



2025

Plataforma web materno360 para la gestión digital de controles prenatales y seguimiento de embarazos del centro de salud del municipio del Palmar

MANUAL TÉCNICO

QUETZALTENANGO

V.0.1

Para más información, por favor contactase

Gilmar Ariel Puac Acabal

(502) 5733 2954

gpuacal@miumg.edu.gt

Contenido

Acerca del proyecto	4
1. Requerimientos del sistema	5
1.1 Requerimientos de hardware	5
1.2 Requerimientos de software	5
2. Instalación del entorno.....	5
3. Ejecución de migraciones:.....	6
4. Estructura del proyecto	7
4.1 Carpetas principales.....	8
5. Componentes del sistema	9
6. Modelo de Base de Datos	10
7. Diagramas UML	10

Acerca del proyecto

El presente manual describe el uso del sistema Materno360, una aplicación web local desarrollada para la gestión y seguimiento de mujeres embarazadas. El sistema está diseñado para el personal médico y administrativo, facilitando el registro, control, antecedentes y generación de informes de pacientes.

1. Requerimientos del sistema

1.1 Requerimientos de hardware

Componente	Requisito mínimo	Recomendado
Procesador	Intel Core i3	Intel Core i5 o superior
Memoria RAM	4 GB	8 GB
Almacenamiento	256 GB SSD	512 GB SSD
Sistema Operativo	Windows 7 / 10 / 11	Windows 10 / 11

1.2 Requerimientos de software

Componente	Versión recomendada	Descripción
PHP	8.2 o superior	Lenguaje base del framework Laravel
Composer	Última versión	Gestor de dependencias de PHP
Laravel	10.x	Framework principal del sistema
MySQL	8.0	Gestor de base de datos
Visual Studio Code	Actualizada	Entorno de desarrollo
Navegador Web	Chrome, Edge, Firefox	Para visualizar el sistema

2. Instalación del entorno

Paso 1. Clonar o descargar el proyecto

Si se obtiene desde un repositorio: <https://github.com/Gilmar29/Sistema-Materno-360.git>

Hay dos opciones para ejecutar el proyecto:

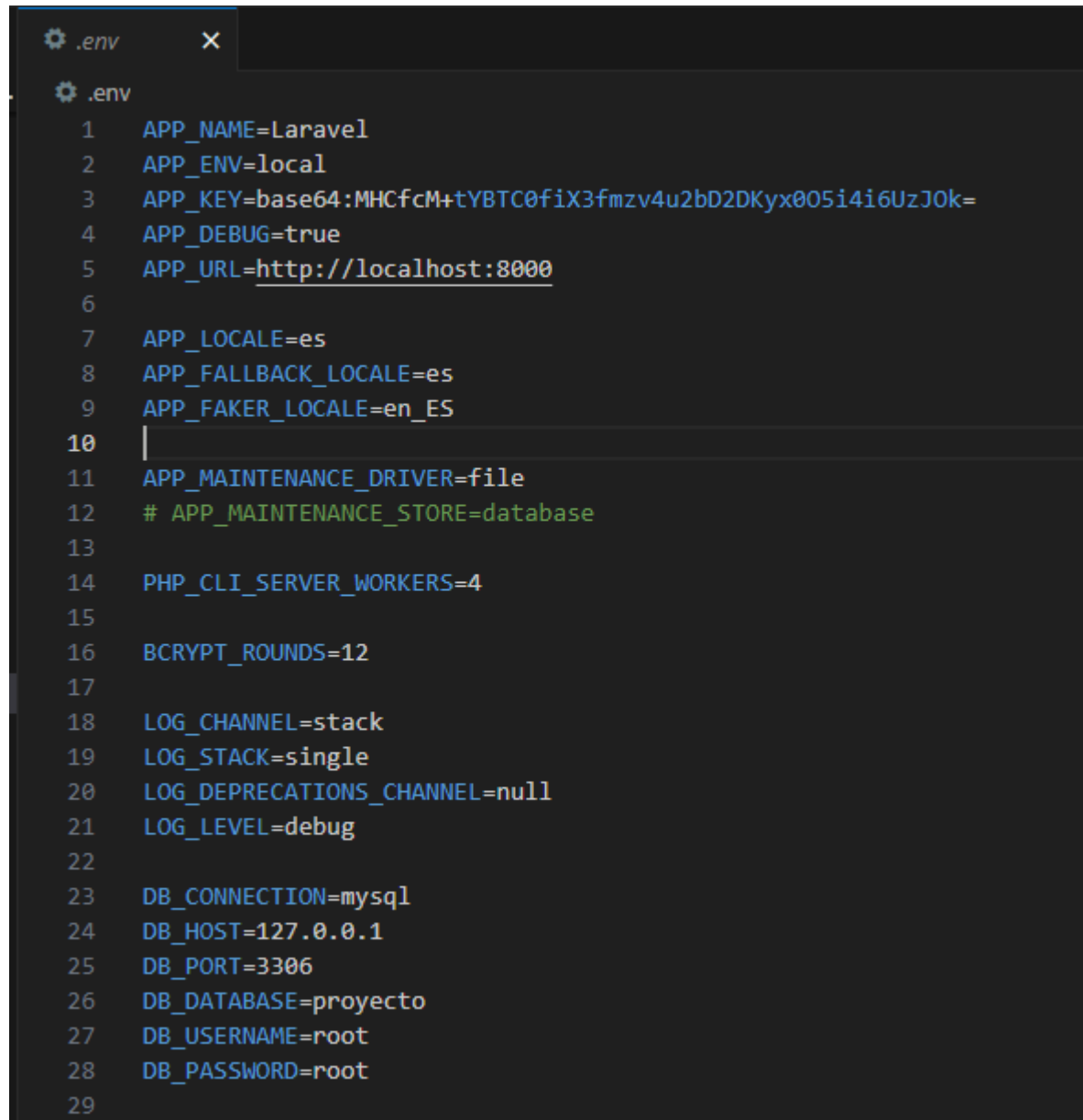
- Clonar
- Descargar el proyecto en formato ZIP

Paso 2. Instalar dependencias

Ejecutar desde la terminal de VS Code:
composer install

Paso 3. Configurar el archivo .env

Se actualizó la conexión a la base de datos con las credenciales de MySQL Workbench:



```
.env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:MHCfcM+tYBTC0fiX3fmzv4u2bD2DKyx005i4i6UzJ0k=
4 APP_DEBUG=true
5 APP_URL=http://localhost:8000
6
7 APP_LOCALE=es
8 APP_FALLBACK_LOCALE=es
9 APP_FAKER_LOCALE=en_ES
10
11 APP_MAINTENANCE_DRIVER=file
12 # APP_MAINTENANCE_STORE=database
13
14 PHP_CLI_SERVER_WORKERS=4
15
16 BCRYPT_ROUNDS=12
17
18 LOG_CHANNEL=stack
19 LOG_STACK=single
20 LOG_DEPRECATED_CHANNEL=null
21 LOG_LEVEL=debug
22
23 DB_CONNECTION=mysql
24 DB_HOST=127.0.0.1
25 DB_PORT=3306
26 DB_DATABASE=proyecto
27 DB_USERNAME=root
28 DB_PASSWORD=root
29
```

3. Ejecución de migraciones:

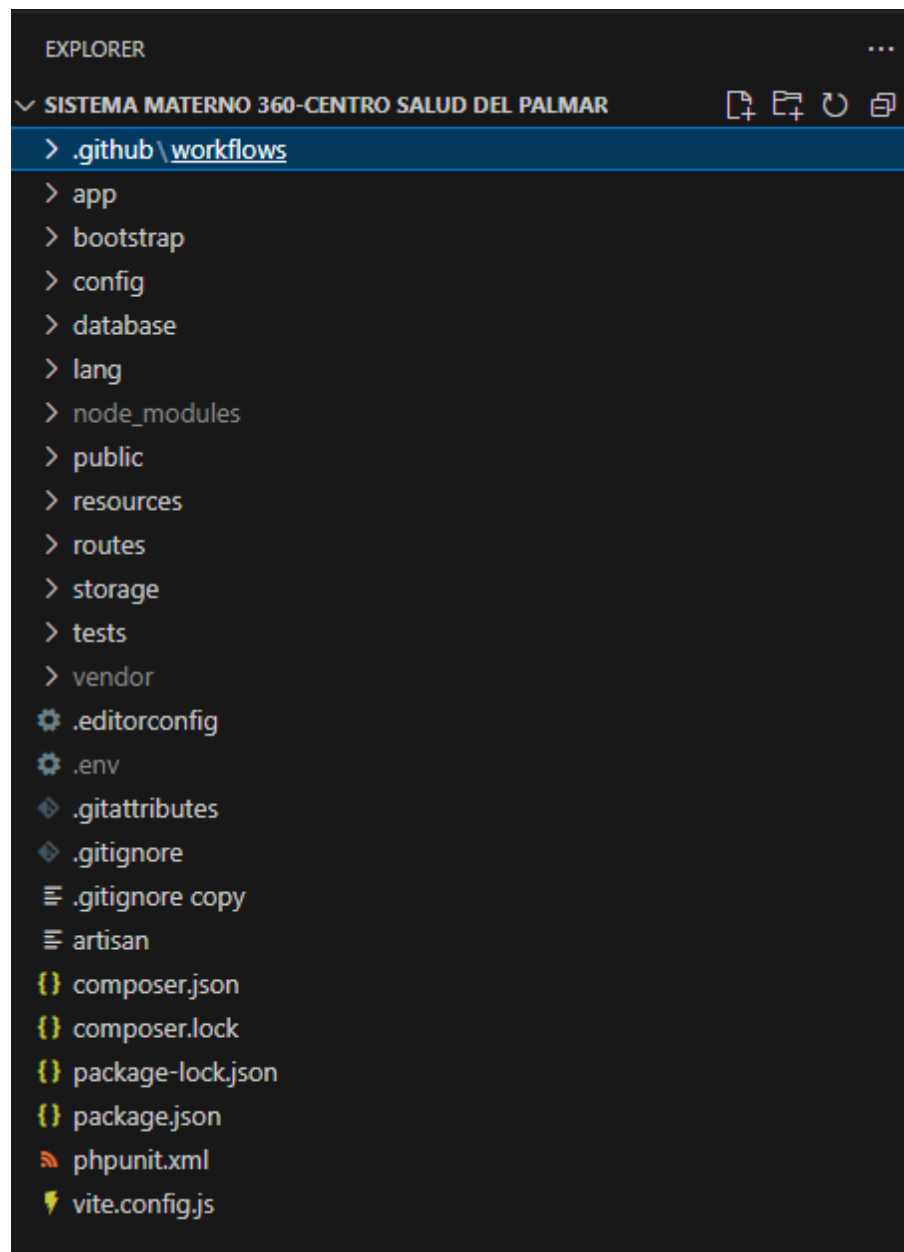
```
php artisan migrate
```

Esto generó todas las tablas necesarias (pacientes, usuarios, roles, antecedentes, controles, etc.) en la base de datos.

El sistema quedó disponible en la dirección: <http://127.0.0.1:8000>

4. Estructura del proyecto

La estructura principal del sistema está dividida en carpetas según la organización del framework Laravel:



4.1 Carpetas principales

Carpeta / Archivo	Descripción	Cómo se genera
.github/workflows	Contiene archivos YAML usados para la integración continua (CI/CD) con GitHub Actions. Por ejemplo, ejecutar pruebas o desplegar automáticamente.	Se crea manualmente o al configurar GitHub Actions.
app/	Aquí está el código principal de la aplicación: controladores, modelos y lógica del negocio.	Se genera por defecto con <code>laravel new</code> o <code>composer create-project</code> .
bootstrap/	Contiene el archivo que inicializa el framework Laravel, cargando configuraciones y dependencias.	Se genera automáticamente con Laravel.
config/	Archivos de configuración del sistema (base de datos, mail, cache, etc.).	Incluido por Laravel.
database/	Contiene migraciones, seeders y factories para definir y poblar la base de datos.	Se crea automáticamente; puedes agregar migraciones con <code>php artisan make:migration</code> .
lang/	Archivos de traducción (mensajes del sistema en varios idiomas).	Generado por Laravel.
node_modules/	Dependencias instaladas de Node.js (para Vite, Vue, React, etc.).	Se genera con <code>npm install</code> .
public/	Carpeta pública del servidor. Aquí se encuentra <code>index.php</code> , punto de entrada de Laravel. También guarda imágenes, CSS y JS compilados.	Parte del esqueleto de Laravel.

resources/	Contiene las vistas (Blade o Vue/React), archivos CSS, JavaScript y componentes front-end.	Se crea por defecto.
routes/	Define las rutas del sistema (web.php, api.php, etc.).	Generado por Laravel.
storage/	Guarda logs, archivos subidos, caché, etc.	Laravel la crea automáticamente.
tests/	Aquí van las pruebas automáticas del sistema (PHPUnit).	Incluido por Laravel.
vendor/	Contiene todas las dependencias de PHP instaladas con Composer.	Se genera al ejecutar composer install.

5. Componentes del sistema

El sistema Materno360 cuenta con los siguientes módulos:

1. Gestión de Pacientes

Permite registrar, editar, eliminar y visualizar la información general de las pacientes embarazadas.

2. Antecedentes Obstétricos

Registra la información médica previa de cada paciente (número de embarazos, partos, cesáreas, abortos, etc.).

3. Controles Médicos

Módulo donde se registran los controles prenatales, resultados y observaciones del médico.

4. Usuarios y Roles

Administración de usuarios del sistema (administrador y médico), con control de permisos según el rol asignado.

5. Reportes Médicos

Generación de reportes en formato PDF, organizados por paciente o por fecha.

6. Seguridad y Autenticación

Acceso restringido mediante login con usuario y contraseña. Se utiliza el sistema de autenticación de Laravel

6. Modelo de Base de Datos

El sistema utiliza una base de datos relacional MySQL con las tablas principales:

- usuarios
- pacientes
- antecedentes_obstetricos
- controles
- roles

Cada tabla incluye claves primarias (id), foráneas (paciente_id, rol_id), y relaciones entre entidades para mantener la integridad de los datos.

7. Diagramas UML

Diagrama de clases API.

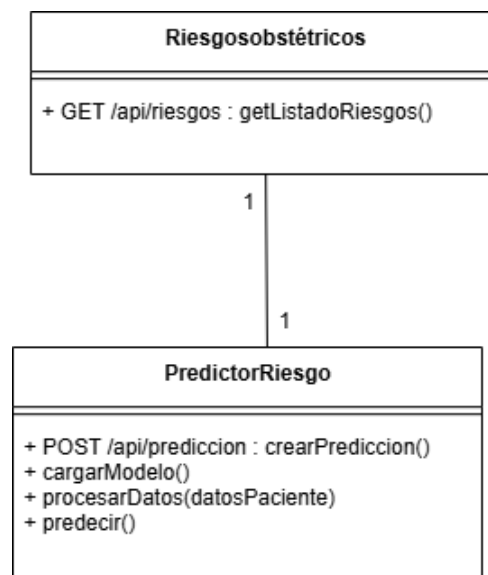


Figura 5 Diagrama de clases API.

Diagrama de clases interfaz gráfica.

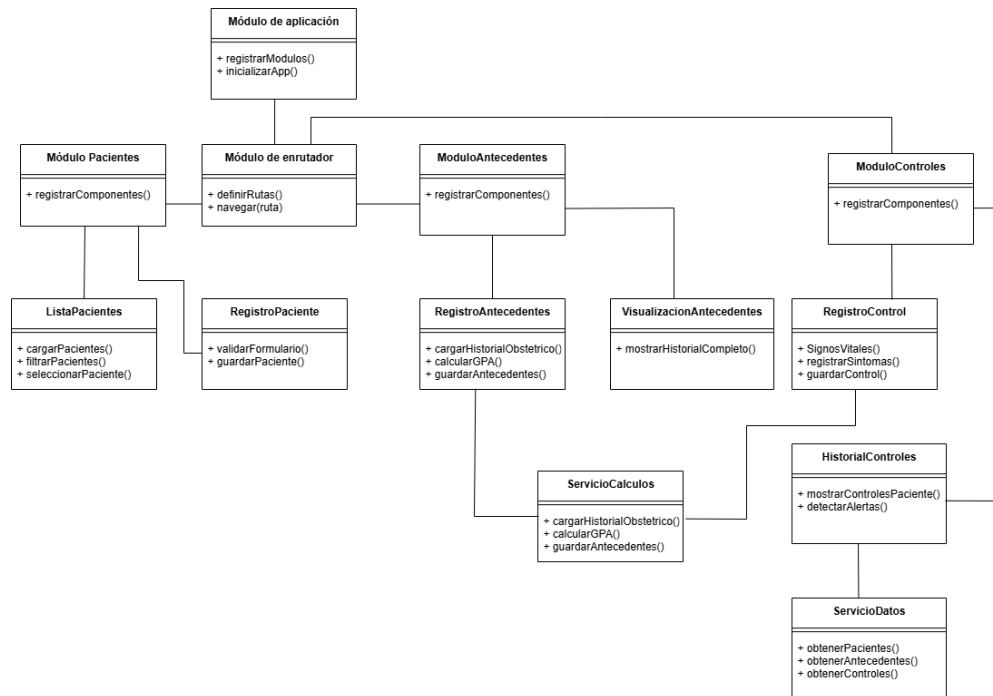


Figura 6 Diagrama de clases interfaz gráfica.

Diagrama de despliegue.

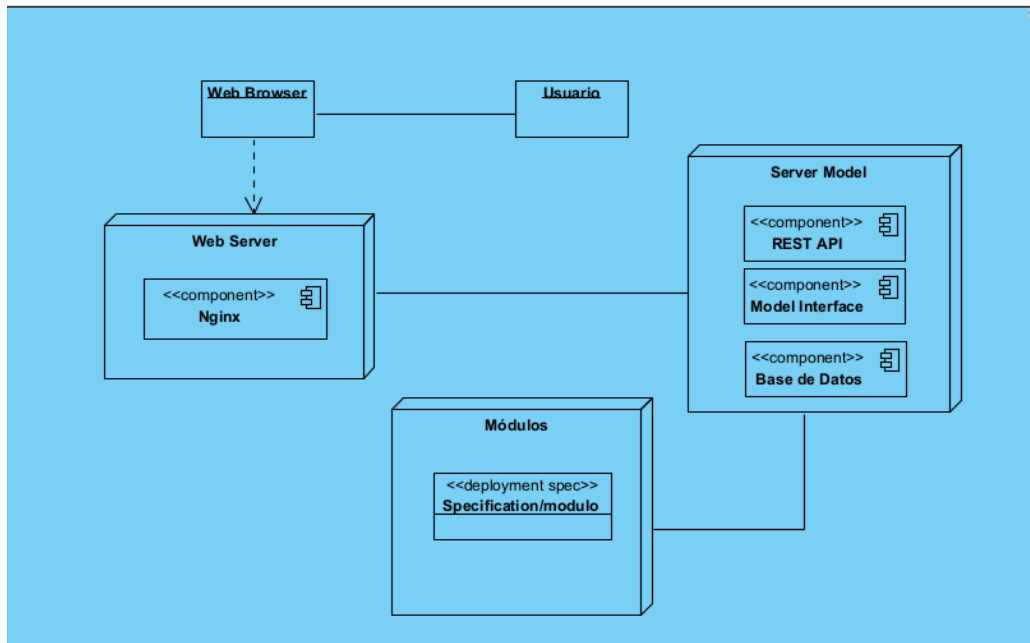


Figura 7 Diagrama de despliegue.

Diagrama de componentes

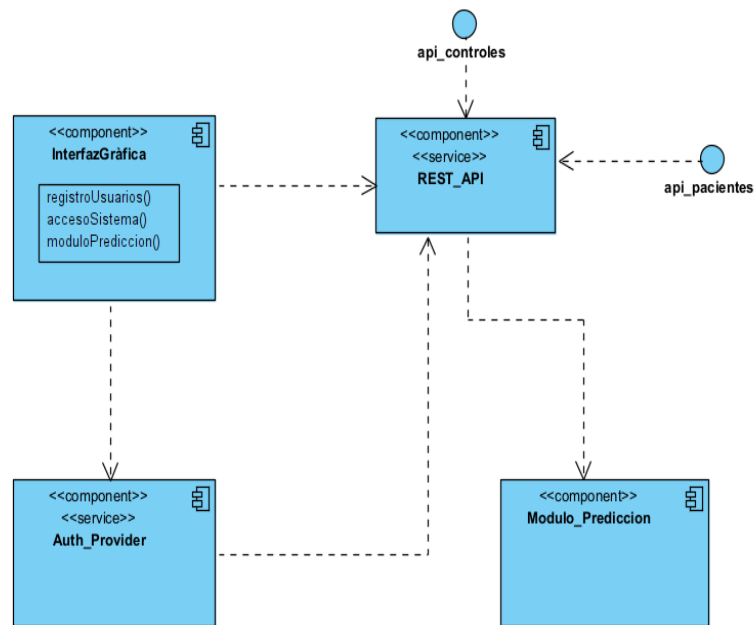


Figura 8 diagrama de componentes.

Diagrama de actividad de Control Prenatal

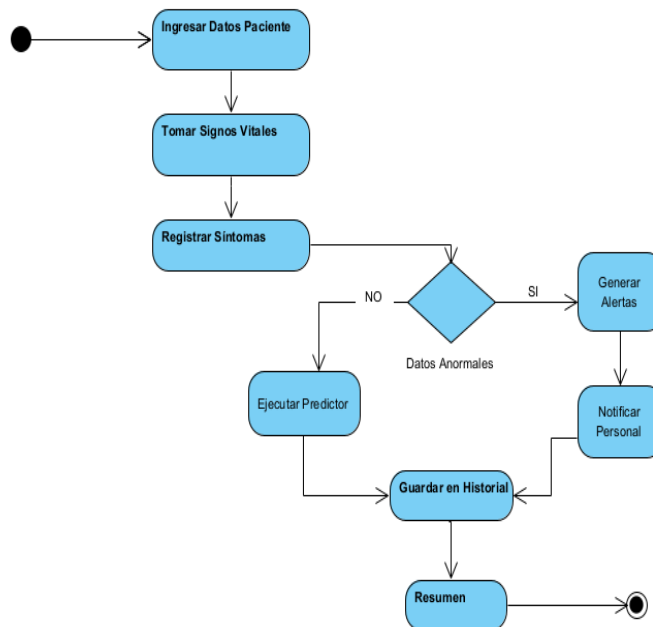


Figura 9 Diagrama de actividad de Control Prenatal.

Diagrama de secuencia de registro de paciente y predicción de riesgos

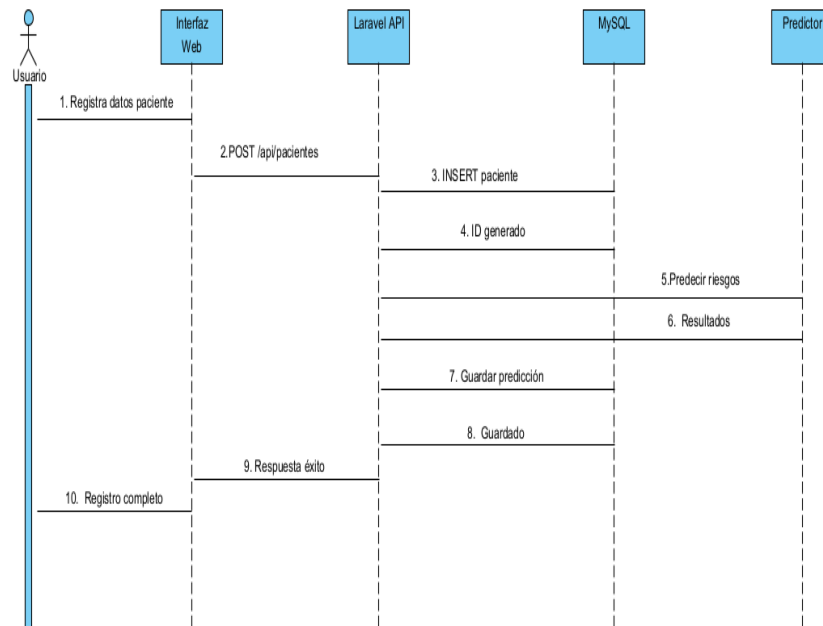


Figura 10 diagrama de secuencia de registro de paciente y predicción de riesgos.

Diagrama de estados acceso al sistema.

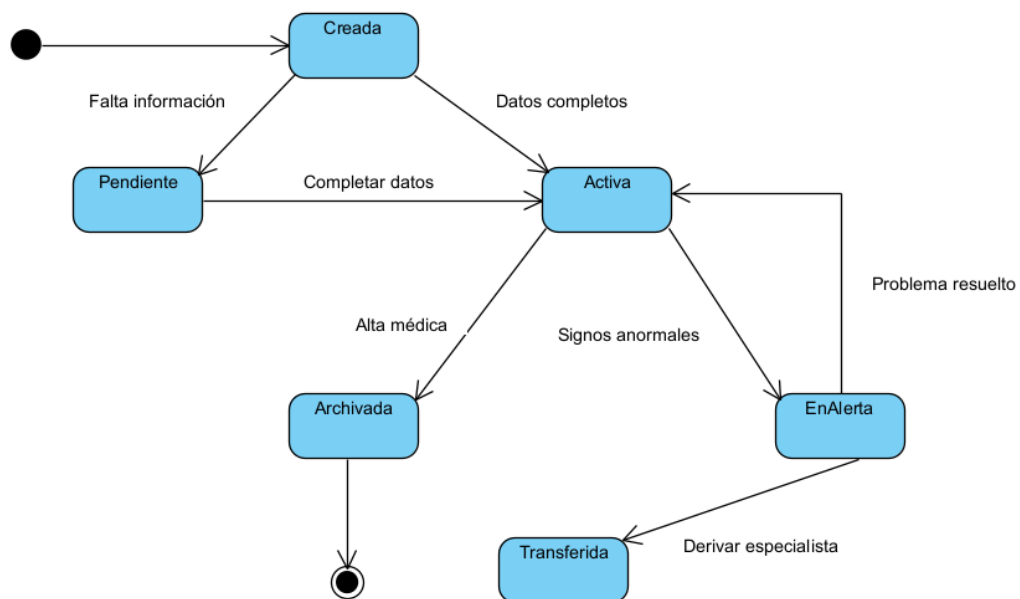


Figura 11 diagrama de estados acceso al sistema.