



**Región de
Murcia**

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

Diseño e implementación de una aplicación para gestionar una clínica dental

**Proyecto Fin de Grado de Desarrollo de
Aplicaciones Multiplataforma**

Autor: Francisco José Verdú Berdejo
Tutor Académico: José María García Nieto
Centro: CIFP Carlos III

Junio de 2021



**Región de
Murcia**

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)



Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

ÍNDICE

1.- Introducción y Objetivo del proyecto

1.1- Introducción

1.1- Objetivo del proyecto

2.- Herramientas y tecnología

2.1- Lenguajes

2.1.1.- Java

2.1.2.- XML

2.1.3.- CSS

2.1.4.- FXML

2.2- Frameworks

2.2.1.- JavaFX

2.2.2.- Hibernate

2.2.3.- JasperReports

2.3.- Otras herramientas

2.3.1.- Eclipse (IDE)

2.3.2.- PgAdmin

2.3.3.- Laragon

2.3.4.- PostgreSQL

2.3.5.- Jaspersoft Studio

2.3.6.- Maven

2.3.7.- JDBC

2.3.7.- Scene Builder

3.- Análisis funcional

3.1.- Documento de Requisitos

3.2.- Diseño de base de datos

3.2.1.- Resumen

3.2.2.- Procedimientos almacenados (stored procedures)

3.2.3.- Triggers (disparadores)

3.2.4.- Vistas (PostgreSQL)

4.- Arquitectura del sistema

4.1- Patrón MVC

4.2.- Estructura del proyecto

5.- Interfaces de usuario

6.- Implementación de la aplicación

7.- Resultado final

8.- Conclusiones

8.1.- Conclusiones

8.2.- Trabajo futuro

9.- Bibliografía

10.- Anexos

10.1.- Anexo A. Instalación Laragon



Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

10.2.- Anexo B. Modelo de base de datos ER (Imagen y modelo .uml adjunto al proyecto)

10.3.- Anexo C. Estructura de Vistas JavaFX (Imagen y modelo .uml adjunto al proyecto)

10.4.- Anexo D. Jerarquía de modelos principales (Imagen y fichero .uml adjunto al proyecto)

10.5.- Anexo E. Jerarquía de controladores creados (Imagen y fichero .uml adjunto al proyecto)

10.6.- Anexo F. Jerarquía de informes creados (Imagen y fichero .uml adjunto al proyecto)

10.7.- Anexo G. Jerarquía del paquete de utilidades creado (Imagen y fichero .uml adjunto al proyecto)

10.8.- Anexo H. Manual de Usuario (Se adjunta en el proyecto un documento PDF que será el manual del usuario)



1. Introducción

1.1 Planteamiento del proyecto

Hoy en día, es indispensable el uso de un software que ayude a gestionar la información. El acceso a la información debe ser rápida, coherente y sin errores.

El objetivo principal del proyecto es el desarrollo de una aplicación de escritorio que facilite la gestión de una clínica dental.

Para ello, haremos uso JavaFX como framework y de otras tecnologías como Java, CSS y PostgreSQL para llevarlo a cabo.

La aplicación permitirá la definición de los artículos y medicamentos que utilizará el usuario, junto a la jerarquía que seguirá, la cual permitirá definir categorías para dichos artículos.

Adicionalmente, también permitirá la definición de los pacientes, médicos y citas como entidades principales para llevar a cabo la gestión de la clínica dental.

En base a todas estas definiciones, el sistema dará acceso al usuario a gestionar las citas de los pacientes e ir construyendo poco a poco el historial clínico del paciente. También permitirá diagnosticar la problemática del paciente y tratarla y posteriormente realizar la facturación del mismo.

Separaremos el proyecto en varias secciones. Primero explicará las distintas tecnologías empleadas para informar al usuario de su uso. Posteriormente se indicará una breve explicación de la estructura que seguirá el código desarrollado. A continuación, hablará del modelo de datos creado que creará los cimientos de la aplicación y finalmente dará paso a explicar el despliegue de la aplicación de escritorio.

1.2. Objetivos del proyecto

Los objetivos del proyecto son:

- Aprender sobre las tecnologías desktop empleadas: Java, CSS, PostgreSQL, XML, FXML, JavaFx.
- Ser capaz de desarrollar una aplicación desktop desde cero, empezando desde su diseño, desarrollo y posterior implementación.



2. Herramientas y tecnologías

2.1. Lenguajes

2.1.1. Java

Java es un lenguaje de programación orientado a objetos que fue comercializado por primera vez por Sun microsystems en 1995. Permite crear aplicaciones que pueden ser utilizadas independientemente de su entorno operativo y hardware. Su sintaxis deriva de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos.

Las aplicaciones de Java son compiladas a bytecode (clase Java), que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura del ordenador donde se utilice.

Es una tecnología muy utilizada y conocida, desde computadores hasta juegos o dispositivos móviles.

2.1.2. XML

XML es el acrónimo de Extensible Markup Language. Se trata de un metalenguaje que se utiliza para decir algo acerca de otro. Es un formato simple basado en texto para representar la información de manera estructurada.

Es sencillo de estructurar, facilitando diferenciar las distintas partes del documento. Permite importarse y exportarse a otras aplicaciones, programas y formatos.

Puede ser utilizado para estructurar bases de datos, editores de texto u hojas de cálculo. La versión 1.0 es la utilizada para este proyecto.

2.1.3. CSS

CSS (Cascading Style Sheets) es un lenguaje utilizado para desarrollar y definir el estilo de la interfaz visual del usuario.

Inicialmente, CSS se diseñó para estructurar mejor el desarrollo de páginas web, pudiendo separar el contenido del documento de los estilos del documento.

CSS cuenta con su propia sintaxis, y se basa en herencia en cascada, es decir, si el elemento padre tiene definidas unas propiedades, los hijos también heredarán estas propiedades.

Con CSS podríamos crear sombras, degradados, modificar colores de fuentes, etc.



2.1.4. Fxml

Fxml es un lenguaje de interfaz de usuario basado en XML. Fue creado por Oracle Corporation para definir la interfaz de usuario de una aplicación JavaFX.

Presenta una alternativa al diseño de interfaces de usuario utilizando código de procedimiento y permite abstraer el diseño del programa de la lógica del programa.

Contiene un gran número de elementos, tanto estáticos como dinámicos, gráficos, objetos 3D. Permite crear interfaces más atractivas y livianas que con Java Swing.

Se suele utilizar con una herramienta que facilita el desarrollo de interfaces visuales, el cual es Scene Builder. Permite crear la interfaz visual del usuario haciendo uso de XML.

2.2. Frameworks y librerías

¿Qué es un framework?

Podemos definir un framework como una estructura de software formada por diferentes componentes que facilitan el desarrollo de una aplicación.

El objetivo de los frameworks es hacer cómodo, elegante y simple el desarrollo de las aplicaciones, sin necesidad de reescribir código.

2.2.1. JavaFX

JavaFX es un framework de código abierto, el cuál facilita desarrollar aplicaciones de una forma simple y vistosa.

Características

JavaFX incluye muchas funcionalidades que facilitan al usuario su trabajo, como puedan ser:

1. Está escrito en Java, por lo que permite aprovechar todas las funcionalidades de Java. Se puede utilizar con cualquier editor de Java.
2. Permite el enlace de datos a través de sus bibliotecas.
3. Puede lograr animaciones interesantes escribiendo pocas líneas de código.
4. Proporciona un amplio conjunto de soporte multimedia, como la reproducción de videos y audios.
5. Ofrece dos formas de crear una interfaz de usuario. O bien haciendo uso de código Java o haciendo uso de FXML.
6. Permite vincular un documento css en el propio fichero FXML.
7. Facilita la utilización de una arquitectura MCV.

En la actualidad, JavaFX es uno de los frameworks más utilizados en Java, contando con una gran comunidad creciente. La versión actual JavaFX, utilizada en este proyecto es la 15.1.



Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

2.2.2. Hibernate

Hibernate es una herramienta de mapeo objeto-relacional (ORM) para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación, mediante archivos XML o anotaciones donde se pueden establecer relaciones.

Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan en la ejecución de dichas sentencias, facilitando la portabilidad de la aplicación a distintas bases de datos.

Ofrece un lenguaje de consultas denominado HQL (Hibernate Query Language), parecido a SQL, pero orientado a objetos e incorpora nociones como herencia, polimorfismo y asociación.

2.2.3. JasperReports

JasperReports es un proyecto open source creado por Teodor Danciu, en el año 2001, como respuesta a una carencia importante de Java.

JasperReports se define como una biblioteca de creación de informes que permite la previsualización e impresión de documentos e inclusive la exportación a formatos tales como PDF, HTML, XLS, CSV y XML.

Está escrito completamente en Java, y puede ser utilizado en gran variedad de aplicaciones Java. Adicionalmente se ha desarrollado un subproyecto que se conforma como un servidor integrado de informes, denominado JasperReports Server.

2.3. Otras herramientas

2.3.1. Eclipse (IDE)

Es una plataforma de software compuesto por un conjunto de herramientas de programación de código abierto y multiplataforma que se utiliza mayormente para el desarrollo de Aplicaciones de Cliente Enriquecido.

Inicialmente fue desarrollado por IBM, pero actualmente está siendo desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro.

Eclipse provee al programador con Frameworks muy ricos para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de datos, diagramas UML, etc.

Es personalizable, gracias a ello, los usuarios pueden cambiar el tema del editor, atajos de teclado, y preferencias. Es gratuito y de código abierto.

Es compatible con varios lenguajes de programación, como C/C++, Python, Ruby Se puede extender a través de plugins y funcionalidades para facilitar el trabajo al desarrollador.



Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

2.2.2. PgAdmin

PgAdmin es una herramienta gráfica open source, para trabajar con PostgreSQL, la base de datos de código abierto más avanzada del mundo y que está disponible en Linux, Solaris, FreeBSD, Mac Os X y Windows.

Proporciona una interfaz muy sencilla de utilizar que integra desarrollo de programación SQL, administración de bases de datos y creación y diseño de bases de datos en un único entorno de desarrollo.

Fue fundado por el programador David Page, quien es el desarrollador principal del proyecto. Comenzó a escribirse en Visual Basic y con el tiempo ha sido desarrollada en distintos lenguajes.

Contiene herramientas visuales con la que se puede crear muchas tareas, entre ellas:

- Ver y crear bases de datos.
- Ver el estado del servidor.
- Administrar usuarios.
- Configurar servidores.
- Crear y ejecutar sentencias SQL.
- Ver y crear otros objetos como vistas, triggers, procedimientos almacenados, etc.

La versión utilizada de PgAdmin en este proyecto es la versión 4.2. En versión Desktop.

2.3.3. Laragon

Laragon es una suite de desarrollo diseñado especialmente para trabajar principalmente con desarrollo web y que funciona sobre Windows.

Me he decantado por la utilización de esta herramienta porque aunque la aplicación es de escritorio, facilita la utilización diversas bases de datos de manera cómodas, y entre ellas PostgreSQL.

La versión utilizada de Laragon en este proyecto es la versión 4.0.

2.3.4. PostgreSQL

PostgreSQL es un sistema de Gestión de Bases de Datos(SGBD) relacional, de código abierto y orientado a objetos.

Es manejado por una comunidad de desarrolladores que trabajan de manera desinteresada. Dicha comunidad es PGDG (PostgreSQL Global Development Group).

2.3.5. Jaspersoft Studio

Es un software de edición de JasperReports. Facilita el diseño y ejecución de plantillas de informes.



Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

Permite escribir expresiones complejas y diseñar componentes visuales de manera sencilla. Con esta herramienta se puede acceder a diferentes tipos de fuentes de datos, incluidos Big Data, CSV, Hibernate, JDBC, JSON NoSQL, XML o la propia fuente de datos personalizada.

Se puede utilizar como complemento de Eclipse o bien como un software instalable aparte.

2.3.6. Maven

Maven es una herramienta de software para la gestión y construcción de proyectos Java creada en 2002 por Jason van Zyl.

Maven utiliza fichero POM (Project Object Model) para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos

Maven puede descargar plugins dinámicamente de un repositorio y también subir código a un repositorio final, dejándolo visible para el resto de usuarios.

Su soporte es principalmente Java. Existen plugins en otros lenguajes, pero mínimos.

La filosofía de Maven es la estandarización de las construcciones generadas por seguir el principio de 'Convención sobre Configuración'.

2.3.7. JDBC

Java Database Connectivity (JDBC) es la especificación de Javasoft de una interfaz de programación de aplicaciones (API) estándar que permite que los programas Java accedan a sistemas de gestión de bases de datos.

Con este tipo de interfaces, se pueden desarrollar aplicaciones que conecten con las bases de datos, se envíen consultas escritas en el lenguaje de consulta estructurada (SQL) y procesen los resultados.

2.3.8. Scene Builder

Scene Builder es un editor de interfaces de JavaFX, simple e intuitivo.

Permite diseñar ficheros FXML arrastrando componentes y ubicándolos en la UI.

Permite separar el diseño de la interfaz de usuario de la lógica.



3. Análisis Funcional

3.1. Documento de Requisitos

Identificador	Requisito
R1	El sistema contará con un menú de herramientas y una botonera con opciones rápidas a los ítems del menú.
R1.1	El usuario debe poder acceder a cualquier opción del menú de herramientas
R1.2	El usuario debe de poder acceder a la botonera que hay en la parte inferior al menú de herramientas
R1.3	El usuario debe de poder visualizar los reportes disponibles.
R1.4	El usuario debe de poder salir de la aplicación.
R2	El sistema contará con la sección Pacientes
R2.1	El usuario podrá crear un paciente
R2.2	El usuario podrá modificar un paciente
R2.3	El usuario podrá eliminar un paciente.
R2.4	El usuario podrá filtrar pacientes
R2.5	El usuario podrá listar todos los pacientes existentes.
R2.6	Si el paciente está siendo utilizado, no se podrá eliminar. Se avisará del motivo.
R3	El sistema contará con la sección Médicos
R3.1	El usuario podrá crear un médico
R3.2	El usuario podrá modificar un médico
R3.3	El usuario podrá eliminar un médico
R3.4	El usuario podrá filtrar médico
R3.5	El usuario podrá listar todos los médicos existentes.
R3.6	Si el médico está siendo utilizado, no se podrá eliminar. Se avisará del motivo.
R4	El sistema contará con la sección Proveedores
R4.1	El usuario podrá crear un proveedor



Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

R4.2	El usuario podrá modificar un proveedor
R4.3	El usuario podrá eliminar un proveedor
R4.4	El usuario podrá filtrar proveedores
R4.5	El usuario podrá listar todos los proveedor existentes.
R4.6	Si el proveedor está siendo utilizado, no se podrá eliminar. Se avisará del motivo.
R5	El sistema contará con la sección Artículos
R5.1	El usuario podrá crear un artículo
R5.2	El usuario podrá modificar un artículo
R5.3	El usuario podrá eliminar un artículo.
R5.4	El usuario podrá filtrar artículos
R5.5	El usuario podrá listar todos los artículos existentes.
R5.6	Si el artículo está siendo utilizado, no se podrá eliminar. Se avisará del motivo.
R6	El sistema contará con la sección Familias de Artículo
R6.1	El usuario podrá crear una familia de artículo
R6.2	El usuario podrá modificar una familia de artículo.
R6.3	El usuario podrá eliminar una familia de artículo.
R6.4	El usuario podrá filtrar familias de artículos.
R6.5	El usuario podrá listar todos las familias de artículos existentes.
R6.6	Si la familia de artículo está siendo utilizado, no se podrá eliminar. Se avisará del motivo.
R7	El sistema contará con la sección Medicamentos
R7.1	El usuario podrá crear un medicamento
R7.2	El usuario podrá modificar un medicamento.
R7.3	El usuario podrá eliminar un medicamento.
R7.4	El usuario podrá filtrar medicamentos.
R7.5	El usuario podrá listar todos los medicamentos existentes.
R7.6	Si el medicamento está siendo utilizado, no se podrá eliminar. Se



	avisará del motivo.
R8	El sistema contará con la sección Tratamientos
R8.1	El usuario podrá crear un tratamiento
R8.2	El usuario podrá modificar un tratamiento
R8.3	El usuario podrá eliminar un tratamiento
R8.4	El usuario podrá filtrar tratamientos
R8.5	El usuario podrá listar todos los tratamientos existentes.
R8.6	Si el tratamiento está siendo utilizado, no se podrá eliminar. Se avisará del motivo.
R9	El sistema contará con la sección Tipos de Tratamiento.
R9.1	El usuario podrá crear un tipo de tratamiento.
R9.2	El usuario podrá modificar un tipo de tratamiento.
R9.3	El usuario podrá eliminar un tipo de tratamiento.
R9.4	El usuario podrá filtrar tipos de tratamiento
R9.5	El usuario podrá listar todos los tipos de tratamiento existentes.
R9.6	Si el tipo de tratamiento está siendo utilizado, no se podrá eliminar. Se avisará del motivo.
R10	El sistema contará con la sección Especialidades
R10.1	El usuario podrá crear una especialidad
R10.2	El usuario podrá modificar una especialidad
R10.3	El usuario podrá eliminar una especialidad
R10.4	El usuario podrá filtrar especialidades
R10.5	El usuario podrá listar todas las especialidades existentes.
R10.6	Si la especialidad está siendo utilizado, no se podrá eliminar. Se avisará del motivo.
R11	El sistema contará con la sección Listado de Grupos Sanguíneos
R11.1	El usuario podrá visualizar el listado de Grupos Sanguíneos.
R11.2	El usuario podrá filtrar Grupos Sanguíneos.
R11.3	El usuario podrá listar todos los Grupos Sanguíneos.



R11.4	El sistema no podrá eliminar ningún Grupo Sanguíneo.
R12	El sistema contará con la sección Listado de Dientes
R12.1	El usuario podrá visualizar el listado de Dientes.
R12.2	El usuario podrá filtrar Dientes
R12.3	El usuario podrá listar todos los Dientes.
R12.4	El sistema no podrá eliminar ningún Diente.
R13	El sistema contará con la sección Reservar Citas
R13.1	El usuario podrá reservar una cita.
R13.2	El usuario podrá visualizar los datos del paciente.
R13.3	El usuario podrá visualizar los datos del médico
R13.4	El usuario podrá seleccionar una hora
R14	El sistema contará con la sección Gestionar Citas.
R14.1	El usuario podrá visualizar el listado de Citas.
R14.2	El usuario podrá filtrar por citas.
R14.3	El usuario podrá listar todas las citas.
R14.4	El usuario podrá anular una cita.
R14.5	El usuario podrá reprogramar una cita.
R14.6	El usuario podrá eliminar una cita.
R14.7	Si la cita está siendo utilizada, no se podrá eliminar. Se avisará sobre ello.
R14.8	Si la cita ha sido confirmada por un médico, cambiará de estado y no será visible dentro de esta sección.
R14.9	El usuario podrá visualizar la información de las citas
R15	El sistema contará con la sección Citas Pendientes de Tratar.
R15.1	El usuario podrá visualizar el listado de Citas Pendientes de Tratar
R15.1	El usuario podrá filtrar por Citas Pendientes de Tratar.
R15.2	El sistema obligará a que el usuario indique un tratamiento antes de guardar.
R15.2	El usuario podrá visualizar el documento en curso si está guardado.



R15.3	Si el usuario registra la visita, entonces cambiará el estado de la cita a Confirmado y no será visible en el panel de gestión de citas.
R15.4	El usuario podrá eliminar la visita actual registrada.
R15.5	Si la visita actual tiene algún registro asociado, no se podrá eliminar. Se avisará.
R15.6	Si se borra la visita actual, entonces la cita cambiará de estado Pendiente.
R15.7	Para poder añadir un tratamiento a la visita, el usuario deberá indicar el diente sobre el que va a realizar el tratamiento
R15.8	Si el usuario intenta guardar la visita sin haber puesto un tratamiento, le dará un aviso de error.
R15.9	El usuario podrá modificar las observaciones del tratamiento a presupuestar.
R15.10	El usuario podrá crear recetas.
R15.11	El usuario podrá añadir documentos a la visita en curso.
R15.12	El usuario podrá seleccionar dientes permanentes o dientes temporales
R16	El sistema tendrá la sección Visitas Pendientes de Facturar
R16.1	El sistema permitirá filtrar Visitas Pendientes de Facturar
R16.2	Cuando el usuario haga click sobre una visita Pendiente de Facturar, se abrirá el escenario para gestionar facturas. Esto se entenderá como un paso previo a la facturación de la visita.
R16.2	El sistema propondrá por defecto en las líneas de la factura el precio que tenga configurado el tratamiento en ese momento.
R16.3	El sistema permitirá modificar el precio de las líneas del paso previo a la factura.
R16.4	El sistema permitirá modificar el importe cobrado al paciente.
R16.5	El sistema obligará a poner una serie en el documento.
R16.6	El sistema obtendrá el nº máximo de documento para la serie especificada y le sumará 1. Este será el nº de la factura.
R16.7	Una vez se haya generado la factura, el usuario no podrá modificar la serie del documento.
R16.8	El usuario podrá visualizar la información del paciente y del médico que ha realizado el tratamiento.



Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

R16.9	El sistema tendrá una opción para aplicar un descuento al cliente. Esto repercutirá en la base imponible, como descuento.
R16.10	El sistema tendrá una opción para establecer un % de IVA. Esto repercutirá en el total de la factura, como aumento.
R16.11	Si el usuario modifica el precio de la línea de la factura, se realizará un recálculo a nivel de factura (cabecera).
R16.12	Si se modifica el % de descuento o % de IVA, el sistema hará un recálculo a nivel de factura (cabecera).
R16.13	El usuario podrá crear la factura a partir de la visita tratada.
R16.14	El usuario podrá añadir una observación a nivel de factura (cabecera).
R17	El sistema tendrá la sección "Facturas"
R17.1	Las características serán similar al punto R16 y toda su extensión.
R18	El sistema tendrá a sección "Historial Clínico"
R19	El usuario podrá modificar el Historial Clínico.
R19.1	El usuario podrá visualizar la trazabilidad de todas las visitas a las que ha acudido el paciente.
R19.2	El usuario podrá desplazarse a la factura en cuestión si se le ha facturado, haciendo click en la malla de datos y pulsado en el botón de ver factura.
R19.3	El usuario podrá desplazarse a la visita en cuestión, haciendo click en la malla de datos y pulsado en el botón ver visita.
R19.4	El usuario podrá imprimir el historial completo del paciente en curso.
R19.5	El usuario podrá filtrar por paciente
R20	El sistema tendrá la opción "Reportes"
R20.1	El usuario podrá hacer uso de todos los informes disponibles
R21	El sistema tendrá la opción "Salir"
R21.1	El usuario podrá cerrar la aplicación



3.2. Diseño de base de datos

3.2.1. Resumen

La aplicación Dentidex utiliza una base de datos relacional para almacenar la información de los usuarios del sistema, artículos y sus jerarquías, pacientes, médicos, proveedores, citas, historiales clínicos, recetas, medicamentos, facturas ... etc.

Al inicio del proyecto se planteó que cada usuario pudiera acceder al sistema por medio de un formulario de login y un usuario y una contraseña. Esto permitiría por una parte que los usuarios de la clínica se validasen en el sistema para hacer cualquier tipo de gestión y limitar responsabilidad. Además por otra parte, el paciente podría validarse por medio de una aplicación móvil y ver online su historial clínico.

En base a lo mencionado, la entidad usuario se guarda a nivel de bases de datos, pero no se le da ningún uso actualmente.

Los artículos se jerarquizarán por medio de la entidad FamiliaArticulo.

Una familia de artículo podría entenderse como una agrupación de varios artículos.

Un artículo estará formado a su vez por una familia de artículo y un medicamento (opcional).

Un proveedor estará formado por un sujeto, que a su vez estará formado por una dirección, y un usuario (opcional).

Un médico estará formado por una especialidad y un sujeto, que a su vez estará formado por una dirección y un usuario (opcional).

Un paciente estará formado por un grupo sanguíneo, un género, un estado civil, un sujeto, que a su vez estará formado por una dirección y un usuario (opcional). Además estará formado un historial clínico (cabecera).

Una cita estará formada un estado (Pendiente, Confirmado, Anulado), un paciente, un médico, un historial clínico detalle y tendrá relación débil con una hora de cita (sólo se copiará la hora).

Un historial clínico podría entenderse como una agrupación de varias líneas de historial clínico..

Una línea de historial clínico estará formado por una cita, varias líneas de historial clínico de receta, varias líneas de historial clínico de documento y varias líneas de historial clínico de tratamientos.

Una línea de historial clínico receta estará formada por una línea de historial clínico y un medicamento.

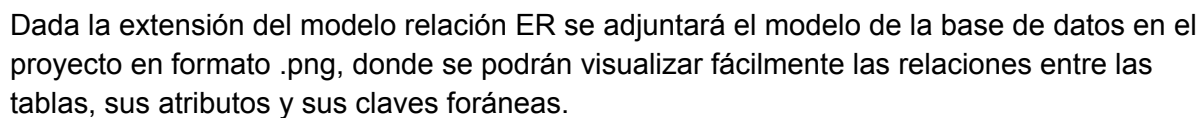
Una línea de historial clínico de tratamiento estará formada por una línea de historial clínico, un diente, un tratamiento y una línea de factura.

Los tratamientos se jerarquizarán por medio del tipo de tratamiento.

Un tratamiento estará formado por un tipo de tratamiento y una línea de historial clínico de tratamiento.

Una línea de factura estará formada por una factura y una línea de historial clínico de tratamiento.

Una factura podría entenderse como una agrupación de líneas de factura.





3.2.2. Procedimientos almacenados (stored procedures)

Para el desarrollo del proyecto se ha tenido que hacer uso de un procedimiento “procesarCita()”. En PostgreSQL, se denomina función al procedimiento.

De modo resumido, este procedimiento se encarga de actualizar el estado de la cita cuando se inserta o se elimina un registro en la tabla HistorialClinicoDetalle.

El usuario puede crear una línea de historial clínico y para ello necesita una cita válida como base. Cuando se crea el registro a partir de la cita previa, se modifica el valor del estado de la cita a ‘Confirmado’ y cuando se elimina, el estado vuelve a ‘Pendiente’.

¿Qué es un procedimiento almacenado?

Un procedimiento almacenado se puede definir como un bloque de código guardado en base de datos que puede ser reutilizado de manera recursiva.

Una sentencia SQL, debe ser escrita en cualquier parte del código donde sea necesaria. En cambio, un procedimiento almacenado sólo necesita escribirse una sola vez y puede ser ejecutado desde cualquier parte del código. Es común utilizarlo junto a triggers.

Tiene varias ventajas:

- Se ejecuta en el motor de la base de datos, por lo que disminuye la sobrecarga de de datos entrantes y salientes y aumenta su velocidad de procesamiento.
- Simplifica la creación y mantenimiento de la lógica de negocio de la aplicación.
- Disminuye la probabilidad de que los datos se corrompan por el uso de programas cliente defectuosos.

3.2.3. Triggers (disparadores)

En adicción a lo mencionado en el punto anterior, hemos hecho uso de triggers para mejorar la integridad de la base de datos.

Se ha creado un trigger en la tabla HistorialClinicoDetalle, el cual dispara la función mencionada en el punto anterior.

El principal motivo de su uso es para mejorar la integridad de los datos. Por lo tanto, se tuvo que crear un trigger que respetase la lógica de negocio de la aplicación.

¿Qué es un trigger?

Un trigger se puede definir como una especie de procedimiento que se ejecuta automáticamente cuando se produce un evento en alguna de las tablas de la base de datos.

Este trigger se desencadena cuando se realiza la operación INSERT o DELETE. El trigger se produce tanto antes como después de ejecutarse el evento.



Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

Su principal utilidad es mejorar la gestión de la base de datos, implementando reglas de negocio para mejorar la integridad de la base de datos. Los trigger permiten sincronizar datos entre tablas, modificar valores o incluso prevenir errores.

3.2.4. Vistas (PostgreSQL)

Para la realización del proyecto he tenido que crear varias vistas. Entre ellas:

- ArticulosVista.
- CitasVista.
- FacturasVista.
- MedicosVista.
- ProveedoresVista.
- PacientesVista.
- SujetosVista.
- RecetasVista.
- TratamientosVista.

El código para la creación de las vistas se facilita en el proyecto, en ficheros .sql.

¿Qué es una vista?

Una vista es una tabla cuyo contenido está definido por una consulta. Al igual que una tabla, una vista consta de un conjunto de columnas y filas de datos con un nombre.

Una de sus mayores ventajas es que si se modifica alguno de los valores de la tabla que subyace bajo la tabla, también se verá reflejado en la vista.

Las vistas también facilitan la gestión de las consultas y permite por una parte, enmascarar las tablas que hay en la base de datos y por otra parte, conceder permisos sólo de lectura, para evitar riesgos.



4. Arquitectura del sistema

4.1. Patrón MVC

La arquitectura Modelo-Vista-Controlador (MVC) es un patrón de arquitectura que está pensado para separar dentro de una aplicación los tres componentes principales, que son los datos (modelos), interfaces de usuario (vistas) y la lógica de negocio de sistema (controladores).

Componentes MVC

Modelo. El componente modelo corresponde a los datos con los que el usuario trabaja. Estos datos son transferidos entre los componentes Vista y Controladores.

Supongamos como ejemplo un objeto Medicamento el cual recuperará la información del producto desde la base de datos, posteriormente lo manipulará y actualizará y finalmente devolverá el dato a la base de datos o bien será utilizado para renderizarlo en una vista..

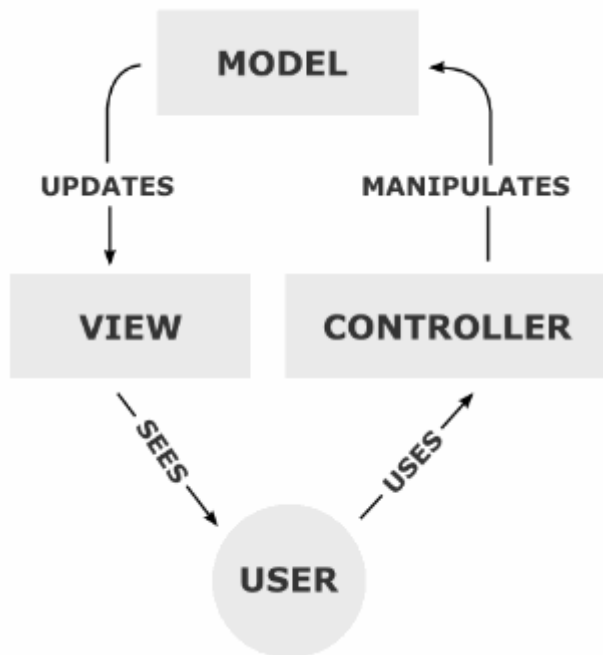
Vista. El componente Vista es utilizado por todas las interfaces de usuario de la aplicación. Por ejemplo, la vista de un producto incluirá todos los elementos con los que interactuará el usuario final (selectores, checkboxes, textbox.... etc).

Controlador. El componente Controlador funciona como una capa intermedia entre los componentes Vista y Modelo, el cual procesa toda la lógica de negocio y las solicitudes entrantes, manipula los datos haciendo uso del componente Modelo e interactúa con la vista para que renderice los datos finales que retorne.

Un ejemplo de ello sería un controlador de un medicamento el cual controlará todas las interacciones y entradas de datos desde la vista Medicamento y actualizará la base de datos utilizando el componente Medicamento.

Por lo tanto, la arquitectura MVC aporta múltiples ventajas, entre ellas:

- Aislar de manera clara la lógica de negocio entre componentes, facilitando el mantenimiento y la escalabilidad de la aplicación
- Permite representar los mismos datos de distintas maneras. Por ello podríamos utilizar distintas vistas.
- Código más legible.
- Pruebas aisladas.



Esquema MVC. Wikipedia

4.2. Estructura del proyecto

Una de las ventajas de JavaFX es que nos permite utilizar el patrón MCV, con el uso de una variante XML propia de esta tecnología denominada FXML.

El Modelo consiste en los distintos objetos que están dentro del dominio de la aplicación.

La Vista consiste en los propios ficheros FXML.

El Controlador es el propio código en Java que define el comportamiento que debe seguir la GUI (Interfaz de Usuario) para interactuar con el usuario.

Basándonos en dicho patrón se ha estructurado el proyecto del siguiente modo:

- Modelos:
 - Artículo.java
 - Cita.java
 - CitaHora.java
 - Diente.java
 - Direccion.java
 - Enumerados.java
 - Especialidad.java
 - EstadoCivil.java
 - Factura.java
 - FacturaDetalle.java
 - FamiliaArticulo.java
 - Genero.java
 - GrupoSanguineo.java
 - HistorialClinico.java



- HistorialClinicoDetalle_Documento.java
- HistorialClinicoDetalle_Receta.java
- HistorialClinicoDetalle_Tratamiento.java
- HistorialClinicoDetalle.java
- ICargaDatos.java
- ICodigoUnico.java
- INombreUnico.java
- IOperable.java
- ISujeto.java
- IUsuario.java
- Medicamento.java
- Medico.java
- Paciente.java
- Proveedor.java
- Sujeto.java.java
- TipoTratamiento.java
- Tratamiento.java
- Usuario.java
- Controladoras:
 - AlmacenShowController.java
 - ArticulosShowController.java
 - CitasHoraSeleccionController.java
 - CitasPendientesSeleccionController.java
 - DientesListadoController.java
 - DientesSeleccionController.java
 - EspecialidadesShowController.java
 - FacturasSeleccionController.java
 - FacturasShowController.java
 - FamiliasArticuloShowController.java
 - FormularioBaseController.java
 - FormularioInicioController.java
 - GestionarCitasReprogramacionShowController.java
 - GestionarCitasShowController.java
 - GruposSanguineoListadoController.java
 - HistorialClinicoDetalleSeleccionController.java
 - HistorialClinicoShowController.java
 - MedicamentosSeleccionController.java
 - MedicamentosShowController.java
 - MedicosSeleccionController.java
 - MedicosShowController.java
 - PacientesSeleccionController.java
 - PacientesShowController.java
 - ProveedoresSeleccionController.java
 - ProveedoresShowController.java
 - ReservarCitasShowController.java



- TiposTratamientoShowController.java
- TratamientosShowController.java
- Vistas:
 - AlmacenShow.fxml
 - ArticulosShow.fxml
 - CitasHoraSeleccion.fxml
 - CitasPendientesSeleccion.fxml
 - DientesListado.fxml
 - DientesSeleccion.fxml
 - EspecialidadesShow.fxml
 - FacturasSeleccion.fxml
 - FacturasShow.fxml
 - FamiliasArticuloShow.fxml
 - FormularioBase.fxml
 - FormularioInicio.fxml
 - GestionarCitasReprogramacionShow.fxml
 - GestionarCitasShow.fxml
 - GruposSanguineoListado.fxml
 - HistorialClinicoDetalleSeleccion.fxml
 - HistorialClinicoShow.fxml
 - MedicamentosSeleccion.fxml
 - MedicamentosShow.fxml
 - MedicosSeleccion.fxml
 - MedicosShow.fxml
 - PacientesSeleccion.fxml
 - PacientesShow.fxml
 - ProveedoresSeleccion.fxml
 - PacientesShow.fxml
 - ProveedoresSeleccion.fxml
 - ReservarCitasShow.fxml
 - TiposTratamientoShow.fxml
 - TratamientosShow.fxml
- Vistas.css:
 - DarkTheme.css
- Util:
 - Alertas.java. **Se hace uso de esta clase para mostrar avisos sencillos al usuario, el típico alert.**
 - DialogBox.java. **Se hace uso de esta clase para poder crear formularios de selección de elementos, como seleccionar pacientes, médicos, dientes, etc.**
 - Eventos.java. **Se hace uso de esta clase para poder crear máscaras, ya sea obligar a los textbox a tener sólo números, decimales, o un límite máximo de caracteres.**
 - Ficheros.java. **Se hace uso de esta clase para la utilización sencilla de ficheros.**



- **Imagenes.java. Se hace uso de esta clase para reconversión de imágenes o bien para obtenerla a través del paquete resources.**
- **Tiempo.java. Se hace uso de esta clase para la conversión de tipos a formatos de fecha.**
- **Informes.java. Se hace uso de esta clase para la renderización e invocación de los informes.**
- Informes:
 - CabeceraDocumento.jrxml
 - InformeFactura.jrxml
 - InformeHistorialClinico.jrxml
 - InformeHistorialClinicoDetalle.jrxml
 - ListadoArticulos.jrxml
 - ListadoCitasPorMedico.jrxml
 - ListadoEspecialidades.jrxml
 - ListadoFacturas.jrxml
 - ListadoFacturasDetalle.jrxml
 - ListadoFamiliasArticulo.jrxml
 - ListadoHistorialClinico.jrxml
 - ListadoHistorialClinicoDetalle.jrxml
 - ListadoHistorialClinicoDetalleRecetas.jrxml
 - ListadoHistorialClinicoDetalleTratamientos.jrxml
 - ListadoMedicamentos.jrxml
 - ListadoMedicos.jrxml
 - ListadoMedicosPorEspecialidad.jrxml
 - ListadoMedicosPorPoblacion.jrxml
 - ListadoPacientes.jrxml
 - ListadoPacientesDetalle.jrxml
 - ListadoPacientesDeudores.jrxml
 - ListadoPacientesPorGenero.jrxml
 - ListadoPacientesPorProvincia.jrxml
 - ListadoPacientesPorTipoSanguineo.jrxml
 - ListadoProveedores.jrxml
 - ListadoRecetasMedicasPorMedico.jrxml
 - ListadoRecetasMedicasPorPaciente.jrxml
 - ListadoTiposTratamiento.jrxml
 - ListadoTratamientos.jrxml

En esta imagen podemos ver a rasgos generales la estructura del proyecto.



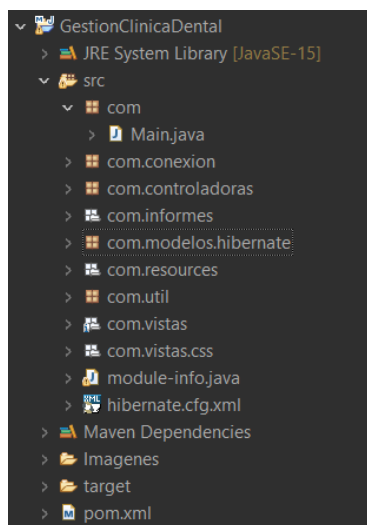
Región de
Murcia

CIFP Carlos III

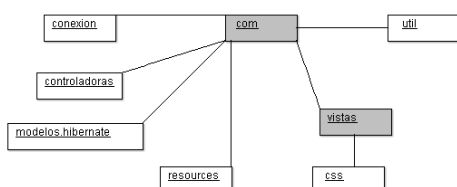
C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)



Aquí se facilita a modo de esquema la visualización del cómo está estructurado el proyecto.



Por medio de estas imágenes podemos ver la extensión del proyecto.
Algunos de los informes creados durante el desarrollo del proyecto.



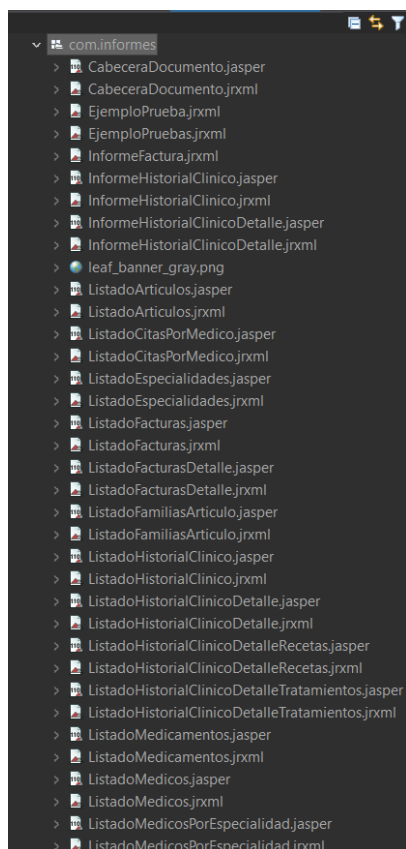
Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)





Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

Controladoras creadas para la realización del proyecto.

```
com.controladoras
> AlmacenShowController.java
> ArticulosShowController.java
> CitasHoraSeleccionController.java
> CitasPendientesSeleccionController.java
> DientesListadoController.java
> DientesSeleccionController.java
> EspecialidadesShowController.java
> FacturasSeleccionController.java
> FacturasShowController.java
> FamiliasArticuloShowController.java
> FormularioBaseController.java
> FormularioInicioController.java
> GestionarCitasReprogramacionShowController.java
> GestionarCitasShowController.java
> GruposSanguineoListadoController.java
> HistorialClinicoDetalleSeleccionController.java
> HistorialClinicoDetalleShowController.java
> HistorialClinicoShowController.java
> MedicamentosSeleccionController.java
> MedicamentosShowController.java
> MedicosSeleccionController.java
> MedicosShowController.java
> PacientesSeleccionController.java
> PacientesShowController.java
> ProveedoresSeleccionController.java
> ProveedoresShowController.java
> ReservarCitasShowController.java
> TiposTratamientoShowController.java
> TratamientosShowController.java
```



Región de
Murcia

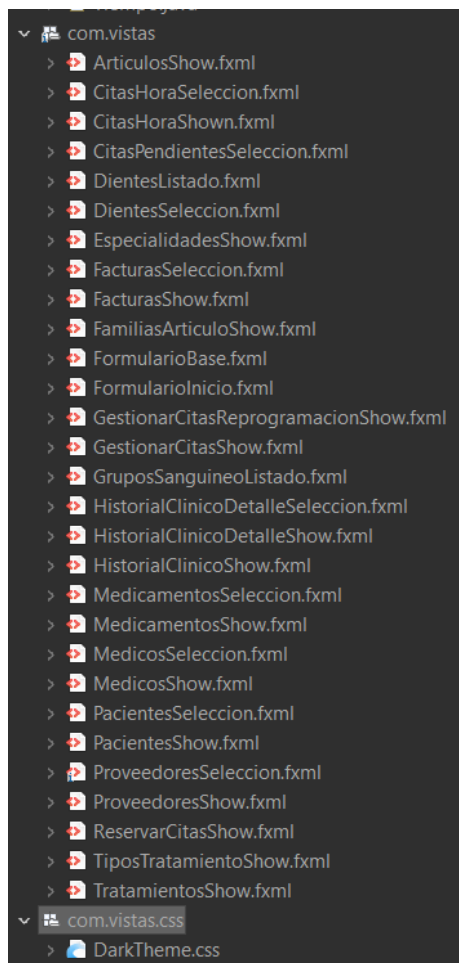
CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

Vistas creadas para la realización del proyecto.





Región de
Murcia

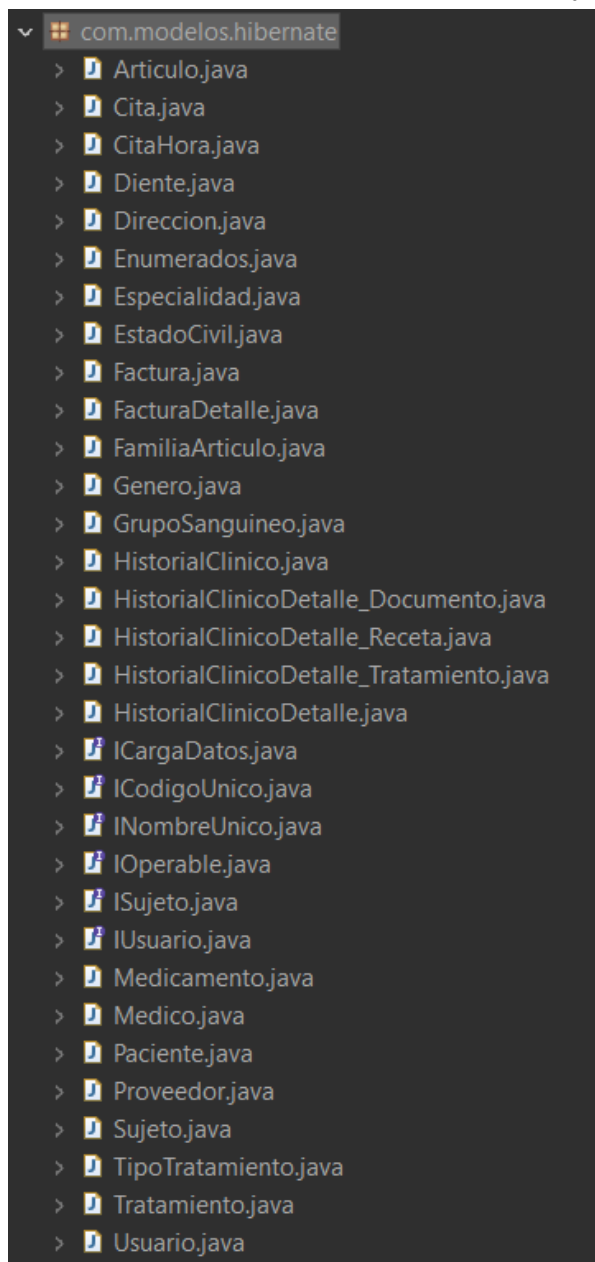
CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

Modelos creados para la realización del proyecto





Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

Inicialmente se pensó que sería adecuado utilizar Hibernate para hacer el mapeado de las entidades y trabajar directamente sobre ello.

De hecho Hibernate llegó a crear las clases que existían en la base de datos de PostgreSQL por medio de ingeniería inversa.

Estuve investigando la utilización de Hibernate y vi que aconsejaban utilizar Hibernate si el modelo de datos lo iba a crear el propio mapeador objeto-relacional.

Como en mi caso no fue así, sino que creé desde 0 el modelo ER, por medio de sentencias SQL, y además el tiempo muy ajustado como para invertirlo en algo que no sabía si iba a ser fructífero, pensé que sería viable desarrollar yo mismo el mapeador dentro de las clases java.

Por lo tanto, tuve que crear varias interfaces que confiriesen cierta genericidad en las clases, para poder hacer uso de los objetos de manera más segura y dinámica.

Un ejemplo de ello es la interface IOperable, que hereda los métodos de la interfaz ICargaDatos. Encontré la necesidad de crear esta interfaz para obligar que las clases que operan con la base de datos tengan sí o sí estos métodos.

```
2
3
4 public interface IOperable extends ICargaDatos{
5
6     public boolean guardar();
7     public boolean esEliminable();
8     public boolean borrar();
9
10
11 }
```

Aquí podemos ver cómo declara los métodos cargar, toString() y getLista(), como tipo genérico.

```
1 import java.util.ArrayList;
2
3
4
5 public interface ICargaDatos {
6     public boolean cargar();
7     public ArrayList<?> getLista();
8     public String toString();
9
10
11 }
```



Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



Ejemplo de definición de la clase Artículo, la cual implementa la interfaz IOperable e ICodigoUnico.

```
4 */
5 @SuppressWarnings("serial")
6 public class Artículo implements java.io.Serializable, IOperable, ICodigoUnico {
7
8     private int id;
9     private FamiliaArticulo familiaArticulo;
10    private String codigo;
11    private String nombre;
12    private Double precioCompra;
13    private Integer cantidadExistencia;
14    private Medicamento medicamento;
15
```

En esta imagen podemos ver cómo se implementaron dichos métodos para poder operar sobre la base de datos.

```
@Override
public boolean guardar() {
    String sql = "";

    if(this.getId() == 0) {
        sql = "INSERT INTO Articulos (Codigo, Nombre, PrecioCompra, CantidadExistencia, Id_FamiliaArticulo, Id_Medicamento) "
            + " VALUES ('" + this.getCodigo().trim() + "', '" +
                this.getNombre().trim() + "', " +
                this.getPrecioCompra().toString() + ", " +
                this.getCantidadExistencia().toString() + ", " +
                (this.getFamiliaArticulo() != null && this.getFamiliaArticulo().getId() > 0 ? this.getFamiliaArticulo().getId() : 0) + ", " +
                (this.getMedicamento() != null && this.getMedicamento().getId() > 0 ? this.getMedicamento().getId() : 0) + " )";
    } else {
        sql = "UPDATE Articulos "
            + " SET Codigo = '" + this.getCodigo().trim() + "',"
            + " Nombre = '" + this.getNombre().trim() + "',"
            + " PrecioCompra = " + this.getPrecioCompra() + ","
            + " CantidadExistencia = " + this.getCantidadExistencia() + ","
            + " Id_FamiliaArticulo = " + (this.getFamiliaArticulo() != null && this.getFamiliaArticulo().getId() > 0 ? this.getFamiliaArticulo().getId() : 0) + ","
            + " Id_Medicamento = " + (this.getMedicamento() != null && this.getMedicamento().getId() > 0 ? this.getMedicamento().getId() : 0) + " "
            + " WHERE Id = " + this.getId();
    }
    return ConexionBBDD.setDatos(sql);
}

@Override
public boolean esEliminable() {
    //Esta clase no tiene dependencias de momento. Por lo tanto siempre será eliminable
    return true;
}

@Override
public boolean borrar() {
    String sql =
        "DELETE "
        + " FROM Articulos"
        + " WHERE ID = " + this.getId();
    return ConexionBBDD.setDatos(sql);
}
```




Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

Al igual que los métodos `getLista()` y `cargar()`, que acceden a la base de datos, mapean los objetos y se devuelve el objeto a sí mismo o bien un `ArrayList` de objeto de un tipo determinado.

```
@Override
public ArrayList<Articulo> getLista() {
    ArrayList<Articulo> lista = new ArrayList<Articulo>();
    try {
        ResultSet rs = ConexionBBDD.getDatos("SELECT * FROM ArticulosVista");
        while(rs.next()) {
            Articulo p = new Articulo();
            p.setId(rs.getInt("Id_Articulo"));
            p.setCodigo(rs.getString("ArticuloCodigo"));
            p.setNombre(rs.getString("ArticuloNombre"));
            p.setPrecioCompra(rs.getDouble("PrecioCompra"));
            p.setCantidadExistencia(rs.getInt("CantidadExistencia"));
            p.setFamiliaArticulo(new FamiliaArticulo());
            p.getFamiliaArticulo().setId(rs.getInt("Id_FamiliaArticulo"));
            p.getFamiliaArticulo().setCodigo(rs.getString("FamiliaArticuloCodigo"));
            p.getFamiliaArticulo().setNombre(rs.getString("FamiliaArticuloNombre"));
            p.setMedicamento(new Medicamento());
            p.getMedicamento().setId(rs.getInt("Id_Medicamento"));
            p.getMedicamento().setFarmaco(rs.getString("Farmaco"));
            p.getMedicamento().setNombreComercial(rs.getString("NombreComercial"));
            p.getMedicamento().setPresentacion(rs.getString("Presentacion"));
            p.getMedicamento().setIndicacion(rs.getString("Indicacion"));
            p.getMedicamento().setViaAdministracion(rs.getString("ViaAdministracion"));

            lista.add(p);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return lista;
}
```

```
@Override
public boolean cargar() {
    try {
        ResultSet rs = ConexionBBDD.getDatos("SELECT * FROM ArticulosVista WHERE Id_Articulo = " + this.getId());
        if (rs.next() == false) return false;
        else {
            setCodigo(rs.getString("ArticuloCodigo"));
            setNombre(rs.getString("ArticuloNombre"));
            setPrecioCompra(rs.getDouble("PrecioCompra"));
            setCantidadExistencia(rs.getInt("CantidadExistencia"));
            setFamiliaArticulo(new FamiliaArticulo());
            getFamiliaArticulo().setId(rs.getInt("Id_FamiliaArticulo"));
            getFamiliaArticulo().setCodigo(rs.getString("FamiliaArticuloCodigo"));
            getFamiliaArticulo().setNombre(rs.getString("FamiliaArticuloNombre"));
            setMedicamento(new Medicamento());
            getMedicamento().setId(rs.getInt("Id_Medicamento"));
            getMedicamento().setFarmaco(rs.getString("Farmaco"));
            getMedicamento().setNombreComercial(rs.getString("NombreComercial"));
            getMedicamento().setPresentacion(rs.getString("Presentacion"));
            getMedicamento().setIndicacion(rs.getString("Indicacion"));
            getMedicamento().setViaAdministracion(rs.getString("ViaAdministracion"));
        }
    } catch (SQLException e) {
        e.printStackTrace();
        return false;
    }
    return true;
}
```



Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

5. Interfaces de usuario

La aplicación cuenta con un menú de 8 ítems submenús y una botonera de 10 ítems.

En la mayoría de escenarios tiene la siguiente estructura:

- Un menú de opciones, que permite crear, modificar o eliminar registros, además de listar los documentos de la entidad en concreto.
- Una caja de búsqueda para filtrar los datos de la malla de datos
- Un formulario para poder añadir o modificar datos.
- Una malla de datos para visualizar los datos registrados.

Se muestran algunas imágenes aleatorias de cómo se visualizan las interfaces de usuario.

Inicio





Software de Gestión de Clínica Dental

Pacientes

Médicos

Citas

Tratamientos

Almacén

Facturas

Reportes

Configuración

Ayuda

Inicio

Pacientes

Reservar Cita

Tratamientos

Médicos

Artículos

Historial Clínico

Citas Pendientes

Visitas Pendientes Facturar

Facturas

PROVEEDORES

Nuevo

Modificar

Borrar

Guardar Cambios

Cancelar Cambios

Listar

BÚSQUEDA

Buscar por ...

Apellido

Nombre

Dirección

Id:

21

Fecha Creación:

2021-05-22 14:19:48.699657

Fotografía:

Nombre:

Alejandro

Apellidos:

Parra Perez

NIF:

9999999999d

Email:

dalkdaskdalkida

Buscar Imagen

Fecha Nacimiento:

18/5/1993

Teléfono

99999999999999

Persona Contacto:

ldksañld

Página Web:

Id_Proveedor	Nombre	Nif	F. Nacimiento	Email	Teléfono	Persona Con...	Página Web	Dirección	Población	CP
1	Alejandro Parr...	9999999999d	1993-05-18 00:00:00.0	dalkdaskdalkida	99999999999999	ldksañld		Av. Sangonera	Murcia	30860



PACIENTES

Nuevo

Modificar

Borrar

Guardar Cambios

Cancelar Cambios

Listar

BUSQUEDA

Buscar por ...

Sujeto

Id:

000000

Dirección

Fecha Creación:

Nombre:

Apellidos:

NIF:

Email:

Fecha Nacimiento:

Teléfono

Profesión

Género:

Estado civil:

Grupo Sanguíneo:

Fotografía:

Buscar Imagen

Id_Paciente	Nombre	Nif	F. Nacimiento	Teléfono	Profesión	Género	Estado Civil	G. Sanguíneo	Dirección	Pais
4	Antonela Ba...	nif26999999999999999	2016-05-10 00:00:00	telefono1	-	Mujer	Otros	A+	Barriada La Est...	Hu...
2	Ernesto Laca...	nif2nif2nif2nif2nif2	1979-05-11 00:00:00	698512354	Abogado	Hombre	Otros	A+	Calle Rosaleda	Mu...
7	Emilio Sanch...	12341234j	1998-05-07 00:00:00	695874123	Estudiante	Hombre	Soltero	B+	Calle La Palmera	Ma...



Región de Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

Visitas

Tratamiento de la Cita en curso.

Datos de la Cita

Id Visita	Fecha Visita	Hora	Id Cita	Id Médico	Nombre Médico
6	25/05/2021	19:00	4	2	Carmelo Buenafuente
Id Paciente	Nombre Paciente	Edad	Género	NIF	
2	Ernesto Lacasito Fin	42	Hombre	nif2nif2nif2ni	

DIENTES PERMANENTES DIENTES TEMPORALES

(18)	(17)	(16)	(15)	(14)	(13)	(12)	(11)	(21)	(22)	(23)	(24)	(25)	(26)	(27)	(28)
(48)	(47)	(46)	(45)	(44)	(43)	(42)	(41)	(31)	(32)	(33)	(34)	(35)	(36)	(37)	(38)

Código Diente: Cuadrante: Reiniciar

Historial Clínico Sintomas Diagnóstico Tratamientos Recetas Ficheros

Hábitos alimenticios

Medicación Actual

Tratamientos Anteriores

Sangrado Excesivo Problema Sanguíneo VIH Embarazada Fuma

Pastillas Anticonceptivas Dificultad Abrir Boca Ruido Abrir/Cerrar Boca Muerde Uñas Consumo Alimentos Cítricos

Peso Kg 0.0

Altura Cm 0.0

Temperatura 0.0

Frecuencia Cardíaca 0

Operaciones

ELIMINAR

GUARDAR

VER PRESUPUESTO

Visualización de una factura

Gestión de la factura en curso

Datos de la Cabecera de la Factura

Serie	Numero	F. Emisión	Id Médico	Nombre Médico
xx	1	29/05/20	2	Carmelo Buenafuente
Id Paciente	Nombre Paciente	Provincia	Poblacion	NIF
2	Ernesto Lacasito Fin	Murcia	Murcia	nif2nif2nif2ni

Observaciones

Detalle de la Factura

Id_FacturaDeta...	Diente	Tratamiento	Tipo	Precio	Cobrado	Observaciones
1	Nº: 21-2	Brackets de zafiro	Ortodoncia	5600.0	2.5	sdsdsd

Pie de la Factura

Dto. Cliente (%)	0.0	Base Imponible	5600.0
% IVA	0.0	Total Factura	5600.0

Operaciones

ELIMINAR

GUARDAR

VER FACTURA



Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



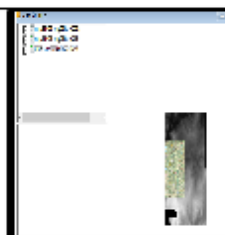
ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

Impresión de un historial clínico en PDF

INFORME HISTORIAL CLÍNICO DEL PACIENTE ACTUAL

Paciente: Ernesto Lacasito Fin(2)

SUJETO					
Fecha de Nacimiento: 1979-05-11 NIF: nif2nif2nif2nif2 Género: Hombre	<table><thead><tr><th>GRUPO SANGUÍNEO</th></tr></thead><tbody><tr><td>Grupo Sanguíneo: A+</td></tr><tr><td>Puede recibir de: A+ AB+</td></tr><tr><td>Puede Donar de: O+ O- A+ A-</td></tr></tbody></table>	GRUPO SANGUÍNEO	Grupo Sanguíneo: A+	Puede recibir de: A+ AB+	Puede Donar de: O+ O- A+ A-
GRUPO SANGUÍNEO					
Grupo Sanguíneo: A+					
Puede recibir de: A+ AB+					
Puede Donar de: O+ O- A+ A-					



HISTORIAL CLÍNICO

Antecedentes Patológicos: dsadsadasdsd

Enfermedades Familiares: dsdsdsdasdasd

Alergias: lññlklkñl

VISITAS

Id Cita: 4 Cita: 2021-05-25 19:00:00 Médico Cita: Carmelo Buenafuente (ID: 2)

Obs. Cita: dasdasdada

Id Visita: 6 Fecha Visita: 2021-05-25

Habitos Alimenticios: sssss

Medicación Actual:

Tratamientos Anteriores: dsdsd

Vista Observaciones: 99999999999999

SÍNTOMAS

- | | |
|--|---|
| <input type="checkbox"/> Sangrado Excesivo | <input type="checkbox"/> Dificultad Abrir Boca |
| <input type="checkbox"/> Problema Sanguíneo | <input checked="" type="checkbox"/> Ruido Abrir Cerrar Boca |
| <input type="checkbox"/> VIH | <input checked="" type="checkbox"/> Muerde Uñas |
| <input checked="" type="checkbox"/> Embarazada | <input checked="" type="checkbox"/> Fuma |
| <input type="checkbox"/> Pastillas Anticonceptivas | <input type="checkbox"/> Consumo Alimentos Cítricos |

Temperatura: 0.00

Frecuencia Cardíaca: 0

Altura Cm: 0.00

Peso Kg: 0.00

TRATAMIENTO

DIENTE

Brackets de zafiro

Incisivo Central(21)

Obs: dsdsd

MEDICAMENTO

INDICACIONES

SOLINITRINA

*Spray: 2 puff cada 5 min hasta un máximo de 3-4
veces
dasdasdasd



Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

Gestionar Citas

Software de Gestión de Clínica Dental

Pacientes Médicos Citas Tratamientos Almacén Facturas Reportes Configuración Ayuda

Inicio Pacientes Reservar Cita Tratamientos Medicos Articulos Historial Clínico Citas Pendientes Visitas Pendientes Facturar Facturas

OPCIONES

Anular Cita

Cambiar a Pendiente

Reprogramar Cita

Eliminar Cita

BUSQUEDA

Buscar por ...

5/5/2020

25/6/2021

Citas Pendientes Citas Anuladas Citas Expiradas

IdCita	IdPaciente	IdMedico	F. Cita	H. Cita	Observaciones Cita
2	5	2	2021-05-22	10:30	UFFFFFF madre mia
1	2	7	2021-05-05	10:00	

Datos de la Cita y del Paciente

Cita Nº: Fecha Cita: Hora Cita:

Código Paciente: --

Código Médico: --

Dirección: Municipio:

Teléfono:

Reserva de una cita

Software de Gestión de Clínica Dental

Pacientes Médicos Citas Tratamientos Almacén Facturas Reportes Configuración Ayuda

Inicio Pacientes Reservar Cita Tratamientos Medicos Articulos Historial Clínico Citas Pendientes Visitas Pendientes Facturar Facturas

OPCIONES

Nuevo Cita

Guardar Cambios

Cancelar Cambios

BUSQUEDA

Buscar por ...

Datos Paciente

IdPaciente: 7 Buscar Paciente

Nombre: Emilio Sanchez Ortega

NIF: 12341234j

Dirección: Calle La Palmera

Teléfono: 695874123

Datos Cita

IdMedico: 5 Buscar Médico

Fecha Cita: 9/6/2021

Observaciones: No se que poner aqui

Hora Disponible

09:00

09:30

10:00

10:30

11:00



6. Implementación de la aplicación

Consiste en implementar la aplicación en un entorno empresarial real.

Para ello hemos de seguir una serie de pasos:

- Instalar una base de datos PostgreSQL. Para ello, he tenido que instalar Laragon, el cual nos ofrece una serie de herramientas para desarrollar. En mi situación me ha bastado con poder utilizar su base de datos. La base de datos debe tener el puerto **5432** abierto y el usuario por defecto es postgres.
- Creamos la base de datos ClinicaDental, las tablas necesarias para que funcione el sistema y le añadimos los valores iniciales en las siguientes tablas para que el usuario pueda trabajar:
 - Dientes
 - Grupos Sanguíneos.
 - Géneros.
 - Especialidades.
 - Medicamentos.
 - EstadosCivil.
 - CitasEstado.
 - CitasHora.
 - Tratamientos.
 - Tipos de Tratamiento.

En los anexos finales, se explicará paso a paso cómo instalar todo el software necesario para implementarlo.



8. Conclusiones

En este capítulo, se analizan las conclusiones finales del proyecto y se reflexiona sobre las posibles tareas de mantenimiento y extensión de la aplicación.

8.1. Conclusiones

Este trabajo ha consistido en el desarrollo de Dentidex, una aplicación de escritorio para gestionar de manera sencilla clínicas dentales. Las principales funcionalidades son las siguientes:

- Registro y definición de artículos, pudiendo agruparlos en jerarquías, o asignarles un precio, cantidad en existencias o medicamento por defecto.
- Gestión de pacientes, médicos y proveedores, que supondrán el pilar para gestionar la clínica.
- Registro de pedidos de facturas.
- Gestión de citas, donde se puede gestionar las citas de la clínica dental.
- Tratamiento de las necesidades del paciente, con la expedición de recetas y la incorporación de documentos, trazando lo que sucedió en la visita del paciente.
- Impresión de documentos y visualización de listados, gracias al módulo de informes.

8.2. Trabajo futuro

Este sistema de gestión conlleva unas expectativas bastante interesantes, ya que se ha diseñado como una base para incorporar nuevas funcionalidades, como la implementación de un sistema de gestión de usuarios y roles, para delimitar las funcionalidades que desempeña cada usuario, o bien gestionar más cómodamente las tarifas de los tratamientos de manera periódica.

Adicionalmente, también se ha replanteado implementar otros módulos como son el de compras, que incluiría la compra del producto y su posterior facturación o el de pagos, que facilite la gestión de los pagos a los proveedores a quien se les compre dicho artículo o servicio.

La aplicación espera posibles futuras actualizaciones que mejorarán la seguridad del sistema, ya sea por medio de copias de seguridad o imponiendo medidas de protección a nivel de base de datos para mantener la privacidad e integridad de los datos. También se están analizando el desarrollo de una aplicación para que los pacientes puedan visualizar por medio de un código y contraseña su historial clínico, permitiéndoles conocer el estado actual de sus pagos o su salud general.



Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

9. Bibliografía

<https://sourceforge.net/projects/care2002/>
http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1561-31942018000600104&lng=es&rm=iso&tlng=es#f2
<https://www.ilerna.es/blog/aprende-con-ilerna-online/sanidad/codigo-internacional-dientes-fdi/>
<https://www.guru99.com/postgresql-vs-mysql-difference.html#:~:text=KEY%20DIFFERENCE%3A,feature%20like%20JSON%2C%20XML%20etc.&text=PostgreSQL%20is%20complete%20ACID%20compliant.used%20with%20InnoDB%20and%20NDB.>
<http://openclinic.sourceforge.net/>
<http://www.odonto.me/>
<https://clinic-cloud.com/>
<https://www.youtube.com/channel/UCM2OliYmPkDDjc2iYXxmpYA>
https://sede.aemps.gob.es/docs/CIMA-REST-API_1_19.pdf
<http://datos.salud.gob.ar/dataset/recetas-medicamentos-esenciales/archivo/06e9493d-ff44-4acb-b058-e8adfde41704>
<https://github.com/nuvlar/FIDE>
<https://www.postgresql.org/ftp/pgadmin/pgadmin4/v5.2/windows/>
<https://www.ibm.com/docs/es/informix-servers/12.10?topic=started-what-is-jdbc>
<https://www.programaenlinea.net/que-es-es-laragon/>
<https://oracle-patches.com/en/coding/what-is-javafx>
<https://www.codejava.net/coding/how-to-create-executable-jar-file-with-resources-and-dependencies-using-maven-in-eclipse>
<https://community.oracle.com/tech/developers/discussion/2478906/combobox-is-refreshed-in-side-a-tableview>
<https://community.jaspersoft.com/project/jaspersoft-studio>
http://chuwiki.chuidiang.org/index.php?title=Ejemplo_sencillo_con_Hibernate
https://www.donasang.org/que-es-la-sang/es_quadre-de-compatibilitat.html
<https://www.javaquery.com/2015/11/how-to-fill-jasper-table-using.html>
<http://migranitodejava.blogspot.com/2011/08/introduccion-hibernate.html>
<https://www.eclipse.org/ide/>
<https://stackoverflow.com/questions/4038616/get-count-of-records-affected-by-insert-or-update-in-postgresql>
<https://code.makery.ch/es/library/javafx-tutorial/part4/>
<https://www.clinicabmurri.com/tipos-de-dentistas/>
<https://junit.org/junit5/>
<https://stackoverflow.com/questions/42357924/javafx-external-directory-and-resources-outside-of-jar-file>
<https://javadesdecero.es/avanzado/modulos-java-ejemplos/>
<https://gluonhq.com/products/scene-builder/>
<https://jdbc.postgresql.org/>
<https://www.pgadmin.org/>
<https://argouml.uptodown.com/windows>
<https://www.microsoft.com/es-es/sql-server/sql-server-downloads>
<https://www.mysql.com/>



Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

<https://stackoverflow.com/questions/14978355/jasper-report-integration-to-java>

<https://stackoverflow.com/questions/28547751/fxml-document-naming-convention>

<https://www.stewshack.com/post/jasper-table-parameters>



Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

10.- Anexos

10.1.- Anexo A. Instalación Laragon

Para instalar Laragon, accedemos por medio del siguiente enlace:

<https://laragon.org/download/index.html>

Download



Laragon is a universal development environment. It has many features to make you more productive:

[Benefits of Laragon](#)

After downloading, You can add **git**, **phpmyadmin**, **Node.js/MongoDB**, **Python/Django/Flask/Postgres**, **Ruby**, **Java**, **Go** using “[Tools > Quick add](#)”

Note: [You can also download from GitHub](#)

[Contents](#)

[Edition](#)

[Why Laragon?](#)

[Tutorials](#)

[Back to Top](#)

Edition

- **Laragon Full:** Apache 2.4, Nginx, MySQL 5.7, PHP 7.2, Redis, Memcached, Node.js 11, npm, yarn, git, ...
[Download Laragon - Full \(130 MB\)](#)
- **Laragon Lite:** Don't include Node.js 11, npm, yarn, git but you can add them easily using “[Tools > Quick add](#)”
[Download Laragon - Lite \(85 MB\)](#)
- **Laragon Portable:** PHP 5.4, MySQL 5.1 - Good for getting started with PHP, then you can add newer versions of PHP/MySQL easily later
[Download Laragon - Portable \(18 MB\)](#)

Why Laragon?

Elegimos la opción Laragon Lite y lo descargamos. una vez descargado, lo instalamos e iniciamos el programa.



Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



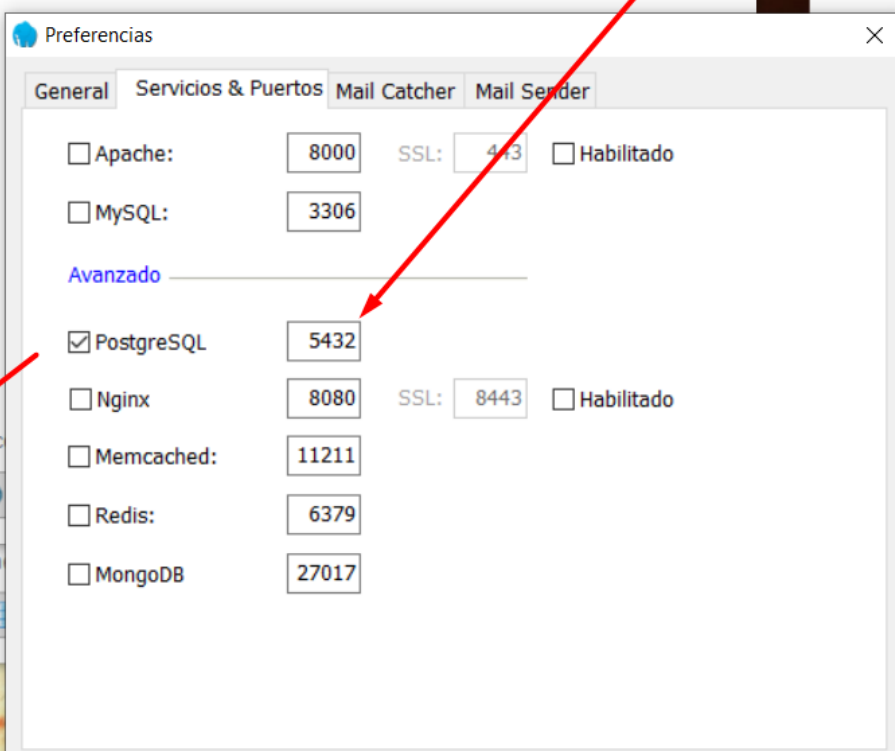
ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

Laragon Full 4.0.15 190/04 php-7.2.19-Win32-VC15-x64 [15] 169.254.120.92



Menú

© Leo K



Una vez abierto el panel principal de Laragon, nos vamos al icono de configuración >> Servicios & Puertos >> Iniciar Todo. Con esto, Laragon iniciará el servidor PostgreSQL, al que nos conectaremos posteriormente.



Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifpcarlos3.es



Laragon Full 4.0.15 190/04 php-7.2.19-Win32-VC15-x64 [15] 169.254.120.92



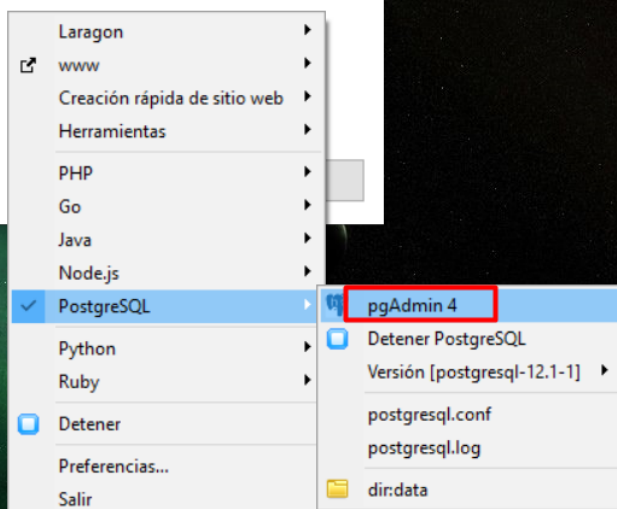
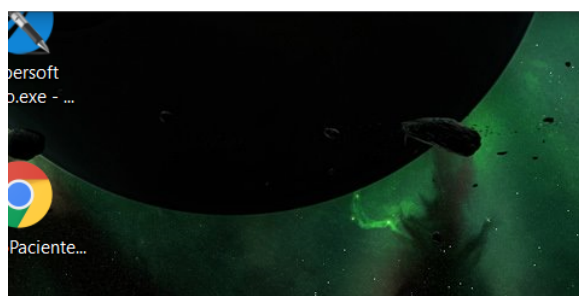
Menú

© Leo K

PostgreSQL postgresql-12.1-1 started

5432

Detener Web Base de Datos



Hacemos click derecho sobre la ventana principal de Laragon, nos desplegará la ventana secundaria, seleccionamos PostgreSQL>>pgAdmin.

Nos abrirá un menú para logearnos a PostgreSQL. Por defecto el usuario es 'postgres' y no tiene contraseña. Desde aquí crearemos una base de datos, denominada por ejemplo ClinicaDental.

De momento, la aplicación no permite modificar dinámicamente los datos de conexión a la base de datos. Esta parte está aún en desarrollo. Se incorporará en versiones futuras.

10.2.- Anexo B. Modelo de base de datos ER (Imagen y modelo .uml adjunto al proyecto)

Dada la dimensión del modelo ER creado, se adjuntará el esquema relacional creado con el modelador ERG de PostgreSQL También se adjuntan el esquema diseñado con ArgoUML, también en los formatos .zargo y .PNG, para poder visualizarlo con buena calidad.

10.3.- Anexo C. Estructura de Vistas JavaFX (Imagen y modelo .uml adjunto al proyecto)

Como bien dijimos en uno de los puntos anteriores, se ha hecho uso de JavaFX para el desarrollo de las interfaces visuales de los usuarios.



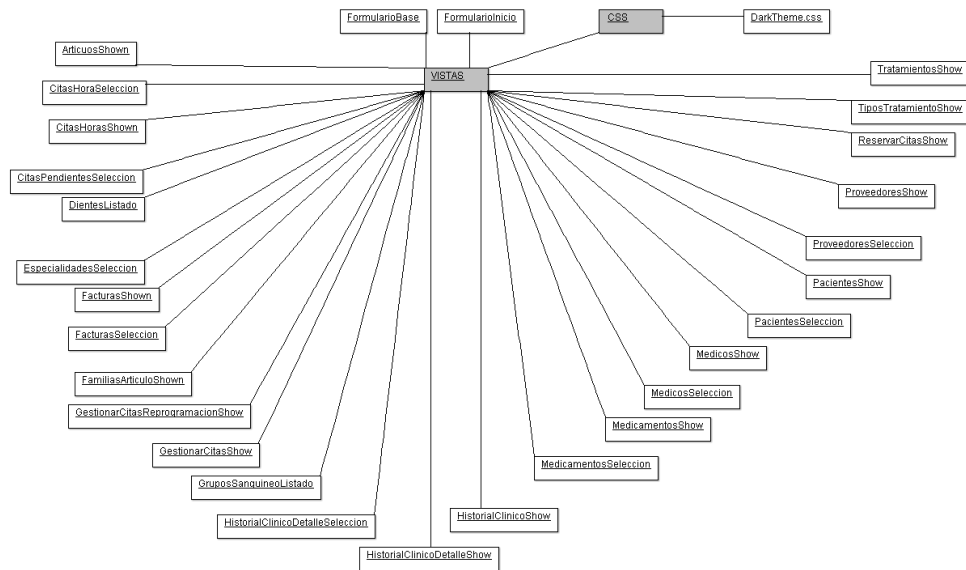
Región de
Murcia

CIFP Carlos III

C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
FAX: 968 320111
30019702@murciaeduca.es
www.cifp-carlos3.es



ENSEÑANZAS COFINANCIADAS POR
FONDO SOCIAL EUROPEO (C.F.E.)

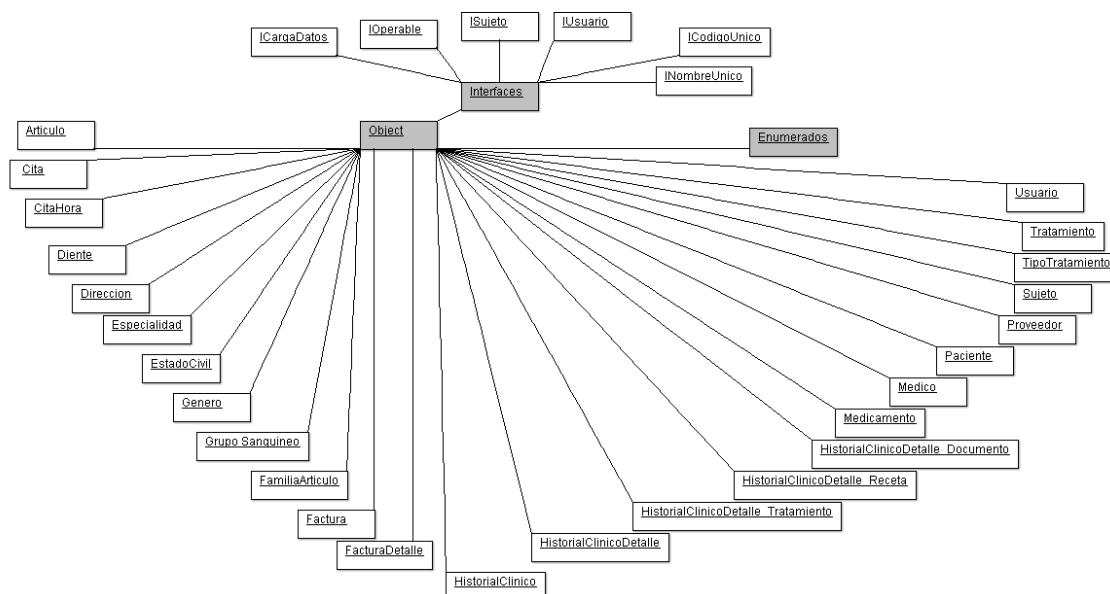




10.4.- Anexo D. Jerarquía de modelos principales(Imagen y fichero .uml adjunto al proyecto)

Uno de los aspectos más importantes son los Modelos. Están situados en la carpeta principal com/modelos/hibernate/. La clase base de la que heredan todos los modelos es la clase Object.

Además se incluyen también las interfaces que se han creado bajo necesidad del desarrollo del proyecto.



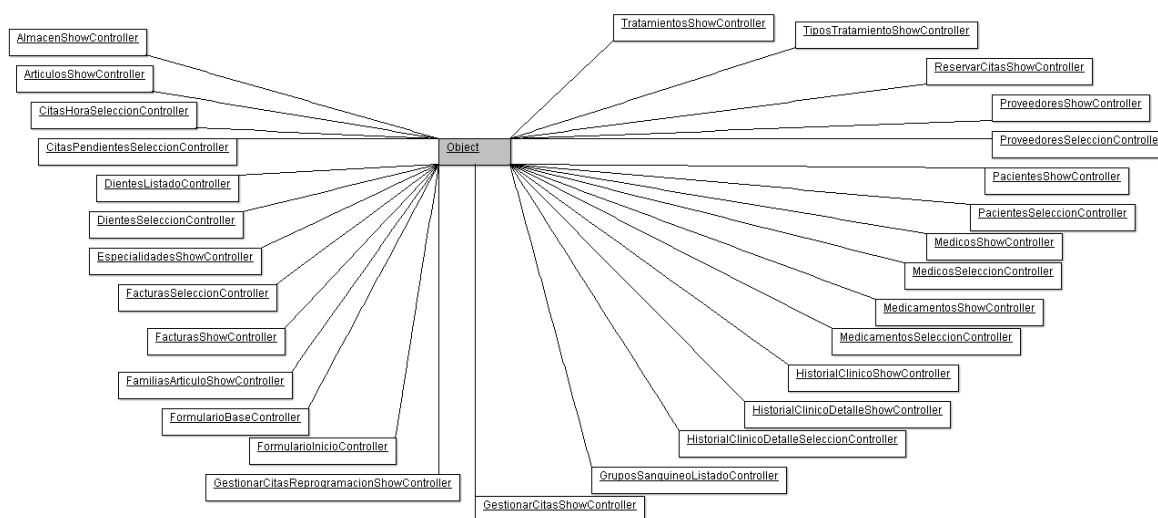
Se adjunta junto al proyecto un fichero .uml y una imagen en png más visible.



10.5.- Anexo E. Jerarquía de controladores creados (Imagen y fichero .uml adjunto al proyecto).

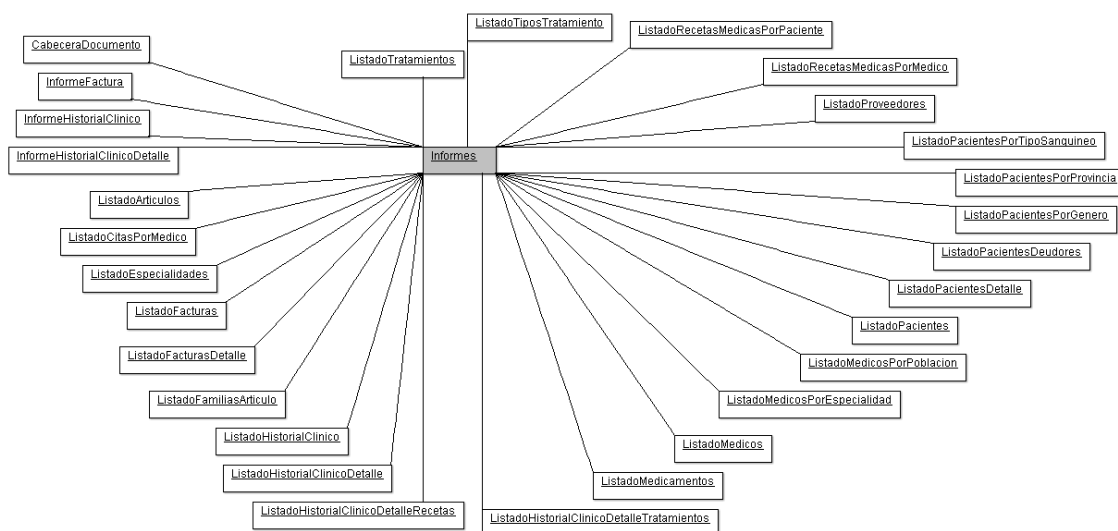
Otro de los aspectos vitales son los controladores. Con ellos representamos toda la lógica de negocio de la aplicación. Todos los controladores heredan también de la clase Object, al igual que cualquier otra clase.

Se adjunta junto al proyecto un fichero .uml y una imagen en png más visible.



10.6.- Anexo F. Jerarquía de informes creados (Imagen y fichero .uml adjunto al proyecto).

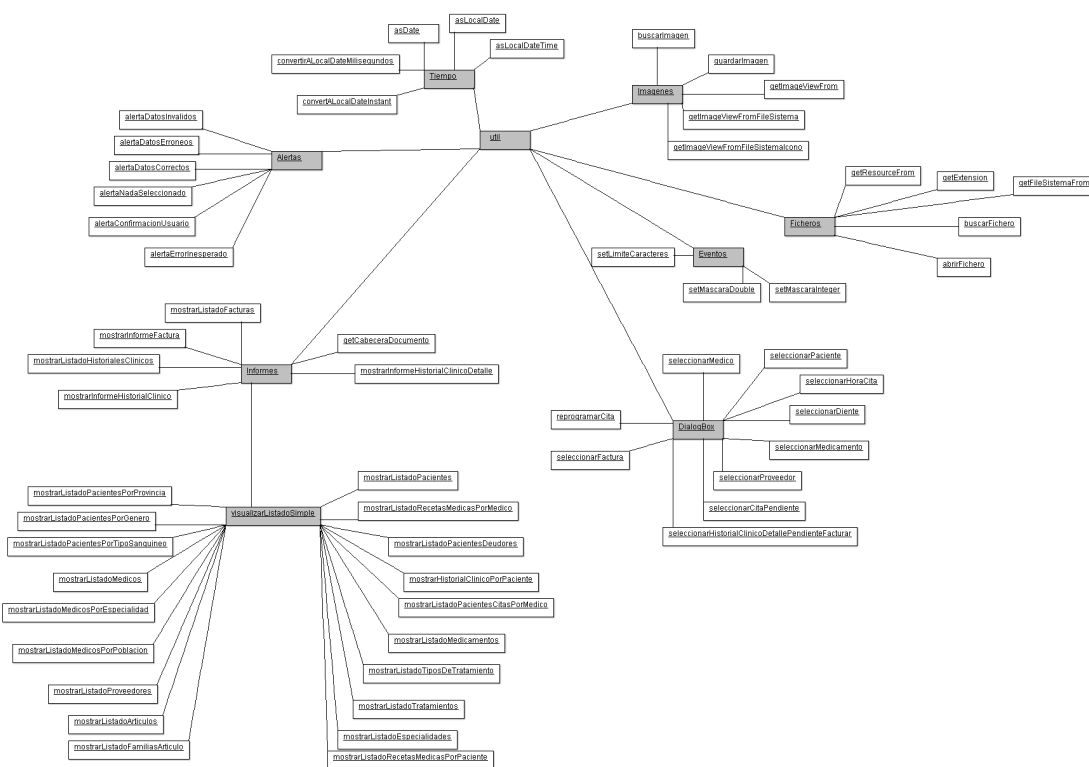
Otro de los aspectos importantes a tener en cuenta son los informes. La necesidad base de ello es porque es necesario reportar la información que se introduce en el sistema. Todos los informes están ubicados en la carpeta /com/informes/.



Se adjunta junto al proyecto un fichero .uml y una imagen en png más visible.

10.7.- Anexo G. Jerarquía del paquete de utilidades creado (Imagen y fichero .uml adjunto al proyecto).

Para la realización del proyecto se ha tenido que crear un paquete que contenga una serie de utilidades que agilicen el desarrollo del mismo, tales como utilidades para fechas, imágenes, visualización de mensajes de aviso al usuario, renderización de informes... etc. Está ubicado en la carpeta /com/util/.



Se adjunta junto al proyecto un fichero .uml y una imagen en png más visible.

10.8.- Anexo H. Manual de Usuario (Se adjunta en el proyecto un documento PDF que será el manual del usuario)

Un requisito indispensable a la hora de realizar un proyecto es crear un manual de usuario, que explique al usuario paso a paso cómo debe operar para utilizar correctamente la aplicación de escritorio.

Se adjunta un manual en .PDF junto al proyecto.