

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



**PERFIL DE PROYECTO DE GRADO**

**SISTEMA WEB DE GESTIÓN DOCUMENTAL DE CASOS  
DE ORTODONCIA Y ORTOPEDIA PARA LA SOCIEDAD  
BOLIVIANA DE ORTODONCIA Y ORTOPEDIA  
DENTOFACIAL – LA PAZ**

Proyecto de Grado para obtener el Título de Licenciatura en Informática  
Mención Ingeniería de Sistemas Informáticos

**POR: CRISTHIAN EDWIN ORTIZ MERCADO**  
**TUTOR METODOLÓGICO: M.SC. ALDO RAMIRO VALDEZ**  
**ALVARADO**  
**ASESOR: LIC. BRIGIDA ALEXANDRA CARVAJAL BLANCO**

**LA PAZ – BOLIVIA**  
**Julio, 2020**

# ÍNDICE

1. INTRODUCCIÓN .....	5
2. ANTECEDENTES.....	6
2.1. ANTECEDENTES INSTITUCIONALES .....	6
2.1.1 MISION.....	7
2.1.2 VISION .....	7
2.2. PROYECTOS SIMILARES .....	7
3. PLANTEAMIENTO DEL PROBLEMA .....	10
3.1. PROBLEMA CENTRAL .....	10
3.2. PROBLEMAS SECUNDARIOS .....	11
4. DEFINICIÓN DE OBJETIVOS.....	11
4.1. OBJETIVO GENERAL .....	11
4.2. OBJETIVOS ESPECÍFICOS .....	12
5. JUSTIFICACIÓN .....	12
5.1. JUSTIFICACIÓN ECONÓMICA.....	12
5.2. JUSTIFICACIÓN SOCIAL .....	12
5.3. JUSTIFICACIÓN TECNOLÓGICA.....	13
6. ALCANCES Y LÍMITES .....	13
6.1. ALCANCES.....	13
6.2. LÍMITES.....	14
7. APORTES.....	15
7.1. PRÁCTICO.....	15
7.2. TEÓRICO .....	15
8. METODOLOGÍA .....	15
8.1 MÉTODO DE INVESTIGACIÓN.....	15
8.2 METODOLOGÍA DE DESARROLLO .....	16
9. MARCO TEÓRICO.....	18
9.1 GESTION DOCUMENTAL .....	18

9.1.1 ARCHIVO CENTRAL .....	18
9.1.2 DOCUMENTO ELECTRONICO .....	18
9.2 REGISTROS ORTODONCICOS .....	21
9.3 INGENIERÍA DE SOFTWARE .....	22
9.3.1 ETAPAS DEL PROCESO .....	23
9.3.2 MODELOS DE PROCESOS DE SOFTWARE .....	25
9.4 METODOLOGÍA DE DESARROLLO SCRUM .....	27
9.4.1 HERRAMIENTAS DE LA METODOLOGÍA .....	28
9.4.2 ROLES Y RESPONSABILIDADES .....	31
9.4.3 PROCESO DE LA METODOLOGÍA .....	34
9.4.4 CONTROL DE EVOLUCIÓN DE PROYECTO .....	37
9.5 INGENIERÍA WEB .....	38
9.5.1 PROCESO DE INGENIERÍA WEB .....	38
9.5.2 HERRAMIENTAS TECNOLÓGICAS .....	39
9.6 METODOLOGÍA UWE .....	42
9.6.1 FASES DE LA METODOLOGÍA UWE .....	42
10. ÍNDICE TENTATIVO .....	47
11. CRONOGRAMA DE AVANCE .....	50
12. BIBLIOGRAFÍA .....	50
ANEXOS .....	53
ANEXO A – ÁRBOL DE PROBLEMAS .....	53
ANEXO B – ÁRBOL DE OBJETIVOS .....	54
ANEXO C – MARCO LÓGICO .....	55

## ÍNDICE DE TABLAS

Tabla 1: Índice Tentativo.....	49
Tabla 2: Elementos de diseño Navegacional .....	50
Tabla 3: Matriz de Marco Lógico .....	56

## ÍNDICE DE FIGURAS

Figura 1: Visión general del proceso de la metodología Scrum .....	36
Figura 2: Análisis de caso de uso .....	43
Figura 3: Modelo de Clases .....	43
Figura 4: Relaciones y enlaces.....	45
Figura 5: Diseño navegacional UWE.....	45
Figura 6: Diagrama de presentación.....	46
Figura 7: Árbol de problemas .....	53
Figura 8: Árbol de objetivos .....	54

## 1. INTRODUCCIÓN

Una gran problemática en cuanto al entorno laboral y académico en nuestro país, es la falta de orden y documentación que existe en los diferentes campos laborales, pero especialmente en salud es de conocimiento público que el manejo de la información médica se encuentra en precarias condiciones y esto puede llegar a generar un estancamiento en cuanto a avances en tratamientos e investigación. Específicamente en el área de odontología (ortodoncia y ortopedia dentofacial). Según los doctores Parra y Mercado “En la ciudad de La paz, al existir varios profesionales trabajando de manera independiente y no estar bajo algún tipo de normativa de documentación por parte de alguna institución, se presentan casos que a criterio profesional pueden ser compartidos académicamente. Lastimosamente estos casos quedan sin documentación y no pueden ser compartidos para que la comunidad de profesionales obtenga información de utilidad para mejorar la formación académica, y así mejorar la calidad de tratamientos dentales ortopédicos y de ortodoncia en la ciudad de La Paz.” (Parra, y Mercado, comunicación personal, enero 2020).

Así mismo la el doctor Aguirre, presidente de la “Sociedad Boliviana de ortodoncia y ortopedia Dentofacial - La Paz” menciona que esta busca ser un soporte para los profesionales dedicados a esta rama de la odontología para consolidar su formación académica y práctica (Aguirre, comunicación personal, marzo 2020).

Un gran parte de los profesionales en odontología trabajan de manera independiente y no se rigen bajo criterios de documentación de casos tratados, estos se documentan para el seguimiento en la medida que el profesional considere necesario. Pero definitivamente se nos encontramos en una nueva era en la cual la transformación digital se acopla a varias áreas del entorno laboral y académico, lo que nos sugiere que es momento de que estas áreas vayan adaptando su entorno a la nueva etapa digital donde contamos con herramientas que facilitan la difusión de información que puede llegar a ser valiosa y estar al alcance de todos.

Por lo que los doctores Paredes, Gandía y Cibrián (2006) expresan lo siguiente, “Actualmente la tecnología digital es una realidad que cada vez se impone más en todos los ámbitos clínicos y, por tanto, existe una incorporación también de los ortodoncistas a la digitalización de los registros ortodóncicos diagnósticos. También afirman que “los registros ortodóncicos son una herramienta fundamental a la hora de realizar un correcto diagnóstico y plan de tratamiento. Estos registros pueden dividirse básicamente en tres grandes grupos: radiografías, fotografías y modelos de estudio. Estos deben realizarse antes, en muchas ocasiones durante, y al finalizar cualquier tratamiento de Ortodoncia”. Tradicionalmente, las fotografías y las radiografías se han realizado en formato papel, tanto fotográfico como radiográfico, mientras que los modelos de estudio se hacen en escayola. Sin embargo, cada vez más se está produciendo un gran cambio hacia la digitalización de estos registros ortodóncicos.

Cualquier ortodoncista con unos años de experiencia profesional, experimenta el problema del almacenamiento físico de los modelos de estudio de ortodoncia en la clínica, además del tiempo que se pierde clasificando y almacenando los mismos, por ello la digitalización de estos parece una idea muy atractiva”.

## **2. ANTECEDENTES**

### **2.1. ANTECEDENTES INSTITUCIONALES**

La Sociedad Boliviana de Ortodoncia es una institución sin fines de lucro que agrupa a todos los especialistas en ortodoncia y ortopedia dentofacial de Bolivia. Reconocida por el colegio de Odontólogos de Bolivia, entre Que se por medio de actividades, cursos, seminarios promueve una labor social y profesional colaborativa para mejorar el ámbito profesional en la especialidad.

### 2.1.1 MISION

La misión de la Sociedad Boliviana de Ortodoncia es consolidar la formación práctica y académica del profesional de la ortodoncia para que cumpla la responsabilidad social que le impone ser un gestor de la salud pública de nuestro país, además la actualización continua de cada uno de nuestros asociados, poniendo especial empeño en difundir y preservar el código de ética profesional inherente al ortodoncista y en fortalecer la imagen del gremio ante la sociedad.

### 2.1.2 VISION

Ser la institución busca ser para los profesionales especialistas un instrumento para mejorar la calidad en el desempeño laboral y conocimientos en el ámbito de ortodoncia y ortopedia dentofacial

## 2.2. PROYECTOS SIMILARES

En una búsqueda de información de sistemas similares en la Carrera de Informática en la Universidad Mayor de San Andrés se presentan los proyectos descritos en la Tabla 2.1:

Descripción	Módulos
<b>“SISTEMA WEB DE CONTROL DE PAGOS, CITAS E HISTORIALES CLÍNICOS CASO: CLÍNICA DENTAL LAVADENT”, Sara Patricia Huanca Cantuta 2015, Universidad Mayor de San Andrés, carrera de informática, presenta el “Sistema Web de control de pagos, citas e historiales clínicos para la clínica dental LAVADENT”, que permita tener un buen manejo de</b>	<ul style="list-style-type: none"><li>• <b>Módulo de programación de citas</b>, comprende el registro de la selección de horarios y citas para los pacientes por médico.</li><li>• <b>Módulo de seguimiento de historias clínicas</b>; dentro de la cual se realizan el control de resultados y de tratamientos hechos a los pacientes.</li><li>• <b>Módulo de control de pagos</b>, permitirá el monitoreo a las cuotas pagadas por los pacientes.</li></ul>

<p>información odontológica en constante actualización, y así garantizar la preferencia de antiguos y nuevos pacientes.</p>	<ul style="list-style-type: none"> <li>• <b>Módulo de inicio de sesión</b></li> </ul>
<p><b>“SISTEMA DE ADMINISTRACION Y CONTROL DE HISTORIALES CLINICOS DE LA U.M.S.A”, Rosmery Lozano Flores, 2014, Universidad Mayor de San Andrés, Carrera de Informática, presenta el Desarrollo de un Sistema de Administración y Control de Historiales Clínicos de los pacientes universitarios, que permite mejorar las tareas de admisión, consulta, búsqueda y elaboración de reportes o informes de manera más rápida y confiable, reduciendo el tiempo en el registro de datos y manejo de la información de los consultorios de la U.M.S.A.</b></p>	<ul style="list-style-type: none"> <li>• <b>Módulo de Admisión</b>, Proceso por el cual se registra pacientes para ser atendidos en los consultorios.</li> <li>• <b>Módulo de Cita médica</b>, Proceso por el cual comienza con el medico verifica el número de pacientes(universitarios) para atender</li> <li>• <b>Módulo de Consulta Médica</b>, Proceso por el cual se asigna pacientes para ser atendidos en los consultorios.</li> <li>• <b>Módulo de Historial clínico</b>, Proceso por el cual comienza con el médico que verifica, registra y actualiza hojas clínicas.</li> <li>• <b>Módulo de Hoja de Enfermería Registro Medico</b>, Proceso comienza cuando la enfermera verifica la receta médica, proporciona atención al paciente y registra en hoja de enfermería.</li> <li>• <b>Módulo de Reportes</b>, Proceso comienza cuando el jefe médico registra y verifica los informes enviados por la enfermera y el médico.</li> </ul>



<p><b>“SISTEMA WEB DE SEGUIMIENTO A HISTORIAS CLINICAS DE LA EMPRESA SPA MEDICO CIME BASADO EN CRM”,</b> María Leonor Gonzales, 2014, Universidad Mayor de San Andrés, Carrera de Informática, presenta el desarrollo de un Sistema Web de Seguimiento a Historias Clínicas para la empresa SPA Medico CIME, que permita, a través del modelado de negocio CRM fidelizar a clientes antiguos y ganar nuevos clientes a través del manejo de información en una constante actualización, y así mejorar el manejo de historiales clínicos dentro del SPA Medico CIME, y de esta manera fidelizar a los clientes.</p>	<ul style="list-style-type: none"> <li>• <b>Módulo de reservas de tratamientos,</b> comprende el registro de tratamientos a los que se inscribe un paciente y a la selección de horarios de sesiones para los pacientes por médico y tratamiento. 11</li> <li>• <b>Módulo de seguimiento a historias clínicas;</b> dentro del cual se realizará el control de resultados de los tratamientos recibidos por lo pacientes.</li> <li>• <b>Módulo de control de materiales,</b> permite que los administrativos, estén siempre informados de la cantidad de materiales existentes y requeridos para la realización de tratamientos.</li> <li>• <b>Módulo de control de pagos,</b> permitirá el monitoreo a las cuotas pagadas por los pacientes.</li> <li>• <b>Módulo de difusión de publicidad,</b> a través del control de las hojas de estadísticas que pertenecen a los historiales clínicos se obtendrán los datos necesarios para la difusión dinámica y personalizada al paciente.</li> </ul>
<p><b>“SISTEMA WEB DE SEGUIMIENTO Y CONTROL DE PACIENTES INTERNOS” CASO: HOSPITAL ARCO IRIS,</b> Wendy Quiroga, 2014, proyecto fue desarrollado en el Hospital “Arco Iris” en el sector de la Unidad de</p>	<ul style="list-style-type: none"> <li>• <b>Registro de los datos del paciente,</b> así como el registro de ingreso, datos de la internación, asignación de infraestructura, seguimiento del paciente, registro de medicamentos administrados y el egreso del paciente, esto a cargo de los doctores.</li> </ul>

<p><b>Cuidados Intensivos Neonatales (UCIN), que realizan el cuidado de los neonatos ayudándoles así a poder registrar virtualmente las historias clínicas.</b></p>	<ul style="list-style-type: none"> <li>• <b>Ubicación del historial clínico</b>, desechando medios digitales y desechando las historias en papel que ocupaban lugares físicos ayudando a los doctores como al personal del U.C.I.N. a la facilidad de acceder a los historiales.</li> <li>• <b>Actualización datos del paciente</b>, tener mayor control y no tener límites para expresar todo lo que se refiere al paciente.</li> <li>• <b>Consultas</b>, estas nos permitirán ver, manejar todo el historial clínico del paciente</li> </ul>
---	--

*Tabla 2. 1: Proyectos Similares*

*Fuente: Elaboración Propia*

### 3. PLANTEAMIENTO DEL PROBLEMA

#### 3.1. PROBLEMA CENTRAL

Los doctores Parra y Mercado afirman que, “Los consultorios particulares en su mayoría no llevan un registro exhaustivo de sus casos de manera regular y menos aún de manera digital”, (Parra, y Mercado, comunicación personal, enero 2020).

Por esto sabemos las condiciones en las que trabajan los profesionales en odontología en la gran mayoría de los casos, por lo que dar el salto a registros digitales conllevaría que todos realizarán una inversión para adquirir software que permita registrar los casos que tratan durante su actividad laboral, por lo que esto limita a que puedan contar con registros digitales de estos para poder compartirlos.

Además, la Sociedad Boliviana de Ortodoncia y Ortopedia Dentofacial busca aportar en la formación de profesionales, pero no cuenta con un espacio en el cual se puedan compartir casos de ortodoncia y ortopedia dentofacial cuyo tratamiento puede ser útil y recibir comentarios de otros profesionales acerca de estos.

Por lo que se genera la interrogante:

¿Cómo compartir casos ortopédicos dentofaciales o de ortodoncia tratados por odontólogos acreditados por la Sociedad Boliviana de Ortodoncia y Ortopedia Dentofacial – La Paz?

### **3.2. PROBLEMAS SECUNDARIOS**

- La sociedad no cuenta con una plataforma digital en la cual profesionales acreditados puedan contribuir con tratamientos de casos de ortodoncia y ortopedia dentofacial, por lo que estos casos quedan sin ser compartidos como aporte con sus colegas.
- No existe un espacio en el cual profesionales pueden realizar comentarios sobre tratamientos compartidos, por lo que hace que no haya un intercambio de ideas y experiencias.
- El problema de estándares de modelo de registro físico de casos de ortodoncia y ortopedia dentofacial, hace que no se tenga una guía de cómo estos deben ser registrados de manera digital.
- No se cuenta con un espacio para el uso de multimedia como imágenes y radiografías digitales de tratamientos, por lo que no pueden ser visualizadas de manera sencilla.
- Los tratamientos compartidos son de carácter académico y profesional, y si de ser compartidos por algún medio podrían estar al alcance de personas no acreditadas por la Sociedad Boliviana de Ortodoncia y Ortopedia Dentofacial – La Paz.

## **4. DEFINICIÓN DE OBJETIVOS**

### **4.1. OBJETIVO GENERAL**

Desarrollar un Sistema Web de Gestión Documental que permita a odontólogos acreditados por la Sociedad Boliviana de Ortodoncia y Ortopedia Dentofacial – La Paz, compartir casos ortopédicos dentofaciales o de ortodoncia en un archivo digital.

## **4.2. OBJETIVOS ESPECÍFICOS**

- Proporcionar a la SBO – La Paz una plataforma que brindar el servicio de archivo para contribuir y compartir casos de ortodoncia y ortopedia dentofacial.
- Generar un espacio en el cual se puedan realizar comentarios a los casos de ortodoncia y ortopedia compartidos por colegas.
- Mostrar una guía de pasos a seguir para poder realizar el registro digital de un caso de ortodoncia y ortopedia dentofacial.
- Facilitar la integración de archivos multimedia como imágenes y radiografías digitales a los archivos digitales que se generen.
- Otorgar a la SBO – La Paz la potestad de otorgar credenciales a profesionales acreditados por ésta para el acceso a los archivos históricos.

## **5. JUSTIFICACIÓN**

### **5.1. JUSTIFICACIÓN ECONÓMICA**

Debido a que se trabajará con software libre, esto implica que el código fuente estará abierto a cualquier desarrollador involucrado con la institución, por lo tanto, esto implica independencia total en cuanto a licencia y de desarrollo. Siendo esta una ventaja al tener software propietario ya que no se deberán hacer gastos adicionales en software de recursos para administrar el sistema. Además, que el acceso que da la Sociedad profesional, hace que estos no inviertan una parte de sus ingresos en software para registro de casos de manera digital.

### **5.2. JUSTIFICACIÓN SOCIAL**

La Sociedad Boliviana de Ortodoncia y Ortopedia Dentofacial, tiene como objetivo consolidar la formación práctica y académica del profesional de la ortodoncia, por lo que una herramienta donde los profesionales puedan contribuir y acceder a diferentes casos tratados

y compartidos por otros colegas es de mucha utilidad para que estos cuenten con documentación de fácil acceso y de esta manera poder contar con una plataforma que provea de información útil para su desempeño profesional y de esta forma mejorar la calidad de tratamientos que estos brindan.

### 5.3. JUSTIFICACIÓN TECNOLÓGICA

El sistema desarrollado, trabajara con el framework de desarrollo en front-end **Angular** basado en JavaScript que permite implementar el concepto de *Single Page Application* (Aplicación de una sola página) orientado a un único punto de entrada a la web haciendo que la carga en el navegador sea solo una vez, optimizando la fluidez de ejecución del sistema web. Así mismo el uso de este framework permitirá aplicar al sistema web lo que se conoce como *responsive web design* (diseño web adaptable) para que el sistema web sea adaptable a teléfonos celulares y tables haciendo el alcance del software por tanto, para los especialistas no será estrictamente necesario tener una computadora, únicamente acceso a internet, ya que gracias a toda la nueva tecnología que actualmente existe como teléfonos móviles, tabletas, computadoras portátiles, a la que muchos ya tienen acceso. De esta forma actualizándose en lo que, a nuevas tecnologías de información y comunicación. Así estos poder realizar aportes usando cualquier herramienta que la actualidad brinda. También se hará uso del framework de desarrollo back-end **Laravel** para la construcción de una API-Restful que se encarga del procesamiento de información, rutas, seguridad y establecer comunicación con el servidor de base de datos **MySQL** para el almacenamiento de datos de los profesionales acreditados por la sociedad y los casos a registrar en el archivo,

## 6. ALCANCES Y LÍMITES

### 6.1. ALCANCES

Los alcances del sistema web de archivos históricos de casos de ortodoncia y ortopedia dentofacial, se definen a través de módulos que se encargan de gestionar estos archivos

históricos para que los profesionales sean capaces de contribuir y tener acceso, descrita a continuación:

- Módulo de administración de usuarios, que comprende el control de credenciales otorgadas por parte de la SBO – La Paz a profesionales acreditados.
- Módulo de administración de contribuciones de un profesional acreditado, en el cual genera y administra sus archivos.
  - Submódulo de captura multimedia, el cual permitirá subir o capturar las fotografías requeridas para la creación de un archivo.
- Módulo de archivo, que comprende un espacio en el cual se pueden acceder a los archivos compartidos por todos los profesionales.
  - Submódulo de comentarios, que generara para cada archivo en el cual colegas podrán dejar opiniones o sugerencias.
- Módulo de inicio de sesión y seguridad

## **6.2. LÍMITES**

Los límites del sistema web de archivos históricos de casos de ortodoncia y ortopedia dentofacial son:

- Las interfaces del sistema estarán restringidas a los tipos de usuarios autorizados del sistema.
- Otra limitante abocada más a los servicios de web de cada navegador, el sistema web no controlara la compatibilidad o ausencia de plugin o complementos necesarios para la funcionalidad que proporcione el sistema.
- El sistema web podrá ser accedido por los profesionales únicamente a través de internet por las características que se plantea obtener.

## **7. APORTES**

### **7.1. PRÁCTICO**

El sistema web de archivos históricos, lograra aportes a la Sociedad Boliviana de Ortodoncia y Ortopedia Dentofacial – La Paz en términos de:

- Agilización de búsqueda y obtención de información de casos contribuidos por otros profesionales.
- Brindar un formato de registro digital de casos de ortodoncia y ortopedia dentofacial.
- Otorgar una herramienta que permita la contribución por parte de profesionales que quieran realizar un aporte a su comunidad.

### **7.2. TEÓRICO**

Repositorio archivístico de Kardex de pacientes de ortodoncia y ortopedia dentofacial como archivo central mediante el sistema web está orientado a la implementación digital de gestión documental de casos de ortodoncia y ortopedia dentofacial, que es desarrollado con la metodología ágil Scrum el cual es un proceso que se aplica de manera regular un conjunto de buenas prácticas, para trabajar colaborativamente en equipo. Para el modelado del sistema web, la metodología UWE basada en el modelo UML, proporciona los elementos y diagramas necesarios para el buen desarrollo del modelado del sistema, basado en las necesidades que tienen los especialistas en ortodoncia y ortopedia dentofacial para registrar archivos históricos de casos tratados.

## **8. METODOLOGÍA**

### **8.1 MÉTODO DE INVESTIGACIÓN**

La metodología dedicada a este proyecto está basada en la método de investigación descriptiva la cual nos ayuda a observar haciendo el estudio en la Sociedad Boliviana de Ortodoncia y Ortopedia Dentofacial – La Paz, acerca de la gestión documental de casos de

ortodoncia y ortopedia que desea implementar, adquiriendo la información necesaria, y así poder determinar las falencias existentes, para luego poder desarrollar un sistema el cual solucione los problemas, para así poder concluir si el sistema podrá subsanar las falencias encontradas en el la gestión documental de la Sociedad Boliviana de Ortodoncia y Ortopedia Dentofacial – La Paz.

La investigación descriptiva consiste en la caracterización de un hecho, fenómeno, individuo o grupo, con el fin de establecer su estructura de comportamiento. Los resultados de este tipo de investigación se ubican en un nivel intermedio en cuanto a la profundidad de los conocimientos se refiere. (Arias, 2012)

Conocida también como la investigación estadística, ya que describen los datos, el objetivo de esta investigación consiste en llegar a conocer las situaciones, costumbres y actitudes a través de la descripción exacta de las actividades y procesos, el objetivo principal es saber él porque y para que se está realizando. La investigación descriptiva nos permite averiguar:

- ¿Cómo se puede compartir casos de ortodoncia y ortopedia dentofacial?
- ¿De qué manera se puede manejar la información de forma rápida evitando pérdida de tiempo?
- ¿Cómo se puede conservar casos tratados y que estos reciban comentarios de colegas especialistas en ortodoncia?

## **8.2 METODOLOGÍA DE DESARROLLO**

La metodología a usar es la de *scrum*, este es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante el proyecto.

En este proceso se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto, scrum también se utiliza cuando es necesario identificar y solucionar ineficiencias.



Este método de trabajo promueve la innovación, motivación que forma parte de este proyecto, los beneficios que se tiene esta metodología son el cumplimiento de las expectativas, flexibilidad en cambios, mayor calidad de software, reducción de riesgos.

Las características del proceso de Scrum en el proyecto son:

La primera y última fase que consiste en procesos definidos, se observara la implementación de gestión documental en la Sociedad Boliviana de Ortodoncia y Ortopedia Dentofacial – La Paz.

Se desarrolla iteraciones llamadas “*Sprint*”, el equipo de desarrollo decide que funcionalidad incluir o no, el tiempo necesario para terminar el proyecto.

Por otra parte, se trabajará con la metodología UWE que está especializada en la especificación de aplicaciones adaptativas y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario.

Siguiendo las fases que tiene la UWE primeramente se adquirirán las características que el sistema web de gestión documental de casos de ortodoncia y ortopedia dentofacial debe tener, se hará un análisis de requerimientos si los requisitos que tiene la institución se cumplirán, posteriormente se hará la programación con lo especificado anteriormente, se realizaran pruebas para su correcto funcionamiento y poder llevarlo al proceso de instalación para ser utilizado por profesionales acreditados por la Sociedad Boliviana de Ortodoncia y Ortopedia Dentofacial – La Paz, con el mantenimiento se mejorara y controlara el sistema ya desarrollado e instalado

## **9. MARCO TEÓRICO**

### **9.1 GESTION DOCUMENTAL**

La gestión documental consiste en la captura, almacenamiento y recuperación de documentos. Cualquier organización ya captura, almacena y recupera documentos todos los días. Los documentos llegan a la empresa en forma de papel y también en formato electrónico.

La industria de los sistemas para la gestión de documentos electrónicos se encarga del desarrollo de tecnologías para tareas tan importantes en el mundo de la información como el manejo de documentos en formato digital, la conversión a dicho formato de los documentos de papel, las publicaciones electrónicas y la gestión de archivos automatizados. (García, 2001).

#### **9.1.1 ARCHIVO CENTRAL**

Unidad administrativa que coordina y controla el funcionamiento de los archivos de gestión y reúne los documentos transferidos por los mismos una vez finalizado su trámite y cuando su consulta es constante.

Se conforman con la documentación que deja de ser utilizada de forma frecuente. Esta fase es de 10 años, pero los archiveros pueden establecer menores plazos en función de muchos factores siempre teniendo en cuenta la finalización de los trámites.

#### **9.1.2 DOCUMENTO ELECTRONICO**

A pesar de que un documento puede valerse de cualquier medio para existir, este se gestiona como una unidad de información independientemente de su formato o el medio en que se almacene.

El término "documento electrónico" es un concepto relativamente nuevo y no menos debatido que el visto anteriormente, se denominará un documento como electrónico si se encuentra físicamente almacenado en un dispositivo electrónico o unidad de almacenamiento

externa de un dispositivo de este tipo, comprensible sin ningún procesamiento adicional, excepto la presentación del monitor o de la página impresa. Cualquier archivo de computadora que tenga un contenido válido para el trabajo de un universo de usuarios, por reducido que sea, se considera un documento electrónico. (García, 2001)

#### **9.1.2.1 PRINCIPALES TECNOLOGIAS PARA LA GESTION DE DOCUMENTOS ELECTRONICOS**

Las tecnologías para la gestión de documentos electrónicos pueden clasificarse de manera general según estas categorías:

- **Gestión de documentos**, El software para la gestión de documentos está orientado a resolver un problema muy común por estos días en las organizaciones pues es difícil recuperar y gestionar información de una manera eficiente sin una estructura común, creada mediante la computadora, el uso de Internet y de las intranets. (García, 2001)
- **Gestión del conocimiento**, el conocimiento no es una entidad física y tangible, la gestión del conocimiento no es realmente un producto, sino un proceso, una forma de relacionar la información y la experiencia, y de colocar el producto obtenido a disposición de todos los trabajadores de la organización para que estos realicen su trabajo de una manera más eficiente. (García, 2001)
- **Imaging**, el término "*imaging*" se ha utilizado para definir el hecho de crear imágenes en formato digital a partir de documentos impresos, así como de asociarles índices, almacenarlos, recuperarlos y distribuirlos como imágenes electrónicas de los originales. (García, 2001)
- **Flujos de trabajo**, las tecnologías para el flujo de trabajo o *workflow* se han diseñado para automatizar ciertos procesos de trabajo internos de la organización. Estos sistemas se apoyan en versiones electrónicas de los documentos que se mueven a través de diferentes departamentos de la organización para su consulta o modificación por diferentes miembros de esta. (García, 2001)

- **Sistemas para el almacenamiento**, todo sistema necesita de formas eficientes de archivar documentos y datos, así como de colocarlos a disposición de sus usuarios; es por ello que el almacenamiento y recuperación es un componente esencial en el desarrollo de los sistemas para la gestión de documentos, esto para todos está claro. (García, 2001)
- **Gestión**, el concepto de gestión de archivos no es nuevo. Se dice que desde que existieron registros para almacenar, existió la gestión de archivos. Como registro se entiende, a los efectos de este trabajo, cualquier información relativa al trabajo de una organización. Todo documento que emplea una organización puede considerársele como un registro archivado. (García, 2001)

#### **9.1.2.2 VENTAJAS DE LOS SISTEMAS PARA LA GESTION DE DOCUMENTOS ELECTRONICOS**

Con todo lo descrito hasta el momento, sin restar un mínimo de importancia a los sistemas para la gestión de documentos en los formatos tradicionales, sobre todo aquellos que se realizan para los documentos en papel, se hace evidente que uno de los campos de la ciencia de la información y la bibliotecología en que más directamente han influido las tecnologías en los últimos tiempos es en la gestión de los documentos electrónicos.

Así, la aparición y masificación de los documentos electrónicos a través de las tecnologías ha permitido: reducir el espacio físico de archivo, mantener un único conjunto de información que contenga todo aquello que fue o es importante para la organización, posibilitar la localización rápida por una gran diversidad de criterios, facilitar el procesamiento paralelo de información contenida en documentos, en lugar de que el usuario deba esperar porque otro termine con ellos para consultarlos, preservar los documentos originales, con la eliminación de los riesgos que trae consigo el deterioro de los documentos por su uso, integrar la documentación en procesos de trabajo, controlar versiones, eliminar las copias incontroladas, distinguir el papel en circulación, controlar el acceso al archivo por niveles de seguridad, eliminar la pérdida de documentación, mejorar, de forma general, el manejo de información

que puede resultar determinante para el desarrollo de la organización, relacionar información por grupos y tipos de usuario y evitar redundancia en la documentación. (García, 2001)

## 9.2 REGISTROS ORTODONCICOS

Para Vigo (2013), “En la especialidad de Ortodoncia y Ortopedia Dento-facial es necesario documentar y tener registros clínicos previos al tratamiento por dos razones: primero porque necesitamos conocer la situación inicial del paciente en cada caso clínico, y segundo porque estos registros son información complementaria al examen clínico facial y dental”.

Como ejemplo, en la Clínica Facal de Ortodoncia se hacen tres tipos de registros obligatorios y previos al diagnóstico de cada caso antes de empezar cualquier tratamiento ortodóncico.

- **Radiografías del Cráneo y de los huesos de la Cara:** Se toman radiografías de los huesos de la cara y cráneo para analizar las proporciones faciales-esqueléticas y no sólo de la relación dental.
- **Modelos de las arcadas dentarias superior e inferior:** Registro de mordida o de relación interdental y registro de la Articulación Temporo-Mandibular (ATM) en su posición fisiológica adecuada por medio del uso de articuladores semiajustables.
- **Fotos faciales y dentales** antes de iniciar el tratamiento ortodóncico.
- **Axiografía:** Registro complementario durante el tratamiento para verificar la posición de la ATM en un estado de estabilidad y reproducibilidad.

Según Loynes (2019), “Cuando una persona se convierte en candidata para un tratamiento de ortodoncia, su dentista considera sus registros cuando reúne un plan de tratamiento. El dentista u ortodoncista reunirá todas las posibles opciones de tratamiento del paciente.

Los registros de ortodoncia también se pueden usar a lo largo del tratamiento para realizar un seguimiento del progreso a lo largo del tiempo, y también se convierten en parte de los registros dentales de una persona”.

Tipos de registros dentales de ortodoncia

- **Radiografías panorámicas**, una radiografía panorámica , también conocida como panorex o pan, es bidimensional y muestra las mandíbulas y los dientes superiores e inferiores en la misma pieza de película.
- **Radiografías cefalométricas**, la radiografía cefalométrica , o cef para abreviar, es una radiografía que captura el perfil lateral de una persona desde la parte superior de su cabeza hasta justo debajo de la punta de la barbilla.
- **Fotos intraorales y extraorales**, se toman fotos dentales intraorales (dentro de la boca) y extraorales (fuera de la boca) de los dientes y la cara de una persona. Estas fotos se utilizan para determinar la clasificación externa actual del perfil de una persona, así como un registro permanente para mantenerlo archivado.

### 9.3 INGENIERÍA DE SOFTWARE

Según Sommerville (2005), para muchas personas el software son solo programas de computadora, sin embargo, nos comenta que son todos aquellos documentos asociados a la configuración de datos que se necesitan para hacer que estos programas operen de manera adecuada. Estos productos de software se desarrollan para algún cliente en particular o para un mercado en general. Para el diseño y desarrollo de proyectos de software se aplican metodologías, modelos y técnicas que permiten resolver los problemas. En los años 50 no existían metodologías de desarrollo, el desarrollo estaba a cargo de los propios programadores. De ahí la importancia de contar con analistas y diseñadores que permitieran un análisis adecuado de las necesidades que se deberían de implementar.

El objetivo principal que busca la ingeniería de software es convertir el desarrollo de software en un proceso formal, con resultados predecibles, que permitan obtener un producto final de alta calidad y satisfaga las necesidades y expectativas del cliente. Según Gacitúa (2003), la Ingeniería de Software es un proceso intensivo de conocimiento, que abarca la captura de requerimientos, diseño, desarrollo, prueba, implantación y mantenimiento. Generalmente a partir de un complejo esquema de comunicación en el que interactúan usuarios y desarrolladores, el usuario brinda una concepción de la funcionalidad esperada y el

desarrollador especifica esta funcionalidad a partir de esta primera concepción mediante aproximaciones sucesivas. Este ambiente de interacción motiva la búsqueda de estrategias robustas para garantizar que los requisitos del usuario serán descubiertos con precisión y que además serán expresados en una forma correcta y sin ambigüedad, que sea verificable, trazable y modificable.

### **9.3.1 ETAPAS DEL PROCESO**

De acuerdo con Pressman (2002), “las etapas metodológicas a llevar a cabo para el desarrollo de Sistemas de Información, se establecen de la siguiente manera:

**Análisis de los requisitos del software:** El proceso de reunión de requisitos se intensifica y se centra especialmente en el software. Dentro del proceso de análisis, es fundamental que, a través de una colección de requerimientos funcionales y no funcionales, el desarrollador o desarrolladores del software comprendan completamente la naturaleza de los programas que deben construirse para desarrollar la aplicación, la función requerida, comportamiento, rendimiento e interconexión. Es de suma importancia que antes de empezar a codificar los programas, se tenga una completa y plena comprensión de los requisitos del software. “

Pressman establece que, “la tarea del análisis de requisitos es un proceso de descubrimiento, refinamiento, modelado y especificación. Se refina en detalle el ámbito del software, y se crean modelos de los requisitos de datos, flujo de información y control, y del comportamiento operativo. Se analizan soluciones alternativas y se asignan a diferentes elementos del software. El análisis de requisitos permite al desarrollador o desarrolladores especificar la función y el rendimiento del software, indica la interfaz del software con otros elementos del sistema y establece las restricciones que debe cumplir el software”.

**El análisis de requisitos del software puede dividirse en cinco áreas de esfuerzo, que son:**

- **Reconocimiento del problema.** Reconocer los elementos básicos del problema tal y como los perciben los usuarios finales.

- **Evaluación y síntesis.** Definir todos los objetos de datos observables externamente, evaluar el flujo y contenido de la información, definir y elaborar todas las funciones del software, entender el comportamiento del software en el contexto de acontecimientos que afectan al sistema.
- **Modelado.** Crear modelos del sistema con el fin de entender mejor el flujo de datos y control, el tratamiento funcional y el comportamiento operativo y el contenido de la información.
- **Especificación.** Realizar la especificación formal del software.
- **Revisión.** Un último chequeo general de todo el proceso.

**Diseño:** Según Pressman (2002), “El diseño del software es realmente un proceso de muchos pasos pero que se clasifican dentro de uno mismo. En general, la actividad del diseño se refiere al establecimiento de las estructuras de datos, la arquitectura general del software, representaciones de interfaz y algoritmos. El proceso de diseño traduce requisitos en una representación de software. El diseño es el primer paso en la fase de desarrollo de cualquier producto o sistema de ingeniería. De acuerdo con Pressman, el objetivo del diseño es producir un modelo o representación de una entidad que se va a construir posteriormente. El diseño, es la primera de las tres actividades técnicas que implica un proceso de ingeniería de software; estas etapas son diseño, codificación y pruebas. Generalmente la fase de diseño produce un diseño de datos, un diseño arquitectónico, un diseño de interfaz, y un diseño procedimental”.

**Generación de Código:** Según Pressman (2002), “Esta actividad consiste en traducir el diseño, en una forma legible por la máquina. La generación de código se refiere tanto a la parte de generación de los ambientes virtuales, como a la parte en la cual se añadirá comportamiento a estos ambientes.”

**Pruebas:** Según Pressman (2002), “Una vez que se ha generado código, comienzan las pruebas del software o sistema que se ha desarrollado. De acuerdo con Pressman, el proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las



sentencias se han comprobado, y en los procesos externos funcionales, es decir, la realización de las pruebas para la detección de errores. En el caso de una herramienta de software, es necesario tener etapas de pruebas tanto para la parte funcional del software, como para la parte aplicativa del mismo<sup>2</sup>.

**Mantenimiento:** Según Pressman (2002), “El software indudablemente sufrirá cambios, y habrá que hacer algunas modificaciones a su funcionalidad. Es de suma importancia que el software de calidad pueda adaptarse con fines de acoplarse a los cambios de su entorno 18 externo. Por medio de la documentación apropiada y atinada del software se pueden presentar las vías para el mantenimiento y modificaciones al mismo”.

### **9.3.2 MODELOS DE PROCESOS DE SOFTWARE**

#### **9.3.2.1 MODELOS TRADICIONALES**

El modelo está formado por un conjunto de fases o actividades en las que no tienen en cuenta la naturaleza evolutiva del software, las cuales son:

**Ciclo de vida** También se le conoce como modelo lineal secuencia o modelo en cascada. Plantea un enfoque sistemático, secuencial para el desarrollo de software, que comienza en un nivel de sistemas y continúa con el análisis, diseño, codificación, pruebas y mantenimiento. Este modelo comprende una primera actividad como lo es La Ingeniería y modelado de sistemas / información, en el cual se establece el sistema de nivel superior y se deben establecer los requisitos de la empresa en la que se encuentra. Los requisitos se recogen del sistema con una pequeña parte de análisis y diseño

**Basado en Prototipos** Este paradigma se inicia con la recolección de requerimientos. El desarrollador y el cliente encuentran y definen los objetivos globales para el software, identifican los requisitos conocidos y las áreas del esquema en donde es obligatoria más definición. Luego aparece un “diseño rápido”. El diseño rápido está centrado en una representación de los aspectos del software que serán visibles para el usuario-cliente. El diseño rápido lleva a la construcción de un prototipo. El prototipo lo evalúa el cliente-usuario

y lo utiliza para refinar los requisitos del software a desarrollar. La interacción ocurre cuando el prototipo satisface las necesidades del cliente, a la vez que permite que el desarrollador comprenda mejor lo que se necesita hacer.

**Modelo DRA** El Desarrollo Rápido de Aplicaciones (DRA) es un modelo de proceso del desarrollo del software lineal secuencial que enfatiza un ciclo de desarrollo extremadamente corto. El modelo DRA es una adaptación a “alta velocidad” del 19 modelo lineal secuencial en el que se logra el desarrollo rápido utilizando un enfoque de construcción basado en componentes. (Pressman, 2002)

### **9.3.2.2 MODELOS EVOLUTIVOS**

El software al igual que todos los sistemas complejos, evoluciona con el tiempo. Los requisitos de gestión y de productos a menudo cambian conforme a que el desarrollo proceda haciendo que el camino que lleva al producto final no sea real; las estrictas fechas tope del mercado hacen que no sea posible finalizar un producto completo, por lo que se debe introducir una versión limitada para cumplir la presión competitiva y de gestión; se comprende perfectamente el conjunto de requisitos de productos centrales o del sistema, pero todavía se tienen que definir los detalles de extensiones del producto o sistema. En estas y en otras situaciones similares, los ingenieros del software necesitan un modelo de proceso que se haya diseñado explícitamente para acomodarse a un producto que evolucione con el tiempo. Los modelos que se adaptan a la evolución son:

- Modelo Espiral
- Evolutivo
- Incremental
- Modelo de desarrollo concurrente

### **9.3.2.3 MODELOS PARA SISTEMAS ORIENTADO A OBJETOS**

Es la construcción de modelos de un sistema por medio de la identificación y especificación de un conjunto de objetos relacionados, que se comportan y colaboran entre sí de acuerdo a

los requerimientos establecidos para el sistema de objetos. Son modelos con alto grado de interactividad y solapamiento entre fases, como ser:

- De agrupamiento
- Fuente
- Basado en Componentes
- Proceso Unificado

#### **9.3.2.4 PROCESOS AGILES**

Un proceso es ágil cuando el desarrollo de software es incremental (entregas pequeñas de software, con ciclos rápidos), cooperativo (cliente y desarrolladores trabajan juntos constantemente con una cercana comunicación), sencillo (el método en sí mismo es fácil de aprender y modificar, bien documentado), y adaptable (permite realizar cambios de último momento). Entre las metodologías ágiles identificadas son:

- *Extreme Programming (XP)*
- Scrum
- Familia de Metodologías *Crystal*
- *Feature Driven Development*
- *Rational Unified Process*, una configuración ágil
- *Dynamic Systems Development Method*
- *Adaptive Software Development*
- *Open Source Software Development*

#### **9.4 METODOLOGÍA DE DESARROLLO SCRUM**

Scrum es una metodología ágil de gestión de proyectos de desarrollo de software, basada en un proceso de trabajo constante, iterativo e incremental. Creada por Jeff Sutherland en 1993, de las metodologías ágiles, es la más utilizada, según una encuesta publicada por VersionOne en 2010 realizada a 4770 entrevistados de 91 países. La misma, revela que el 58% de los

encuestados, utiliza Scrum como metodología para la gestión de proyectos de desarrollo de Software.

Jeff Sutherland aplicó el modelo Scrum al desarrollo de software en 1993 en Easel Corporation (Empresa que en los macro-juegos de compras y fusiones se integraría en VMARK, luego en Informix y finalmente en Ascential Software Corporation). En 1996 lo presentó junto con Ken Schwaber como proceso formal, también para gestión del desarrollo de software en OOPSLA 96. Más tarde, en 2001 serían dos de los promulgadores del Manifiesto ágil. En el desarrollo de software Scrum está considerado como modelo ágil por la Agile Alliance.

Sus principales características son:

- Equipos auto dirigidos
- Utiliza reglas para crear un entorno ágil de administración de proyectos
- No prescribe prácticas específicas de ingeniería
- Los requerimientos se capturan como ítems de la lista Product Backlog
- El producto se construye en una serie de Sprints de un mes de duración

## **9.4.1 HERRAMIENTAS DE LA METODOLOGÍA**

### **9.4.1.1 PILA DE PRODUCTO**

La Pila de Producto, o *Product Backlog*, es un artefacto del marco de trabajo para la gestión agile de proyectos de desarrollo de software, SCRUM. Y que es, en líneas generales, una lista ordenada u priorizada de las tareas que componen un proyecto de aplicación.

Aunque SCRUM no lo define, el formato que más se utiliza para la tarjeta de trabajo que compone una Pila de Producto es la Historia de Usuario. Sin que haya mayores problemas en utilizar Casos de Uso, o una lista de tareas.

Lo importante es que el propio esfuerzo de realizar la división en tareas implica una organización del trabajo y una primera visión del alcance del proyecto. Es decir, qué es lo

que se quiere obtener después de semanas o meses de trabajo. La Pila de Producto, según Scrum, es propiedad del Dueño del Producto. Es decir, el cliente final o su representante. Esto suena un tanto utópico ya que es muy difícil encontrar a un cliente que le pueda o quiera dedicar el tiempo y dedicación que requiere una gestión *Agile* de un proyecto.

Teniendo en cuenta que es un artefacto vivo y que va a sufrir modificaciones tanto de prioridad como de alcance durante todo el proyecto, se construye una primera Pila de Producto que comprenda la descripción de las funcionalidades esperadas a alto nivel. En este momento el *Product Backlog* empieza a mostrar sus ventajas.

Es un punto inicial para gerencia, si hubiese que presentar una oferta al cliente, en la estimación en tiempo y dinero del coste del proyecto. Ya que la definición del alcance la genera el propio cliente, o las personas que más saben sobre lo que quieren que realice la futura aplicación.

Por otra parte, el equipo adquiere la primera visión del proyecto. Cuál es el objetivo y la motivación que ha llevado al cliente a pensar en que un desarrollo de software puede ser una ventaja o una solución a sus necesidades. En definitiva, porque es una mejora.

Y el cliente puede observar de forma visual y cómoda todo el trabajo que representa el construir su proyecto, decidir las funciones que más valor le aportan y detectar que cosas puede dejar en duda, por si no fuera necesario o interesante de construir.

Otra gran ventaja de utilizar Pilas de Producto es que permite encapsular una metodología *Agile*, como *Scrum*, dentro de un proceso formal de gestión de proyectos. Es decir, puedo cumplir con los artefactos clásicos de Toma de requisitos, Análisis funcional, Análisis Técnico, etc. Y, simultáneamente construir el *Product Backlog* (aunque con más esfuerzo) para empezar a desarrollar y a obtener software que funcione de forma iterativa.

#### **9.4.1.2 PRODUCT BACKLOG LIST**

La *Product Backlog list* es una lista priorizada que define el trabajo que se va a realizar en el proyecto. Cuando un proyecto comienza es muy difícil tener claro todos los requerimientos

sobre el producto. Sin embargo, suelen surgir los más importantes que casi siempre son más que suficientes para un Sprint.

La *Product Backlog* List puede crecer y modificarse a medida que se obtiene más conocimiento acerca del producto y del cliente. Con la restricción de que solo puede cambiarse entre Sprints. El objetivo es asegurar que el producto definido al terminar la lista es el más correcto, útil y competitivo posible y para esto la lista debe acompañar los cambios en el entorno y el producto.

Existe un rol asociado con esta lista y es el de *Product Owner*. Si alguien quiere realizar cualquier modificación sobre la lista, por ejemplo: agregar o incrementar la prioridad de sus elementos tiene que convencer al *Product Owner*.

#### **9.4.1.3 SPRINTS**

Esta herramienta es el procedimiento de adaptación de las cambiantes variables del entorno (requerimientos, tiempo, recursos, conocimiento, tecnología). Son ciclos iterativos en los cuales se desarrolla o mejora una funcionalidad para producir nuevos incrementos. Durante un *Sprint* el producto es diseñado, codificado y probado. Y su arquitectura y diseño evolucionan durante el desarrollo.

El objetivo de un *Sprint* debe ser expresado en pocas palabras para que sea fácil de recordar y esté siempre presente en el equipo. Es posible definir una serie de restricciones que el equipo deba aplicar durante un *Sprint*.

Un *Sprint* tiene una duración planificada de entre una semana y un mes. No es posible introducir cambios durante el Sprint, por lo tanto, para planificar su duración hay que pensar en cuanto tiempo puedo comprometerme a mantener los cambios fuera del *Sprint*. Dependiendo del tamaño del sistema, la construcción de un release puede llevar entre 3 y 8 *Sprints*. Por otra parte, podrían formarse equipos para desarrollar en forma paralela distintos grupos de funcionalidad.

El ***Sprint Backlog*** Es el punto de entrada de cada Sprint. Es una lista que tiene los ítems de la *Product Backlog* List que van a ser implementados en el siguiente Sprint.

Los ítems son seleccionados por el *Scrum Team*, el Scrum Master y el *Product Owner* en la *Sprint Planning Meeting* a partir de la priorización de los ítems y los objetivos que se marcaron para ese Sprint. A partir de los objetivos a cumplir durante el *Sprint* el *Scrum Team* determina que tareas debe desempeñar para cumplir el objetivo. De esto surge el *Sprint Backlog*. Es importante destacar que es el equipo quien se organiza para alcanzar el objetivo. El *Manager* no asigna tareas a los individuos y tampoco toma decisiones por el equipo. El equipo puede agregar nuevas tareas o remover tareas innecesarias en cualquier momento si lo considera necesario para cumplir el objetivo. Pero el *Sprint Backlog* solo puede ser modificado por el equipo. Las estimaciones se actualizan cada vez que aparece nueva información.

#### **9.4.2 ROLES Y RESPONSABILIDADES**

Para orquestar este proceso, SCRUM distingue actores con diferentes papeles dentro del proceso. De forma general, podemos distinguir propietario del producto o *Product Owner*, master de scrum o *Scrum Master*, equipo de desarrollo o *Scrum Team* y cliente o usuario.

##### **9.4.2.1 PRODUCT OWNER**

El *Product Owner* es la única persona responsable de delinear el producto más valioso posible para la fecha deseada. Esto se logra gestionando el flujo de trabajo hacia el equipo, que a su vez se lleva a cabo seleccionando y refinando ítems del *Product Backlog*. El *Product Owner* mantiene el *Product Backlog* y asegura que todos sepan qué hay en él y cuáles son las prioridades. El *Product Owner* puede ser ayudado por otros individuos, pero el rol debe ser ocupado por una única persona.

Ciertamente, el *Product Owner* no es responsable de todo. El Equipo Scrum completo es responsable de ser lo más productivo posible, de mejorar sus prácticas, de hacer las preguntas correctas, de ayudar al *Product Owner*. El Equipo de Desarrollo es responsable de determinar

cuánto trabajo puede ser tomado en un *Sprint*, y de producir un Incremento de Producto al finalizar el mismo.

De todas formas, el *Product Owner*, en Scrum, se encuentra en una posición única. Suele ser la persona más cercana al “costado del negocio” de todo el proyecto. Es típicamente el encargado de “sacar el producto” y quien se espera hará el mejor trabajo posible en cuanto a satisfacer a todas las partes interesadas. El *Product Owner* lleva adelante esta tarea mediante la gestión del *Product Backlog* y asegurándose que el *Product Backlog* y el avance contra éste se mantengan visibles.

El Product Owner, al decidir sobre qué debe hacer y qué posponer el Equipo de Desarrollo, toma las decisiones de alcance versus fechas que llevan al mejor producto posible.

#### **9.4.2.2 SCRUM MASTER**

El *Scrum Master* es un “líder servicial”, que ayuda al resto del equipo Scrum a seguir su proceso. Debe tener una buena comprensión de Scrum y la habilidad de capacitar a otros en sus sutilezas, trabaja junto al Product Owner para que éste logre crear y mantener el *Product Backlog*, junto al equipo de desarrollo para encontrar e implementar las prácticas técnicas que les permitirán tener un Incremento de producto 'Hecho' al final de cada Sprint. También trabaja con el Equipo Scrum completo para evolucionar la definición de hecho.

El Scrum Master también es responsable de velar por la remoción de los impedimentos al avance del equipo. Estos impedimentos pueden ser externos al equipo, como por ejemplo la falta de apoyo de otro equipo, o internos, como ser que el *Product Owner* no sepa preparar el *Product Backlog* de forma adecuada.

Así también fomenta la auto organización. Los problemas deben ser resueltos por el equipo siempre que sea posible. El *Scrum Master* actúa como coach para el Equipo scrum, ayudando a sus miembros a ejecutar el proceso scrum. Los ayuda a trabajar juntos y a aprender el framework Scrum, al tiempo que los protege de distracciones tanto internas como externas. Puede facilitar reuniones y ayuda a mantener al equipo scrum en el buen camino, productivo



y creciendo en sus capacidades, es responsable de asegurar que Scrum sea comprendido e implementado, tanto dentro como fuera del equipo. Ayuda a personas fuera del equipo a entender el proceso y a comprender qué interacciones con el equipo son valiosas y cuáles no.

#### **9.4.2.3 SCRUM TEAM**

El *Scrum Team* Es un grupo de personas encargadas de desarrollar e implementar la funcionalidad pactada. El equipo se auto administra y auto organiza. Las personas que lo componen, deben utilizar su ingenio para incrementar la funcionalidad cumpliendo con los requerimientos a lo largo de cada iteración. Los miembros del equipo son responsables del éxito de cada *sprint*.

La auto administración y auto organización es un aspecto crucial en Scrum. Es indispensable que todos los miembros del *Team* estén comprometidos con su trabajo y para ello es necesario contar con la motivación de los mismos. En un *Team* que se auto organiza y auto administra, no hay una persona encargada de controlar que todos los miembros estén cumpliendo sus tareas en tiempo y forma. Sino que cada uno de los integrantes es responsable de realizar sus tareas y además debe asegurarse que el resto de los miembros hagan lo propio con sus tareas respectivas. El *Team* debe trabajar en conjunto como un bloque. Sus resultados son producto del trabajo colectivo y no de esfuerzos individuales.

Idealmente, se recomienda que el *Team* esté formado por no más de siete personas. A medida que el *Team* aumenta en cantidad de personas, aumenta la posibilidad que el *Team* no funcione como tal, dado que habrá miembros que perderán la motivación y el compromiso necesario para el correcto funcionamiento del *Team*.

En algunas situaciones, cuando se cuente con un Team formado por más de diez personas, es recomendable dividir este equipo en distintos sub equipos. La subdivisión puede hacerse según las responsabilidades (un equipo de desarrolladores, un equipo de qa, un equipo de analistas funcionales) o según las necesidades del negocio en sub equipos que incluyan todos los roles. Cuando se armen sub equipos formados solamente por desarrolladores, qa o

analistas funcionales, es necesario encontrar una forma para hacer interactuar a los distintos sub equipos. Para ello puede realizarse, semanalmente, una reunión de Scrum con un representante de cada subequipo. De esta forma los equipos estarán al tanto de las actividades de los otros equipos.

La composición del Team es algo que no puede ser absoluta, dado que cada proyecto de *software* tiene sus propias características. Es decir, algunos proyectos necesitarán mayor presencia de personas para ocupar ciertos roles que otros. Por ejemplo, en un proyecto relacionado a *data mining*, es necesario contar con algún experto en dicha tecnología y un administrador de bases de datos, pero en un proyecto empresarial estos roles pueden no ser necesarios. Existe la posibilidad que el objetivo de un sprint sea “Encontrar la mayor cantidad de fallas en el sistema”, con lo cual, en ese caso, el *Team* puede estar formado solamente por personal de *testing*.

También puede ser factible, que el objetivo de un sprint sea “Presentar el diseño de la refactorización necesaria para utilizar una nueva librería de un tercero”, por lo tanto, en ese caso, es muy probable que el *Team* esté formado por un grupo de arquitectos, diseñadores y desarrolladores sin contar con analistas funcionales ni personal de *testing*.

### **9.4.3 PROCESO DE LA METODOLOGÍA**

#### **9.4.3.1 PRE – GAME**

El proceso comienza con la fase de *Pre-game*, en la que se realiza de forma conjunta con el cliente una definición sencilla y clara de las características que debe tener el sistema que vaya a ser desarrollado, definiendo las historias de usuario que van a guiar el proceso de desarrollo la cual incluye dos sub-fases:

- **Planning** Consiste en la definición del sistema que será construido. Para esto se crea la lista *Product Backlog* a partir del conocimiento que actualmente se tiene del sistema. En ella se expresan los requerimientos priorizados y a partir de ella se estima el esfuerzo requerido. La *Product Backlog List* es actualizada constantemente con

ítems nuevos y más detallados, con estimaciones más precisas y cambios en la prioridad de los ítems.

- **Architecture** El diseño de alto nivel del sistema se planifica a partir de los elementos existentes en la *Product Backlog List*. En caso de que el producto a construir sea una mejora a un sistema ya existente, se identifican los cambios necesarios para implementar los elementos que aparecen en la lista Product Backlog y el impacto que pueden tener estos cambios. Se sostiene una *Design Review Meeting* para examinar los objetivos de la implementación y tomar decisiones a partir de la revisión. Se preparan planes preliminares sobre el contenido de cada release.

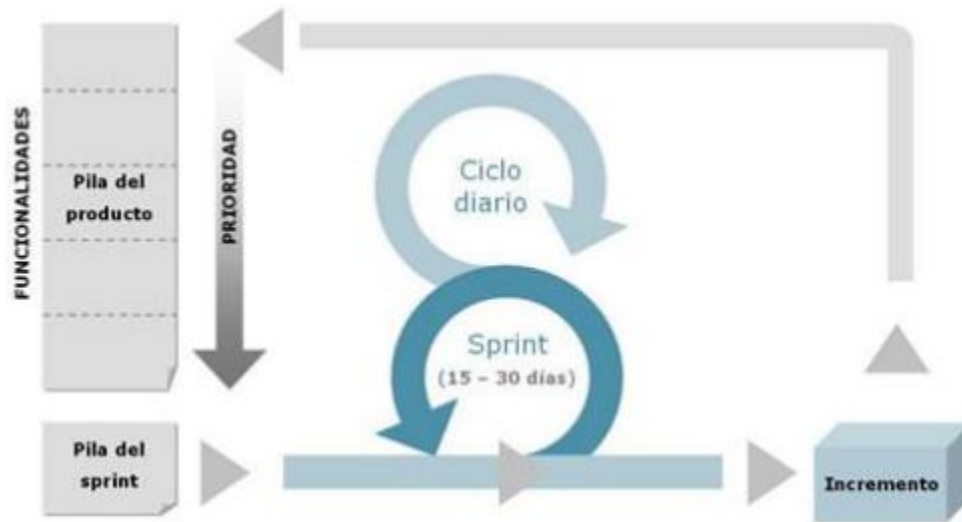
#### 9.4.3.2 GAME

La fase llamada **Game** es la parte ágil del Scrum, en esta fase se espera que ocurran cosas impredecibles. Para evitar el caos Scrum define prácticas para observar y controlar las variables técnicas y del entorno, así también como la metodología de desarrollo que hayan sido identificadas y puedan cambiar. Este control se realiza durante los *Sprints*. Dentro de variables de entorno encontramos: tiempo, calidad, requerimientos, recursos, tecnologías y herramientas de implementación. En lugar de tenerlas en consideración al comienzo del desarrollo, Scrum propone controlarlas constantemente para poder adaptarse a los cambios en forma flexible.

#### 9.4.3.3 POST – GAME

El Post-Game es la fase que contiene el cierre del release. Para ingresar a esta fase se debe llegar a un acuerdo respecto a las variables del entorno por ejemplo que los requerimientos fueron completados. El sistema está listo para ser liberado y es en esta etapa en la que se realiza integración, pruebas del sistema y documentación. El sprint es por tanto el núcleo central que proporciona la base de desarrollo iterativo e incremental tal como se ve en la

Siguiente figura:



*Figura 1: Visión general del proceso de la metodología Scrum*

*Fuente: Schwaber Beedle, 2006*

Los elementos que conforman el desarrollo Scrum son:

### Las reuniones

- **Planificación de *sprint*:** Jornada de trabajo previa al inicio de cada *sprint* en la que se determina cuál va a ser el trabajo y los objetivos que se deben cumplir en esa iteración.
- **Reunión diaria:** Breve revisión del equipo del trabajo realizado hasta la fecha y la previsión para el día siguiente.
- **Revisión de *sprint*:** Análisis y revisión del incremento generado.

### Los elementos

- ***Pila del producto:*** lista de requisitos de usuario que se origina con la visión inicial del producto y va creciendo y evolucionando durante el desarrollo.
- ***Pila del sprint:*** Lista de los trabajos que debe realizar el equipo durante el *sprint* para generar el incremento previsto.

- **Incremento:** Resultado de cada *sprint*

#### 9.4.4 CONTROL DE EVOLUCIÓN DE PROYECTO

Scrum controla de forma empírica y adaptable la evolución del proyecto, empleando las siguientes prácticas de la gestión ágil:

- **Revisión de las Iteraciones:** Al finalizar cada iteración se lleva a cabo una revisión con todas las personas implicadas en el proyecto. Este es el periodo máximo que se tarda en reconducir una desviación en el proyecto o en las circunstancias del producto.
- **Desarrollo incremental:** Durante el proyecto, las personas implicadas no trabajan con diseños o abstracciones. El desarrollo incremental implica que al final de cada iteración se dispone de una parte del producto operativa que se puede inspeccionar y evaluar.
- **Desarrollo evolutivo** Los modelos de gestión ágil se emplean para trabajar en entornos de incertidumbre e inestabilidad de requisitos. En Scrum se toma a la inestabilidad como una premisa, y se adoptan técnicas de trabajo para permitir esa evolución sin degradar la calidad de la arquitectura que se irá generando durante el desarrollo. El desarrollo Scrum va generando el diseño y la arquitectura final de forma evolutiva durante todo el proyecto.
- **Auto-organización:** Durante el desarrollo de un proyecto son muchos los factores impredecibles que surgen en todas las áreas y niveles. La gestión predictiva confía la responsabilidad de su resolución al gestor de proyectos. En Scrum los equipos son auto organizados (no auto dirigidos), con margen de decisión suficiente para tomar las decisiones que consideren oportunas.
- **Colaboración:** Las prácticas y el entorno de trabajo ágiles facilitan la colaboración del equipo. Ésta es necesaria, porque para que funcione la auto organización como un control eficaz cada miembro del equipo debe colaborar de forma abierta con los demás, según sus capacidades y no según su rol o su puesto.

## **9.5 INGENIERÍA WEB**

La ingeniería web se debe al crecimiento desenfrenado que está teniendo la Web está ocasionando un impacto en la sociedad y el nuevo manejo que se le está dando a la información en las diferentes áreas en que se presenta ha hecho que las personas tiendan a realizar todas sus actividades por esta vía.

El desarrollo de aplicaciones Web posee determinadas características que lo hacen diferente del desarrollo de aplicaciones o software tradicional y sistemas de información.

Según Murugesan, Deshpande, Hansen y Ginige., promotores iniciales del establecimiento de la Ingeniería Web como nueva disciplina, dan la siguiente definición:

“Es el proceso utilizado para crear, implantar y mantener aplicaciones y sistemas Web de alta calidad. Esta breve definición nos lleva a abordar un aspecto clave de cualquier proyecto como es determinar qué tipo de proceso es más adecuado en función de las características del mismo.

El desarrollo de aplicaciones Web posee determinadas características que lo hacen diferente del desarrollo de aplicaciones o software tradicional y sistemas de información. La ingeniería de la Web es multidisciplinar.”

### **9.5.1 PROCESO DE INGENIERÍA WEB**

Según Pressman (2001), “las actividades que formarían parte del marco de trabajo incluirían las tareas abajo mencionadas. Dichas tareas serían aplicables a cualquier aplicación Web, independientemente del tamaño y complejidad de la misma.

**Comunicación con el cliente:** La comunicación con el cliente se caracteriza por medio de dos grandes tareas: el análisis del negocio y la formulación. El análisis del negocio define el contexto empresarial organizativo para las WebApps y otras aplicaciones de negocio. La formulación es una actividad de recopilación de requisitos que involucran a todos los participantes.

**Planeación:** Se crea el plan del proyecto para el incremento de la Web App. El plan consiste de una definición de tareas y un calendario de plazos respecto al período establecido para el desarrollo del proyecto.

**Modelado:** Las labores convencionales de análisis diseño de la ingeniería del software se adaptan al desarrollo de las Web App, se mezclan y luego se funden en una actividad de modelado de la Web. El intento es desarrollar análisis rápido y modelos de diseño que definan requisitos y al mismo tiempo representen una Web App que los satisfará.

**Construcción:** Las herramientas y la tecnología Web se aplican para construir la Web App que se ha modelado. Una vez que se construye el incremento de Web App se dirige a una serie de pruebas rápidas para asegurar que se descubran los errores en el diseño.

### 9.5.2 HERRAMIENTAS TECNOLÓGICAS

Las tecnologías abarcan un amplio conjunto de descripción de contenido y lenguaje de modelación, por ejemplo: HTML, VRML, XML, etc. lenguajes de programación por ejemplo java, php, jsp, etc. recursos de desarrollo basados en componentes por ejemplo corba, com, activeX, .net, etc. Frameworks de trabajo para optimizar el desarrollo en diferentes lenguajes, navegadores, herramientas multimedia, herramientas de auditoría de sitio, herramientas de conectividad de base de datos, herramientas de seguridad, servidores y utilidades de servidor, y herramientas de administración y análisis de sitio.

#### 9.5.2.1 PHP

PHP (acrónimo de PHP: *Hypertext Preprocessor*), es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor.

El PHP está enfocado principalmente a la programación de scripts del lado del servidor, por lo que se puede hacer cualquier cosa que pueda hacer otro programa CGI, como recopilar datos de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies. Aunque PHP puede hacer mucho más.

Existen principalmente dos campos principales donde se usan scripts de PHP.

**Scripts del lado del servidor.** Este es el campo más tradicional y el foco principal. Son necesarias tres cosas para que esto funcione. El analizador de PHP (módulo CGI o servidor), un servidor web y un navegador web. Es necesario ejecutar el servidor con una instalación de PHP conectada. Se puede acceder al resultado del programa de PHP con un navegador, viendo la página de PHP a través del servidor. Todo esto se puede ejecutar en su máquina si está experimentado con la programación de PHP. Véase la sección sobre las instrucciones de instalación para más información.

**Scripts desde la línea de comandos.** Se puede crear un script de PHP y ejecutarlo sin necesidad de un servidor o navegador. Solamente es necesario el analizador de PHP para utilizarlo de esta manera. Este tipo de uso es ideal para scripts que se ejecuten con regularidad empleando cron (en \*nix o Linux) o el Planificador de tareas (en Windows). Estos scripts también pueden usarse para tareas simples de procesamiento de texto.

#### **9.5.2.2 LARAVEL**

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti". Fue creado en 2011 y tiene una gran influencia de *frameworks* como *Ruby on Rails*, Sinatra y ASP.NET MVC.2

Laravel tiene como objetivo ser un framework que permita el uso de una sintaxis elegante y expresiva para crear código de forma sencilla y permitiendo multitud de funcionalidades. Intenta aprovechar lo mejor de otros *frameworks* y aprovechar las características de las últimas versiones de PHP.

#### **9.5.2.3 MySQL**

MySQL es actualmente el sistema de gestión de base de datos relacional más popular de código abierto. Para administrar y procesar grandes cantidades de datos, así como acceder a ellos la base de datos, MySQL es una herramienta importante y eficaz.



MySQL es muy utilizado en aplicaciones web, como Joomla, Wordpress, Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación.

#### **9.5.2.4 ANGULAR**

Angular (comúnmente llamado Angular 2+ o Angular 2) es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript.

Angular se basa en clases tipo "Componentes", cuyas propiedades son las usadas para hacer el binding de los datos. En dichas clases tenemos propiedades (variables) y métodos (funciones a llamar).

Angular es la evolución de AngularJS aunque incompatible con su predecesor.

## **9.6 METODOLOGÍA UWE**

La propuesta de Ingeniería Web basada en UML (UWE (Koch, 2000)) es una metodología detallada para el proceso de autoría de aplicaciones con una definición exhaustiva del proceso de diseño que debe ser utilizado. Este proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el Proceso Unificado de Modelado.

UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario.

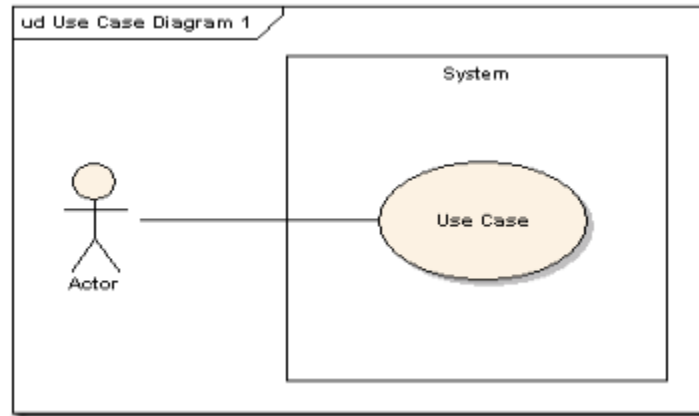
Otras características relevantes del proceso y método de autoría de UWE son el uso del paradigma orientado a objetos, su orientación al usuario, la definición de una meta-modelo (modelo de referencia) que da soporte al método y el grado de formalismo que alcanza debido al soporte que proporciona para la definición de restricciones sobre los modelos.

### **9.6.1 FASES DE LA METODOLOGÍA UWE**

Las fases de la metodología UWE, son procesos o actividades que se utilizan y permiten identificar las necesidades de la aplicación o sistema web a desarrollar; estas actividades se describen y representan en cuatro fases que son:

### 9.6.1.1 ANÁLISIS DE REQUISITOS

Como en otras metodologías, la primera fase o actividad es la del análisis de requisitos funcionales, que permite visualizar los procesos y funciones que debe cumplir el sistema web, esta fase se ve reflejada en los casos de uso.



*Figura 2: Análisis de caso de uso*

*Fuente: M. Gonzales, 2013*

### 9.6.1.2 DISEÑO CONCEPTUAL

El diseño conceptual se basa en el análisis de requisitos del paso anterior. Esto incluye los objetos involucrados entre los usuarios y la aplicación. Este modelo propone construir un



*Figura 3: Modelo de Clases*

*Fuente: M. Gonzales, 2013*

modelo de clases con estos objetos, ignorándolos los aspectos de navegación: Presentación e Interacción, que serán tratados posteriormente. Los principales elementos de modelado son; las clases, asociaciones y

### **9.6.1.3 DISEÑO NAVEGACIONAL**

El diseño navegacional no es solo útil para la generación de la documentación de la estructura de la aplicación, sino que también permite mejorar la estructura de navegabilidad.

El modelo de la navegación comprende de:

- El modelo de espacio de navegación que especifica qué objetos puede ser visitados a través de la aplicación Web.
- El modelo de estructura de navegación que define como se alcanzan estos objetos a través de la Web.
- En el proceso de construir el modelo espacial de navegación las decisiones del diseñador están basadas en el modelo conceptual y los requisitos de la aplicación definidos en el modo de caso de uso.

Cuando hablamos de un sistema web, es necesario conocer la relación y los enlaces entre las páginas web, es por eso que en la fase de diseño se describen a través de diagramas la navegación del sistema cumpliendo con lo que se diseñó en los casos de uso, los elementos que se utiliza para el diseño de diagramas son:






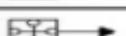
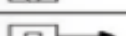



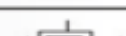
ICONO	DESCRIPCION
	Hiperlance jerarquico inferido
	Hiperlance jerarquico inferido
	Indice
	Visita guiada
	Visita guiada indexada
	Acceso al calorio
	Acceso simultaneo
	Indice Multiple
	Visita guiada multiple
	Visita guiada indexada multiple
	Ejemplo de acceso a partir de una relacion N:M, en este caso utilizando un indice (podria ser cualquier otra primitiva de acceso simple)

Figura 4: Relaciones y enlaces

Fuente: Ingenieria web basada en UML, Instituto de Informática

En la figura a continuación se muestra un ejemplo del diseño navegacional:

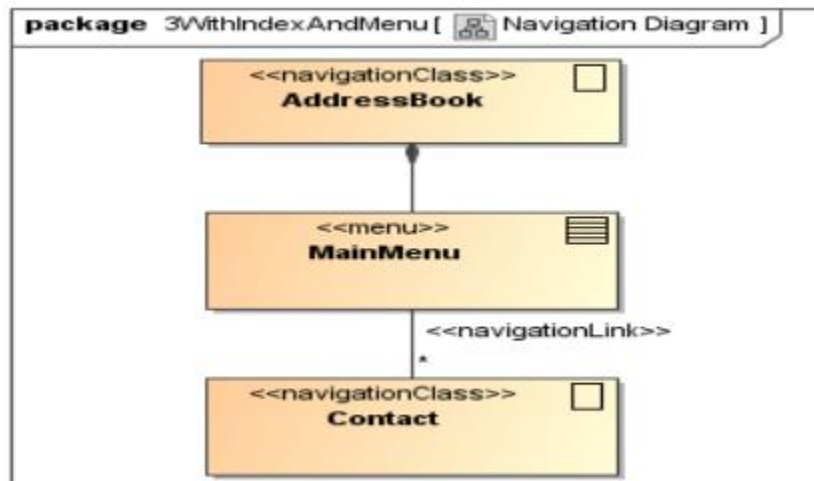


Figura 5: Diseño navegacional UWE

Fuente: Ingenieria web basada en UML, Instituto de Informática

#### 9.6.1.4 DISEÑO DE PRESENTACIÓN

Este modelo permite una visión amplia de los procesos de la página web que se representan en los diagramas de navegación; pueden interpretarse también con las interfaces del sistema web, para el caso se tiene estereotipos o iconos que ayudan al diseño de los diagramas de presentación. El diagrama de presentación de la metodología UWE, permite al usuario comprender y analizar, sobre el área de trabajo al que se someterá con la implantación del sistema. En la siguiente figura, se muestra la aplicación de los iconos que pertenecen a los diagramas de presentación.

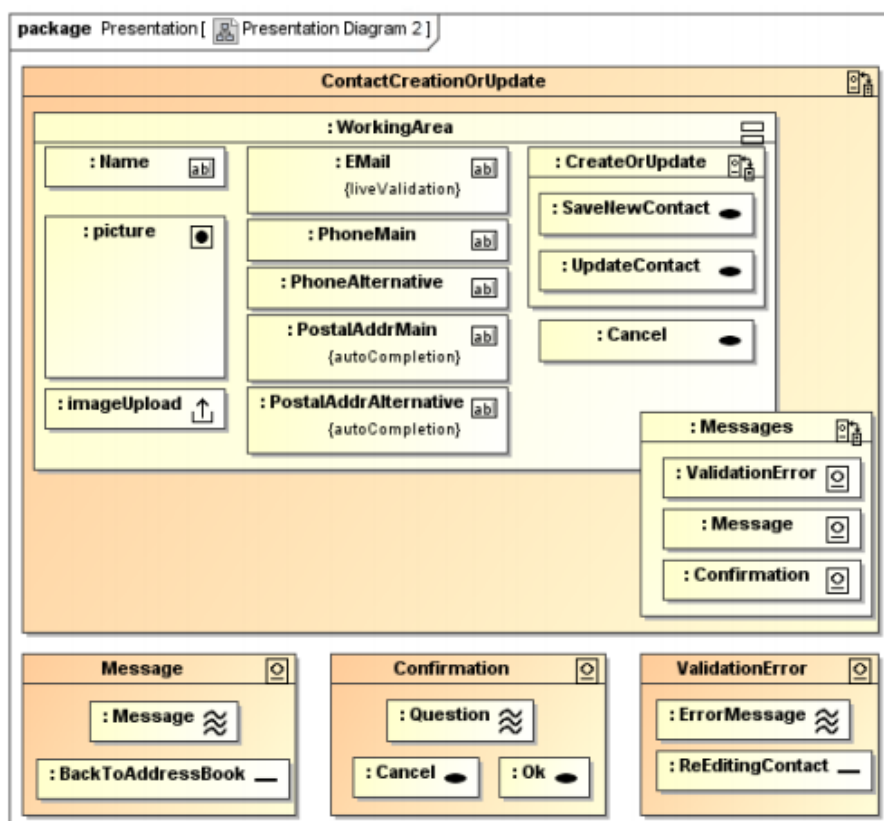


Figura 6: Diagrama de presentación

Fuente: Ingeniería web basada en UML, Instituto de Informática



## 10. ÍNDICE TENTATIVO

### **CAPÍTULO I MARCO INTRODUCTORIO**

- 1.1 INTRODUCCIÓN
- 1.2 ANTECEDENTES
  - 1.2.1 INSTITUCIONALES
    - 1.2.1.1 MISIÓN
    - 1.2.1.2 VISIÓN
  - 1.2.2 PROYECTOS SIMILARES.
- 1.3 PLANTEAMIENTO DEL PROBLEMA
  - 1.3.1 PROBLEMA CENTRAL
  - 1.3.2 PROBLEMAS SECUNDARIOS
- 1.4 DEFINICIÓN DE OBJETIVOS
  - 1.4.1 OBJETIVO GENERAL
  - 1.4.2 OBJETIVOS ESPECÍFICOS
- 1.5 JUSTIFICACIÓN
  - 1.5.1 ECONÓMICA
  - 1.5.2 SOCIAL
  - 1.5.3 TECNOLÓGICA
- 1.6 ALCANCES Y LÍMITES
  - 1.6.1 ALCANCES
  - 1.6.2 LÍMITES
- 1.7 APORTES
  - 1.7.1 APORTE PRÁCTICO
  - 1.7.2 APORTE TEÓRICO
- 1.8 METODOLOGÍA
  - 1.8.1 MÉTODO DE INVESTIGACIÓN
  - 1.8.2 METODOLOGÍA DE DESARROLLO

### **CAPÍTULO II MARCO TEÓRICO**

- 2.1 GESTION DOCUMENTAL
- 2.2 DOCUMENTO ELECTRONICO
- 2.3 REGISTROS ORTODONCICOS
- 2.4 INGENIERÍA DE SOFTWARE
  - 2.4.1 ETAPAS DEL PROCESO
  - 2.4.2 MODELOS DE PROCESOS DE SOFTWARE
    - 2.4.2.1 MODELOS TRADICIONALES
    - 2.4.2.2 MODELOS EVOLUTIVOS
    - 2.4.2.3 MODELOS PARA SISTEMAS ORIENTADO A OBJETOS
    - 2.4.2.4 PROCESOS AGILES
- 2.5 METODOLOGÍA DE DESARROLLO SCRUM
  - 2.5.1 HERRAMIENTAS DE LA METODOLOGÍA
    - 2.5.1.1 PILA DE PRODUCTO
    - 2.5.1.2 PRODUCT BACKLOG LIST
    - 2.5.1.3 SPRINTS
  - 2.5.2 ROLES Y RESPONSABILIDADES
    - 2.5.2.1 PRODUCT OWNER
    - 2.5.2.2 SCRUM MASTER

2.5.2.3	SCRUM TEAM
2.5.3	PROCESO DE LA METODOLOGÍA
2.5.3.1	PRE – GAME
2.5.3.2	GAME
2.5.3.3	POST – GAME
2.5.4	CONTROL DE EVOLUCIÓN DE PROYECTO
2.6	INGENIERÍA WEB
2.6.1	PROCESO DE INGENIERÍA WEB
2.6.2	HERRAMIENTAS TECNOLÓGICAS
2.7	METODOLOGÍA UWE
2.7.1	FASES DE LA METODOLOGÍA UWE
2.7.1.1	ANÁLISIS DE REQUISITOS
2.7.1.2	DISEÑO CONCEPTUAL
2.7.1.3	DISEÑO NAVEGACIONAL
2.7.1.4	DISEÑO DE PRESENTACIÓN
2.8	INFORMACIÓN INSTITUCIONAL
2.8.1	FORMATO DE REGISTRO DE CASOS
<b>CAPÍTULO III MARCO APLICATIVO</b>	
3.1	INTRODUCCIÓN
3.2	ANÁLISIS DE LA SITUACIÓN ACTUAL
3.3	IDENTIFICACIÓN DE ROLES SCRUM
3.4	FASE DE PRE-GAME
3.4.1	CREACIÓN DEL PRODUCT BACKLOG (PILA DE PRODUCTO)
3.4.2	IDENTIFICACIÓN DE ROLES DE USUARIO
3.5	FASE DEL GAME
3.5.1	DESARROLLO DEL SPRINT 1: MÓDULO DE INICIO DE SESIÓN
3.5.1.1	PLANIFICACIÓN DEL SPRINT 1
3.5.1.2	ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 1
3.5.1.3	DISEÑO NAVEGACIONAL
3.5.1.4	DISEÑO DE PRESENTACIÓN
3.5.1.5	PANTALLAS DEL SPRINT 1
3.5.1.6	PRUEBA UNITARIA DEL SPRINT 1
3.5.2	DESARROLLO DEL SPRINT 2: MÓDULO DE ADMINISTRACION DE USUARIOS
3.5.2.1	PLANIFICACIÓN DEL SPRINT 2
3.5.2.2	ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 2
3.5.2.3	DISEÑO NAVEGACIONAL
3.5.2.4	DISEÑO DE PRESENTACIÓN
3.5.2.5	PANTALLAS DEL SPRINT 2
3.5.2.6	PRUEBA UNITARIA DEL SPRINT 2
3.5.3	DESARROLLO DEL SPRINT 3: MÓDULO DE REGISTRO DE CASOS
3.5.3.1	PLANIFICACIÓN DEL SPRINT 3
3.5.3.2	ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 3
3.5.3.3	DISEÑO NAVEGACIONAL
3.5.3.4	DISEÑO DE PRESENTACIÓN
3.5.3.5	PANTALLAS DEL SPRINT 3



3.5.3.6	PRUEBA UNITARIA DEL SPRINT 3
3.5.4	DESARROLLO DEL SPRINT 4: MÓDULO DE ARCHIVO Y COMENTARIOS
3.5.4.1	PLANIFICACIÓN DEL SPRINT 4
3.5.4.2	ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 4
3.5.4.3	DISEÑO NAVEGACIONAL
3.5.4.4	DISEÑO DE PRESENTACIÓN
3.5.4.5	PANTALLAS DEL SPRINT 4
3.5.4.6	PRUEBA UNITARIA DEL SPRINT 4
3.5.5	DISEÑO CONCEPTUAL
3.6	FASE DEL POST – GAME
3.6.1	PRUEBAS DE INTEGRACIÓN
3.6.2	PRUEBAS DE STRESS
<b>CAPÍTULO IV CALIDAD Y SEGURIDAD</b>	
4.1	CALIDAD DE SOFTWARE
4.2	SEGURIDAD
<b>CAPÍTULO V ANÁLISIS COSTO BENEFICIO</b>	
5.1	COCOMO II
5.2	COSTO DEL SISTEMA
5.2.1	COSTO DE DESARROLLO DEL SOFTWARE
5.2.2	COSTO DE IMPLEMENTACIÓN
5.2.3	COSTO DE ELABORACIÓN DEL PROYECTO
5.2.4	COSTO TOTAL DEL SOFTWARE
5.3	VALOR ACTUAL NETO
5.3.1	COSTO / BENEFICIO
5.4	TASA INTERNA DE RETORNO
<b>CAPÍTULO VI CONCLUSIONES Y RECOMENDACIONES</b>	
6.1	CONCLUSIONES
6.2	RECOMENDACIONES
<b>BIBLIOGRAFÍA</b>	

*Tabla 1: Índice Tentativo*

*Fuente: Elaboración Propia*

## 11. CRONOGRAMA DE AVANCE

CRONOGRAMA DE AVANCE PROYECTO DE GRADO																									
ACTIVIDADES	DURACIÓN EN DÍAS	DEL 1 DE JULIO AL 30 DE NOVIEMBRE																							
		JULIO				AGOSTO				SEPTIEMBRE				OCTUBRE				NOVIEMBRE				DIC.			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2		
Redacción del Capitulo II - Marco Teórico																									
Desarrollo del Capitulo III - Marco Aplicativo																									
Metodología Scrum PRE-GAME																									
GAME																									
POST-GAME																									
CIERRE																									
Redacción del Capitulo III - Marco Aplicativo																									
Redacción del Capitulo IV - Calidad y																									
Redacción del Capitulo V - Análisis																									
Redacción del Capitulo VI - Conclusiones y Recomendaciones																									

*Tabla 2: Elementos de diseño Navegacional*

*Fuente: Ingeniería web basada en UML, Instituto de Informática*

## 12. BIBLIOGRAFÍA

Arias F., (2012) El proyecto de Investigación – Introducción a la metodología científica. 6ta edición, editorial Episteme, Venezuela.

Duarte B., (2017) Sistema Web de Gestión de Pacientes Odontológicos ODONTOWEB. Universidad Nacional de Ingeniería, Facultad de Electrónica y computación, Nicaragua. [Consulta: 10 de julio, 2020], Recuperado de: <<http://ribuni.uni.edu.ni/1494/1/80742.pdf>>.

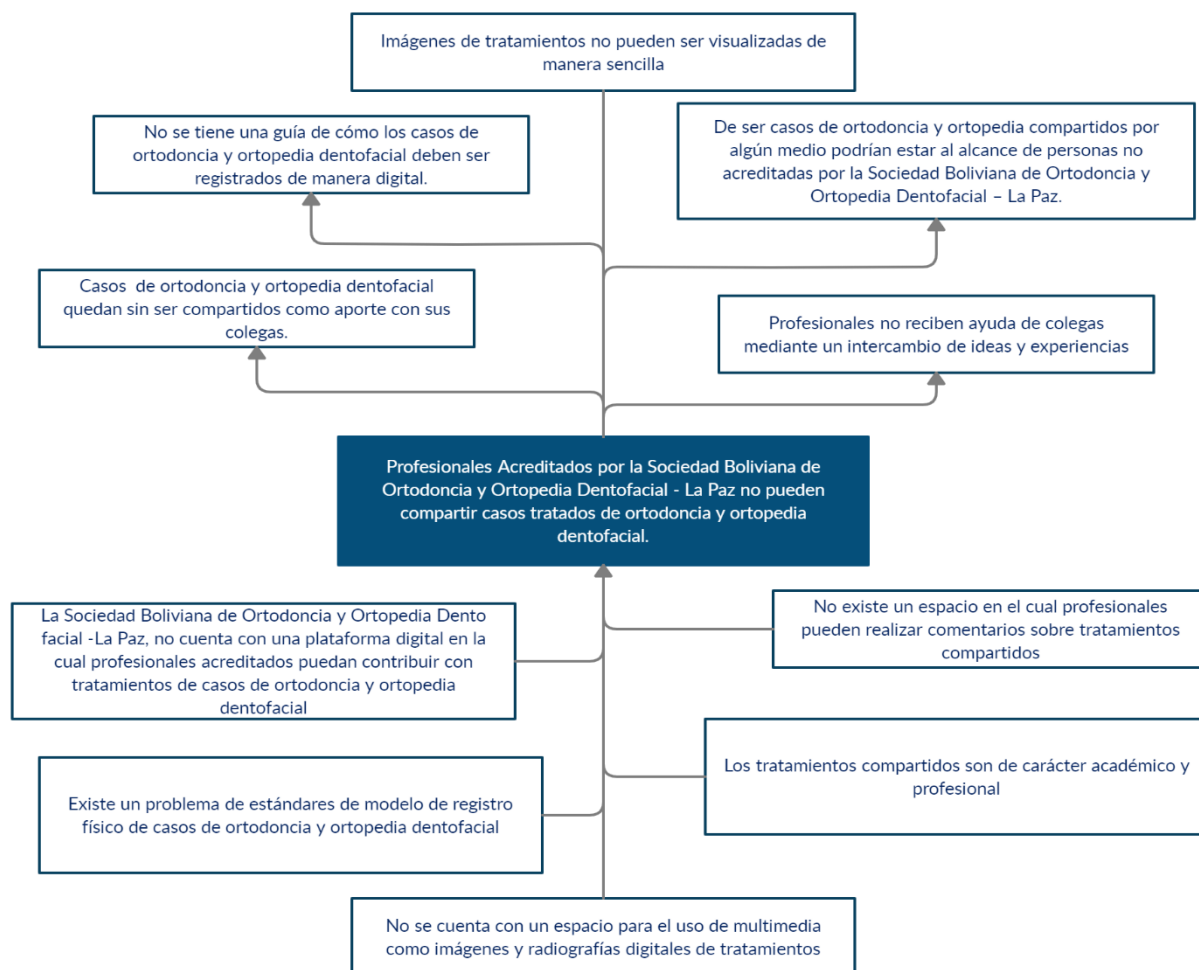
García A. (2001), La gestión de documentos electrónicos como respuesta a las nuevas condiciones del entorno de información. [Consulta: 29 de julio, 2020], Recuperado de: <[http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S1024-94352001000300003#:~:text=Se%20trata%20de%20eficientes%20sistemas,im%C3%A1genes%20electr%C3%B3nicas%20de%20documentos%20impresos.](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352001000300003#:~:text=Se%20trata%20de%20eficientes%20sistemas,im%C3%A1genes%20electr%C3%B3nicas%20de%20documentos%20impresos.)>.

- Gonzales M, (2013), Sistema web de seguimiento a historias clínicas para la empresa Spa Medico CIME basado en CRM, Licenciatura en informática, La Paz, Universidad Mayor de San Andrés, Facultad de Ciencias Puras y Naturales, Carrera de Informática.
- Huanca C., (2015), Sistema web de control de pagos, citas e historiales clínicos caso: Clínica Dental LAVADENT, Licenciatura en informática, La Paz, Universidad Mayor de San Andrés, Facultad de Ciencias Puras y Naturales, Carrera de Informática.
- Kenneth E. Kendall y Julie E. Kendall, 2005, Análisis y Diseño de Sistemas, 5ta Edición, México, Editorial Pearson Educación.
- Loynes C. (2019), ¿Qué son los registros dentales de ortodoncia?, Revista Digital *Disciplined*, [Consulta: 29 de julio, 2020], Recuperado de: <<https://disciplined.com/que-son-los-registros-dentales-de-ortodoncia/>>.
- Lozano R., (2014), Sistema de administración y control de historiales clínicos para consultorios de la UMSA, Licenciatura en informática, La Paz, Universidad Mayor de San Andrés, Facultad de Ciencias Puras y Naturales, Carrera de Informática.
- Mallea J. (2009), Ingeniería Web Universidad Autónoma Juan Misael Saracho, Facultad de Ciencias y Tecnología, Carrera de Ingeniería Informática. Tarija, Bolivia. [Consulta: 29 de julio, 2020], Recuperado de: <<http://juanedgarmallea.blogspot.com/2009/06/ingenieria-web.html>>.
- Murugesan S., Deshpande Y., Hansen S., Ginige. A., (2016) *Web Engineering: A New Discipline for Development of Web-Based Systems."* *Lecture Notes in Computer Science*.
- Ortegón E, Pacheco J. y Prieto A, 2005, Metodología del Marco Lógico para la planificación, el seguimiento y la evaluación de proyectos y programas, 2da Edición, Naciones Unidas.
- Paredes V, Gandía J. y Cibrián R. (2006). Registros diagnósticos digitales en ortodoncia. Situación actual. [Consulta: 10 de julio de 2020], Recuperado de <[http://scielo.isciii.es/scielo.php?script=sci\\_arttext&pid=S1698-69462006000100020/](http://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S1698-69462006000100020/)>.
- Pressman R., (2010), Ingeniería del Software un enfoque práctico. 7ma Edición, [Consulta: 10 de julio de 2020], Recuperado de: <<http://cotana.informatica.edu.bo/downloads/Id-Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF>>

- Quiroga W, (2013), Sistema web de seguimiento y control de pacientes internos, Licenciatura en informática, La Paz, Universidad Mayor de San Andrés, Facultad de Ciencias Puras y Naturales, Carrera de Informática.
- Sanz D., García E. (2011), Metodologías para el Desarrollo de Aplicaciones Web: UWE, [Consulta: 18 de julio de 2020], Recuperado de: <<https://jorgeportella.files.wordpress.com/2011/11/analisis-diseo-y-desarrollodeaplicacionesweb.pdf>>.
- Sommerville I. (2005), Ingeniería del Software. Séptima edición. [Consulta: 29 de julio de 2020], Recuperado de: <<https://es.scribd.com/document/357706285/Ingenieria-de-Software-lan-Sommerville-pdf>>.
- Vigo, S (2013), Registros Ortodóncicos y Equipo de Diagnóstico. [Consulta: 20 de junio de 2020], Recuperado de: <<http://www.facalortodoncia.com/la-clinica/equipo-diagnostico/>>.

## ANEXOS

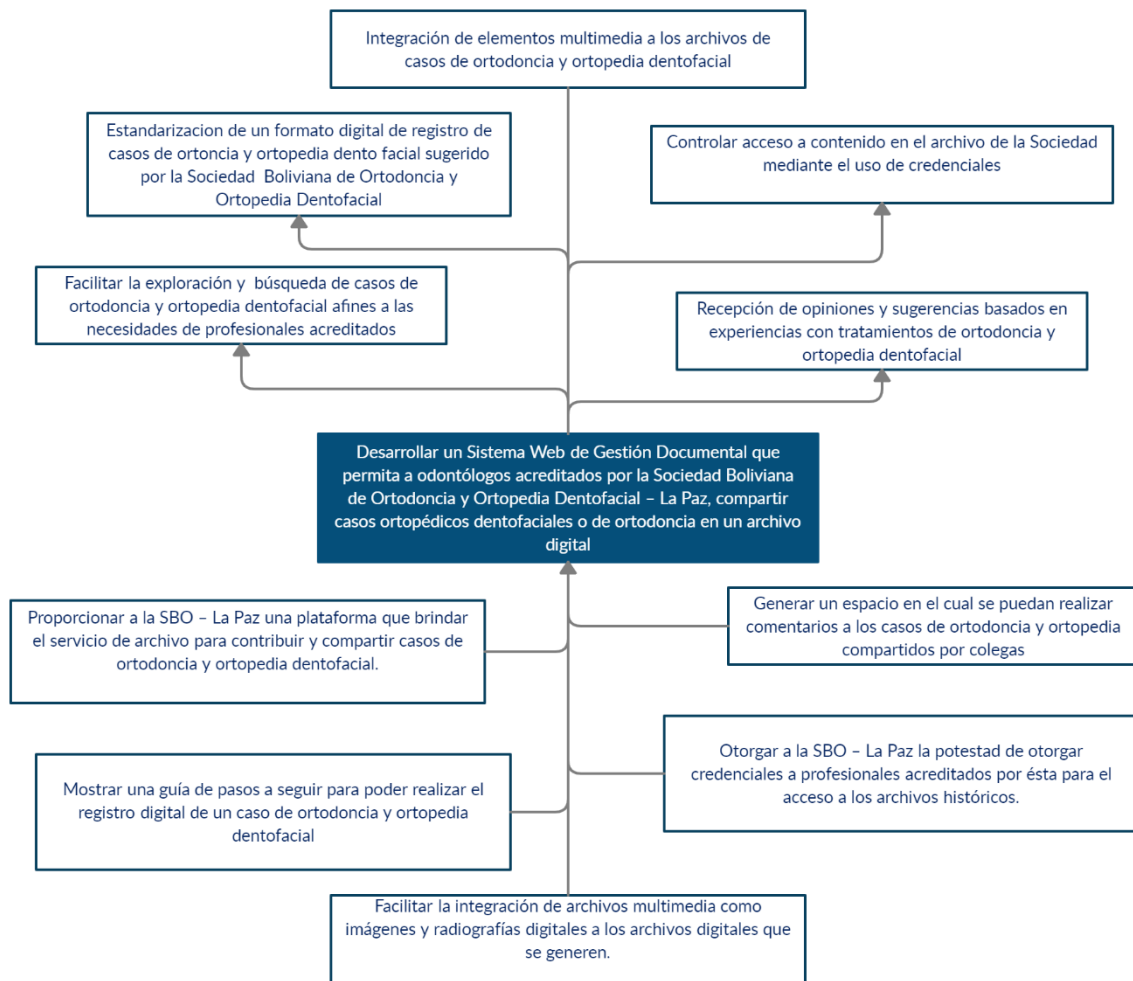
### ANEXO A – ÁRBOL DE PROBLEMAS



*Figura 7: Árbol de problemas*

*Fuente: Elaboración Propia*

## ANEXO B – ÁRBOL DE OBJETIVOS



*Figura 8: Árbol de objetivos*

*Fuente: Elaboración Propia*

## ANEXO C – MARCO LÓGICO

RESUMEN NARRATIVO	INDICADORES	MÉTODOS DE VERIFICACIÓN	SUPUESTOS
<b>FIN</b> Lograr que profesionales compartan casos de ortodoncia tratados mediante la Sociedad Boliviana de Ortodoncia y Ortopedia Dentofacial - La Paz	Índice de conformidad de profesionales acreditados por la Sociedad Boliviana de Ortodoncia y Ortopedia Dentofacial – La Paz	Informes recopilados de encuestas a profesionales acreditados por la sociedad Boliviana de Ortodoncia y Ortopedia Dentofacial – La Paz	Los profesionales cuentan con acceso a internet  Los casos cuentan con características necesarias para ser compartidos
<b>PROPÓSITO</b> Generar un repositorio de Kardex de tratamientos mediante la implementación un sistema web de gestión documental de casos de ortodoncia y ortopedia dentofacial para la Sociedad Boliviana de Ortodoncia y ortopedia dentofacial - La Paz	Medios d		
<b>PRODUCTO</b> -Control de credenciales de acceso -Búsqueda y visualización de casos en el archivo -Integración de elementos multimedia a casos de ortodoncia registrados.		Reportes semanales de avance. Documentación generada de acuerdo a la metodología SCRUM. Visto bueno del área administrativa de la Sociedad Boliviana de Ortodoncia y Ortopedia dentofacial -La Paz.	

<b>ACTIVIDADES</b>	
- Fase De Elaboración o ( <i>Pre-Game</i> )	Recopilación de información.
-Fase De Construcción o ( <i>Game</i> )	Análisis, Diseño y modelado del sistema web. Desarrollo de la solución.
-Fase De Transición ( <i>Post-Game</i> )	Implementación del sistema, pruebas y capacitación.

*Tabla 3: Matriz de Marco Lógico*

*Fuente: Elaboración Propia*