



UNIVERSIDAD  
TECNOLÓGICA  
METROPOLITANA  
FACULTAD DE INGENIERÍA  
Escuela de Informática

# Computación Paralela y Distribuida

Proyecto Aplicación Móvil

Integrantes:

- Javier López
- Alex Bidart
- Shu-yi Wong

Profesor: Sebastián Salazar

Fecha: 23 de Diciembre de 2020



UNIVERSIDAD  
TECNOLÓGICA  
METROPOLITANA  
FACULTAD DE INGENIERÍA  
Escuela de Informática

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Problemática</b>	<b>3</b>
<b>3. Resolución de la Problemática</b>	<b>4</b>
<b>4. Tecnología usada</b>	<b>9</b>
<b>5. Conclusión</b>	<b>11</b>

# 1. Introducción

Hoy en día, el uso de plataformas digitales es el pan de cada día, en especial las plataformas como aplicaciones móviles, siendo las más solicitadas, llevando esta demanda a que la mayoría de plataformas web tengan un acceso mediante aplicaciones móviles, es por ello que este informe tiene como objetivo explicar en mejor medida el desarrollo de una aplicación móvil que entregue los resultados de una plataforma web, como es una Api Rest, de manera más clara y atrayente para el usuario.

Se muestra las diferentes herramientas necesarias para la creación de dicho programa, las explicaciones de cada una de sus funcionalidades, además de la muestra de las pantallas realizadas con el fin de entregar un resultado más apropiado y finalmente una conclusión al respecto del trabajo realizado.

## 2. Problemática

Con el fin de realizar un trabajo exitoso, se requiere cumplir ciertos requisitos que se mencionan a continuación:

- Creación de dos vistas, la primera corresponde a la de un formulario que ingrese los siguientes datos para realizar una operación:
  - Indicador (uno de los realizados).
  - País uno. Selección de un país.
  - País dos. Selección de un país (Distinto al anterior).
  - Año de inicio. Debe ser un año válido (dentro del rango posible).
  - Año de término. Debe ser un año válido posterior al año de inicio.
  - Botón con la etiqueta de “graficar”. Al ejecutar este botón se debe renderizar un gráfico de líneas con la comparación año a año (entre los intervalos seleccionados) de ambos países (en el gráfico, estos deben diferenciarse por colores).
- La segunda es una vista con los nombres de los integrantes del trabajo, el cual queda a libre elección su diseño.
- También se solicita un “Splash Screen” que aparezca al iniciar la aplicación con la finalidad de cargar los datos correspondientes de manera efectiva.

Cabe destacar que es necesario que la aplicación móvil pueda ser conectada a la red también a la Api Rest referenciada con el fin de obtener los datos solicitados de manera clara y concisa.

### 3. Resolución de la Problemática

Antes de la programación, fue necesario llevar a cabo una larga investigación y documentación para obtener un resultado exitoso, dentro de esta investigación, se obtuvo que para finalizar la aplicación correctamente es necesario tener los siguientes paquetes de datos.

- `http`<sup>1</sup>. Este paquete corresponde al lenguaje Dart permite que la aplicación se conecte a algún repositorio `http` y pueda hacer solicitudes, como por ejemplo, conectarse a la `api rest` y solicitar o enviar los datos.
- `charts_flutter`<sup>2</sup>. Este paquete es de Flutter y permite la creación y visualización de gráficos de diferentes tipos.
- `validators`<sup>3</sup>. Este paquete de Dart corresponde a un validador de datos `string`, la cual es bastante útil y efectiva para que los datos sean más correctos.

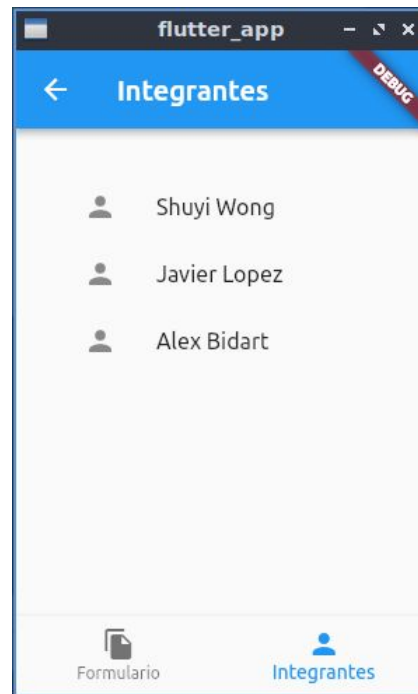
En primera instancia se procedió a la creación de las diferentes vistas solicitadas, iniciando por la de integrantes encontrada en `./screens/Integrantes.dart` la cual consta con una clase llamada `UserListApp` con un contenido básico de estructura que en su cuerpo llama a la clase `"UserList()"` la cual tiene una lista de usuarios de tipo `"User"` con los nombres de los tres integrantes que conforman el grupo de trabajo. Esta lista es mostrada en pantalla mediante un widget de muestra de lista como se puede observar en la siguiente imagen.

---

<sup>1</sup> "http | Dart Package - Pub.dev." 16 jul. 2020, <https://pub.dev/packages/http>. Se consultó el 23 dic. 2020.

<sup>2</sup> "charts\_flutter | Flutter Package - Pub.dev." 28 feb. 2020, [https://pub.dev/packages/charts\\_flutter](https://pub.dev/packages/charts_flutter). Se consultó el 23 dic. 2020.

<sup>3</sup> "validators | Dart Package - Pub.dev." 26 jun. 2020, <https://pub.dev/packages/validators>. Se consultó el 23 dic. 2020.



*Imagen 3.1: "Vista Integrantes" (Fuente: Creación Propia)*

Posteriormente se dio a la creación de la pantalla formulario la cual tiene una estructura en base a un widget llamado Form para la creación de formularios, este contiene dos tipos de widgets, cuatro espacios de TextFormField y uno de tipo DropdownButton, con el fin de ingresar correctamente los datos solicitados.

- En primer lugar se creó el espacio para ingresar los códigos de los países de forma texto dentro de un mismo container.
- Luego se creó una lista que contiene cada uno de los posibles Indicadores solicitados, estos se muestran mediante una barra de menú que se expande hacia abajo al cliquear, donde uno debe elegir el indicador que desea buscar.
- Posterior a eso, se creó otro espacio idéntica a de los países para ingresar las fechas anuales.
- Finalmente, el formulario posee un botón llamado Graficar el cual termina de ingresar los datos y los lleva a un cálculo interno donde debe ingresar a la red, conectarse a la api rest y posteriormente entregar una nueva pestaña con un gráfico de barras con los resultados solicitados.



Imagen 3.2: “Vista Formulario” (Fuente: Creación Propia)

En cuanto a la validación de la pantalla, cada uno de los espacios tiene una validación en base a presuntos condicionales, donde los datos ingresados por texto no pueden estar vacíos, en caso de esto la aplicación muestra los siguientes errores



Imagen 3.3: “Vista Formulario, errores sin datos” (Fuente: Creación Propia)

Además, otra validación característica es que los valores ingresados de países y años no pueden ser los mismos, en caso de que el usuario ingrese los mismos valores, la pantalla muestra los errores visualizados en la siguiente imagen.



The screenshot shows a mobile application window titled 'flutter\_app' with a 'Formulario' screen. The screen has a white background and a blue header. A red 'DEBUG' banner is in the top right corner. The form contains the following elements:

- Países:** Two input fields, both containing 'CL'. Below the second field is a red error message: 'Pais incorrecto'.
- Indicador:** A dropdown menu with 'TDA' selected.
- Fechas:** Two input fields, both containing '2000'. Below the second field is a red error message: 'Fecha Incorrecta'.
- Buttons:** A purple 'Graficar' button is centered below the date fields.
- Bottom Bar:** Two icons are visible: 'Formulario' (a blue document icon) and 'Integrantes' (a grey person icon).

Imagen 3.4: “Vista Formulario, errores datos identicos” (Fuente: Creación Propia)

Es necesario aclarar, que los datos ingresados en el formulario son guardados en una clase llamada “Model” con el fin de tener los datos de manera más ordenada para luego utilizarlos en el llamado json del servicio POST de la api rest.

Siguiendo a la definición de las vistas, se creo el Splash Screen referenciado en el código base de la aplicación, de forma que este sea presentado correctamente al iniciar la aplicación, este contempla el logo de la universidad y tiene una duración de 3 segundos.





Imagen 3.5: “Splash Screen” (Fuente: Creación Propia)

Para un manejo más cómodo de la aplicación, se dio a la creación de un BottomNavigatorBar, o también dicho una barra de navegación inferior, para poder pasar de vistas de manera cómoda y rápida.

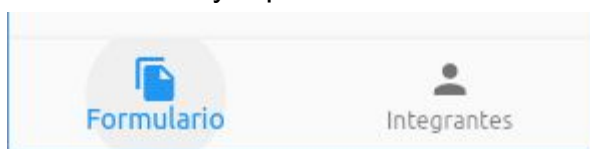


Imagen 3.5: “Barra de Navegación” (Fuente: Creación Propia)

## 4. Tecnología usada

Para todo el desarrollo de la aplicación se utilizó el entorno Android Studio



*Imagen 4.1: “Logo Android Studio” (Fuente: google.cl)*

Android Studio es un entorno de desarrollo integrado para plataformas Android, el cual cuenta con diferentes herramientas y librerías que ayudan al desarrollador a crear de manera más fácil y eficiente aplicaciones móviles de carácter android, usa lenguajes de programación como Java, Kotlin y C++, no obstante se pueden descargar plug ins para la programación en otros lenguajes.

Se utilizó Flutter para el desarrollo de toda la aplicación y se usó el lenguaje Dart para ella.



*Imagen 4.1: “Logo Flutter y Dart” (Fuente: google.cl)*

Flutter es un SDK de desarrollo de aplicaciones móviles, se utiliza mayoritariamente para el desarrollo de interfaces de usuario en Android, iOS y Web, utiliza lenguajes como C, C++ y Dart.

Dart es un lenguaje de programación de código abierto, es utilizado para la programación Web en su mayor medida gracias a su estructura flexible

## 5. Conclusión

Como conclusión, el trabajo conllevó diferentes problemáticas a los programadores gracias al nuevo entorno y lenguaje de programación que se utilizó, siendo una grata experiencia de aprendizaje, emocionante y bastante interesante, puesto que este tipo de programación es una de las más solicitadas dentro del mercado hoy en día, si bien, en el ámbito laboral puede que dependiendo del puesto que se obtenga no sea utilizado este conocimiento, no quita el peso de que corresponde a algo sumamente favorable al desarrollo del perfil del programador.

Actualmente la aplicación está acotada en los datos ya existentes en la API REST y en caso de que exista alguna modificación en aquella API o su base de datos, es necesario redefinir algunas validaciones en la aplicación móvil frente a la modificación. También los accesos y nuevas funcionalidades que se deseen aportar es necesario realizar un nuevo estudio y modificaciones correspondientes.

No obstante, si la aplicación llega a manos de expertos más capacitados, es posible la creación de un sistema que ayude de manera eficiente entidades que necesiten un manejo de datos como este, así como en el caso que se tiene ahora en entidades respecto a la migración, también para personas naturales que lo soliciten, esta aplicación sería bastante útil y posiblemente solicitada para la toma de decisiones correctas.