



**UNIVERSIDAD ANDRÉS  
BELLO FACULTAD DE  
INGENIERÍA**

**INGENIERÍA CIVIL EN  
INFORMÁTICA**

**ENTREGA HITO 2:**

**PLANIFICACION DE RUTAS DE  
HOSPITALIZACION  
DOMICILIARIA**

**Ítalo Cararo Álvarez**

**Fabian Cárdenas Correa**

**Jordan Cárdenas Mercado**

**André Ducheylard Lolic**

**Alex Soto Herrera**

## **Índice:**

Introducción.....	3
Planificación de las rutas.....	4
Implementación de la base de datos.....	4
Casos de uso aplicación móvil.....	5
Requerimientos no funcionales.....	14
Hallazgos encontrados.....	14
Vistas preliminares de la aplicación móvil.....	15
Conclusión.....	17

### **Introducción**

El presente documento referente al sprint 2 del proyecto, tendrá como fin evidenciar los avances planteados en el sprint 1, que consta con la implementación de una base de datos para la comunicación entre las aplicaciones web y móvil, una demostración a priori de la aplicación móvil complementado de algunos casos de uso, y además, evidenciar los hallazgos encontrados donde se muestra todo lo que se logro encontrar del proyecto anterior.

### **Planificación de las rutas**

El algoritmo de la planificación de las rutas principalmente corre en base a dos Apis, MapQuest y LocationIQ para hacer el cálculo de las distancias entre las direcciones, luego de calcular las distancias se ejecuta un algoritmo para optimizar por distancia más una heurística basada en la prioridad del paciente, esta última valuada en 5 posibles prioridades: “muy alta, alta, normal, baja, muy baja”, esta luego de generar la ruta se redirecciona automáticamente a Google maps para mostrar las rutas en el orden de prioridad obtenido.

### **Implementación de la base da datos**

La implementación de la base de datos se basa principalmente en una migración desde lo que era la aplicación web implementada en una base de datos local junto con Django, luego de dicha migración, paso a ser implementada en Amazon Web Service montada para el uso de ambas aplicaciones, tanto la web como la móvil, todo esto para unificar la base de datos y que no sé tuviera que hacer una base de datos adicional para la implementación de la aplicación móvil y tener que generar una tercera base de datos que las interconectara.

### Casos de uso aplicación móvil

Como se mostrará a continuación se mostrará parte de los casos de usos totales, como una ejemplificación del flujo “normal” que debería hacer el usuario (encargado de la visita) para llegar a la finalidad del caso de uso.

Especificación de caso de uso: Atender Paciente

## 1. Brief Description

El objetivo es explicar cómo se realiza una atención domiciliaria desde que el doctor va en camino hasta que el doctor termina de revisar al paciente.

## 2. Basic Flow of Events

Paso	Rol	Acción.
1	Sistema	Muestra login con Usuario y contraseña para que ingrese el doctor
2	Usuario	Doctor ingresa su usuario y contraseña
3	Sistema	Valida que el usuario y la contraseña estén correctas
4	Sistema	El sistema muestra el Home.
5	Usuario	Selecciona la opción de Iniciar Ruta
6	Sistema	Muestra un mapa con la ruta que debe seguir el doctor hasta la casa de los pacientes.
7	Usuario	Sigue la ruta en su automóvil hasta la casa del paciente.

8	Sistema	Se muestra la vista “Destino alcanzado”
9	Usuario	El usuario pulsa el botón “Iniciar Visita”
10	Sistema	Muestra la vista “Visita”
11	Usuario	Presiona el botón “Terminar visita”
12	Sistema	Muestra la vista “Visita finalizada” y pide al usuario rellenar el formulario de la visita.
13	Usuario	Rellena el formulario y presiona el botón “Guardar”
14	Sistema	Guarda la información dada por el usuario en la base de datos.

### 3. Alternative Flows

#### 3.1 No se valida el usuario

Paso	Rol	Acción
3.1.1	Sistemas	El sistema no valida al usuario y/o contraseña.
3.1.2	Sistema	Envía el mensaje “Usuario y/o contraseña incorrecta”.
3.1.3	Sistema	Vuelve al paso 1.

#### 7.1 Doctor se equivoca en ruta.

Paso	Rol	Acción
7.1.1	Usuario	Desvía de la ruta marcada por el mapa.
7.1.2	Sistema	Actualiza la ruta para llegar a su destino.

#### 4. Preconditions:

- 4.1 El doctor se comunicó previamente para agendar la cita con el paciente
- 4.2 El paciente se encuentra en su domicilio
- 4.3 El doctor cuenta con un automóvil
- 4.4 El doctor y el paciente se encuentran registrados en la base de datos
- 4.5 El doctor presiona correctamente los botones.
- 4.6 El doctor no se equivocará al momento de llenar el formulario al final de la visita.

**5. Postconditions:**

- 5.1 El doctor revisa si tiene citas próximas a realizar
- 5.2 El formulario queda guardado en la base de datos

**6. Special Requirements:**

- 6.1 Tanto el doctor como el paciente tienen que estar disponibles a la hora de la visita.
- 6.2 El doctor debe tener conexión a internet al momento de usar la App.

Especificación de caso de uso: Revisar el historial del paciente

## 1. Brief Description

El objetivo es mostrar el proceso para llegar a ver un historial de un paciente agendado para hoy.

## 2. Basic Flow of Events

Paso	Rol	Acción.
1	Sistema	Muestra login con Usuario y contraseña para que ingrese el doctor.
2	Usuario	Doctor ingresa su usuario y contraseña
3	Sistema	Valida que el usuario y la contraseña estén correctas
4	Sistema	El sistema muestra el Home.
5	Usuario	Selecciona la opción de "Visitas Asignadas".



6	Sistema	Muestra la vista "Visitas Asignadas".
7	Usuario	Selecciona el paciente al cual desea ver el historial.
8	Sistema	Se muestra la vista "Detalle Paciente"
9	Usuario	El usuario pulsa el botón "Historial de visita"
10	Sistema	Muestra la vista "Historial de visita"
11	Usuario	Elige la visita que quiera mirar.
12	Sistema	Muestra la vista "Visita finalizada" y pide al usuario rellenar el formulario de la visita.
13	Usuario	Rellena el formulario y presiona el botón "Guardar"

### 3. Alternative Flows

#### 3.1 No se valida el usuario

Paso	Rol	Acción
3.1.1	Sistemas	El sistema no valida al usuario y/o contraseña.
3.1.2	Sistema	Envía el mensaje “Usuario y/o contraseña incorrecta”.
3.1.3	Sistema	Envía el mensaje “El usuario no tiene permisos para acceder”.
3.1.4	Sistema	Vuelve al paso 1.

#### 4. Preconditions:

- 4.1 El doctor y el paciente se encuentran registrados en la base de datos
- 4.2 El doctor presiona correctamente los botones.

#### 5. Postconditions:

----

#### 6. Special Requirements:

- 6.1 El doctor debe tener conexión a internet al momento de usar la App.

Especificación de caso de uso: Editar la información de un doctor.

## 1. Brief Description

El objetivo es mostrar el proceso para que un doctor pueda editar sus datos.

## 2. Basic Flow of Events

Paso	Rol	Acción.
1	Sistema	Muestra login con Usuario y contraseña para que ingrese el doctor.
2	Usuario	Doctor ingresa su usuario y contraseña
3	Sistema	Valida que el usuario y la contraseña estén correctas
4	Sistema	El sistema muestra el Home.
5	Usuario	Selecciona la opción de "Ver perfil".
6	Sistema	Muestra la vista "Ver Perfil".
7	Usuario	Selecciona el botón "Editar perfil"
8	Sistema	Se muestra la vista "Editar Perfil"
9	Usuario	El usuario modifica los datos.

10	Usuario	Selecciona el botón “Guardar Cambios”
11	Sistema	Valida los datos ingresados.

### 3. Alternative Flows

#### 3.1 No se valida el usuario

Paso	Rol	Acción
3.1.1	Sistemas	El sistema no valida al usuario y/o contraseña.
3.1.2	Sistema	Envía el mensaje “Usuario y/o contraseña incorrecta”.
3.1.3	Sistema	Envía el mensaje “El usuario no tiene permisos para acceder”.
3.1.4	Sistema	Vuelve al paso 1.

#### 3.2 Doctor se equivoca en ingresar algún dato.

Paso	Rol	Acción
11.1.1	Usuario	Ingresa incorrectamente un dato.
11.1.2	Sistema	Envía el mensaje “Datos ingresados inválidos”.
11.1.3	Sistema	Vuelve al paso 8.

**4. Preconditions:**

- 4.1 El doctor se encuentra registrado en la base de datos.
- 4.2 El doctor presiona correctamente los botones.

**5. Postconditions:**

---

**6. Special Requirements:**

- 6.1 El doctor debe tener conexión a internet al momento de usar la App.

## **Requerimientos no funcionales**

- Uso de base de datos PostgreSQL
- Uso de las Apis MapQuest y LocationIQ
- Se trabaja en los servidores de Amazon Web Service

## **Hallazgos encontrados**

1-Los datos no están asociados a las tablas correspondientes, están esparcidos en diferentes tablas. Por ejemplo; Existen datos presentes en algunas tablas que tendrían mejor sentido y resultarían en un trabajo más eficiente si se movieran a otras tablas que tuvieran mayor relación con dicho atributo.

2-No hay una coherencia o hilo en las tablas creadas. Por ejemplo; Hay mucha redundancia de datos y ciclos entre tablas que dificultan el entendimiento y el trabajo sobre las mismas.

3-Está conectada para que funcione fuertemente con el framework Django, haciendo la estabilidad muy difícil y casi nula. Por ejemplo; Si se quisiera agregar una comuna extra al alcance de los equipos médicos, se tendría que modificar los usuarios y permisos de las tablas internas de Django ya que dicho modelo que refleja las comunas está programado en una sección errónea del framework.

4-Hay tablas que no tienen una conexión con el diagrama central o con las tablas de pacientes/usuarios. por ejemplo; tablas de visita\_tiempos, visita\_llamar, biblioteca\_archivo. En la práctica ninguna de esas tablas aparece conectada en el modelo de Django, pero logran funcionar igual.

5-No hay una ventana clara para poder agregar funciones o soluciones en un futuro, debido a lo "cerrado de mente" con la que se modelo la base de datos. Por ejemplo; Al intentar agregar un algoritmo de optimización de ruta, se hace muy difícil el manejo de datos tanto para escritura como lectura ya que no en todas las tablas se pueden modificar/agregar atributos. (está relacionado con el punto 3)

6- Enlazado con el punto 3, existen problemas con la migración de la base de datos, ya que Django la trabaja con mysqlite, dicho problema se presentó al momento de migrar a MySQL afectando las tablas internas auto-generadas por Django. Se desconoce si otros sistemas de bases de datos presentan problemas al igual que con MySQL

### **Soluciones aplicadas**

Para los hallazgos 1 y 2 no se encuentra ninguna solución factible, ya que modificar la estructura de la base de datos hacia inoperable la versión web.

En el caso de los hallazgos 3 y 5 se han modificar algunas tablas solamente para lograr un bypass a los problemas que han surgidos.

Para el hallazgo número 4 todavía no encontramos una solución, ya que aún no hemos trabajado con dichas tablas, pero para la entrega del hito 3 se deberían de trabajar.

6- Finalmente se migró a PostgreSQL ya que no presentó ningún problema.

## Vistas preliminares de la aplicación móvil



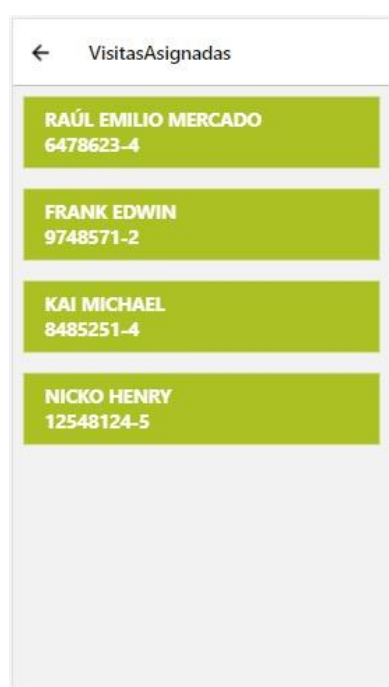
Vista login



Vista home



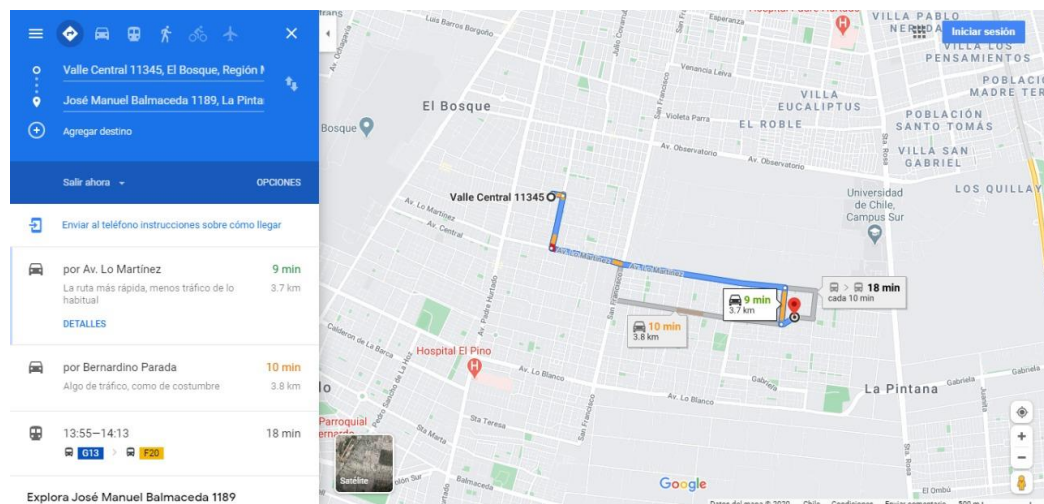
Vista Perfil



Vista visitasAsignadas



Vista Mapa



Vista ruta desplegada "nota: se abre Google maps para mostrar la ruta"



### **Conclusión**

En resumen, se logró solucionar gran parte de las problemáticas planteadas dentro de la sección de hallazgos, se avanzó una parte considerable en la programación de las vistas de la aplicación móvil. La implementación de la base de datos alojada en AWS es uno de los más grandes avances ya que nos permite conectar la aplicación móvil junto con la web y así lograr intercambio de datos, consultas, entre otros. Para el hito 3, se tiene previsto montar las Apis a servicios en la nube como AMW o Azure.