (tales como el análisis de tuberculosis o anemia).

## **2.3. Entregables del proyecto**

A continuación, se indica y describen cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables.

### **2.3.1. Modelo de negocio**

Un modelo de negocio es una herramienta previa al plan de negocio que te permitirá definir con claridad qué vas a ofrecer al mercado, cómo lo vas a hacer, a quién se lo vas a vender, cómo se lo vas a vender y de qué forma vas a generar ingresos. Es una herramienta de análisis que te permitirá saber quién eres, cómo lo haces, a qué coste, con qué medios y qué fuentes de ingresos vas a tener. Definir tu modelo de negocio es saber cuál es tu ADN, cómo está hecho, cómo se puede modificar, cómo pulir, cómo cambiar, cómo moldear.

### **2.3.2. Visión**

Un documento de visión define el alcance y el objetivo de alto nivel de un programa, producto o proyecto. Una declaración clara del problema, una propuesta de solución y las características de alto nivel de un producto ayudan a establecer expectativas y reducir riesgos. Este tema ofrece un esquema del contenido potencial de un documento de visión.

### **2.3.3. Glosario**

Un glosario es una recopilación de definiciones o explicaciones de palabras que versan sobre un mismo tema u ordenada de forma alfabética. Es común que se los incluya como anexo al final de libros, investigaciones, tesis o enciclopedias, poco conocidos, de difícil interpretación o que no sean comúnmente utilizados en el contexto en que aparecen. De esta forma, un glosario no es lo mismo que un diccionario y tienen la característica de dar conceptos, tienen una ligera diferencia: en el glosario solo podemos encontrar términos propios de un campo o de un libro específico; mientras que el diccionario sirve para comprenderlo

### **2.3.4. Modelo de casos de uso**

Un modelo de caso de uso es un modelo de las funciones previstas del sistema y su entorno y sirve como un contrato entre el cliente y los desarrolladores. Los casos de uso sirven como hebra de unión a lo largo del desarrollo del sistema. El mismo modelo de caso de uso es el resultado de la disciplina de Requisitos y se utiliza como entrada para disciplinas de Prueba, Diseño y Análisis.

### **2.3.5. Especificaciones de requisitos de software (con casos de uso)**

La especificación de requisitos de software (ERS) es una descripción completa del comportamiento del sistema que se va a desarrollar. Incluye un conjunto de casos de uso que describe todas las interacciones que tendrán los usuarios con el software. Los casos de uso también son conocidos como requisitos funcionales. Además de los casos de uso, la ERS también contiene requisitos no funcionales (complementarios). Los requisitos no funcionales son requisitos que imponen restricciones en el diseño o la implementación, como, por ejemplo, restricciones en el diseño o estándares de calidad.

### **2.3.6. Especificaciones de caso de uso con prototipo**

La especificación de un caso de uso debe describir el modo en que un actor interactúa con el sistema. Es una narración que describe el rol desempeñado por el actor en su interacción con el sistema.

Un prototipo es una representación limitada del diseño de un producto que permite a las partes responsables de su creación experimentar, probarlo en situaciones reales y explorar su uso. Un prototipo puede ser cualquier cosa, desde un trozo de papel con sencillos dibujos a un complejo software.

### **2.3.7. Diagrama de robustez**

Es una técnica que permite analizar los pasos de un caso de uso para validar su lógica y asegurar que es lo bastante robusto.

### **2.3.8. Diagrama de secuencia**

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. A menudo es útil para complementar a un diagrama de clases, pues el diagrama de secuencia se podría describir de manera informal como "el diagrama de clases en movimiento", por lo que ambos deben estar relacionados entre sí (mismas clases, métodos, atributos...).

### **2.3.9. Arquitectura C4**

Los diagramas de arquitectura de software son una manera fantástica de comunicar cómo se planea construir un sistema de software (diseño inicial) o cómo funciona un sistema de software existente (documentación retrospectiva, intercambio de conocimientos y aprendizaje).

### **2.3.10. Diagrama de clases**

En ingeniería de software, un diagrama de clases en Lenguaje Unificado de Modelado (UML) es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones (o métodos), y las relaciones entre los objetos.

### **2.3.11. Diagrama de colaboración**

El diagrama de colaboración es un tipo de diagrama de interacción cuyo objetivo es describir el comportamiento dinámico del sistema de información que muestra cómo interactúan los objetos entre sí, es decir, con qué otros objetos tiene vínculos o intercambia mensajes de un objeto determinado.

#### **2.3.12. Diagrama de componentes**

Los diagramas de componentes UML representan las relaciones entre los componentes individuales del sistema mediante una vista de diseño estática. Pueden ilustrar aspectos de modelado lógico y físico. En el contexto del UML, los componentes son partes modulares de un sistema independientes entre sí, que pueden reemplazarse con componentes equivalentes.

#### **2.3.13. Diagrama de despliegue**

Los diagramas de despliegue se utilizan para visualizar los procesadores/nodos/dispositivos de hardware de un sistema, los enlaces de comunicación entre ellos y la colocación de los archivos de software en ese hardware.

#### **2.3.14. Diagrama de estados**

Los diagramas de estado UML pueden ayudar a visualizar el ciclo de vida de cada objeto de forma clara y comprensible. Aunque este diagrama solo consta de unos pocos elementos, si se utiliza correctamente, puede contribuir notablemente al resultado final.

#### **2.3.15. Demo**

Las demos o versiones demo son prototipos, versiones incompletas o de evaluación de un determinado programa informático con el fin de mostrar la idea de funcionamiento y demostrar sus funcionalidades. Se utilizan para que los usuarios o potenciales clientes puedan probar el software antes de comenzar a utilizarlo en un ambiente real.

### **2.4. Evaluación del Plan de Desarrollo del Software**

El Plan de Desarrollo de Software se revisará en las semanas de prácticas calificadas 4,5 y 6, la entrega final del proyecto será en la fecha del examen final y la subsanación y observaciones en la fecha de exámenes sustitutorios.

## **3. Organización del Proyecto**

### **3.1. Participantes del proyecto**

- Sanchez S. Cristhian (20180517A)
- Lopez C. Ronaldo (20181469K)
- Navio T. Jose (20175526F)
- Valero M. Andro Erick (20172757G)
- Raymundo M. Abraham (20180562G)
- Alderete V. Davis (20181139K)

### 3.2. Interfaces Externas

---

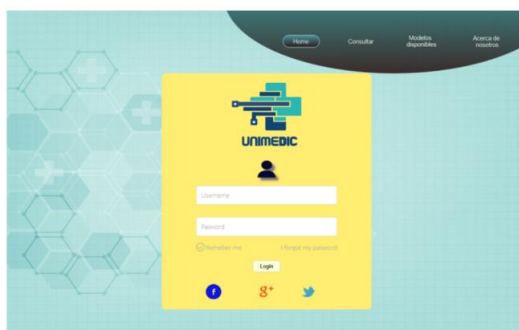


Figura 2: Login

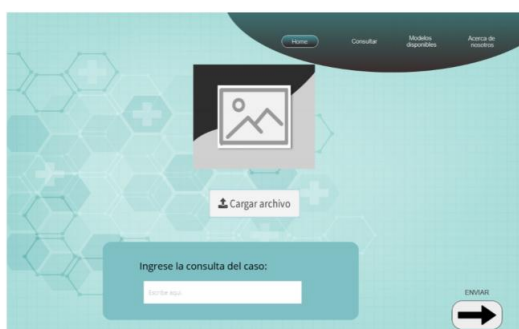


Figura 3: Ventana de consultas

### 3.3. Roles y Responsabilidades

Integrante	Responsabilidades
Sanchez S. Cristhian	Modelo de Negocio. Diagrama de secuencia. Entrenamiento y prueba de los modelos de Deep Learning.
Lopez C. Ronaldo	Glosario. Especificaciones de requisitos de software (con casos de uso). Diagrama de clases. Diagrama de componentes. Diseño y prueba del servidor.
Navio T. Jose	Modelo de casos de uso. Diagrama de colaboración. Entrenamiento y prueba de los modelos de Deep Learning.
Valero M. Andro Erick	Especificaciones de caso de uso con prototipo. Diagrama de despliegue. Diseño de las interfaces Web y Mobile.
Raymundo M. Abraham	Documento de visión. Arquitectura C4 (gráficos y documentación). Diseño de las interfaces web y Mobile.
Alderete V. Davis	Documento de visión. Diagrama de robustez. Diagrama de estados. Diseño de las interfaces web y Mobile.

## 4. Gestión de proceso

### 4.1. Plan de Proyecto

Se tiene diversas fases del proyecto que va organizadas según el requerimiento entregado en la problemática.

#### 4.1.1. Plan de las Fases

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La siguiente tabla muestra una la distribución de tiempos y el número de entregables de cada fase.

Fase	N° Entregables	Duración
Primer entregable	3	1 semana
Segundo entregable	6	2 semanas
Tercer entregable	7	1 semana
Cuarto entregable	Todos	5 semanas



4.1.2. Calendario del proyecto

	Semana 1	Semana 2	Semana 4	Semana 5	Semana 6
<i>Vision, glosario, casos de uso y especificaciones del proyecto</i>					
<i>Diseño de las interfaces Web y Mobile</i>					
<i>Entrenamiento y prueba de los modelos de Deep Learning</i>					
<i>Diseño y prueba del servidor</i>					
<i>Integración de todos los módulos</i>					
<i>Pruebas unitarias y de calidad</i>					
<i>Despliegue de la aplicación</i>					

Leyenda:

Equipo 1

Valero  
Alderete  
Raymundo

Equipo 2

Lopez

Equipo 3

Sánchez  
Navío

TODOS