
Herramienta de apoyo al análisis de datos para diagnóstico de enfermedad celiaca

A tool for supporting data analysis related to the diagnosis of the celiac disease

Por

Jose Antonio Garrido Sualdea
Alvaro Noriega Moreno



**UNIVERSIDAD COMPLUTENSE
MADRID**

Grado en Ingeniería del Software
FACULTAD DE INFORMÁTICA

Mercedes García Merayo
**Herramienta de apoyo al análisis de datos para
diagnóstico de enfermedad celiaca**

**A tool for supporting data analysis related to
the diagnosis of the celiac disease**

MADRID, 2019–2020

Agradecimientos

Quiero agradecer a mi familia por estar ahí siempre apoyándome y preocupándose, a mi pareja sentimental por ser un pilar fundamental y sobre todo a mi compañero Álvaro con el cual he compartido casi todos los años de carrera apoyándonos mutuamente en este camino tan duro.

José Antonio Garrido Sualdea

Agradecer a mi familia y a mi pareja por el apoyo recibido y por poder contar con ellos en los buenos y malos momentos durante este largo periodo de formación universitaria. También agradecer a mi compañero José Antonio por haber estado ahí siempre y con el que he trabajado codo con codo durante toda la carrera.

Álvaro Noriega Moreno

Queremos agradecerle a nuestra tutora Mercedes todo el esfuerzo desempeñado. No ha sido un camino fácil y eso es de agradecer ya que eso ha hecho dar lo mejor de nosotros mismos y dar nuestra mejor versión. Gracias por guiarnos y ayudarnos a finalizar esta última etapa de la carrera.

Los autores.

Sobre TEF_LON

TEFLON(CC0 1.0(DOCUMENTACIÓN) MIT(CÓDIGO))ES UNA PLANTILLA DE L^AT_EX CREADA POR DAVID PACIOS IZQUIERDO CON FECHA DE ENERO DE 2018. CON ATRIBUCIONES DE USO CC0.

Esta plantilla fue desarrollada para facilitar la creación de documentación profesional para Trabajos de Fin de Grado o Trabajos de Fin de Máster. La versión usada es la 1.3.

V:1.3 OVERLEAF V2 WITH PDFL_AT_EX, MARGIN 1IN, NO-BIB

Contacto

Autor: DAVID PACIOS IZQUIERO

Correo: dpacios@ucm.es

ASCII: asciifdi@gmail.com

DESPACHO 110 - FACULTAD DE INFORMÁTICA

Índice general

	Página
1. Introducción	2
1.1. Objetivos	3
1.2. Estructura del trabajo	3
2. Introduction	5
2.1. Objectives	6
2.2. Work structure	6
3. Análisis de requisitos	8
3.1. Inicio de sesión y registro	8
3.2. Carga de datos	8
3.3. Creación de consultas	9
3.4. Gestión de consultas	10
4. Entornos de Desarrollo	11
4.1. Frameworks y Librerías	11
4.2. Lenguajes de Programación	13
4.2.1. Java	13
4.2.2. SQL	13

4.2.3. XML	14
4.3. Base de Datos	14
4.4. Control de Versiones	14
4.5. Maquetación y Prototipos	16
5. Diseño e Implementación	17
5.1. Fase de análisis	17
5.2. Fase de investigación	17
5.3. Fase de Instalación y Configuración	18
5.3.1. Crear una cuenta en Google Drive:	18
5.3.2. Instalación de la última versión de java:	18
5.3.3. Actualización de las variables de entorno de windows	18
5.3.4. Configurar la codificación UTF-8 para el intérprete de windows	19
5.3.5. Descargar e Instalar XAMPP	19
5.3.6. Iniciar phpMyAdmin y cargar la BBDD	19
5.3.7. Descargar NetBeans	21
5.3.8. Descargar Eclipse	21
5.4. Fase de Diseño	21
5.4.1. Módulo Inicio y Registro	21
5.4.2. Módulo de cargar datos	23
5.4.3. Módulo Principal	23
5.4.4. Barra menú	29
5.4.5. Módulo Visualizar Consulta y Exportar Datos a Fichero Excel	32
5.5. Implementación de Funcionalidad y Dinamismo	32
5.5.1. Carga Dinámica	32

A tool for supporting data analysis related to the diagnosis of the celiac disease	UCM
<hr/>	
5.5.2. Integración	33
5.5.3. Negocio	35
5.5.4. Presentación	36
6. Contribuciones al Proyecto	38
6.1. José Antonio Garrido Sualdea	38
6.2. Álvaro Noriega Moreno	40
7. Conclusiones	41
8. Conclusions	43
9. Trabajo Futuro	44

Resumen

Este trabajo tiene como objetivo facilitar el filtrado de datos médicos de pacientes que potencialmente pueden sufrir la enfermedad celiaca, para su análisis. La aplicación permite cargar archivos Excel que contienen los datos poco organizados y difíciles de visualizar y transformarlos de cara a su manipulación. También permite almacenar estos datos en tablas de forma organizada, pudiendo añadir más datos según se vayan añadiendo más y clasificarlos en diferentes tablas gracias a la carga dinámica. A parte permite al usuario añadir filtros y restricciones para consultar la información deseada, permitiendo exportar los resultados. Ha sido desarrollada en Java, utilizando una librería que permite la manipulación de ficheros Excel.

Palabras Clave

Base de datos, enfermedad celiaca, patrones de diseño, consultas SQL.

Abstract

The objective of this work is to facilitate the filtering of medical data of patients who may potentially suffer from celiac disease, to analyze them later. The application allows you to upload Excel files that contain data that is poorly organized and difficult to visualize and transform them for manipulation. It also allows you to store this data in an organized way, being able to add more data as more is added and classify it in different tables thanks to the dynamic loading. Besides, it allows the user to add filters and restrictions to consult the desired information, allowing the results to be exported. It has been developed in Java, using a library that allows the manipulation of Excel files.

Keywords

Databases, celiac disease, design patterns, SQL queries.

Capítulo 1

Introducción

La enfermedad celiaca afecta al sistema inmunológico, dañando el intestino delgado debido a la ingestión de alimentos que contienen gluten. Tras la ingestión de gluten el sistema inmunológico responde dañando las vellosidades intestinales que se encargan de absorber los nutrientes y las vitaminas, causando diferentes síntomas. Los síntomas pueden ocurrir en el sistema digestivo o en otras partes del cuerpo, afectando de manera diferente a cada persona [8].

No se conoce la causa exacta de la enfermedad, aunque se sabe que factores ambientales y genéticos influyen en el desarrollo de la enfermedad celiaca. Los genes compatibles con el desarrollo de esta enfermedad son los que pertenecen al sistema HLA, el cual se encarga de ayudar al sistema inmunológico a identificar las proteínas externas a nuestro organismo. Respecto a los factores ambientales destaca la elevada presencia de gluten en la industria alimentaria, debido al uso de cereales. Este factor ambiental podría ser una explicación al notable aumento de casos en los últimos años.

En la investigación llevada a cabo para este proyecto, nos dimos cuenta de que la celiacía, es una enfermedad difícil de identificar. Hay numerosos casos de personas a las que no se les diagnostica esta enfermedad durante mucho tiempo. Es necesario hacer un análisis exhaustivo de los datos disponibles de estos pacientes para tratar de identificar patrones que permitan diagnosticar la enfermedad. Las bases de datos pueden ayudar en esta tarea, ayudando a analizar datos de distintos pacientes sospechosos, para así poder hacer un diagnóstico correcto. Cabe destacar el beneficio que genera en el ámbito de la medicina un uso apropiado de esta tecnología, posibilitando el acceso a información general y/o datos de diferentes pacientes para establecer reglas o patrones que ayuden a determinar causas y efectos.

Decidimos realizar este proyecto ya que a ambos siempre nos ha interesado el campo de la medicina y todo lo que conlleva ayudar a los demás; además, debido que procedemos de la rama de software, nos motiva en gran medida el desarrollo y gestión de proyectos que utilizan bases de datos. Este proyecto para nosotros fue un desafío, ya que no se trataba de una aplicación de una asignatura que se acabara olvidando, sino que va a tener su aplicación en el mundo real, va a servir para ayudar a los demás y se ha diseñado de

tal manera que pueda ser escalable y ampliable en cuanto a su funcionalidad por otros estudiantes o desarrolladores de software.

1.1. Objetivos

El principal objetivo consiste en desarrollar una aplicación intuitiva capaz de utilizar datos médicos de pacientes, con los que a priori no se podía interactuar, realizando conversiones a una base de datos con el fin de poder filtrarlos para su posterior uso en investigación. Aunque se parte de una colección de datos específica, la aplicación esta preparada para que se puedan añadir más datos si en el futuro se requieren, y con ello conseguir una aplicación mas robusta que permita contrastar mas datos clínicos.

Al comienzo del proyecto, teniendo en cuenta la información médica proporcionada por un profesional del estudio de la enfermedad celiaca y nuestro conocimiento de las tecnologías de desarrollo software, determinamos una serie de objetivos.

- Ser capaces de extraer los datos relevantes del archivo proporcionado.
- Construir una base de datos relacional bien organizada.
- Permitir a profesionales médicos que no tienen porque tener conocimientos técnicos para explotar la información disponible de sus pacientes, diseñar consultas para filtrar los resultados deseados.
- Mostrar los datos de manera intuitiva y sencilla.

1.2. Estructura del trabajo

El esquema del presente trabajo se encuentra organizado en X capítulos, cuyo contenido se describe a continuación:

- Capítulo 1 - Introducción: expone la motivación del trabajo realizado junto a los objetivos planteados al inicio del mismo, así como la estructura del documento.
- Capítulo 2 - Introduction: traducción de la introducción al inglés.
- Capítulo 3 - Análisis de requisitos: en este capitulo se presentan los requisitos que debe cumplir la aplicación desarrollada.
- Capítulo 4 - Entornos de Desarrollo: en esta sección se explican los patrones de programación utilizados en el desarrollo de la aplicación.
- Capítulo 5 - Diseño e Implementación: en este capitulo se exponen las fases por las que se han pasado para llevar acabo el desarrollo del proyecto

- Capítulo 6 - Contribuciones al Proyecto : en esta sección se exponen las contribuciones cada integrante del equipo ha aportado al proyecto.
- Capítulo 7 - Conclusiones: en este capítulo se refleja la impresión personal respecto al resultado del proyecto y la experiencia adquirida.
- Capítulo 8 - Conclusions: traducción de las conclusiones al inglés.
- Capítulo 9 - Trabajo Futuro: en esta sección se proponen diferentes ampliaciones posibles para desarrollar en un futuro.

Capítulo 2

Introduction

Celiac disease affects the immune system, damaging the small intestine due to ingesting foods that contain gluten. After ingesting gluten, the immune system responds by damaging the intestinal villi that are responsible for absorbing nutrients and vitamins, causing different symptoms. Symptoms can occur in the digestive system or in other parts of the body, affecting each person differently [8].

The exact cause of the disease is not known, although environmental and genetic factors are known to influence the development of celiac disease. The genes compatible with the development of this disease, belong to the HLA system, which is in charge of helping the immune system to identify proteins outside our body. Regarding environmental factors, the high presence of gluten in the food industry stands out, due to the use of cereals. This environmental factor could be an explanation for the notable increase in cases in recent years.

In the research carried out for this project, we realized that celiac disease is a difficult disease to identify. There are numerous cases of people who are not diagnosed with this disease for a long time. An exhaustive analysis of the available data from these patients is necessary to try to identify patterns that allow the disease to be diagnosed. The databases can help in this task, helping to analyze data from different suspicious patients, in order to make a correct diagnosis. It should be noted the benefit that an appropriate use of this technology generates in the field of medicine, allowing access to general information and / or data from different patients to establish rules or patterns that help determine causes and effects.

We decided to carry out this project since we have always been interested in the field of medicine and all that it entails helping others; Furthermore, since we come from the software branch, we are greatly motivated by the development and management of projects that use databases. This project was a challenge for us, since it was not an application of a subject that would end up being forgotten, but it will have its application in the real world, it will serve to help others and it has been designed in such a way so that it can be scalable and expandable in terms of functionality by other students or software developers.

2.1. Objectives

The main objective is to develop an intuitive application capable of using medical data of patients, with which a priori could not be interacted with, making conversions to a database in order to be able to filter them for later use in research. Although it is based on a specific data collection, the application is prepared so that more data can be added if it is required in the future, and thus achieve a more robust application that allows to contrast more clinical data.

At the beginning of the project, taking into account the medical information provided by a professional in the study of celiac disease and our knowledge of software development technologies, we determined a series of objectives.

- Be able to extract the relevant data from the provided file.
- Build a well-organized relational database.
- Allow medical professionals who do not have to have technical knowledge to exploit the information available from their patients, to design consultations to filter the desired results.
- Show the data in an intuitive and simple way.

2.2. Work structure

The scheme of this work is organized in 9 chapters, the content of which is described below:

- Chapter 1 - Introduction: this section sets out the motivation for the work carried out, the objectives set at the beginning and the structure of the document.
- Chapter 2 - Introduction: translation of the introduction into English.
- Chapter 3 - Requirements analysis: this chapter presents the requirements that the developed application must meet.
- Chapter 4 - Development Environments: this section explains the programming patterns used in the development of the application.
- Chapter 5 - Design and Implementation: this chapter describes the phases that have been passed for the development of the project
- Chapter 6 - Contributions to the Project: this section describes the contributions each team member has made to the project.
- Chapter 7 - Conclusions: this chapter reflects the personal impression regarding the result of the project and the experience acquired.

- Chapter 8 - Conclusions: translation of the conclusions into English.
- Chapter 9 - Future Work: this section describes different possible extensions to develop in the future.

Capítulo 3

Análisis de requisitos

En este capítulo se presentan los requisitos que debe cumplir la aplicación desarrollada.

3.1. Inicio de sesión y registro

Introduciendo el nombre y la contraseña el usuario puede registrarse en la aplicación si no hay ningún usuario registrado con el mismo nombre. Una vez registrado deberá volver a introducir su nombre y contraseña para iniciar la sesión. El usuario registrado puede iniciar la aplicación utilizando su nombre y contraseña.

3.2. Carga de datos

- El usuario que ha iniciado sesión en la aplicación puede elegir continuar sin cargar un nuevo archivo. En este caso se utilizarán los datos almacenados en la base de datos de una sesión anterior.
- El usuario que ha iniciado sesión en la aplicación puede elegir continuar con la carga de un archivo de datos. Ello permite al usuario cargar los datos un fichero local a través de un panel integrado en la interfaz. Este panel permite al usuario navegar entre los directorios locales de su equipo y seleccionar únicamente archivos de formato “.csv”.
- El usuario que ha cargado un archivo, podrá ver una lista de los campos que se han almacenado y añadir campos que no se han almacenado pero que están presentes en el nuevo archivo.

3.3. Creación de consultas

La aplicación permite la creación de consultas a medida.

- Selección de bloque de información: el usuario puede elegir el bloque de datos (tabla) de la base de datos de la que se desea recuperar información.
- Selección de parámetros: se permite al usuario elegir un dato de información del bloque seleccionado previamente para aplicar una restricción.
- Aplicar restricciones numéricas con operadores: el usuario tiene la posibilidad de elegir entre diferentes operadores. Con los operadores $>$, $<$, $=$, $>=$, $<=$, se permite al usuario introducir un valor entero o decimal dependiendo del tipo del parámetro seleccionado. Solo se permite introducir valores decimales en las columnas de tipo double.
- Aplicar restricciones numéricas utilizando un rango: el usuario tiene la posibilidad de elegir un rango de valores numéricos. Se permite al usuario introducir un valores enteros o decimales dependiendo el tipo del parámetro seleccionado. Solo se permite introducir valores decimales en las columnas de tipo double. Se obliga al usuario a establecer ambos límites del rango.
- Aplicar restricciones alfabéticas: se permite al usuario elegir los valores alfabéticos que aplicar en la restricción. Los valores alfabéticos entre los que puede elegir el usuario corresponden a los datos que contiene un parámetro de tipo varchar. El usuario puede elegir uno o varios valores.
- Aplicar restricciones a fechas: para un dato de tipo date se permite al usuario aplicar un rango de fechas. Se obliga al usuario a establecer ambos límites para el rango.
- Eliminar restricción: se permite al usuario eliminar cualquier restricción añadida. Puede seleccionar una o varias restricciones para eliminar.
- Selección de información a mostrar: el usuario puede elegir los datos que desea filtrar.
- Añadir filtro: el usuario puede añadir un filtro siempre que haya seleccionado los datos que desea filtrar. Una vez añadido un filtro no se le permite al usuario volver a seleccionar los datos correspondientes a ese filtro.
- Eliminar filtro: se permite al usuario eliminar cualquier filtro añadido. Puede seleccionar uno o varios filtros para eliminar.
- Ejecutar consulta: el usuario puede ejecutar cualquier consulta siempre que haya diseñado como mínimo un filtro y una restricción.

3.4. Gestión de consultas

El usuario puede realizar diferentes acciones para gestionar las consultas almacenadas.

- Guardar una nueva consulta: al usuario se le permite guardar una consulta introduciendo obligatoriamente una descripción. No se permite introducir la misma descripción a diferentes consultas.
- Abrir consulta: el usuario puede seleccionar una consulta de las que se han guardado previamente en la base de datos.
- Guardar una consulta modificada: se permite al usuario guardar los cambios realizados en una consulta cargada en la aplicación.
- Eliminar una consulta: el usuario puede eliminar la consulta actual si esta ha sido previamente guardada.
- Crear consulta: se pregunta al usuario si desea crear una nueva consulta. En caso afirmativo se proporciona al usuario una consulta vacía, en caso contrario se mantendrá con la consulta actual.
- Regresar al menú de carga: permite al usuario regresar al menú de selección de archivos. El usuario perderá la consulta si no la ha guardado previamente.

Visualización de resultados y exportación

- Exportación de resultados: el usuario puede exportar los resultados. Deberá escribir un nombre y seleccionar en el panel de exploración de archivos la ubicación donde se generará un archivo con formato ".xlsx" (excel) con los resultados. Es obligatorio que el usuario escriba un nombre para el archivo.

Capítulo 4

Entornos de Desarrollo

En esta sección se abordarán los patrones de programación aplicados además de las librerías usadas. También se hablará sobre los diferentes lenguajes de programación utilizados, de la base de datos empleada y del control de versiones.

4.1. Frameworks y Librerías

Se utiliza IBM Rational Software Architect Designer que hace uso de los siguientes patrones.

- Modelo vista controlador(MVC): es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Se trata de un patrón valido para todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo. El Modelo que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia. La Vista, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos de interacción con éste. El Controlador, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.
- Dispatcher: Se desea una vista que maneje una petición y genere una respuesta, gestionando una cantidad limitada de procesamiento de negocio. Cuando la vista es estática o generada de un modelo de presentación existente hay poco procesamiento de negocio antes de generar la vista. Ventajas: Aprovechamiento de marcos y librerías.
- Singleton: En ingeniería de software, singleton, o instancia única, es un patrón de diseño que permite restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase solo tenga una instancia y proporcionar un punto de acceso global a ella. El

patrón singleton se implementa creando en nuestra clase un método que crea una instancia del objeto solo si no existe alguna. Para asegurar que la clase no puede ser instanciada nuevamente se regula el alcance del constructor [11].

- Context: Se desea evitar utilizar información del sistema específica del protocolo fuera de su contexto relevante. Al controlador frontal le pueden llegar datos de entrada en formato específico de un protocolo concreto; incluir datos de este tipo fuera de su contexto puede provocar un fuerte acoplamiento. Utilizamos un objeto contexto para encapsular el estado de una forma independientemente del protocolo para ser compartida por toda la aplicación. Ventajas:
 - mejora la reusabilidad y mantenibilidad.
 - Mejora las pruebas.
 - reduce las restricciones en la evolución de GUIs.
- Factoría abstracta [5]: Es un patrón que define una interfaz para crear familias de objetos relacionados o dependientes sin especificar las clases concretas.
 - Aísla al cliente de las clases concretas.
 - Ayuda a controlar la clase de objetos que crea una aplicación.
 - Permite cambiar fácilmente de familia de productos.
 - Promueve la consistencia entre productos, haciendo que una aplicación utilice objetos de una sola familia a la vez
 - La inclusión de nuevos tipos de producto es difícil.
- Service Application: Se desea centralizar la lógica de negocio a través de distintos componentes de la capa de negocio y servicios. Ventajas:
 - Centraliza lógica del negocio y de workflow reutilizable
 - Mejora la reusabilidad de la lógica de negocio.
 - Evita duplicación de código.
- Transfer: Queremos independizar el intercambio de datos entre capas. Los datos tienen que fluir entre la capa de negocio y las otras capas, dicho flujo debería ser independiente de los mecanismos de representación de cada capa. Utilizamos un objeto transferencia para mover los elementos de datos entre capas. Ventajas:
 - ayuda a independizar capas.
- Application Controller: Se desea centralizar y modularizar la gestión de acciones y de vistas. Se utiliza un controlador de aplicación para centralizar la recuperación e invocación de componentes de procesamiento de peticiones, tales como comandos y vistas. La redirección a vista puede delegarse al controlador frontal o al controlador de la aplicación e incluirse en el action/command. Ventajas:
 - Mejora la modularidad.
 - Mejora la reutilización
 - Mejora la extensibilidad

- Librería Apache POI

Apache POI es una API de java que permite manipular varios formatos de archivos basados en los estándares Open Office XML (OOXML) y el formato de documento compuesto OLE 2 de Microsoft (OLE2). En resumen, permite leer y escribir archivos MS Excel, MS Word y MS PowerPoint utilizando Java. Esta librería se compone de varios elementos:

- **HSSFWorkbook**: es la representación de alto nivel de un libro (Workbook) que será nuestro documento excel. Es el primer objeto que construiremos si vamos a leer o escribir una hoja excel.
- **HSSFSheet**: es la representación de alto nivel de una hoja excel, podemos elegir la hoja de la excel usando el **HSSFWorkbook**.
- **HSSFRow**: representación de celda de una fila de la hoja excel.
- **HSSFCell**: representación de una celda en una fila de una hoja de la excel, la utilizaremos para manejar el contenido de la celda.

4.2. Lenguajes de Programación

Para el desarrollo software de esta aplicación, se ha hecho uso de diferentes lenguajes y sistemas.

4.2.1. Java

Es un lenguaje de programación orientada a objetos, es compatible con diferentes sistemas operativos y posee una gran API bien documentada acerca del propio lenguaje. Utiliza una máquina virtual (JVM). Java es un lenguaje semicompilado que funciona con una máquina virtual, la cual se conoce como la Java virtual machine, o JVM. La máquina virtual de Java es necesaria para poder ejecutar los programas escritos en este lenguaje, y se encuentra disponible para diversos sistemas operativos, como son OS X, Windows, y diversas distribuciones de Linux. Esta máquina virtual lo que hace es interpretar los programas de Java, transformarlos a lenguaje máquina para la PC, y así la PC puede ejecutar el programa. Bien la PC no ejecuta directamente el programa, sino que ejecuta la JVM, y la JVM interpreta el programa precompilado, razón por la cual Java se considera un lenguaje semicompilado [2].

4.2.2. SQL

Es un lenguaje de dominio específico utilizado en programación, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales. Originalmente basado en el álgebra relacional y en el cálculo relacional, SQL consiste en un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje

de control de datos. El alcance de SQL incluye la inserción de datos, consultas, actualizaciones y borrado, la creación y modificación de esquemas y el control de acceso a los datos [12].

4.2.3. XML

El XML es una adaptación del SGML (Standard Generalized Markup Language), un lenguaje que permite la organización y el etiquetado de documentos. Esto quiere decir que el XML no es un lenguaje en sí mismo, sino un sistema que permite definir lenguajes de acuerdo a las necesidades. Cuando se desarrolla un programa con interfaz gráfica es necesario organizar todos los documentos y XML es de gran ayuda en estos casos: permite agruparlas, etiquetarlas, especificar su ubicación y relacionarlas con otros datos, según las necesidades de los diseñadores [1].

4.3. Base de Datos

Nos hemos decantado por el uso de MySQL [3] como sistema gestor de base de datos debido a que estamos familiarizados con su entorno y además por la multitud de ventajas que tiene:

- Es gratuita
- Es muy fácil de usar
- Es una base de datos muy rápida
- Utiliza varias capas de seguridad. Contraseñas encriptadas, derechos de acceso y privilegios para los usuarios.
- Pocos requerimientos y eficiencia de memoria. Tiene una baja fuga de memoria y necesita pocos recursos de CPU o RAM.
- Es compatible con Linux y Windows.

Aquí se puede ver reflejado el esquema de la base de datos 4.1 4.2

4.4. Control de Versiones

Para llevar a cabo el control de versiones nos hemos servido de la herramienta Google Drive.

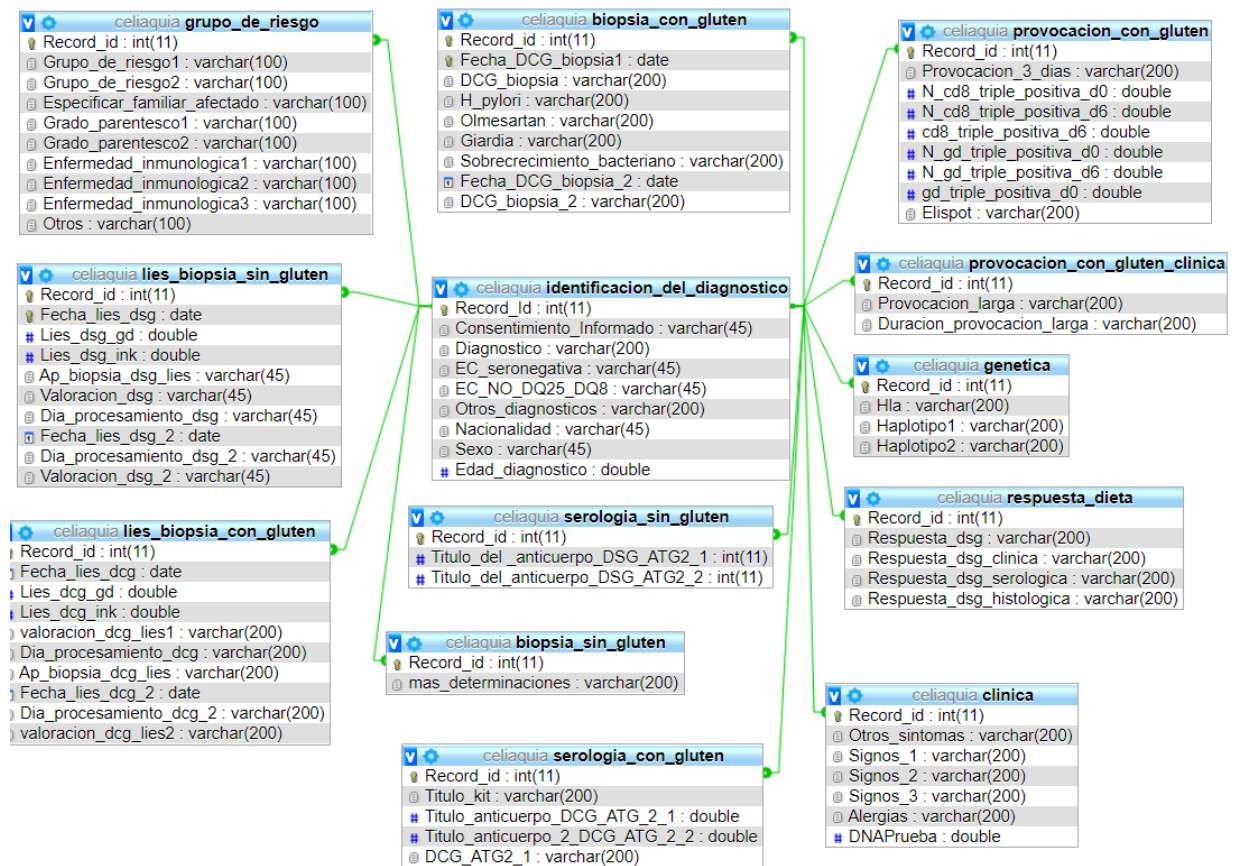


Figura 4.1: Vista gráfica del esquema de la BBDD parte 1

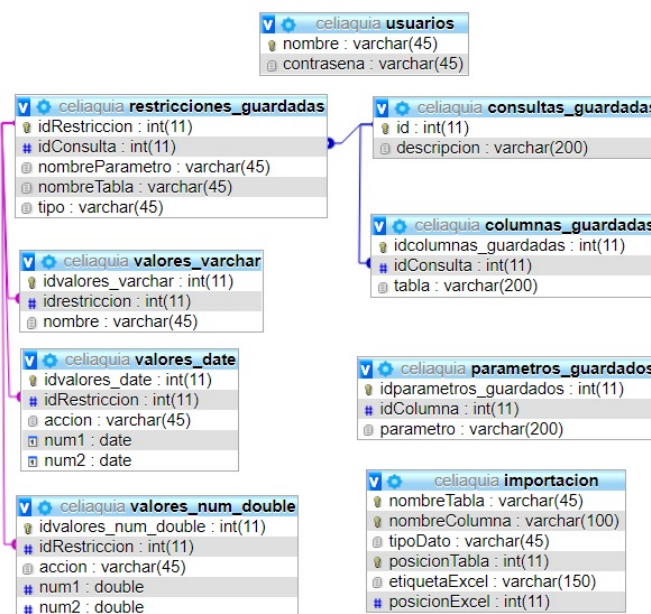


Figura 4.2: Vista gráfica del esquema de la BBDD parte 2

Google Drive es un servicio de alojamiento de archivos que fue introducido por la empresa estadounidense Google el 24 de abril de 2012 [6].

Permite guardar cualquier tipo de archivo. Se puede acceder a tus archivos de Drive desde cualquier smartphone, tablet o computadora. Vayas donde vayas, tus archivos te siguen. A su vez, se pueden compartir archivos y carpetas. Es posible invitar fácilmente a otros usuarios a ver y descargar todos los archivos que quieras, o puedes invitarlos a que trabajen en ellos, sin necesidad de enviar archivos adjuntos por correo electrónico [7].

4.5. Maquetación y Prototipos

Para la realización de los diferentes prototipos y diseños gráficos del software, nos hemos ayudado de la herramienta NetBeans. NetBeans es un framework que simplifica el desarrollo de aplicaciones para Java Swing. El paquete de NetBeans IDE para Java SE contiene lo que se necesita para empezar a desarrollar plugins y aplicaciones basadas en la plataforma NetBeans; no se requiere un SDK adicional.

La plataforma ofrece servicios reusables comunes para las aplicaciones de escritorio, permitiendo a los desarrolladores centrarse en la lógica de sus aplicaciones [9].

Capítulo 5

Diseño e Implementación

En este capítulo se tratarán las fases por las que hemos tenido que pasar a lo largo del proyecto para poder llevarlo a cabo. Primero se habla sobre la fase de análisis, después la fase de investigación y para finalizar los programas que tuvimos que instalar y configurar para desarrollar la aplicación, junto con la fase de diseño llevada a cabo.

5.1. Fase de análisis

Durante la fase de análisis del proyecto planteamos los diferentes problemas que queríamos resolver. Nuestro objetivo principal era mejorar la gestión de los datos recogidos en un estudio de investigación en celiaquía. Los datos provenían de la exportación de una base de datos mal organizada, la cual exportaba dichos datos a un archivo Excel con columnas concatenadas sin orden. Por ello decidimos realizar una aplicación que permitiese filtrar los datos y ordenarlos para visualizarlos de manera intuitiva permitiendo crear archivos Excel con la información más compacta y estructurada.

5.2. Fase de investigación

Tras analizar el volumen de datos y los errores y celdas vacías presentes en estos, lo que nos dificultaría mucho la automatización de la carga de estos datos en nuestra base de datos, decidimos utilizar tecnologías que dominábamos y hemos aprendido y utilizado durante nuestro periodo de formación universitaria. Se tuvo que investigar cómo manejar en nuestra aplicación archivos Excel u otro tipo de hojas de cálculo, por lo que se buscó información sobre el uso de la librería Apache POI, concretamente XSSFWorkbook, que permite trabajar con ficheros Excel en formato .xlsx en Java. Además, tuvimos que revisar los conocimientos que teníamos sobre JavaSwing y profundizar en los frameworks de desarrollo de interfaces, incluido el lenguaje y framework de la base de datos. También se tuvo que hacer una revisión a nivel teórico-práctico de los patrones de desarrollo

software que se han utilizado para la implementación de la aplicación

5.3. Fase de Instalación y Configuración

En esta fase de instalación y configuración preparamos todo el software necesario para llevar a cabo el desarrollo de la aplicación. A continuación detallamos los pasos a seguir.

5.3.1. Crear una cuenta en Google Drive:

Se crearon dos carpetas, una para el control de versiones en caso de tener que actualizar un modulo del proyecto, y otra carpeta en la que se realizaban copias de seguridad de los proyectos de forma íntegra con independencia de los cambios que se hicieran, fácil de usar y versátil debido a que Google Drive soporta todo tipo de archivos.

5.3.2. Instalación de la última versión de java:

Gracias a la tecnología Java, podemos trabajar en un entorno informático mucho más seguro. Actualizando a la versión de Java más reciente, se mejoró la seguridad del sistema; las versiones anteriores no incluyen las últimas actualizaciones de seguridad [4].

5.3.3. Actualización de las variables de entorno de windows

Para poder ejecutar el archivo ejecutable desde la consola de windows(.jar) es necesario añadir al PATH de las variables de entorno de windows la ruta de Java para poder compilar el archivo de la aplicación. Para ello nos dirigiremos al explorador de archivos de Windows10 y sobre “este equipo” hacemos click sobre el botón derecho y pulsamos la opción “propiedades”. A continuación se nos abrirá una nueva ventana, donde nos dirigiremos con el ratón sobre la columna de la izquierda y pulsaremos sobre “Configuración avanzada del sistema”. Dentro de la pestaña de opciones avanzadas haremos click sobre el botón “Variables de entorno”. En el apartado de “Variables de usuarios para Admin” seleccionaremos la celda donde esta la variable “Path” y le pulsaremos el botón “Editar”. Después comprobamos que no este escrita ya la ruta donde se encuentra Java ya instalado, y en caso de no ser así tendríamos que pulsar el botón “Nuevo” y pegar la ruta donde esta Java instalado, que deberemos buscar en los archivos de programa, carpeta Java, en la carpeta jdk y dentro de la carpeta bin.

5.3.4. Configurar la codificación UTF-8 para el intérprete de windows

Debido a que tratamos con datos de una BBDD, en la cual se pueden incluir caracteres raros, como tildes, es necesario incluir en la configuración regional del sistema la versión beta que incluye la codificación UTF-8. Para ello seguiremos los siguientes pasos: En la barra de búsqueda de Windows10 buscaremos “configuración”. Una vez abierto tendremos que pinchar sobre “Hora e idioma”. Se nos abrirá una ventana nueva donde tendremos que pinchar en el apartado “Región e idioma” que se encuentra localizado en la columna izquierda de la ventana. Una vez realizado estos pasos debemos de pinchar en “Configuración de idioma administrativo”. Se abrirá una ventana nueva, donde dentro de la pestaña “Administrativo” deberemos pulsar sobre el botón “Cambiar configuración regional del sistema”. Para finalizar en la ventana que se nos abra debemos seleccionar la casilla “Versión beta: Use UTF-8 Unicode para la compatibilidad de idioma en todo el mundo”.

5.3.5. Descargar e Instalar XAMPP

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl.

Te permite instalar de forma sencilla Apache en tu propio ordenador, sin importar tu sistema operativo (Linux, Windows, MAC o Solaris), y lo mejor de todo es que su uso es gratuito. XAMPP es una herramienta de desarrollo que te permite probar tu trabajo (páginas web o programación por ejemplo) en tu propia computadora sin necesidad de tener que acceder a internet [10]. Podemos descargarlo gratuitamente desde la página del proyecto Apache Friends. Una vez descargado el paquete, puedes ejecutar el archivo .exe haciendo doble clic en él. Se recomienda desactivar el programa antivirus hasta que todos los componentes estén instalados, ya que puede obstaculizar el proceso de instalación. Iniciamos el asistente de instalación y seguimos la configuración estándar para un servidor de prueba local, con la cual se instalan todos los componentes disponibles. Durante el proceso de instalación es frecuente que el asistente avise del bloqueo de Firewall. En la ventana de diálogo puedes marcar las casillas correspondientes para permitir la comunicación del servidor Apache en una red privada o en una red de trabajo. Recuerda que no se recomienda usarlo en una red pública [13]. A continuación se muestra en la figura 5.1 la ventana que se despliega al ejecutar Xampp. Deberemos de hacer click sobre los botones “start” de Apache y MySQL.

5.3.6. Iniciar phpMyAdmin y cargar la BBDD

Debemos iniciar un navegador e introducir la dirección <http://localhost/phpmyadmin/> (debemos recordar que no necesitamos conexión a internet, ya que trabajamos con una

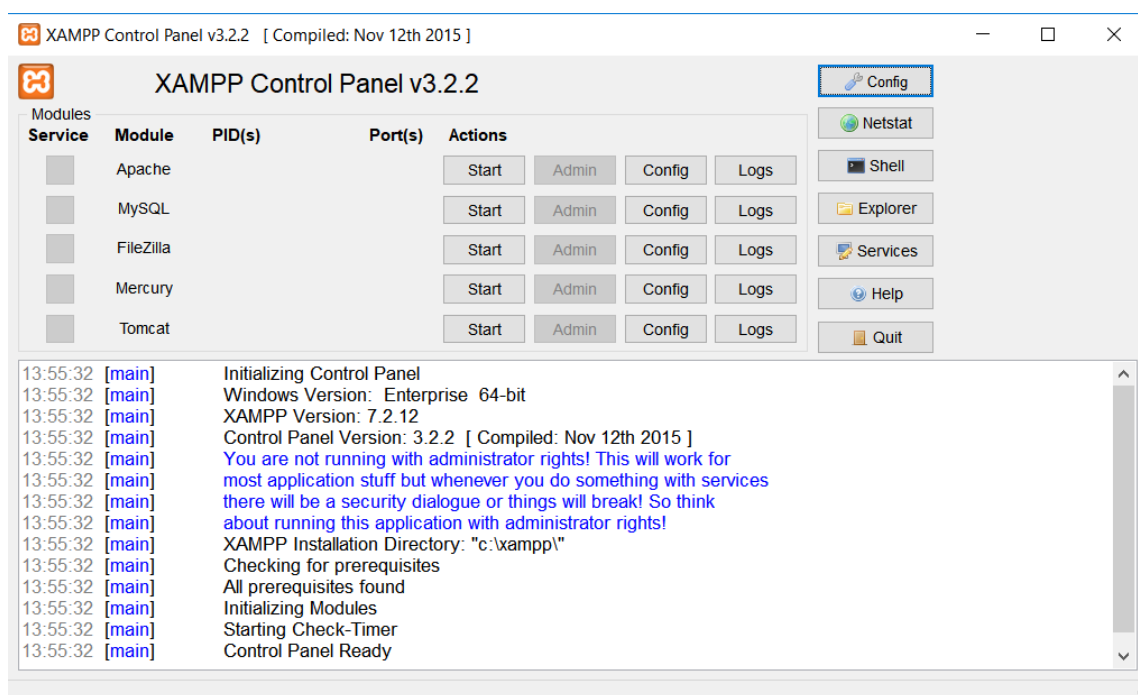


Figura 5.1: Vista gráfica de la ejecución de Xampp

BBDD en local). Aquí procedemos a la creación de la base de datos, como se puede ver en la figura 5.2. La interfaz es bastante clara y facil de entender. En la parte izquierda pulsaríamos en “nuevo” y bastaria con introducir el nombre y la codificación de la BBDD para empezar a crear tablas de forma rápida y cómoda.

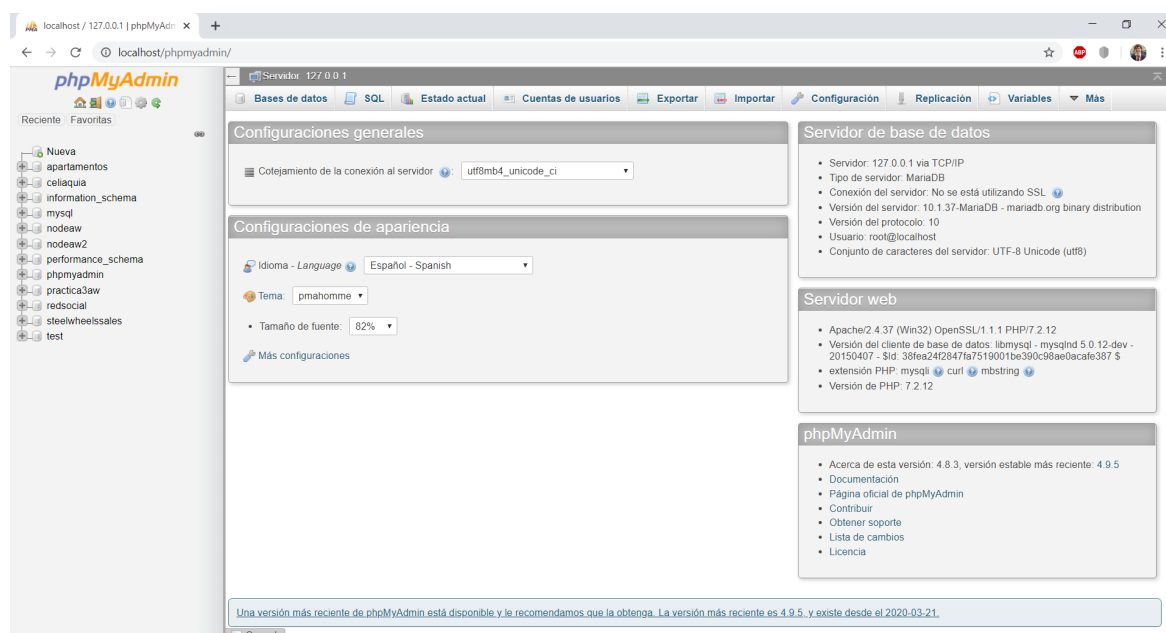


Figura 5.2: Vista gráfica phpMyAdmin

5.3.7. Descargar NetBeans

Hacemos uso de Netbeans para diseñar la parte gráfica de la aplicación con Java Swing. Es una herramienta potente ya que permite realizar el diseño de Frames utilizando la tecnología *drag and drop*. Para ello simplemente tendremos que dirigirnos a la página oficial de NetBeans y descargarnos la última versión. Procederemos a seguir el asistente de instalación y seguiremos los pasos recomendados que vienen por defecto para ello. Una vez instalado, para empezar a diseñar, tendremos que crear un proyecto, seleccionando como categoría: “Java Application”. Una vez generado el proyecto simplemente nos tendremos que dirigir a la pestaña de “Diseño” y empezar a trabajar. A la izquierda se encuentran todos los componentes de Swing, aunque en algunos casos, como en JCalendar, tuvimos que importarlos.

5.3.8. Descargar Eclipse

Por último tuvimos que hacer uso de esta herramienta para desarrollar código en Java debido a que es el software que usamos por defecto en la facultad y con el que más horas hemos trabajado y estamos más familiarizados. El proceso de instalación fue sencillo, simplemente basta con dirigirse a la página oficial, descargarse la última versión de Eclipse y seguir el asistente de instalación. Para la creación del proyecto bastaba con crear un proyecto Java sobre un workspace ya creado e ir añadiendo clases. Este era el momento de establecer la conexión del programa con la BBDD. Para ello hemos hecho uso de una conexión JDBC donde los DriverManager se encargan de gestionar la conectividad con la BBDD

5.4. Fase de Diseño

En esta fase llevamos a cabo diferentes diseños gráficos para ver como sería el aspecto de la aplicación mediante el editor gráfico de NetBeans. Para ello realizamos varios bocetos hasta poder reunir un marco global para del diseño. A partir de entonces, se fueron desarrollando la sucesivas vistas que mostraremos a continuación.

5.4.1. Módulo Inicio y Registro

En la figura 5.3 se muestra la pantalla de inicio. Se presenta un formulario con dos campos a rellenar, el nombre y la contraseña y dos botones para iniciar sesión o para registrarse en la aplicación.

En el caso de querer iniciar sesión se deberá introducir el nombre de usuario y la contraseña. Estos datos se cotejarán con la BBDD y en caso de estar registrado, se procederá

al acceso a la aplicación. En caso de introducir mal los datos se mostrará un mensaje de advertencia.

En el caso de querer registrarse aparecerá la pantalla que se muestra en la figura 5.4. Después el proceso es idéntico, debiendo introducir el nombre nuevo y la contraseña nueva deseada. En caso de haber otro usuario registrado con el mismo nombre se mostrará un mensaje de advertencia. En caso de ser exitoso, se mostrará un mensaje de bienvenida y se redirigirá a la pantalla de inicio donde se podrá iniciar sesión.

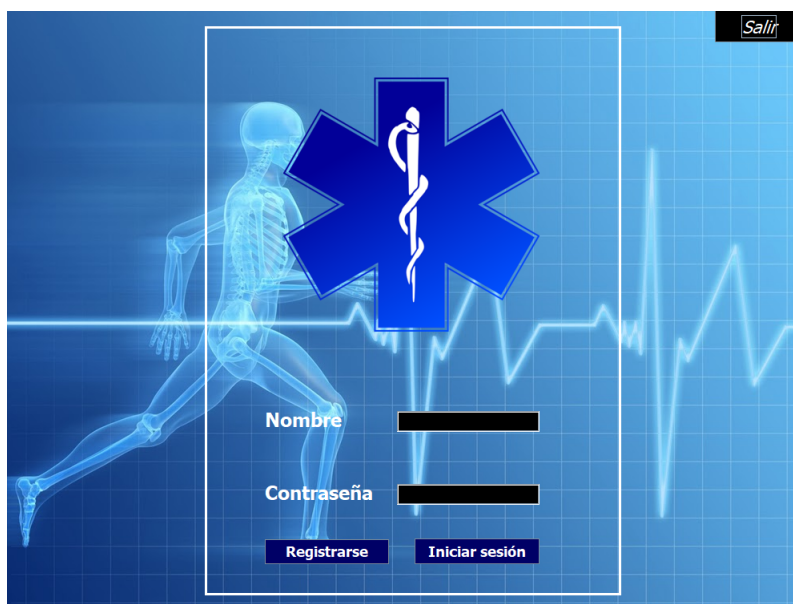


Figura 5.3: Vista gráfica Módulo Inicio Sesión



Figura 5.4: Vista gráfica Módulo Registro

5.4.2. Módulo de cargar datos

La pantalla que se muestra en la figura 5.5 es la que se muestra cuando se inicia sesión en la aplicación. Tenemos dos opciones, seleccionar una BBDD a cargar o continuar con la que esté ya cargada en la BBDD.

Si deseamos cargar otra BBDD con diferentes datos, solamente deberemos seleccionar un archivo.

Si deseamos continuar con los datos que ya estaban cargados previamente en la BBDD, simplemente deberemos de pulsar en el botón “Continuar sin cargar nuevo archivo”. Si hemos seleccionado un archivo a cargar, tras la carga del archivo, nos aparecerá la

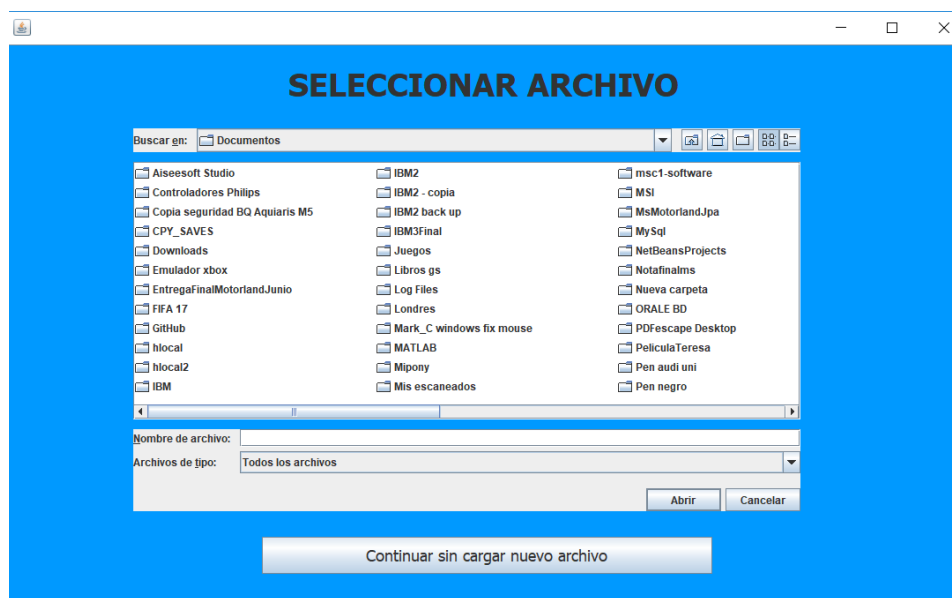


Figura 5.5: Vista gráfica Seleccionar Archivo

pantalla que se muestra en la figura 5.6 con un recuadro con los nombres de las columnas de los datos que se han almacenado. Aquí podremos añadir otra columna a almacenar, indicando el nombre de la cabecera en el Excel, el nombre que daremos al campo en la tabla, la tabla a la que pertenece y el tipo del dato. Al pulsar el botón “añadir” se añadirá el campo y se regresará a esta pantalla por si queremos añadir otro campo. Cuando no deseemos añadir ningún campo más, pulsaremos en el botón “finalizar”.

5.4.3. Módulo Principal

En la figura 5.7 se muestra el corazón de la aplicación, donde conseguimos en una misma pantalla reunir todo el proceso para poder realizar las consultas deseadas. A continuación se irán detallando las diferentes partes para poder explicar su funcionalidad de forma precisa.

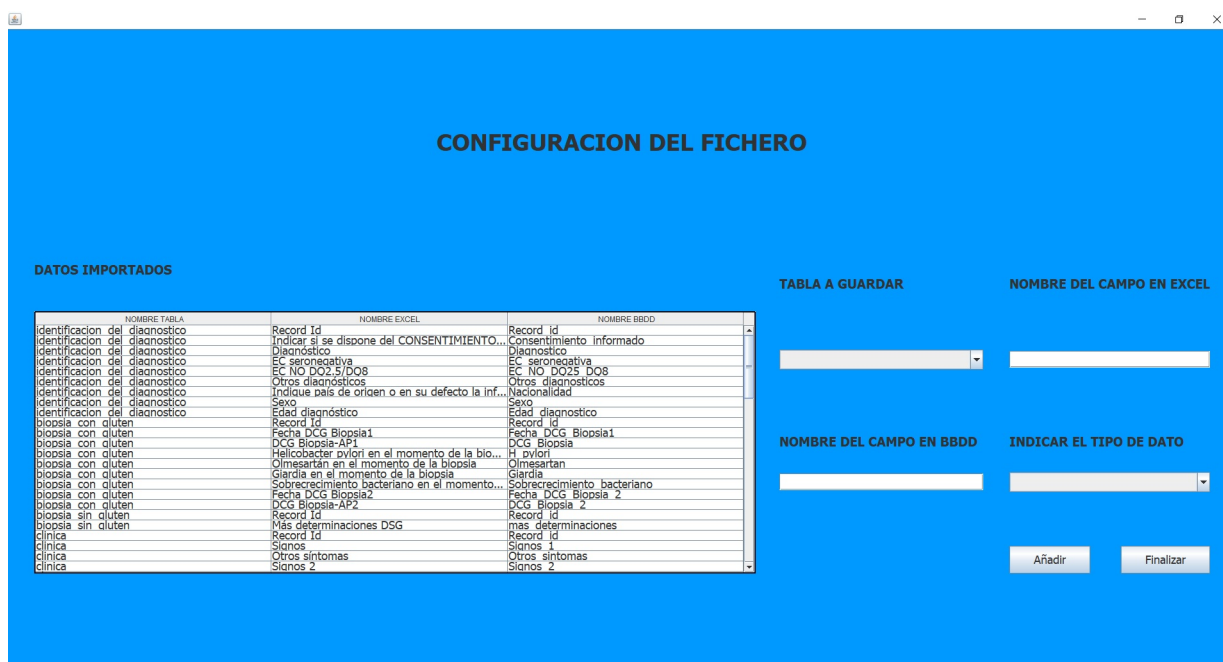


Figura 5.6: Vista gráfica Configuración del fichero

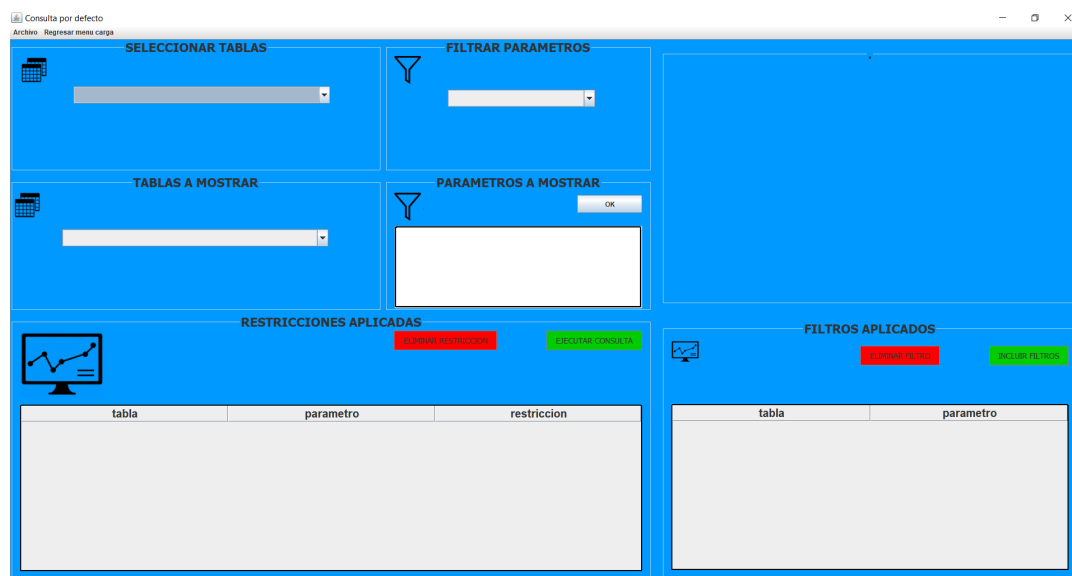


Figura 5.7: Vista gráfica Pantalla Principal

Módulo Seleccionar Tablas

Se deberá elegir sobre el bloque sobre el que queremos realizar una consulta (figura 5.8).

Módulo Filtrar parámetros

En este caso se seleccionarán los parámetros de interés del bloque seleccionado previamente sobre el que se podrá establecer alguna restricción.

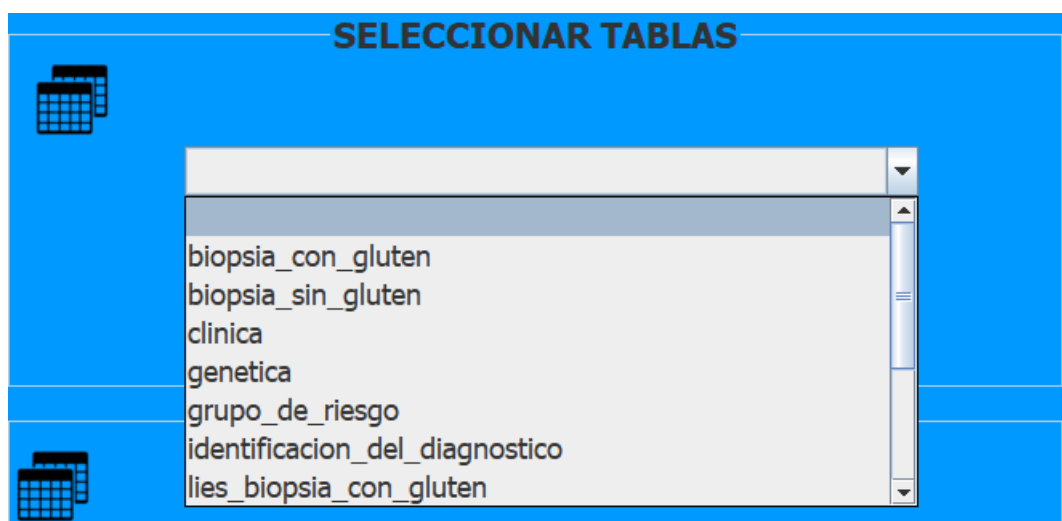


Figura 5.8: Vista gráfica Seleccionar Tablas

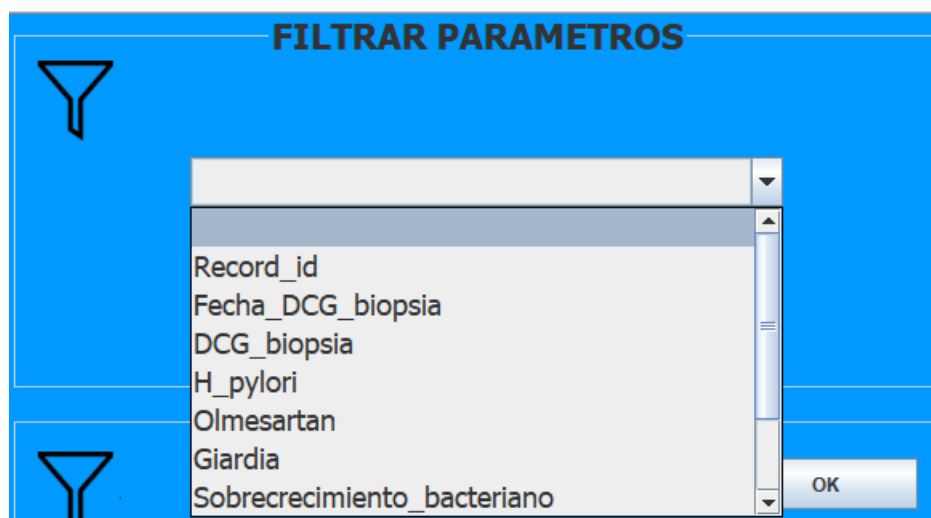


Figura 5.9: Vista gráfica Filtrar Parámetros

Módulo Aplicación de Restricciones Numéricas

El panel de la figura 5.10 se activará en el la superior derecha cuando uno de los parámetros seleccionados a filtrar corresponda con un valor numérico. Podremos filtrar aquellos valores que queramos que sean mayores, menores o distintos del valor dado. También podremos acotar un rango de búsqueda si seleccionamos el operador “Rango”; el primer valor del intervalo debe ser menor que el segundo valor del intervalo.

Módulo Aplicación Restricciones Alfabéticas

El panel de la figura 5.11 se activará en el la superior derecha cuando uno de los parámetros seleccionados a filtrar corresponda con valores tipificados. Por lo tanto, solamente tendremos que seleccionar todas aquellas opciones de interés.

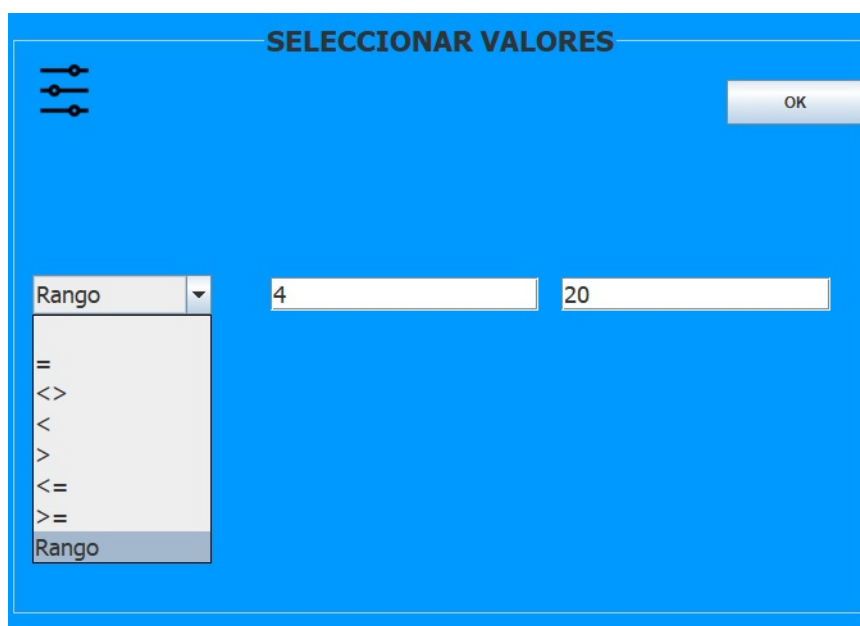


Figura 5.10: Vista gráfica restricciones numéricas

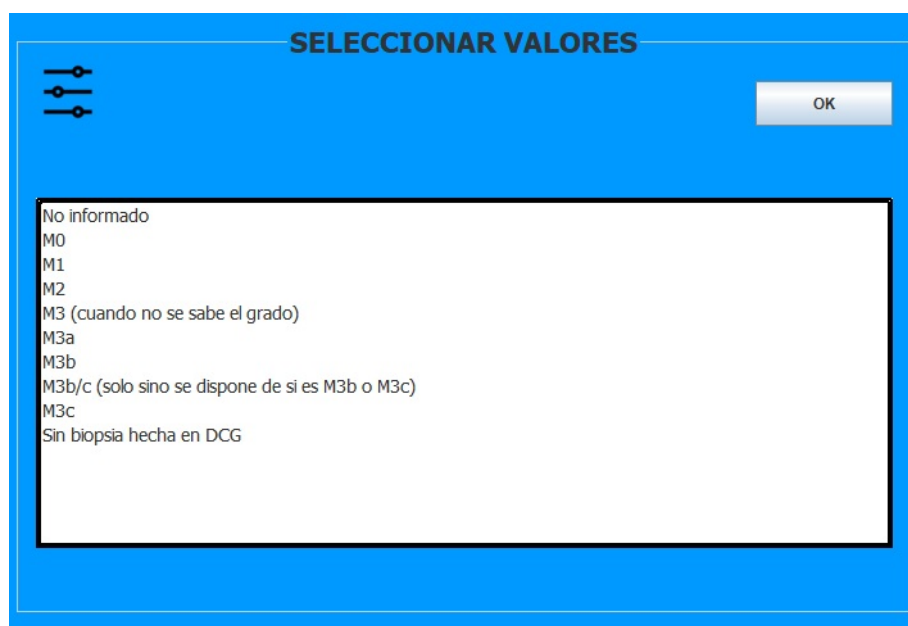


Figura 5.11: Vista gráfica restricciones alfabéticas

Módulo Aplicación Restricciones Fecha

El panel de la figura 5.12 se activará en la parte superior derecha cuando uno de los parámetros seleccionados a filtrar corresponda con fechas. Podremos filtrar aquellas fechas que queramos que sean mayores, menores o distintas de la fecha dada, seleccionando el año el mes y el día. También podremos acotar un rango de búsqueda para las fechas si seleccionamos el operador “Rango”; el primer valor del intervalo debe ser menor que el segundo valor del intervalo. En la figura 5.12 tenemos un ejemplo de selección de rango.



Figura 5.12: Vista gráfica restricciones Fecha

Módulo datos seleccionados

En esta parte de la interfaz 5.13 se deberán seleccionar los bloques de datos cuyos parámetros queremos filtrar, es decir, que resultados queremos que se muestren. Una vez seleccionado el bloque correspondiente del desplegable aparecerán sus parámetros seleccionables como se muestra en la figura 5.14.

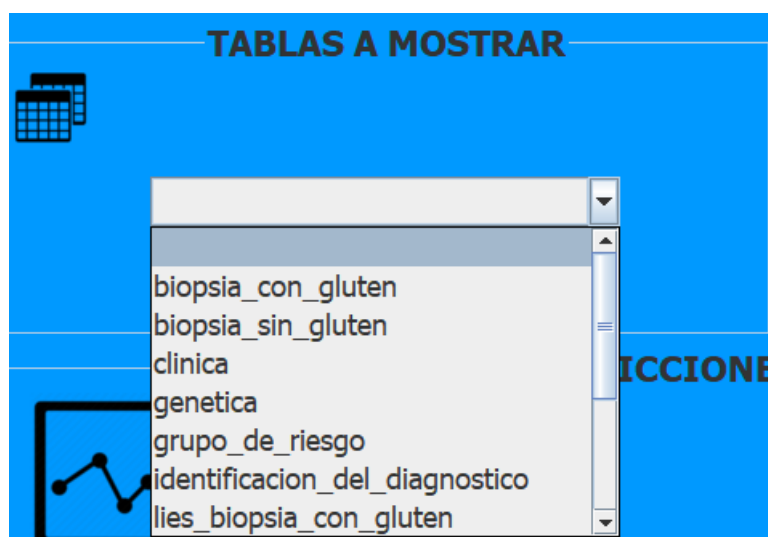


Figura 5.13: Vista gráfica bloques de parámetros a mostrar

Módulo Parámetros a Mostrar

En esta pantalla que reflejamos en la figura 5.14, como bien hemos indicado antes, deberemos seleccionar los parámetros que deseamos que aparezcan en la salida de información, pudiendo seleccionar un parámetro o varios a la vez si corresponden al mismo bloque. Una vez seleccionado, se hará click en el botón Ok y automáticamente aparecerán en la selección de parámetros seleccionados como se muestra en la figura 5.15.

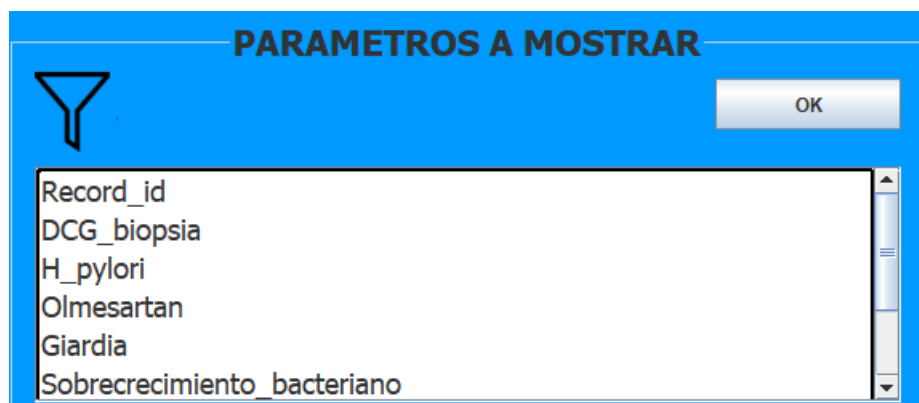


Figura 5.14: Vista gráfica parámetros a mostrar

Módulo Filtros Aplicados

Aquí se reflejarán los datos que se mostrarán al ejecutar la consulta. Primero aparecerá el nombre del bloque y luego el parámetro o la lista de parámetros escogidos. Tenemos la opción de poder borrar los filtros aplicados cuando queramos, tan solo deberemos seleccionar las filas deseadas y seleccionar esta opción. También se ha añadido un botón realmente útil, que nos permite agregar aquellos filtros que se encuentren en el bloque de restricciones aplicadas, ahorrando tiempo y aportando comodidad.

Módulo Restricciones Aplicadas

En la figura 5.16 se muestra la interfaz en la que se lleva a cabo el registro de las restricciones aplicadas de los datos numéricos, fechas y datos tipificados, clasificados correctamente por nombre de bloque y parámetro. Podremos tener la opción de eliminar cualquier restricción deseada. Tan solo deberemos seleccionar las filas deseadas y hacer click en el botón ELIMINAR RESTRICCIÓN. Por último para ejecutar la consulta deseada una vez que hayamos aplicado las restricciones y filtros deseados, tan solo deberemos hacer click sobre el botón EJECUTAR CONSULTA. El resultado aparecerá en otra pantalla.

tabla	parametro
biopsia con gluten	Record id
clinica	Sintomas abreviados1
identificacion del diagnostico	Sexo
biopsia sin gluten	Fecha DSG biopsia

Figura 5.15: Vista gráfica Filtros Aplicados

tabla	parametro	restriccion
biopsia con gluten	Record id	> 7
clinica	Sintomas abreviados1	[Dispepsia]
identificacion del diagnostico	Sexo	[Mujer]
biopsia sin gluten	Fecha DSG biopsia	Rango 2018-05-22 2020-05-22

Figura 5.16: Vista gráfica Restricciones Aplicadas

5.4.4. Barra menú

En la parte superior de la aplicación, en la pantalla principal, podemos la barra de menú, tal como se muestra en la figura 5.17, con las opciones *abrir*, *guardar*, *guardar como*, *crear una nueva consulta* o en su defecto *regresar al panel de carga de ficheros de BBDD*.

Módulo Abrir Consulta Guardada

En este módulo tenemos la opción de cargar una consulta guardada previamente. Se refleja en la figura 5.18 y como se puede apreciar, aparecerá un lista con todas las consultas almacenadas, de las cuales podremos elegir una. Automáticamente se cargará esa consulta

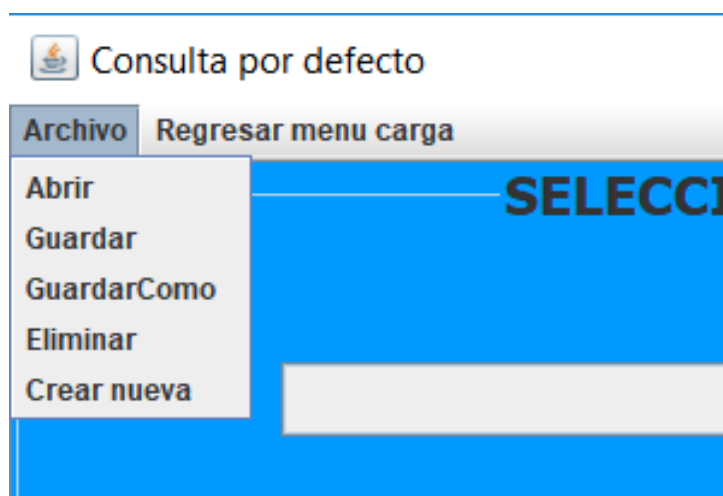


Figura 5.17: Vista gráfica Restricciones Aplicadas

en la sesión actual y podremos trabajar con ella, ya bien sea para consultar datos o para modificarla.



Figura 5.18: Vista gráfica Abrir Consulta Guardada

Módulo Guardar Consulta

Este módulo se encarga de almacenar la consulta que estemos generando actualmente en la sesión. Si partimos de una consulta por defecto y todavía no hemos guardado nada, es decir, si es la primera vez que vamos a guardar, automáticamente saltará la ventana de “Guardar Como” 5.19, cuyo funcionamiento explicaremos a continuación. Si la consulta ya

está almacenada en la BBDD y quedan cambios pendientes de guardar, automáticamente se refrescará la información en la BBDD.

Módulo Guardar Como archivo

En este módulo, como bien hemos hecho referencia a ello antes, nos permitirá almacenar la consulta actual en la BBDD. Tendremos dos casuísticas. La primera de ellas consiste en querer guardar por primera vez la consulta creada, introduciendo la descripción correspondiente y pulsando el botón de OK. La segunda casuística sería la de almacenar esa consulta específica con otro nombre y en dicho caso, una vez escrita la nueva descripción, y pulsado el botón de OK, trabajaremos en la sesión actual con la nueva descripción.



Figura 5.19: Vista gráfica Guardar Consulta

Módulo Eliminar Archivo

Este módulo nos permitirá borrar de la BBDD la consulta actual que está en sesión. La aplicación mostrará un mensaje de advertencia preguntándonos si estamos seguros de querer borrarlo. En caso de haber iniciado una nueva consulta por defecto y no la hayamos guardado en la BBDD todavía, la aplicación lanzará un mensaje de error anunciando que no podemos borrar algo que no esta guardado.

Módulo Crear Nueva Sesión

Esta opción nos brindará la oportunidad de crear una consulta nueva. Siempre que seleccionemos este botón en el desplegable del menú, nos preguntará si estamos seguros de querer empezar una nueva consulta y si hemos guardado los cambios anteriores.

Módulo Regresar al menú de carga

Con esta opción tendremos la posibilidad de regresar al menú de carga (figura 5.5), si deseamos cargar en la BBDD los datos de otros pacientes o de diferentes fechas.

5.4.5. Módulo Visualizar Consulta y Exportar Datos a Fichero Excel

La figura 5.20 muestra la pantalla que aparecerá cuando ejecutemos la consulta de la figura 5.16. Se mostrarán los resultados con sus correspondientes bloques y parámetros. Tendremos la posibilidad de exportar dichos datos a un fichero Excel seleccionando la ubicación del archivo.



Record_id	Fecha_DCG_biopsia	Sexo
1	2018-01-22	Mujer
2	2018-01-23	Mujer
13	2019-03-20	Mujer
16	2016-05-11	Mujer
17	2016-09-19	Mujer
18	2017-02-15	Mujer
27	2016-03-30	Mujer
30	2016-06-01	Mujer
49	2016-02-05	Mujer
53	2016-10-27	Mujer
65	2018-01-30	Mujer
105	2016-01-19	Mujer
129	2017-03-13	Mujer
152	2017-02-14	Mujer
171	2015-06-01	Mujer
198	2016-01-21	Mujer
206	2015-06-09	Mujer
219	2017-06-20	Mujer
222	2016-10-11	Mujer
227	2016-02-29	Mujer
556	2016-02-18	Mujer

Figura 5.20: Vista gráfica Consulta y Exportar Datos a Fichero Excel

5.5. Implementación de Funcionalidad y Dinamismo

La implementación de la aplicación se ha dividido en tres capas: integración, presentación y negocio.

5.5.1. Carga Dinámica

La carga de los datos la hacemos a partir de un archivo de formato ".csv", con todas las columnas en una misma fila separadas por comas. Para facilitar la lectura de los datos transformamos este archivo en un archivo ".xlsx", con los datos que nos interesan y separados por columnas. Utilizando este archivo leemos los datos y los insertamos en las tablas correspondientes. También permite añadir nuevos campos. Los campos que nos interesan están almacenados en una tabla de la base de datos. La carga dinámica utiliza las siguientes clases.

- **CsvToExcel**: esta clase es la encargada de crear el nuevo archivo ".xlsx". Esta clase recibe un array de objetos que contienen la información de las columnas que se desean insertar. La clase se encarga de recorrer las cabeceras del archivo y comprobar que estos nombres de columnas están en el array de importación. En caso de no encontrar la columna, borra esa columna de su tabla correspondiente, de la tabla de importación, de las consultas guardadas y del array de importación. En caso de encontrarla guarda la posición en la que se encuentra. Después de esto, recorre todas las filas del archivo ".csv", separándolas por comas y añadiendo cada dato al nuevo archivo ".xlsx".
- **ExcelReading**: esta clase crea las consultas INSERT para ejecutar en BD y añadir los datos a las tablas. Recorre el array de importación, y para cada objeto importación comprueba a que tabla pertenece y que posición del fichero Excel ocupa. Para cada tabla crea sentencias INSERT concatenando los datos a insertar en la tabla. Cuando el siguiente objeto del array de importación no pertenece a la misma tabla, antes de leerlo hace una llamada al DAO para ejecutar la sentencia INSERT.
- **ExcelReading2**: esta clase se utiliza cuando queremos insertar campos que no están en el array importación, es decir, columnas nuevas que aparecen en el fichero Excel pero que no se han insertado previamente. Esta clase recibe como parámetros información relativa a la columna nueva y se encarga de buscar el nombre de la columna entre las cabeceras. En caso de encontrarlo, hace un INSERT en la tabla importación con la información relativa a esa columna y modifica la tabla a la que corresponde esa columna añadiendo el nuevo campo. Después de esto, crea una sentencia UPDATE para añadir los datos en la tabla correspondiente. Si el nombre no está repetido y el tipo de datos es correcto, hará una llamada al DAO para ejecutar esta sentencia. En caso de algún error o de que el nombre esté repetido, borrará la columna de la tabla correspondiente y de la tabla importación.

5.5.2. Integración

Es la capa encargada de acceder a los datos.

- **DAOFactory**: clase encargada de crear los objetos DAO.
- **DAO**: clase que contiene métodos insert y read para manipular datos de las tablas de la base de datos. Tenemos un DAO para cada tabla de la base de datos. Tablas para la gestión de los datos médicos, la gestión de los usuarios y la gestión de las consultas almacenadas.
- **TransactionFactory**: clase encargada de crear los objetos Transaction.
- **Transaction**: clase que contiene el conjunto de operaciones que se ejecutan sobre la base de datos.
- **Transaction Manager**: clase que permite la concurrencia en nuestra aplicación, usa una transacción cada vez que se vaya a llevar a cabo una operación con la base de datos.

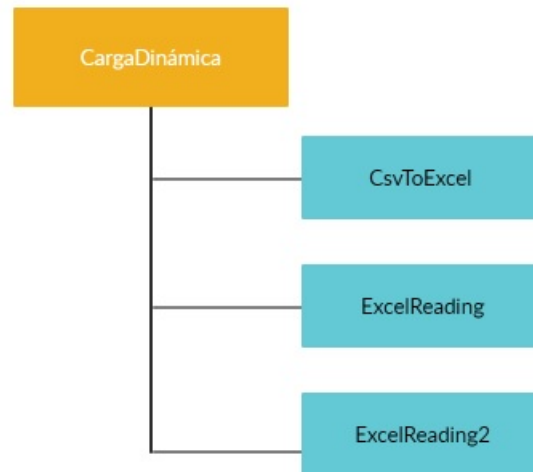


Figura 5.21: Esquema Carga

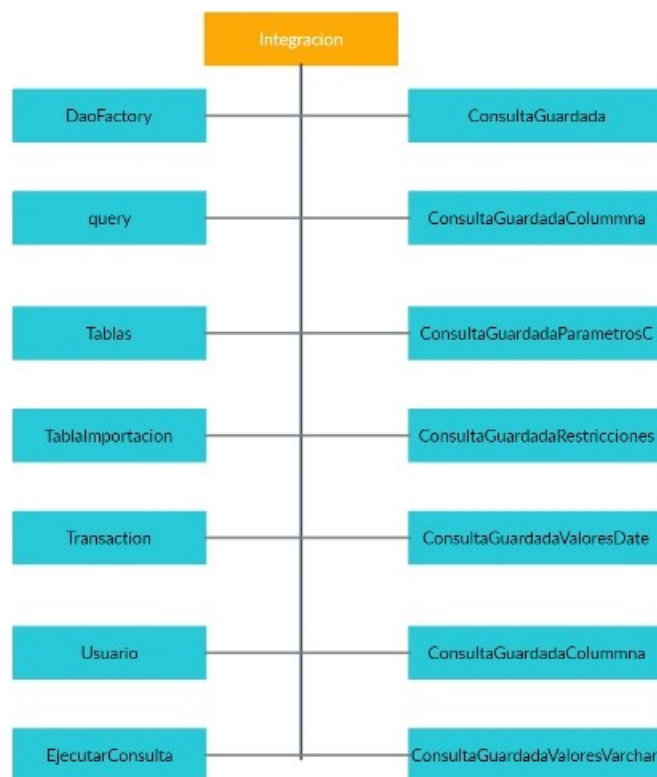


Figura 5.22: Esquema integracion

5.5.3. Negocio

En esta capa se centraliza la lógica de nuestra aplicación. En ella se llevan a cabo las operaciones de negocio.

- Transfer: clase que guarda todos los datos de un objeto para que puedan ser utilizados y transmitidos por otras clases. Una clase transfer para los usuarios, que contiene su nombre y contraseña y una clase transfer para las consultas guardadas.
 - TConsultaGuardada: es el objeto transfer mas complejo de la aplicación. Es el objeto que se utiliza durante una sesión, donde se almacenan todos los filtros y restricciones que el usuario desea aplicar. Sobre este objeto se cargan los datos relativos a una consulta guardada.
 - TImportacion: es el objeto que contiene la información sobre las columnas a importar. Contiene el nombre de la tabla a la que pertenece la columna, el nombre de la columna, el tipo de dato, su posición en la tabla y su nombre y su posición en el archivo Excel.

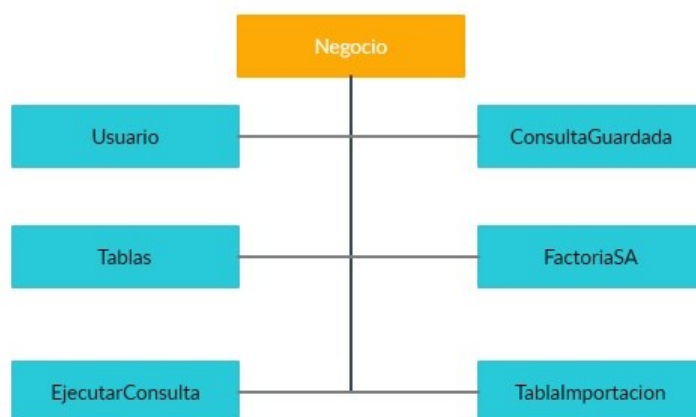


Figura 5.23: Esquema negocio

- Servicio de aplicación: encapsula todas las reglas de negocio en una clase. Es el encargado de crear las transacciones y de utilizar los DAOs para obtener los datos con los que hacer las operaciones de nuestra aplicación. Tenemos servicios de aplicación para los diferentes transfers de datos de consultas y de usuario. Entre los diferentes servicios de aplicación cabe destacar la funcionalidad de estos:
 - ASEjecutarConsulta: es el encargado de generar la consulta SQL a través del método prepararConsultas. Este método parte de un objeto TConsultaGuardada, recorre sus restricciones y parámetros, desglosándolos y concatenándolos en varios Strings. Un String “select” para los filtros, un String “from” para las tablas y un String “where” para las restricciones. Devuelve un String con la query SQL y el método ejecutarConsulta se encargará de crear la transacción y llamar al DAO.

- SAConsultaGuardada: El metodo insertarConsultaGuardada se encarga de guardar un objeto TConsultaGuardada, llamando a cada DAO para guardar la información relativa a columnas, restricciones, parámetros y valores. El método mostrarConsultaGuardada se encarga de generar un objeto TConsultaGuardada partiendo de los datos guardados de una consulta.

5.5.4. Presentación

En esta capa centralizamos las interfaces de usuario, el controlador y sus comandos y el dispatcher. Esta capa se compone de:

- Controlador: es la clase intermedia entre la vista y el modelo. Genera un comando dado el evento, ejecuta el comando, obtiene la instancia del despachador y el dispatcher ejecuta la información devuelta por el comando.

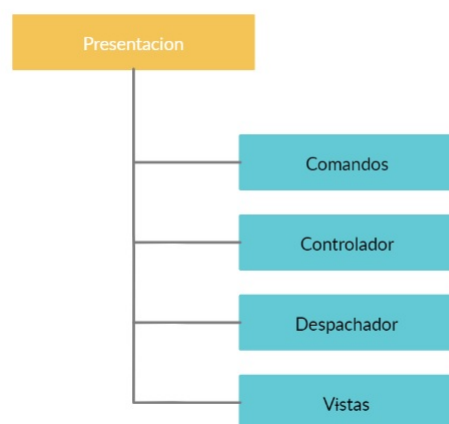


Figura 5.24: Esquema presentación

- Dispatcher: el modelo se comunica con la vista a través del controlador, delegando en el dispatcher. Según el contexto recibido, con una variable entera que representa un comando y los datos, el dispatcher se encarga de llamar a los métodos de las diferentes vistas.
- Comandos: en nuestra aplicación tenemos un comando para cada evento. Un comando para cada operación y comandos encargados de abrir y cerrar las vistas. Cabe destacar la funcionalidad de los siguientes comandos:
 - ComandoMenu: esta clase se encarga de llamar a la clase encargada de leer el archivo Excel seleccionado y de devolver un contexto con una variable entera MENU, y con los datos de las tablas cargada, para que el dispatcher genere la instancia de la vista principal. En caso de ser un archivo incorrecto devuelve un contexto con una variable entera VISTACARGADATOS, para que el dispatcher cargue de nuevo la vista de carga de datos.

- ComandoEjecutarConsulta: en esta clase con los datos recibidos rellena un objeto TConsultaGuardada y hace una llamada al ASEjecutarConsulta pasando como parámetro el objeto. El servicio de aplicación devolverá el resultado y el comando devolverá un contexto con una variable entera EJECUTARCONSULTA y los datos del resultado obtenido.
- Vista: esta clase contiene un atributo privado con la interfaz y los diferentes métodos abrir y update para cada interfaz.
- GUI: existe una por cada vista e implementa la interfaz de usuario utilizando Java Swing.

El código de la aplicación descrita y una guía para hacerla funcionar, se puede descargar desde el siguiente enlace:

<https://drive.google.com/drive/folders/1V7hdiWtomaE2Zmw2InK5SzgkhJHR1CJE?usp=sharing>

Capítulo 6

Contribuciones al Proyecto

En esta sección se exponen las contribuciones cada integrante del equipo ha aportado al proyecto. Este proyecto ha sido un trabajo en equipo por lo que todo el desarrollo ha sido realizado por ambos integrantes, sin embargo aquí se especifica a que tarea ha dedicado mas tiempo cada uno.

6.1. José Antonio Garrido Sualdea

Participación en la elaboración de la memoria.

Investigación de la herramienta Netbeans para el desarrollo de las vistas de la aplicación. Es una plataforma software muy potente ya que aparte de permitir desarrollar código en Java, también posee una herramienta para realizar el diseño gráfico aplicando la técnica “drag and drop”. Para ello tuve que ver bastantes vídeo tutoriales a parte de leer diferentes páginas web.

Desarrollo de la lógica de presentación con la investigación de diversos tipos de patrones. En concreto se busca que la vista maneje una petición y genere una respuesta. Por otra parte también buscamos el desacoplamiento dentro de la aplicación a la hora de encapsular la información de entrada independientemente del protocolo para mejorar la reusabilidad y mantenimiento.

Ayuda al desarrollo de la lógica de negocio, ya que al ser la capa que esta en contacto con la de presentación, tuve que documentarme para poder ejercer una buena comunicación entre capas. A esto refiere el tratamiento de enviar y recoger datos para la vista.

Investigación de la librería Apache Poi para carga del fichero. Este proceso conlleva la transformación de un fichero base en formato csv en formato .xlsx para la clasificación de la información en columnas y con ello poder procesar los datos.

Investigación de la transformación de ficheros Excel para el tratamiento de los datos de

Herramienta de apoyo al análisis de datos para diagnóstico de enfermedad celiaca

A tool for supporting data analysis related to the diagnosis of the celiac disease UCM

forma universal, siendo cualesquiera que sean las columnas. Aquí interviene la parte de carga dinámica.

Depuración y corrección de errores en el código.

6.2. Álvaro Noriega Moreno

Participación en la elaboración de la memoria.

Investigación del uso de la librería Apache Poi para la creación de ficheros, así como las clases a utilizar para crear un fichero .xlsx y sus diferentes columnas para exportar los datos. Durante la investigación se realizaron pruebas de como usar esta librería para mas tarde incluirlas en el proyecto. Participación en el desarrollo de la carga dinámica gracias a los conocimientos adquiridos durante la investigación para transformar el fichero original (csv) en un archivo .xlsx y así poder obtener los datos de los ficheros.

Desarrollo de la lógica relativa a la creación de las consultas a partir de las restricciones y filtros de la vista, para la extracción de los datos resultantes de la base de datos.

Desarrollo de la lógica de negocio relativa a las consultas guardadas y a la obtención de datos, encargada de rellenar los objetos necesarios a partir de los datos obtenidos de la base de datos.

Investigación y configuración de las diferentes librerías y clases de Java para la conexión y uso de Mysql en el proyecto. Para la investigación se realizaron diferentes pruebas y se utilizaron proyectos de ejemplo vistos en la universidad

Creación y configuración de las tablas necesarias de la base de datos, así como el estudio de las diferentes columnas necesarias para cada tabla y de las relaciones entre las diferentes tablas.

Depuración y corrección de errores en el código.

Capítulo 7

Conclusiones

En este capítulo dejamos reflejado el esfuerzo empleado en la investigación y desarrollo de la aplicación, las impresiones que nos ha causado, el desarrollo personal que hemos conseguido a nivel funcional e intelectual, la adquisición de experiencia referida a saber comunicarse y organizarse con los compañeros de equipo.

Las horas empleadas en investigación sobre la enfermedad celiaca para poder sacarle partido a la aplicación nos sirvieron para aprender bastante acerca de esta enfermedad. Muchas personas pueden alcanzar cierto grado de conocimiento sobre este tema, pero la verdad es que una vez que nos embarcamos en búsqueda de información, nos dimos cuenta de que es más profundo de lo que pensábamos. Ahora somos más conscientes de lo que realmente es esta enfermedad.

Además de lo que aprendimos al realizar la investigación sobre la enfermedad celiaca, el desarrollo de esta aplicación nos ha ayudado a adquirir cierta experiencia en el ámbito del desarrollo software, en la gestión de herramientas, en la mejora de la planificación, etcétera. Si hubiera algo menos que destacar diríamos que se pecó de infravalorar la cantidad de trabajo que conllevaba, ya que aunque ya hemos realizado más proyectos durante la carrera, es verdad que quizás no hemos trabajado con tantos módulos o tanto volumen de datos. Pero al final gracias a la constancia y a las técnicas aprendidas en la carrera hemos sido capaces de reconducirlo llegando a buen puerto. Así que estos nos ha ayudado positivamente como superación personal y profesional a parte de ayudarnos a mejorar a trabajar en equipo.

Respecto a la primera impresión que nos causó el proyecto con el resultado final, la verdad es que nos ha sorprendido bastante. No eramos conscientes de la gran utilidad que tiene esta herramienta de apoyo en la medicina. Gracias a esta aplicación, el gran número de datos que estaban almacenados en grandes torres de ordenador cogiendo polvo, hemos conseguido darles una segunda vida. Hemos conseguido que esos valores sirvan para poder contrastar diferentes características entre pacientes con el único objetivo de realizar investigaciones para tratar de hallar patrones o similitudes entre ellos y para poder predecir si unas personas u otras pueden ser más propensas a padecer dicha enfermedad.

Estamos también orgullosos de haber realizado un código bien estructurado para posibles versiones nuevas de otros compañeros que quieran seguir dando utilidad o hacer seguir creciendo esta aplicación añadiendo distintas funcionalidades.

Capítulo 8

Conclusions

Thanks to all the time spent investigating about the celiac disease, we learned a lot of different things about it. Many people can reach a certain grade of knowledge about this topic, but the truth is that is deeper than what we thought. Now we are more conscious about what this disease really is.

Apart from the information we learned while conducting research on celiac disease, the development of this application has helped us to gain some experience in the field of software development, tool management, improving planning, etc. If there was something wrong to highlight, we would say that the amount of work involved was underestimated, because although we have already carried out more projects during the degree, it is true that perhaps we have not worked with so many modules or so much volume of data. But in the end, thanks to the perseverance and the techniques learned in the degree we have been able to redirect it to a successful conclusion. So these have helped us positively as a personal and professional improvement in addition to helping us improve teamwork.

Regarding to the first impression caused by the final result of the project, the truth is that we were quite surprised about it. We were not aware of the great utility of this support tool in medicine. Thanks to this application, the large number of data that was stored in the computer towers gathering dust, we were able to manage it and use it for different tasks. We've made that these values can be used to contrast different characteristics between patients with the purpose of performing research and try to find patterns or similarities between them in order to be able to predict whether some people or others may be more prone to suffering from this disease.

We are also proud of have done a well-structured code which will be useful to develop future versions of it by our colleagues next year., who will be able to add different functionalities to it.

Capítulo 9

Trabajo Futuro

En este capítulo proponemos diferentes ampliaciones que a nos han parecido interesantes destacar para una posible ampliación y mejora de la aplicación. Entre las ampliaciones encontramos algunas mas fáciles de desarrollar y algunas que requieren de mayor dedicación.

Para el futuro hemos planteado una serie de módulos que supondrían una mejora y ampliación de las funcionalidades de nuestra aplicación.

Una posible ampliación de nuestra aplicación sería la creación de un modulo que permitiese al usuario insertar datos médicos de manera manual. Proporcionaríamos al usuario registrado una nueva interfaz en la que pueda ir seleccionando bloques y rellenando sus respectivos parámetros. Esta funcionalidad permitiría añadir pacientes adicionales además de los cargados desde el fichero de datos.

Otra extensión consistiría en un modulo en el que el usuario introdujese una dirección de correo electrónico y la aplicación adjuntaría a un email un archivo Excel con los resultados y los datos seleccionados por el usuario que sería enviado a la dirección introducida. Esta funcionalidad agilizaría el proceso para compartir datos médicos y resultados de consultas entre diferentes médicos de manera sencilla y rápida.

Por último se podría añadir la posibilidad de cargar un fichero Excel desde un repositorio externo. En la interfaz de carga de datos el usuario podría introducir la url de la ubicación de un archivo Excel localizado en un servicio de alojamiento de archivos online, como por ejemplo Google Drive. Esta característica eliminaría la restricción de que el archivo a cargar se encuentre alojado en el equipo desde el que se está ejecutando la aplicación.

Índice de figuras

4.1. Vista gráfica del esquema de la BBDD parte 1	15
4.2. Vista gráfica del esquema de la BBDD parte 2	15
5.1. Vista gráfica de la ejecución de Xampp	20
5.2. Vista gráfica phpMyAdmin	20
5.3. Vista gráfica Módulo Inicio Sesión	22
5.4. Vista gráfica Módulo Registro	22
5.5. Vista gráfica Seleccionar Archivo	23
5.6. Vista gráfica Configuración del fichero	24
5.7. Vista gráfica Pantalla Principal	24
5.8. Vista gráfica Seleccionar Tablas	25
5.9. Vista gráfica Filtrar Parámetros	25
5.10. Vista gráfica restricciones numéricas	26
5.11. Vista gráfica restricciones alfabéticas	26
5.12. Vista gráfica restricciones Fecha	27
5.13. Vista gráfica bloques de parámetros a mostrar	27
5.14. Vista gráfica parámetros a mostrar	28
5.15. Vista gráfica Filtros Aplicados	29
5.16. Vista gráfica Restricciones Aplicadas	29

5.17. Vista gráfica Restricciones Aplicadas	30
5.18. Vista gráfica Abrir Consulta Guardada	30
5.19. Vista gráfica Guardar Consulta	31
5.20. Vista gráfica Consulta y Exportar Datos a Fichero Excel	32
5.21. Esquema Carga	34
5.22. Esquema integracion	34
5.23. Esquema negocio	35
5.24. Esquema presentación	36

Bibliografía

- [1] Definición de xml, 2013. <https://definicion.de/xml/>.
- [2] Introducción a java, 2013. <http://codigoprogramacion.com/cursos/java/47-introjjava.html#.XrlTYmgzZPZ>.
- [3] Por qué elegir el gestor de base de datos mysql, 2018. <https://fp.uoc.fje.edu/blog/por-que-elegir-el-gestor-de-base-de-datos-mysql/>.
- [4] Descarga gratuita de java, 2020. <https://www.java.com/es/download/>.
- [5] Factoría abstracta, 2020. <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/191>.
- [6] Google drive, 2020. https://es.wikipedia.org/wiki/Google_Drive.
- [7] Google drive, 2020. https://www.google.com/intl/es-419_ALL/drive/.
- [8] Mediplus. enfermedad celiaca, 2020. <https://medlineplus.gov/spanish/celiacdisease.html>.
- [9] Netbeans, 2020. <https://es.wikipedia.org/wiki/NetBeans>.
- [10] Que es xampp y que necesita para ser instalado, 2020. <https://sites.google.com/site/portafoliovicenciosr/poll>.
- [11] Singleton, 2020. <https://es.wikipedia.org/wiki/Singleton>.
- [12] Sql, 2020. <https://es.wikipedia.org/wiki/SQL>.
- [13] Xampp: instalación y primeros pasos, 2020. <https://www.ionos.es/digitalguide/servidores/herramientas/instala-tu-servidor-local-xampp-en-unos-pocos-pasos/>.