

TECNOLÓGICO DE COSTA RICA

ÁREA DE INGENIERÍA EN COMPUTADORES

CE3101 – Bases de Datos

Tarea Corta 3:

GasStationPharmacy

REALIZADO POR:

Campos Abarca Alejandro 2015035779

López Salas, Luis 2015088115

Murillo Morgan, Francisco J. 2015147156

Guzmán, Gerald 2016177609

PROFESOR:

Marco Rivera

I SEMESTRE DEL 2019 - GRUPO 2

Contenidos

[Manual de Usuario 3](#_Toc2719524)

[Introducción 7](#_Toc2719525)

[Descripción de los métodos: 7](#_Toc2719526)

[Estructuras de Datos 8](#_Toc2719527)

[Problemas Conocidos: 8](#_Toc2719528)

[Problemas Encontrados: 8](#_Toc2719529)

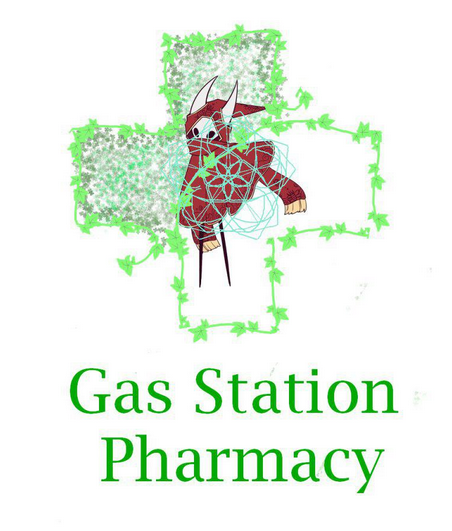
[Trabajo En Equipo: 9](#_Toc2719530)

[Conclusiones y Recomendaciones: 9](#_Toc2719531)

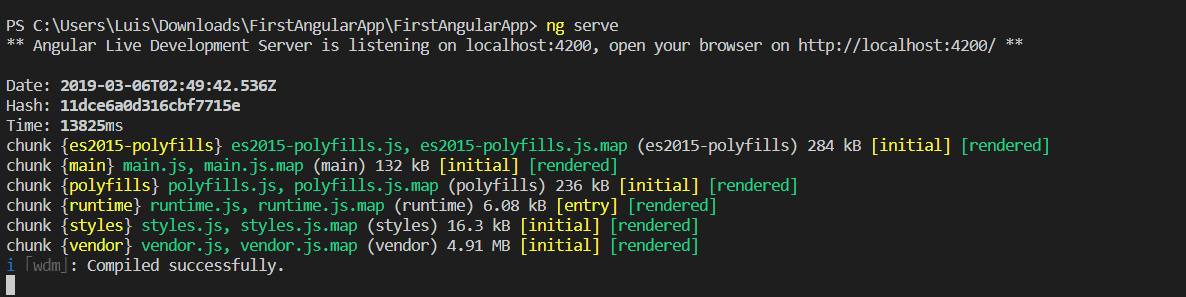
[Diagrama de Clases 10](#_Toc2719532)

[Bibliografía: 11](#_Toc2719533)

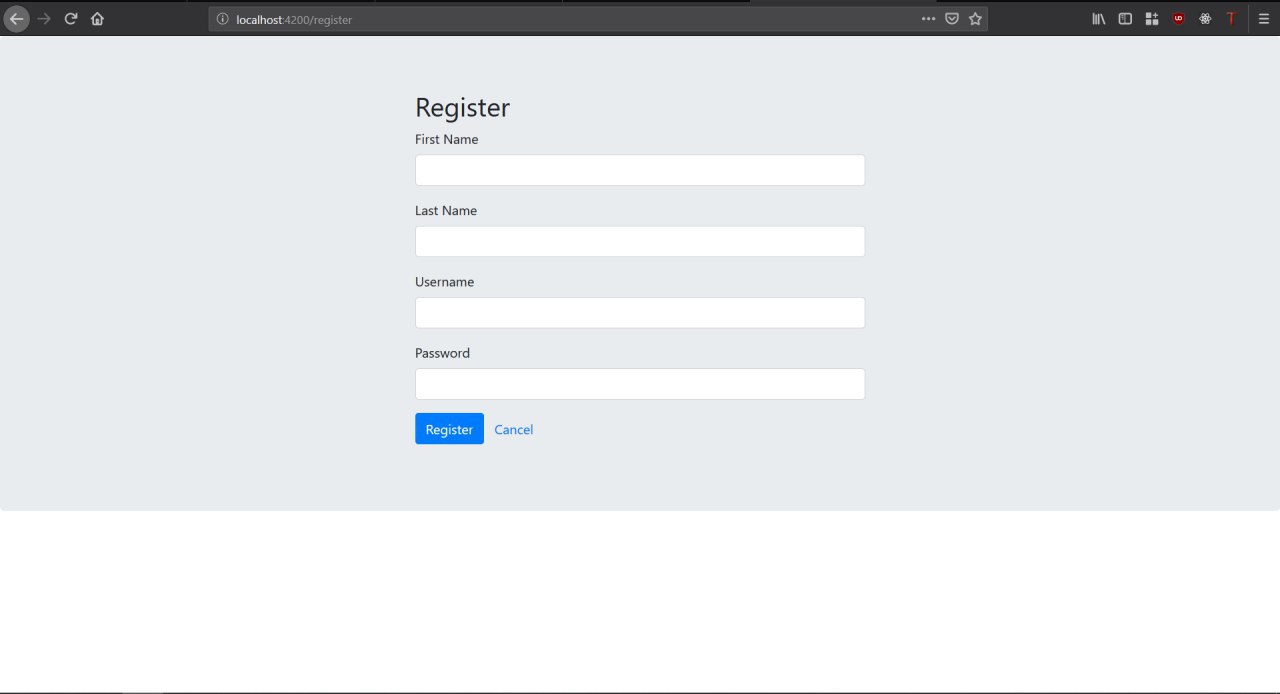
# Manual de Usuario



Para correr el servidor primero se debe de hacer un llamado a “ng serve” el cual crea una instancia del servidor para que corra el angular. De la siguiente forma:



Una vez que esto ya se cumple podemos ingresar a la primera pantalla que se tiene que es el Register:



Adonde hay 3 espacios:

1-Fisrt Name: Primer Nombre donde se debe escribir el primer nombre de la persona a inscribir.

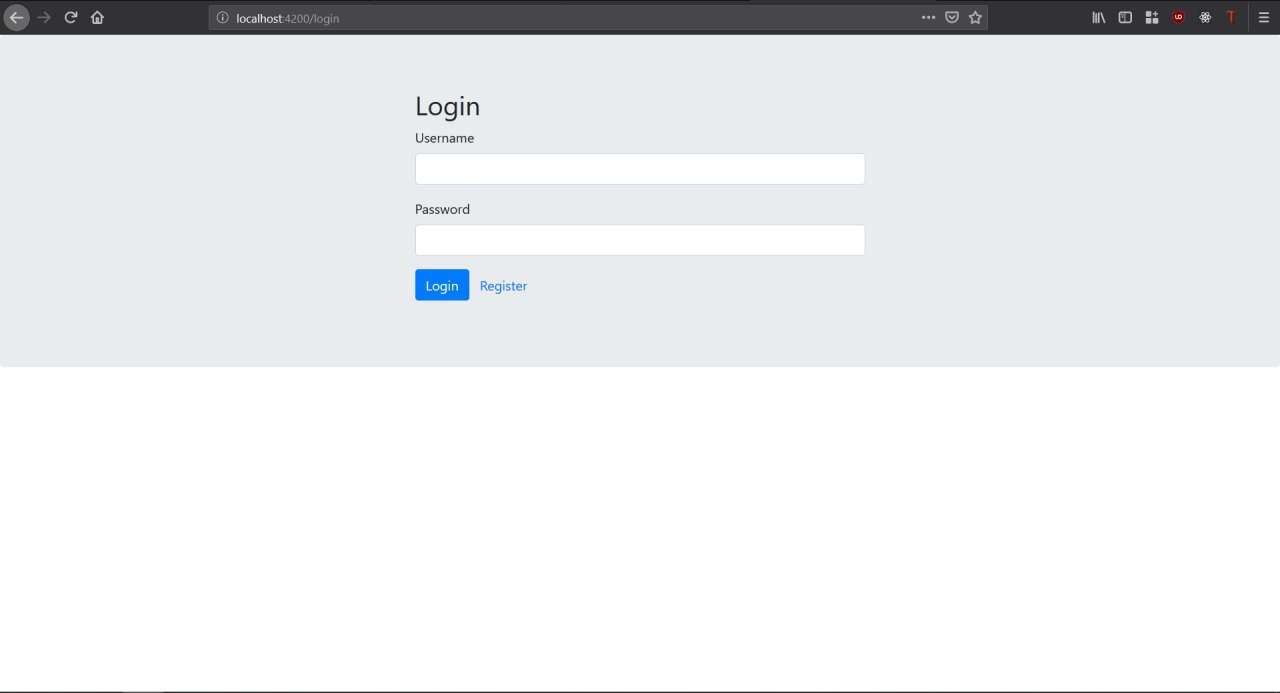
2-Last Name: Apellido de la persona a la cual se va a inscribir.

3-Username: nombre de usuario a la persona que se va a inscribir.

4-Password: Contraseña de la persona a la cual se va a escribir.

De aquí se tienen dos opciones que son los dos botones ubicados al final de la pantalla. Se puede inscribir a una nueva persona con los datos insertados o se puede cancelar la operación, esto depende de si se desea registrar a una nueva persona o no.

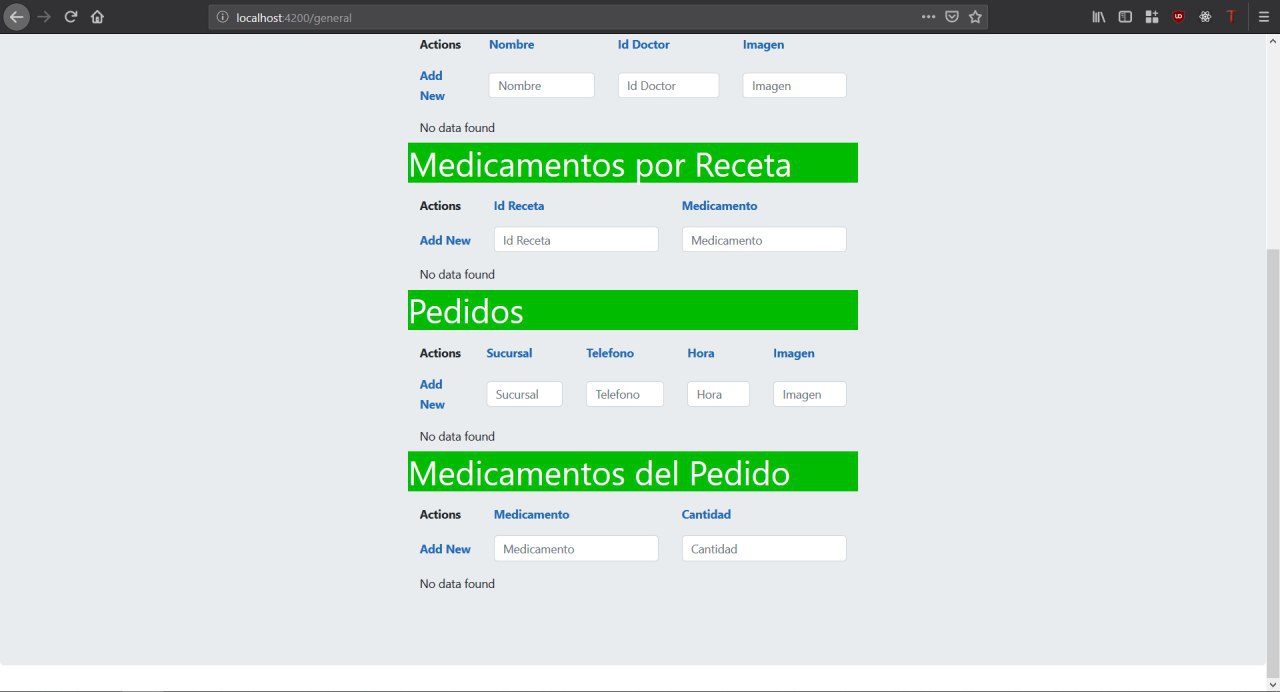
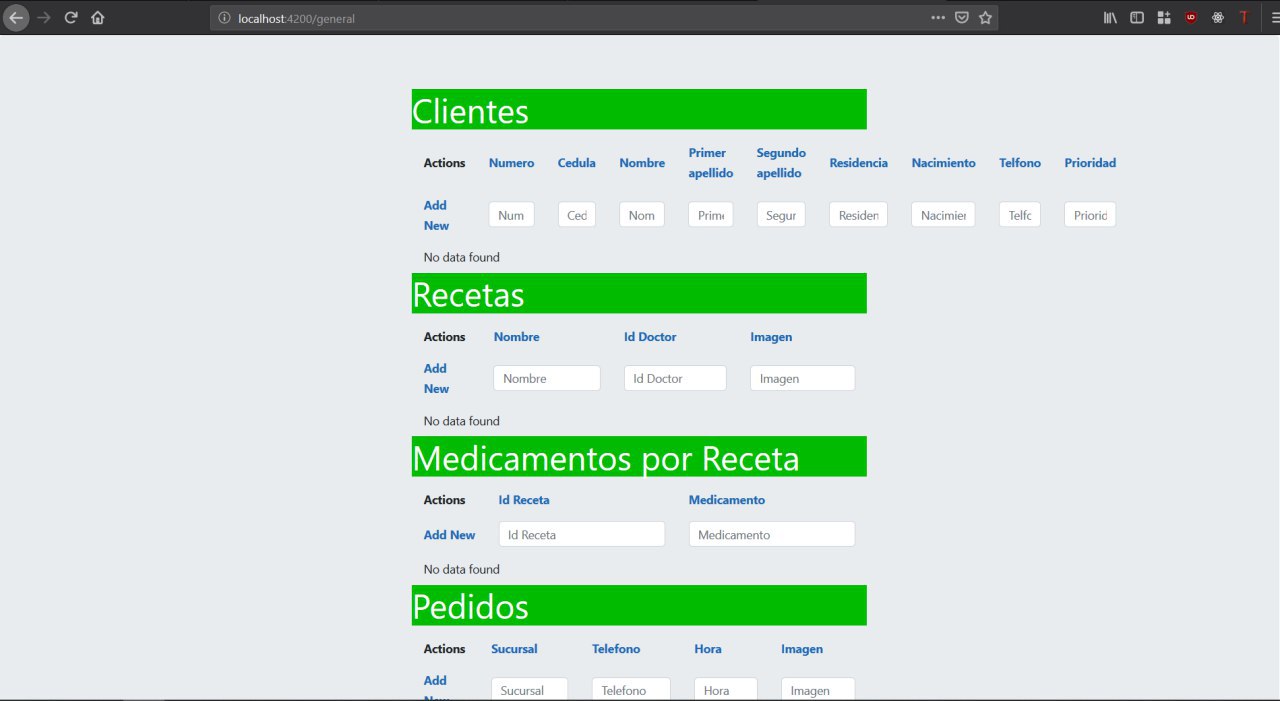
Otra pantalla que se puede acceder si ya se cuenta con un usuario es el log-in:



El log-in es parecido al Register, pero este solo se tiene dos campos:

1. Username: nombre de usuario asignado previamente.
2. Password: Es la contraseña previamente asignada, los dos tiene que coincidir para que se acepte el intento de log-in.

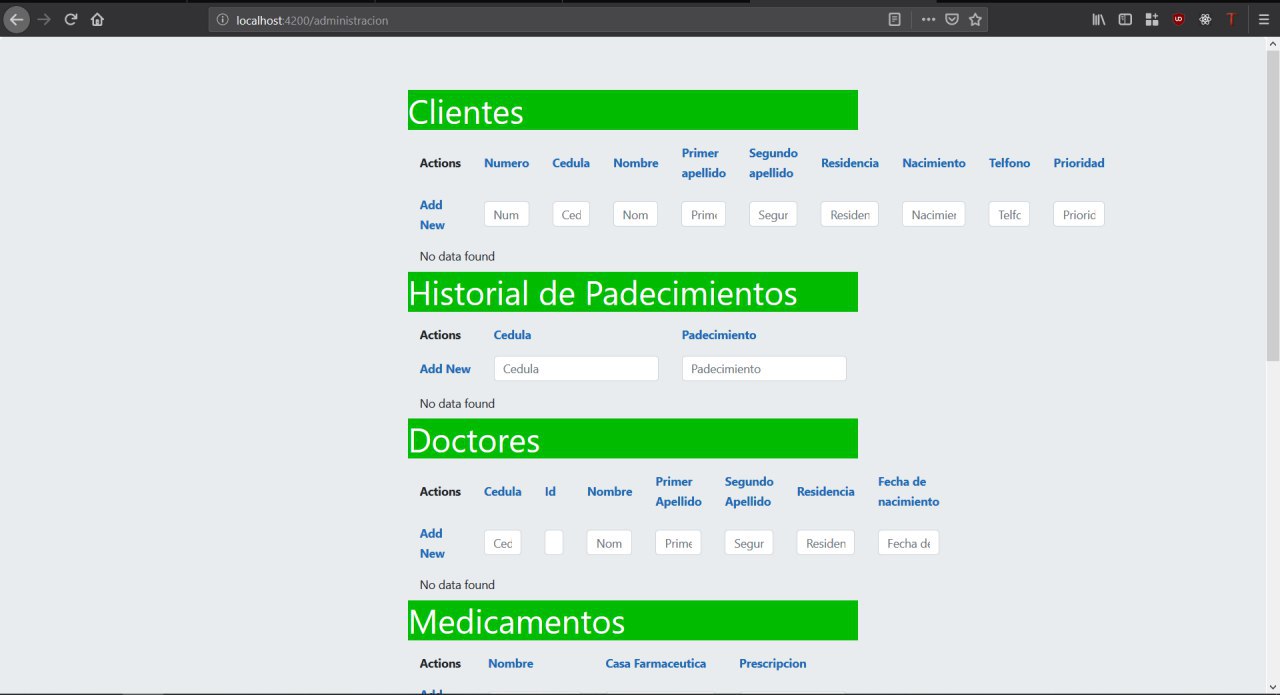
Si se lograra hacer iniciar una sesión en el sistema ya se podrían comenzar a crear varios valores como es posible desde la vista general:



En esta pestaña se ve que se puede hacer bastantes acciones, vamos a tomar una por una.

1. La creación de clientes, esto permite crear nuevos clientes a partir de los datos ingresados en las casillas, estas incluyen, Numero, Cedula, Nombre, Primer y Segundo Apellido, Fecha de Nacimiento, Teléfono y su prioridad.
2. La creación de recetas, esto se hace para poder ingresar recetas posteriormente, en este caso las recetas contienen 3 atributos, Nombre, doctor e imagen.
3. Medicamentos por receta, son los medicamentos que se le pueden agregar a una receta, incluye el ID de la receta y el medicamento a el cual le pertenece.
4. Pedidos, son los pedidos que se pueden hacer, para estos se debe especificar la sucursal, teléfono, hora e imagen (en caso de necesitar receta).
5. Medicamentos por pedidos, estos indican cuales son los medicamentos que se van a dar en un pedido, incluye el medicamento y una cantidad del mismo.

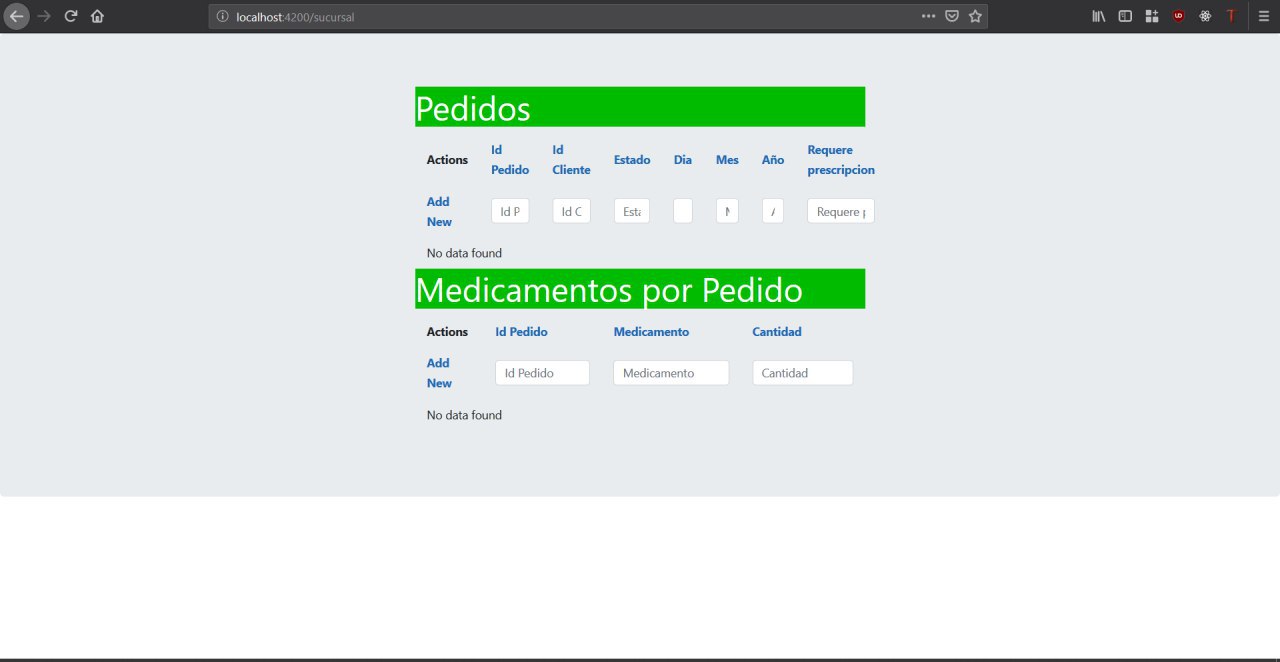
Faltan dos paginas de ver, la sucursal y la administración, para la administración es la vista que tienen los gerentes para hacer cambio y es la siguiente:



Esto incluye cuatro campos y son los siguientes:

1. Clientes, son los todos los clientes que se pueden ver y es capaz de crear cualquier cliente, estos tienen los mismos datos que en la página general.
2. Historial de padecimientos, son los padecimientos que tiene un cliente, para esto se especifica la cedula y su padecimiento.
3. Doctores, son los doctores que se pueden registrar en el sistema, para estos se pide, Cedula, ID, Nombre, Primer y segundo apellido, residencia y una fecha de nacimiento.
4. Medicamentos, son los medicamentos a registrar para estos se pide el nombre de medicamente, su casa farmacéutica y si esta ocupa prescripción.

Por ultimo se tiene la vista de sucursal:



Esta es la más simple de todas las vistas, solo incluye los pedidos y los medicamentos por pedido. Se hace referencia a todas las acciones que lleva acabo una sucursal.

1. Pedidos, son los pedidos que se pueden hacer. Para estos se pude un ID de pedido, un id de cliente, estado, día mes año del pedido además si este requiere una prescripción o no.
2. Medicamentos por pedido. Son los medicamentos que se tienen dentro de un pedido. Para estos se pide, un Id del pedido, los medicamentos que se tienen en el pedido, así como una cantidad de estos mismos.

# Introducción

En el curso de bases de datos se pidió analizar y efectuar un caso de estudio adonde se quiso simular una base de datos. Esto con el propósito de familiarizar a los estudiantes con el funcionamiento y desarrollo pertinentes los sistemas de bases de datos. Dado esto no se utilizó una base de datos como tal, si no se creó un modelo a partir de un WebService con modularidad REST para el manejo de todo dato.

# Descripción de los métodos:

Class1: crea el webservice, se encarga de recibir todas las solicitudes que se le mande a el servidor. No recibe ningún parámetro en su invocación ya que recibe todos los parámetros se pasan por parte de solicitudes HTTP.

Describe: Se utiliza para demarcar las intenciones que se tienen para la forma en que la página se va a desplegar en un navegador. Recibe varios parámetros pasados por angular y se encarga por parte del programador destacar cual es el propósito de cada atributo para así saber que es y cómo se debería de desplegar ese mismo atributo. Por ejemplo, un header (encabezamiento) debería estar demarcado como tal.

Dado a la naturaleza del proyecto, se utilizaron pocos métodos ya que, no se encontró el uso para más. Esto dado a que desde un punto de vista de funcionamiento no se hace la necesidad de más métodos. Se puede ver si se simplifica la viste de la tarea, solo se ocupan almacenar y acceder a datos conforme se necesiten. Esto lo hacen ya los frameworks y sus llamadas son realmente simples.

# Estructuras de Datos

En el proyecto no se utilizaron estructuras de datos propias. Se utilizó un framework Nancy el cual se encargó de todo manejo de datos por parte del servidor. Además, en las páginas HTML no se vio necesario el manejo de ninguna estructura de datos. Lo cual implica que toda estructura de datos utilizada en el proyecto no fue desarrollada para el proyecto, si no que su uso es general y proviene de bibliotecas externas. Esto sin hacer uso de ninguna tecnología ASPX como detalla la descripción de la tarea.

# Problemas Conocidos:

Al ser un prototipo o “mock-up” se sabía que este proyecto iba a tener una serie de problemas intrínsecos a su diseño y su funcionamiento. Uno de estos se ve a la hora de manejar la información, ya que esta no se almacenar por medio de una base de datos si no se va a almacenar en de archivos JSON. Lo cual limita la fluidez de los datos siendo esto la forma en que estos datos se relacionan entre sí e implica que se complique el diseño de almacenamiento de estos. Sería más fácil utilizar una base de datos ya que la manipulación de los datos, así como la relación que se tendrían entre estos datos seria directa. Ya que para este tipo de uso fueron creadas las bases de datos.

Por ser una aplicación local este programa se debe de correr a nivel local, es decir, dentro de una misma maquina o una red de computadores, lo cual limita el tiempo y recursos a la hora de desarrollar el proyecto por dos razones: Todo el equipo debe estar en una misma ubicación lo cual no es siempre factible por lo que se pierde tiempo ya que hay que elegir horas especificas adonde todos concuerden y es un poco pesado estar corriendo varios programas en una misma máquina, puede incluso llegar a retrasar las horas de programación porque hay que pasar actualizando todas las máquinas y estar solucionando problemas no relacionados a la tarea de cada individuo.

# Problemas Encontrados:

Cuando se intentó implementar el “log-in” se tuvo un problema ya que a la hora de diseñarse se pensó que el sistema debería de pedir que el usuario iniciara sesión apenas entre a la página, sin embargo, este debe permitir hacer ciertas acciones (como consultas) sin que se tenga a una cuenta activa en sesión. Esto implicó una remodelación del sistema de loggeo hacia el programa, adonde permita que un usuario acceso a ciertas partes del programa, pero no a otras. Para hacer esto se hace un sistema adonde detecte si el usuario estaba en sesión y permitirle acceder a diferentes partes. [3]

# Trabajo En Equipo:

|  |  |  |  |
| --- | --- | --- | --- |
| Integrantes | Actividad | Dia | Horas |
| Luis | Creación del App Móvil | 12/02/19 | 3 |
| Luis | Creación de Layouts | 16/02/19 | 4 |
| Luis | Edición de Layouts | 18 /02/19 | 5 |
| Luis | Pruebas de Servicio REST | 20/02/19 | 4 |
| Luis | Solución de errores Android | 24/02/19 | 6 |
| Alejandro | Creación del WebService | 24/02/19 | 4 |
| Alejandro | Modificaciones al WebService | 26/02/19 | 6 |
| Alejandro | Ligues dentro de los archivos JSON | 28/02/19 | 7 |
| Francisco | Creación de páginas con angular | 25/02/19 | 8 |
| Gerald | Modificaciones a paginas HTML | 26/02/19 | 4 |
| Francisco | Página Web mandar JSON | 02/03/19 | 5 |
| Gerald | Página Web recibir JSON | 04/03/19 | 4 |
| Francisco | Página Web últimos toques | 03/03/19 | 6 |

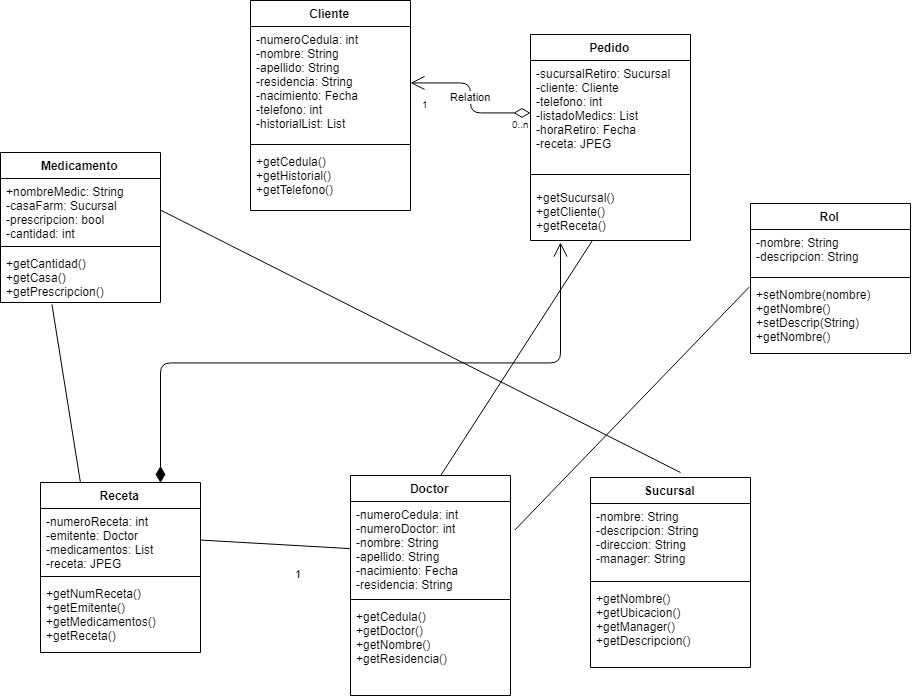
# Conclusiones y Recomendaciones:

La tarea abarca varios ámbitos importantes para entender el funcionamiento de una base de datos a un nivel apto para principiantes. Tiene todas las funciones principales como son subir y almacenar, editar y borrar datos. Si tiene ciertos inconvenientes que principalmente vienen de la manera a la cual se deben ligar los datos entre sus diferentes archivos (JSON, pero entiéndase tablas). Es complicado dado que no hay una forma de hacerlo directo, sino que se deben hacer conexiones a partir de búsquedas dentro de los mismos archivos y por parte del programa. Es decir, se deben de buscar estos por referencia y una vez que se encuentren se devuelven.

También ayuda al estudiante a aplicar y aprender como es el uso de HTML, así como las herramientas bootsrap y css. Lo cual va a ayudar con el resto del curso dado a que siempre se deben utilizar las páginas web tanto locales como no, más cuando ya se utilicen bases de datos como tales. Por ende, esta tarea corta se puede ver como una introducción en caliente (es decir aplicada) del resto del curso, ya que es aplicada y a su vez es más simple que los proyectos del resto del curso.

Seria de beneficio hacia la tarea y hacia el aprendizaje de los estudiantes que la tarea no fuera tan complicada para que así en vez de complicarse en cuanto a un diseño el cual puede estar más complejo de lo debido. Si esto no fuera así, se podría hacer enfoque más directo a la parte importante de la tarea la cual es modelar una base de datos. Ya que gran parte de la tarea ha sido la conexión entre varias “tablas” en vez de confeccionar un modelo el cual se viera mejor reflejado como una base de datos.

# Diagrama de Clases



En el diagrama anterior se planteó la solución de la tarea a través de clases que se creyeron necesarias para los módulos de gestión que son la base del servicio REST.

Clase Doctor: Esta clase es la encargada de mantener la información relacionada a los doctores en las farmacias correspondientes la cual es: Número Cédula, Número de doctor, Nombre, Apellidos, Lugar de Residencia, Fecha de nacimiento. Esta clase es la que más relaciones tiene con las demás.

Clase Cliente: esta clase contiene la siguiente información de cliente: Número Cédula, Nombre, Apellidos, Lugar de Residencia, Fecha de nacimiento, teléfono y una breve historial de los padecimientos. La clase cliente se relaciona con Pedido solamente.

Clase Pedido: esta clase contiene la siguiente información: sucursal de recojo, cliente, teléfono preferido, listado de medicamentos, hora de recojo, imagen de la receta. Esta clase se relaciona con Cliente, Doctor, y Receta

Clase Sucursal: esta clase contiene la información: nombre, descripción, dirección y

administrador. Se relaciona con Medicamento únicamente.

Clase Medicamento: esta clase guarda la información: nombre,

casa farmacéutica, requiere prescripción (Si/No), cantidad disponible. Se relaciona con Sucursal y Receta.

Clase Receta: esta clase guarda: número de receta, doctor que emitió la receta, medicamentos prescritos y debe permitir cargar una imagen de la receta física. Se relaciona con Medicamento y Doctor.

Clase Rol: esta clase guarda únicamente: nombre y descripción y se relaciona únicamente con Doctor

# Bibliografía:

[3] Angular 6 - User Registration and Login Example & Tutorial. (2018, May 16). Retrieved March 1, 2019, from <http://jasonwatmore.com/post/2018/05/16/angular-6-user-registration-and-login-example-tutorial>

AngularJS (2019-02-05). Recuperado de: https://angular.io/Bootstrap Themes & Templates (2019-02-05). Recuperado de: https://getbootstrap.com/docs/4.1/getting-started/introduction