



## PROYECTO DE GRADO

Presentado ante la ilustre UNIVERSIDAD DE LOS ANDES como requisito final para  
obtener el Título de INGENIERO DE SISTEMAS

Desarrollo del Nuevo Sistema de Información Web para el Manejo  
de Historias Clínicas Odontológicas de la Facultad de Odontología  
de la Universidad de Los Andes

Por

Br. Joanelly González

Tutor: Domingo Hernández

Cotutora: Norelkys Espinoza.

Octubre 2017

©20017 Universidad de Los Andes Mérida, Venezuela

C.C. Reconocimiento

# Desarrollo del Nuevo Sistema de Información Web para el Manejo de Historias Clínicas Odontológicas de la Facultad de Odontología de la Universidad de Los Andes

Br. Joanelly González

Proyecto de Grado — Sistemas Computacionales, 177 páginas

**Resumen:** El presente trabajo, tiene como propósito fundamental desarrollar el nuevo sistema de Información Web para el manejo de Historias Clínicas Odontológicas, en la Facultad de Odontología específicamente en la Clínica Integral del Adulto, la cual presta un servicio médico-odontológico a la comunidad merideña con el fin del resguardar la información y evitar pérdidas de tiempo en atención, búsqueda y consultas.

Para la realización de esta investigación se utilizó la metodología de análisis y diseño de sistemas de información descrita por Jonás Montilva, aplicando las fases descritas en el Método White Watch, que van desde el modelado de negocios, ingeniería de requisitos, diseño del sistema, hasta la implementación y pruebas del mismo. Se partió de un estudio de campo, para conocer la realidad y las necesidades de las nuevas funcionalidades a incorporar en el registro y almacenamiento de la Historia Clínica. Posterior a esto, se determinaron las bases teóricas que sustentan el sistema propuesto. En la fase de ingeniería de requisitos se realizaron reuniones semanales con el personal interno de la clínica integral del adulto para la recolección de las nuevas incorporaciones a la historia clínica del paciente. Para el modelado se utilizó UML 2.0 (Unified Modeling Language), como herramienta de desarrollo se utilizó el framework Laravel, y como manejador de base de datos PostgreSQL. Los objetivos propuestos al inicio de la investigación fueron cumplidos y se obtuvo como conclusión que el sistema de información web facilita y agiliza de una manera eficaz la gestión de control y manejo de historias clínicas odontológicas.

**Palabras clave:** Historias Clínicas, Sistema de Información, Método White Watch, UML 2.0

# Índice

Dedicatoria .....	iv
Índice de Figuras.....	viii
Índice de Tablas.....	x
Agradecimientos.....	xi
Capítulo 1 Introducción.....	1
1.1 Antecedentes .....	2
1.2 Contexto de la Organización .....	4
1.3 Planteamiento del Problema .....	5
1.4 Justificación .....	6
1.5 Objetivos .....	6
1.6 Objetivo General.....	6
1.7 Objetivos Específicos.....	7
1.8 Metodología.....	7
1.9 Alcance .....	9
Capítulo 2 Marco Teórico .....	10
2.1 Conceptos Básicos Empleados en Odontología.....	10
2.1.1 Historia Clínica.....	10
2.1.2 Diagnostico.....	11
2.1.3 Diagnóstico Clínico.....	11
2.1.4 Pronostico.....	11
2.1.5 Tratamiento.....	11
2.1.6 Radiografía .....	11
2.1.7 Odontograma.....	12
2.2 Conceptos empleados en el desarrollo de sistemas y descripción de algunas herramientas requeridas para el diseño. ....	12
2.2.1 Sistema.....	12
2.2.1.1 Elementos de un Sistema (Senn James, 1993) .....	12
2.2.1.2 Clasificación de los Sistemas (Senn James, 1993) .....	13

2.2.1.3 Sistemas de Información .....	13
2.2.1.4 Diseño de Sistemas de Información.....	14
2.2.1.5 Beneficios de los sistemas de Información (Kendall K. 1997) .....	14
2.2.2 Software Propietario .....	15
2.2.3 Software Libre.....	15
2.2.4 HTML.....	15
2.2.5 Framework.....	15
2.2.6 Modelo de Proceso “Reloj” (White Watch).....	16
2.2.7 UML.....	18
2.2.8 Modelo Vista Controlador .....	21
2.2.8.1 Modelo.....	21
2.2.8.2 Vistas .....	22
2.2.8.3 Controlador.....	22
2.2.9 Laravel .....	23
2.2.10 PostgreSQL.....	24
2.2.10.1 Base de Datos .....	24
2.2.10.2 Base de Datos relacionales .....	25
Capítulo 3 Modelado y Análisis de Requerimientos.....	26
3.1 Análisis del Dominio de la aplicación.....	26
3.1.1 Identificación del proceso .....	27
3.1.2. Modelado del proceso de negocio .....	29
3.2 Descubrimiento de requisitos funcionales: .....	29
3.2.1 Identificación de Actores. ....	30
3.2.2 Casos de Uso.....	32
3.2.2.1 Definición de los Casos de Uso.....	32
3.2.2.2 Descripción de los Casos de Uso.....	33
3.2.2.3 Diagramas de Casos de Uso .....	40
3.2.2.3.1 Casos de Uso del actor: Administrador.....	40
3.2.2.3.2 Casos de Uso del actor: jefe de Clínica. ....	41
3.2.2.3.3 Casos de Uso del actor: Profesor Tutor. ....	42
3.2.2.3.4 Casos de Uso del actor: Estudiante. ....	43

3.3 Descubrimiento de requisitos No funcionales: .....	43
Capítulo 4 Diseño del Sistema .....	45
4.1 Definición de la estructura inicial de la aplicación. ....	45
4.1.1 Definición de las metas de diseño: .....	45
4.1.1.1 Requisitos relacionados con la capa de presentación: .....	46
4.1.1.2 Requisitos relacionados con la capa de lógica de negocios: .....	46
4.1.1.3 Requisitos relacionados con la capa de datos: .....	47
4.1.2 Identificación de los subsistemas: .....	47
4.1.3 Descripción de la vista arquitectónica. ....	47
4.1.3.1 Vistas de Uso .....	48
4.1.3.2 Vista de comportamiento.....	48
4.1.3.3 Vista de datos .....	49
4.1.3.3.1 Modelo de la Base de Datos Relacional del sistema SIHCO.....	51
4.1.3.3.2 Definición física de la Base de Datos.....	52
4.1.3.4 Vista de componentes .....	54
4.1.3.5 Vista de despliegue .....	55
4.2 Diseño de la interfaz de usuario. ....	55
4.2.1 Grafo de navegación. ....	55
4.2.1.1 Estilo Visual de las Páginas. ....	57
Capítulo 5 Implementación y Prueba del Sistema .....	60
5.1 Construcción del Sistema.....	60
5.2 Gestión de Configuración. ....	72
5.2 Pruebas del sistema. ....	73
5.2.1 Prueba de caja negra.....	73
5.2.2 Prueba de caja blanca. ....	75
5.2.3 Prueba de Navegación.....	82
Capítulo 6 Conclusiones y Recomendaciones .....	84
6.1 Conclusiones.....	84
6.2 Recomendaciones.....	85
Referencias Bibliográficas .....	86
Anexos .....	88

Anexo A. ....	89
Casos de Usos del Sistemas Web .....	89
Anexo B. ....	93
Diagramas de Secuencia para los casos de Uso del Sistema Web .....	93
Anexo C. ....	97
Diccionario de Datos .....	97
Anexo D. ....	103
Vistas del Sistema Web .....	103

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

# Índice de Figuras

Figura 2.1. Modelo del Proceso White Watch por Jonas Montilva, 2010. ....	17
Figura 2.2. Modelo MVC .....	23
Figura. 3.1. Modelo del proceso para la atención de un paciente .....	28
Figura 3.2. Cadena de Valor de SIHCO.....	29
Figura.3.3. Jerarquía de Actores .....	31
Figura 3.4. Caso de uso para el rol Administrador .....	41
Figura 3.5. Caso de uso rol: Profesor Jefe de Clínica.....	41
Figura 3.6. Caso de uso rol: Profesor Tutor. ....	42
Figura 3.7. Caso de uso rol: Estudiante. ....	43
Figura 4.1. Diagrama de secuencia: Anexar Historia para el rol Estudiante. ....	48
Figura 4.2. Diagrama de secuencia: Validar Historia para el rol Profesor Tutor. ....	49
Figura 4.3. Diagrama de Clases .....	50
Figura 4.4. Diagrama de Componentes del sistema web.....	54
Figura 4.5. Diagrama de Despliegue del sistema web .....	55
Figura 4.3. Grafo de Navegación. ....	56
Figura 4.4 Menú de acceso para el estudiante de 2do año .....	58
Figura 4.5 Menú de acceso para el estudiante de 3er año .....	59
Figura 4.5 Menú de acceso para el estudiante de 3er año .....	59
Figura 5.1. Vista de acceso al sistema.....	61
Figura 5.2. Vista del menú para el rol jefe de clínica .....	62
Figura 5.3 Vista de Selección de Paciente para ingresar historia .....	63
Figura 5.4 Vista de Ingreso de Consulta.....	63
Figura 5.5 Vista de Datos Personales .....	64
Figura 5.6 Vista de Antecedente Familiar.....	64
Figura 5.7 Vista de Ingreso de Odontograma .....	65
Figura 5.8 Vista de Ingreso de Control de Placa .....	65
Figura 5.9 Vista de Ingreso de Imagenología .....	66
Figura 5.10 Vista de Ingreso de Imágenes radiográficas .....	67

Figura 5.11 Vista de Ingreso de Imágenes radiográficas (Panorámicas).....	67
Figura 5.12 Vista de Ingreso del Examen Muscular .....	68
Figura 5.13 Vista de Ingreso del Examen Muscular .....	68
Figura 5.14 Vista de Ingreso del Examen de la Oclusión.....	69
Figura 5.15 Vista de Ingreso del Examen de la Oclusión.....	70
Figura 5.16 Vista de Ingreso prótesis parcial removible superior.....	70
Figura 5.17 Vista de Ingreso prótesis parcial removible superior.....	71
Figura 5.18 Vista de Ingreso de una cita.....	71
Figura 5.19 Vista de Consultas de las citas.....	72
Figura 5.20. Guardar Consulta sin campos obligatorios.....	75
Figura 5.21 Insertar Historia: Antecedente Familiar por el rol estudiante de 2do año .....	76
Figura 5.22 Insertar Historia: Antecedente Familiar por el rol estudiante de 2do año .....	76
Figura 5.23 Consultar Historia: Antecedente Familiar por el rol estudiante de 2do año.....	77
Figura 5.24 Consultar Historia: Antecedente Familiar por el rol estudiante de 2do año.....	77
Figura 5.25 Agregar Usuario por parte del rol Jefe de Clínica.....	79
Figura 5.26 Agregar Usuario por parte del rol Jefe de Clínica.....	80
Figura 5.27 Consultar Usuario por parte del rol Jefe de Clínica.....	80
Figura 5.28 Consultar Usuario por parte del rol Jefe de Clínica.....	81
Figura A.1 Caso de uso ingresar al Sistema .....	89
Figura A.2. Caso de uso anexar historia para el caso del rol estudiante .....	90
Figura A.3. Caso de uso generalizado para CRUD de pacientes y CRU de Historias .....	91
Figura A.4. Detalles de los casos de uso CRUD usuario y CRUD paciente .....	92
Figura B.1. Diagrama de secuencia para el CU2, ingresar paciente .....	93
Figura B.2. Diagrama de secuencia para el caso de uso CU5, ingresar usuario.....	94
Figura B.3. Diagrama de secuencia para la consulta de usuario .....	95
Figura B.4. Diagrama de Secuencia para el caso de uso, CU12, consultar estadísticas .....	96
Figura D.1 Vista de Consulta de Usuarios .....	103
Figura D.2 Vista de Consulta de Pacientes.....	104
Figura D.3 Vista de Ingreso de usuario .....	104
Figura D.4 Vista de Ingreso de Pacientes .....	105

# Índice de Tablas

Tabla 3.1. Casos de uso del sistema web.....	32
Tabla 3.2. Caso de uso Ingresar al sistema.....	33
Tabla 3.3. Descripción Caso de Uso Agregar Usuario.....	34
Tabla 3.4. Descripción caso de uso Modificar usuario.....	34
Tabla 3.5. Descripción caso de uso: Eliminar Usuario .....	35
Tabla 3.6. Descripción caso de uso: Consultar Historia.....	35
Tabla 3.7. Descripción caso de uso: Consultar Historia.....	36
Tabla 3.8. Descripción caso de uso: Agregar Paciente .....	36
Tabla 3.9. Descripción caso de uso: Modificar Paciente .....	37
Tabla 3.10. Descripción caso de uso: Eliminar Paciente .....	38
Tabla 3.11. Descripción caso de uso: Anexar Historia .....	38
Tabla 3.12. Descripción caso de uso: Validar Historia .....	39
Tabla 3.13. Descripción caso de uso: Modificar Historia .....	40
Tabla 4.1. Base de datos relacional.....	51
Tabla 4.2. Definición física de la tabla Persona .....	52
Tabla 4.3. Definición física de la tabla Usuario .....	52
Tabla 4.4. Definición física de la tabla Paciente.....	54
Tabla 5.1: Pruebas Caja Negra.....	74
Tabla 5.2: Pruebas de Navegación .....	83
Tabla C.1. Descripción de la tabla historia .....	98
Tabla C.2. Descripción de la tabla Consulta .....	98
Tabla C.3. Descripción de la tabla datos clínicos .....	100
Tabla C.4. Descripción de la tabla exámenes complementarios .....	101
Tabla C.5. Descripción de la tabla examen clínico .....	102

# Capítulo 1

## Introducción

Durante el último siglo se ha producido un aumento espectacular en el ámbito de la informática y la comunicación verdaderamente sorprendente, ante los beneficios prácticos que le ofrece al hombre para el desarrollo de sus actividades. Es así como la tecnología brinda una serie de ventajas competitivas que permiten el desarrollo de la globalización, modernización, e integración donde todos los elementos a nivel internacional, nacional, regional y local, pueden coordinarse para brindar procesos integrados y efectivos en los sectores económicos, sociales, culturales y políticos. En ese aspecto de integración, los sistemas de información y las distintas aplicaciones de software surgieron como herramientas para condicionar datos importantes, aumentando los niveles de eficiencia y efectividad a través del Internet desde el punto de vista de rentabilidad y competitividad contribuyendo con el desarrollo personal y organizacional.

Es así como Venezuela, en su deseo de desarrollo y crecimiento ha logrado incorporar en pequeñas, medianas y grandes empresas, sistemas de información y software para optimizar las operaciones administrativas como operativas, siguiendo el esquema de acceder datos para que el computador realice las actividades de clasificación, de tal manera que sea un trabajo más fácil y organizado.

En tal sentido, es de hacer notar que los médicos, odontólogos, bioanalistas, entre otros profesionales de la salud, trabajan directamente con pacientes, que deben ser tratados con el respeto que le corresponde de manera individual, por ello en la consulta se lleva una historia médica de cada caso en particular, de manera que al ser atendidos se lleve un control de su salud pasada y presente para poder asumir decisiones al respecto de la situación de cada uno de ellos.

Tal es el caso de los Odontólogos, quienes atienden en sus consultorios a una cantidad de pacientes, cada uno con problemas más o menos similares pero particulares en cuanto a las condiciones que cada uno presenta, y cuya prioridades en la atención se definen a través del triaje para definir el tratamiento terapéutico más recomendable con el propósito de mejorar su estado de salud bucal , de allí

que se lleve una historia clínica para registrar todas las eventualidades y servir de guía al profesional en la toma de decisiones.

En la Facultad de Odontología de la Universidad de Los Andes, se lleva una historia médica por paciente. En tal sentido, en el Archivo de registros clínicos se encuentran cada una de las historias archivadas en carpetas y deben ser consultadas en cada visita del paciente, ocasionando algunas veces pérdida de tiempo en su búsqueda y llenado, como también en el seguimiento del proceso de tratamiento, lo cual afecta negativamente si se toma en cuenta el tiempo que se requiere para cada caso. Por lo cual se considera necesario asumir el cambio incorporando sistemas de información que no solo pondrían en la palestra a la institución en cuanto a los procesos administrativos, organizacionales y tecnológicos, sino que además su actividad de extensión sería reconocida como un servicio de calidad, puesto que la presente innovación facilitaría las operaciones dentro de la misma, redundando en una mejor atención al paciente odontológico.

## 1.1 Antecedentes

www.bdigital.ula.ve

En relación con la automatización de procesos se han realizado diversos estudios, a continuación, se describen algunos de ellos que guardan cierta relación con el presente trabajo:

Duque (2009) realizó un estudio titulado: “Software Para la Gestión de Control de Historias Clínicas Odontológicas”. Para este estudio el objetivo fue diseñar un software para la gestión de control de las historias clínicas odontológicas.

El estudio se desarrolló tomando en cuenta el proceso de una investigación positivista, proyecto factible a partir de hechos objetivos y proponer alternativas de solución que son posibles de ejecutar. La investigación se considera no experimental, ya que las variables del sistema de información, así como sus dimensiones e indicadores, fueron analizados en su estado natural, sin la intervención del investigador.

La metodología utilizada fue la de Montilva (2010), las herramientas de diseño del sistema fue el lenguaje de programación Python, El manejador de base de datos SQLite.

Los resultados de esta investigación lograron abarcar los objetivos propuestos en dicho estudio y se obtuvo un sistema de información automatizado que facilita y agiliza de una manera eficaz la gestión de control de historias odontológicas.

Este proyecto proporcionó un aporte considerable, como lo fue el tratamiento de la metodología en el análisis y aplicación de cada fase de la misma, para lograr una mejor claridad y aplicación para el diseño de esta investigación.

Villareal (2003), en su trabajo especial de grado titulado: “Sistema de Información Web para el manejo de Historias Clínicas Odontológicas”, el objetivo de esta investigación fue desarrollar un sistema bajo el ambiente Cliente-Servidor, de fácil y rápido manejo, capaz de agilizar el proceso de registro, búsqueda de información y control de citas de los pacientes que acuden a la Clínica Integral de Adulto de la Facultad de Odontología de la Universidad de Los Andes.

La metodología utilizada fue la del autor Montilva (2010), metodología estructurada o ciclo de vida de un sistema de información.

Los resultados de este trabajo determinaron que el sistema web cumple con todos los requerimientos y objetivos iniciales propuestos, y se adquirió un sistema de información el cual mejora la velocidad de respuesta al momento de la búsqueda y el almacenamiento de las historias de los pacientes junto con las respectivas imágenes radiográficas.

Se utilizó como técnica de recolección de datos la entrevista no estructurada realizada a las personas encargadas en la clínica Integral de Adulto (Profesores y Estudiantes), donde se describió la situación actual del mismo referente a la forma de llenado de las historias clínicas de cada paciente, así como el proceso de triaje que realiza cada paciente al momento de asistir a dicha Institución.

El estudio y desarrollo de este trabajo proporciona aportes considerables para la sustentación de este proyecto, tales como: La utilización de un software prototipo como modelo de referencia para el diseño y estructura, así como también el análisis del diseño del sistema, tratamiento de cada fase de la metodología, presentación y evaluación del mismo para lograr obtener claridad de los objetivos y resultados que se deseen alcanzar en este proyecto.

## 1.2 Contexto de la Organización

Los Orígenes de la Facultad de Odontología de la Universidad de Los Andes remontan al año 1942 cuando la escuela de Dentistería (que pertenecía a la Escuela de Medicina) se transforma en la Facultad de Odontología. Al efectuarse este cambio, los estudios de Odontología se elevan a 4 años, en vez de 3 y se otorga al final de la carrera el Título de Doctor de Odontología. En 1967 el plan de estudios se establece en 5 años y se crea el Departamento de Odontología Preventiva y Social, incorporando al pensum las materias humanísticas indispensables para una buena formación profesional.

En el año 1998 comienza a regir un nuevo plan de estudios, con un régimen de estudio por anualidad, el cual ha sido modificado en el año 2007 y recientemente en el año 2016, y otorga el título de Odontólogo. La duración de la carrera según el régimen de estudio es de 5 años con períodos lectivos de 32 semanas. Está dividido en 6 líneas curriculares: 1. Biopatológica, 2. Odontología Social, 3. Practica Odontológica, 4. Desarrollo Humano, 5. Clínica, 6. Investigación, las cuales son regentada administrativamente por los departamentos que lo conforman: Departamento de Biopatología, Departamento de Odontología Preventiva y Social, Departamento de Medicina Oral, Departamento de Odontología Restauradora y Departamento de Investigación.

La Facultad de Odontología de La Universidad de Los Andes se encuentra ubicada en el centro del Estado Mérida, específicamente en el Rectorado y La Casona, ubicados en la calle 24 y calle 23 entre las avenidas 2 y 3. Es en La Casona donde se encuentran el Laboratorio de Computación y el Decanato.

Dentro del Pensum de Estudios, las asignaturas clínicas que lo conforman engloban una serie de actividades de atención odontológica, que se desarrollan en atención a cubrir las necesidades que, en esta materia, presenta la comunidad merideña, como también ser parte fundamental de la formación de los estudiantes de dicha Institución, la cual debe complementarse con el registro de los datos clínicos de los pacientes allí tratados en forma de Historias clínicas bajo la supervisión de Profesores de esta Ilustre Institución.

### 1.3 Planteamiento del Problema

El registro de los datos clínicos del paciente consiste en realizar una serie de preguntas estructuradas, cuyas respuestas son los datos que serán reflejadas en su Historia Clínica. Esta práctica y el llenado de la Historia se llevan a cabo bajo la supervisión de una o varios profesores de la Facultad de Odontología.

El proceso de registro, almacenamiento y búsqueda de las historias clínicas se lleva a cabo de forma manual, lo que trae como consecuencia desorganización de las historias, descuido, olvidos y por ende trabajos poco satisfactorios. Estos factores suelen ocasionar retrasos en la consulta y dificultad en la realización de un seguimiento detallado del tratamiento integral de cada paciente.

En la búsqueda de una solución eficiente para dicha problemática Villareal (2003) propuso el desarrollo de un sistema de información web bajo ambiente Cliente-Servidor que permitiría la manipulación de las historias clínicas y un mayor control sobre los pacientes, este sistema fue desarrollado bajo software privativo, con el lenguaje ASP, y SyBase como manejador de base de datos, además se utilizaron complementos (“plugins”) para la visualización e interacción de las distintas secciones gráficas de la historia clínica como lo son el Odontograma, el examen clínico, control de placa, entre otras, estas secciones se limitaron a su funcionamiento solo en ciertas versiones del navegador Internet Explorer desde la 5.5 hasta la versión 7 ya que versiones más nuevas estas funcionalidades se perdieron, por lo que el sistema no se implementó en su totalidad y todo el proceso de registro y consultas de los pacientes continúa llevándose de forma manual.

Debido a las dificultades presentadas por dicho sistema en cuanto a sus funcionalidades, y por obligación del Decreto 3.390 publicado en gaceta oficial Nro 38.095 el 28 de Diciembre de 2004 el cual determina que toda entidad pública debe establecer una política de migración de sus sistemas a software libre, se hace necesario la creación de un nuevo sistema de información Web en el que se realizará el manejo de historias Clínicas Odontológicas, tomando en cuenta el trabajo previo realizado por parte de Villareal (2003).

## **1.4 Justificación**

En la Facultad de Odontología de la Universidad de Los Andes, se lleva a cabo actualmente el proceso de registro, almacenamiento y búsqueda de las historias clínicas de forma manual, debido a las limitaciones presentadas por el sistema desarrollado por Villareal (2003), el cual permaneció apegado a versiones anteriores del navegador Internet Explorer quedando algunas funcionalidades obsoletas por lo que no se implementó dicho sistema, además de estar desarrollado bajo software privativo por lo que la Universidad de Los Andes como ente público debe cumplir con el Decreto 3.390.

De allí la importancia del diseño de un nuevo sistema web para la gestión de las Historias Clínicas Odontológicas, el cual permitirá llevar un proceso organizado en cuanto a la información de cada uno de los pacientes, mejorando los tiempos de búsqueda, atención y consultas por parte de los estudiantes de la carrera de Odontología, además brindará soporte a los profesores encargados de evaluar y validar cada Historia Clínica ingresada por los estudiantes, lo que permitiría evitar la desorganización de las historias, incluso pérdidas de algunos registros de pacientes, lo que conlleva a retrasos en la consulta y dificultad en la realización de un seguimiento detallado del tratamiento integral de cada paciente.

## **1.5 Objetivos**

Para la realización del presente trabajo, se estructuró el objetivo general, y los objetivos específicos de la siguiente manera

### **1.6 Objetivo General**

Desarrollar el nuevo Sistema Web de Historias Clínicas Odontológicas de la Facultad de Odontología sobre una plataforma de arquitectura abierta y de software libre.

## 1.7 Objetivos Específicos

- Analizar e Identificar el problema en el sistema web actual.
- Identificar y analizar las ideas preliminares y los requerimientos actuales por parte de los profesores de la Facultad de Odontología, para ser incluidos en el Sistema Web.
- Construir un modelo funcional jerárquico del sistema a implantar con las nuevas herramientas tecnológicas.
- Automatizar el proceso de registro, búsqueda de información y control de citas de los pacientes con las nuevas herramientas tecnológicas.
- Desarrollar la base de datos en el sistema manejador PostgreSQL.
- Controlar los elementos gráficos de diagnósticos asociados a cada paciente (radiografías y modelos dentales) con las nuevas herramientas tecnológicas.
- Importar y validar los datos del sistema actual al sistema nuevo.

## 1.8 Metodología

Para el desarrollo de la presente investigación se utilizó el Modelo de Proceso “Reloj”, propuesto por el Profesor Montilva (2010), conocido “White Watch”. Esta metodología plantea un conjunto de procesos bien estructurados y organizados que describen las fases, actividades, tareas y recursos necesarios para desarrollar y mantener las aplicaciones del software. La estructura de este modelo de proceso es principalmente jerárquica.

Este Modelo comprende 8 fases de desarrollo las cuales son ejecutables secuencialmente en el sentido de las agujas del reloj, comenzando por el análisis del dominio y finalizando en la fase de entrega del sistema.

El modelo de procesos está organizado en dos grupos de procesos complementarios:

- Los procesos gerenciales que incluyen los procesos de soporte, y
- Los procesos técnicos de desarrollo del producto de software

Esta manera de estructurar el marco metodológico permite que la ejecución de los procesos de desarrollo sea cíclica, iterativa y controlada. Los procesos gerenciales determinan, supervisan y controlan los ciclos, las iteraciones y la entrega de los productos parciales y finales del proyecto.

Cada ciclo de procesos técnicos produce una nueva versión del sistema (enfoque progresivo o evolutivo) o un nuevo subsistema del sistema en desarrollo, si se ha seleccionado el enfoque de desarrollo incremental. En cada ciclo se puede iterar entre los procesos técnicos a fin de corregir errores, introducir nuevos requisitos o, simplemente, mejorar el producto en desarrollo.

Son los procesos gerenciales quienes determinan el alcance de cada iteración y cuántas iteraciones se realizan por ciclo y describen las actividades que el líder del proyecto debe realizar para:

- Planificar, organizar y controlar el proceso de desarrollo del proyecto
- Asegurar la calidad del sistema mediante validaciones y verificaciones
- Gestionar los cambios en las especificaciones del producto

Los procesos técnicos son los que prescriben lo que debe hacer el grupo de desarrollo para elaborar un producto de software pequeño y poco complejo. Contemplan actividades relacionadas con el Modelado de Negocios o dominio, la Ingeniería de requisitos, el Diseño de software, el Aprovisionamiento de Componentes, la Implementación, las Pruebas y la Entrega de producto parcial o final.

El sistema web fue modelado con Lenguaje Unificado de Modelado (UML), es un lenguaje gráfico que permite visualizar, especificar, construir y documentar un sistema, ofreciendo una manera estándar de aspectos conceptuales tales como: procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados; todos estos los elementos gráficos que se diseñan sirven como documentación del sistema para futuras revisiones y mejoras del mismo.

## 1.9 Alcance

Desarrollar el nuevo sistema de Información Web para el manejo de Historias Clínicas Odontológicas de los pacientes tratados en la Facultad de Odontología de la Universidad de Los Andes, que permita el almacenamiento y recuperación de las Historias Clínicas, controlando algunos elementos gráficos de diagnósticos asociados a cada paciente, como las radiografías, el examen de la oclusión, y las partes gráficas de la evaluación periodontal, consultar e imprimir informes con carácter estadísticos, por medio del análisis del software prototipo desarrollado por Villareal (2003) a fin de establecer los requisitos funcionales y no funcionales que formarán parte del nuevo sistema, además de implementar los nuevos requerimientos presentes en la Historia Clínica actual.

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

## Capítulo 2

### Marco Teórico

Este capítulo abarca los fundamentos teóricos necesarios para el entendimiento y comprensión del proyecto, entre los cuales se definen algunos términos básicos empleados en la práctica odontológica, los sistemas de información, el Modelo de Proceso “Reloj” para el desarrollo de aplicaciones empresariales y el lenguaje de modelado unificado (UML) empleado para el desarrollo del sistema web. Generando una base teórica sobre las historias clínicas y herramientas de diseño que otorga al lector una idea del contenido del documento.

#### 2.1 Conceptos Básicos Empleados en Odontología

Con la finalidad de proveer una mejor comprensión, se detallan a continuación conceptos utilizados en la práctica Médica y en Odontología relacionados a la atención del paciente.

##### 2.1.1 Historia Clínica

La historia clínica es el registro ordenado, secuencial y permanente de todos los fenómenos clínicos. En ella quedan plasmados todos los datos obtenidos mediante el interrogatorio y la exploración (directa e indirecta), así como los relativos al diagnóstico, tratamiento y seguimiento del paciente, tal como se ha indicado con anterioridad.

Entre los objetivos de la historia clínica, desde una perspectiva asistencial, se encuentran la identificación de los pacientes y el conocimiento de enfermedades personales o familiares más importantes, con el fin de adaptar el tratamiento odontológico a las condiciones de salud física y mental del paciente, además prevenir las complicaciones frente a los tratamientos instaurados

(quimioprofilaxis). Como se señalaba en un apartado anterior, la historia clínica constituye asimismo una garantía médico legal de atención adecuada al paciente. Kustner (1999).

### **2.1.2 Diagnóstico**

“Identificación de un proceso o enfermedad mediante la evaluación específica de signos clínicos, síntomas, anamnesis, pruebas del laboratorio y técnicas especiales. Algunos tipos de diagnóstico son: diagnóstico clínico, diagnóstico de enfermería, diagnóstico de laboratorio, diagnóstico diferencial y diagnóstico físico” Mosby (2003, p.467).

### **2.1.3 Diagnóstico Clínico**

Según Mosby (2003) el diagnóstico clínico es un “Diagnóstico basado solamente en el conocimiento obtenido a través de la historia clínica y de la exploración física, sin utilizar pruebas de laboratorio o radiografías. (p.467).

### **2.1.4 Pronóstico**

Predicción de la posible evolución de una enfermedad basándose en el estado de la persona y en el curso habitual de la enfermedad observado en situaciones similares. Mosby (2003, p. 1269)

### **2.1.5 Tratamiento**

“Asistencia y cuidados proporcionados a un paciente para combatir, mejorar o prevenir la enfermedad, trastorno o lesión.” Mosby (2003, p. 1595)

### **2.1.6 Radiografía**

Producción de imágenes sombreadas sobre una emulsión fotográfica mediante la acción de la radiación ionizante. Las estructuras que son relativamente radiopacas (permiten pasar pocos rayos X a su través), como los huesos y las cavidades rellenas con un contraste radiopaco, dejan una sombra en la película. Mosby (1996)

### 2.1.7 Odontograma

Es un esquema de las arcas dentarias, diseñadas de manera gráfica y geoméricamente. En él se encuentran las características anatómicas de los dientes, así como de las particularidades y modificaciones, con fines de identificación Moreno (2008, pág. 118).

Permiten asentar los datos del examen del paciente, tratamientos realizados y pendientes por realizar, además, forman una parte imprescindible de la historia clínica odontológica del paciente. Por otra parte, sirve para registrar la información dental de un paciente adulto o infantil, así como la identificación propia del paciente, en el caso de la odontología forense, puede servir como identificación de una persona que haya padecido y no se pueda identificar por el rostro, huellas digitales o características particulares. Hernández (2014).

## 2.2 Conceptos empleados en el desarrollo de sistemas y descripción de algunas herramientas requeridas para el diseño.

En esta sección se abarcarán conceptos referentes al desarrollo de sistemas, se definirá que es un sistema, elementos de un sistema, los sistemas de información, además se describirán algunas herramientas requeridas para el diseño del presente trabajo.

### 2.2.1 Sistema

Senn James, 1993. Es un conjunto de elementos relacionados entre sí, para alcanzar un objetivo, operando sobre entradas.

Un sistema, es una entidad formada por componentes organizados que interactúan de forma en que las propiedades del conjunto no pueden deducirse por completo, estas propiedades son conocidas como propiedades emergentes. Pérez, J (2008).

#### 2.2.1.1 Elementos de un Sistema (Senn James, 1993)

**Entrada:** Constituyen la fuerza alimentadora de todo sistema. Puede estar constituida por información, energía y materiales. Esto es lo que procede del medio ambiente y entra al sistema.

**Proceso:** Es la acción que permite la transformación del insumo conforme a los fines y objetivos del sistema. Es el resultado neto de todas las actividades que se realizan para producir bienes o servicios.

**Salida:** Es el resultado final de la operación o procedimiento de un sistema, a través de salidas el sistema exporta resultados de sus operaciones hacia su medio ambiente.

**Retroalimentación:** Es un mecanismo mediante el cual una parte de la energía de salida retorna a la entrada.

**Ambiente:** Es el medio que envuelve externamente al sistema.

### 2.2.1.2 Clasificación de los Sistemas (Senn James, 1993)

#### En cuanto a su constitución:

Sistemas físicos o concretos. Están compuestos por equipos, maquinarias, objetos y cosas.

Sistemas abstractos. Están compuestos por conceptos, planes, hipótesis e ideas.

#### En cuanto a su naturaleza:

Sistemas cerrados. Son los sistemas que no presentan intercambio con el medio ambiente que los rodea, pues son herméticos a cualquier influencia ambiental.

Sistemas abiertos. Pueden ser comprendidos como un conjunto de partes en constantes interacción (lo que destaca la característica de la interdependencia de las partes) constituyendo un todo sinérgico (todo es mayor que la suma de las partes), orienta hacia determinados propósitos y en permanente relación de interdependencia con el ambiente externo.

### 2.2.1.3 Sistemas de Información

Según Kendall, K. (1997), señala que un sistema de información, “es un conjunto de elementos que interactúan entre sí, con el fin de apoyar las actividades de una empresa o negocio”.

Partiendo de esta definición, indica que las organizaciones con éxito utilizan la información como instrumento eficaz para la administración y han adquirido sistemas de información que responden a las necesidades de las personas.

Actualmente la necesidad de información en las organizaciones es mucha y su existencia es vital para alcanzar el éxito, los sistemas han evolucionado en su uso, comenzando con la sistematización de procesos operativos de las organizaciones como apoyo a este nivel para brindar información que sirva de base en el proceso de toma de decisiones.

#### **2.2.1.4 Diseño de Sistemas de Información**

Kendall, K. (1997), el diseño de sistemas se define como el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un sistema, con suficientes detalles como para permitir su interpretación y realización física.

#### **2.2.1.5 Beneficios de los sistemas de Información (Kendall K. 1997)**

- Acceso rápido a la información y por ende mejora en la atención a los usuarios.
- Mayor motivación en los medios para anticipar los requerimientos de las directivas.
- Generación de informes e indicadores, que permiten corregir fallas difíciles de detectar y controlar con un sistema manual.
- Posibilidad de planear y generar proyectos institucionales soportados en sistemas de información que presentan elementos claros y sustentados.
- Evita pérdida de tiempo recopilando información que ya está almacenado en bases para encontrar y manipular la información.
- Impulso a la creación de grupos de trabajo e investigación debido a la facilidad para encontrar y manipular la información.
- Soluciona el problema de falta de comunicación entre las diferentes instancias. En el ámbito directivo se hace más efectiva la comunicación.
- Organización en el manejo de archivos e información clasificada por temas de interés general y particular.
- Generación de nuevas dinámicas, utilizando medios informáticos como el correo electrónico, multimedia, tele-conferencia, acceso directo a base de datos y redes nacionales e internacionales.
- Aumento de la productividad, gracias a la liberación de tiempos en búsquedas y generación de información repetida.

### **2.2.2 Software Propietario**

Son programas de computación cuya licencia establece restricciones de uso, redistribución o modificación por parte de los usuarios, o requiere de autorización expresa del licenciador. (Decreto Nro. 3.390, 2004).

### **2.2.3 Software Libre**

Son programas de computación cuya licencia garantiza al usuario acceso al código fuente del programa y lo autoriza a ejecutarlo con cualquier propósito, modificarlo y redistribuir tanto el programa original como sus modificaciones en las mismas condiciones de licenciamiento acordadas al programa original, sin tener que pagar regalías a los desarrolladores previos. (Decreto Nro. 3.390, 2004).

### **2.2.4 HTML**

HTML es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla que corresponde a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto. Se trata de un formato abierto que surgió a partir de las etiquetas SGML (Standard Generalized Markup Language). Concepto traducido generalmente como “Estándar de Lenguaje de Marcado Generalizado” y que se entiende como un sistema que permite ordenar y etiquetar diversos documentos dentro de una lista. Este lenguaje es el que se utiliza para especificar los nombres de las etiquetas que se utilizarán al ordenar, no existen reglas para dicha organización, por eso se dice que es un sistema de formato abierto. Pérez & Gardey (2012)

### **2.2.5 Framework**

“Son software que ofrecen una infraestructura para la creación de otros programas. El “framework” contiene librerías de código y módulos ya listos que resumen las tareas de creación de elementos recurrentes en el desarrollo de aplicaciones, a la vez que define una arquitectura para el desarrollo de software” (Álvarez, s.f).

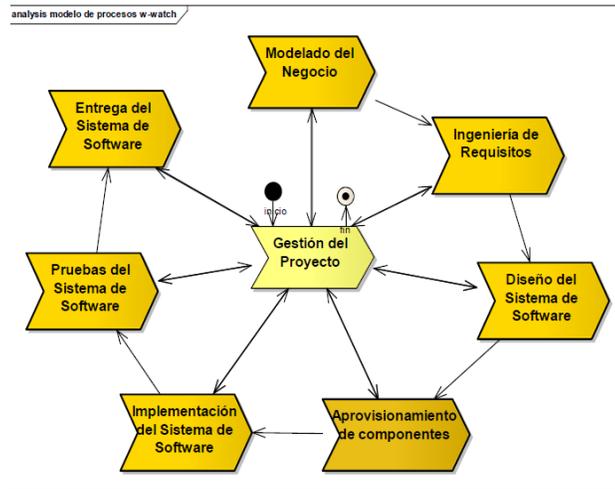
Existen diversos tipos de “framework”, El denominado “framework” multimedia, por ejemplo, es utilizado por lo general en editores para audio y video o en reproductores de medios, aunque además resulta útil para aplicaciones vinculadas a conversores de medios y videoconferencias.

El “framework” para aplicaciones web, en tanto, se utiliza para respaldar el desarrollo de servicios web, aplicaciones web y sitios web de carácter dinámico. Entre otras ventajas, se caracteriza por facilitar la reutilización de código y por brindar estructuras destinadas a plantillas y bibliotecas para tener acceso a bases de datos, entre los cuales destaca Laravel, Ruby on Rails, Django, AngularJS.

El “framework” de hojas de estilo en cascada (CSS o Cascading Syle Sheets), tal como se identifica a una librería de estilos genéricos. Estas herramientas pueden subdividirse de acuerdo a sus características en frameworks de CSS multipropósito o en frameworks de CSS con propósitos específicos.

### **2.2.6 Modelo de Proceso “Reloj” (White Watch)**

El método “White Watch” se enfoca particularmente en el desarrollo de aplicaciones empresariales. Una aplicación empresarial es aquella que apoya los procesos de negocio de una empresa de forma distribuida. Dos tipos de aplicaciones empresariales son las de comercio electrónico y sistemas de información web (SIW) que dan soporte a uno o más conjuntos de procesos de negocio, por medio de una interfaz que permite el intercambio de información a través de una red Intranet, Extranet o Internet. En la Figura 2.1 se muestra el diagrama de flujo de trabajo del Modelo de Procesos del método White Watch, a continuación, se describen cada una de las ocho fases del marco metodológico que han de ser aplicadas por el grupo para desarrollar una aplicación empresarial Montilva (2010):



**Figura 2.1. Modelo del Proceso “White Watch” por Montilva (2010).**

Fase 1. Modelado de Negocio: El propósito de esta fase es permitir al equipo de desarrollo, conseguir buena comprensión del entorno donde se va a desarrollar el sistema antes de comenzar la definición y especificación de los requerimientos del mismo. Establecer el alcance del sistema, Modelar las actividades de cada proceso del sistema, Identificar actores, y definir los roles del sistema.

Fase 2. Ingeniería de Requisitos: Su propósito es descubrir y definir los requisitos funcionales-no funcionales a ser satisfechos por el sistema en desarrollo. Se elaboran los diagramas de casos de uso, se elabora el diagrama preliminar de clases, se integran diagramas en documento de especificación de requisitos (DER).

Fase 3. Diseño del sistema de Software: En esta fase se define la estructura del sistema, además se diseña la interfaz Usuario/Sistema con sus perfiles, se elabora el diagrama jerárquico de las pantallas del sistema, se establece las características estéticas que debe tener la interfaz gráfica de la aplicación. Se diseña la base de datos, conceptual, luego implementable y por último el diseño físico. Se diseñan los módulos o componentes del Sistema.

Fase 4. Aprovechamiento de Componentes: En esta fase se instala la plataforma requerida para desarrollar el sistema (servidores web, sistemas manejadores de base de datos). Los componentes del software son codificados y probados individualmente, y se ejecutan pruebas sobre los módulos o componentes desarrollados.

Fase 5. Implementación del sistema de Software: Todos los componentes son ensamblados y se crea la base de datos en el SDBD seleccionado, en este caso PostgreSQL.

Fase 6. Pruebas del Sistema de Software: En esta fase se prueba el sistema en su ambiente operacional. Se Corrigen los errores detectados.

Fase 7. Entrega del sistema de Software: En esta fase se instala la aplicación, se diseñan y ejecutan las pruebas de instalación, se elabora la documentación o manual para el sistema, además se entrenan a los usuarios que interactuaran con el sistema.

www.bdigital.ula.ve

### 2.2.7 UML

El Lenguaje Unificado de Modelado (UML – Unified Modeling Language) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar cada una de las partes que comprende el desarrollo de un software (Rumbaugh, Jacobson & Booch, 2000).

Consta de un conjunto de notaciones gráficas que son usadas para modelar la estructura, funcionalidad, comportamiento e implementación de un sistema y organizar los modelos producidos y de un lenguaje formal que permite expresar formalmente restricciones acerca de los elementos modelados de un sistema. UML nace de la unificación de tres metodologías de análisis y diseño orientada a objeto, la metodología de Grady Booch para la descripción de conjuntos de objetos y sus relaciones, la técnica de modelado orientada a objetos (OMT: Object-Modeling Technique) de James Rumbaugh y la aproximación de Ivar2.3 UML 22 Jacobson (OOSE: Object Oriented Software Engineering) mediante la metodología de casos de uso. De estas tres metodologías de partida, las de Booch y Rumbaugh pueden ser descritas como centradas en objetos, ya que sus aproximaciones se enfocan hacia el modelado de los objetos que componen el sistema, su relación y colaboración. Por otro

lado, la metodología de Jacobson es más centrada a usuario, ya que todo en su método se deriva de los escenarios de uso, Guillen (2003).

UML 2.0 consta de 13 tipos de diagramas clasificados en dos grupos:

1. Diagramas para Modelado Estructural:

- Diagrama de Clases
- Diagramas de Objetos
- Diagramas de Componentes
- Diagramas de Estructura Compuesta
- Diagramas de Despliegue
- Diagramas de Paquetes

2. Diagramas para Modelado del Comportamiento:

- Diagramas de Casos de usos
- Diagramas de Interacción
- Diagramas de Secuencias
- Diagramas de Estado
- Diagramas de Actividades

Los diagramas para modelado estructural se enfatizan en los elementos que deben existir en el sistema modelado, mientras que para el modelado del comportamiento se enfatizan en lo que sucede en el sistema modelado.

En este trabajo nos limitaremos a explicar los diagramas utilizados como lo son: diagrama de casos de uso, diagrama de secuencia, diagrama de clases, diagrama de componente y diagrama de despliegue.

**Diagramas de Casos de Uso:**

Es una descripción de un proceso fin-a-fin relativamente largo que típicamente incluye varias etapas o transacciones, NO es una etapa o actividad individual de un proceso, describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista de un usuario; permiten definir los límites del sistema y las relaciones entre el sistema y su entorno.

Se representa en el diagrama por una elipse, denota un requerimiento solucionado por el sistema. Cada caso de uso es una operación completa desarrollada por los actores y por el sistema en un dialogo. El conjunto de casos de uso representa la totalidad de operaciones desarrolladas por el sistema.

**Actor:**

Es un usuario (en el sentido amplio de la palabra: ser humano, o aplicación) del sistema, que necesita o usa algunos de los casos de uso. Se representa mediante, un acompañado de nombre significativo, si es necesario.

**Relaciones en un diagrama de caso de uso:**

Entre los elementos de un diagrama de casos de uso se pueden presentar tres tipos de relaciones, representadas por líneas dirigidas entre ellos (del elemento dependiente al independiente).

Comunica: Relación entre un actor y un caso de uso, denota la participación del actor en el caso de uso determinado.

Usa: Relación entre dos casos de uso, denota la inclusión del comportamiento de un escenario en otro.

Extiende: Relación entre dos casos de uso, denota cuando un caso de uso es una especialización de otro.

**Diagrama de Secuencia:**

Muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo. Esta descripción es importante ya que permite detallar los casos de uso, aclarándolos al nivel de mensajes de los objetos existentes, como también muestra el uso de los mensajes de las clases diseñadas en el contexto de una operación.

**Diagrama de clases:**

Son utilizados durante el proceso de análisis y diseño de sistemas informáticos, donde se crea el diseño conceptual de la información que se manejará en el sistema, estos diagramas son estructuras estáticas que dan una visión general del conjunto de clases existentes en el sistema modelado y las relaciones existentes entre cada una de ellas. Son estáticos porque muestra las relaciones entre las distintas clases, pero no especifica que sucede cuando ocurre alguna interacción entre ellas.

### **Diagrama de Componentes:**

Representa las dependencias entre componentes de software, incluyendo componentes de código fuente, componentes del código binario, y componentes ejecutables. Un módulo de software se puede representar como un componente.

### **Diagrama de Despliegue:**

El diagrama de despliegue muestra como está configurado el hardware y el software del sistema, para ello se modelan los nodos físicos y los medios de comunicación entre ellos y luego se especifica por cada nodo los componentes que en el residen.

## **2.2.8 Modelo Vista Controlador**

La arquitectura de diseño de modelo-vista-controlador (MVC) especifica que una aplicación consta de un modelo de datos, de información de presentación y de información de control. Es un patrón importante puesto que se utiliza tanto en componentes gráficos básicos hasta sistemas empresariales; aunque en un principio el MVC fue creado para aplicaciones de escritorio, se ha adaptado como arquitectura para diseñar e implementar aplicaciones web en los principales lenguajes de programación y la mayoría de los "frameworks" modernos (o alguna adaptación del MVC), entre ellos podemos mencionar a Ruby on Rails, Django, AngularJS y muchos otros más, estos "frameworks" se diferencian básicamente en la interpretación de como las funciones de la arquitectura se dividen entre cliente y servidor. Alvarez (2014). MVC no pretende discriminar entre capa de negocio y capa de presentación, pero si pretende separar la capa visual gráfica de su programación y acceso a datos, algo que mejora el desarrollo y mantenimiento sus componentes en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

### **2.2.8.1 Modelo**

Se encarga de los datos, generalmente consultando la base de datos, por tanto, abarcará el acceso a la información y también actualizará el estado, por lo que tendremos todas las gestiones de las operaciones en las tablas como los "selects", "updates", "inserts", entre otros comandos SQL correspondientes.

### 2.2.8.2 Vistas

Son las representaciones visuales de los datos como su nombre nos hace entender, es todo lo que está asociado con la interfaz gráfica. Ni el modelo ni el controlador se preocupan de cómo se verán los datos, esa responsabilidad es únicamente de la vista, es decir, es el código que nos permite visualizar los estados de nuestra aplicación en HTML. En las vistas nada más se manejan códigos HTML y PHP que nos permite mostrar la salida.

### 2.2.8.3 Controlador

Se encarga de controlar o administrar cuidadosamente los recursos necesarios para completar una tarea al recibir las solicitudes del usuario y se encarga de solicitar los datos al modelo y de comunicárselos a la vista de la aplicación.

En la Figura 2.2 se explica el ciclo de vida de la arquitectura MVC en el cual:

- El usuario realiza una petición.
- El controlador captura la petición del usuario.
- El controlador llama al modelo.
- El modelo interactúa con la base de datos, y retorna la información al controlador.
- El controlador recibe la información y la envía a la vista.
- La vista procesa la información recibida y la entrega de una manera visualmente entendible al usuario.

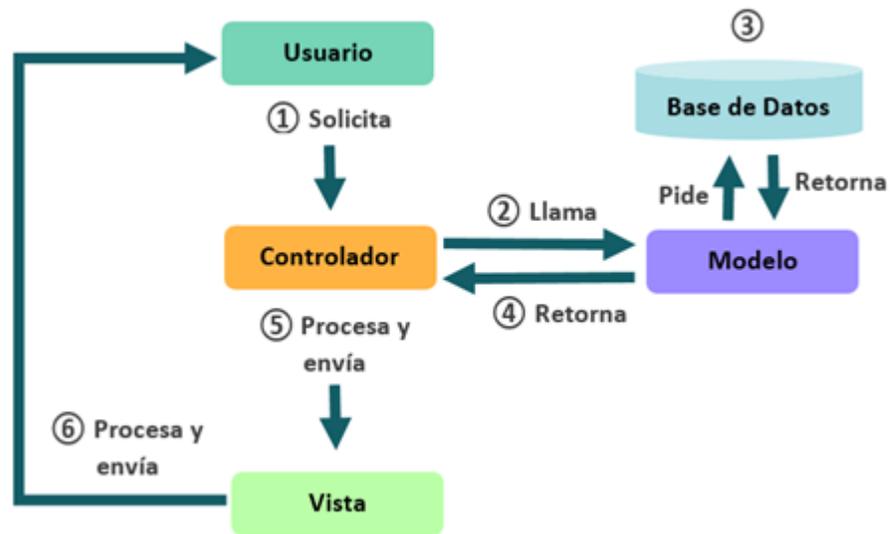


Figura 2.2. Modelo MVC

### 2.2.9 Laravel

Laravel fue creado por Taylor Otwell en el 2011, definido como "framework", está hecho para la arquitectura tradicional Modelo Vista Controlador (MVC), plantea el uso de rutas con cierres (Routes with closures) para programar la interacción HTTP (Hypertext Transfer Protocol) o Protocolo de transferencia de hipertexto directamente como una función anónima asociada a una ruta, esto tiene la ventaja de reducir la cantidad de código, especialmente cuando sólo necesitamos incluir una funcionalidad.

Al ser un código modular y extensible con su estructura de paquetes y "drivers" nos facilitan ampliar su funcionalidad, robustez y seguridad, hace que el uso de los datos en Laravel no sea complejo, mediante el ORM ("Object-Relational Mapping") llamado "Eloquent" basado en el patrón "active record" la interacción con las bases de datos es totalmente orientada a objetos, La funcionalidad interna del ORM es realizar la correspondencia entre los objetos de PHP (Hypertext Preprocessor) a las tablas en la base de datos, para el caso en que la persistencia de los datos de la aplicación es proporcionada por una base de datos, ofreciendo compatibilidad con la gran mayoría de las bases de datos del mercado actual y permitiendo la migración de datos de una forma fácil y segura-Otro punto a favor es el manejo de rutas de la aplicación, las URL (Uniform Resource Locator) amigables y control de enlaces auto-actualizables facilitando el mantenimiento de un sitio web, cuenta también con una herramienta de interfaces llamada Artisan que permite programar tareas por líneas de comando, las aplicaciones estarán

blindadas con el uso de "Middleware", ya que se encarga de analizar y filtrar las llamadas HTTP en el servidor, se encarga de verificar la información relacionada a un usuario registrado, de evitar problemas de tipo scripts de sitios cruzados (XSS) y otras medidas de seguridad.

## **2.2.10 PostgreSQL**

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarles a otras bases de datos comerciales.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. ([www.postgresql.org.es](http://www.postgresql.org.es)).

### **2.2.10.1 Base de Datos**

Según Márquez (2011), "Una base de datos es un conjunto de datos almacenados en memoria externa que están organizados mediante una estructura de datos. Cada base de datos ha sido diseñada para satisfacer los requisitos de información de una empresa u otro tipo de organización, como, por ejemplo, una universidad o un hospital".

Actualmente, y debido al desarrollo tecnológico, la mayoría de las bases de datos están en formato digital, siendo este un componente electrónico, por tanto, se ofrece un gran rango de soluciones a la problemática de almacenamiento de datos con programas que manipulen ese conjunto de datos.

### 2.2.10.2 Base de Datos relacionales

Una base de datos relacional es una colección de elementos de datos organizados en un conjunto de tablas formalmente descritas desde la que se puede acceder a los datos o volver a montarlos de muchas maneras diferentes sin tener que reorganizar las tablas de la base. La base de datos relacional fue inventada por E.F. Codd en IBM en 1970.

Es el modelo utilizado actualmente para representar y administrar datos dinámicamente. Su idea primordial es el uso de "relaciones". Estas relaciones podrían considerarse como conjuntos de datos llamados "tuplas". Viendo cada relación como si fuese una tabla que está formada por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla). Besemmel (1998)

El lenguaje más universal para generar las consultas, insertar, actualizar y borrar datos, a bases de datos relacionales es SQL, Structured Query Language o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales. A diferencia de otros modelos como el jerárquico y el de red, el modelo relacional el dónde y cómo se almacenen los datos no tienen relevancia. Esto le da importante ventaja al ser más fácil de utilizar y entender a usuario esporádico en base de datos. La información puede ser recuperada o almacenada mediante "consultas" brindando una amplia flexibilidad y poder para administrar la información. Los sistemas gestores de bases de datos (del inglés Database Management System o DBMS), es un software muy específico que permiten almacenar y acceder a los datos de forma rápida y estructurada, destinado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Entre las características de los sistemas gestores de base de datos encontramos mínima redundancia, los datos son independientes lógicamente y físicamente, integridad de los datos, optimiza las consultas complejas, brinda seguridad de acceso, auditoría, respaldo y recuperación de datos, proporciona fácil acceso a través de lenguajes de programación estándar. (EcuRed).

## Capítulo 3

### Modelado y Análisis de Requerimientos

En este capítulo se identifica y detalla el sistema de negocios para el cual se desarrolla el sistema Web, además se presenta la definición y especificación de los requisitos de la aplicación, cumpliendo con las dos primeras fases de la metodología del proceso “reloj” (White Watch).

En la elaboración de un proyecto de software el análisis de la institución y de los requerimientos e ideas del cliente, le permiten al grupo de desarrollo obtener:

- Un mejor entendimiento del o los problemas a resolver.
- Una definición del grupo de actores que van a interactuar con el sistema.
- La identificación de los requerimientos de hardware y software necesarios para la implementación del sistema, acordes con las necesidades del cliente.

#### 3.1 Análisis del Dominio de la aplicación

Para este caso el análisis se basó en el estudio del modelo actual de la Historia Clínica, además de la revisión de la primera versión del SIHCO, así como en las entrevistas semanales con un grupo de profesores de la Facultad de Odontología de la Universidad de Los Andes y el tutor del proyecto. Reuniones en las cuales, se planteaban las necesidades no realizadas en el sistema anterior y las posibles soluciones del nuevo sistema a realizar.

### 3.1.1 Identificación del proceso

A continuación, se describe el proceso llevado a cabo por una persona para ser atendido en la Clínica Integral del Adulto (Facultad de Odontología).

Primero deberá pasar por la oficina de admisión, donde serán recaudados sus datos personales, posteriormente será evaluado para determinar su condición clínica, la cual, establecerá la próxima instancia a la que tendrá que asistir, bien sea la o emergencia de ser necesario. Estos procesos se explican con más detalles a continuación:

#### **Admisión (Pre-registro):**

Cuando una persona desea ser atendida en la Clínica Integral del Adulto, deberá primero pasar por el departamento de admisión, donde se registrará de forma única en el sistema, para ellos deberá suministrar la siguiente información: apellidos, nombres, fecha de nacimiento, ocupación, teléfono y dirección, una vez registrados estos datos, esa persona pasa a ser paciente, culminando así la primera etapa.

#### **Clínica de Diagnósticos (Triaje):**

En esta etapa se evaluará la condición clínica del paciente, evaluación que se reflejará a la hora de remitir al paciente a las distintas instancias con las que cuenta la Facultad de Odontología. Esta evaluación es realizada por estudiantes cursantes del último año de la carrera de Odontología, bajo la supervisión del profesor encargado. Si el estado del paciente amerita atención inmediata, el mismo será remitido a emergencia donde será tratado a la brevedad posible, luego, el paciente será remitido a la Clínica de Diagnostico o a alguna de las Clínicas Integrales, según sea el caso.

**Clínica Integral I:** Destinada para actividades de diagnóstico y tratamientos simples, como control de placa, higiene oral y radiología periapical.

**Clínica Integral II:** En esta clínica se desarrollan actividades de diagnóstico y tratamientos medios, como restauraciones compuestas, endodoncia, y ajuste oclusal, entre otros.

**Clínica Integral III:** la tercera y última de las clínicas en las cuales se desarrollan actividades de diagnóstico y tratamiento complejas, como amalgamas con refuerzos internos, incrustaciones complejas, y cirugía.

La estructuración de las tres clínicas no implica que el paciente debe pasar por las clínicas I y/o II para poder asistir a la Clínica Integral III, aunque el número de la clínica (I, II, III) implican cierta jerarquía en los tratamientos que pueden realizar, no existe dependencias entre ellas. Véase Figura 3.1.

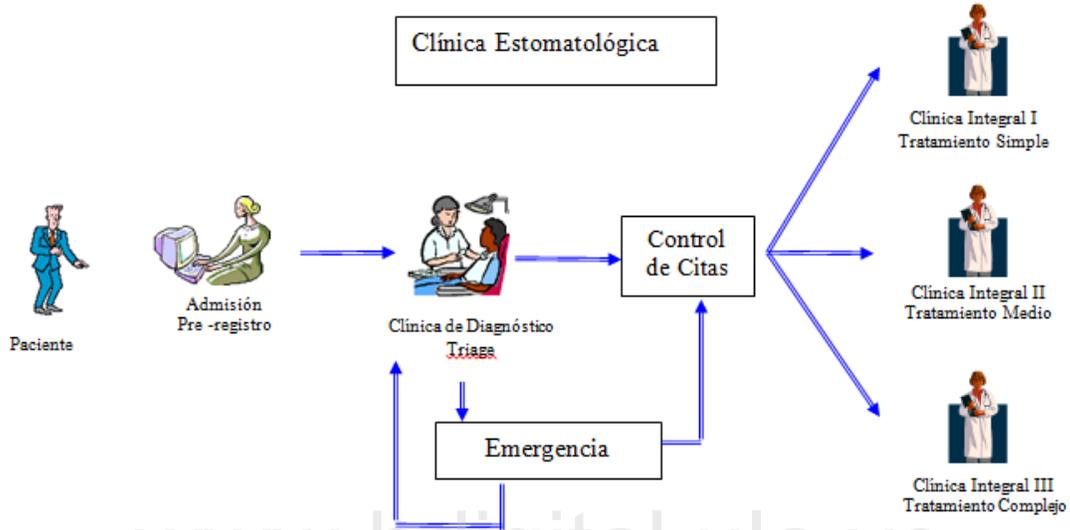
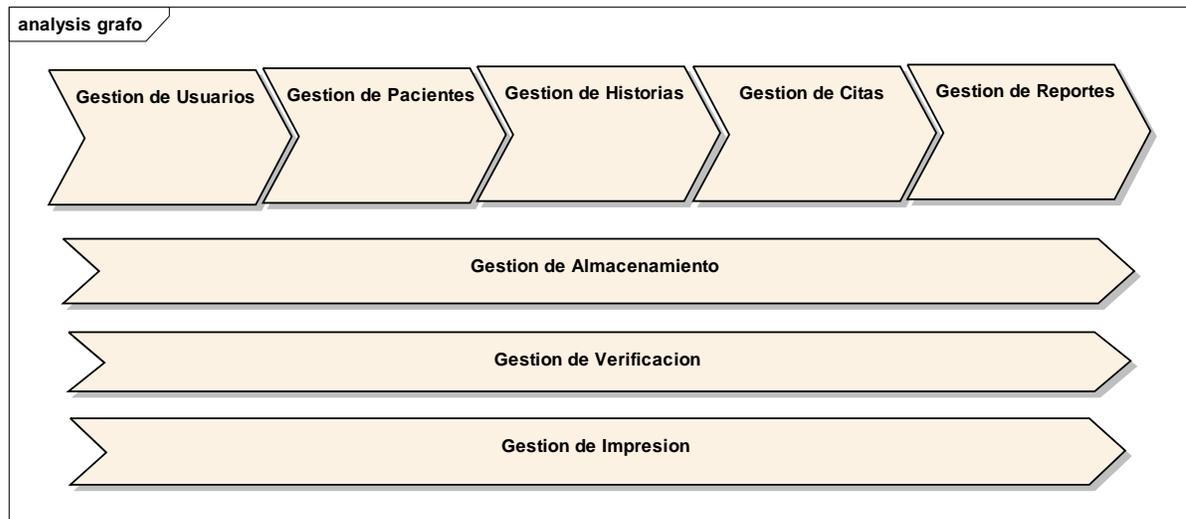


Figura. 3.1. Modelo del proceso para la atención de un paciente

Una vez que el paciente ha sido tratado, el estudiante que lo asistió, deberá dirigirse al profesor tutor y mostrar la historia clínica del paciente evaluado con sus respectivas actualizaciones (registro manual), quien decidirá el ingreso o no de esos datos al sistema. De ser aprobado, el estudiante ingresará la nueva información a la historia clínica. Posteriormente el profesor tutor ingresará al sistema para verificar y validar dichos datos.

### 3.1.2. Modelado del proceso de negocio



**Figura 3.2. Cadena de Valor de SIHCO**

En la Figura 3.2 se puede observar la cadena de valor que forma parte del sistema de negocio, muestra la relación entre los procesos primarios o fundamentales y los procesos que sirven de apoyo.

- Gestión de usuarios: Es el proceso encargado de llevar el crear, actualizar, eliminar y visualizar los usuarios del sistema.
- Gestión de historias: Es el proceso que se encarga de llevar el control de registro, consulta y validación de las historias ingresadas en el sistema.
- Gestión de citas: Es el proceso que se encarga de manejar el control perteneciente a las citas generadas por las distintas clínicas su inserción, modificación y reprogramación.

### 3.2 Descubrimiento de requisitos funcionales:

Para el comienzo del desarrollo del sistema, es necesario conocer el conjunto de requerimientos, características o condiciones que establecen los odontólogos, en cuanto a los datos de insumo del paciente, del proceso o tratamiento, así como del producto obtenido luego de ser asistida la

persona, todo esto con el propósito de satisfacer las exigencias y cubrir las expectativas de los actores involucrados en relación a aligerar el registro para llevar el control de la historia de cada uno de los pacientes. Cabe destacar que el requerimiento más importante es que el software sea fácil de utilizar y brinde seguridad en cuanto al archivo y registro de las historias odontológicas. En el sistema de Historias Clínicas Odontológicas se determinaron los siguientes requisitos:

1. Gestión de usuarios
2. Gestión de Pacientes
3. Acceder a los componentes y procesos de la aplicación limitado por los roles y privilegios de los usuarios.
4. Registrar las Historias Clínicas Odontológicas asociadas a los pacientes
5. Consultar las Historias Clínicas Odontológicas asociadas a los pacientes
6. Actualizar y/o modificar datos en la Historia Odontológica
7. Validar la Historia Clínica Odontológica asociada a los pacientes
8. Emitir reportes mensuales

Debido a la gran cantidad de usuarios que accederán a sistema y al tipo de información que se maneja, se hace necesaria la creación de un mecanismo que identifique, jerarquice y a la vez restrinja el ingreso al mismo, garantizando así la integridad de la Base de Datos y de la información allí contenida. Esta parte del proyecto nos permitirá identificar a los distintos actores del sistema y su interacción con el mismo, para así poder definir los grupos de usuarios y sus respectivas interfaces.

A continuación, se describen los actores que van a interactuar con el sistema:

### **3.2.1 Identificación de Actores.**

Los actores de este proyecto representan a las personas que van a interactuar con el sistema, se identificaron a cuatro tipos de actores: Administrador, Profesor tutor, Profesor Jefe de Clínica, y Estudiante.

El actor “Administrador” no necesariamente estará vinculado directamente con las actividades clínicas de la institución, dado que sus funciones principales se remiten al funcionamiento y mantenimiento del sistema, y a la creación, modificación y eliminación de los usuarios.

El actor “Profesor Tutor” representa al grupo de profesores que laboran en la Clínica Integral del Adulto y se encargan de atender a los pacientes y supervisar a los estudiantes que interactúan con el sistema. Estarán a cargo de la validación de las Historias Clínicas ingresadas al sistema.

El actor “Profesor Jefe de Clínica” representa al grupo de profesores que además de gozar con los privilegios del actor “Profesor Tutor”, podrán modificar la historia de un paciente de considerarlo necesario mientras que la historia no haya sido validada. Por otra parte, el Actor “Jefe de Clínica” estará a cargo del buen funcionamiento de la Clínica Integral del Adulto, para ello el sistema le brindará algunos informes tales como: La cantidad de pacientes tratados por clínicas, los distintos tratamientos realizados por clínicas, entre otros.

El actor “Estudiante” representa al grupo de estudiantes que laboran en la Clínica Integral de Adulto y están encargados de atender a los pacientes y registrar las Historias Clínicas.

Como se puede apreciar en la Figura 3.3 el modelo Jerárquico de los actores antes descritos.

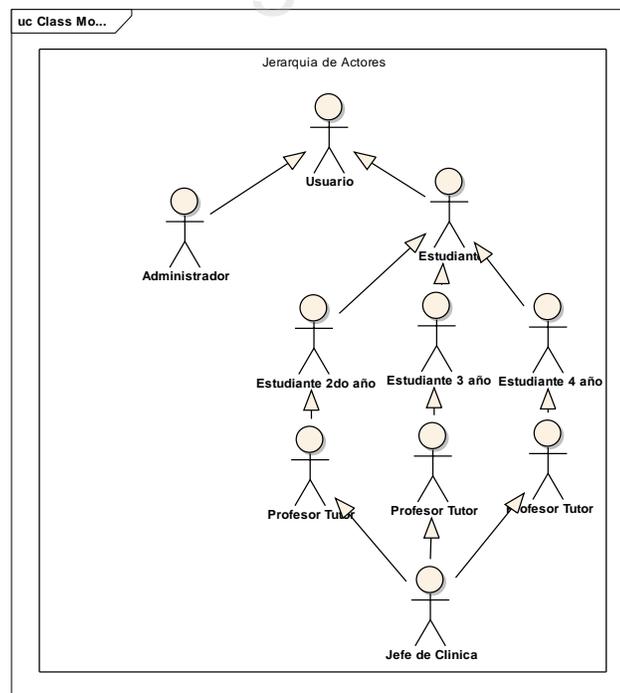


Figura.3.3. Jerarquía de Actores

### 3.2.2 Casos de Uso

Con los diagramas de Casos de uso buscamos representar la forma como los distintos actores anteriormente descritos, interactúan con el sistema en desarrollo. Estos casos de uso forman parte de la elaboración de la arquitectura de la aplicación, específicamente se emplean para describir la vista de uso contenida en la vista arquitectónica, más adelante en el capítulo 4 detallaremos la arquitectura de la aplicación.

#### 3.2.2.1 Definición de los Casos de Uso

Para el presente trabajo de acuerdo a los actores involucrados y a los requerimientos establecidos, se determinaron los siguientes casos de uso:

Caso de Uso	Descripción
CU1. Ingresar al Sistema	Permite acceder al sistema a los usuarios registrados
CU2. Ingresar Paciente	Permite ingresar un paciente al sistema
CU3. Modificar Paciente	Permite modificar pacientes registrados
CU4. Eliminar Paciente	Permite eliminar pacientes registrados
CU5. Ingresar Usuario	Permite ingresar usuarios al sistema
CU6. Modificar Usuario	Permite modificar usuarios registrados
CU7. Eliminar Usuario	Permite eliminar usuarios registrados
CU8. Anexar Historia	Permite ingresar la historia clínica de pacientes
CU9. Consultar Historia	Permite consultar las historias clínicas de pacientes
CU10. Modificar Historia	Permite modificar las historias clínicas de pacientes
CU11. Consultar Estadísticas	Muestra datos estadísticos en relación a las Historias Clínicas tratadas
CU12. Validar Historia	Permite validar las historias clínicas de pacientes

**Tabla 3.1. Casos de uso del sistema web**

### 3.2.2.2 Descripción de los Casos de Uso

En la descripción de los casos de uso aparece los actores involucrados, las precondiciones y postcondiciones, los flujos básicos y los alternativos de cada uno de los casos de uso existentes. Se realizar la descripción desde la Tabla 3.2 hasta la Tabla 3.13.

<b>CU1. Ingresar al Sistema</b>	
<b>Actores Principales</b>	Administrador, Profesor Jefe de Clínica, Profesor Tutor, Estudiante
<b>Condiciones de Entrada</b>	Estar registrado en el sistema
<b>Condiciones de Salida</b>	Ingreso exitoso al sistema
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. El actor selecciona la pestaña Ingresar.</li> <li>2. El sistema muestra la ventana solicitando los datos para ingresar.</li> <li>3. El actor ingresa sus datos y selecciona el botón Entrar.</li> </ol>
<b>Flujos Alternativos</b>	<ol style="list-style-type: none"> <li>3.a El actor ingresa mal un dato.</li> <li>3.b El sistema solicita que ingrese de nuevo los datos por que alguno esta incorrecto.</li> <li>4. El actor olvido su clave y selecciona olvide clave.</li> <li>5. El sistema solicitara correo para enviarle su clave.</li> </ol>
<b>Notas</b>	

**Tabla 3.2. Caso de uso Ingresar al sistema**

<b>CU5. Agregar Usuario</b>	
<b>Actores Principales</b>	Administrador, Profesor Jefe de Clínica
<b>Condiciones de Entrada</b>	Haber entrado al sistema
<b>Condiciones de Salida</b>	Mensaje informativo con el registro exitoso del usuario

<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>4. El actor selecciona la pestaña, Registrar Usuario.</li> <li>5. El sistema le muestra un formulario para ingresar los datos del usuario.</li> <li>6. El actor introduce los datos del usuario.</li> <li>7. El actor guarda el nuevo registro</li> </ol>
<b>Flujos Alternativos</b>	3.a El sistema pide los campos obligatorios que estén en blanco
<b>Notas</b>	

**Tabla 3.3. Descripción Caso de Uso Agregar Usuario**

<b>CU7. Modificar Usuario</b>	
<b>Actores Principales</b>	Administrador, Profesor Jefe de Clínica
<b>Condiciones de Entrada</b>	Haber entrado al sistema
<b>Condiciones de Salida</b>	Mensaje informativo que se actualizaron los datos
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. El actor selecciona la pestaña, Modificar Usuario.</li> <li>2. El sistema muestra la lista de usuarios.</li> <li>3. El actor selecciona la acción editar</li> <li>4. El sistema le muestra un formulario para modificar los datos.</li> <li>5. El actor ingresa los nuevos datos y guarda.</li> <li>6. El sistema muestra un mensaje informando que se actualizaron los datos</li> </ol>
<b>Flujos Alternativos</b>	6.a El sistema muestra los campos que tengan error y pide que los modifiquen.
<b>Notas</b>	

**Tabla 3.4. Descripción caso de uso Modificar usuario**

<b>CU8. Eliminar Usuario</b>	
<b>Actores Principales</b>	Administrador, Profesor Jefe de Clínica
<b>Condiciones</b>	Haber entrado al sistema

<b>de Entrada</b>	
<b>Condiciones de Salida</b>	Usuario Eliminado
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. El actor selecciona la pestaña, Modificar Usuario.</li> <li>2. El sistema muestra la lista de usuarios.</li> <li>3. El actor selecciona la acción eliminar</li> <li>4. El sistema elimina el usuario</li> </ol>
<b>Flujos Alternativos</b>	6.a El sistema muestra la lista de los usuarios registrados.
<b>Notas</b>	

**Tabla 3.5. Descripción caso de uso: Eliminar Usuario**

<b>C11. Consultar Estadística</b>	
<b>Actores Principales</b>	Administrador, Profesor Jefe de Clínica
<b>Condiciones de Entrada</b>	Haber entrado al sistema
<b>Condiciones de Salida</b>	Lista de Reportes asociados a los datos registrados en el sistema
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. El actor selecciona la pestaña, Consultar Reportes.</li> <li>2. El sistema carga los datos en los distintos reportes.</li> </ol>
<b>Flujos Alternativos</b>	Ninguno
<b>Notas</b>	

**Tabla 3.6. Descripción caso de uso: Consultar Historia.**

<b>CU9. Consultar Historia</b>	
<b>Actores Principales</b>	Administrador, Profesor Jefe de Clínica, Profesor Tutor

<b>Condiciones de Entrada</b>	Haber entrado al sistema
<b>Condiciones de Salida</b>	Información de la Historia de un Paciente
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. El actor selecciona la pestaña, Consultar Historia.</li> <li>2. El sistema carga la lista de pacientes registrados en el sistema.</li> <li>3. El actor selecciona el paciente asociado a la Historia que desea consultar.</li> <li>4. El sistema muestra la Historia Clínica del Paciente por secciones.</li> <li>5. El actor selecciona la sección de la historia a consultar.</li> <li>6. El sistema carga los datos que tenga almacenados</li> </ol>
<b>Flujos Alternativos</b>	Ninguno
<b>Notas</b>	

**Tabla 3.7. Descripción caso de uso: Consultar Historia**

<b>CU2. Agregar Paciente</b>	
<b>Actores Principales</b>	Administrador, Profesor Jefe de Clínica, Profesor Tutor, Estudiante
<b>Condiciones de Entrada</b>	<ol style="list-style-type: none"> <li>1. Haber entrado al sistema</li> <li>2. El paciente no debe existir en el sistema</li> </ol>
<b>Condiciones de Salida</b>	Mensaje informativo con el registro exitoso del paciente
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. El actor selecciona la pestaña, Registrar Paciente.</li> <li>2. El sistema le muestra un formulario para ingresar los datos del paciente.</li> <li>3. El actor introduce los datos del paciente.</li> <li>4. El actor guarda el nuevo registro</li> </ol>
<b>Flujos Alternativos</b>	3.a El sistema pide los campos obligatorios que estén en blanco
<b>Notas</b>	

**Tabla 3.8. Descripción caso de uso: Agregar Paciente**

<b>CU3. Modificar Paciente</b>	
<b>Actores Principales</b>	Administrador, Profesor Jefe de Clínica, Profesor Tutor, Estudiante
<b>Condiciones de Entrada</b>	<ol style="list-style-type: none"> <li>1. Haber entrado al sistema</li> <li>2. El paciente debe estar registrado en el sistema.</li> </ol>
<b>Condiciones de Salida</b>	Mensaje informativo que se actualizaron los datos
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. El actor selecciona la pestaña, Paciente.</li> <li>2. El sistema muestra la lista de pacientes.</li> <li>3. El actor selecciona la acción editar.</li> <li>4. El sistema le muestra un formulario para modificar los datos.</li> <li>5. El actor ingresa los nuevos datos y guarda.</li> <li>6. El sistema muestra un mensaje informando que se actualizaron los datos</li> </ol>
<b>Flujos Alternativos</b>	6.a El sistema muestra los campos que tengan error y pide que los modifiquen.
<b>Notas</b>	

**Tabla 3.9. Descripción caso de uso: Modificar Paciente**

<b>CU4. Eliminar Paciente</b>	
<b>Actores Principales</b>	Administrador, Profesor Jefe de Clínica
<b>Condiciones de Entrada</b>	<ol style="list-style-type: none"> <li>1. Haber entrado al sistema</li> <li>2. El paciente debe estar registrado en el sistema.</li> </ol>
<b>Condiciones de Salida</b>	Paciente Eliminado
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. El actor selecciona la pestaña, Paciente.</li> <li>2. El sistema muestra la lista de pacientes.</li> <li>3. El actor selecciona la acción eliminar</li> <li>4. El sistema elimina el paciente</li> </ol>
<b>Flujos Alternativos</b>	6.a El sistema muestra la lista de los pacientes registrados.

Notas	
-------	--

**Tabla 3.10. Descripción caso de uso: Eliminar Paciente**

<b>CU8. Anexar Historia</b>	
<b>Actores Principales</b>	Profesor jefe de Clínica, Profesor Tutor, Estudiante
<b>Condiciones de Entrada</b>	<ol style="list-style-type: none"> <li>1. Haber entrado al sistema</li> <li>2. El paciente debe estar registrado previamente en el sistema</li> </ol>
<b>Condiciones de Salida</b>	Insertar Información de la Historia de un Paciente
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. El actor selecciona la pestaña, Anexar Historia.</li> <li>2. El sistema carga la lista de pacientes registrados en el sistema.</li> <li>3. El actor podrá buscar por número de cedula.</li> <li>4. El actor selecciona el paciente asociado a la Historia que desea ingresar.</li> <li>5. El sistema muestra la Historia Clínica en una barra de navegación que permitirá acceder de forma directa a las distintas partes de la historia para ingresar los nuevos datos.</li> <li>6. El sistema confirma los datos almacenados</li> </ol>
<b>Flujos Alternativos</b>	Ninguno
Notas	

**Tabla 3.11. Descripción caso de uso: Anexar Historia**

<b>CU12. Validar Historia</b>	
<b>Actores Principales</b>	Profesor jefe de Clínica, Profesor Tutor
<b>Condiciones de Entrada</b>	<ol style="list-style-type: none"> <li>3. Haber entrado al sistema</li> <li>4. El paciente debe estar registrado previamente en el sistema</li> </ol>
<b>Condiciones de Salida</b>	Validar la Historia de un Paciente
<b>Flujo de</b>	<ol style="list-style-type: none"> <li>7. El actor selecciona la pestaña, Consultar Historia.</li> </ol>

<b>Eventos</b>	<p>8. El sistema carga la lista de pacientes registrados en el sistema.</p> <p>9. El actor podrá buscar por número de cedula.</p> <p>10. El actor selecciona el paciente asociado a la Historia que desea ingresar.</p> <p>11. El sistema muestra la Historia Clínica en una barra de navegación que permitirá acceder de forma directa a las distintas partes de la historia para validar las distintas secciones de la historia.</p> <p>12. El actor selecciona la sección que desee validar.</p> <p>13. El sistema confirma la validación de la Historia Clínica.</p>
<b>Flujos Alternativos</b>	Ninguno
<b>Notas</b>	

**Tabla 3.12. Descripción caso de uso: Validar Historia**

<b>CU10. Modificar Historia</b>	
<b>Actores Principales</b>	Profesor Jefe de Clínica, Profesor Tutor, Estudiante
<b>Condiciones de Entrada</b>	<p>5. Haber entrado al sistema</p> <p>6. El paciente debe estar registrado previamente en el sistema</p>
<b>Condiciones de Salida</b>	Modificar la Historia de un Paciente
<b>Flujo de Eventos</b>	<p>14. El actor selecciona la pestaña, Consultar Historia.</p> <p>15. El sistema carga la lista de pacientes registrados en el sistema.</p> <p>16. El actor podrá buscar por número de cedula.</p> <p>17. El actor selecciona el paciente asociado a la Historia que desea ingresar.</p> <p>18. El sistema muestra la Historia Clínica en una barra de navegación que permitirá acceder de forma directa a las distintas partes de la historia para modificar las distintas secciones de la historia para el caso en el que la historia no haya sido validada.</p> <p>19. El actor selecciona la sección que desee modificar.</p> <p>20. El sistema confirma la modificación de la Historia Clínica.</p>
<b>Flujos</b>	Ninguno

Alternativos	
Notas	

Tabla 3.13. Descripción caso de uso: Modificar Historia

### 3.2.2.3 Diagramas de Casos de Uso

De acuerdo a los casos de uso definidos en la Tabla 3.1, se muestran los diagramas de caso de uso para los distintos actores del sistema. Para la descripción se utilizan distintos estereotipos de tipo <<CRUD>> que permiten resumir las actividades de creación, actualización, visualización y eliminación, además se cuenta con el estereotipo <<CR>> que indica solo las actividades de creación y visualización. Véase Figura 3.4, 3.5, 3.6, 3.7 y el Anexo A

#### 3.2.2.3.1 Casos de Uso del actor: Administrador.

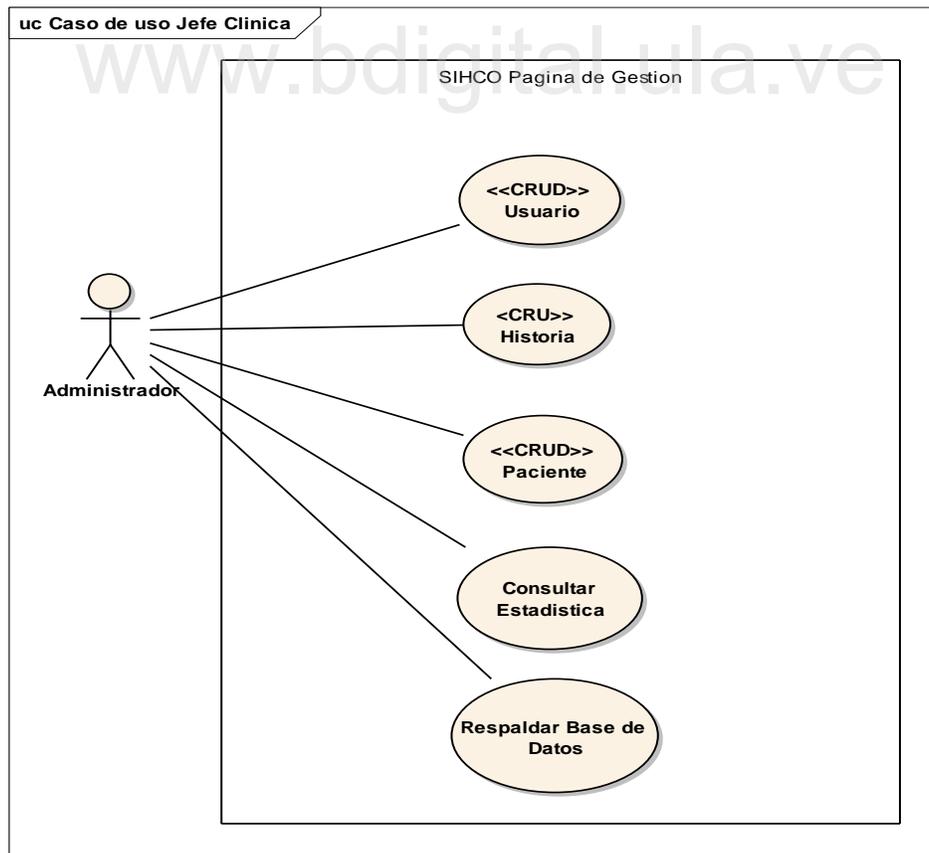


Figura 3.4. Caso de uso para el rol Administrador

3.2.2.3.2 Casos de Uso del actor: jefe de Clínica.

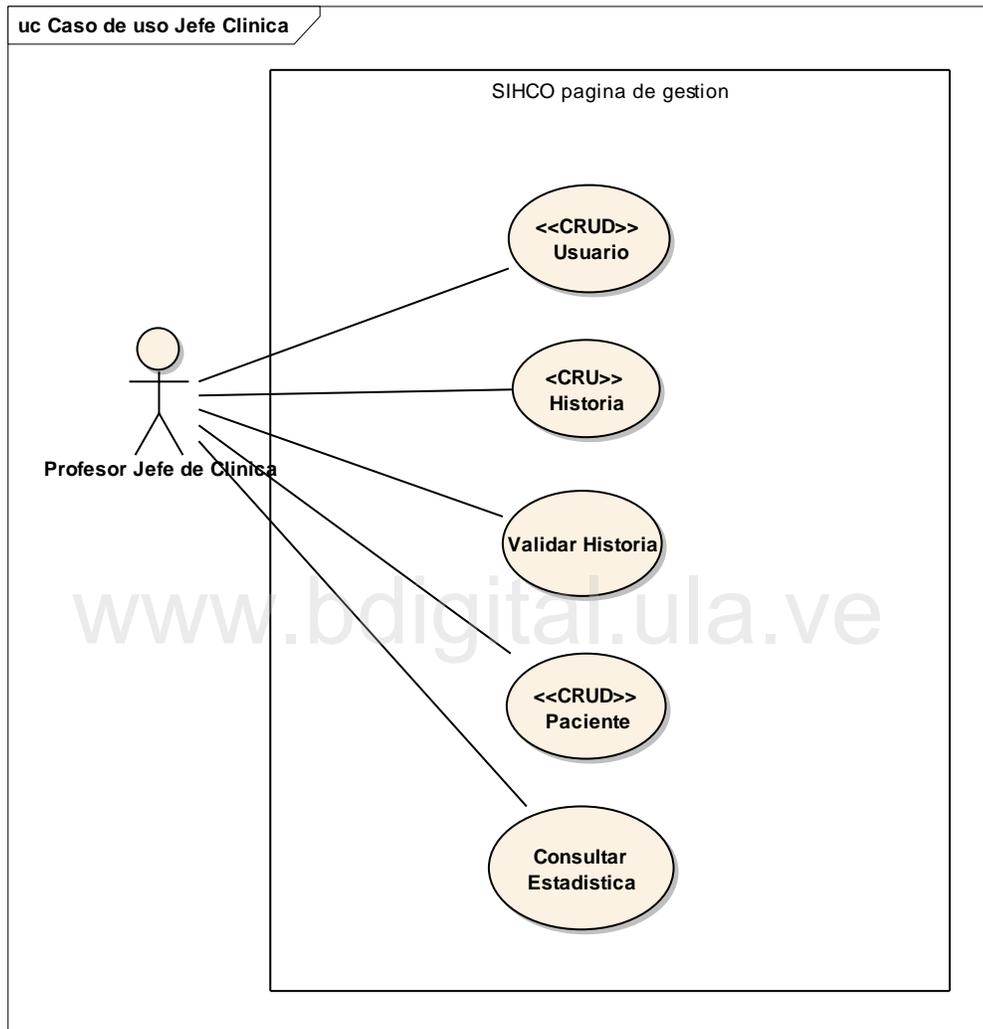


Figura 3.5. Caso de uso rol: Profesor Jefe de Clínica

### 3.2.2.3.3 Casos de Uso del actor: Profesor Tutor.

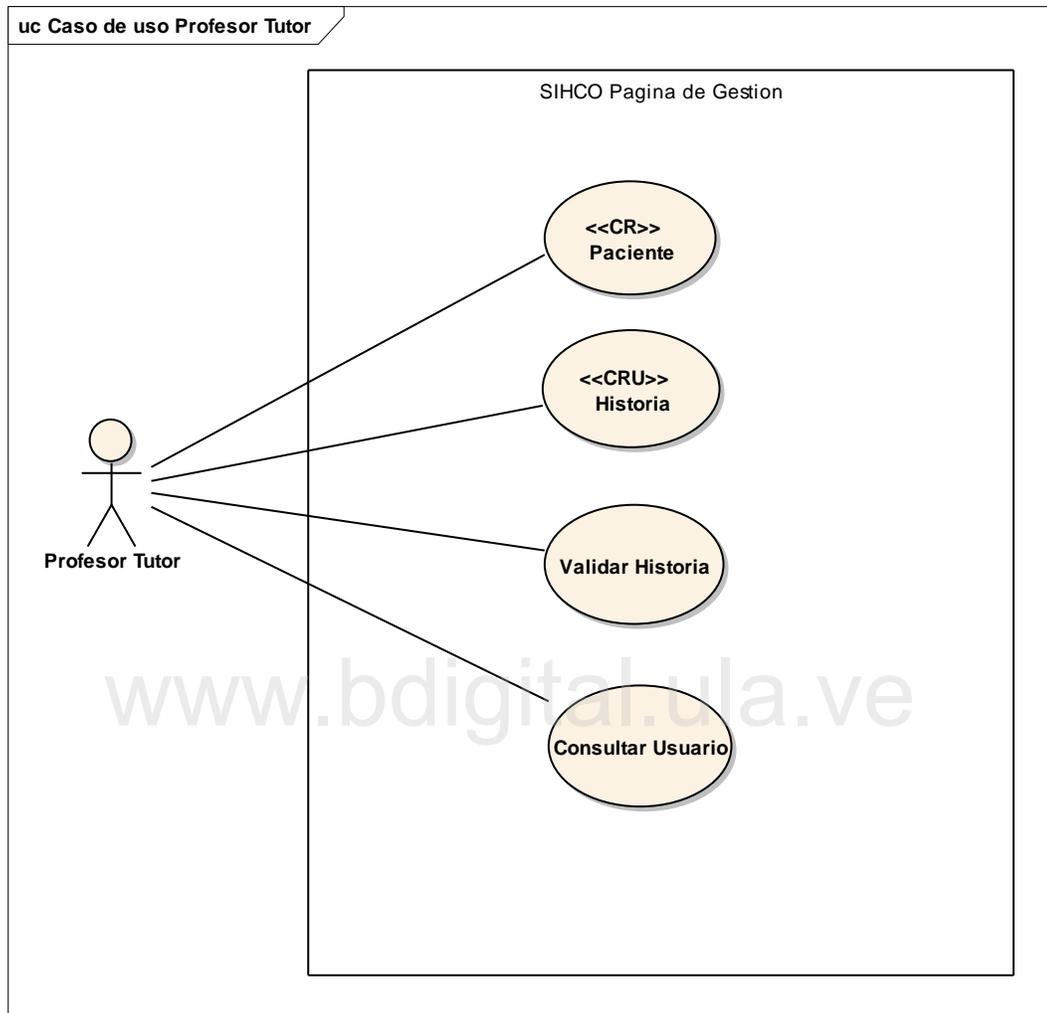


Figura 3.6. Caso de uso rol: Profesor Tutor.

### 3.2.2.3.4 Casos de Uso del actor: Estudiante.

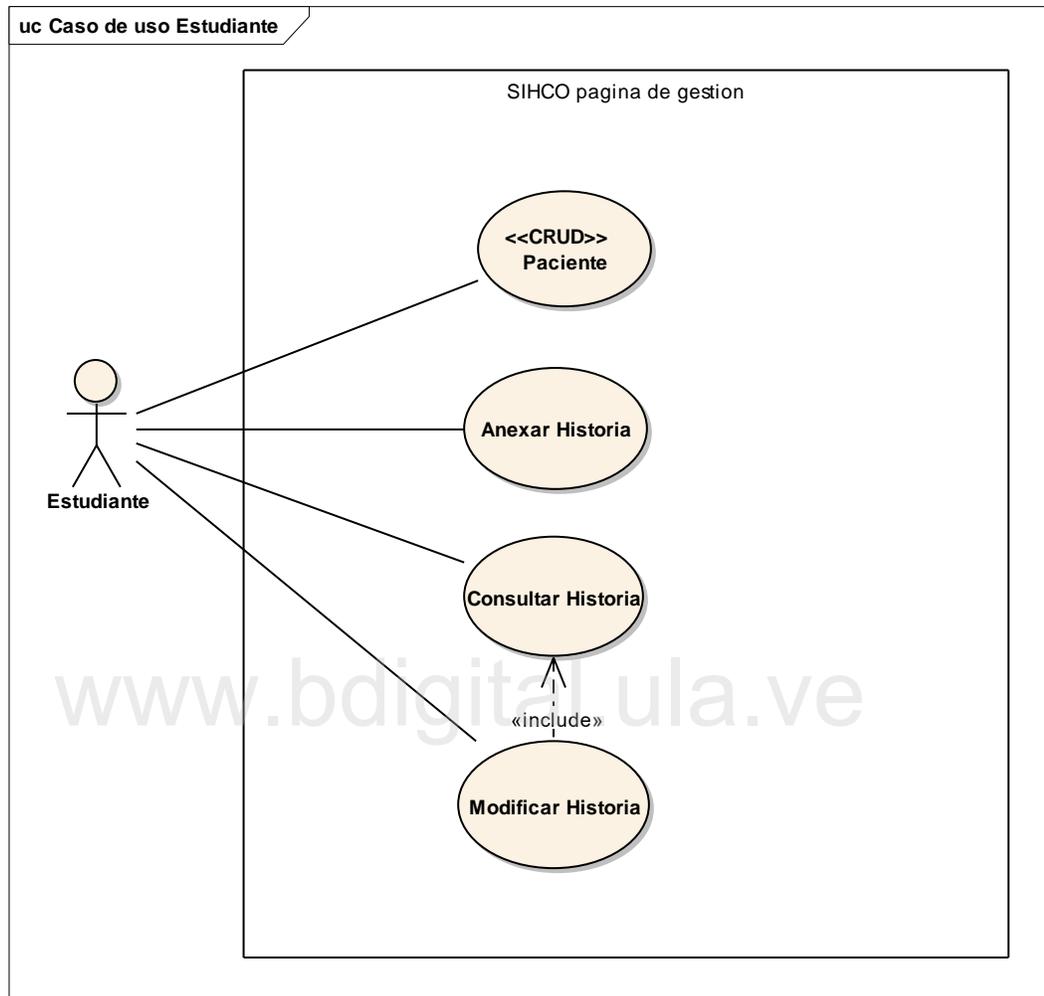


Figura 3.7. Caso de uso rol: Estudiante.

### 3.3 Descubrimiento de requisitos No funcionales:

No se refieren directamente a las propiedades funcionales del sistema, sino a sus propiedades emergentes o a restricciones adicionales en el sistema o en el proyecto de desarrollo de software.

Las propiedades emergentes Son aquellas que resultan del sistema como un todo y que es muy difícil o imposible atribuirle a un componente particular de éste. Por ejemplo, la fiabilidad, tiempo de respuesta, usabilidad, capacidad de almacenamiento, entre otros.

Para el desarrollo del sistema se establecieron los siguientes:

1. Para el servidor: Sistema Operativo Linux, con servicio Apache Http Sever, PostgreSQL como manejador de base de datos, PHP como lenguaje de programación para el código ejecutado.
2. El acceso al sistema web sólo puede realizarse a través de la intranet de la Facultad de Odontología.
3. Las respuestas a las consultas al servidor de base de datos deben ser lo más rápidas y eficientes posible, tomando en cuenta que los usuarios no deben notar algún error en el sistema.
4. El tiempo de respuesta del sistema al solicitar un reporte de datos deben ser lo más rápidas y eficientes posible, debido a los usuarios no deben notar algún error en el sistema.
5. El sistema debe contar con un manual de usuario estructurado adecuadamente.
6. El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.
7. La aplicación web debe poseer un diseño “Responsive” a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivos tableta y teléfonos inteligentes.
8. Debe especificarse un plan de recuperación ante fallas para el sistema a ser desarrollado.
9. El sistema debe ser robusto, no susceptible a fallas menores.
10. Poseer una política de respaldo de la base de datos a fin de evitar la pérdida completa de las historias clínicas de los pacientes. Para satisfacer este requisito no funcional la política de respaldo será diaria.

## Capítulo 4

### Diseño del Sistema

En este capítulo se elabora un diseño de la arquitectura de la aplicación que sea apropiada a los requisitos especificados y que establezca los subsistemas de la aplicación, los componentes de cada subsistema, las conexiones entre estos componentes y las restricciones que regulan la arquitectura. En esta fase se relacionan los requisitos obtenidos previamente junto con: la arquitectura del sistema, la identificación de subsistemas, la descripción de las vistas arquitectónicas, el diseño de la interfaz de usuario/sistema, el diseño de la base de datos y la instalación de la plataforma de desarrollo, que son la base fundamental para llevar a cabo la arquitectura solicitada.

#### **4.1 Definición de la estructura inicial de la aplicación.**

##### **4.1.1 Definición de las metas de diseño:**

Para la definición de las metas de diseño nos basaremos en la arquitectura de software de tres capas propuesta por el modelo de proceso “reloj” (White Watch). La arquitectura por capas es un modelo de desarrollo de software el cual tiene como objetivo la separación o desacoplamiento de las partes que componen un sistema. La primera capa es de presentación, en esta se implementa el interfaz usuario/sistema, la segunda capa es la de lógica de negocios donde se realiza la funcionalidad de la aplicación empresarial, esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él, y por último se encuentra la capa que se encarga de la administración de los datos de la aplicación, conocida como capa de datos. De esta forma con la arquitectura por capas, es

sencillo y mantenible crear diferentes interfaces sobre un mismo sistema sin requerirse cambio alguno en la capa de datos o lógica.

El patrón de diseño Modelo, Vista y Controlador (MVC), está asociado a la idea de 3 capas, el mismo se centra en la secuencia de ejecución, desde que se produce un evento en la capa de presentación hasta que el mismo es atendido de forma completa, separando los datos y la lógica del negocio de la interfaz de usuario, guardando la relación con la arquitectura de capas, en el que la capa de presentación serían los componentes de la Vista, la capa lógica serían los Controladores que reciben las peticiones del usuario y se envían respuestas tras el proceso, y por último la capa de datos sería los Modelos que se encarga de acceder a ellos, y realizar el proceso de almacenamiento de datos.

Para obtener una mayor eficiencia en el diseño del sistema se deben determinar cuáles de los requisitos obtenidos en el capítulo de modelado y análisis de requerimientos están relacionados directamente con el diseño arquitectónico del sistema.

#### **4.1.1.1 Requisitos relacionados con la capa de presentación:**

Los requisitos relacionados con esta capa son los que definen la interacción del sistema con los usuarios, los mensajes de usuarios y captura de los datos. De la lista de requisitos expuestos en el apartado anterior se determina que los requisitos relacionados con esta capa son aquellos que hacen referencia a que datos se deben mostrar en los reportes, como deben ser presentados, los datos que debe capturar el sistema cada vez que el usuario desee ingresar, insertar, y consultar la Historia Clínica Odontológica. Los requisitos son 2,3,4 y 8 de la lista de requisitos funcionales descrita en el capítulo 3 en el apartado 3.2 del capítulo anterior y 3, 6 y 7 de la lista de requisitos no funcionales descrita en el capítulo 3 en el apartado 3.3.

#### **4.1.1.2 Requisitos relacionados con la capa de lógica de negocios:**

En esta capa se encuentran los requisitos que implementan la funcionalidad del sistema, como son las funciones que requieren los usuarios y automatizan los flujos de trabajo, además de aquellos que manejan los datos. De los requisitos descritos en el capítulo anterior se encuentran aquellos que se encargan de manejar los datos y realizar cálculos para la información. Los mismos son 5 y 6 de la lista de requisitos funcionales expuesta en el capítulo 3 en el apartado 3.2 y los requisitos 1, 2 de la lista de los requisitos no funcionales descrita en el capítulo 3 en el apartado 3.3.

#### **4.1.1.3 Requisitos relacionados con la capa de datos:**

Esta capa está asociada con aquellos requisitos que se relacionan con la administración de los datos de la aplicación. Los requisitos asociados a esta capa son los descritos en el capítulo 3 en el apartado 3.2 específicamente 2,3,4, 5, 8 y 10 para los requisitos funcionales.

#### **4.1.2 Identificación de los subsistemas:**

El sistema web en estudio forma parte de la Clínica Integral del Adulto ubicada en la Facultad de Odontología de la Universidad de Los Andes, el cual permite realizar el manejo y gestión las Historias Clínicas odontológicas de los pacientes allí tratados. El sistema propuesto se puede dividir en dos subsistemas, lo cual permite una mejor comprensión y otorga una mayor eficiencia a la hora de diseñarlo, estos subsistemas son:

1. Sistema de Registro: Comprende todas las páginas que muestran el registro del usuario y paciente, en cuanto a datos personales se refiere. Se vincula a los casos de usos descritos en las Tablas 3.2, 3.3, 3.4, 3.7, 3.8 y 3.9 del capítulo anterior, en los que se describen los procesos para registrar usuarios y pacientes de manera administrativa.
2. Sistema de administración: Comprende todas las páginas que muestran los ingresos de las Historias Clínicas odontológicas, Reportes mensuales. Este subsistema está vinculado a los casos de usos descritos en las Tablas 3.5, 3.6, 3.10, 3.11 y 3.13, del capítulo anterior.

#### **4.1.3 Descripción de la vista arquitectónica.**

La vista arquitectónica representa un aspecto parcial de una arquitectura de software que muestran propiedades específicas del sistema. Para la descripción de la arquitectura de la aplicación se detalla a través de la elaboración de las diferentes perspectivas o vistas que la componen, como son: Uso, Comportamiento, datos, componentes y despliegue.

### 4.1.3.1 Vistas de Uso

Para la definición de las vistas de uso se determinan los casos de usos asociados, se definen un conjunto de escenarios de utilización y se revisa la consistencia y coherencia de las vistas. Para el presente trabajo las vistas de uso son los descritos en el capítulo 3 en el apartado 3.2.2.3 de los diagramas de caso de usos.

### 4.1.3.2 Vista de comportamiento

Para la vista de comportamiento se definen los elementos de acción y reacción entre la aplicación y los actores, para ello se realizan diagramas de secuencias que permiten representar los mensajes intercambiados por un conjunto de objetos durante un escenario. En esta sección se mostrarán los diagramas correspondientes a los casos de uso descritos en las Tablas 3.11 y 3.12. Véase Figura 4.1, 4.2 y Anexo B.

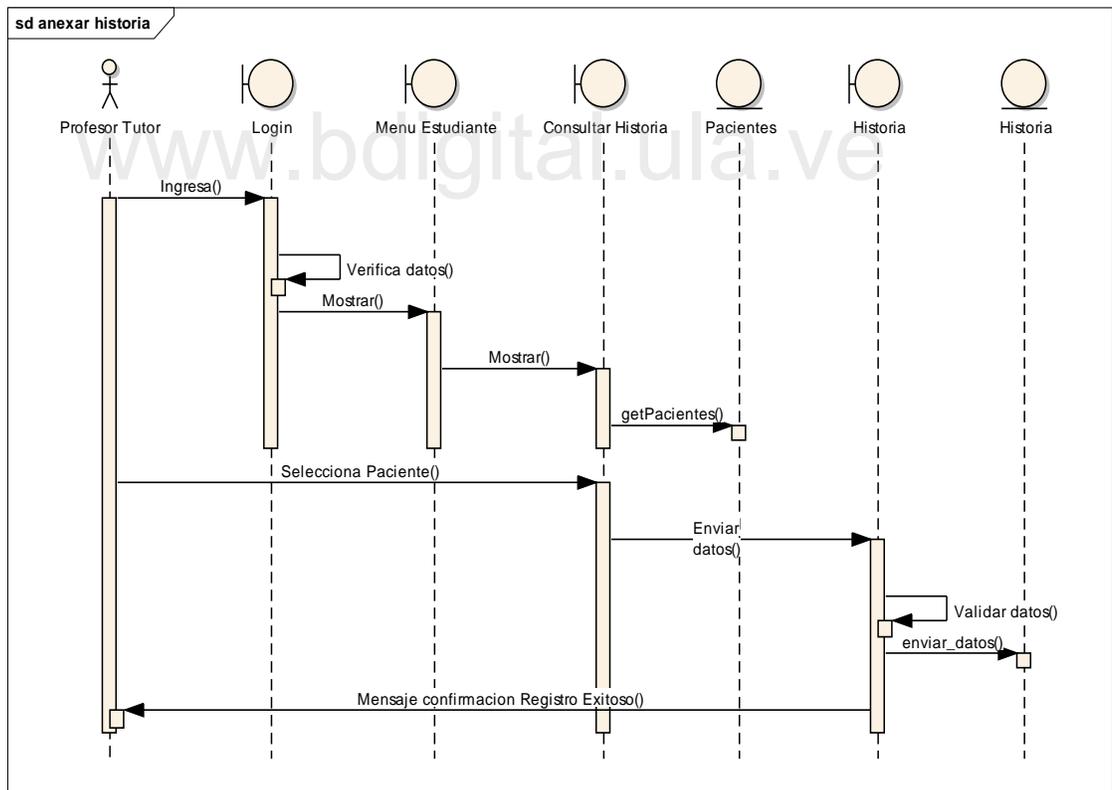


Figura 4.1. Diagrama de secuencia: Anexar Historia para el rol Estudiante.

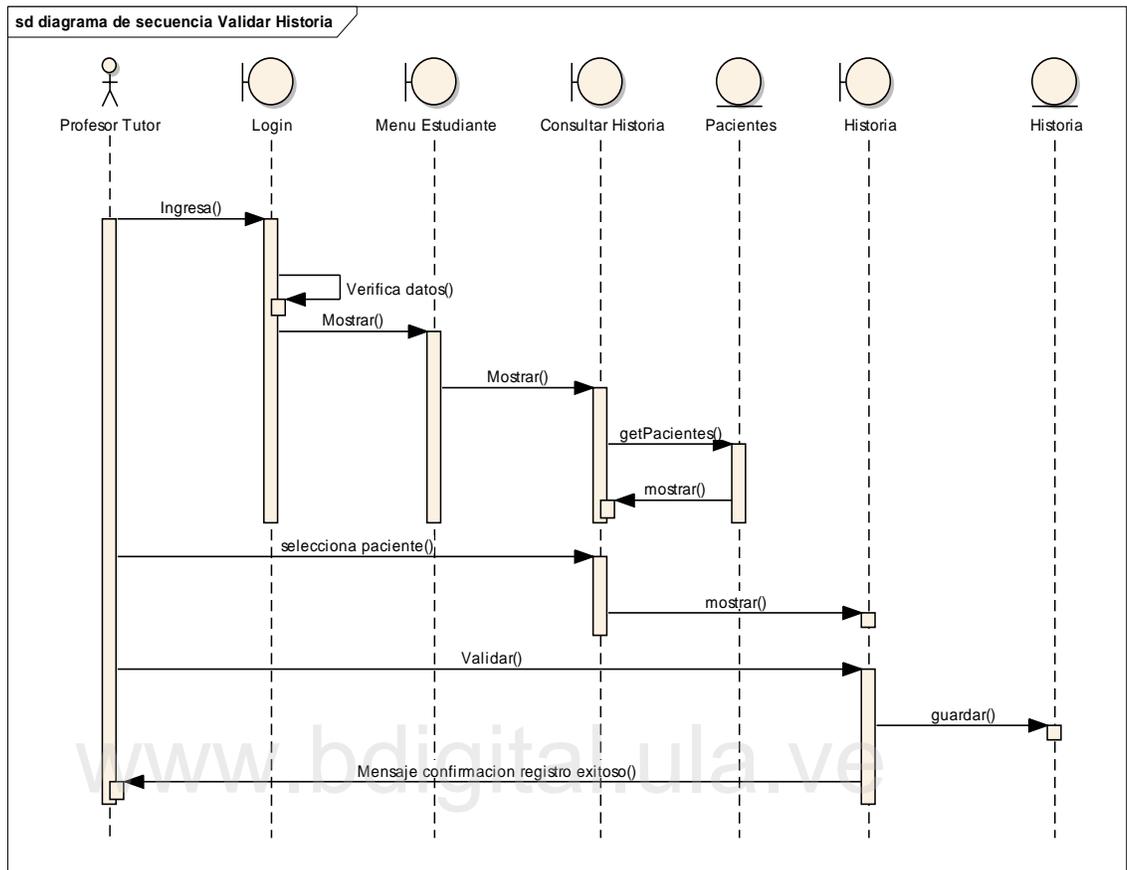


Figura 4.2. Diagrama de secuencia: Validar Historia para el rol Profesor Tutor.

#### 4.1.3.3 Vista de datos

En la vista de datos se construye el diagrama de clases, se determina la asociación entre clases de objetos y subsistemas de la aplicación, para el presente trabajo siguiendo la metodología incremental de Montilva (2010) se realizaron 3 versiones, para efecto de este documento presentaremos la versión final, cabe resaltar de igual forma que el diagrama de clase para la primera versión abarcaba pocas relaciones entre clases, solo la inserción de la primera parte de la historia conocida como datos básicos, luego en la segunda versión, se abarco las dos secciones de la historia datos básico y datos pre clínicos y para finalizar en la tercera versión se abarco toda la historia con la última sección de los datos clínicos como se puede apreciar en el Figura 4.3.

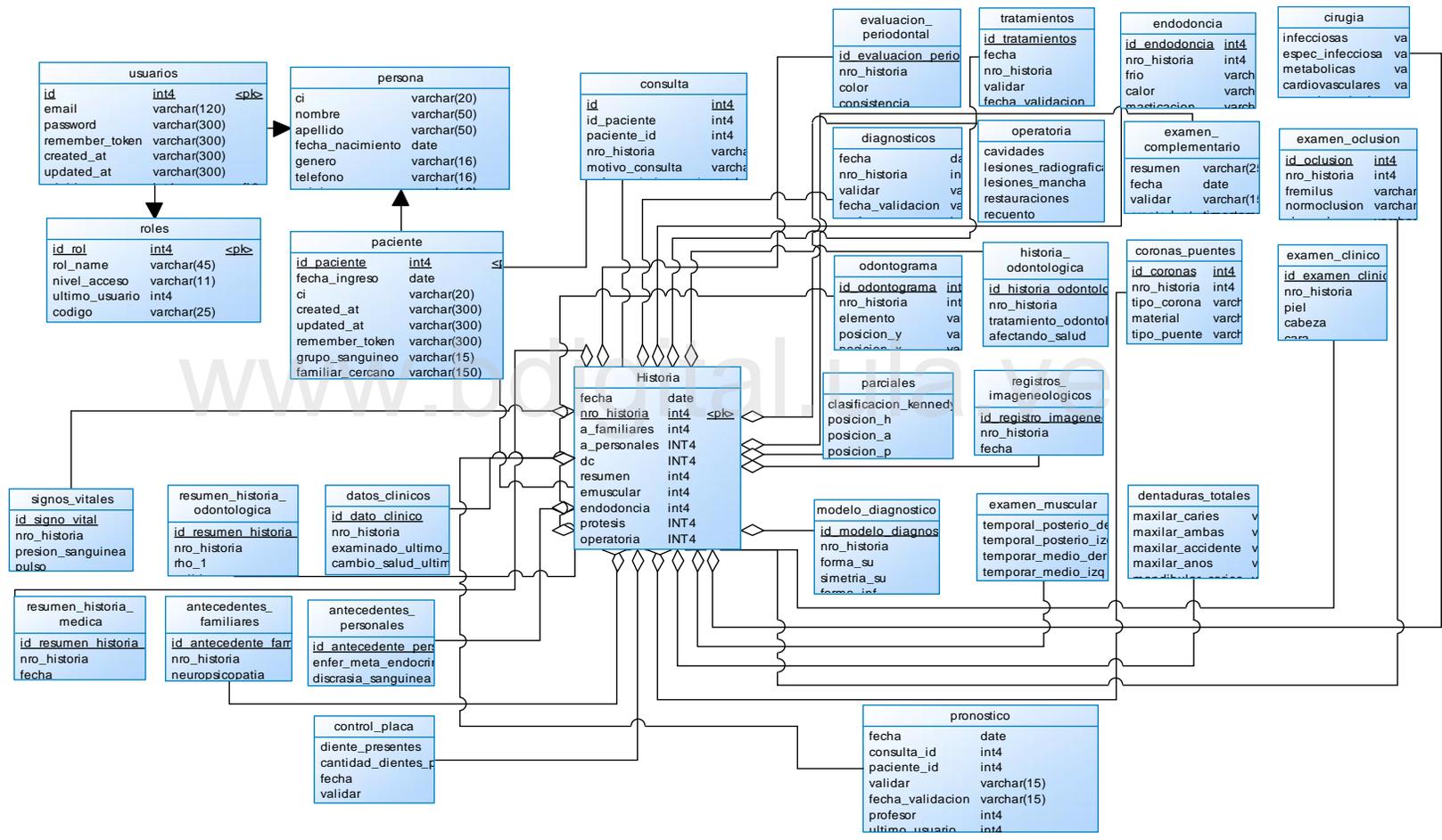


Figura 4.3. Diagrama de Clase

#### 4.1.3.3.1 Modelo de la Base de Datos Relacional del sistema SIHCO.

Una base de datos relacional es un conjunto de una o más tablas estructuradas en registros (tuplas) y campos (columnas), que se vinculan entre sí por un campo en común. A esta manera de construir la base de datos se le denomina modelo relacional. Para lograr obtener la base de datos relacional, las entidades mostradas en el diagrama de clases de la sección 4.3, deben pasar por el proceso de normalización, esto permite que la base de datos sea utilizada de manera óptima.

El proceso de normalización consiste en verificar el cumplimiento de ciertas reglas que aseguran la eliminación de redundancias e inconsistencias. Esto se logra mediante la aplicación de ciertos procedimientos y en ocasiones se traduce a la separación de los datos en diferentes relaciones. Las relaciones resultantes deben cumplir ciertas características:

- Se debe conservar la información:
  1. Conservación de los atributos.
  2. Conservación de las tuplas, evitando la aparición de tuplas que no estaban en las relaciones originales.
- Se deben conservar las dependencias.

Este proceso se lleva a cabo aplicando una serie de reglas llamadas “formas normales”. Estas reglas permiten crear bases de datos libres de redundancias e inconsistencias, en la cual se aplica generalmente hasta la tercera forma normal; por lo tanto, se puede observar en la Tabla 4.1, que las columnas de las diferentes entidades son atómicas y cumplen con la eliminación de la redundancia.

Entidad	Relación
Persona	(ci, nombre, apellido, fecha_nacimiento, genero, teléfono, celular, dirección, created_at, updated_at, remember_token, <u>id_persona</u> )
Usuario	(id, email, password, remember_token, rol_id, valor_id, persona_id)
Paciente	( <u>id_paciente</u> , fecha_ingreso, grupo_sanguineo, nivel_educacional, lee_escribe, zona_residencia, convive, situacion_laboral, lugar_nacimiento, ocupacion, estado_civil, persona_id)
Rol	( <u>id_rol</u> , rol_name, nivel_acceso, código)

**Tabla 4.1. Base de datos relacional**

#### 4.1.3.3.2 Definición física de la Base de Datos.

Tabla Persona:

En esta tabla se almacenan los datos personales que son comunes a los diferentes tipos de persona (usuario, paciente).

Nombre	Tipo	Descripción
ci	character varying(20)	Numero de cedula
nombre	character varying(50)	Nombres (s)
apellido	character varying(50)	Apellido (s)
fecha nacimiento	date	Fecha de nacimiento
genero	character varying(10)	Género: Masculino o Femenino
teléfono	character varying(16)	Número de Teléfono
celular	character varying(16)	Número de Teléfono
dirección	character varying(200)	Dirección de lugar donde vive

**Tabla 4.2. Definición física de la tabla Persona**

Tabla usuario:

Se registran los datos específicos que caracterizan al tipo de usuario bien sea administrador, profesor tutor, profesor jefe de clínica, y estudiante (de 2do año, 3er año o 4to año)

Nombre	Tipo	Descripción
email	character varying(50)	Email del usuario
password	character varying(300)	Contraseña de acceso
rol_id	integer	Rol al que pertenece el usuario
valor_id	integer	Estatus del usuario: Activo o Inactivo
persona_id	integer	Identificador de la tabla persona

**Tabla 4.3. Definición física de la tabla Usuario**

Tabla Paciente:

Se registran los datos específicos para al tipo de persona paciente.

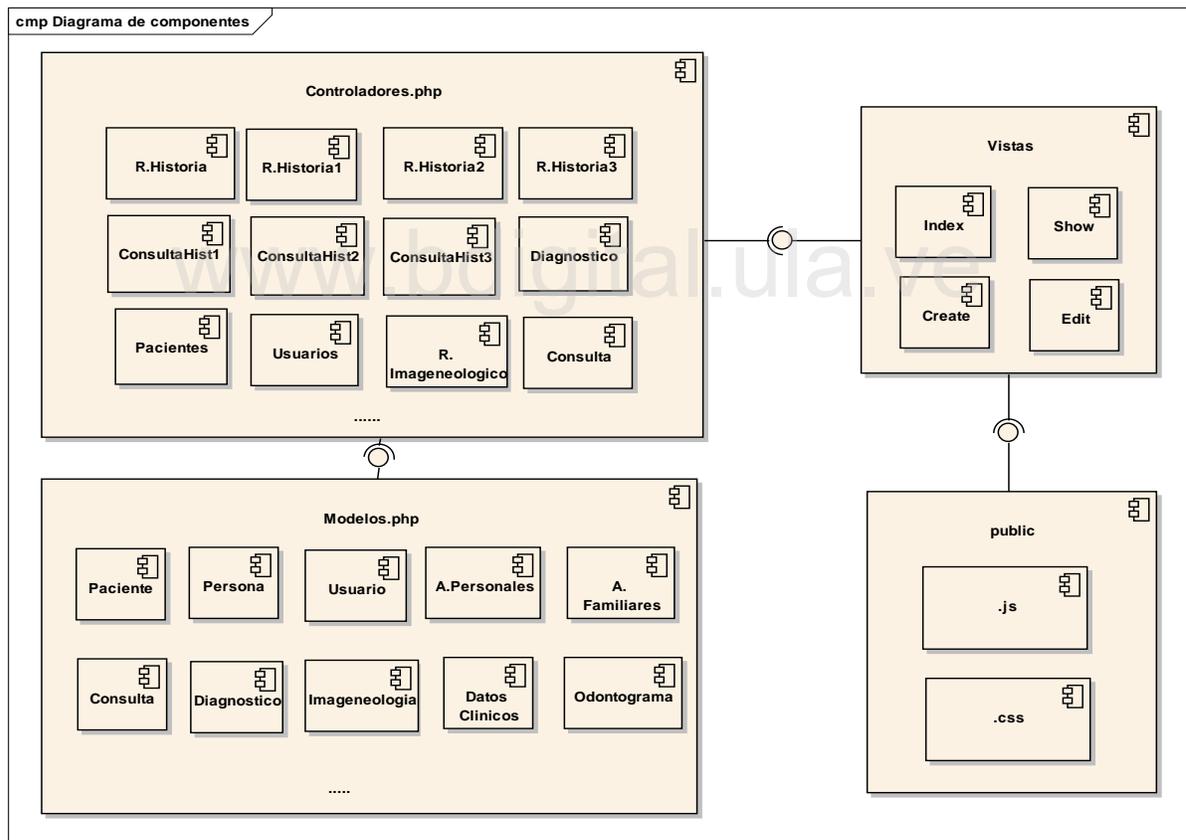
Nombre	Tipo	Descripción
fecha_ingreso	date	Fecha de ingreso del paciente
nro_historia	integer	Numero de historia del paciente
grupo_sanguíneo	character varying(15)	Grupo sanguíneo del paciente: A+, O+, entre otros.
nivel_educacional	integer	Nivel educacional del paciente (primaria completa, secundaria complete, universitario completa)
lee_escribe	integer	Si lee o escribe el paciente
zona_residencia	integer	zona de residencia urbana, periurbana, rural
convive	integer	Con quien convive el paciente (solo, con hijos, esposa e hijos)
situación_laboral	integer	Situación laboral del paciente (desempleado, estudiante, empleado)
lugar_nacimiento	character varying(50)	Lugar de Nacimiento del paciente
ocupación	character varying(50)	Ocupación del paciente
estado_civil	character varying(15)	Estado civil del paciente (soltero, casado, viudo, divorciado)
familiar_paciente_id	integer	Referencia a la tabla familiar_paciente donde se encuentra la información asociada al familiar más cercano del paciente
persona_id	integer	Referencia a la tabla persona

**Tabla 4.4. Definición física de la tabla Paciente**

Las tablas restantes serán mostradas en el Anexo C.

#### 4.1.3.4 Vista de componentes

Para el desarrollo del sistema Web se hizo uso de los componentes de Laravel siendo un componente un elemento reusable e independiente que cualquier aplicación puede utilizar, suministrando servicios por medio de interfaces bien definidas. En la Figura 4.4, se puede observar el diagrama de componentes de SIHCO, reflejando solo los componentes principales de dicho sistema.



**Figura 4.4. Diagrama de Componentes del sistema web**

### 4.1.3.5 Vista de despliegue

En la Figura 4.5 se observa el diagrama de despliegue del sistema web, en el que se observa la distribución física de los componentes mencionados en la sección 4.2.2, obteniendo una arquitectura de tres capas, donde el cliente representa la capa de presentación, Este a su vez se comunica con el servidor principal que contiene dos servidores virtuales, uno representando la capa de negocio (Servidor Web Postgres), que se comunica con la capa de datos (Servidor Base Datos).

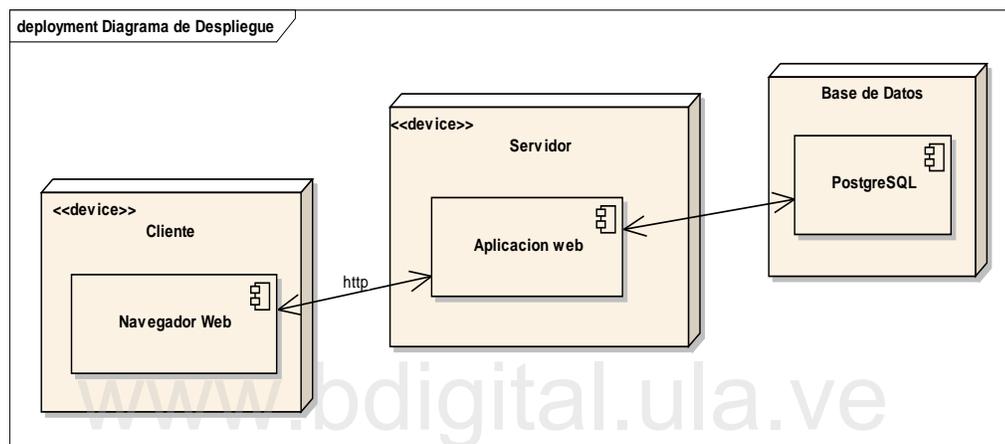


Figura 4.5. Diagrama de Despliegue del sistema web

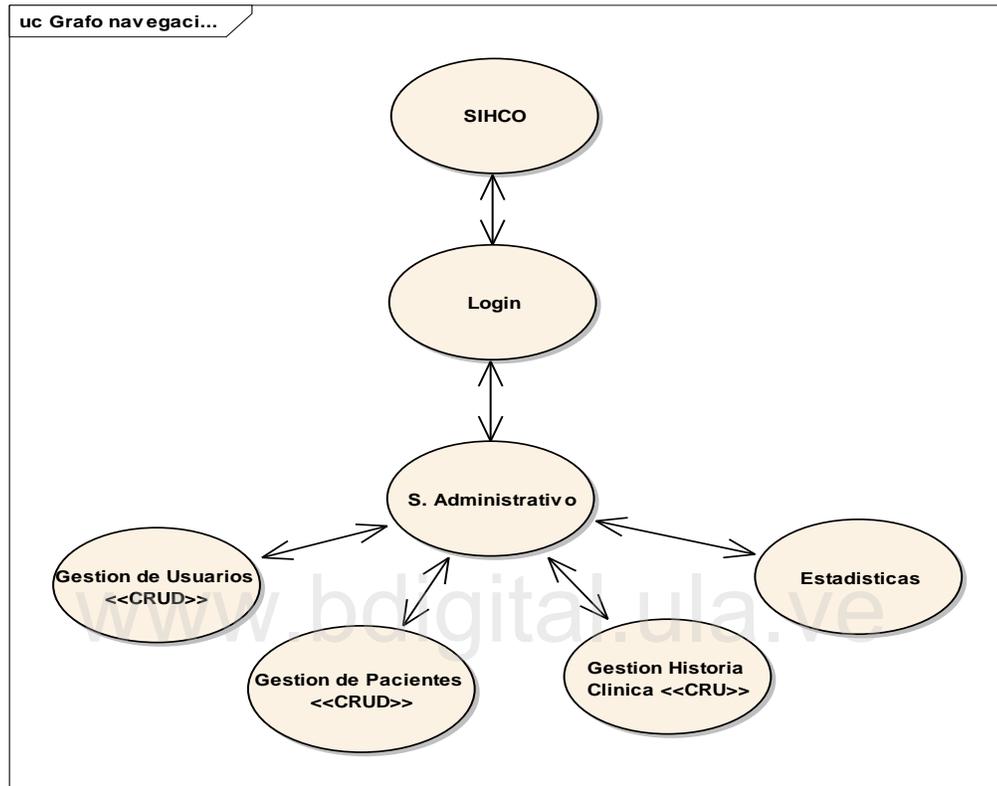
## 4.2 Diseño de la interfaz de usuario.

En esta sección definiremos la estructura de navegación, y el estilo de las páginas Web. Es importante señalar que las paginas fueron diseñadas tratando de conservar el mayor parecido posible con el sistema manual de historias clínicas actual de la Clínica Integral del Adulto, de tal manera que los usuarios se familiaricen de forma rápida y fácil con el sistema.

### 4.2.1 Grafo de navegación.

El sistema se compone por una serie de ventanas, que permiten al usuario relacionarse con la aplicación por medio de: ingreso al sistema, consulta, modificación, eliminación, con los que se le permite al

usuario obtener la información que se encuentra almacenada en la base de datos. La Figura 4.3, muestra el flujo de ventanas del sistema.



**Figura 4.3. Grafo de Navegación.**

- Ventana principal: Esta es la primera vista que acceden los usuarios al ingresar con la URL que da el acceso al sistema.
- Ventana de acceso: Esta ventana es donde el sistema le solicita al usuario el email y la contraseña para ingresar al mismo.
- Página de módulos: Muestra los distintos módulos a los cuales tiene acceso el usuario dependiendo del permiso del mismo.
- Página de submódulos: En esta ventana se refleja los procesos que se pueden llevar a cabo dentro del módulo solicitado por el usuario.

- **Página Resultado:** Esta ventana permite visualizar el resultado de la consulta que haya solicitado el usuario.
- **Alertas Ventanas emergentes:** Muestran información de alerta al usuario, en caso de fallas de algún procedimiento realizado.

#### **4.2.1.1 Estilo Visual de las Páginas.**

El sistema hace uso de un estilo fresco y renovado utilizando logos y colores asociados a la Universidad de Los Andes y en la Facultad de Odontología, con el fin de mantener un estándar a nivel organizacional. La interfaz del sistema está diseñada para que el usuario acceda a la información de forma rápida y sencilla. A continuación, se describen las partes que componen el estilo de las páginas.

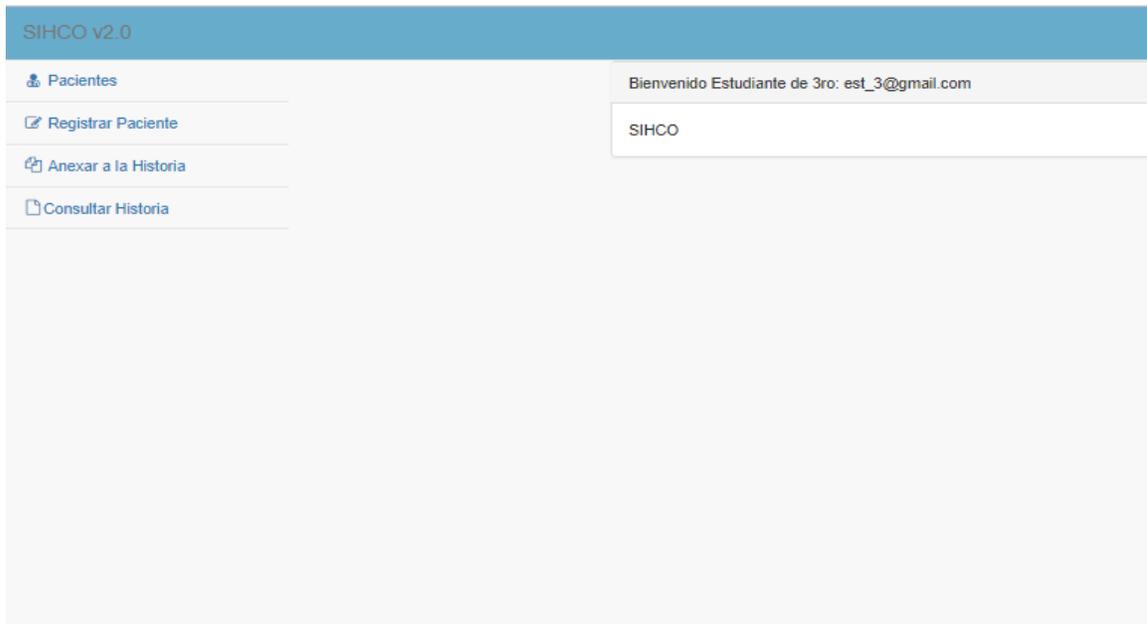
- **Barra de Módulos:** Muestra enlaces hacia los módulos disponibles dependiendo del tipo de usuario que acceda.
- **Botones:** Son los que le permiten al usuario realizar la acción deseada sobre el módulo o sub módulo que solicite.
- **Inicio:** Muestra una ventana de bienvenida al usuario de dicha sesión.
- **Web:** Retorna al usuario a la página principal de la aplicación manteniéndose en la sesión.
- **Ingresar usuario:** Muestra una ventana donde se ingresa la información del usuario, indicando el tipo de rol, y el profesor responsable.
- **Consultar usuario:** Muestra una ventana donde se despliega la información de los usuarios registrados.
- **Ingresar paciente:** Muestra una ventana donde se ingresa la información del paciente, nombres, apellidos, cedula de identidad, dirección, entre otros.
- **Consultar Historia:** Muestra una ventana donde el usuario selecciona al paciente que desea consultar la historia.
- **Anexar Historia:** Muestra una ventana donde el usuario selecciona al paciente que desea agregar la historia.
- **Reportes:** Muestra una ventana donde se podrán visualizar la información de: usuarios registrados, pacientes registrados e información relacionada a la historia clínica.

- Salir: Al dar clic sobre dicho botón hará que el usuario abandone la sesión. 4.3 Diagrama de Clases.

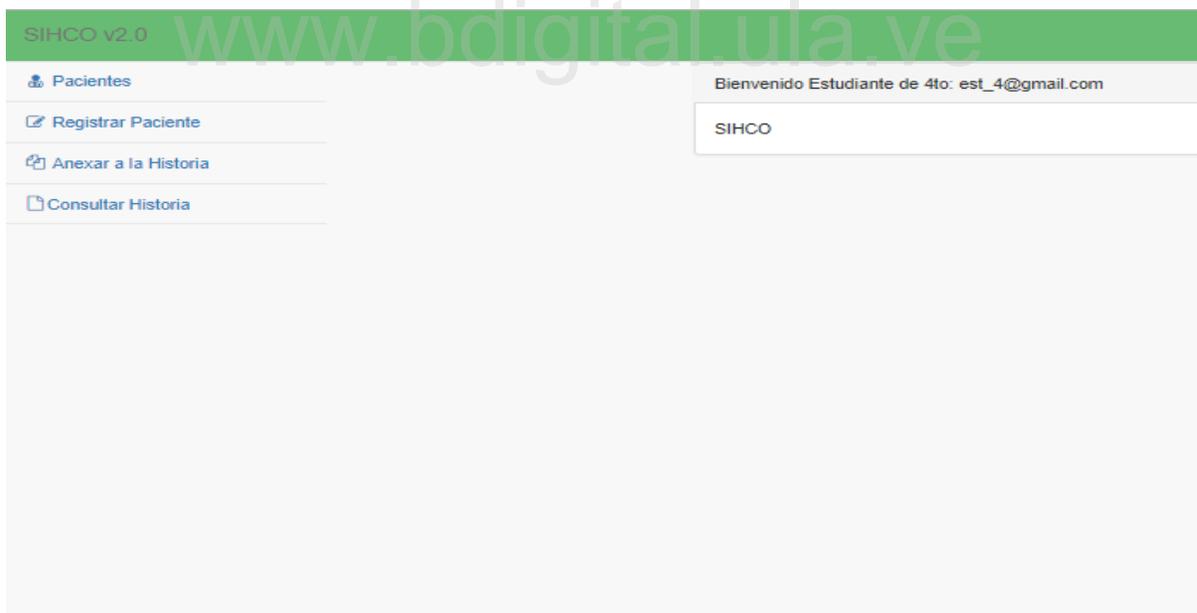
Para la diferenciación de los distintos roles que accederán al sistema se realizó la identificación por colores en la barra superior, en las Figuras 4.4, 4.5, y 4.6 se visualiza el menú de acceso una vez son validados los usuarios por el sistema en este caso los estudiantes de segundo año, tercer año y cuarto año.



**Figura 4.4 Menú de acceso para el estudiante de 2do año**



**Figura 4.5 Menú de acceso para el estudiante de 3er año**



**Figura 4.6 Menú de acceso para el estudiante de 4to año**

## Capítulo 5

# Implementación y Prueba del Sistema

Este capítulo comprende la implementación del diseño del sistema web, a través el ensamblaje de las tres capas (Capa de presentación, Capa Lógica de Negocio, y Capa de Datos) descritas en los capítulos anteriores, tomando como base los componentes desarrollados en el capítulo 4, con la finalidad de cumplir con los requerimientos solicitados por los usuarios del sistema. Para la implementación de este trabajo se realizó una gestión de configuración que permite manejar los distintos cambios en el sistema, produciendo versiones estables a través del proceso iterativo que nos proporciona la metodología empleada, esta gestión de configuración aporta beneficios ya sea el de brindar apoyo a los métodos de desarrollo de software, mantener la integridad del producto, asegurar la completitud y correctitud de los elementos bajo configuración, entre otros.

### 5.1 Construcción del Sistema.

Una vez diseñado el sistema, se procede a realizar la integración de los componentes asociados a la interfaz Usuario/Sistema con la capa de presentación, la cual a su vez está constituida por el código del lado del Cliente y del lado del Servidor.

Para el código del lado del cliente, el Sistema Web de Historias Clínicas Odontológicas está formado por elementos visuales (HTML) y códigos que se deben ejecutar en el navegador a través de funciones desarrolladas en lenguajes de programación (JavaScript, PHP). Por otra parte, el código del lado del servidor incluye la lógica de programación de las páginas apoyado en el Framework Laravel que cuenta con distintas herramientas para la manipulación de las pantallas diseñadas en la aplicación.

En el desarrollo de la interfaz gráfica del sistema se usó una serie de elementos como son, botones de acción, cajas de texto, text-área, etiquetas, hipervínculos, entre otros. Los cuales permiten una interacción de fácil manejo para el usuario final, por otra parte, se adaptaron funciones para generar distintos métodos de búsqueda y ordenamiento de los datos presentados.

En la Figura 5.1, se observa la pantalla de acceso al sistema, la cual está formada por dos campos: Email, y contraseña., el cual facilita a la aplicación la autenticación del usuario, dicho formulario debe cumplir con ciertas validaciones: el correo electrónico debe ser único y estar almacenado en la base de datos del sistema y una contraseña de tipo alfanumérica con longitud mínima de 6 caracteres. Si la autenticación es exitosa el sistema redirige a la pantalla principal donde se podrá observar las distintas funcionalidades a través de un menú (ver Figura 5.2), de acuerdo al rol de usuario registrado, en caso contrario el sistema retorna la misma pantalla de acceso con un mensaje de error indicando el campo el cual no aprobó la autenticación.

Acceder

Direccion E-Mail

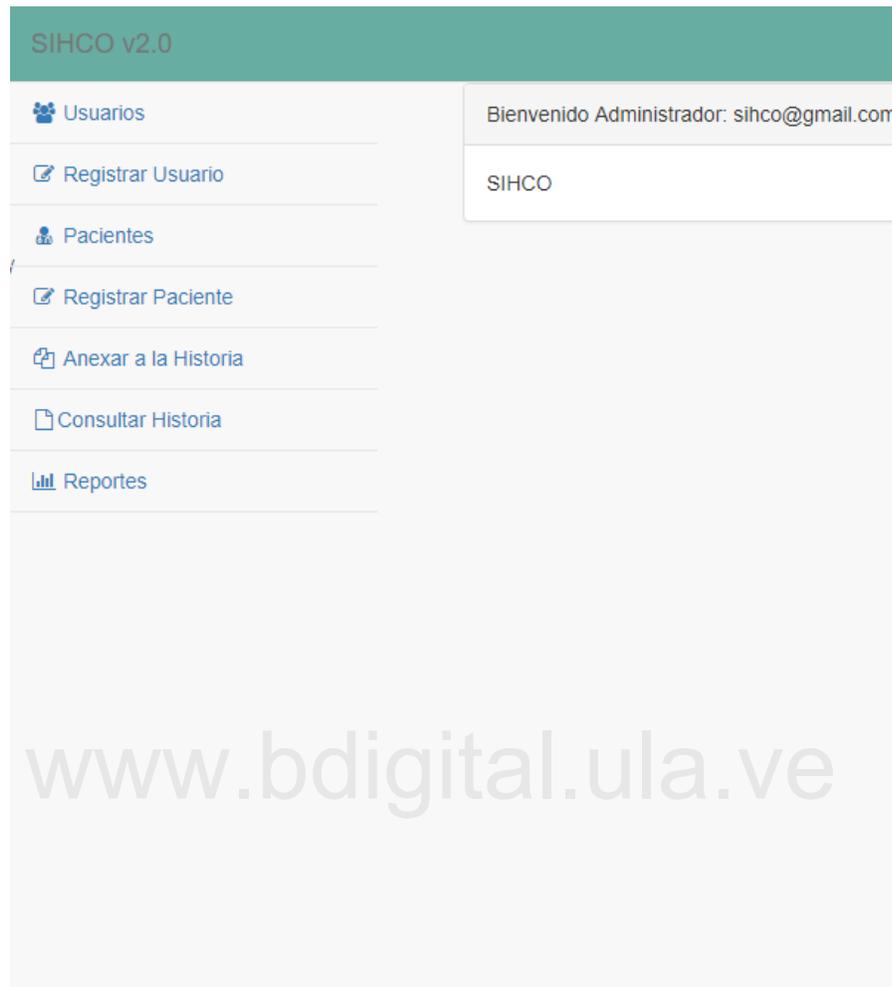
Contraseña

Recuerdame

[Entrar](#) [Olvido su Contraseña?](#)

www.bdigital.ula.ve

**Figura 5.1. Vista de acceso al sistema**



**Figura 5.2. Vista del menú para el rol jefe de clínica**

Como podemos apreciar en la Figura 5.2, este usuario cuenta con un grupo de botones ubicado en el “Menu” que permitirán su interacción con el sistema. A continuación, mostraremos como fluye el sistema al pulsar sobre “Anexar a la Historia”.

SIHCO v2.0

- Usuarios
  - Registrar Usuario
- Pacientes
  - Registrar Paciente
  - Anexar a la Historia
  - Consultar Historia
  - Reportes

Pacientes

Buscar Paciente  Ordenar Por Nombre

Hist. Nro	Nombre	Apellido	Sexo	C.I
13	Mario	perez	M	999
14	Luisa	Caceres	F	9166522
19	Estafania	Fernandez	F	222111
20	pedrito	perez	F	6666
16	Luis	Hidalgo	M	9662233

1

**Figura 5.3 Vista de Selección de Paciente para ingresar historia**

SIHCO v2.0

- Usuarios
  - Registrar Usuario
- Pacientes
  - Registrar Paciente
  - Anexar a la Historia
  - Consultar Historia
  - Reportes

Motivo de la Consulta

Nombre  C.I.  Fecha Consulta

Motivo de la Consulta

Enfermedad Actual

**Figura 5.4 Vista de Ingreso de Consulta**

Al momento de guardar la consulta, la información devuelta por el sistema nos mostrara una página con los datos del paciente de nuestra elección.

SIHCO v2.0

Usuarios

- Registrar Usuario
- Pacientes
- Registrar Paciente
- Anexar a la Historia
- Consultar Historia
- Reportes

I al VII | VIII al XIII | XIII al XXIV | Diagnostico/Pronostico | Tratamiento

III) Ant. Familiares  
 IV) Ant. Personales  
 V) Datos Clínicos Seleccionados  
 Res. de la Historia Medica  
 VI) Signos Vitales  
 VII) Historia Odontologica  
 Res. de la Historia Odontologica

Consultar Datos del Paciente

Apellido: perez  
 Sexo: M  
 Fecha de Nacimiento: 999  
 Ocupación: test  
 Telefono: test  
 Celular: test  
 Grupo Sanguineo: A  
 Telefono Familiar: 333  
 Direccion: test  
 Familiar Cercano: test

**Figura 5.5 Vista de Datos Personales**

En la figura anterior podemos observar la aparición de una barra de navegación, la cual nos permitirá acceder de forma rápida a cualquier parte de la Historia Clínica del paciente seleccionado, siguiendo con la presentación, iremos a la sección de insertar los Antecedentes familiares (ver Figura 5.6).

SIHCO v2.0

Usuarios

- Registrar Usuario
- Pacientes
- Registrar Paciente
- Anexar a la Historia
- Consultar Historia
- Reportes

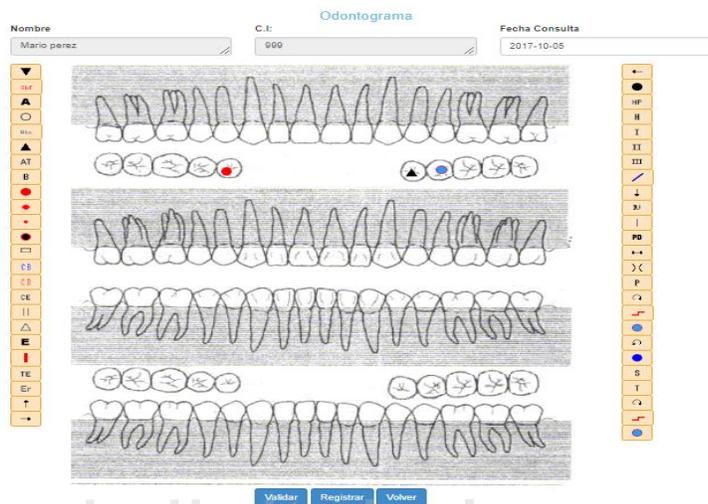
Antecedente Familiar

Nombre: Mario perez  
 C.I.: 999  
 Fecha Consulta: 2017-10-05

Enfermedad	Circulo	Acciones
<input checked="" type="checkbox"/> 1-Enfermedades Cardiovasculares		
Hipertension	Padre	Eliminar
<input type="checkbox"/> 2-Enfermedades Renales		
<input checked="" type="checkbox"/> 3-Neuropsicopatias	<input checked="" type="checkbox"/> 4-Enfer. Metabolicas y Endocrinas	<input checked="" type="checkbox"/> 5-Discrasias Sanguineas
<input type="checkbox"/> 6-Enfermedades Alergicas		
<input type="checkbox"/> 7-Fiebre Reumatica	<input type="checkbox"/> 8- Artritis Reumatoidea	
<input type="checkbox"/> 9-Cancer		
<input type="checkbox"/> 10-Enfermedades Infecciosas		
<input type="checkbox"/> 11-Enfermedades de Transmision sexual		

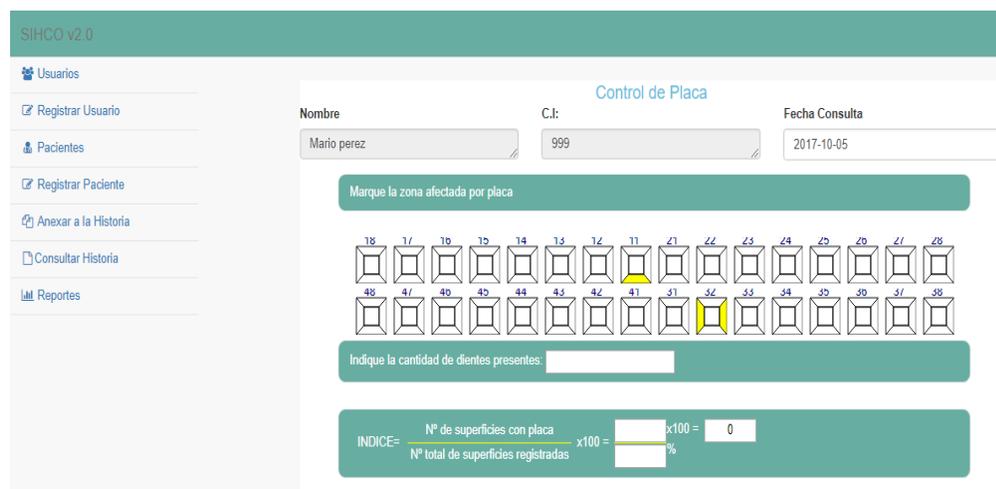
**Figura 5.6 Vista de Antecedente Familiar**

En la Figura 5.7 se muestra el ingreso del Odontograma del paciente, en el que podemos observar dos barras en sentido vertical en ambos lados, lo que permite seleccionar las distintas alteraciones que deben ser reflejadas en la zona del diente.



www.bdigital.ula.ve  
**Figura 5.7 Vista de Ingreso de Odontograma**

En la Figura 5.8 se muestra la página de ingreso de control de placa del paciente, en donde el usuario puede marcar la zona del diente afectado luego ingresar la cantidad de dientes presentes y al presionar calcular, se realiza el cálculo de índice de placa utilizando la ecuación de Nro. de dientes con placa entre Nro. de superficies presentes por 100.



**Figura 5.8 Vista de Ingreso de Control de Placa**

- [Registrar Usuario](#)
- [Pacientes](#)
- [Registrar Paciente](#)
- [Anexar a la Historia](#)
- [Consultar Historia](#)
- [Ver citas](#)
- [Citas](#)
- [Reportes](#)

### Imagenología

Nombre:  C.I.:  Fecha Consulta:

Ordenado Por:

Motivo del Examen:

Numero y tipo de Imagen:

Panorámica  Periapicales  Interproximales

Otros

Agregar Imágenes

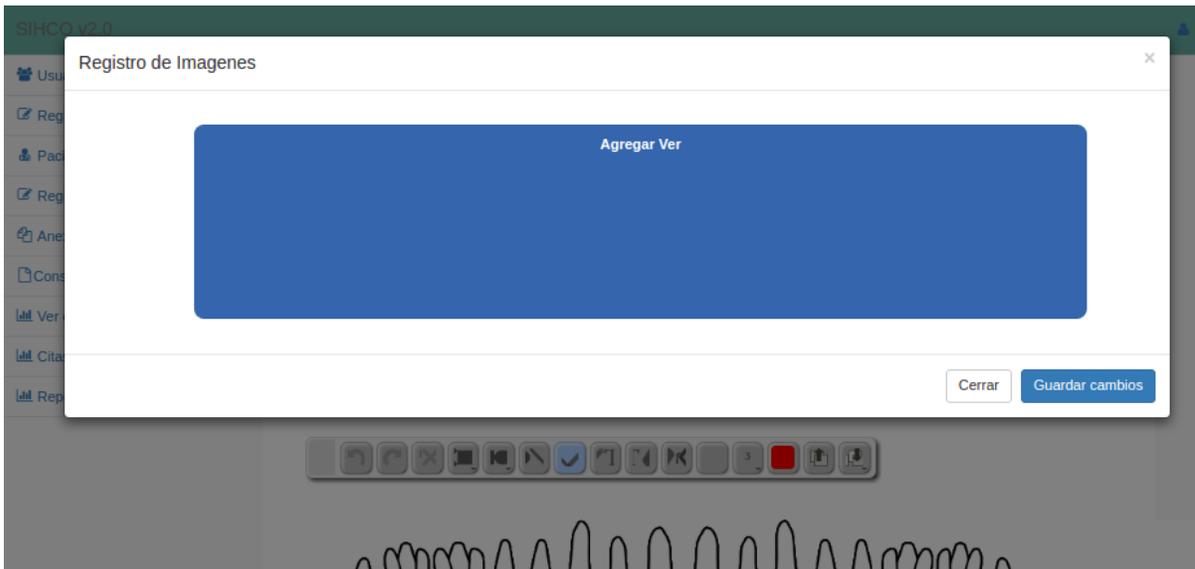
**Figura 5.9 Vista de Ingreso de Imagenología**

En la Figura 5.9 se puede observar la sección del ingreso de informes radiográficos donde el estudiante cuenta con el panel de dibujo para indicar que piezas faltan entre otras cosas, para luego hacer el posterior informe radiográfico.



**Figura 5.10 Vista de Ingreso de Imágenes radiográficas**

En la Figura 5.10 se observa la sección de ingreso de imágenes radiográficas indicadas por las zonas, periapicales e interproximales.



**Figura 5.11 Vista de Ingreso de Imágenes radiográficas (Panorámicas)**

En la Figura 5.11 se observa la sección de ingreso de imágenes radiográficas panorámicas.

- Usuarios
- Registrar Usuario
- Pacientes
- Registrar Paciente
- Anexar a la Historia
- Consultar Historia
- Ver citas
- Citas
- Reportes

### Examen Muscular

**Nombre**

**C.I:**

**Fecha Consulta**

**A. Palpación de los músculos extraorales:** (use una presión de 1 kg. Marque el código en el sitio correspondiente utilizando la siguiente escala para medir el dolor: 0 = Ausencia de dolor / presión solamente; 1 = Dolor leve; 2 = Dolor moderado; 3 = Dolor severo)

1 Temporal posterior (detrás de la sien)	Der. Izq.	<input type="text"/> <input type="text"/>	
2 Temporal medio (centro de la sien)		<input type="text"/> <input type="text"/>	
3 Temporal anterior (delante de la sien)		<input type="text"/> <input type="text"/>	
4 Origen del masetero (debajo del arco zigomático)		<input type="text"/> <input type="text"/>	
5 Cuerpo del masetero (en la mejilla)		<input type="text"/> <input type="text"/>	Temporal/Digastrico posterior
6 Inserción del masetero (angulo de la mandibula)		<input type="text"/> <input type="text"/>	
7 Región mandibular posterior (estilohioido/digastrico post.)		<input type="text"/> <input type="text"/>	
8 Región submandib. (ptergoideo medio/Suprahioido /Digastrico anterior)		<input type="text"/> <input type="text"/>	
9 Esternocleidomastoideo (a lo largo de todo el músculo)		<input type="text"/> <input type="text"/>	
10 Musculatura posterior del cuello y cervical		<input type="text"/> <input type="text"/>	Masetero/Digastrico anterior

Figura 5.12 Vista de Ingreso del Examen Muscular

10 Musculatura posterior del cuello y cervical   Masetero/Digastrico anterior

Esternocleidomastoideo

Musculatura posterior del cuello

**B. Palpación de los músculos intraorales:** (use una presión de 0.5 kg. Marque el código en el sitio correspondiente utilizando la siguiente escala para medir el dolor: 0 = Ausencia de dolor / presión solamente; 1 = Dolor leve; 2 = Dolor moderado; 3 = Dolor severo)

1 Área del pterigoideo lateral (por encima de los molares superiores)	Der. Izq.	<input type="text"/> <input type="text"/>	
2 Tendón del temporal (tendón)		<input type="text"/> <input type="text"/>	

Pterigoideo lateral

Figura 5.13 Vista de Ingreso del Examen Muscular

En las Figuras 5.12 y 5.13 se puede observar parcialmente el formulario de preguntas para realizar el examen muscular a un paciente.

SIHCO v2.0

Examen de la Oclusion

Nombre: Mario perez      C.I.: 999      Fecha Consulta:

1. Fremilus  
Especifique

2. Oclusion

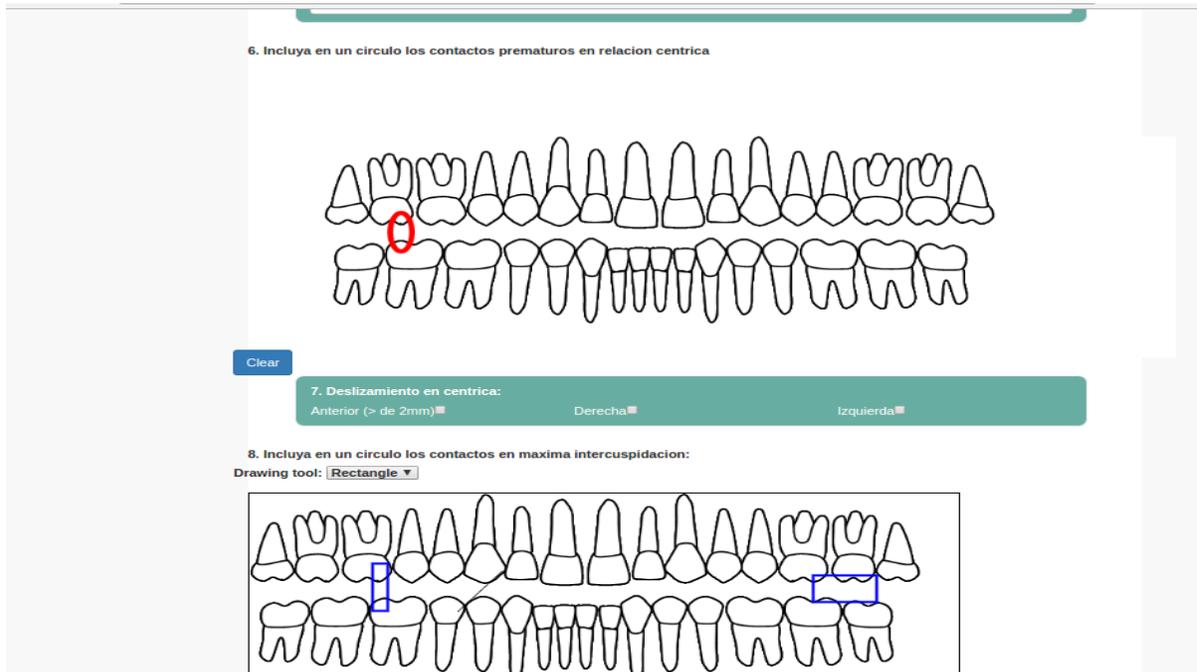
1- Normoclusion      2- Clase II      3- Clase III  
4- Abierta:      5- A tope:      6- Cruzada:  
Anterior      Anterior      Anterior  
Posterior: Derecha      Izquierda      Posterior: Derecha      Izquierda

3. Sobremordida

Horizontal      Vertical  
mm      mm

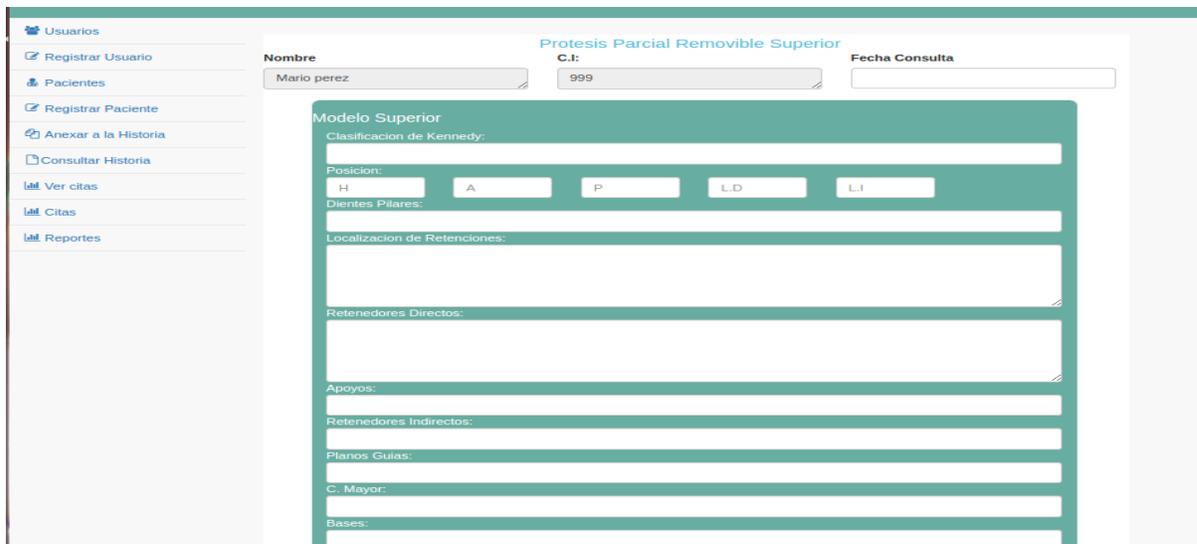
4. Desviacion de la linea media dental:  
Ninguna      Hacia la Derecha      Hacia la Izquierda

Figura 5.14 Vista de Ingreso del Examen de la Oclusión



**Figura 5.15 Vista de Ingreso del Examen de la Oclusión**

En la Figura 5.14 y 5.15 se puede observar parcialmente el cuestionario de preguntas a realizar a un paciente para el examen de la oclusión, además se cuenta con secciones de dibujo que permite marcar y sombrear los contactos en las diferentes zonas de las piezas dentarias.



**Figura 5.16 Vista de Ingreso prótesis parcial removible superior**

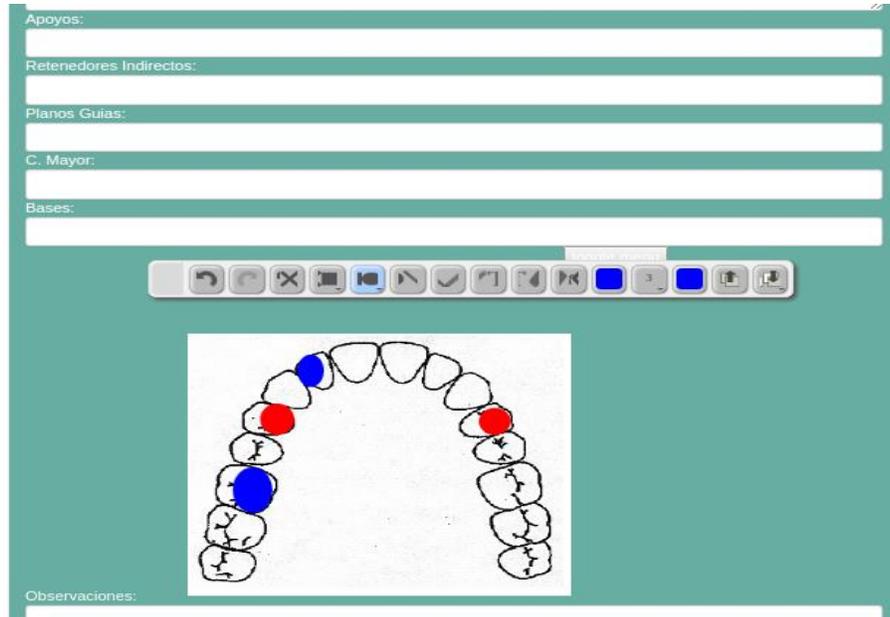


Figura 5.17 Vista de Ingreso prótesis parcial removible superior

En las Figuras 5.16 y 5.17 se observa el formulario de registro para la prótesis parcial removible superior, se cuenta además con un panel de dibujo donde los usuarios pueden realizar el bosquejo de las prótesis.

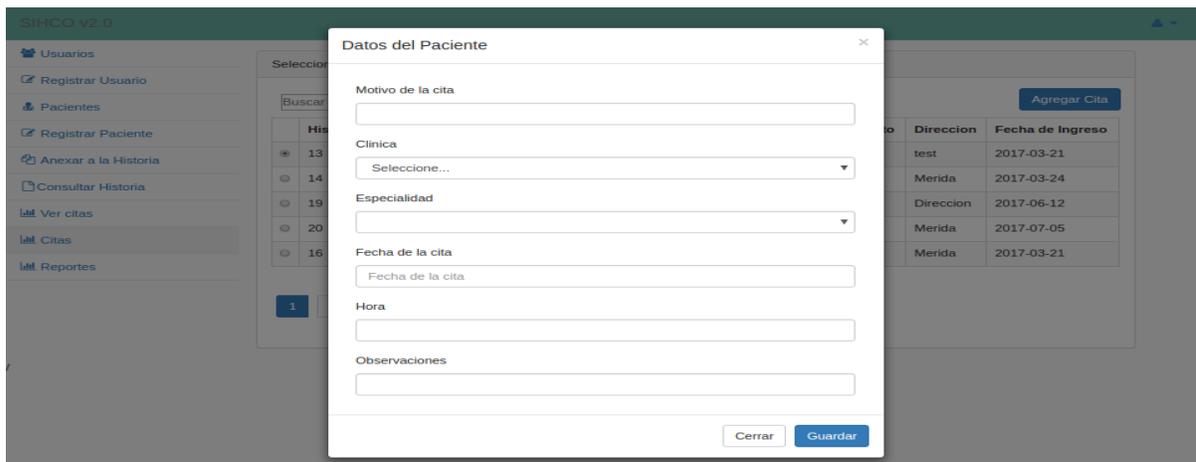


Figura 5.18 Vista de Ingreso de una cita

**Figura 5.19 Vista de Consultas de las citas**

En las Figuras 5.18 y 5.19 se observa los módulos de gestión de citas, para ingresar y consultar dichas citas por clínica y además por especialidad, en la sección de consulta se pueden visualizar las citas por días permitiendo editar dicha cita y eliminarla.

## 5.2 Gestión de Configuración.

Durante el proceso de construcción de un software, los cambios son inevitables. Los cambios provocan confusión e incertidumbre, sobre todo cuando no se han analizado o pronosticado correctamente. Es importante considerar ciertas modificaciones que pueden ocurrirle al software dentro de todo el proceso de ingeniería. La Gestión de la Configuración es el conjunto de procesos destinados a asegurar la validez de todo producto obtenido durante cualquiera de las etapas del desarrollo de un Sistema de Información (S.I.), a través del estricto control de los cambios realizados sobre los mismos y de la disponibilidad constante de una versión estable de cada elemento para toda persona involucrada en el

citado desarrollo. Estos dos elementos (control de cambios y control de versiones de todos los elementos del S.I.) facilitan también el mantenimiento de los sistemas al proporcionar una imagen detallada del sistema en cada etapa del desarrollo. La gestión de la configuración se realiza durante todas las fases del desarrollo de un sistema de información, incluyendo el mantenimiento y control de cambios, una vez realizada la puesta en producción. (EcuRed).

Para el presente trabajo se realizó el control de versiones a través de la herramienta Bitbucket la misma es un servicio de alojamiento basado en web, el cual nos brinda el apoyo y servicio para alojar nuestra aplicación, esta herramienta se basa en Git el sistema de control de versiones distribuido de código abierto (Álvarez, s.f). Este control de versiones fue de gran utilidad e importancia durante el desarrollo del sistema ya que permitía depurar las versiones estables e incluso poder regresar a cambios anteriores cuando una funcionalidad no estaba operando de manera correcta.

## 5.2 Pruebas del sistema.

www.bdigital.ula.ve

En todo proyecto es de gran importancia minimizar las fallas del sistema, por ello se debe probar adecuadamente tanto los programas como los procedimientos y base de datos. Para la detección de errores se tomaron en cuenta los tipos de prueba: caja negra y caja blanca.

### 5.2.1 Prueba de caja negra.

En este tipo de prueba se hace énfasis en los requisitos funcionales, a muy alto nivel, para diseñar las pruebas sin necesidad de conocer como están desarrollas las funciones internas de la aplicación. Las pruebas se aplican sobre el sistema empleando un conjunto de datos de entrada y observando la salida, para garantizar que la función se está desempeñando correctamente.

Las pruebas se realizaron en el servidor de desarrollo para evitar que los posibles errores no afecten el ambiente de trabajo a los estudiantes y profesores de la Facultad de Odontología. En la Tabla 5.1 se describen algunas de las pruebas realizadas.

Modulo	Parámetro de entrada	Salida esperada	Resultado
Acceso al	Clave incorrecta	Vuelve a solicitar los	Satisfactorio

Sistema		datos para ingresar al sistema	
Guardar nuevo usuario	No ingresar algún dato obligatorio y pulsar ingresar	Mensaje de error solicitando los datos Obligatorios	Satisfactorio
Guardar nuevo paciente	No ingresar algún dato obligatorio y pulsar ingresar	Mensaje de error solicitando los datos Obligatorios	Satisfactorio
Guardar Consulta	No ingresar algún dato obligatorio y pulsar ingresar	Mensaje de error solicitando los datos Obligatorio	Satisfactorio
Guardar Antecedente personal	No ingresar algún dato obligatorio y pulsar ingresar	Mensaje de error solicitando los datos Obligatorios	Satisfactorio
Guardar Consulta	No ingresar la fecha de la consulta	Mensaje de error indicando que debe ingresar la fecha	Satisfactorio

**Tabla 5.1: Pruebas Caja Negra**

En la Figura 5.20 se puede visualizar la prueba realizada al momento de insertar una consulta sin el campo de fecha y enfermedad actual, en el cual el sistema le indica al usuario los campos obligatorios para proceder con el registro.

SHCO v2.0

Usuarios

- Registrar Usuario
- Pacientes
- Registrar Paciente
- Anexar a la Historia
- Consultar Historia
- Reportes

Motivo de la Consulta

Nombre: Mario perez

C.I.: 999

Fecha Consulta:  Introduzca una fecha

Motivo de la Consulta: Dolor de Muela

Enfermedad Actual:  Introduzca una enfermedad actual de la consulta

Registrar

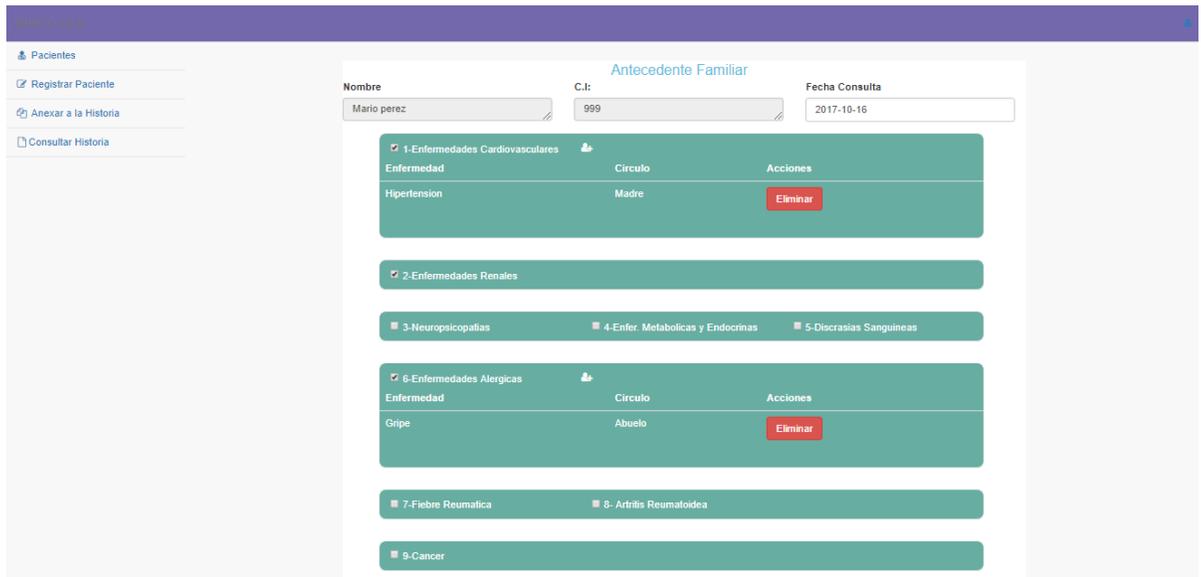
**Figura 5.20. Guardar Consulta sin campos obligatorios**

### 5.2.2 Prueba de caja blanca.

Son pruebas de bajo nivel, para el desarrollo de este tipo de prueba se conoce el código y lo que el sistema debe hacer, para ello se proporciona un conjunto de entrada para obtener los valores de salida esperados. Se probó cada uno de los módulos para verificar el correcto comportamiento:

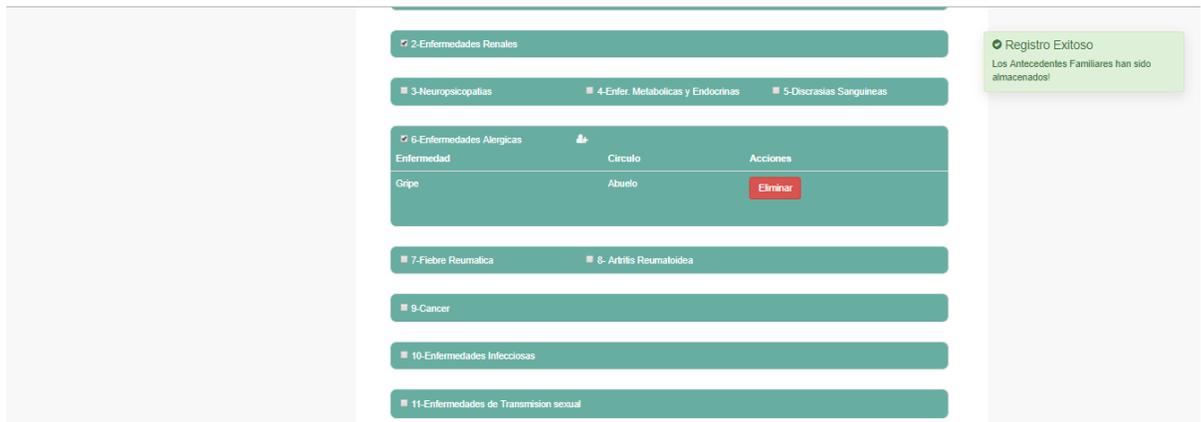
Módulo de Anexar Historia: Se realizó el ingreso de una historia (ficticia) de un paciente, comprobando que almacenara toda la información de manera correcta luego de pasar por las validaciones necesarias.

Módulo de Agregar Usuario: se realiza el ingreso de un nuevo usuario, comprobando que almacenara toda la información satisfactoria luego de pasar por las validaciones necesarias.



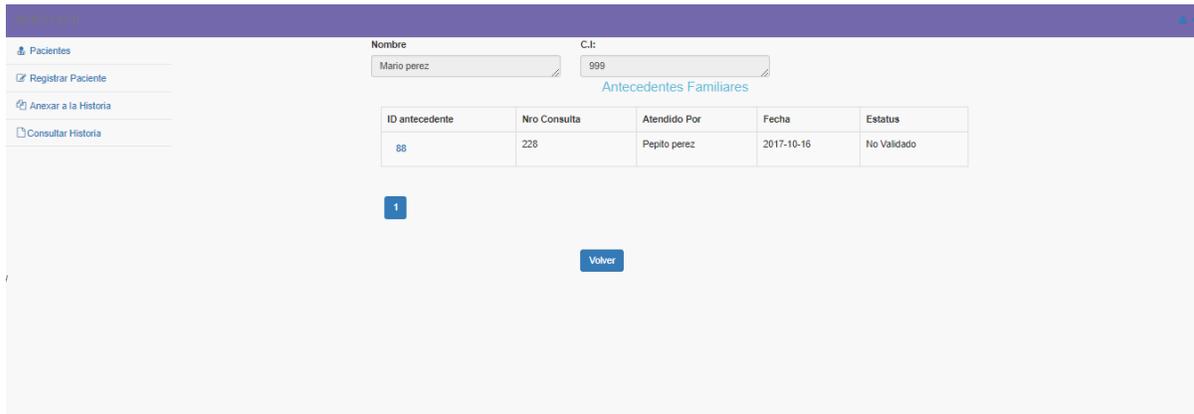
**Figura 5.21 Insertar Historia: Antecedente Familiar por el rol estudiante de 2do año**

En la figura anterior se puede apreciar el cambio de color en la barra de navegación, esto se realizó siguiendo los requerimientos del sistema el cual se hacía necesario la diferenciación de los distintos roles al momento de ingresar al sistema.



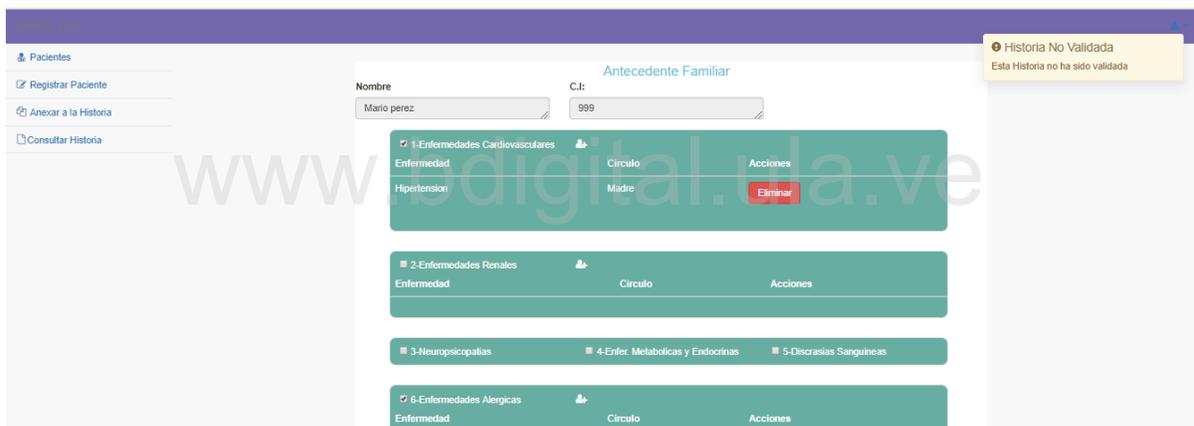
**Figura 5.22 Insertar Historia: Antecedente Familiar por el rol estudiante de 2do año**

En la Figura 5.22 se muestra el resultado obtenido luego de presionar el botón guardar. Para garantizar que dicho antecedente se haya almacenado de manera exitosa se ingresa en la opción de “Consultar Historia” y se selecciona al paciente el cual se realizó el ingreso de los datos.



**Figura 5.23 Consultar Historia: Antecedente Familiar por el rol estudiante de 2do año**

En la Figura 5.23 y 5.24 se puede apreciar que los datos almacenados corresponden a los realizados anteriormente.



**Figura 5.24 Consultar Historia: Antecedente Familiar por el rol estudiante de 2do año**

En la Figura 5.24 se puede apreciar además un mensaje de aviso de que la historia no se encuentra validada lo cual permite visualizar el estatus y poder apreciar el momento en el cual la historia esta validada por parte del profesor tutor a cargo.

Para el módulo de anexar historia la función utilizada para almacenar los datos de manera satisfactoria fue:

```
public function antecedentefamiliar(Request $req)
{
    $data = $req->all();
```

```

DB::beginTransaction();

try {

    $consulta = intval($req->input('consulta_id'));

    $data['ultimo_usuario'] = Auth::user()->id;
    $verificar = DB::table('antecedentes_familiares')
        ->where('paciente_id', $data['paciente_id'])
        ->where('consulta_id', $data['consulta_id'])
        ->where('fecha', $data['fecha'])
        ->count();
    if ($verificar > 0) {
        $id = DB::table('antecedentes_familiares')
            ->where('paciente_id', $data['paciente_id'])
            ->where('consulta_id', $data['consulta_id'])
            ->where('fecha', $data['fecha'])
            ->pluck('antecedentes_familiares.id_antecedente_familiar');

        $data['id_antecedente_familiar'] = $id[0];

        DB::table('antecedentes_familiares')
            ->where('paciente_id', $data['paciente_id'])
            ->where('consulta_id', $data['consulta_id'])
            ->where('fecha', $data['fecha'])
            ->delete();
    }

    $consulta2 = AntecedentesFamiliares::create($data);

} catch (Exception $ex) {

```

```

DB::rollback();

echo $ex;

die();
}

DB::commit();

}

```

Para el módulo de Agregar usuario se realizaron las pruebas similares a la sección anterior.

The screenshot shows a web interface for 'SHCO v2.0' with a sidebar menu on the left containing options like 'Registrar Usuario', 'Registrar Paciente', 'Anexar a la Historia', 'Consultar Historia', 'Ver citas', 'Citas', and 'Reportes'. The main content area is titled 'Registrar Usuario' and contains a form with the following fields and values:

Nombre	pedro
Apellido	perez
Cedula	1111
Fecha de Nacimiento	2017-10-16
Sexo	Masculino
Direccion	Merida
Telefono	0274-8574424
Celular	04126669866
Tipo de usuario	Administrador
Valor	Activo
Dirección de correo	prueba@gmail.com
Contraseña	*****

A 'Registrar' button is located at the bottom of the form.

**Figura 5.25 Agregar Usuario por parte del rol Jefe de Clínica**

En la Figura 5.25 se muestra el formulario para el registro de un nuevo usuario. Al momento de darle guarda el sistema informa del registro exitoso de dicho usuario (ver Figura 5.26).

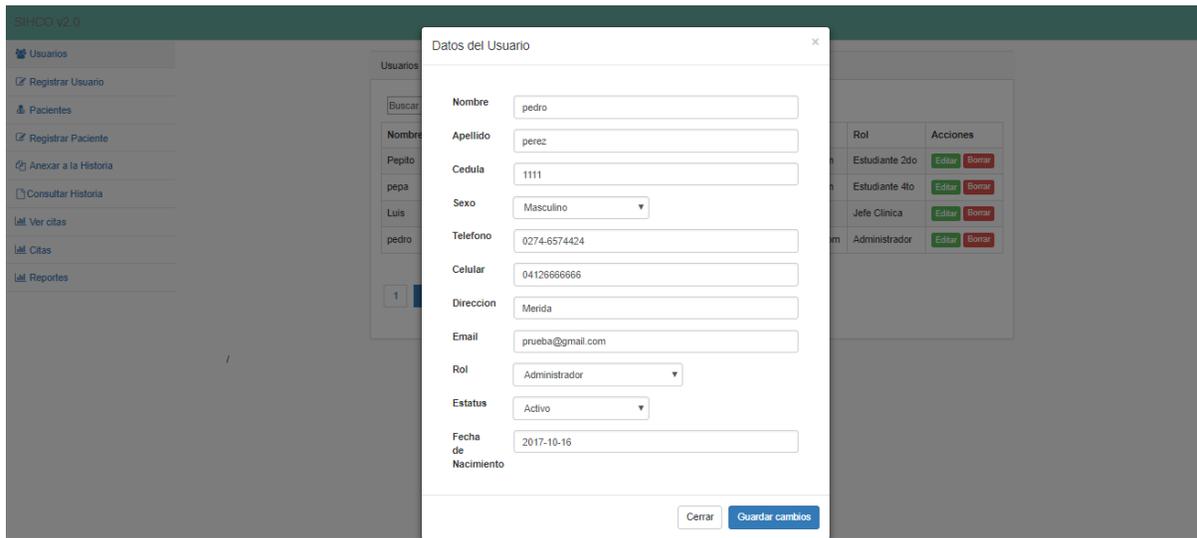
**Figura 5.26 Agregar Usuario por parte del rol Jefe de Clínica**

Para verificar que el usuario se haya registrado de manera exitosa en la base de datos, se realizó la consulta de todos los usuarios registrados lo cual arrojó los resultados esperados.

Nombre	Apellido	Sexo	C.I	Celular	Fecha de Nacimiento	Email	Rol	Acciones
Pepito	perez	M	22222222	0414253366	2017-09-12	est_2@gmail.com	Estudiante 2do	<a href="#">Editar</a> <a href="#">Borrar</a>
pepa	plg	M	66552222	04247255555	2017-09-12	est_4@gmail.com	Estudiante 4to	<a href="#">Editar</a> <a href="#">Borrar</a>
Luis	Perez	M	44332222	04245664444	2017-10-06	luis@gmail.com	Jefe Clinica	<a href="#">Editar</a> <a href="#">Borrar</a>
pedro	perez	M	1111	04126666666	2017-10-16	prueba@gmail.com	Administrador	<a href="#">Editar</a> <a href="#">Borrar</a>

**Figura 5.27 Consultar Usuario por parte del rol Jefe de Clínica**

Como se puede apreciar en la Figura 5.27 el ultimo registro de los usuarios es el usuario registrado anteriormente, se realiza la consulta de dicho usuario para comprobar que toda la información suministrada sea la correcta (ver Figura 5.28)



**Figura 5.28 Consultar Usuario por parte del rol Jefe de Clínica**

Para el módulo de registrar usuario, se realizaron varios registros comprobando que los registrara de manera exitosa, la función utilizada fue:

```
public function register(Request $request)
```

```
{
```

```
    $flag = DB::table('usuarios')
```

```
        ->where('ci', $request->input('ci'))
```

```
        ->count();
```

```
    $flag2 = DB::table('persona')
```

```
        ->where('ci', $request->input('ci'))
```

```
        ->count();
```

```
    $flag3 = DB::table('usuarios')
```

```
        ->where('email', $request->input('email'))
```

```
        ->count();
```

```
    if ($flag) {
```

```
        return redirect()->back()->with('duplicated_user', 'El usuario ya existe!');
```

```

}

if ($flag3) {
    return redirect()->back()->with('user_registered', 'Error email duplicado');
}

if ($flag2) {

    $persona = DB::table('persona')
        ->where('ci', $request->input('ci'))
        ->pluck('persona.id_persona');
    $this->create($request->all(), $persona[0]);
    return redirect()->back()->with('user_registered', 'El usuario se creó satisfactoriamente');
}

$this->createPerson($request->all());
$persona = DB::table('persona')
    ->where('ci', $request->input('ci'))
    ->pluck('persona.id_persona');

$this->create($request->all(), $persona[0]);

return redirect()->back()->with('user_registered', 'El usuario se creó
satisfactoriamente');
}

```

### 5.2.3 Prueba de Navegación.

En tipo de prueba se verifica que todas las secciones del sistema web sean funcionales y permitan al usuario moverse a través del mismo.

Modulo	Condición de ejecución	Resultado
Acceso al Sistema: Autenticar el usuario	Ingresa correo, contraseña y pulsa Ingresar	Funciona Correctamente
Acceso a: Consultar usuarios	Selecciona la pestaña "Usuarios"	Funciona Correctamente
Acceso a: Guardar usuario	Selecciona la pestaña "Agregar Usuario"	Funciona Correctamente
Acceso a: Anexar Historia	Selecciona la pestaña "Anexar Historia"	Funciona Correctamente
Acceso a: Consultar Historia	Selecciona la pestaña "Consulta Historia"	Funciona Correctamente
Guardar Consulta	Ingresa fecha de la consulta, motivo de la consulta y enfermedad actual, pulsa guardar	Funciona Correctamente
Guarda sección de la Historia	Ingresa los datos respectivos a la historia, ingresa la fecha, pulsa guardar	Funciona Correctamente

**Tabla 5.2: Pruebas de Navegación**

## Capítulo 6

### Conclusiones y Recomendaciones

#### 6.1 Conclusiones.

Basado en la metodología White Watch se desarrolló el sistema web de historias clínicas odontológicas para la Facultad de Odontología de la Universidad de Los Andes, el cual posee múltiples ventajas con respecto al sistema desarrollado anteriormente, como son:

- Desarrollo del nuevo sistema web bajo herramientas de software libre.
- Implementación y mejoramiento de la interfaz gráfica para el anexo y consulta de las historias clínicas de los pacientes.
- Implementación de nuevos módulos de la historia clínica del paciente requeridos por el personal encargado de la atención clínica del paciente: estudiantes y profesores.

El sistema web desarrollado permite llevar el control del tratamiento integral del paciente, a través de módulos que facilitan el registro de elementos gráficos (radiografías, panorámicas, modelos dentales, entre otros), de esta forma se logra almacenar de manera digital las distintas imágenes que contiene la historia de un paciente, lo que garantiza prolongar su existencia evitando así la pérdida de tiempo y dinero. Además, la aplicación cuenta con nuevas funciones que facilitan la implementación de los tests realizados a los pacientes (Nicotina, Farsgestrom), que aportan datos valiosos para visualización y análisis de datos estadísticos. Es importante resaltar que el sistema facilita al usuario el registro, búsqueda de información y el control de citas de los pacientes atendidos en la Clínica Integral del Adulto.

Con respecto al objetivo que indicaba: Importar y validar los datos del sistema anterior al nuevo, en vista de que la nueva historia cambió, con relación a la historia del año 2003 implementada en

dicho sistema, se tomó la decisión de no realizar la importación puesto que quedaban muchos campos de las tablas vacíos por ser campos nuevos incorporados en la nueva historia, se planteó dejar las historias anteriores en papel, y almacenar en el nuevo sistema las nuevas historias de los pacientes.

Para el control de versiones se utilizó la herramienta Bitbucket el cual utiliza Git para todo el manejo del desarrollo de la aplicación.

A través de las pruebas caja negra y caja blanca, se verificó que el sistema web cumple con los requerimientos solicitados, por lo tanto, se concluye que la aplicación se comporta de manera esperada.

## **6.2 Recomendaciones.**

Agregar un módulo al sistema que permita a los pacientes no registrados acceder de manera remota al sistema web para realizar un pre registro y brindarles una visualización de las fechas y horarios de atención por parte de los estudiantes y profesores de la Facultad de Odontología para las distintas Clínicas (I, II, III).

Agregar un módulo que permita consultar los costos y tratamientos llevados a cabo en las clínicas.

Adaptar el sistema web actual a las distintas Facultades vinculadas en el área de la salud, de la Universidad de los Andes, con el fin de aportar una herramienta automatizada para el control de los pacientes.

Establecer mecanismos de enseñanza y ayuda por parte de los administradores del sistema, hacia los estudiantes y profesores en cuanto a la inserción de los datos de la historia, para que exista un correcto uso de la aplicación.

## Referencias Bibliográficas

- Villareal G. (2003), Sistema de Información Web para el Manejo de Historias Clínicas Odontológicas (Tesis de Pregrado). Universidad de Los Andes, Mérida Venezuela.
- Montilva, J. (2010). Desarrollo de aplicaciones empresariales, el método watch. Mérida, Venezuela.
- Duque, K. (2009). Software Para la Gestión de Control de Historias Clínicas Odontológicas (Tesis de Pregrado). Universidad Rafael Urdaneta, Maracaibo, Venezuela.
- Hernández, A. L. (2014). ESTOMATOLOGIA: PREVENCIÓN BUCAL. [<http://analeticiahernandezflores.blogspot.com/2014/09/uso-del-odontograma.html>]
- Guillen, P. R. (2003). Análisis y Diseño Orientado a Objetos. <http://www.itl.laguna.edu.mx/academico/carreras/sistemas/Analisis%20y%20dise%F1o%20orientado%20a%20objetos/rumbaugh.pdf>.
- Álvarez, M. A. (2014). Qué es MVC. <http://www.desarrolloweb.com/articulos/que-es-mvc.html>.
- Pérez, J. (2008). Definición de sistema. <https://definicion.de/sistema/>
- Besembel, I. M. (1998). Base de Datos Relacionales.
- Álvarez, M. A. (2014). Framework. <https://desarrolloweb.com/wiki/framework.html>
- Márquez, M. (2011). Base de Datos.
- Pérez J & Gardey A. 2012. Definición de HTML. <https://definicion.de/html/>
- Kendall, K. E. (2005). Análisis y Diseño de sistemas. Sexta edición, Pearson Educación, México.
- Kustner, E. (1999). La Historia Clínica Odontológica. Editorial Masson S.A. Barcelona-España.

- Rumbaugh J, Jacobson I, Booch G. (2000). El lenguaje unificado de modelado. Pearson Educación, Madrid.
- EcuRed. Base de Datos. [https://www.ecured.cu/Bases\\_de\\_datos](https://www.ecured.cu/Bases_de_datos).
- Moreno, S. (2008). Gestión del área de trabajo en el gabinete bucodental. Málaga-España.
- Decreto Nro. 3.390, publicado en Gaceta Oficial Nro.38.095. Caracas, Venezuela. 28 de diciembre de 2004.

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

**Anexos**

## Anexo A.

### Casos de Usos del Sistemas Web

En la Figura C.1 se muestra el caso de uso CU1.Ingresar al Sistema, en él podemos observar a los distintos actores accediendo al sistema, para lo cual deberán identificarse y de ser validada la información ingresada concerniente al email del usuario y la clave, el sistema automáticamente desplegará el menú respectivo a su jerarquía.

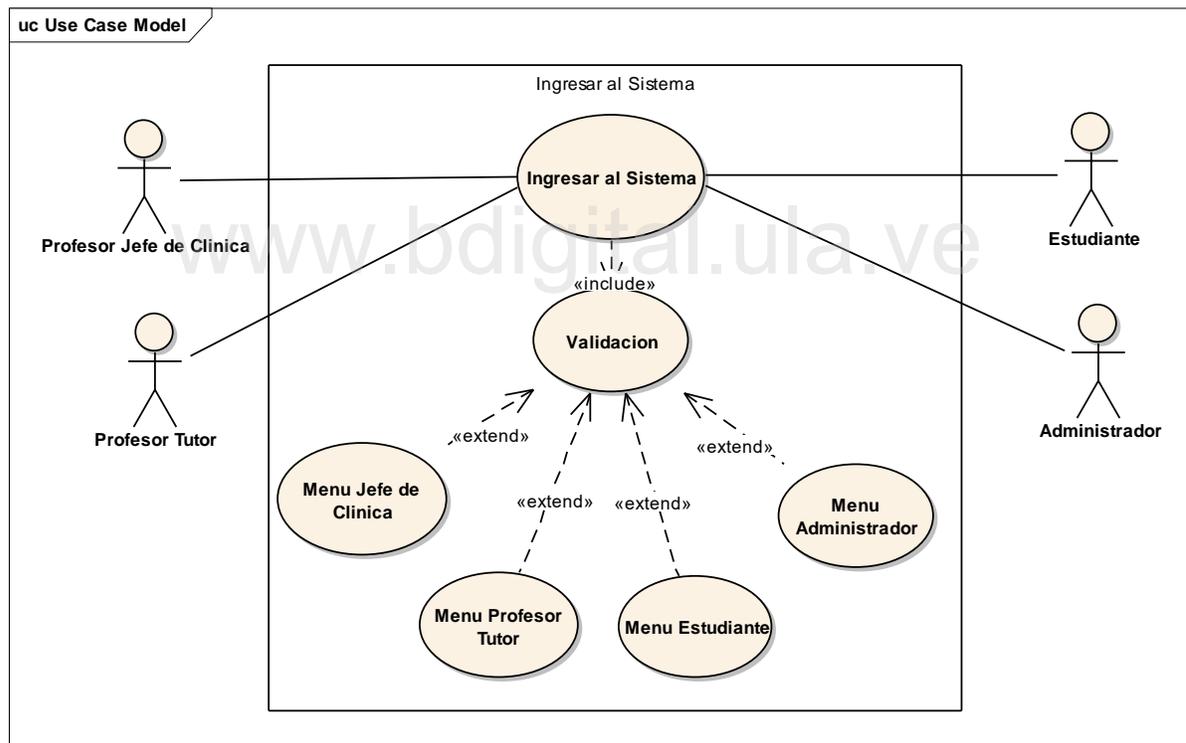


Figura A.1 Caso de uso ingresar al Sistema

En la Figura C.2 se especifica el caso de uso anexar historia para el rol del estudiante, se puede observar la jerarquía que presenta dicho rol, especializándose en los roles, estudiantes de segundo año, estudiante de tercer año y estudiante de cuarto año, lo cual indica que el estudiante de segundo año solo puede acceder a la sección 1 de la historia del paciente, el estudiante de tercer año accede a la sección 1 y además accede a la sección 2, mientras que el estudiante de cuarto año puede acceder a toda la historia de dicho paciente.

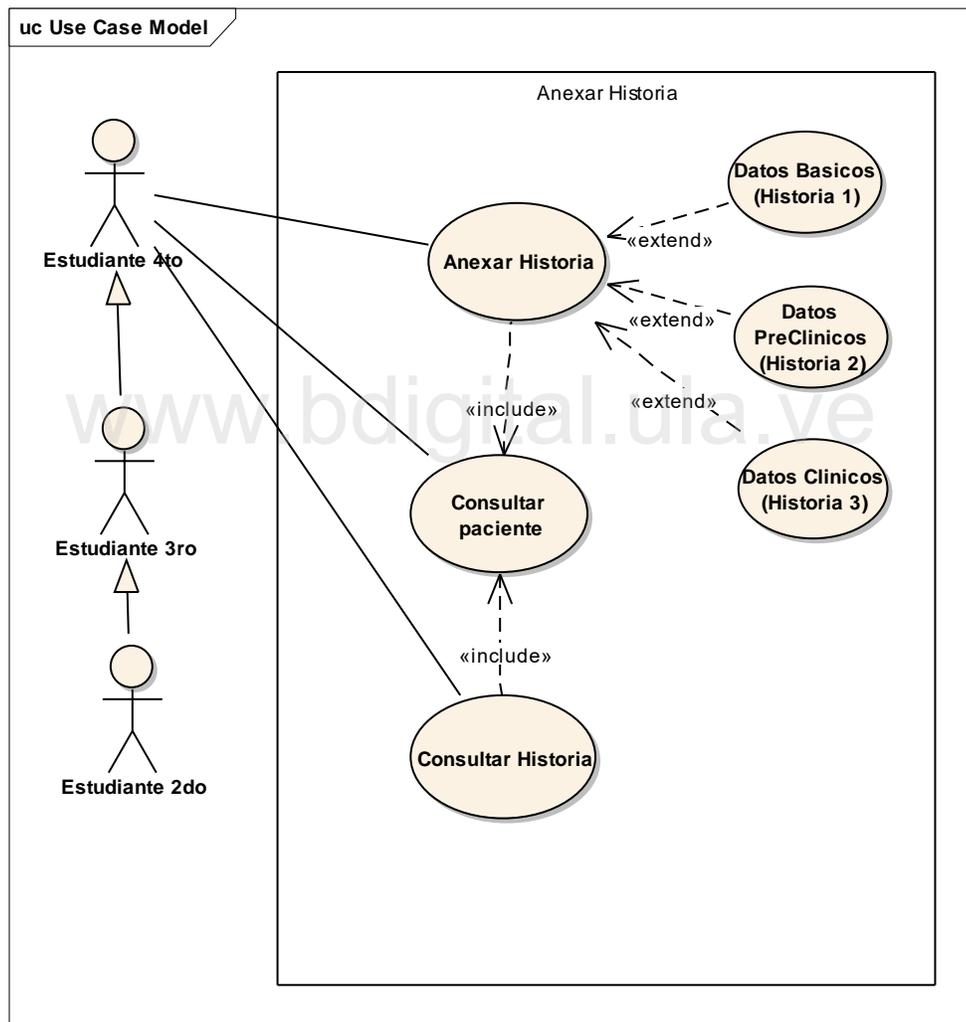


Figura A.2. Caso de uso anexar historia para el caso del rol estudiante

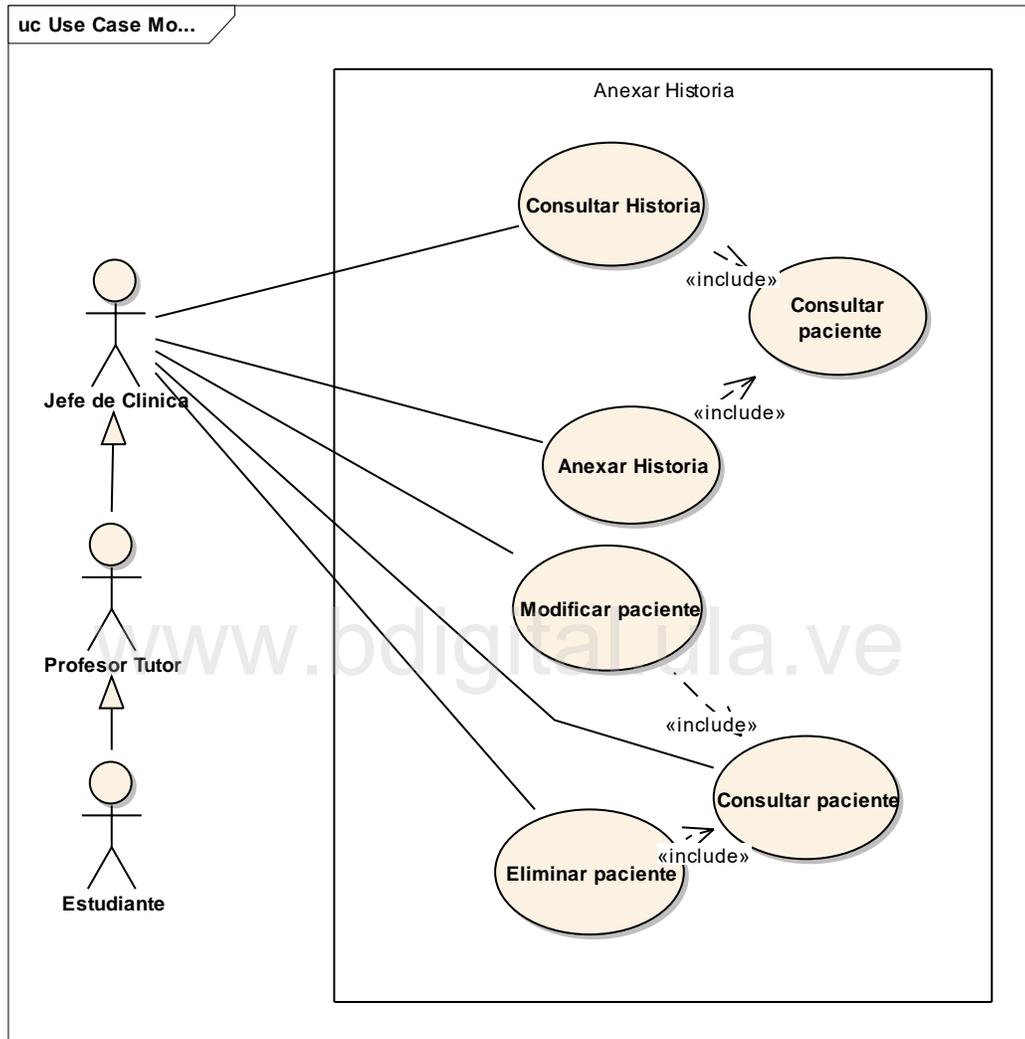


Figura A.3. Caso de uso generalizado para CRUD de pacientes y CRU de Historias

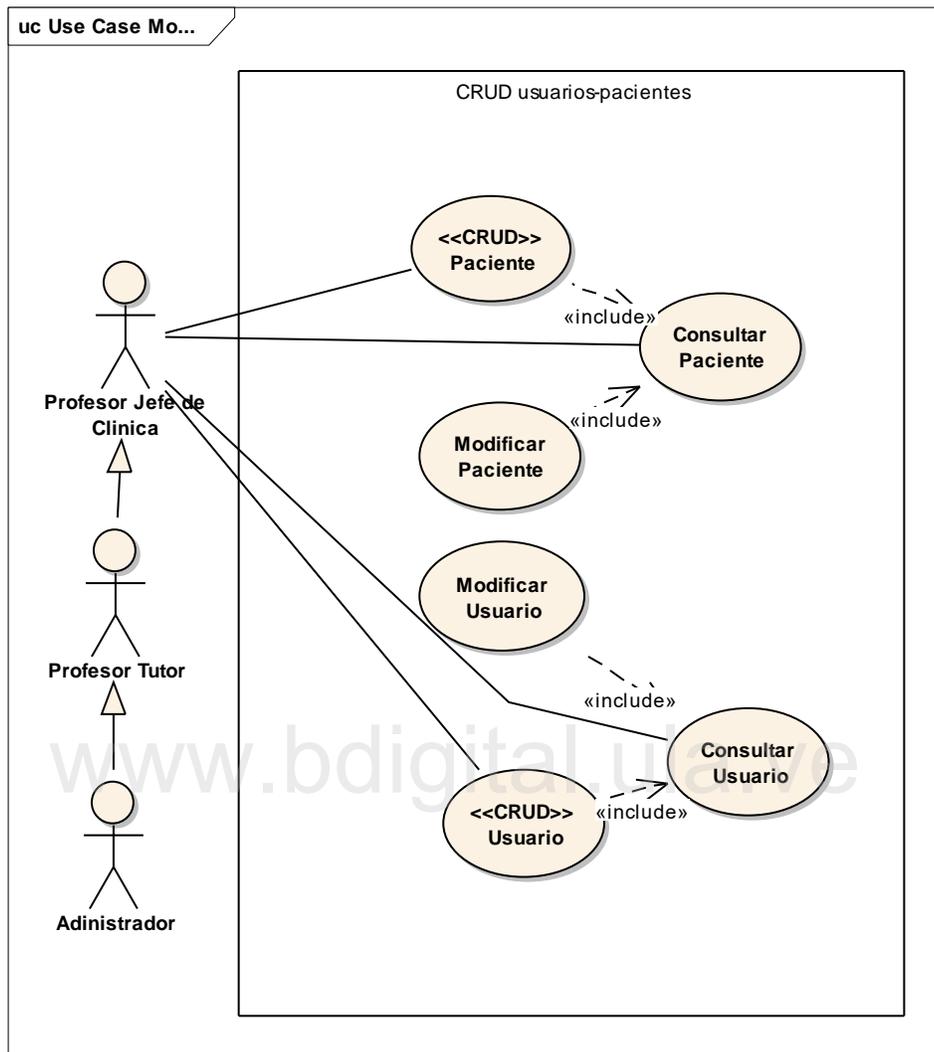


Figura A.4. Detalles de los casos de uso CRUD usuario y CRUD paciente

## Anexo B.

### Diagramas de Secuencia para los casos de Uso del Sistema Web

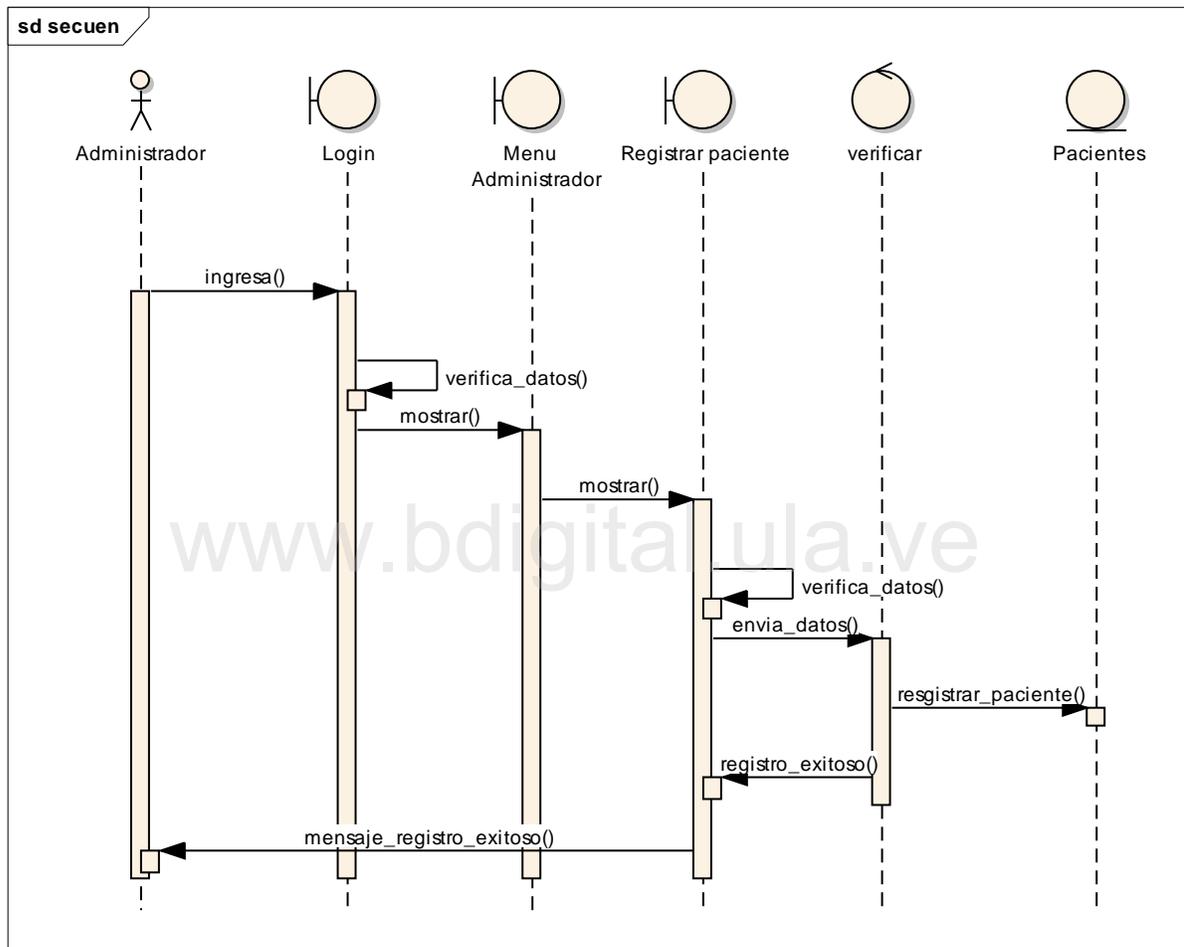


Figura B.1. Diagrama de secuencia para el CU2, ingresar paciente

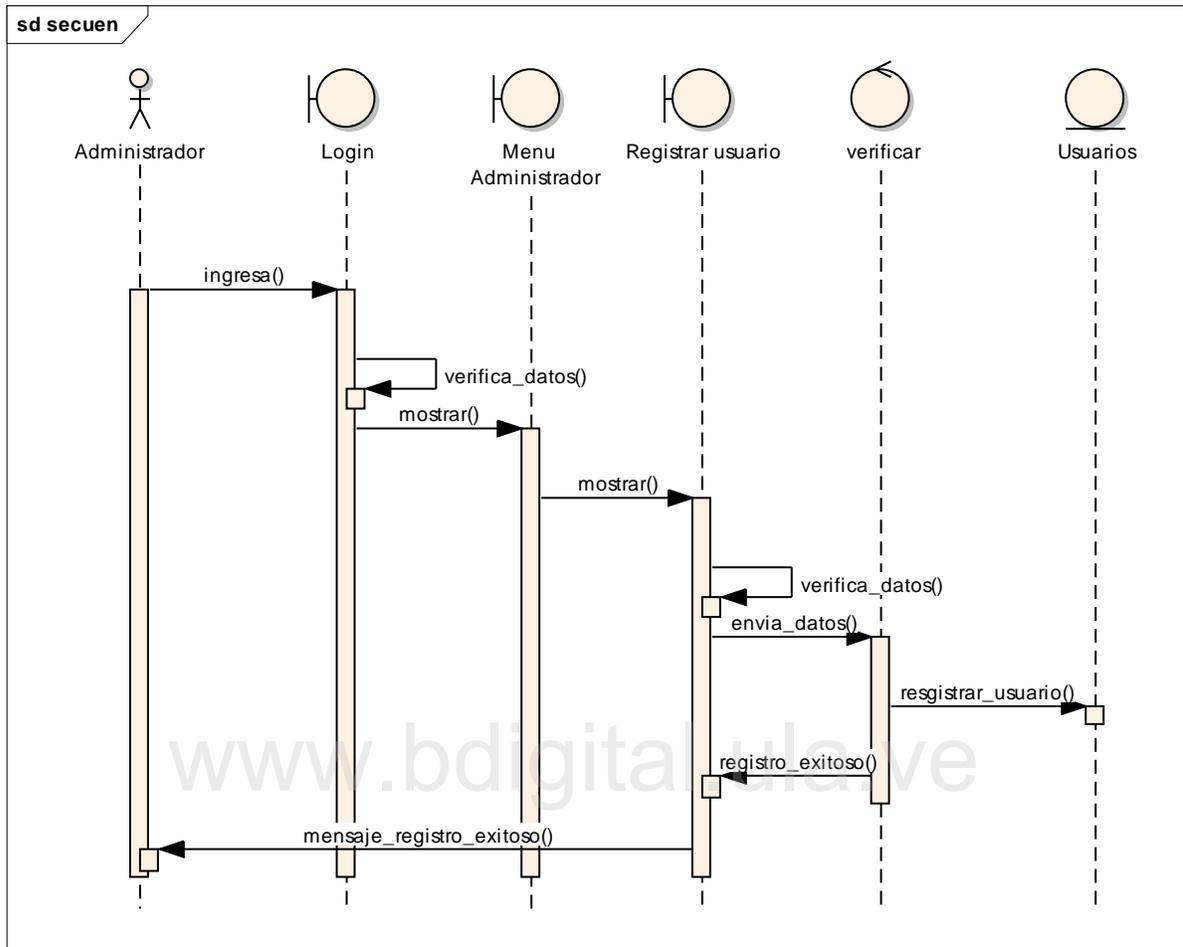


Figura B.2. Diagrama de secuencia para el caso de uso CU5, ingresar usuario.

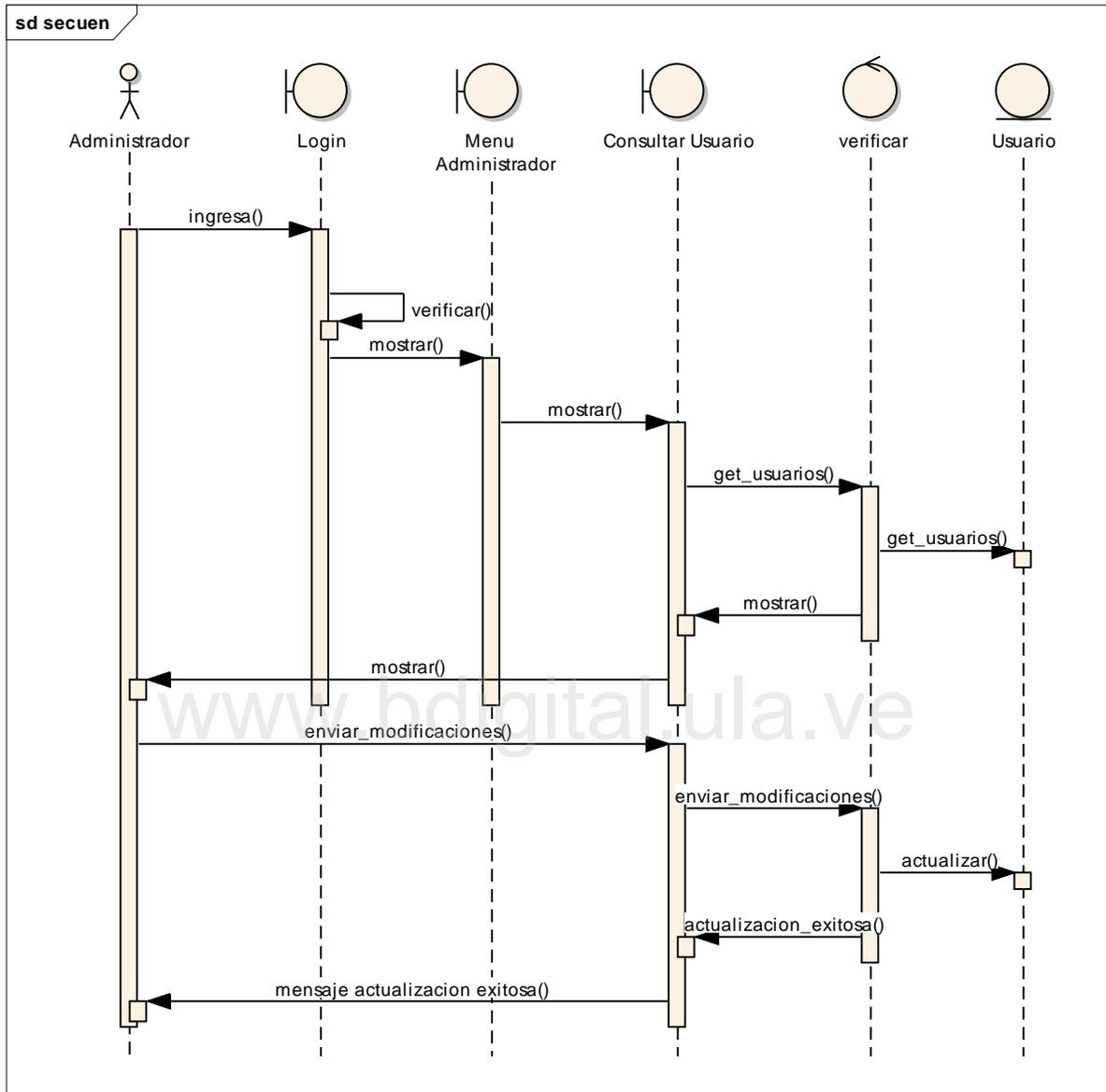


Figura B.3. Diagrama de secuencia para la consulta de usuario

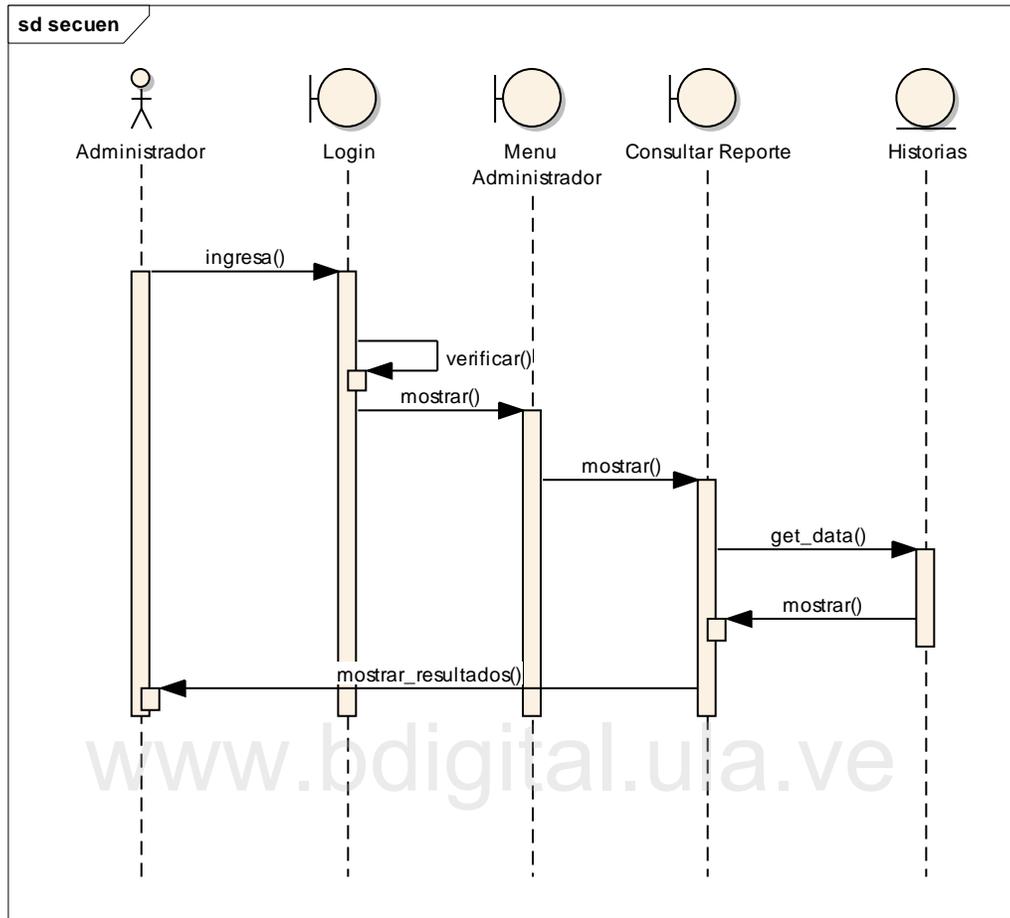


Figura B.4. Diagrama de Secuencia para el caso de uso, CU12, consultar estadísticas

## Anexo C.

### Diccionario de Datos

#### Tabla Historia

Esta tabla contiene información de las entidades de la historia clínica que han sido ingresadas o no y de estar llenas si han sido validadas. Se indica con 0 para no ingresado, 1 para ingresado, 2 ingresado y validado.

Nombre	Tipo	Descripción
id_historia	integer	identificador único
af	carácter varying (5)	Antecedentes familiares
ap	carácter varying (5)	Antecedentes personales
dc	carácter varying (5)	Datos clínicos
sv	carácter varying (5)	Signos Vitales
ho	carácter varying (5)	Historia Odontológica
rhc	carácter varying (5)	Resumen Historia Clínica
rho	carácter varying (5)	Resumen Historia Odontológica
ec	carácter varying (5)	Examen Clínico
Fp	carácter varying (5)	Evaluación periodontal
Odonto	carácter varying (5)	Odontograma
Placa	carácter varying (5)	Control de placa
Radio	carácter varying (5)	Imagenología
eo	carácter varying (5)	Examen Oclusión
Em	carácter varying (5)	Examen Muscular
md	carácter varying (5)	Modelos de diagnósticos
Ecom	carácter varying (5)	Examen complementario
Fcpf	carácter varying (5)	Coronas y puentes fijos

Dtl	carácter varying (5)	Dentaduras totales
Protesis	carácter varying (5)	Prótesis parcial removible
endodoncia	carácter varying (5)	Endodoncia
operatoria	carácter varying (5)	Operatoria
cirugía	carácter varying (5)	Cirugía
diapre	carácter varying (5)	Diagnóstico clínico
diaDef	carácter varying (5)	Diagnóstico definitivo
pronostico	carácter varying (5)	Pronostico
tratamiento	carácter varying (5)	Tratamiento

**Tabla C.1. Descripción de la tabla historia**

### Tabla Consulta

En esta tabla se almacena la información acerca de las causas por las que el paciente acude a la Clínica Integral.

Nombre	Tipo	Descripción
id	integer	Identificador único
nro_historia	integer	numero historia del paciente
motivo_consulta	character varying(200)	Motivo de la consulta
enfermedad_actual	character varying(200)	Enfermedad Actual
ultimo_usuario	integer	Referencia al estudiante que ingresa la consulta
profesor	integer	Referencia del profesor que realiza la validación
validar	character varying(15),	Contiene información acerca de la validez de la historia
fecha_validacion	character varying(15),	Contiene la fecha que se realizó la validación.

**Tabla C.2. Descripción de la tabla Consulta**

### Tabla Datos Clínicos

Esta tabla contiene las respuestas concernientes al interrogatorio del paciente con respecto a los datos relevantes acerca de las intervenciones quirúrgicas. Las respuestas son reflejadas en Si / No

Nombre	Tipo	Descripción
id_dato_clinico	integer	Identificador único
examinado_ultimo_ayo	integer	Examinado el último año
cambio_salud_ultimo_ayo	integer	Ha habido un cambio en su salud el último año
sangra_largo_tiempo	character varying(5)	Sangra por largo tiempo al cortarse
perdida_peso	character varying(5)	Pérdida de peso en los últimos meses
grave_enfermo	character varying(5)	Ha estado gravemente enfermo
hospitalizado	character varying(5)	Ha sido hospitalizado
marcapaso	character varying(5)	Posee marcapaso
tranfusión_sanguinea	character varying(5)	Le han realizado transfusión sanguínea
asma	character varying(5)	Sufre de asma
aspirina	character varying(5)	Alérgico a la aspirina
penicilina	character varying(5)	alérgico a la penicilina
yodo	character varying(5)	alérgico al yodo
sulfonamida	character varying(5)	alérgico a la sulfonamida
otros_medicamentos	character varying(5)	alérgico a otros medicamentos
trastorno_visual	character varying(5)	sufre de trastorno visual
sinusitis	character varying(5)	sufre de sinusitis
hemorragia_nasal	character varying(5)	sufre de hemorragia nasal
dolor_pecho	character varying(5)	siente dolor en el pecho cuando realiza esfuerzo moderado
tos	character varying(5)	tiene tos persistente
tos_sangre	character varying(5)	tose algunas veces con sangre
dificultad_tragar	character varying(5)	tiene dificultad para tragar
indigestión	character varying(5)	sufre de indigestión frecuente
vomito	character varying(5)	vomita con frecuencia

orines	character varying(5)	orina más de seis veces al día
sediento	character varying(5)	esta siempre sediento
desmayo	character varying(5)	tiene tendencia a desmayarse
moretones	character varying(5)	se le producen moretones
desorden_sangre	character varying(5)	tiene desorden en la sangre
cansancio	character varying(5)	se cansa fácilmente
embarazo	character varying(5)	Mujer: está embarazada
pastilla_anticonceptiva	character varying(5)	Está tomando pastillas anticonceptivas
desarreglos_mensuales	character varying(5)	presenta desarreglos mensuales
cicatrizacion_lenta	character varying(5)	posee cicatrización lenta
espec_grav_enfermo	character varying(50)	especifique si la enfermedad
espec_otros_medicamentos	character varying(50)	especifique a que otros medicamentos en alérgicos
barbitúricos	character varying(5)	alérgico a barbitúricos (píldoras para dormir)
espec_fecha_hospitalizacion	character varying(15)	especifique la fecha de la hospitalización
validar	character varying(15)	indica el estatus de la historia
consulta_id	integer	referencia a la consulta
ultimo_usuario	integer	referencia al estudiante que ingreso los datos
profesor	integer	referencia al profesor que realizo la validación
fecha	character varying(15)	indica la fecha del examen
fecha_validacion	character varying(15)	indica la fecha de la validación
nro_historia	integer	referencia a la historia del paciente

**Tabla C.3. Descripción de la tabla datos clínicos**

### Tabla exámenes complementarios

En esta tabla se almacena la información concerniente a exámenes complementarios para la atención del paciente.

Nombre	Tipo	Descripción
id_examen_complementario	character varying(20)	Número de cédula
resumen	character varying(50)	Nombres (s)
paciente_id	character varying(50)	Apellido (s)
consulta_id	date	Fecha de nacimiento
profesor	character varying(10)	Género: Masculino o Femenino
validar	character varying(16)	Número de teléfono
fecha_validacion	character varying(16)	Número de teléfono

**Tabla C.4. Descripción de la tabla exámenes complementarios**

### Tabla examen clínico

Esta tabla contiene la información acerca de la revisión física realizada al paciente

Nombre	Tipo	Descripción
id_examen_clinico	character varying(20)	identificador único
piel	character varying(150)	indica las características de la piel
cabeza	character varying(150)	indica las características de la cabeza
cara	character varying(150)	Indica las características de la cara
cuello	character varying(150)	indica las características del cuello
labios	character varying(150)	indica las características de los labios
carrillos	character varying(150)	indica las características de los carrillos
mucosa_labial_bucal	character varying(150)	indica la característica de la mucosa labial

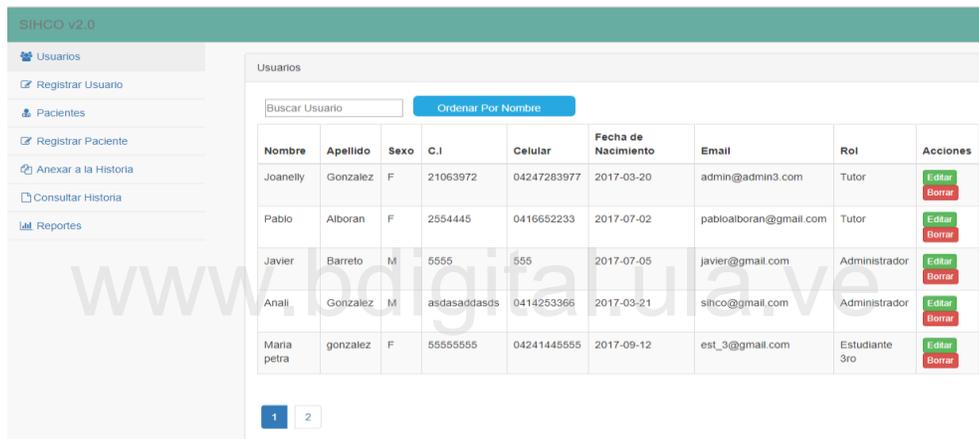
lengua	character varying(150)	indica las características de la lengua
piso_boca	character varying(150)	indica las características del piso de la boca
paladar_duro_blando	character varying(150)	indica el paladar duro blando
orofaringe	character varying(150)	indica las características de la orofaringe
pigmentación	character varying(150)	indica las características de la pigmentación
viscosidad	character varying(150)	indica las características de la viscosidad
observaciones	character varying(150)	indica otras observaciones
fecha		indica la fecha del examen
ultimo_usuario	character varying(15)	referencia al estudiante que ingreso los datos
nro_historia	integer	indica el número de historia del paciente
consulta_id	integer	indica el número de consulta
profesor	integer	indica el profesor que realizo la validación
validar	character varying(15)	indica el estatus del examen
fecha_validacion	character varying(15)	indica la fecha en la que se realizó la validación

**Tabla C.5. Descripción de la tabla examen clínico**

## Anexo D.

### Vistas del Sistema Web

En la Figura D.1 se observa al momento de hacer clic en la sección Usuario de la barra mostrada a un lado de la pantalla. En esta sección se podrá actualizar la información de los usuarios, además de eliminarlos.



Nombre	Apellido	Sexo	C.I.	Celular	Fecha de Nacimiento	Email	Rol	Acciones
Joanelly	Gonzalez	F	21063972	04247283977	2017-03-20	admin@admin3.com	Tutor	<a href="#">Editar</a> <a href="#">Borrar</a>
Pablo	Alboran	F	2554445	0416652233	2017-07-02	pabloalboran@gmail.com	Tutor	<a href="#">Editar</a> <a href="#">Borrar</a>
Javier	Barreto	M	5555	555	2017-07-05	javier@gmail.com	Administrador	<a href="#">Editar</a> <a href="#">Borrar</a>
Anail	Gonzalez	M	asdasaddasds	0414253366	2017-03-21	sihco@gmail.com	Administrador	<a href="#">Editar</a> <a href="#">Borrar</a>
Maria petra	gonzalez	F	55555555	04241445555	2017-09-12	est_3@gmail.com	Estudiante 3ro	<a href="#">Editar</a> <a href="#">Borrar</a>

**Figura D.1 Vista de Consulta de Usuarios**

En la Figura D.2 se observa la sección de gestión de los Pacientes, al momento de hacer clic en Pacientes de la barra vertical del lado izquierdo de la pantalla. En esta sección de igual forma se puede actualizar la información de los pacientes incluso su eliminación.

SIHCO v2.0

Usuarios

- Registrar Usuario
- Pacientes
- Registrar Paciente
- Anexar a la Historia
- Consultar Historia
- Reportes

Pacientes

Buscar Paciente Ordenar Por Nombre

Hist. Nro	Nombre	Apellido	Sexo	C.I	Celular	Fecha de Nacimiento	Direccion	Fecha de Ingreso	Acciones
13	Mario	perez	M	999	test	2017-03-21	test	2017-03-21	<a href="#">Editar</a> <a href="#">Borrar</a>
14	Luisa	Caceres	F	9166522	000	2017-03-21	Merida	2017-03-24	<a href="#">Editar</a> <a href="#">Borrar</a>
19	Estafania	Fernandez	F	222111	2222	2017-06-12	Direccion	2017-06-12	<a href="#">Editar</a> <a href="#">Borrar</a>
20	pedrito	perez	F	6666	21111	2017-07-05	Merida	2017-07-05	<a href="#">Editar</a> <a href="#">Borrar</a>
16	Luis	Hidalgo	M	9662233	04247523366	2017-03-21	Merida	2017-03-21	<a href="#">Editar</a> <a href="#">Borrar</a>

**Figura D.2 Vista de Consulta de Pacientes**

En la Figura D.3 y D.4 se observan los formularios de registros tanto de paciente como de usuarios, accediendo de igual forma a través del menú desplegado al lado izquierdo de la pantalla.

SIHCO v2.0

Usuarios

- Registrar Usuario
- Pacientes
- Registrar Paciente
- Anexar a la Historia
- Consultar Historia
- Reportes

Registrar Usuario

Nombre

Apellido

Cedula

Fecha de Nacimiento

Sexo Seleccione...

Direccion

Telefono

Celular

Tipo de usuario Seleccione...

Valor Seleccione...

Dirección de correo

Contraseña

**Figura D.3 Vista de Ingreso de usuario**

**Figura D.4 Vista de Ingreso de Pacientes**

A continuación, se adjuntas los scripts de creación de cada una de las tablas de la base de datos:

```

CREATE TABLE public.antecedentes_familiares
(
  id_antecedente_familiar          integer          NOT NULL          DEFAULT
  nextval('antecedentes_familiares_id_antecedente_familiar_seq'::regclass),
  neuropsicopatia character varying(5) DEFAULT 'N'::character varying,
  enfer_meta_endocrina character varying(5) DEFAULT 'N'::character varying,
  discrasia_sanguinea character varying(5) DEFAULT 'N'::character varying,
  fiebre_reumatica character varying(5) DEFAULT 'N'::character varying,
  artritis_reumatoidea character varying(5) DEFAULT 'N'::character varying,
  otro character varying(5) DEFAULT 'N'::character varying,
  espec_otro character varying(200),
  validar character varying(20),
  profesor integer,
  ultimo_usuario integer,
  fecha date,

```

```

nro_historia integer,
consulta_id integer,
enfer_renal integer,
created_at timestamp without time zone,
updated_at timestamp without time zone,
fecha_validacion character varying(50),
CONSTRAINT antecedentes_familiares_pkey PRIMARY KEY (id_antecedente_familiar),
CONSTRAINT consulta_id FOREIGN KEY (consulta_id)
REFERENCES public.consulta (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT enfer_renal_id FOREIGN KEY (enfer_renal)
REFERENCES public.paciente_enfer_renal (id_paciente_enfer_renal) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT paciente_id FOREIGN KEY (nro_historia)
REFERENCES public.paciente (nro_historias) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT profesor_id FOREIGN KEY (profesor)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT ultimo_usuario FOREIGN KEY (ultimo_usuario)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE public.antecedentes_familiares
OWNER TO postgres;

CREATE TABLE public.antecedentes_personales
(

```

id_antecedente_personal	integer	NOT	NULL	DEFAULT
-------------------------	---------	-----	------	---------

```

nextval('antecedentes_personale_id_antecedente_personal_seq'::regclass),
enfer_meta_endocrina character varying(5) DEFAULT 'N'::character varying,
discrasia_sanguinea character varying(5) DEFAULT 'N'::character varying,
fiebre_reumatica character varying(5) DEFAULT 'N'::character varying,
artritis_reumatoidea character varying(5) DEFAULT 'N'::character varying,
adenopatia character varying(5) DEFAULT 'N'::character varying,
otro character varying(5),
cefalea character varying(5) DEFAULT 'N'::character varying,
convulsiones character varying(5) DEFAULT 'N'::character varying,
parestesias character varying(5) DEFAULT 'N'::character varying,
edema_miembro_inf character varying(5) DEFAULT 'N'::character varying,
erupciones_piel character varying(5) DEFAULT 'N'::character varying,
ictericia character varying(5) DEFAULT 'N'::character varying,
trastorno_audicion character varying(5) DEFAULT 'N'::character varying,
intervencion_quirurgica character varying(5) DEFAULT 'N'::character varying,
trauma_accidente character varying(5) DEFAULT 'N'::character varying,
fractura character varying(5) DEFAULT 'N'::character varying,
accidente_transito character varying(5) DEFAULT 'N'::character varying,
espec_otro character varying(200),
espec_meta_endocrina character varying(200),
espec_trauma_accidente character varying(200),
espec_fractura character varying(200),
espec_accidente_transito character varying(200),
consulta_id integer,
validar character varying(20),
ultimo_usuario integer,
profesor integer,
nro_historia integer,
fecha date,
espec_accidente_trans character varying(200),

```

```

espec_interv_quirurgica character varying(200),
created_at timestamp without time zone,
updated_at timestamp without time zone,
fecha_validacion character varying(50),
CONSTRAINT antecedentes_personales_pkey PRIMARY KEY (id_antecedente_personal),
CONSTRAINT consulta_id FOREIGN KEY (consulta_id)
REFERENCES public.consulta (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT paciente_id FOREIGN KEY (nro_historia)
REFERENCES public.paciente (nro_historias) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT profesor_id FOREIGN KEY (profesor)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT ultimo_usuario FOREIGN KEY (ultimo_usuario)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE public.antecedentes_personales
OWNER TO postgres;
CREATE TABLE public.cirugia
(
infeciosas character varying(2) DEFAULT 'N'::character varying,
espec_infecciosa character varying(200),
metabolicas character varying(2) DEFAULT 'N'::character varying,
cardiovasculares character varying(2) DEFAULT 'N'::character varying,
gastrointestinales character varying(2) DEFAULT 'N'::character varying,
genito_urinaria character varying(2) DEFAULT 'N'::character varying,

```

neurologica character varying(2) DEFAULT 'N'::character varying,  
espec\_patologico character varying(200),  
causa character varying(200),  
causa\_quirurgico character varying(200),  
hubo\_complicacion character varying(200),  
obser\_quirurgico character varying(2) DEFAULT 'N'::character varying,  
alegia character varying(2) DEFAULT 'N'::character varying,  
cual\_alergia character varying(200),  
dosis character varying(200),  
medicamento character varying(2) DEFAULT 'N'::character varying,  
cual\_medicamento character varying(200),  
hematologicos character varying(2) DEFAULT 'N'::character varying,  
causa\_hema character varying(200),  
transfusionales character varying(2) DEFAULT 'N'::character varying,  
causa\_trans character varying(200),  
embarazada character varying(2) DEFAULT 'N'::character varying,  
otros\_no\_especificados character varying(200),  
cirugia character varying(2) DEFAULT 'N'::character varying,  
tipo character varying(150),  
complicaciones\_cirugia character varying(2) DEFAULT 'N'::character varying,  
espec\_complicaciones character varying(200),  
examenes\_complementarios character varying(200),  
consulta\_id integer,  
validar character varying(20),  
ultimo\_usuario integer,  
profesor integer,  
nro\_historia integer,  
fecha date,  
id\_cirugia integer NOT NULL DEFAULT nextval('cirugia\_id\_cirugia\_seq'::regclass),  
fecha\_validacion character varying(15),  
created\_at timestamp without time zone,

```

updated_at timestamp without time zone,
fecha_hospitalizacion character varying(15),
fecha_hospitalizacion_quirurgico character varying(15),
hospitalizado character varying(2) DEFAULT 'N'::character varying,
operado character varying(2) DEFAULT 'N'::character varying,
CONSTRAINT cirugia_pkey PRIMARY KEY (id_cirugia),
CONSTRAINT cirugia_consulta_id_fkey FOREIGN KEY (consulta_id)
    REFERENCES public.consulta (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT cirugia_paciente_id_fkey FOREIGN KEY (nro_historia)
    REFERENCES public.paciente (nro_historias) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT cirugia_profesor_fkey FOREIGN KEY (profesor)
    REFERENCES public.usuarios (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT cirugia_ultimo_usuario_fkey FOREIGN KEY (ultimo_usuario)
    REFERENCES public.usuarios (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.cirugia
    OWNER TO postgres;

CREATE TABLE public.citas
(
    id_cita integer NOT NULL DEFAULT nextval('citas_id_cita_seq'::regclass),
    nro_historia integer,
    especialidad character varying(15),
    clinica integer,

```

```

observacion character varying(10),
validar character varying(15),
ultimo_usuario integer,
estatus character varying(20),
profesor integer,
motivo character varying(100),
fecha_cita character varying(25),
hora character varying(20),
CONSTRAINT citas_pkey PRIMARY KEY (id_cita),
CONSTRAINT citas_profesor_fkey FOREIGN KEY (profesor)
    REFERENCES public.usuarios (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT citas_ultimo_usuario_fkey FOREIGN KEY (ultimo_usuario)
    REFERENCES public.usuarios (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.citas
    OWNER TO postgres;

```

```

CREATE TABLE public.consulta
(
    id integer NOT NULL DEFAULT nextval('consulta_id_consulta_seq'::regclass),
    nro_historia character varying(45) NOT NULL,
    motivo_consulta character varying(200) NOT NULL,
    enfermedad_actual character varying(200) NOT NULL,
    fecha_consulta date NOT NULL,
    ultimo_usuario integer,
    created_at timestamp without time zone,

```

```

updated_at timestamp without time zone,
profesor integer,
validar character varying(15),
fecha_validacion character varying(15),
CONSTRAINT consulta_pkey PRIMARY KEY (id)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.consulta
  OWNER TO postgres;

```

```

CREATE TABLE public.control_placa
(
  diente_presentes character varying(2),
  cantidad_dientes_placa character varying(3),
  fecha date,
  ultimo_usuario integer,
  nro_historia integer,
  consulta_id integer,
  profesor integer,
  validar character varying(15),
  created_at timestamp without time zone,
  updated_at timestamp without time zone,
  fecha_validacion character varying(50),
  id_control_placa          integer          NOT          NULL          DEFAULT
  nextval('control_placa_id_control_placa_seq'::regclass),
  CONSTRAINT control_placa_pkey PRIMARY KEY (id_control_placa),
  CONSTRAINT control_placa_consulta_id_fkey FOREIGN KEY (consulta_id)
  REFERENCES public.consulta (id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,

```

```

CONSTRAINT control_placa_paciente_id_fkey FOREIGN KEY (nro_historia)
  REFERENCES public.paciente (nro_historias) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT control_placa_profesor_fkey FOREIGN KEY (profesor)
  REFERENCES public.usuarios (id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT control_placa_ultimo_usuario_fkey FOREIGN KEY (ultimo_usuario)
  REFERENCES public.usuarios (id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.control_placa
  OWNER TO postgres;

CREATE TABLE public.coronas_puentes
(
  id_coronas integer NOT NULL DEFAULT nextval('coronas_puentes_id_coronas_seq'::regclass),
  tipo_corona character varying(50),
  material character varying(50),
  tipo_puente character varying(50),
  retenedores character varying(50),
  panticos character varying(50),
  dientes_pilares character varying(50),
  conectores character varying(50),
  material_puente character varying(50),
  fecha date,
  ultimo_usuario integer,
  nro_historia integer,
  consulta_id integer,

```

```

profesor integer,
validar character varying(15),
created_at timestamp without time zone,
updated_at timestamp without time zone,
fecha_validacion character varying(50),
CONSTRAINT id_corona PRIMARY KEY (id_coronas),
CONSTRAINT consulta_id FOREIGN KEY (consulta_id)
REFERENCES public.consulta (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT paciente_id FOREIGN KEY (nro_historia)
REFERENCES public.paciente (nro_historias) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT profesor_id FOREIGN KEY (profesor)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT ultimo_usuario_id FOREIGN KEY (ultimo_usuario)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
oids=false
);
ALTER TABLE public.coronas_puentes
OWNER TO postgres;

CREATE TABLE public.datos_clinicos
(
id_dato_clinico integer NOT NULL DEFAULT
nextval('datos_clinicos_id_dato_clinico_seq'::regclass),
examinado_ultimo_ayo character varying(5) DEFAULT 'N'::character varying,
cambio_salud_ultimo_ayo character varying(5) DEFAULT 'N'::character varying,

```

sangra\_largo\_tiempo character varying(5) DEFAULT 'N'::character varying,  
perdida\_peso character varying(5) DEFAULT 'N'::character varying,  
grave\_enfermo character varying(5) DEFAULT 'N'::character varying,  
hospitalizado character varying(5) DEFAULT 'N'::character varying,  
marcapaso character varying(5) DEFAULT 'N'::character varying,  
transfusion\_sanguinea character varying(5) DEFAULT 'N'::character varying,  
asma character varying(5) DEFAULT 'N'::character varying,  
aspirina character varying(5) DEFAULT 'N'::character varying,  
penicilina character varying(5) DEFAULT 'N'::character varying,  
yodo character varying(5) DEFAULT 'N'::character varying,  
sulfonamida character varying(5) DEFAULT 'N'::character varying,  
otros\_medicamentos character varying(5) DEFAULT 'N'::character varying,  
trastorno\_visual character varying(5) DEFAULT 'N'::character varying,  
sinusitis character varying(5) DEFAULT 'N'::character varying,  
hemorragia\_nasal character varying(5) DEFAULT 'N'::character varying,  
dolor\_pecho character varying(5) DEFAULT 'N'::character varying,  
tos character varying(5) DEFAULT 'N'::character varying,  
tos\_sangre character varying(5) DEFAULT 'N'::character varying,  
dificultad\_tragar character varying(5) DEFAULT 'N'::character varying,  
indigestion character varying(5) DEFAULT 'N'::character varying,  
vomito character varying(5) DEFAULT 'N'::character varying,  
orines character varying(5) DEFAULT 'N'::character varying,  
sediento character varying(5) DEFAULT 'N'::character varying,  
desmayo character varying(5) DEFAULT 'N'::character varying,  
moretones character varying(5) DEFAULT 'N'::character varying,  
desorden\_sangre character varying(5) DEFAULT 'N'::character varying,  
cansancio character varying(5) DEFAULT 'N'::character varying,  
embarazo character varying(5) DEFAULT 'N'::character varying,  
pastilla\_anticonceptiva character varying(5) DEFAULT 'N'::character varying,  
desarreglos\_mensuales character varying(5) DEFAULT 'N'::character varying,  
cicatrizacion\_lenta character varying(5) DEFAULT 'N'::character varying,

```

espec_grav_enfermo character varying(200),
espec_otros_medicamentos character varying(200),
validar character varying(20),
consulta_id integer,
nro_historia integer,
ultimo_usuario integer,
profesor integer,
fecha date,
barbituricos character varying(5) DEFAULT 'N'::character varying,
espec_fecha_hospitalizacion character varying(20),
created_at timestamp without time zone,
updated_at timestamp without time zone,
fecha_validacion character varying(50),
CONSTRAINT id_dato_clinico PRIMARY KEY (id_dato_clinico),
CONSTRAINT consulta_id FOREIGN KEY (consulta_id)
REFERENCES public.consulta (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT paciente_id FOREIGN KEY (nro_historia)
REFERENCES public.paciente (nro_historias) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT profesor_id FOREIGN KEY (profesor)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT ultimo_usuario FOREIGN KEY (ultimo_usuario)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE public.datos_clinicos

```

OWNER TO postgres;

CREATE TABLE public.dentaduras\_totales

```
(  
  maxilar_caries character varying(2) DEFAULT 'N'::character varying,  
  maxilar_ambas character varying(2) DEFAULT 'N'::character varying,  
  maxilar_accidente character varying(2) DEFAULT 'N'::character varying,  
  maxilar_anos character varying(2) DEFAULT 'N'::character varying,  
  mandibular_caries character varying(2) DEFAULT 'N'::character varying,  
  mandibular_ambas character varying(2) DEFAULT 'N'::character varying,  
  mandibular_accidente character varying(2) DEFAULT 'N'::character varying,  
  mandibular_anos character varying(2) DEFAULT 'N'::character varying,  
  dentadura_ninguna character varying(2) DEFAULT 'N'::character varying,  
  dentadura_cuanta character varying(5),  
  dentadura_buena character varying(2) DEFAULT 'N'::character varying,  
  dentadura_mala character varying(2) DEFAULT 'N'::character varying,  
  dentadura_uso character varying(2) DEFAULT 'N'::character varying,  
  razon_reemplazo character varying(100),  
  tiempo_uso character varying(100),  
  tipo_detadura_sup character varying(2) DEFAULT 'N'::character varying,  
  tipo_detadura_inf character varying(2) DEFAULT 'N'::character varying,  
  tipo_detadura_ambas character varying(2) DEFAULT 'N'::character varying,  
  tipo_detadura_nunguna character varying(2) DEFAULT 'N'::character varying,  
  uso_dentadura character varying(50),  
  material_diente character varying(50),  
  forma_diente character varying(50),  
  material_base character varying(50),  
  eval_comodidad character varying(50),  
  eval_efic_masticatoria character varying(50),  
  eval_estetica character varying(50),  
  eval_pronunciacion character varying(50),
```

eval\_dolor character varying(50),  
eval\_dist\_adecuada character varying(50),  
eval\_mas character varying(50),  
eval\_menos character varying(50),  
eval\_estabilidad character varying(50),  
eval\_occlusion character varying(50),  
eval\_retencion character varying(50),  
eval\_estetica\_dent character varying(50),  
eval\_dimension\_vertical character varying(50),  
eval\_prese\_cam character varying(50),  
eval\_higiene character varying(50),  
dentadura\_maxilar character varying(2),  
dentadura\_mandibular character varying(2),  
expresion\_facial character varying(50),  
labios character varying(50),  
relacion\_labio\_alveolar character varying(50),  
reabsorcion\_sup character varying(50),  
reabsorcion\_inf character varying(50),  
relacion\_reborde character varying(50),  
piso\_boca character varying(50),  
resilencia\_sup character varying(50),  
resilencia\_inf character varying(50),  
insercion\_sup character varying(50),  
insercion\_inf character varying(50),  
abertura\_boca character varying(50),  
tam\_sup character varying(50),  
tam\_inf character varying(50),  
contorno\_boveda character varying(50),  
contorno\_paladar character varying(50),  
anchura character varying(50),  
depresibilidad character varying(50),

palatino\_ausente character varying(50),  
palatino\_presente character varying(50),  
palatino\_tam character varying(50),  
mandibular\_ausente character varying(50),  
mandibular\_presente character varying(50),  
mandibular\_tam character varying(50),  
espacio character varying(50),  
mm character varying(50),  
posicion character varying(50),  
seleccion\_sup character varying(150),  
seleccion\_inf character varying(150),  
material character varying(150),  
diente\_sup character varying(50),  
diente\_inf character varying(50),  
relacion\_inter character varying(150),  
relacion\_borde character varying(150),  
espacio\_disp character varying(150),  
espacion\_disp\_occlusal character varying(150),  
seleccion\_arti\_color character varying(150),  
seleccion\_arti\_marca character varying(150),  
formula\_anterior character varying(150),  
formula\_posterior character varying(150),  
formula\_marca character varying(150),  
numero\_cita character varying(150),  
comodidad character varying(150),  
eficiencia\_masticatoria character varying(150),  
post\_estetica character varying(150),  
pronunciacion character varying(150),  
validar character varying(20),  
consulta\_id integer,  
nro\_historia integer,

```

ultimo_usuario integer,
profesor integer,
fecha date,
created_at timestamp without time zone,
updated_at timestamp without time zone,
fecha_validacion character varying(50),
id_dentadura_total          integer          NOT          NULL          DEFAULT
nextval('dentaduras_totales_id_dentadura_total_seq'::regclass),
dentadura_si_usuado character varying(2) DEFAULT 'N'::character varying,
eval_extension character varying(50),
mm_vertical character varying(5),
CONSTRAINT dentaduras_totales_pkey PRIMARY KEY (id_dentadura_total),
CONSTRAINT dentaduras_totales_consulta_id_fkey FOREIGN KEY (consulta_id)
REFERENCES public.consulta (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT dentaduras_totales_paciente_id_fkey FOREIGN KEY (nro_historia)
REFERENCES public.paciente (nro_historias) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT dentaduras_totales_profesor_fkey FOREIGN KEY (profesor)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT dentaduras_totales_ultimo_usuario_fkey FOREIGN KEY (ultimo_usuario)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE public.dentaduras_totales
OWNER TO postgres;

```

```

CREATE TABLE public.diagnostico
(
    fecha_tratamiento character varying(15),
    diagnostico character varying(200),
    diagnostico_id integer,
    id_diagnostico integer NOT NULL DEFAULT nextval('diagnostico_id_diagnostico_seq'::regclass),
    especialidad character varying(15),
    CONSTRAINT diagnostico_pkey PRIMARY KEY (id_diagnostico),
    CONSTRAINT diagnostico_diagnostico_id_fkey FOREIGN KEY (diagnostico_id)
        REFERENCES public.diagnosticos (id_diagnosticos) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.diagnostico
    OWNER TO postgres;

CREATE TABLE public.diagnosticos
(
    fecha date,
    consulta_id integer,
    nro_historia integer,
    validar character varying(15),
    fecha_validacion character varying(15),
    profesor integer,
    ultimo_usuario integer,
    id_diagnosticos integer NOT NULL DEFAULT nextval('diagnosticos_id_diagnosticos_seq'::regclass),
    tipo character varying(15),
    CONSTRAINT diagnosticos_pkey PRIMARY KEY (id_diagnosticos)
)

```

```

WITH (
  OIDS=FALSE
);
ALTER TABLE public.diagnostics
  OWNER TO postgres;

CREATE TABLE public.diagrama_riesgo
(
  id_diagrama_riesgo          integer          NOT          NULL          DEFAULT
  nextval('diagrama_riesgo_id_diagrama_riesgo_seq'::regclass),
  bop character varying(15),
  pd character varying(15),
  tooth character varying(15),
  bl_age character varying(15),
  sys_age character varying(15),
  envir character varying(15),
  nro_historia integer,
  consulta_id integer,
  ultimo_usuario integer,
  validar character varying(15),
  profesor integer,
  fecha date,
  created_at timestamp without time zone,
  updated_at timestamp without time zone,
  fecha_validacion character varying(50),
  CONSTRAINT diagrama_riesgo_pkey PRIMARY KEY (id_diagrama_riesgo),
  CONSTRAINT diagrama_riesgo_consulta_id_fkey FOREIGN KEY (consulta_id)
  REFERENCES public.consulta (id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT diagrama_riesgo_paciente_id_fkey FOREIGN KEY (nro_historia)

```

```

REFERENCES public.paciente (nro_historias) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT diagrama_riesgo_profesor_fkey FOREIGN KEY (profesor)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT diagrama_riesgo_ultimo_usuario_fkey FOREIGN KEY (ultimo_usuario)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.diagrama_riesgo
  OWNER TO postgres;

CREATE TABLE public.dientes_con_placa
(
  id_diente integer NOT NULL DEFAULT nextval('dientes_con_placa_id_diente_seq'::regclass),
  valor_diente integer,
  cara_diente integer,
  estado_diente integer,
  tipo_diente integer,
  control_placa_id integer,
  created_at timestamp with time zone,
  updated_at timestamp with time zone,
  CONSTRAINT dientes_con_placa_pkey PRIMARY KEY (id_diente),
  CONSTRAINT dientes_con_placa_control_placa_id_fkey FOREIGN KEY (control_placa_id)
REFERENCES public.control_placa (id_control_placa) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (

```

```

        OIDS=FALSE
    );
ALTER TABLE public.dientes_con_placa
    OWNER TO postgres;

CREATE TABLE public.elementos_odontograma
(
    id_elemento          integer          NOT          NULL          DEFAULT
nextval('elementos_odontograma_id_elemento_seq'::regclass),
    elemento             character varying(50),
    odontograma_id      integer,
    posicion_x           character varying(15),
    posicion_y           character varying(15),
    CONSTRAINT elementos_odontograma_pkey PRIMARY KEY (id_elemento),
    CONSTRAINT elementos_odontograma_odontograma_id_fkey FOREIGN KEY (odontograma_id)
REFERENCES public.odontograma (id_odontograma) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.elementos_odontograma
    OWNER TO postgres;

CREATE TABLE public.endodoncia
(
    id_endodoncia integer NOT NULL DEFAULT nextval('endodoncia_id_endodoncia_seq'::regclass),
    frio character varying(2) DEFAULT 'N'::character varying,
    calor character varying(2) DEFAULT 'N'::character varying,
    masticacion character varying(2) DEFAULT 'N'::character varying,
    palpacion character varying(2) DEFAULT 'N'::character varying,

```

created\_at timestamp without time zone,  
updated\_at timestamp without time zone,  
fecha\_validacion character varying(50),  
percusion\_vertical character varying(2) DEFAULT 'N'::character varying,  
dolor\_localizado character varying(2) DEFAULT 'N'::character varying,  
dolor\_difuso character varying(2) DEFAULT 'N'::character varying,  
dolor\_provocado character varying(2) DEFAULT 'N'::character varying,  
dolor\_espontaneo character varying(2) DEFAULT 'N'::character varying,  
dolor\_constante character varying(2) DEFAULT 'N'::character varying,  
dolor\_intermitente character varying(2) DEFAULT 'N'::character varying,  
dolor\_palpitante character varying(2) DEFAULT 'N'::character varying,  
dolor\_referido character varying(2) DEFAULT 'N'::character varying,  
ausencia\_dolor character varying(2),  
diente\_tratar character varying(50),  
observacion character varying(250),  
infla\_intraoral character varying(2) DEFAULT 'N'::character varying,  
infla\_extraoral character varying(2) DEFAULT 'N'::character varying,  
fistula character varying(2) DEFAULT 'N'::character varying,  
caries character varying(2) DEFAULT 'N'::character varying,  
obsturaciones character varying(2) DEFAULT 'N'::character varying,  
coronas\_metal character varying(2) DEFAULT 'N'::character varying,  
fractura\_corona character varying(2) DEFAULT 'N'::character varying,  
fisura character varying(2) DEFAULT 'N'::character varying,  
cambio\_color character varying(2) DEFAULT 'N'::character varying,  
movilidad character varying(2) DEFAULT 'N'::character varying,  
polipo\_pulpar character varying(2) DEFAULT 'N'::character varying,  
faceta\_articular character varying(2) DEFAULT 'N'::character varying,  
labios character varying(2) DEFAULT 'N'::character varying,  
lengua character varying(2) DEFAULT 'N'::character varying,  
paladar\_duro character varying(2) DEFAULT 'N'::character varying,  
paladar\_blando character varying(2) DEFAULT 'N'::character varying,

frenillos character varying(2) DEFAULT 'N'::character varying,  
mucosa\_oral character varying(2) DEFAULT 'N'::character varying,  
piso\_boca character varying(2) DEFAULT 'N'::character varying,  
glandulas\_salivales character varying(2) DEFAULT 'N'::character varying,  
sist\_linfatico character varying(2) DEFAULT 'N'::character varying,  
musculos\_masticatorios character varying(2) DEFAULT 'N'::character varying,  
placa character varying(2) DEFAULT 'N'::character varying,  
calculos character varying(2) DEFAULT 'N'::character varying,  
gingivitis character varying(2) DEFAULT 'N'::character varying,  
periodontitis character varying(2) DEFAULT 'N'::character varying,  
bolsa\_periodontal character varying(2) DEFAULT 'N'::character varying,  
test\_cavidad character varying(2) DEFAULT 'N'::character varying,  
prueba\_electrica character varying(2) DEFAULT 'N'::character varying,  
frio\_sensibilidad character varying(2) DEFAULT 'N'::character varying,  
calor\_sensibilidad character varying(2) DEFAULT 'N'::character varying,  
espacio\_periodontal character varying(2) DEFAULT 'N'::character varying,  
perdida\_alveolar character varying(2) DEFAULT 'N'::character varying,  
zona\_radiolucida character varying(2) DEFAULT 'N'::character varying,  
zona\_radiolucida\_lateral character varying(2) DEFAULT 'N'::character varying,  
zona\_radiolucida\_furca character varying(2) DEFAULT 'N'::character varying,  
reabsorcion\_interna character varying(2) DEFAULT 'N'::character varying,  
reabsorcion\_externa character varying(2) DEFAULT 'N'::character varying,  
radiopacidad\_apical character varying(2) DEFAULT 'N'::character varying,  
raiz\_formacion character varying(2) DEFAULT 'N'::character varying,  
nucleo character varying(2) DEFAULT 'N'::character varying,  
fractura\_radicular character varying(2) DEFAULT 'N'::character varying,  
tratamiento\_endodontico character varying(2) DEFAULT 'N'::character varying,  
otros character varying(2) DEFAULT 'N'::character varying,  
bacteriana character varying(2) DEFAULT 'N'::character varying,  
quimica character varying(2) DEFAULT 'N'::character varying,  
fisica character varying(2) DEFAULT 'N'::character varying,

```

trauma character varying(2) DEFAULT 'N'::character varying,
fines_protesicos character varying(2) DEFAULT 'N'::character varying,
fines_terapeuticos character varying(2) DEFAULT 'N'::character varying,
ultimo_usuario integer,
validar character varying(15),
fecha character varying(15),
nro_historia integer,
consulta_id integer,
profesor integer,
CONSTRAINT endodoncia_pkey PRIMARY KEY (id_endodoncia),
CONSTRAINT endodoncia_consulta_id_fkey FOREIGN KEY (consulta_id)
REFERENCES public.consulta (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT endodoncia_paciente_id_fkey FOREIGN KEY (nro_historia)
REFERENCES public.paciente (nro_historias) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT endodoncia_profesor_fkey FOREIGN KEY (profesor)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT endodoncia_ultimo_usuario_fkey FOREIGN KEY (ultimo_usuario)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE public.endodoncia
OWNER TO postgres;

CREATE TABLE public.enfermedades_cardiovasculares
(

```

```

id_enfermedad_cardiovascular          integer          NOT          NULL          DEFAULT
nextval('enfermedades_cardiovasculares_id_enfermedad_cardiovascular_seq'::regclass),
enfermedad character varying(50),
ultimo_usuario integer,
created_at timestamp without time zone,
updated_at timestamp without time zone,
CONSTRAINT          enfermedades_cardiovasculares_pkey          PRIMARY          KEY
(id_enfermedad_cardiovascular),
CONSTRAINT          enfermedades_cardiovasculares_ultimo_usuario_fkey          FOREIGN          KEY
(ultimo_usuario)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE public.enfermedades_cardiovasculares
OWNER TO postgres;

```

```

CREATE TABLE public.enfermedades_patologicas
(
id_enfermedad_patologica          integer          NOT          NULL          DEFAULT
nextval('enfermedades_patologicas_id_enfermedad_patologica_seq'::regclass),
enfermedad character varying(50),
valor_id integer,
ultimo_usuario integer,
created_at timestamp without time zone,
updated_at timestamp without time zone,
CONSTRAINT id_enfermedad_patologica PRIMARY KEY (id_enfermedad_patologica),
CONSTRAINT enfermedades_patologicas_ultimo_usuario_fkey FOREIGN KEY (ultimo_usuario)
REFERENCES public.usuarios (id) MATCH SIMPLE
)

```

```

        ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT valor_id FOREIGN KEY (valor_id)
    REFERENCES public.valores_listas (id_valor) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.enfermedades_patologicas
    OWNER TO postgres;

CREATE TABLE public.enfermedades_renales
(
    enfermedad character varying(50),
    ultimo_usuario integer,
    id_enfermedad_renal integer NOT NULL DEFAULT
nextval('enfermedades_renales_id_enfermedad_renal_seq'::regclass),
    created_at timestamp without time zone,
    updated_at timestamp without time zone,
    CONSTRAINT enfermedades_renales_pkey PRIMARY KEY (id_enfermedad_renal),
    CONSTRAINT enfermedades_renales_ultimo_usuario_fkey FOREIGN KEY (ultimo_usuario)
    REFERENCES public.usuarios (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.enfermedades_renales
    OWNER TO postgres;

CREATE TABLE public.evaluacion_periodontal

```

```

(
  id_evaluacion_periodontal      integer      NOT      NULL      DEFAULT
nextval('evaluacion_periodontal_id_evaluacion_periodontal_seq'::regclass),
  color character varying(150),
  consistencia character varying(150),
  contorno character varying(150),
  textura character varying(150),
  grosor character varying(150),
  ancho_encia character varying(150),
  posicion character varying(150),
  sangramiento character varying(150),
  cepillado character varying(150),
  espontaneo character varying(150),
  otro character varying(150),
  haliatosis character varying(150),
  sensibilidad character varying(150),
  fecha date,
  ultimo_usuario integer,
  profesor integer,
  nro_historia integer,
  consulta_id integer,
  validar character varying(15),
  created_at timestamp without time zone,
  updated_at timestamp without time zone,
  fecha_validacion character varying(50),
  CONSTRAINT id_perio PRIMARY KEY (id_evaluacion_periodontal),
  CONSTRAINT consulta_id FOREIGN KEY (consulta_id)
  REFERENCES public.consulta (id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT paciente_id FOREIGN KEY (nro_historia)
  REFERENCES public.paciente (nro_historias) MATCH SIMPLE

```

```

        ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT profesor_id FOREIGN KEY (profesor)
    REFERENCES public.usuarios (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT ultimo_usuario_id FOREIGN KEY (ultimo_usuario)
    REFERENCES public.usuarios (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.evaluacion_periodontal
    OWNER TO postgres;

```

```

CREATE TABLE public.examen_clinico
(
    id_examen_clinico          integer          NOT NULL          DEFAULT
nextval('examen_clinico_id_examen_clinico_seq'::regclass),
    piel character varying(150),
    cabeza character varying(150),
    cara character varying(150),
    cuello character varying(150),
    labios character varying(150),
    carrillos character varying(150),
    mucosa_labial_bucal character varying(150),
    lengua character varying(150),
    piso_boca character varying(150),
    paladar_duro_blando character varying(150),
    orofaringe character varying(150),
    pigmentacion character varying(150),
    viscosidad character varying(50),

```

```

observaciones character varying(150),
fecha date,
ultimo_usuario integer,
nro_historia integer,
consulta_id integer,
profesor integer,
validar character varying(15),
created_at timestamp with time zone,
updated_at timestamp with time zone,
fecha_validacion character varying(50),
CONSTRAINT id_examenn PRIMARY KEY (id_examen_clinico),
CONSTRAINT consulta_id FOREIGN KEY (consulta_id)
    REFERENCES public.consulta (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT paciente_id FOREIGN KEY (nro_historia)
    REFERENCES public.paciente (nro_historias) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT profesor_id FOREIGN KEY (profesor)
    REFERENCES public.usuarios (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT ultimo_usuario_id FOREIGN KEY (ultimo_usuario)
    REFERENCES public.usuarios (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.examen_clinico
    OWNER TO postgres;

CREATE TABLE public.examen_complementario

```

```

(
  resumen character varying(250),
  fecha date,
  ultimo_usuario integer,
  nro_historia integer,
  consulta_id integer,
  profesor integer,
  validar character varying(15),
  created_at timestamp with time zone,
  updated_at timestamp with time zone,
  fecha_validacion character varying(50),
  id_examen_complementario          integer          NOT          NULL          DEFAULT
nextval('examen_complementario_id_examen_complementario_seq'::regclass),
  CONSTRAINT examen_complementario_pkey PRIMARY KEY (id_examen_complementario),
  CONSTRAINT examen_complementario_consulta_id_fkey FOREIGN KEY (consulta_id)
  REFERENCES public.consulta (id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT examen_complementario_paciente_id_fkey FOREIGN KEY (nro_historia)
  REFERENCES public.paciente (nro_historias) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT examen_complementario_profesor_fkey FOREIGN KEY (profesor)
  REFERENCES public.usuarios (id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT examen_complementario_ultimo_usuario_fkey FOREIGN KEY (ultimo_usuario)
  REFERENCES public.usuarios (id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.examen_complementario

```

OWNER TO postgres;

```
CREATE TABLE public.examen_muscular
(
    temporal_posterio_der character varying(5),
    temporal_posterio_izq character varying(5),
    temporal_medio_der character varying(5),
    temporal_medio_izq character varying(5),
    temporal_anterior_der character varying(5),
    temporal_anterior_izq character varying(5),
    origen_masetero_der character varying(5),
    origen_masetero_izq character varying(5),
    cuerpo_masetero_der character varying(5),
    cuerpo_masetero_izq character varying(5),
    insercion_masetero_der character varying(5),
    insercion_masetero_izq character varying(5),
    region_mandibular_der character varying(5),
    region_mandibular_izq character varying(5),
    region_submandibular_der character varying(5),
    region_submandibular_izq character varying(5),
    esternocleidomastoideo_der character varying(5),
    esternocleidomastoideo_izq character varying(5),
    musculatura_der character varying(5),
    musculatura_izq character varying(5),
    area_pterigoide_der character varying(5),
    area_pterigoide_izq character varying(5),
    tendon_temp_der character varying(5),
    tendon_temp_izq character varying(5),
    polo_lateral_der character varying(5),
    polo_lateral_izq character varying(5),
    insercion_pos_der character varying(5),
```

```

insercion_pos_izq character varying(5),
atm_apertura_der character varying(50),
atm_cierre_der character varying(50),
atm_lateralidad_der character varying(50),
atm_lateralidad_izq character varying(50),
atm_protrusion_der character varying(50),
atm_apertura_izq character varying(50),
atm_cierre_izq character varying(50),
atm_izq_lateralidad_der character varying(50),
atm_izq_lateralidad_izq character varying(50),
atm_protrusion_izq character varying(50),
fecha date,
ultimo_usuario integer,
nro_historia integer,
consulta_id integer,
profesor integer,
validar character varying(15),
id_examen_muscular          integer          NOT          NULL          DEFAULT
nextval('examen_muscular_id_examen_muscular_seq'::regclass),
created_at timestamp without time zone,
updated_at timestamp without time zone,
fecha_validacion character varying(50),
CONSTRAINT examen_muscular_pkey PRIMARY KEY (id_examen_muscular),
CONSTRAINT consulta_id FOREIGN KEY (consulta_id)
    REFERENCES public.consulta (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT paciente_id FOREIGN KEY (nro_historia)
    REFERENCES public.paciente (nro_historias) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT profesor_id FOREIGN KEY (profesor)
    REFERENCES public.usuarios (id) MATCH SIMPLE

```

```

        ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT ultimo_usuario_id FOREIGN KEY (ultimo_usuario)
REFERENCES public.usuarios (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.examen_muscular
    OWNER TO postgres;

CREATE TABLE public.examen_occlusion
(
    id_occlusion integer NOT NULL DEFAULT nextval('examen_occlusion_id_occlusion_seq'::regclass),
    fremilus character varying(150),
    normocclusion character varying(2) DEFAULT 'N'::character varying,
    clase_dos character varying(2) DEFAULT 'N'::character varying,
    clase_tres character varying(2) DEFAULT 'N'::character varying,
    abierta_anterior character varying(2) DEFAULT 'N'::character varying,
    abierta_posterior_derecha character varying(2) DEFAULT 'N'::character varying,
    abierta_posterior_izquierda character varying(2) DEFAULT 'N'::character varying,
    a_tope_anterior character varying(2) DEFAULT 'N'::character varying,
    a_tope_posterior_derecha character varying(2) DEFAULT 'N'::character varying,
    a_tope_posterior_izquierda character varying(2) DEFAULT 'N'::character varying,
    cruzada_anterior character varying(2) DEFAULT 'N'::character varying,
    cruzada_posterior_derecha character varying(2) DEFAULT 'N'::character varying,
    cruzada_posterior_izquierda character varying(2) DEFAULT 'N'::character varying,
    sobremordida_horizontal character varying(5),
    sobremordida_vertical character varying(5),
    desviacion_derecha character varying(2) DEFAULT 'N'::character varying,
    desviacion_izquierda character varying(2) DEFAULT 'N'::character varying,

```

dimension\_normal character varying(50),  
dimension\_aumentada character varying(50),  
dimension\_disminuida character varying(50),  
dvr character varying(15),  
dvo character varying(15),  
total character varying(15),  
deslizamiento\_anterior character varying(2) DEFAULT 'N'::character varying,  
deslizamiento\_derecha character varying(2) DEFAULT 'N'::character varying,  
deslizamiento\_izquierda character varying(2) DEFAULT 'N'::character varying,  
funcion\_grupo character varying(2) DEFAULT 'N'::character varying,  
guia\_canina character varying(2) DEFAULT 'N'::character varying,  
guia\_incisiva character varying(2) DEFAULT 'N'::character varying,  
maxima\_apertura character varying(10),  
maxima\_lat\_derecha character varying(10),  
maxima\_lat\_izquierda character varying(10),  
maxima\_protrusion character varying(10),  
max\_apertura character varying(2),  
max\_lat\_derecha character varying(2),  
max\_lat\_izquierda character varying(2),  
max\_protrusion character varying(2),  
fecha date,  
consulta\_id integer,  
nro\_historia integer,  
profesor integer,  
validar character varying(15),  
fecha\_validacion character varying(15),  
created\_at timestamp without time zone,  
updated\_at timestamp without time zone,  
ultimo\_usuario integer,  
desviacion\_ninguna character varying(2) DEFAULT 'N'::character varying,  
contacto\_en\_maxima text,

```

interferencia_derecha text,
interferencia_izquierda text,
interferencia_protrusivo text,
contacto_prematuro text,
CONSTRAINT examen_occlusion_pkey PRIMARY KEY (id_occlusion),
CONSTRAINT examen_occlusion_consulta_id_fkey FOREIGN KEY (consulta_id)
REFERENCES public.consulta (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT examen_occlusion_paciente_id_fkey FOREIGN KEY (nro_historia)
REFERENCES public.paciente (nro_historias) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT examen_occlusion_profesor_fkey FOREIGN KEY (profesor)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT examen_occlusion_ultimo_usuario_fkey FOREIGN KEY (ultimo_usuario)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE public.examen_occlusion
OWNER TO postgres;

CREATE TABLE public.familiar_paciente
(
id_familiar_paciente integer NOT NULL DEFAULT
nextval('familiar_paciente_id_familiar_paciente_seq'::regclass),
nombre character varying(20),
parentesco character varying(20),
direccion character varying(200),

```

```

telefono character varying(15),
CONSTRAINT familiar_paciente_pkey PRIMARY KEY (id_familiar_paciente)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.familiar_paciente
    OWNER TO postgres;

```

```

CREATE TABLE public.historia_odontologica
(
    id_historia_odontologica          integer          NOT          NULL          DEFAULT
nextval('historia_odontologica_id_historia_odontologica_seq'::regclass),
tratamiento_odontologico character varying(5) DEFAULT 'N'::character varying,
afectando_salud character varying(5) DEFAULT 'N'::character varying,
apariencia_dientes character varying(5) DEFAULT 'N'::character varying,
nervios_tratamiento character varying(5) DEFAULT 'N'::character varying,
reaccion_anestesico character varying(5) DEFAULT 'N'::character varying,
dolor_reciente_dentadura character varying(5) DEFAULT 'N'::character varying,
sangra_encia character varying(5) DEFAULT 'N'::character varying,
control_dental_reg character varying(150) DEFAULT 'N'::character varying,
ultima_visita_odo character varying(50),
cepillo_dental character varying(5) DEFAULT 'N'::character varying,
hilo_dental character varying(5) DEFAULT 'N'::character varying,
espec_control_dental character varying(200),
espec_tratamiento_odon character varying(200),
frecuencia_cepillado character varying(200),
tipo_cepillo character varying(20),
frecuencia_hilo_dental character varying(200),
otros_medios character varying(5) DEFAULT 'N'::character varying,
cuales_medios character varying(200),

```

dieta\_habitual character varying(200),  
alimento\_azucar character varying(5) DEFAULT 'N'::character varying,  
frecu\_aliment\_azucar character varying(200),  
cirugia\_cavidad character varying(5) DEFAULT 'N'::character varying,  
espec\_cirugia character varying(200),  
exodoncia character varying(5) DEFAULT 'N'::character varying,  
espec\_exodoncia character varying(200),  
complic\_pos\_exo character varying(5) DEFAULT 'N'::character varying,  
tipo\_complicacion character varying(50) DEFAULT 'N'::character varying,  
ortodoncia character varying(5) DEFAULT 'N'::character varying,  
cuando\_ortodoncia character varying(100),  
endodoncia character varying(5) DEFAULT 'N'::character varying,  
cuando\_endodoncia character varying(50) DEFAULT 'N'::character varying,  
cuales\_dientes\_endodoncia character varying(50) DEFAULT 'N'::character varying,  
periodoncia character varying(5) DEFAULT 'N'::character varying,  
tipo\_tratamiento character varying(50),  
protesis\_dental character varying(5) DEFAULT 'N'::character varying,  
mastica\_alimentos\_blandos character varying(5) DEFAULT 'N'::character varying,  
mastica\_alimentos\_duros character varying(5) DEFAULT 'N'::character varying,  
risa\_sonrisa character varying(5) DEFAULT 'N'::character varying,  
satisfecho\_tratamiento character varying(5) DEFAULT 'N'::character varying,  
mastica\_solo\_lado character varying(5) DEFAULT 'N'::character varying,  
sensibilidad\_dental character varying(5) DEFAULT 'N'::character varying,  
dolor\_atm\_derecha character varying(5) DEFAULT 'N'::character varying,  
dolor\_atm\_izquierda character varying(5) DEFAULT 'N'::character varying,  
ausencia\_dientes character varying(5) DEFAULT 'N'::character varying,  
otras\_causas character varying(5) DEFAULT 'N'::character varying,  
dificultad\_boca character varying(5) DEFAULT 'N'::character varying,  
rechina\_diente character varying(50) DEFAULT 'N'::character varying,  
razon\_mastica character varying(50) DEFAULT 'N'::character varying,  
articulacion\_temporomandibular character varying(5) DEFAULT 'N'::character varying,

abrir\_boca character varying(5) DEFAULT 'N'::character varying,  
 cerrar\_boca character varying(5) DEFAULT 'N'::character varying,  
 masticar character varying(5) DEFAULT 'N'::character varying,  
 bostezar character varying(5) DEFAULT 'N'::character varying,  
 reir character varying(5) DEFAULT 'N'::character varying,  
 dolor\_cabeza character varying(5) DEFAULT 'N'::character varying,  
 frecuencia\_dolor\_cabeza character varying(50) DEFAULT 'N'::character varying,  
 ciclo\_dolor\_cabeza character varying(50) DEFAULT 'N'::character varying,  
 morderse\_uyas character varying(5) DEFAULT 'N'::character varying,  
 morderse\_labios character varying(5) DEFAULT 'N'::character varying,  
 droga character varying(5) DEFAULT 'N'::character varying,  
 morder\_fosforos character varying(5) DEFAULT 'N'::character varying,  
 respirar\_boca character varying(5) DEFAULT 'N'::character varying,  
 tabaco character varying(5) DEFAULT 'N'::character varying,  
 abre\_objeto\_dientes character varying(5) DEFAULT 'N'::character varying,  
 lengua\_contra\_dientes character varying(5) DEFAULT 'N'::character varying,  
 bebida\_alcoholica character varying(5) DEFAULT 'N'::character varying,  
 ultimo\_usuario integer,  
 nro\_historia integer,  
 consulta\_id integer,  
 profesor integer,  
 fecha date,  
 condicion\_sexual\_id integer,  
 cuando\_articulacion character varying(50) DEFAULT 'N'::character varying,  
 fecha\_tratamiento character varying(20),  
 validar character varying(15),  
 tipo\_protesis character varying(30),  
 created\_at timestamp without time zone,  
 updated\_at timestamp without time zone,  
 fecha\_validacion character varying(50),  
 CONSTRAINT historia\_odontologica\_pkey PRIMARY KEY (id\_historia\_odontologica),

```

CONSTRAINT condicion_sexual_id FOREIGN KEY (condicion_sexual_id)
  REFERENCES public.valores_listas (id_valor) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT consulta_id FOREIGN KEY (consulta_id)
  REFERENCES public.consulta (id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT paciente_id FOREIGN KEY (nro_historia)
  REFERENCES public.paciente (nro_historias) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT profesor_id FOREIGN KEY (profesor)
  REFERENCES public.usuarios (id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT ultimo_usuario FOREIGN KEY (ultimo_usuario)
  REFERENCES public.usuarios (id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.historia_odontologica
  OWNER TO postgres;

CREATE TABLE public.listas
(
  id_lista integer NOT NULL DEFAULT nextval('listas_id_lista_seq'::regclass),
  lista character varying(45),
  CONSTRAINT pk_id_lista PRIMARY KEY (id_lista)
)
WITH (
  OIDS=FALSE
);

```

```
ALTER TABLE public.listas
```

```
OWNER TO postgres;
```

```
CREATE TABLE public.modelo_diagnostico
```

```
(
```

```
id_modelo_diagnostico integer NOT NULL DEFAULT  
nextval('modelo_diagnostico_id_modelo_diagnostico_seq'::regclass),  
forma_su character varying(50),  
simetria_su character varying(50),  
forma_inf character varying(50),  
simetria_inf character varying(50),  
dientes_ausentes character varying(5),  
grado character varying(50),  
kennedy_su character varying(50),  
kennedy_inf character varying(50),  
corona_tam character varying(15),  
corona_forma character varying(15),  
giroversiones character varying(15),  
migraciones character varying(15),  
extrusiones character varying,  
inclinados character varying,  
faceta_desgaste character varying(15),  
rebordes character varying(15),  
cuspidate character varying(15),  
contacto_prematuro character varying(15),  
relaciones_bordes character varying(15),  
espacio_disponible character varying(50),  
espacio_apoyo character varying(50),  
profesor integer,  
ultimo_usuario integer,  
fecha date,
```

```

validar character varying(5),
nro_historia integer,
consulta_id integer,
modif_su character varying(15),
modif_inf character varying(15),
created_at timestamp without time zone,
updated_at timestamp without time zone,
fecha_validacion character varying(50),
CONSTRAINT modelo_diagnostico_pkey PRIMARY KEY (id_modelo_diagnostico),
CONSTRAINT modelo_diagnostico_paciente_id_fkey FOREIGN KEY (nro_historia)
REFERENCES public.paciente (nro_historias) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT modelo_diagnostico_profesor_fkey FOREIGN KEY (profesor)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT modelo_diagnostico_ultimo_usuario_fkey FOREIGN KEY (ultimo_usuario)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE public.modelo_diagnostico
OWNER TO postgres;

CREATE TABLE public.odontograma
(
id_odontograma integer NOT NULL DEFAULT
nextval('odontograma_id_odontograma_seq'::regclass),
nro_historia integer,
consulta_id integer,

```

```

elemento character varying(50),
posicion_y character varying(20),
posicion_x character varying(20),
ultimo_usuario integer,
created_at timestamp without time zone,
updated_at timestamp without time zone,
fecha_validacion character varying(50),
validar character varying(15),
profesor integer,
fecha character varying(50),
CONSTRAINT odontograma_pkey PRIMARY KEY (id_odontograma),
CONSTRAINT odontograma_nro_historia_fkey FOREIGN KEY (nro_historia)
REFERENCES public.paciente (nro_historias) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.odontograma
OWNER TO postgres;

```

```

CREATE TABLE public.operatoria
(
  cavidades character varying(5) DEFAULT 'N'::character varying,
  lesiones_radiograficas character varying(2) DEFAULT 'N'::character varying,
  lesiones_mancha character varying(2) DEFAULT 'N'::character varying,
  restauraciones character varying(2) DEFAULT 'N'::character varying,
  recuento character varying(2) DEFAULT 'N'::character varying,
  abundante character varying(2) DEFAULT 'N'::character varying,
  bocadillos character varying(2) DEFAULT 'N'::character varying,
  fosas character varying(2) DEFAULT 'N'::character varying,

```

drogas character varying(2) DEFAULT 'N'::character varying,  
 flujo\_salival character varying(2) DEFAULT 'N'::character varying,  
 factores\_reduccion character varying(2) DEFAULT 'N'::character varying,  
 raices\_expuestas character varying(2) DEFAULT 'N'::character varying,  
 dentrifrico\_una character varying(2) DEFAULT 'N'::character varying,  
 dentrifrico\_dos character varying(5) DEFAULT 'N'::character varying,  
 enjuague\_bucal character varying(5) DEFAULT 'N'::character varying,  
 dentrifrico\_de\_calcio character varying(5) DEFAULT 'N'::character varying,  
 adecuado\_flujo character varying(5) DEFAULT 'N'::character varying,  
 flujo\_salival\_ml character varying(200),  
 riesgo\_caries character varying(200),  
 consulta\_id integer,  
 validar character varying(20),  
 ultimo\_usuario integer,  
 profesor integer,  
 nro\_historia integer,  
 fecha date,  
 id\_operatoria integer NOT NULL DEFAULT nextval('operatoria\_id\_operatoria\_seq'::regclass),  
 fecha\_validacion character varying(15),  
 created\_at timestamp with time zone,  
 updated\_at timestamp without time zone,  
 aparatologia character varying(2) DEFAULT 'N'::character varying,  
 vive\_trabaja character varying(2) DEFAULT 'N'::character varying,  
 dentrifrico\_fluorado character varying(2) DEFAULT 'N'::character varying,  
 barniz character varying(2) DEFAULT 'N'::character varying,  
 clorhexidina character varying(2) DEFAULT 'N'::character varying,  
 goma\_mascar character varying(2) DEFAULT 'N'::character varying,  
 CONSTRAINT operatoria\_pkey PRIMARY KEY (id\_operatoria),  
 CONSTRAINT operatoria\_consulta\_id\_fkey FOREIGN KEY (consulta\_id)  
 REFERENCES public.consulta (id) MATCH SIMPLE  
 ON UPDATE NO ACTION ON DELETE NO ACTION,

```

CONSTRAINT operatoria_paciente_id_fkey FOREIGN KEY (nro_historia)
  REFERENCES public.paciente (nro_historias) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT operatoria_profesor_fkey FOREIGN KEY (profesor)
  REFERENCES public.usuarios (id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT operatoria_ultimo_usuario_fkey FOREIGN KEY (ultimo_usuario)
  REFERENCES public.usuarios (id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.operatoria
  OWNER TO postgres;

CREATE TABLE public.paciente
(
  nro_historias integer NOT NULL DEFAULT nextval('paciente_id_paciente_seq'::regclass),
  fecha_ingreso date NOT NULL,
  ci character varying(20),
  ultimo_usuario integer,
  created_at character varying(300),
  updated_at character varying(300),
  remember_token character varying(300),
  id_pacientes integer NOT NULL DEFAULT nextval('paciente_nro_historia_seq'::regclass),
  grupo_sanguineo character varying(15),
  familiar_cercano character varying(150),
  parentesco character varying(100),
  direccion_familiar character varying(150),
  telefono_familiar character varying(16),

```

```

nivel_educacional integer,
lee_escribe integer,
zona_residencia integer,
convive integer,
situacion_laboral integer,
lugar_nacimiento character varying(50),
ocupacion character varying(50),
estado_civil character varying(15),
persona_id integer,
familiar_paciente_id integer,
CONSTRAINT paciente_pkey PRIMARY KEY (nro_historias),
CONSTRAINT paciente_ultimo_usuario_fkey FOREIGN KEY (ultimo_usuario)
    REFERENCES public.usuarios (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT persona_id FOREIGN KEY (persona_id)
    REFERENCES public.persona (id_persona) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.paciente
    OWNER TO postgres;

CREATE TABLE public.paciente_enfer_cardiovascular
(
    id_paciente_enfer_cardiovascular          integer          NOT          NULL          DEFAULT
nextval('paciente_enfer_cardiovascular_id_paciente_enfer_cardiovascu_seq'::regclass),
    paciente_id integer,
    enfermedad_cardiovascular_id integer,
    circulo_hereditario integer, --

```

```

consulta_id          integer          NOT          NULL          DEFAULT
nextval('paciente_enfer_cardiovascular_consulta_id_seq'::regclass),
ultimo_usuario integer,
antecedente character varying(15),
created_at timestamp without time zone,
updated_at timestamp without time zone,
CONSTRAINT id_enfer PRIMARY KEY (id_paciente_enfer_cardiovascular),
CONSTRAINT circulo_id_car FOREIGN KEY (circulo_hereditario)
REFERENCES public.valores_listas (id_valor) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT paciente_enfer_cardiovascular_consulta_id_fkey FOREIGN KEY (consulta_id)
REFERENCES public.consulta (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT paciente_enfer_cardiovascular_ultimo_usuario_fkey FOREIGN KEY
(ultimo_usuario)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT paciente_enfer_cardiovascular_ultimo_usuario_fkey1 FOREIGN KEY
(ultimo_usuario)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT paciente_id FOREIGN KEY (paciente_id)
REFERENCES public.paciente (id_paciente) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE public.paciente_enfer_cardiovascular
OWNER TO postgres;
COMMENT ON COLUMN public.paciente_enfer_cardiovascular.circulo_hereditario IS '

```

;

```
CREATE TABLE public.paciente_enfer_patologicas
(
  id_paciente_enfer_patologica          integer          NOT          NULL          DEFAULT
  nextval('paciente_enfer_patologicas_id_paciente_enfer_patologica_seq'::regclass),
  paciente_id integer,
  enfermedad_patologica_id integer,
  consulta_id integer,
  circulo_familiar_id integer,
  ultimo_usuario integer,
  antecedente character varying(15),
  created_at timestamp without time zone,
  updated_at timestamp without time zone,
  CONSTRAINT id_paciente_enfer_patolo PRIMARY KEY (id_paciente_enfer_patologica),
  CONSTRAINT circulo_id FOREIGN KEY (circulo_familiar_id)
    REFERENCES public.valores_listas (id_valor) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT consulta_id FOREIGN KEY (consulta_id)
    REFERENCES public.consulta (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT enfer_patolo_id FOREIGN KEY (enfermedad_patologica_id)
    REFERENCES public.enfermedades_patologicas (id_enfermedad_patologica) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT paciente_enfer_patologicas_ultimo_usuario_fkey FOREIGN KEY (ultimo_usuario)
    REFERENCES public.usuarios (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT paciente_id FOREIGN KEY (paciente_id)
    REFERENCES public.paciente (id_paciente) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
```

```

)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.paciente_enfer_patologicas
    OWNER TO postgres;

CREATE TABLE public.paciente_enfer_renal
(
    paciente_id integer,
    enfermedad_renal_id integer,
    circulo_hereditario integer, --
    id_paciente_enfer_renal          integer          NOT          NULL          DEFAULT
nextval('paciente_enfer_renal_id_paciente_enfer_renal_seq'::regclass),
    ultimo_usuario integer,
    consulta_id integer,
    antecedente character varying(15),
    created_at timestamp without time zone,
    updated_at timestamp without time zone,
    CONSTRAINT id_enfer_renal PRIMARY KEY (id_paciente_enfer_renal),
    CONSTRAINT circulo_id_renal FOREIGN KEY (circulo_hereditario)
        REFERENCES public.valores_listas (id_valor) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT paciente_enfer_renal_consulta_id_fkey FOREIGN KEY (consulta_id)
        REFERENCES public.consulta (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT paciente_enfer_renal_ultimo_usuario_fkey FOREIGN KEY (ultimo_usuario)
        REFERENCES public.usuarios (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT paciente_id FOREIGN KEY (paciente_id)
        REFERENCES public.paciente (nro_historias) MATCH SIMPLE

```

```

        ON UPDATE NO ACTION ON DELETE NO ACTION
    )
    WITH (
        OIDS=FALSE
    );
ALTER TABLE public.paciente_enfer_renal
    OWNER TO postgres;
COMMENT ON COLUMN public.paciente_enfer_renal.circulo_hereditario IS '
';

```

```

CREATE TABLE public.parciales
(
    clasificacion_kennedy character varying(200),
    posicion_h character varying(50),
    posicion_a character varying(50),
    posicion_p character varying(50),
    posicion_ld character varying(50),
    posicion_li character varying(50),
    dientes_pilares character varying(50),
    localizacion_retenciones character varying(200),
    retenedores_directos character varying(200),
    apoyos character varying(200),
    retenedores_indirectos character varying(200),
    planos_guias character varying(200),
    c_mayor character varying(200),
    bases character varying(200),
    imagen text,
    observaciones character varying(200),
    consulta_id integer,
    validar character varying(20),

```

```

ultimo_usuario integer,
profesor integer,
nro_historia integer,
fecha date,
orientacion character varying(15),
id_parciales integer NOT NULL DEFAULT nextval('parciales_id_parciales_seq'::regclass),
created_at timestamp without time zone,
updated_at timestamp without time zone,
fecha_validacion character varying(15),
CONSTRAINT parciales_pkey PRIMARY KEY (id_parciales),
CONSTRAINT parciales_consulta_id_fkey FOREIGN KEY (consulta_id)
REFERENCES public.consulta (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT parciales_paciente_id_fkey FOREIGN KEY (nro_historia)
REFERENCES public.paciente (nro_historias) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT parciales_profesor_fkey FOREIGN KEY (profesor)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT parciales_ultimo_usuario_fkey FOREIGN KEY (ultimo_usuario)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.parciales
OWNER TO postgres;

CREATE TABLE public.persona
(

```

```

ci character varying(20) NOT NULL,
ultimo_usuario integer,
nombre character varying(50),
apellido character varying(50),
fecha_nacimiento date,
genero character varying(16),
telefono character varying(16),
celular character varying(16),
direccion character varying(300),
created_at character varying(300),
updated_at character varying(300),
remember_token character varying(300),
id_persona integer NOT NULL DEFAULT nextval('persona_id_persona_seq'::regclass),
CONSTRAINT persona_pkey PRIMARY KEY (id_persona),
CONSTRAINT persona_ultimo_usuario_fkey FOREIGN KEY (ultimo_usuario)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.persona
OWNER TO postgres;

CREATE TABLE public.pronostico
(
  fecha date,
  consulta_id integer,
  nro_historia integer,
  validar character varying(15),
  fecha_validacion character varying(15),

```

```

profesor integer,
ultimo_usuario integer,
especialidad character varying(15),
id_pronostico integer NOT NULL DEFAULT nextval('pronostico_id_pronostico_seq'::regclass),
pronostico character varying(200),
CONSTRAINT pronostico_pkey PRIMARY KEY (id_pronostico)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.pronostico
  OWNER TO postgres;

```

```

CREATE TABLE public.registros_imageneologicos
(
  id_registro_imageneologico integer NOT NULL DEFAULT
  nextval('registros_imageneologicos_id_registro_imageneologico_seq'::regclass),
  consulta_id integer,
  nro_historia integer,
  fecha timestamp without time zone,
  ultimo_usuario integer,
  created_at timestamp without time zone,
  updated_at timestamp without time zone,
  ordenado_por character varying(50),
  motivo_examen character varying(100),
  panoramica character varying(2) DEFAULT 'N'::character varying,
  apta_interpretar character varying(2) DEFAULT 'N'::character varying,
  defecto_imagen character varying(100),
  radiolucida character varying(200),
  radiopaca character varying(200),
  mixta character varying(200),

```

seno\_maxilar\_lado character varying(25),  
seno\_maxilar\_borde character varying(25),  
seno\_maxilar\_forma character varying(25),  
cuello\_condilo character varying(25),  
fosa\_nasal\_lado character varying(25),  
fosa\_nasal\_borde character varying(25),  
fosa\_nasal\_forma character varying(25),  
arco\_lado character varying(25),  
discontinuidad character varying(2) DEFAULT 'N'::character varying,  
presencia\_radiopacidad character varying(50),  
espacio\_aereo\_lado character varying(50),  
espacio\_aereo\_bordes character varying(50),  
espacio\_aereo\_forma character varying(50),  
espacio\_aereo\_numero character varying(50),  
espacio\_aereo\_ubicacion character varying(50),  
anomalia\_tam character varying(50),  
espec\_anomalia\_tam character varying(100),  
supernumerarios character varying(150),  
taurodontismo character varying(150),  
hipercementosis character varying(150),  
otras character varying(150),  
radiolucides\_coronal character varying(200),  
region\_apical character varying(200),  
cresta\_alveolar character varying(150),  
dientes\_inclinados character varying(100),  
espacios\_abiertos character varying(150),  
hallazgos\_adicionales character varying(200),  
uno\_dos character varying(150),  
uno\_uno character varying(150),  
dos\_uno character varying(150),  
interpretacion character varying(200),

```

recomendaciones character varying(200),
profesor integer,
validar character varying(15),
fecha_validacion character varying(15),
periapical character varying(2) DEFAULT 'N'::character varying,
interproximal character varying(2) DEFAULT 'N'::character varying,
otro character varying(2) DEFAULT 'N'::character varying,
espec_otro character varying(50),
alta_densidad character varying(2) DEFAULT 'N'::character varying,
baja_densidad character varying(2) DEFAULT 'N'::character varying,
borrosa character varying(2) DEFAULT 'N'::character varying,
distorsion character varying(2) DEFAULT 'N'::character varying,
defecto character varying(2) DEFAULT 'N'::character varying,
imagen_ppal text,
CONSTRAINT registros_imageneologicos_pkey PRIMARY KEY (id_registro_imageneologico),
CONSTRAINT registros_imageneologicos_consulta_id_fkey FOREIGN KEY (consulta_id)
REFERENCES public.consulta (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT registros_imageneologicos_paciente_id_fkey FOREIGN KEY (nro_historia)
REFERENCES public.paciente (nro_historias) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT registros_imageneologicos_usuario_id_fkey FOREIGN KEY (ultimo_usuario)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE public.registros_imageneologicos
OWNER TO postgres;

```

```

CREATE TABLE public.resumen_historia_medica
(
  id_resumen_historia_medica      integer      NOT      NULL      DEFAULT
  nextval('resumen_historia_medica_id_resumen_historia_medica_seq'::regclass),
  fecha date,
  consulta_id integer,
  nro_historia integer,
  ultimo_usuario integer,
  validar character varying(20),
  profesor integer,
  rhm_1 character varying(350),
  created_at timestamp without time zone,
  updated_at timestamp without time zone,
  fecha_validacion character varying(50),
  CONSTRAINT resumen_historia_medica_pkey PRIMARY KEY (id_resumen_historia_medica),
  CONSTRAINT consulta_id FOREIGN KEY (consulta_id)
    REFERENCES public.consulta (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT paciente_id FOREIGN KEY (nro_historia)
    REFERENCES public.paciente (nro_historias) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT profesor_id FOREIGN KEY (profesor)
    REFERENCES public.usuarios (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT ultimo_usuario FOREIGN KEY (ultimo_usuario)
    REFERENCES public.usuarios (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);

```

```

ALTER TABLE public.resumen_historia_medica
  OWNER TO postgres;
CREATE TABLE public.resumen_historia_odontologica
(
  id_resumen_historia_odontologica integer NOT NULL DEFAULT
nextval('resumen_historia_odontologica_id_resumen_historia_odontolog_seq'::regclass),
  rho_1 character varying(200),
  consulta_id integer,
  nro_historia integer,
  ultimo_usuario integer,
  profesor integer,
  validar character varying(20),
  fecha date,
  created_at timestamp without time zone,
  updated_at timestamp without time zone,
  fecha_validacion character varying(50),
  CONSTRAINT id_ PRIMARY KEY (id_resumen_historia_odontologica),
  CONSTRAINT consulta_id FOREIGN KEY (consulta_id)
  REFERENCES public.consulta (id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT paciente_id FOREIGN KEY (nro_historia)
  REFERENCES public.paciente (nro_historias) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT profesor_id FOREIGN KEY (profesor)
  REFERENCES public.usuarios (id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT ultimo_usuario FOREIGN KEY (ultimo_usuario)
  REFERENCES public.usuarios (id) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (

```

```

        OIDS=FALSE
    );
ALTER TABLE public.resumen_historia_odontologica
    OWNER TO postgres;

CREATE TABLE public.roles
(
    id_rol integer NOT NULL DEFAULT nextval('roles_id_rol_seq'::regclass),
    rol_name character varying(45) NOT NULL,
    nivel_acceso character varying(11),
    ultimo_usuario integer DEFAULT nextval('roles_ultimo_usuario_seq'::regclass),
    codigo character varying(25),
    CONSTRAINT pk_id_rol PRIMARY KEY (id_rol)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.roles
    OWNER TO postgres;

CREATE TABLE public.signos_vitales
(
    id_signo_vital integer NOT NULL DEFAULT nextval('signos_vitales_id_signo_vital_seq'::regclass),
    presion_sanguinea character varying(15),
    pulso character varying(15),
    temperatura character varying(15),
    frecuencia_respiratoria character varying(15),
    validar character varying(20),
    consulta_id integer,
    nro_historia integer,
    ultimo_usuario integer,

```

```

profesor integer,
fecha date,
created_at timestamp without time zone,
updated_at timestamp without time zone,
fecha_validacion character varying(50),
CONSTRAINT id_signo_vital PRIMARY KEY (id_signo_vital),
CONSTRAINT consulta_id FOREIGN KEY (consulta_id)
    REFERENCES public.consulta (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT paciente_id FOREIGN KEY (nro_historia)
    REFERENCES public.paciente (nro_historias) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT profesor_id FOREIGN KEY (profesor)
    REFERENCES public.usuarios (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT ultimo_usuario FOREIGN KEY (ultimo_usuario)
    REFERENCES public.usuarios (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.signos_vitales
    OWNER TO postgres;

CREATE TABLE public.test_fagerstrom
(
    id_test integer NOT NULL DEFAULT nextval('test_fagerstrom_id_test_seq'::regclass),
    primer_cigarrillo character varying(50),
    fumar_en_lugares character varying(2),
    molesta_mas character varying(50),

```

```

cigarro_por_dia character varying(15),
fuma_frecuencia character varying(2),
fuma_enfermo character varying(2),
total character varying(15),
validar character varying(15),
profesor integer,
ultimo_usuario integer,
fecha date,
nro_historia integer,
consulta_id integer,
created_at timestamp without time zone,
updated_at timestamp without time zone,
fecha_validacion character varying(50),
CONSTRAINT test_fagerstrom_pkey PRIMARY KEY (id_test),
CONSTRAINT test_fagerstrom_consulta_id_fkey FOREIGN KEY (consulta_id)
REFERENCES public.consulta (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT test_fagerstrom_paciente_id_fkey FOREIGN KEY (nro_historia)
REFERENCES public.paciente (nro_historias) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT test_fagerstrom_profesor_fkey FOREIGN KEY (profesor)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT test_fagerstrom_ultimo_usuario_fkey FOREIGN KEY (ultimo_usuario)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.test_fagerstrom

```

```
OWNER TO postgres;
```

```
CREATE TABLE public.tratamiento
```

```
(  
  id_tratamiento integer NOT NULL DEFAULT nextval('tratamiento_id_tratamiento_seq'::regclass),  
  fecha_tratamiento character varying(15),  
  tratamiento_realizado character varying(200),  
  tratamiento_id integer,  
  CONSTRAINT tratamiento_pkey PRIMARY KEY (id_tratamiento),  
  CONSTRAINT tratamiento_tratamiento_id_fkey FOREIGN KEY (tratamiento_id)  
    REFERENCES public.tratamientos (id_tratamientos) MATCH SIMPLE  
    ON UPDATE NO ACTION ON DELETE NO ACTION  
)
```

```
WITH (
```

```
  OIDS=FALSE
```

```
);
```

```
ALTER TABLE public.tratamiento
```

```
  OWNER TO postgres;
```

```
CREATE TABLE public.tratamientos
```

```
(  
  id_tratamientos integer NOT NULL DEFAULT  
nextval('tratamientos_id_tratamientos_seq'::regclass),  
  fecha date,  
  consulta_id integer,  
  nro_historia integer,  
  validar character varying(15),  
  fecha_validacion character varying(15),  
  profesor integer,  
  ultimo_usuario integer,  
  especialidad character varying(15),
```

```

CONSTRAINT tratamientos_pkey PRIMARY KEY (id_tratamientos),
CONSTRAINT tratamientos_consulta_id_fkey FOREIGN KEY (consulta_id)
REFERENCES public.consulta (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT tratamientos_paciente_id_fkey FOREIGN KEY (nro_historia)
REFERENCES public.paciente (nro_historias) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT tratamientos_profesor_fkey FOREIGN KEY (profesor)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT tratamientos_ultimo_usuario_fkey FOREIGN KEY (ultimo_usuario)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.tratamientos
OWNER TO postgres;

CREATE TABLE public.usuarios
(
  id integer NOT NULL DEFAULT nextval('usuario_id_usuario_seq'::regclass),
  email character varying(120) NOT NULL,
  password character varying(300),
  remember_token character varying(300),
  created_at character varying(300),
  updated_at character varying(300),
  rol_id integer,
  valor_id integer,
  ultimo_usuario integer,

```

```

ci character varying(20),
persona_id integer,
CONSTRAINT usuario_pkey PRIMARY KEY (id),
CONSTRAINT persona_id FOREIGN KEY (persona_id)
REFERENCES public.persona (id_persona) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT usuarios_rol_id_fkey FOREIGN KEY (rol_id)
REFERENCES public.roles (id_rol) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT usuarios_ultimo_usuario_fkey FOREIGN KEY (ultimo_usuario)
REFERENCES public.usuarios (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT usuarios_valor_id_fkey FOREIGN KEY (valor_id)
REFERENCES public.valores_listas (id_valor) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE public.usuarios
OWNER TO postgres;

```

```

CREATE TABLE public.valores_listas
(
id_valor integer NOT NULL DEFAULT nextval('valores_listas_id_valor_seq'::regclass),
lista_id integer NOT NULL DEFAULT nextval('valores_listas_lista_id_seq'::regclass),
valor character varying(45),
codigo character varying(45),
CONSTRAINT pk_id_valor_lista PRIMARY KEY (id_valor),
CONSTRAINT fk_lista_id FOREIGN KEY (lista_id)
REFERENCES public.listas (id_lista) MATCH SIMPLE

```

```
        ON UPDATE NO ACTION ON DELETE NO ACTION
    )
    WITH (
        OIDS=FALSE
    );
ALTER TABLE public.valores_listas
    OWNER TO postgres;
```

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)