
APP RECORDATORIO DE MEDICAMENTOS

“Medicapp”

EV2 Arquitectura de la solución, almacenamiento y plan de pruebas.

NOMBRE: Victor Contreras, Rodrigo Lagos, Ignacio Parra, Mauricio Toro

CARRERA: Ingeniería Informática

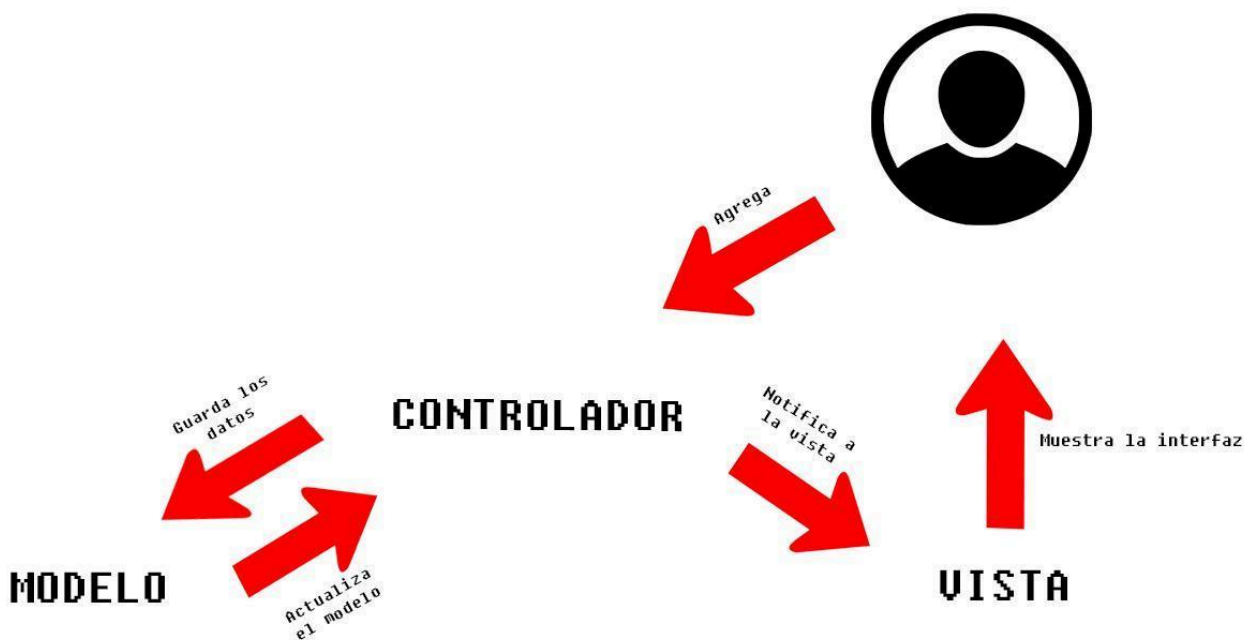
ASIGNATURA: Taller de diseño y desarrollo de soluciones

PROFESOR: Christopher Muñoz Parra

FECHA: --

1- Patrón de arquitectura de la solución:

El patrón que tomaremos en cuenta es el Modelo Vista Controlador (MVC), pensamos en usar el Modelo Vista Presentador (MVP) pero como nuestra aplicación es simple y no requiere de mucha codificación gráfica, suponemos que es más sencilla este patrón.



Estructura del Proyecto en MVC para la Aplicación de Alertas de Medicamentos

El proyecto de la aplicación de alertas de medicamentos se basa en el patrón de arquitectura Modelo-Vista-Controlador (MVC), que es adecuado para mantener una lógica clara y ordenada, separando las responsabilidades entre las capas de datos, control y presentación. A continuación, se describe la estructura propuesta y cómo se adapta a las funcionalidades de la aplicación:

1. ./apk

Descripción: Este archivo es responsable de abrir el instalador en el teléfono, permitiendo la instalación de la aplicación.

Adaptación: Aunque este archivo no forma parte del MVC, se incluirá un instalador que permita a los usuarios instalar la aplicación de manera sencilla.

2. ./main/main.

Descripción: Archivo principal que conecta todas las partes del proyecto. Importa funciones del controlador y vistas para mostrar la interfaz gráfica.

Adaptación: Este archivo será el punto de entrada de la aplicación, gestionando la lógica para agregar y gestionar medicamentos, así como enlazando las vistas donde los usuarios visualizarán sus alertas y horarios de medicación.

3. ./controller/controller.

Descripción: Contiene las funciones solicitadas por el código de la aplicación y ejecuta consultas a la base de datos. También se encarga de las validaciones para evitar errores.

Adaptación: En la aplicación, este archivo será esencial para:

Crear y gestionar alertas y recordatorios.

Manejar la lista de medicamentos.

Validar entradas para evitar duplicados y errores en los horarios.

Permitir que los usuarios marquen las dosis como "tomadas" o "omitidas".

Implementar el botón de emergencia para alertar a un contacto.

4. ./controller/consultas.

Descripción: Contiene las consultas SQL a la base de datos.

Adaptación: Este archivo se utilizará para interactuar con la base de datos MySQL, realizando consultas SQL para almacenar, actualizar y recuperar registros de medicamentos, y otra información necesaria. MySQL utiliza un modelo relacional, lo que permite estructurar los datos en tablas con relaciones claras entre ellos.

5. ./controller/conexión.

Descripción: Abre la conexión con el motor de la base de datos.

Adaptación: Este archivo se encargará de establecer la conexión con la base de datos MySQL, inicializando la conexión y facilitando el acceso a las demás clases que interactúan con la base de datos. Utilizará un controlador de base de datos MySQL, como **JDBC** en Java, para gestionar las conexiones.

6. ./model/

Descripción: Contiene las clases que representan los datos, como la clase usuario.

Adaptación: En la app crearemos varias clases, ejemplo:

- **Medicamento:** Almacena información sobre nombre, dosis, frecuencia, etc.
- **Usuario:** Contiene la información del usuario.
- **Alerta:** Guarda datos sobre cada alarma o recordatorio programado.

Estas clases gestionarán los datos que se enviarán a la base de datos o que serán utilizados por el controlador.

7. ./views/

Descripción: Esta carpeta alberga todas las vistas necesarias para la interacción con el usuario.

Adaptación: Las vistas en la aplicación incluirán:

Recordatorios de medicamentos.

Botones de acción como "Tomar" o "Posponer" dosis.

Notificaciones emergentes para recordatorios.

Botón de emergencia.

Información sobre los medicamentos (descripción, horarios, etc.).

Es fundamental que las vistas sean claras y sencillas, considerando que la app está dirigida principalmente a adultos mayores.

2- Estándares de Programación Segura

Ahora presentaremos los estándares o consideraciones que definimos para el desarrollo de este proyecto, se considerarán los roles y perfiles para el sistema, junto con sus privilegios de cada uno y buenas prácticas que nos llevarán a un desarrollo óptimo de la aplicación.

Perfiles y Roles

La app de recordatorios de medicamentos tendrá dos roles específicos hasta ahora:

| Roles | Perfiles | Privilegios |
|--------------------|-----------------------------|--|
| Editor-programador | Integrante del grupo | Acceso al código fuente Acceso a la base de datos Acceso a al sistema en general |
| Usuario Final | Persona común, Adulto mayor | Vista e interacción con la app |

Buenas Prácticas de Desarrollo

Durante el desarrollo de la aplicación, se seguirán las siguientes buenas prácticas de programación para garantizar seguridad, simplicidad y eficiencia:

1. **Simplicidad en el Diseño:** El sistema evita el uso de procesos complejos, como inicios de sesión o validaciones extensivas de usuario. En su lugar, se centra en la simplicidad ayudando así en la experiencia del usuario ya que está enfocada principalmente en adultos mayores.
2. **Validaciones de Entrada:** Aunque se solicita solo el nombre del usuario, la app incluirá validaciones mínimas como asegurar que el campo no esté vacío o que no se ingresen caracteres inválidos.
3. **Almacenamiento Seguro de Datos:** El nombre del paciente será almacenado de forma segura en el dispositivo, utilizando métodos de almacenamiento local que garantizan que la información no se comparta ni se envíe a servidores externos, minimizando así riesgos de exposición de datos, en cuanto a la información sobre sus medicamentos y el contacto de emergencia se guardan en una base de datos sql.
4. **Requisitos de Seguridad:** Aunque no se maneja información sensible, se asegura que la aplicación no pida más datos de los necesarios y que toda la

información se mantenga en el dispositivo, reduciendo vulnerabilidades potenciales.

3- Algunos servicios de almacenamiento en Cloud

Google firebase

Es una plataforma de desarrollo de aplicaciones móviles y web que ofrece una amplia gama de herramientas y servicios en la nube para facilitar la creación y gestión de aplicaciones. utiliza una base de datos NoSQL en tiempo real que almacena datos en formato JSON y sincroniza los cambios en tiempo real con todos los clientes conectados. Es ideal para aplicaciones que requieren colaboración o actualización de datos instantánea. Nosotros no manejaremos un flujo super instantáneo de datos por lo que no utilizaremos este servicio

Amazon Web Services (AWS)

AWS es una plataforma de servicios en la nube de Amazon que ofrece una amplia gama de servicios como almacenamiento, bases de datos, análisis, aprendizaje automático y más. AWS es conocido por su escalabilidad, fiabilidad y seguridad. Los servicios de almacenamiento más comunes en AWS incluyen **Amazon S3** (almacenamiento de objetos) y **Amazon RDS** (almacenamiento relacional). AWS es ideal para aplicaciones que requieren gran capacidad de almacenamiento, alta disponibilidad y la capacidad de manejar grandes cantidades de tráfico de usuarios, considerando que nuestro proyecto no necesita manejar un gran flujo de datos tampoco utilizaremos este servicio.

Microsoft Azure

Microsoft Azure es la plataforma en la nube de Microsoft que ofrece servicios similares a AWS, incluyendo almacenamiento, bases de datos, análisis y servicios de inteligencia artificial. Azure ofrece **Azure Blob Storage** (almacenamiento de objetos) y **Azure SQL Database** (base de datos relacional) entre otros. Es una opción popular para desarrolladores que ya utilizan productos de Microsoft como Windows Server, Active Directory o SQL Server, integrándose fácilmente con otras herramientas de la suite Microsoft.

Servicio de almacenamiento que utilizaremos

Railway

Railway permite el almacenamiento de bases de datos, similar a AWS y Azure. Railway es una plataforma de desarrollo que facilita la implementación de aplicaciones y servicios en la nube, y proporciona integraciones con diversas bases de datos. Algunas de las bases de datos que puedes utilizar en Railway incluyen PostgreSQL, MySQL (la que utilizaremos) y MongoDB.

A diferencia de AWS y Azure, que ofrecen una amplia gama de servicios en la nube y opciones de bases de datos, Railway se centra más en la simplicidad y la facilidad de uso para desarrolladores, lo que lo hace ideal para proyectos más pequeños, ósea es ideal para nuestro proyecto.



GIT HUB

Para el desarrollo de la aplicación, (aunque no es un servicio de almacenamiento en si) también utilizaremos GitHub para administrar de manera efectiva el control de versiones y facilitar la colaboración simultánea entre los miembros del equipo. Entre las ventajas de GitHub, se destaca el servicio gratuito que permite una búsqueda rápida de repositorios, así como diversas herramientas de colaboración que mejoran la eficiencia del trabajo en equipo, entre otras funcionalidades que facilitan el proyecto.



4- Plan de Pruebas

Criterios de aceptación

| ID | Historia de usuario | Característica/funcionalidad | Criterios de aceptación |
|----|--|---|---|
| 1 | Como usuario Necesito registrar un medicamento con su nombre, dosis, frecuencia y stock, para posteriormente recibir alertas de cuando debo tomarlo. Para asegurarme de seguir correctamente las indicaciones de mi médico. | Permitir al usuario registrar un nuevo medicamento ingresando el nombre, dosis, frecuencia y stock. | <ul style="list-style-type: none"> La aplicación debe permitir ingresar el nombre del medicamento, la dosis, la frecuencia y el stock. Al registrar un medicamento, la aplicación debe crear automáticamente alertas según la frecuencia especificada. La aplicación debe permitir configurar las alarmas para que se envíen a las horas indicadas, con una notificación visible. Las alertas deben funcionar aunque la aplicación no esté en primer plano. |
| 2 | Como usuario Necesito poder ver, editar o eliminar los medicamentos que he registrado. para mantener mis tratamientos de forma actualizada. | Permitir al usuario ver, editar o eliminar medicamentos registrados. | <ul style="list-style-type: none"> La aplicación debe permitir ver una lista de todos los medicamentos registrados. La aplicación debe permitir editar o eliminar los medicamentos registrados. |

| | | | |
|---|--|---|--|
| 3 | <p>Como usuario Quiero ver un calendario con los horarios de mis medicamentos para tener una visión clara de cuándo debo tomar cada dosis y llevar un control de las dosis tomadas, omitidas o pendientes.</p> | <p>Proporcionar un calendario que muestre los horarios de los medicamentos.</p> | <ul style="list-style-type: none"> La aplicación debe permitir visualizar un calendario donde se muestre cada alarma programada para tomar los medicamentos. Cada registro en el calendario debe mostrar el medicamento, la dosis y la frecuencia. El usuario debe poder visualizar la dosis como tomada, omitida o pendiente. El calendario debe actualizarse automáticamente cuando el usuario interactúe con las dosis (por ejemplo, si marca una dosis como tomada). |
| 4 | <p>Como usuario Quiero poder marcar que he pospuesto o he tomado un medicamento Con la finalidad de llevar un registro de cumplimiento y evitar tomar una dosis dos veces.</p> | <p>Permitir al usuario marcar que ha tomado un medicamento.</p> | <ul style="list-style-type: none"> La aplicación debe permitir al usuario marcar cada dosis de medicamento como "tomada" o "pospuesta" con un solo toque o clic al momento de recibir la alerta. |
| 5 | <p>Como usuario Necesito tener un botón de emergencia Para poder notificar rápidamente a un contacto de emergencia si necesito ayuda.</p> | <p>Implementar un botón de emergencia visible en la pantalla principal.</p> | <ul style="list-style-type: none"> La aplicación debe tener un botón de emergencia claramente visible en la pantalla principal. Al presionar el botón de emergencia, la aplicación debe realizar una llamada al contacto de emergencia configurado. |
| 6 | <p>Como usuario Quiero recibir recordatorios para renovar mis recetas médicas antes de que se terminen. Con la finalidad de no quedarme sin medicamentos.</p> | <p>Enviar notificaciones cuando un medicamento esté por acabarse.</p> | <ul style="list-style-type: none"> La aplicación debe emitir una notificación cuando el medicamento esté pronto a acabarse (5 dosis) |
| 7 | <p>Como usuario Necesito tener acceso a la información sobre mis medicamentos, como efectos secundarios y contraindicaciones. Para entender mejor los medicamentos que estoy tomando y estar al tanto de posibles riesgos.</p> | <p>Proporcionar información breve sobre los medicamentos.</p> | <ul style="list-style-type: none"> La aplicación debe proporcionar una sección para cada medicamento registrado donde se muestre información sobre el medicamento, incluidos efectos secundarios y contraindicaciones. |

| | | | |
|---|---|---|---|
| 8 | <p>Como usuario Quiero poder configurar un contacto de emergencia dentro de la aplicación. Con la finalidad de llamarlo rápidamente en caso de necesitar ayuda</p> | <p>Permitir al usuario ingresar y editar un contacto de emergencia.</p> | <ul style="list-style-type: none"> • La aplicación debe permitir al usuario ingresar y guardar la información de un contacto de emergencia. • El usuario debe poder editar y actualizar el contacto de emergencia en cualquier momento. |
| 9 | <p>Como usuario Quiero poder ajustar las preferencias de notificación (sonido, vibración, etc.) Para poder recibir alertas en un formato que sea más fácil de notar y entender para mí.</p> | <p>Permitir la personalización de las notificaciones.</p> | <ul style="list-style-type: none"> • La aplicación debe permitir al usuario personalizar la notificación • Las opciones de notificación deben incluir sonido, vibración, o una combinación de estos. • El usuario debe poder realizar cambios en las preferencias de notificación en cualquier momento y los cambios deben aplicarse inmediatamente. • La aplicación debe proporcionar una opción para probar las diferentes configuraciones de notificación antes de guardarlas. |

A continuación se mostrará los planes de prueba que se le hará y los métodos que aplicaremos, usaremos la caja negra y se plantea sus entrada y salidas

| ID | Nombre | Entrada | Procedimiento | Salida esperada | Salida alternativa |
|----|-------------------------|--|---|--|---|
| 1 | Agregar medicamento | El usuario registrará un medicamento | El usuario abre la aplicación y selecciona "Agregar Alarma". El sistema muestra un formulario con campos para el nombre del medicamento, dosis, y stock. El usuario introduce los datos correspondientes. El usuario establece la hora o frecuencia para recibir alertas. El usuario presiona "Guardar". | El sistema guarda la información del medicamento. Programa las alertas según la frecuencia establecida. | Flujo alternativo 1: Si el usuario deja algún campo vacío o introduce datos no válidos, el sistema muestra un mensaje de error indicando qué debe corregirse. Flujo alternativo 2: Si el medicamento ya existe en el registro, el sistema alerta al usuario sobre el duplicado y le pregunta si desea continuar o cancelar la operación. |
| 2 | Ver Alerta medicamentos | El usuario pulsa sobre uno de sus medicamentos | El usuario puede seleccionar un medicamento para editar o eliminar. El usuario realiza las acciones deseadas (ver, editar o eliminar). | El sistema actualiza y muestra la lista de medicamentos conforme a las acciones del usuario. | Si no hay alertas registradas, aparecerá un mensaje indicando que no existen registros. |
| 3 | Ver calendario | El usuario pulsa la opción "ver calendario". | El usuario selecciona el ícono de un calendario. El sistema muestra un calendario con los horarios de cada alarma programada. Cada registro en el calendario muestra el medicamento, la dosis y el estado (tomada, omitida, pendiente). El usuario cierra la vista del calendario una vez revisada la información. | El calendario se muestra con la información de las dosis, incluyendo el estado de cada una. | Si el usuario no programó ningún medicamento el sistema básicamente no mostrará datos o registros. |

| | | | | | |
|---|--|---|--|--|---|
| 4 | Marcar medicamento como tomado o pospuesto | El usuario pulsa una de las opciones disponibles: "Tomado" o "Posponer 15 min". | <p>El sistema envía una notificación de alerta cuando es hora de tomar el medicamento. El usuario recibe la notificación y ve las opciones "Tomado" y "Posponer 15 min". Si el usuario selecciona "Tomado":</p> <ul style="list-style-type: none"> El sistema actualiza el estado del medicamento a "tomado". Se refleja el cambio en el calendario y en la lista de medicamentos. <p>Si el usuario selecciona "Posponer 15 min":</p> <ul style="list-style-type: none"> El sistema reprograma la alerta para 15 minutos más tarde. | <p>Si se selecciona "Tomado":</p> <ul style="list-style-type: none"> El estado del medicamento se actualiza correctamente <p>Si se selecciona "Posponer 15 min":</p> <ul style="list-style-type: none"> Se muestra un mensaje de confirmación de que la alerta ha sido reprogramada. | El usuario intenta seleccionar una opción que no está disponible. |
| 5 | Botón de emergencia | El usuario podrá llamar automáticamente al contacto ya registrado | El usuario pulsará el ícono de una sirena color rojo. | El sistema llama automáticamente al usuario ya registrado | No llama a nadie porque no se ha agregado un contacto con antelación. El sistema indica que debe configurarse uno antes de usar esta función. |
| 6 | Recordatorio renovación de medicamentos | Cantidad de dosis restantes de los medicamentos. | El sistema monitorea la cantidad de dosis restantes de los medicamentos. Cuando un medicamento llega a 5 dosis o menos, el sistema emite una notificación. | El usuario recibe una notificación recordando la necesidad de renovar la receta del medicamento. | No se emite notificación si la dosis es mayor a 5 |

| | | | | | |
|---|--------------------------------|--|--|--|--|
| 7 | Información sobre medicamentos | Selección de una alerta medicamento registrado. | El usuario abre la aplicación y selecciona un medicamento registrado. El usuario selecciona la opción de información. El sistema muestra una sección con información breve sobre el medicamento. | Se presenta información breve sobre indicaciones de uso, efectos secundarios o contraindicaciones del medicamento. | Mensaje de error si el medicamento seleccionado no está registrado. |
| 8 | Agregar contacto de emergencia | el usuario selecciona contacto emergencia | El usuario abre la aplicación y selecciona el ícono de contacto. El sistema muestra campos para ingresar los datos del contacto de emergencia. El usuario introduce el nombre y número telefónico del contacto. El usuario presiona "Guardar". | El contacto se guarda correctamente y está disponible para futuras llamadas de emergencia. | El usuario digital mal el número telefónico. El usuario no agrega ninguna información. |
| 9 | Preferencias de notificación | El usuario selecciona la opción de personalización de notificaciones | El usuario abre la configuración de notificaciones en la aplicación. El sistema muestra opciones para personalizar notificaciones (sonido, vibración). El usuario selecciona sus preferencias. El usuario presiona "Guardar" para aplicar las nuevas preferencias. | Las preferencias se guardan y aplican inmediatamente en la aplicación. | Mensaje de error si la configuración no se puede aplicar debido a un problema técnico. |

