Title: Administrative system of crimes in Jalisco
SW Architecture

History				
Issue status (Index)	**Maturity/Date** (draft/invalid/valid) (dd-mmm-yyyy)	**Author** Department	**Check/Release** Department	**Description**
1.0	Draft 13-Abr-24	N/A	N/A	Architecture of the Administrative system of crimes in Jalisco.

Table of Contents

1 Purpose

The purpose of this document is to establish traceability and maintain a record of the architecture of our software project "Administrative System Of Crimes In Jalisco." These points will be based on previously generated documents that contain the system requirements.

2 Definitions and abbreviations

Definitions

Software: Set of programs and routines that allow the computer to perform certain tasks.

Interface: In computing, an interface is used to refer to the functional connection between two systems, programs, devices, or components of any kind, which provides communication at different levels, allowing the exchange of information.

Victim: A physical person who suffers harm caused by a subject.

Perpetrator: It refers to a murderer, a rapist, or an assailant, as they all generate victims (the individuals who are murdered, raped, or assaulted).

Detective: A police officer or private investigator whose role is to gather information as evidence to solve a crime.

Witness: A person who has information about a crime or event and provides testimony or evidence regarding what they have observed or experienced.

Connection with a database: a connection with a database enables a software application to interact with and manipulate data stored in the database.

Function: a function is a block of organized, reusable code that performs a specific task or set of tasks within a software program.

Query: a query is a request for specific information from a database, typically expressed in a query language such as SQL.

Abbreviations

SW – Software.

HW – Hardware.

IDE: An integrated development environment (IDE) is a software application that helps programmers develop software code efficiently.

GUI - Graphical User Interface.

DBMS - Database Management System.

UI - User Interface.

OOP - Object-Oriented Programming.

References

N°	Document name	Reference
1	Timing plan.	

3 Realization constraints and targets

- That the software adheres strictly to the requirements, nothing beyond what is written.
- It must allow the user to access information at any time.
- It must be intuitive.
- Storing essential information for each case.
- The software does not generate detailed case reports.
- It will not save the sentence given to the perpetrator.
- Delivery during the code development stage, as it might happen that our tests indicate our software is not functioning as it should, which could delay us in terms of deliverables.
- There's a risk that we may have to correct some aspect of our project overall based on our supervisor's recommendation, which would also delay us.

4 SW Functional Architecture

Below are the functions that will allow us to execute our software; however, as a quick way to describe the behavior of our software, it should store relevant information about a criminal case, allowing us to view, modify, and delete this information.

4.1 Table of functions

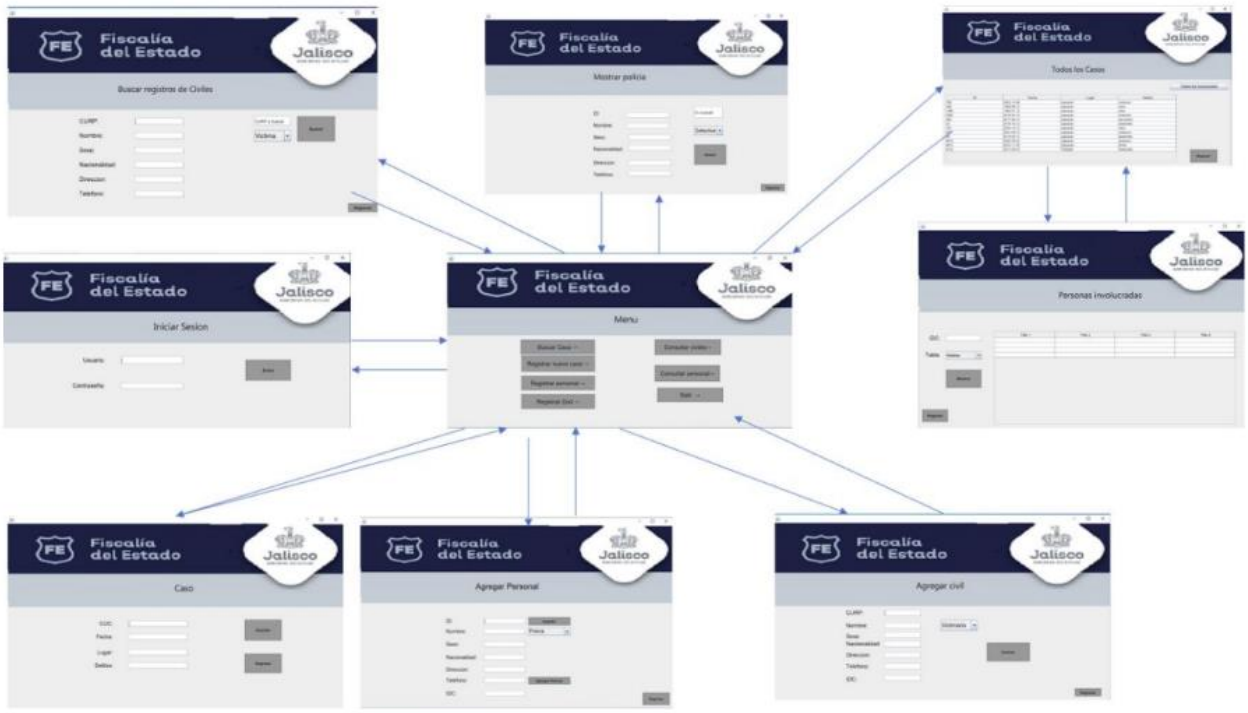
Function Name	Description
conectar	contain all the procedures for connecting to the database return an exception if it cannot establish a connection.
insertarCaso	Insert the case into the database.
insertarVictima	Insert the victim into the database.
validarUsuario	Check if the user exists in the database.

mtrCaso	Allow displaying the case selected by the user.
mtrInvolucrado	Display the individuals involved in the case selected by the user.
uptIdc	Update the information of a case.
uptCualquiera	Update information of any involved individual in the case.
mtrC	Generate an output that displays the case data.
insertarVictimario	Insert the perpetrator into the database.
insertPolicia	Insert the police into the database.
btnInsertActionPerformed	This is an event that allows us to create a button for the user.
buscarCasoActionPerformed	Search for the case.
searchCivilActionPerformed	Search for a civilian.
newCasoActionPerformed	Allow creating a new case.
getBackActionPerformed	Allow returning to the previous interface.
newCivilActionPerformed	Allow adding a civilian to the database.

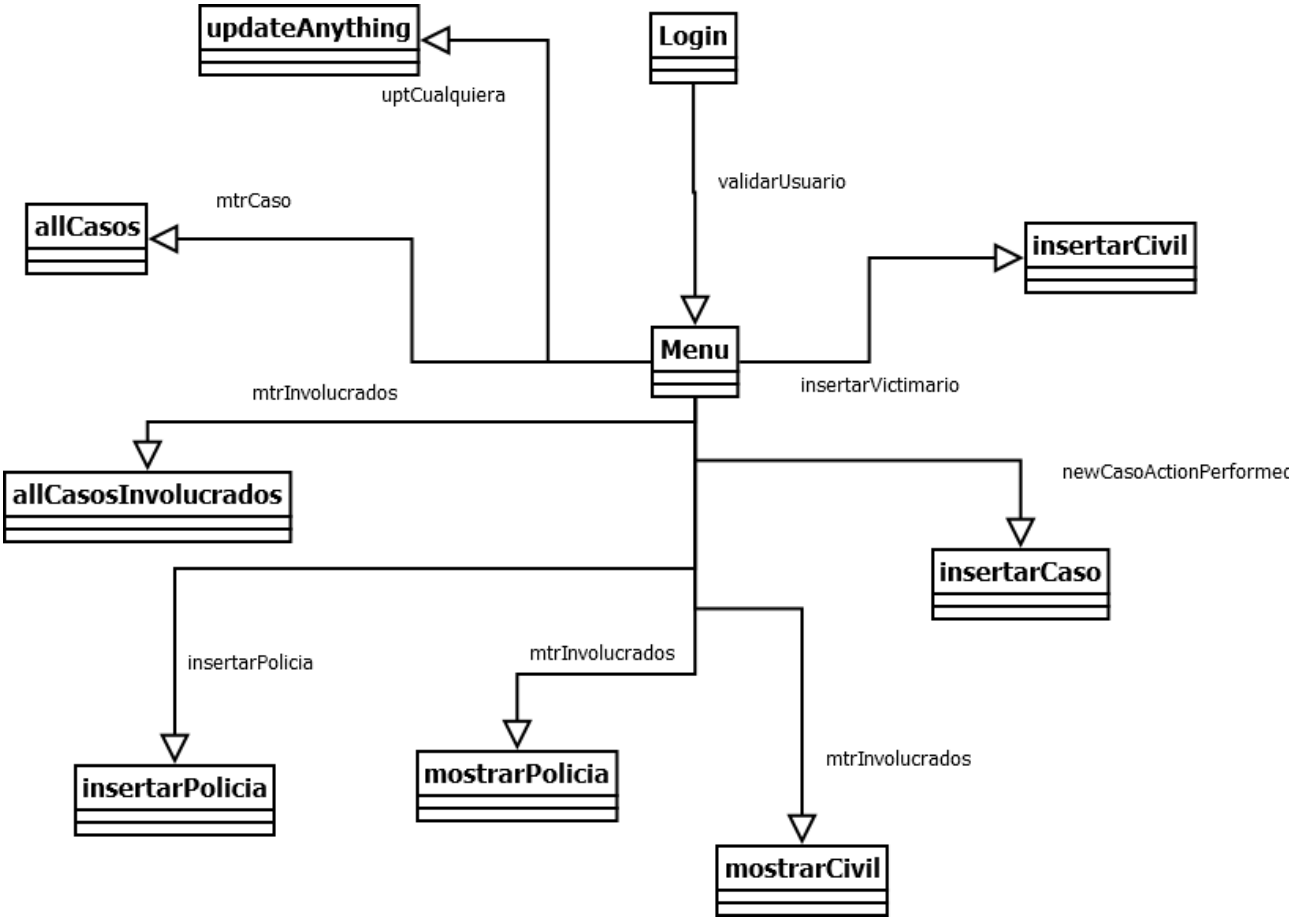
4.2 Table of functional Interfaces

Signal/Event	Description
Login	It's the home screen where credentials are entered.
Menu	It's the interface through which the user will control the entire system.
allCasos	Display all cases registered in the database.
allCasosInvolucrados	Display the individuals involved in the specified case.
updateAnything	Update information of a record in any table except for the case table.
insertarPolicia	Insert the police officer into the database.
insertarCivil	Insert the civilian into the database.
insertarCaso	Insert the case into the database.
insertarDetective	Insert the detective into the database.
mostrarPolicia	Display the data of all police officers related to a case.
mostrarCivil	Display the information of a civilian related to a case.

4.3 Functional Interface


Fiscalía del Estado
Jalisco
Buscar registros de Civiles
Mostrar policía
Todos los Casos
Iniciar Sesion
Menu
Personas involucradas
Caso
Agregar Personal
Agregar civil

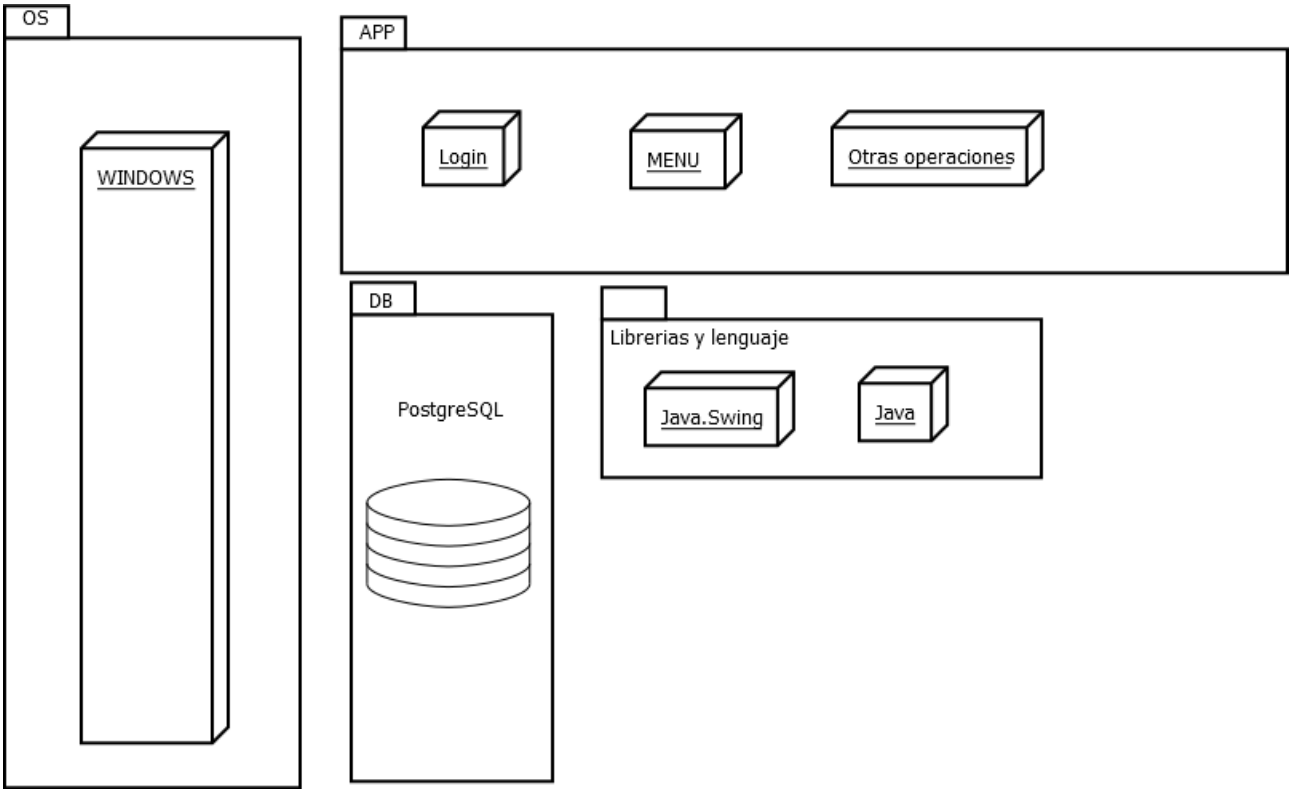
4.4 Functional Interaction



5 SW Physical Architecture

Our project has 4 physical layers if we want to view it in this way. In the first layer, we have the operating system where our program runs; this operating system will only be Windows. Then, we will have our application where all interactions with the user will take place. Next, we have the database where we will store all our data with the help of a database management system, which in our case is PostgreSQL. Finally, we have the libraries and the language we use, such as Java language and the Swing library.

5.1 Physical Decomposition



5.2 Table of SW components

SWC Name	Description	Estimated Size (KB)
conexionSQL	responsible for containing the necessary operations to establish a connection with our PostgreSQL database.	963
crudSQL	It is responsible for containing all the functions that will send or receive information. In other words, these functions perform the necessary queries between our system and the database, and it also has functions that display the data in our program.	12996
interface	It contains all the functions necessary for our end user to interact with our program and is responsible for the visual aspect of the program.	50000
variables	This is a class that is responsible for creating a template for all objects, whether they are cases, detectives, victims, witnesses, etc.	2210
Database	It is responsible for saving all the necessary data of our system.	3500

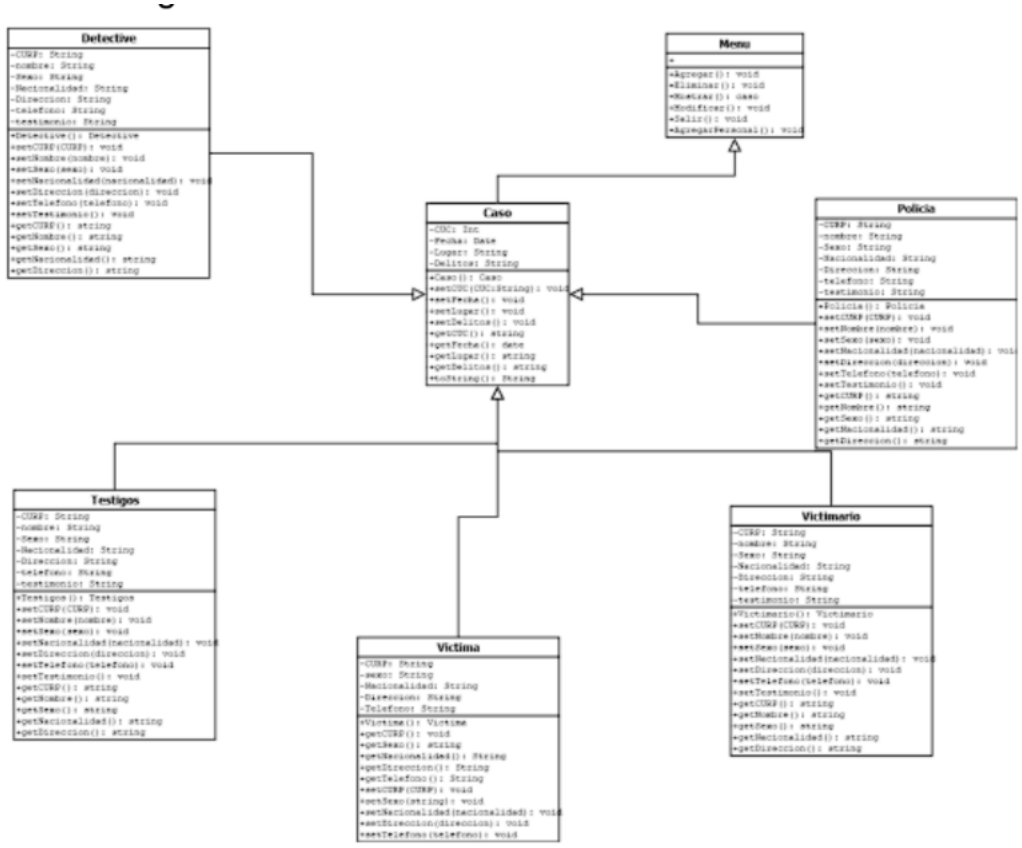
6 SW Requirements Allocation

We use a GitHub repository to store our documents and our project in general, the document that we use as a basis for this is the “arquitectura de software” document that is within this repository: https://github.com/VicHGC/Ingenieria-de-software/

7 SW Integration Plan

Integration Step	Description	SW components	Comments
1.- Create the database.	Integrate all the necessary tables for our system as specified in the data design.	Database	
2.- We create a connection and validate that it is correct.	We will establish our connection with the database and provide validation for any response it returns.	Database conexionSQL	
3.- Creation of the data class.	In this stage, we will create the class that will receive all the data from the database.	Variables	
4.- Integration of query functions.	In this stage, we will create all the functions that send or receive information from the database.	crudSQL	
5- Creation of the user interface.	All previously created functions will be integrated and connected to their respective graphical interfaces.	interface	

8 Class Diagram



9 State machine diagram.