

*Axel Iván Alba Muñoz*

*Elmer Fabian Gallegos Solís*

*Carlos Ángel Muñoz Castañeda*

*Christian Joan Rodríguez Dimitri*

PRoyecto final - entrega

Sistema ALOIS

15 de abril de 2019

INGENIERÍA EN SISTEMAS COMPUTACIONALES – 8°A

**Requerimientos**

Se requiere una base de datos distribuida para poder administrar distintos asilos, enfocándose en aquellos que son públicos.

Debemos de llevar un control de cada paciente, por lo tanto, hay que conocer sus datos personales, como nombre, dirección, y diagnóstico, además es necesario saber en qué habitación se encuentran y las fechas de ingreso y egreso de estas habitaciones, además hay que saber quién es el responsable del paciente, ya que a él se le hacen los cargos y en general hay que contactarlo para ciertas situaciones especiales.

Para los cargos a los encargados de los pacientes, es necesario saber datos personales como su nombre y RFC, adicionalmente hay que registrar datos de contacto como dirección, correo electrónico y teléfono, por otra parte, los cargos se hacen de acuerdo con la habitación en la que están hospedados los pacientes.

Por otro lado, requerimos llevar un registro de medicamentos de cada paciente, así como que médico fue el que lo recetó, sobre el personal es necesario conocer que enfermera atiende a cada paciente y poder consultar los datos del personal administrativo.

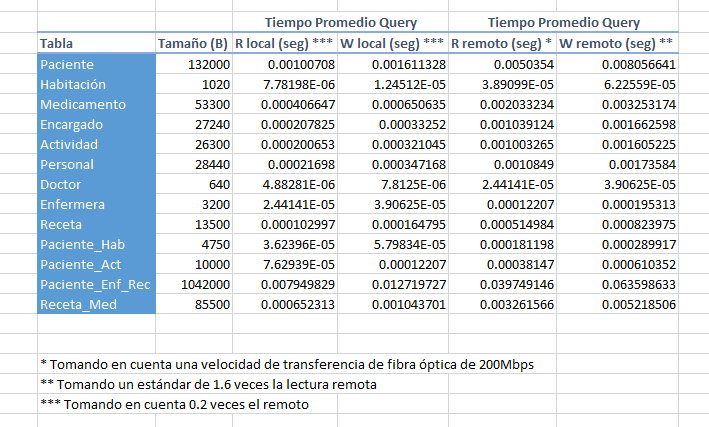
Para los médicos es necesario saber sus datos personales, especialidad y contacto tales como teléfono, celular, por otro lado, es vital conocer de cada medicamento su existencia y precio unitario además lógicamente de su nombre, descripción y conocer a qué hora y con qué cantidad deben los pacientes de tomar sus medicamentos.

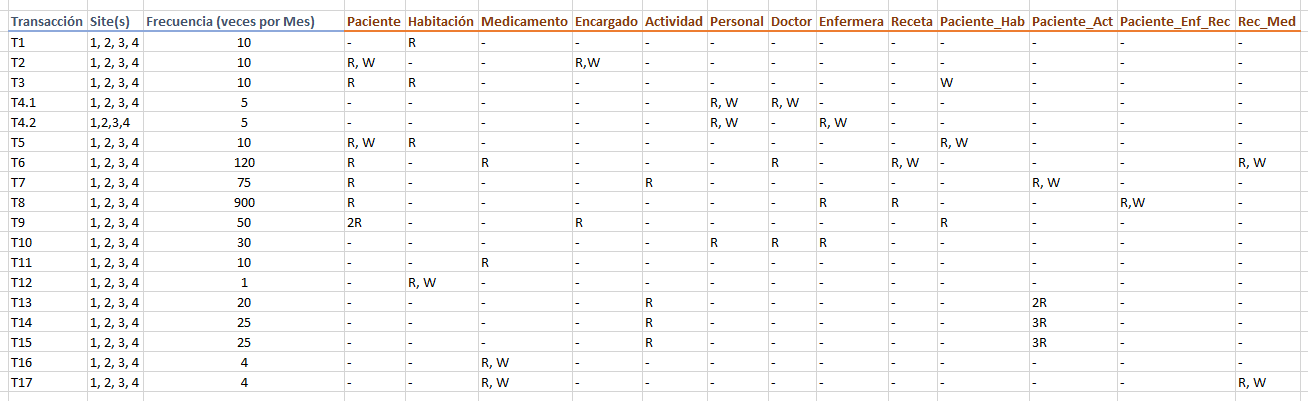
Finalmente, la mayoría de los pacientes tienen al menos algún deterioro mental, en este caso para enfocarnos en el Alzheimer se desarrollarán juegos que ayuden a disminuir el impacto que esta enfermedad tiene sobre el paciente, pero, además con el fin de mejorar esta terapia es indispensable tener el registro de las partidas jugadas, con el fin de poder observar el desarrollo que tendrá el paciente.

**Listado de Transacciones**

1. Listar las habitaciones disponibles en determinado momento.
2. Registro de pacientes que se dan de alta en el asilo.
3. Asignación de una habitación para un paciente.
4. Registro de nuevo personal en el asilo.
   1. Registro de un nuevo Doctor en el asilo.
   2. Registro de un nuevo enfermero o enfermera en el asilo.
5. Dar de baja un paciente y liberar la habitación correspondiente.
6. Asignación de medicamentos a un paciente recetados por un doctor.
7. Asignación de actividades de la aplicación a los pacientes que van a realizarlas.
8. Registrar el servicio de un enfermero(a) a un paciente y si administró o no medicamentos.
9. Mostrar la información completa de un paciente determinado.
10. Mostrar la información del personal médico del asilo.
11. Listar los medicamentos que no estén en existencia en el inventario.
12. Actualizar los precios de las habitaciones.
13. Obtener el historial de actividades de un paciente, así como un promedio de evaluación.
14. Listar las actividades completadas por un paciente.
15. Listar las actividades pendientes de un paciente.
16. Agregar nuevos medicamentos al inventario de farmacia.
17. Eliminar los medicamentos que ya no se distribuyan en el asilo.

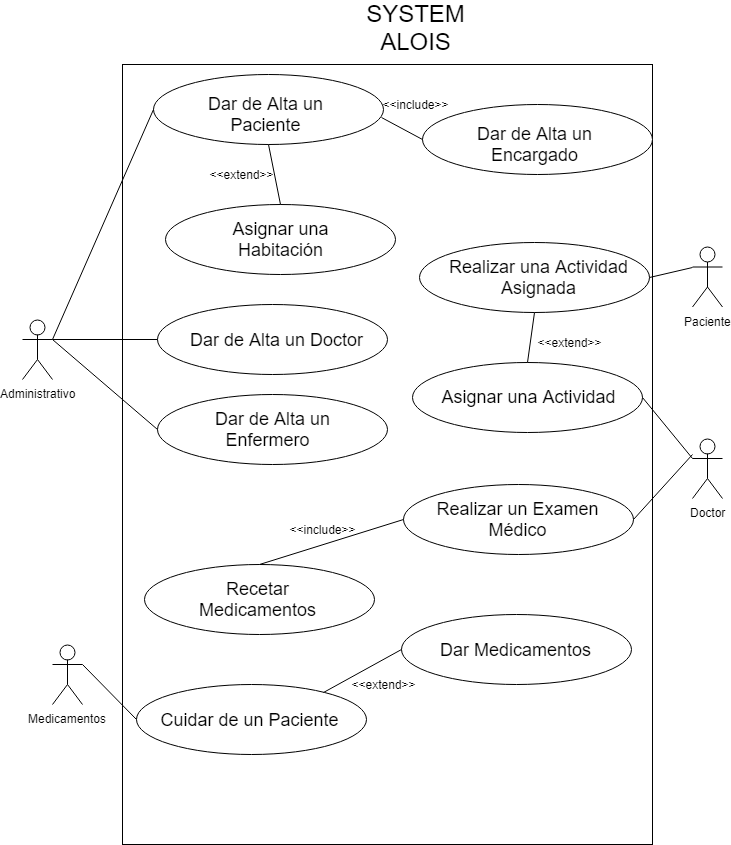
**Volumen de Datos**



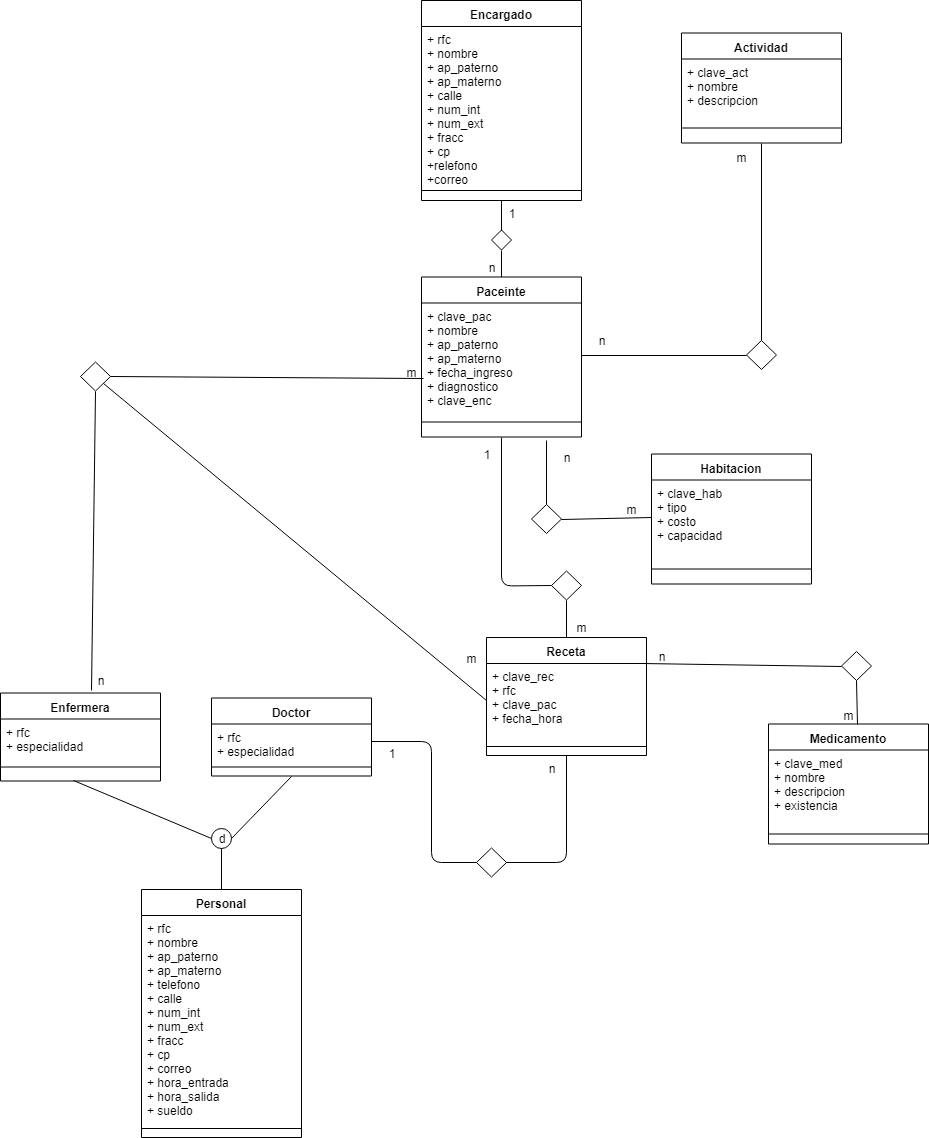


**Modelado**

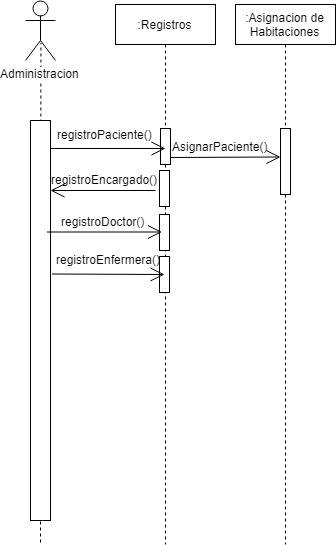
*Diagrama de Casos de Uso*

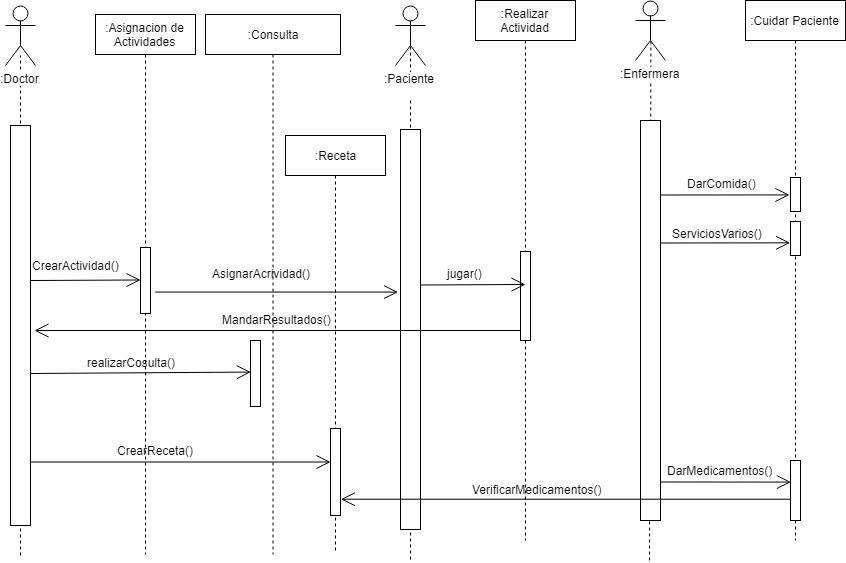


*Diagrama de Clases*

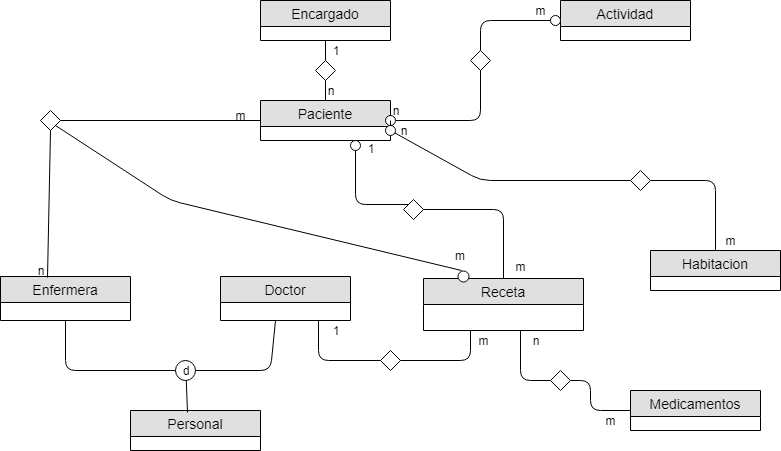


*Diagrama de Secuencias*





**Modelo Entidad-Relación**



**Transformación a Tablas**

Paciente

- clave\_pac : smallint (pk)

- nombre : varchar(25)

- ap\_paterno : varchar(25)

- ap\_materno : varchar(25)

- diagnostico : varchar(1000)

- clave\_enc : varchar(13) (fk)

Personal

- rfc : varchar(13) (pk)

- nombre : varchar(25)

- ap\_paterno : varchar(25)

- ap\_materno : varchar(25)

- telefono : tinyint

- calle : varchar(50)

- num\_int : tinyint

- num\_ext : tinyint

- fracc : varhcar(50)

- cp : smallint

- correo: varchar(20)

- hora\_entrada : time(3)

- hora\_salida : time(3)

- sueldo : smallint

Enfermera

- rfc : varchar(13) (fk)

- especialidad: varchar(15)

Doctor

- rfc : varchar(13) (fk)

- especialidad : varchar(25)

Medicamento

- clave\_med : smallint (pk)

- nombre : varchar(25)

- descripción : varchar(500)

- existencia : smallint

Encargado

- rfc : varchar(13) (pk)

- nombre : varchar(25)

- ap\_paterno : varchar(25)

- ap\_materno : varchar(25)

- calle : varchar(50)

- num\_int : tinyint

- num\_ext : tinyint

- fracc : varchar(50)

- cp : smallint

- teléfono : tinyint

- correo : varchar(20)

Actividad

- clave\_act : smallint (pk)

- nombre : varchar(20)

- descripcion : varchar(500)

Habitación

- clave\_hab : tinyint (pk)

- tipo : varchar(10)

- costo : smallint

- capacidad : smallint

Receta

- clave\_rec : smallint (pk)

- rfc : varchar(13) (fk)

- clave\_pac : smallint (fk)

- fecha\_hora : datetime

Paciente\_Hab

- clave\_pac : smallint (fk)

- clave\_hab : tinyint (fk)

- ingreso : datetime

- salida : datetime

Paciente\_Act

- clave\_pac : smallint (fk)

- clave\_act : smallint (fk)

- estado : varchar(10)

- tiempo : time(0)

- evaluacion : tinyint

Paciente\_Enf\_Rec

- clave\_pac : smallint (fk)

- rfc : varchar(13) (fk)

- servicio : varchar(500)

- clave\_rec : smallint (fk)

Receta\_Med

- clave\_rec : smallint (fk)

- clave\_med : smallint (fk)

- cantidad : tinyint

- dosis : varchar(50

**Normalización**

* **Paciente (clave\_pac, nombre, ap\_paterno, ap\_materno, diagnostico, clave\_enc)**

*Primera Forma Normal – 1NF*

Se dividió el nombre en sus componentes de interés.

No existen datos multivalor que pudieran dividirse en diferentes atributos.

*Segunda Forma Normal – 2NF*

clave\_pac → nombre, ap\_paterno, ap\_materno, diagnostico, clave\_enc

Todos los atributos son determinados por la llave.

No existe una mejor llave, de hecho, no existe ninguna otra llave candidata.

*Tercera Forma Normal – 3NF*

No existen dependencias transitivas, por lo que la llave determina directamente a los demás atributos.

* **Habitación (clave\_hab, tipo, costo, capacidad)**

*Primera Forma Normal – 1NF*

Ninguno de los atributos es multivaluado.

Todos los atributos son atómicos.

*Segunda Forma Normal – 2NF*

clave\_hab → tipo, costo, capacidad

Todos los atributos son determinados por la llave.

No existe una mejor llave, de hecho, no existe ninguna otra llave candidata.

*Tercera Forma Normal – 3NF*

No existen dependencias transitivas, por lo que la llave determina directamente a los demás atributos.

* **Medicamento (clave\_med, nombre, descripción, existencia)**

*Primera Forma Normal – 1NF*

Ninguno de los atributos es multivaluado.

Todos los atributos son atómicos.

*Segunda Forma Normal – 2NF*

clave\_med → nombre, descripción, existencia

Todos los atributos son determinados por la llave.

No existe una mejor llave, de hecho, no existe ninguna otra llave candidata.

*Tercera Forma Normal – 3NF*

No existen dependencias transitivas, por lo que la llave determina directamente a los demás atributos.

* **Encargado (rfc, nombre, ap\_paterno, ap\_materno, calle, num\_int, num\_ext, fracc, cp, telefono, correo)**

*Primera Forma Normal – 1NF*

Se dividió el nombre en sus componentes de interés.

Se dividió la dirección en los componentes necesarios para el sistema.

No existen datos multivalor que pudieran dividirse en diferentes atributos.

*Segunda Forma Normal – 2NF*

rfc → nombre, ap\_paterno, ap\_materno, calle, num\_int, num\_ext, fracc, cp, telefono, correo

Todos los atributos son determinados por la llave.

No existe una mejor llave.

*Tercera Forma Normal – 3NF*

No existen dependencias transitivas, por lo que la llave determina directamente a los demás atributos.

* **Actividad (clave\_act, nombre, descripcion)**

*Primera Forma Normal – 1NF*

Ninguno de los atributos es multivaluado.

Todos los atributos son atómicos.

*Segunda Forma Normal – 2NF*

clave\_act → nombre, descripción

Todos los atributos son determinados por la llave.

No existe una mejor llave, de hecho, no existe ninguna otra llave candidata.

*Tercera Forma Normal – 3NF*

No existen dependencias transitivas, por lo que la llave determina directamente a los demás atributos.

* **Personal (****rfc, nombre, ap\_paterno, ap\_materno, teléfono, calle, num\_int, num\_ext, fracc, cp, correo, hora\_entrada, hora\_salida, sueldo)**

*Primera Forma Normal – 1NF*

Se dividió el nombre en sus componentes de interés.

Se dividió la dirección en los componentes necesarios para el sistema.

No existen datos multivalor que pudieran dividirse en diferentes atributos.

*Segunda Forma Normal – 2NF*

rfc → nombre, ap\_paterno, ap\_materno, teléfono, calle, num\_int, num\_ext, fracc, cp, correo, hora\_entrada, hora\_salida, sueldo

Todos los atributos son determinados por la llave.

No existe una mejor llave.

*Tercera Forma Normal – 3NF*

No existen dependencias transitivas, por lo que la llave determina directamente a los demás atributos.

* **Enfermera (rfc, especialidad)**

*Primera Forma Normal – 1NF*

Ninguno de los atributos es multivaluado.

Todos los atributos son atómicos.

*Segunda Forma Normal – 2NF*

rfc → especialidad

Al ser una tabla que hereda los atributos de la tabla Personal, el rfc que toma como llave foránea es única entre los registros de esta tabla.

Todos los atributos son determinados por la llave.

*Tercera Forma Normal – 3NF*

No existen dependencias transitivas, por lo que la llave determina directamente a los demás atributos.

* **Doctor (rfc, especialidad)**

*Primera Forma Normal – 1NF*

Ninguno de los atributos es multivaluado.

Todos los atributos son atómicos.

*Segunda Forma Normal – 2NF*

rfc → especialidad

Al ser una tabla que hereda los atributos de la tabla Personal, el rfc que toma como llave foránea es única entre los registros de esta tabla.

Todos los atributos son determinados por la llave.

*Tercera Forma Normal – 3NF*

No existen dependencias transitivas, por lo que la llave determina directamente a los demás atributos.

* **Receta (clave\_rec, rfc, clave\_pac, fecha\_hora)**

*Primera Forma Normal – 1NF*

Ninguno de los atributos es multivaluado.

Todos los atributos son atómicos.

*Segunda Forma Normal – 2NF*

clave\_rec → rfc, clave\_pac, fecha\_hora

Todos los atributos son determinados por la llave.

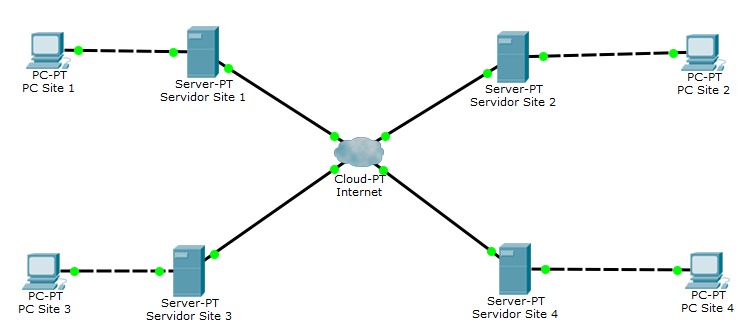
No existe una mejor llave.

*Tercera Forma Normal – 3NF*

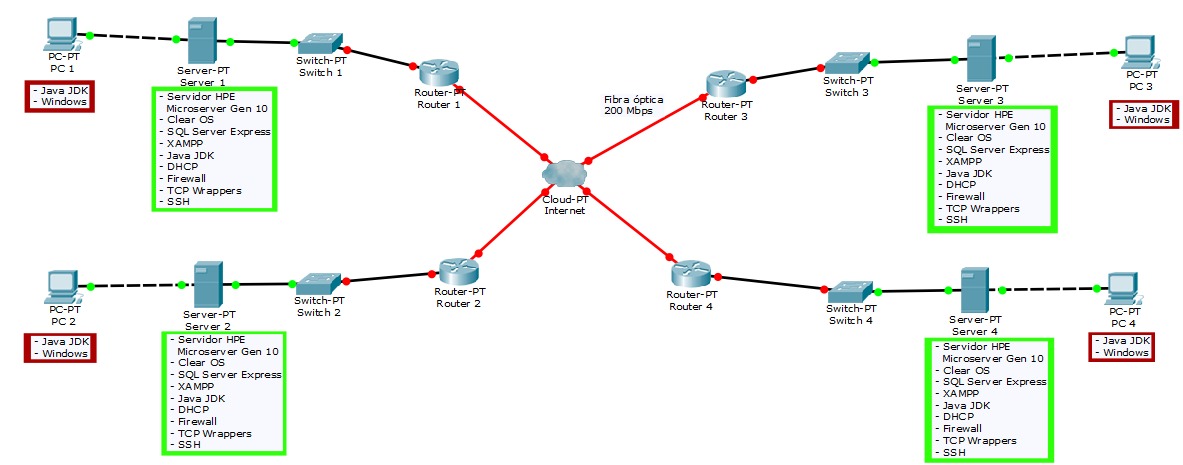
No existen dependencias transitivas, por lo que la llave determina directamente a los demás atributos.

**Diseño de Red**

*Diagrama de Red de Alto Nivel*

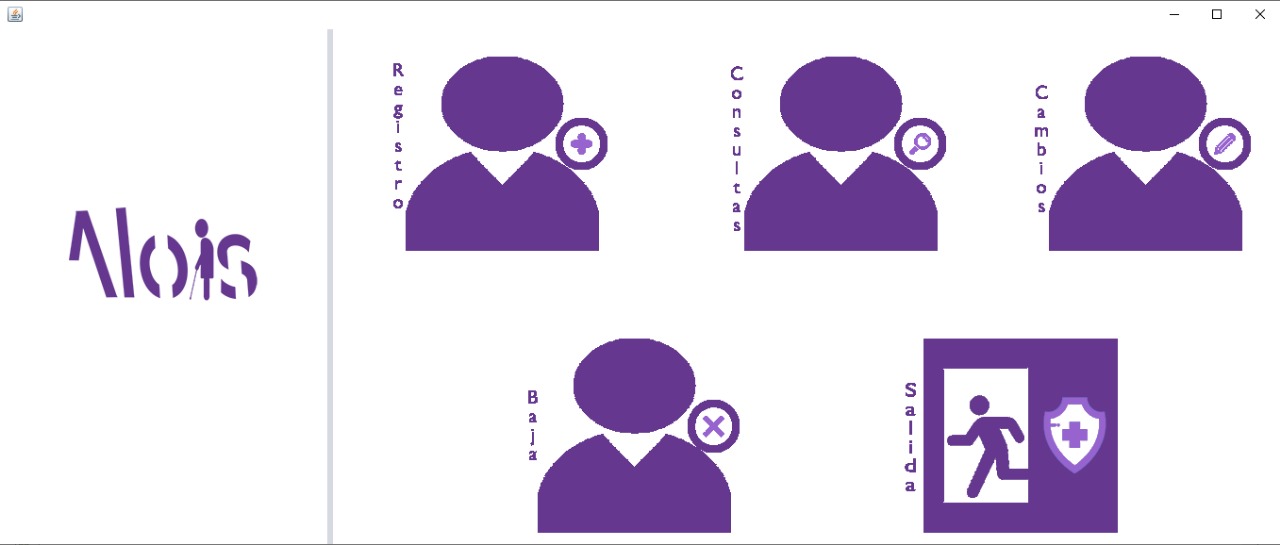


*Diagrama de Red de Bajo Nivel*

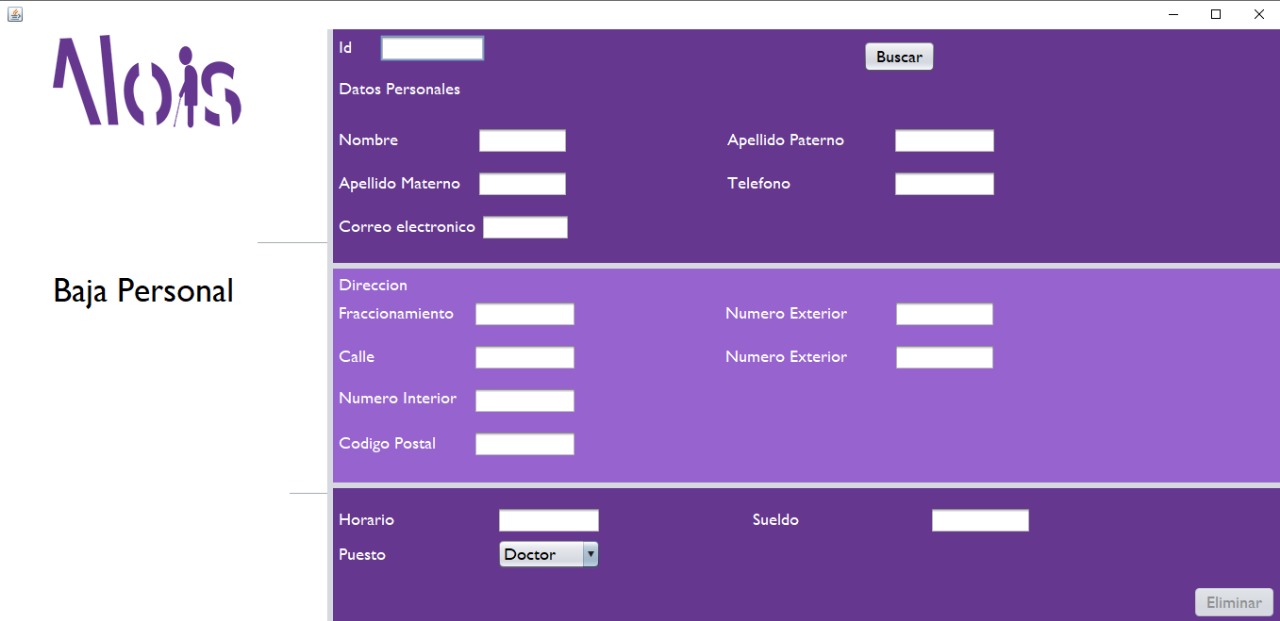


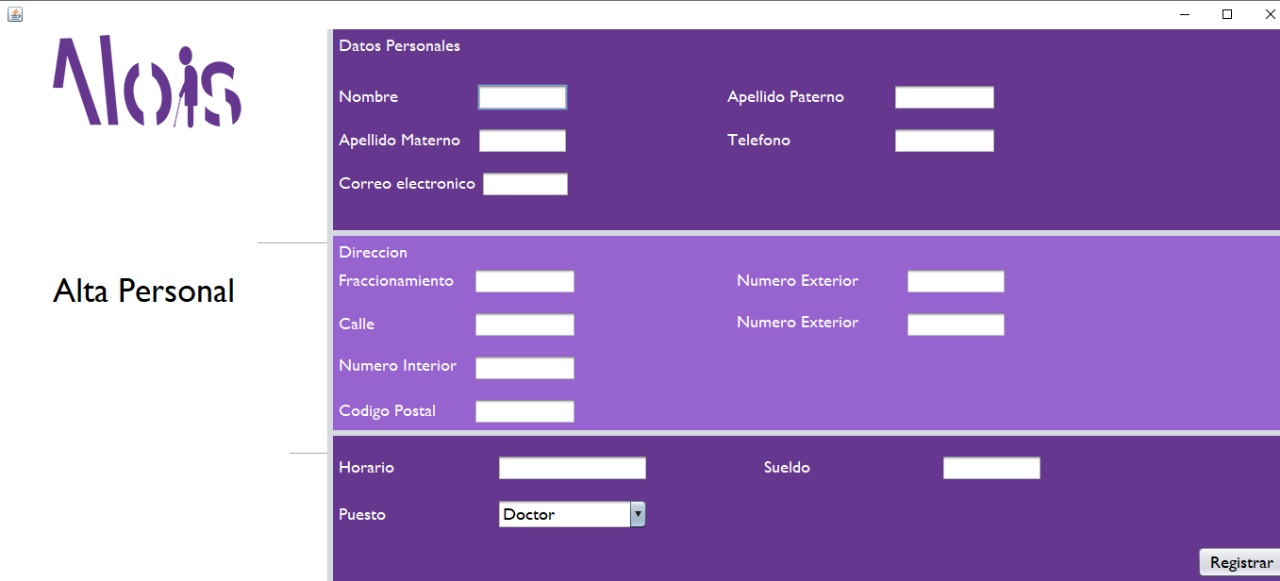
**Diseño de Interface Preliminar**

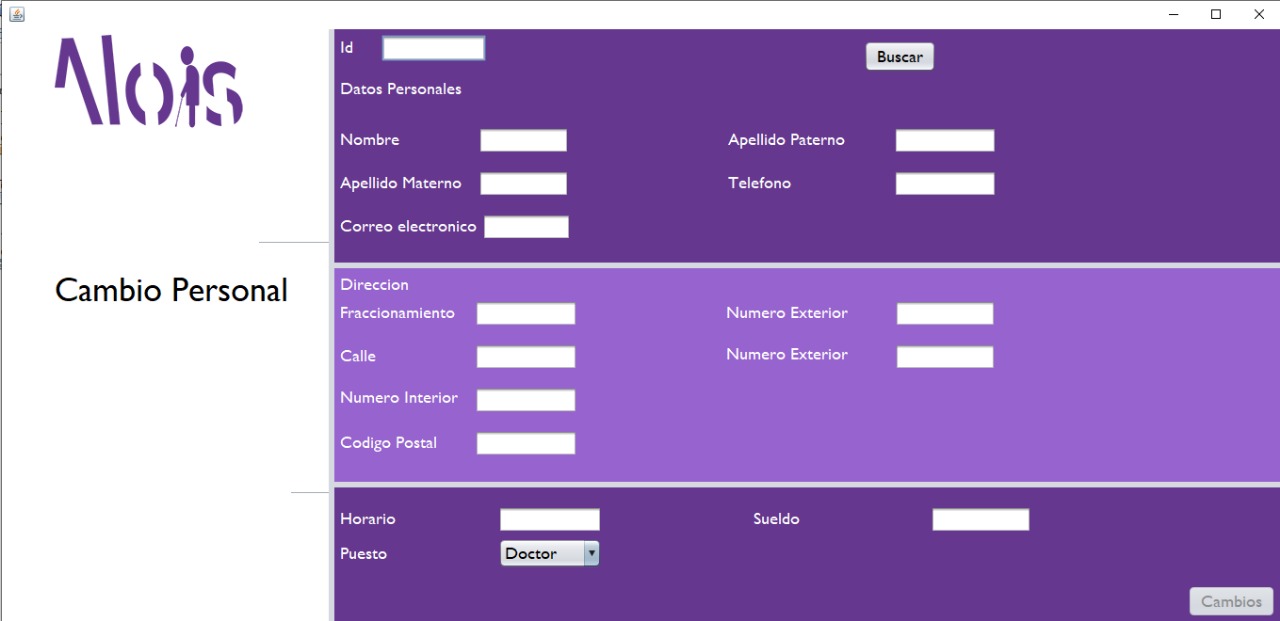
*Sistema de Administración de Asilo*

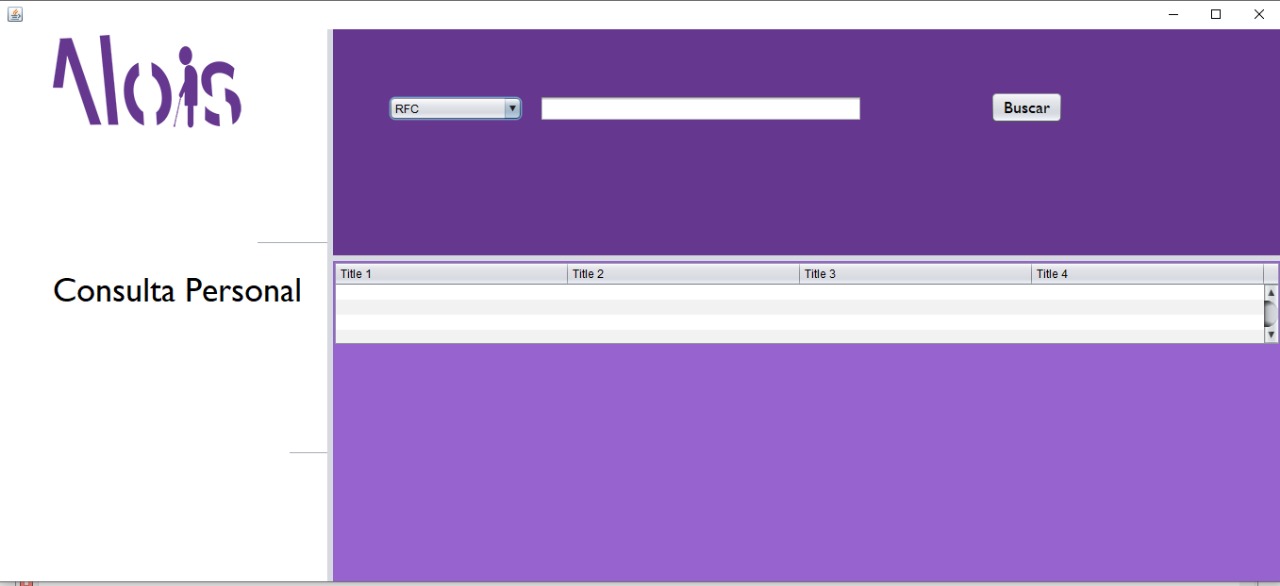
****

****

****

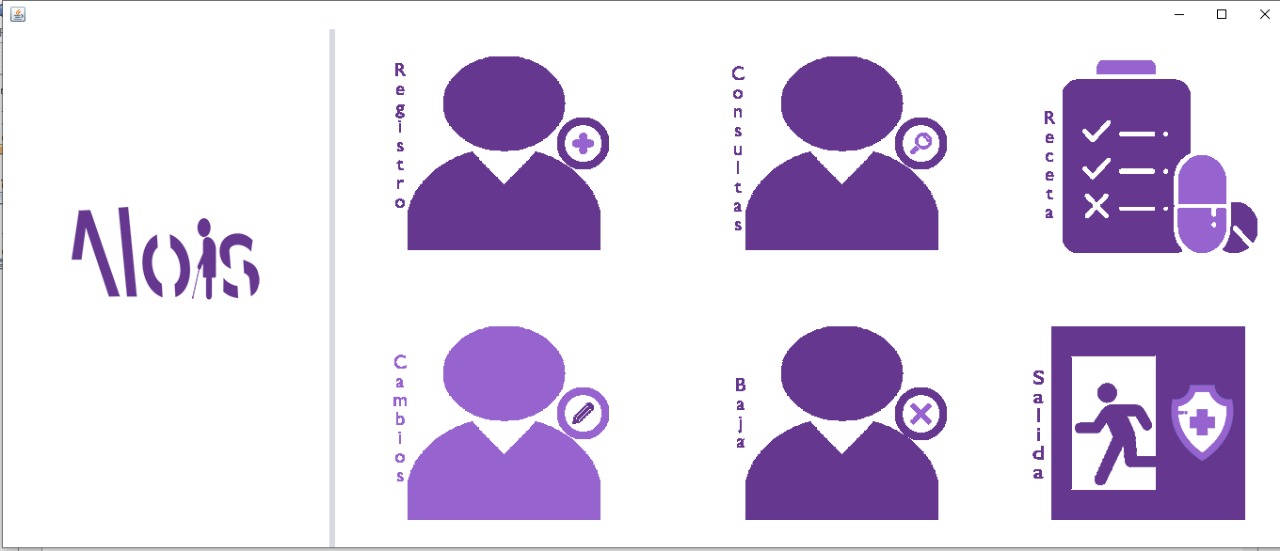
****

****

****

****

****

****

*Aplicación Móvil*

****

Para el sistema de administración de los asilos, la aplicación será utilizada en computadoras de escritorio o incluso computadoras portátiles, ya que son las que permiten un mejor manejo para el mismo.

Para la aplicación de apoyo al Alzheimer, se utilizarán dispositivos móviles con pantallas táctiles que le permitan a los usuarios un manejo más sencillo de la interfaz gráfica y pueda realizar las actividades más cómodamente y en cualquier lugar en donde esté.

La elección de color se escogió en base a una investigación de la gama de colores representativa de la enfermedad del Alzheimer, el cual se concluyó que era el morado. De igual manera, la fuente utiliza principalmente en el logotipo y en la aplicación móvil fue Sans Forgetica, la cual, según una investigación del RMIT, es una fuente que permite recordar de manera más sencilla el texto leído, de esta manera, desde la fuente contribuimos con el apoyo a esta enfermedad, así como el uso de la fuente Gill Sans en el sistema de administración, cuya tipografía es sumamente entendible enfocado en los usuarios que lo utilizan, los cuales necesitan un sistema sencillo de manejar.

**Diseño de Distribución de Base de Datos**

El sistema está diseñado para trabajar de la misma manera en todos los asilos que harán uso de él, puesto que les permitirá llevar un control administrativo de sus áreas. Por este motivo, la base de datos del sistema deberá de tener una estructura igual en los 4 sitios en los que vamos a implementarlo.

En cuestión a la base de datos, de acuerdo a lo planteado anteriormente tendremos una distribución que permita a los 4 sitios hacer su propia administración, por lo que en los cuatro se tendrán las mismas tablas de tal manera que puedan manejar su propia información de manera similar. Esto implica una fragmentación horizontal de la base de datos, cada sitio tendrá las mismas tablas con los mismos atributos, pero conteniendo diferente información.

A pesar de que existe una fragmentación horizontal en las tablas, las tablas como tal no estarán replicadas en ninguno de los sitios, puesto que cada sitio almacenará información diferente administrativamente, a pesar de que tendrán acceso a la información almacenada en los demás sitios.

La idea posteriormente es que la información completa del sistema pueda ser obtenida mediante querys o consultas enfocadas a este tipo de fragmentación.

Para el sistema de nombramiento, se manejará solamente un identificador del sitio únicamente, puesto que no se manejarán copias estrictas de las tablas, el cual estará colocado como un prefijo de la tabla dependiendo del sitio al que pertenezca. De esta manera, y apoyándonos con una tabla que nos de las direcciones IP de los sitios con respecto al prefijo de la tabla, podremos localizar el sitio al cual pertenece la tabla que se utilizará o que accederemos a su información.

