

UNIVERSIDAD NACIONAL DE MOQUEGUA

INGENIERÍA DE SISTEMAS E INFORMÁTICA



**TRABAJO DE UNIDAD 2 - "FINANZAS SIEMPRE A LA
MANO"**

Presentado por:

Jefferson Aldair Camargo Mamani - 2021204075

Asignatura:

Programacion de dispositivos moviles I

Profesor:

Juan Carlos Clares Perca

Ilo 30 de diciembre de 2024

Índice

1. Introducción	3
2. Objetivos	4
3. Justificación	5
4. Procedimiento de desarrollo	6
4.1. Login	6
4.1.1. interfaz del login	7
4.1.2. Logica del login	8
4.2. Registrarse	8
4.2.1. Interfaz para Registrarse	9
4.2.2. Logica para Registrarse	10
4.2.3. Base de datos firebase	10
4.3. Menu principal	10
4.3.1. Interfaz del menu principal	11
4.3.2. Logica del menu principal	12
4.4. Sección de insertar - actualizar - borrar	12
4.4.1. Interfaz del Crud	13
4.4.2. Logica del CRUD	14
4.4.3. Formulario	14
4.4.4. Almacenamiento de datos	15
4.4.5. Algoritmo para gestionar los datos	15
4.4.6. Creacion del activador	16
4.5. Sección de consultar	16
4.5.1. Interfaz de la seccion de consultas	17
4.5.2. Logica de la seccion de consultas	18
4.5.3. Publicacion del Exel	19
4.6. Sección de graficos	19
4.6.1. Interfaz de las seccion de graficos	20
4.6.2. Logica de la seccion de graficos	21
4.6.3.Codigo para los graficos	21
5. Ejecución de la aplicación	23

Índice de figuras

1.	Captura de la interfaz del login.	7
2.	Captura de los bloques del login.	8
3.	Captura de la interfaz para registrarse.	9
4.	Captura de los bloques para registrarte.	10
5.	Captura de la base de datos en Firebase.	10
6.	Captura de la interfaz del menu principal.	11
7.	Captura de los bloques del menu principal.	12
8.	Captura de la interfaz del CRUD.	13
9.	Captura de los bloques del CRUD.	14
10.	Captura del formulario de donde se recolecta los datos.	14
11.	Captura del exel utilizado para visualizar los datos.	15
12.	Captura del Apps Script.	15
13.	Captura del activador	16
14.	Captura de la interfaz de la seccion de consultas	17
15.	Captura de los bloques de la seccion de consultas	18
16.	Captura de la publicacion del exel	19
17.	Captura de la interfaz de la seccion graficos	20
18.	Captura de los bloques de seccion de graficos	21
19.	Captura de los archivos .html en el appInventor	21
20.	Captura de los archivos .html	22
21.	Prueba del uso del login	23
22.	Prueba del uso para crear nuevos usuarios	24
23.	Prueba de la creacion de un nuevo dato	24
24.	Prueba de su visualizacion	25
25.	Prueba de los graficos	25

1. Introducción

En un mundo donde la gestión financiera personal es clave para alcanzar metas económicas y mantener estabilidad, contar con herramientas prácticas y accesibles es fundamental. Finanzas Siempre a la Mano surge como una solución móvil diseñada para simplificar el control de los ingresos y egresos, brindando a los usuarios la capacidad de registrar, analizar y planificar sus finanzas de manera intuitiva y eficiente.

La aplicación ofrece una interfaz amigable y segura, optimizada para dispositivos móviles, que permite llevar un registro detallado de transacciones financieras y acceder a gráficos interactivos para visualizar tendencias de gastos e ingresos. Además, incluye opciones avanzadas de personalización y análisis, empoderando a los usuarios para tomar decisiones informadas sobre su situación económica.

Finanzas Siempre a la Mano no solo se enfoca en la simplicidad de uso, sino también en la seguridad y la precisión, convirtiéndose en el aliado perfecto para quienes buscan mantener un control constante y eficiente de sus finanzas desde la comodidad de su dispositivo móvil.

2. Objetivos

Objetivo General

Desarrollar una aplicación móvil que permita a los usuarios gestionar sus finanzas personales de manera sencilla, segura y eficiente, facilitando el registro, actualización, eliminación y análisis de transacciones financieras para tomar decisiones económicas informadas.

Objetivos Específicos

- **Facilitar la administración financiera:** Diseñar una interfaz intuitiva para el registro rápido de ingresos y egresos, con campos clave como fecha, concepto e importe.
- **Proveer herramientas de análisis financiero:** Implementar cuadros estadísticos y gráficos interactivos que muestren patrones de ingresos y egresos, comparaciones por periodos y distribución por categorías.
- **Optimizar la accesibilidad:** Crear una plataforma móvil que permita acceder a la información financiera desde cualquier lugar en tiempo real.
- **Promover la sincronización de datos:** Implementar almacenamiento en la nube para asegurar la continuidad y disponibilidad de la información en múltiples dispositivos.
- **Mejorar la toma de decisiones financieras:** Generar reportes automáticos que resuman el estado financiero del usuario y sugieran áreas de mejora basadas en los patrones observados.

3. Justificación

En un mundo donde las finanzas personales juegan un papel crucial en la estabilidad económica de los individuos, es fundamental contar con herramientas tecnológicas que permitan gestionar ingresos y egresos de manera efectiva. Sin embargo, muchas de las soluciones existentes en el mercado presentan limitaciones como interfaces poco intuitivas, falta de accesibilidad móvil, y escasas opciones de análisis financiero que ayuden a los usuarios a comprender sus patrones de gasto e ingresos.

La aplicación ***Finanzas Siempre a la Mano*** se justifica en la necesidad de ofrecer una solución moderna, segura y accesible que facilite el control financiero personal desde dispositivos móviles. Este sistema no solo permite registrar y gestionar transacciones de manera eficiente, sino que también proporciona herramientas de análisis visual mediante cuadros estadísticos y gráficos interactivos que ayudan a los usuarios a tomar decisiones informadas sobre su economía.

Adicionalmente, la seguridad de los datos financieros es una preocupación importante. Por ello, esta aplicación implementará un sistema robusto de autenticación y gestión de contraseñas para proteger la información sensible de los usuarios. La inclusión de sincronización en la nube garantiza la disponibilidad de los datos en cualquier momento y desde cualquier dispositivo, haciendo que la herramienta sea versátil y adaptable a las necesidades modernas.

En resumen, ***Finanzas Siempre a la Mano*** responde a la creciente demanda de herramientas financieras personales que combinen accesibilidad, seguridad y análisis avanzado. Su implementación busca empoderar a los usuarios para gestionar sus finanzas de forma eficiente y lograr una mejor estabilidad económica.

4. Procedimiento de desarrollo

4.1. Login

En la sección de Inicio de Sesión (Login), el sistema solicita al usuario ingresar su nombre de usuario y contraseña como datos de acceso. Estos datos se almacenan de forma segura en una base de datos en la nube proporcionada por Firebase, garantizando la integridad y confidencialidad de la información. La interfaz cuenta con dos botones principales:

1. **Ingresar::** Permite al usuario acceder al sistema tras validar sus credenciales contra los registros almacenados en la base de datos.
2. **Registrarse::** Ofrece la opción de crear una nueva cuenta, donde el usuario puede ingresar un nombre de usuario único y una contraseña, los cuales se guardan directamente en la base de datos de Firebase.

4.1.1. interfaz del login

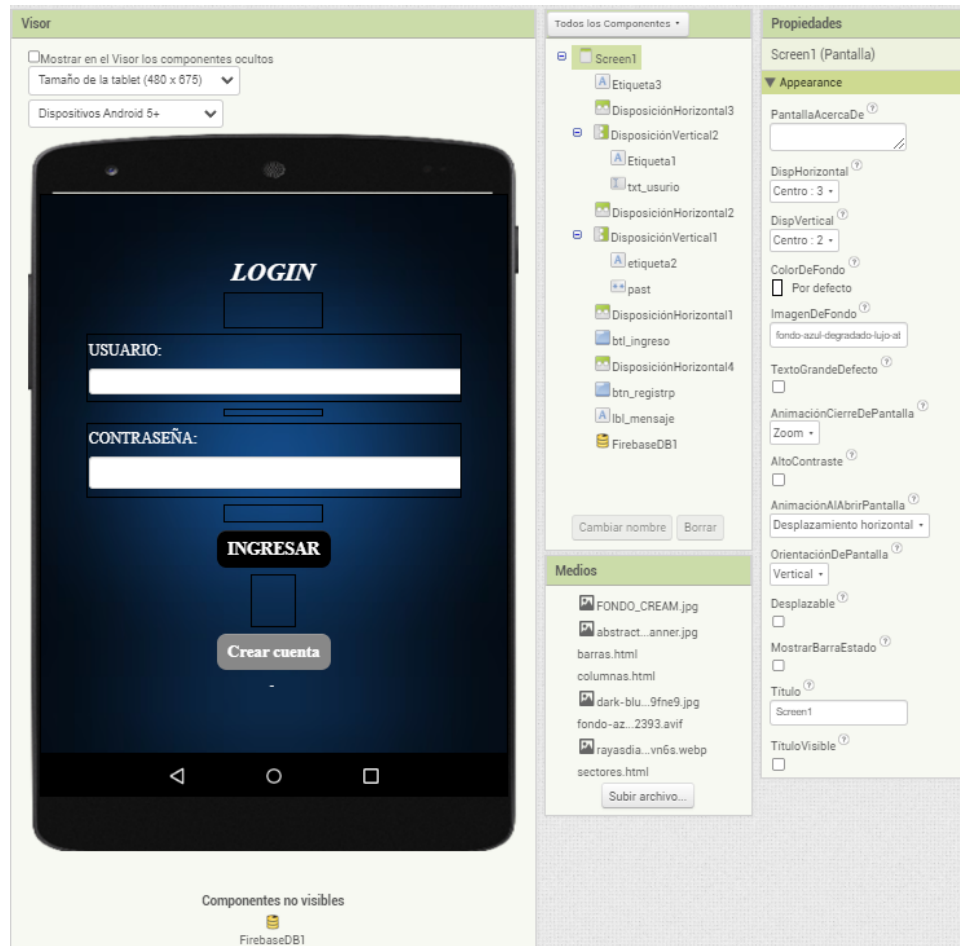


Figura 1: Captura de la interfaz del login.

4.1.2. Logica del login

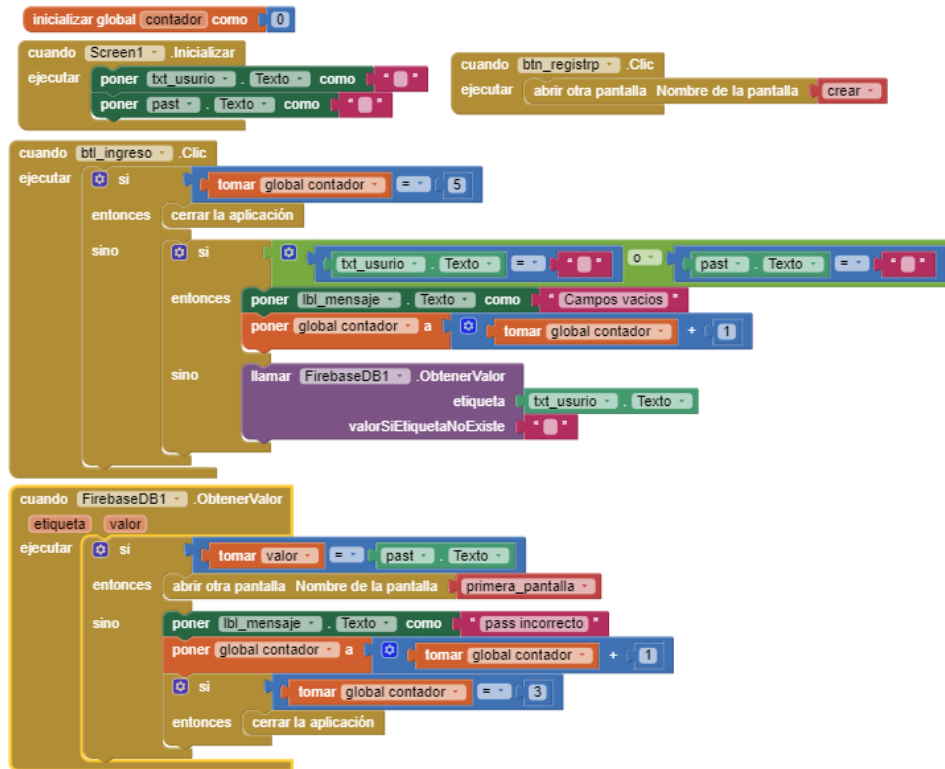


Figura 2: Captura de los bloques del login.

4.2. Registrarse

En la funcionalidad de registro de usuarios, el sistema permite a las personas crear una cuenta nueva para acceder a la aplicación. Este proceso incluye la introducción de un nombre de usuario único, una contraseña, y la confirmación de la contraseña para garantizar que el ingreso sea correcto y evitar errores tipográficos.

Al completar los campos requeridos y proceder con el registro, el sistema valida los datos ingresados y los almacena de manera segura en la base de datos de Firebase, una plataforma en la nube que asegura la integridad y disponibilidad de la información. Si el proceso de registro es exitoso, la aplicación muestra un mensaje de confirmación que notifica al usuario que la cuenta se ha creado correctamente. Este flujo está diseñado para ser intuitivo y seguro, proporcionando una experiencia de usuario confiable y eficiente en el manejo de credenciales.

4.2.1. Interfaz para Registrarse

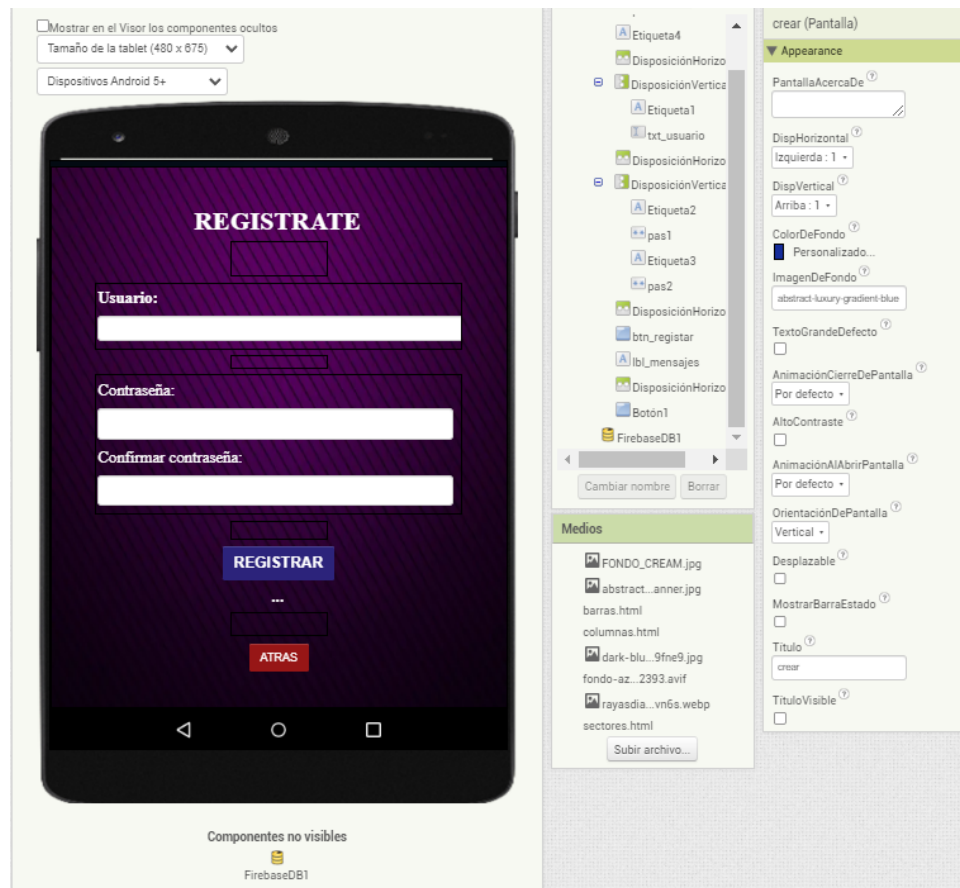


Figura 3: Captura de la interfaz para registrarse.

4.2.2. Logica para Registrarse

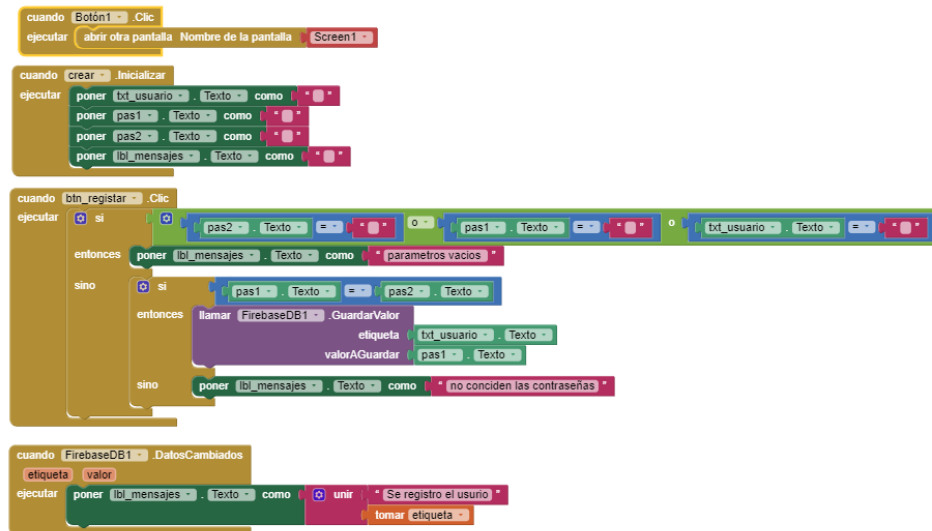


Figura 4: Captura de los bloques para registrarte.

4.2.3. Base de datos firebase

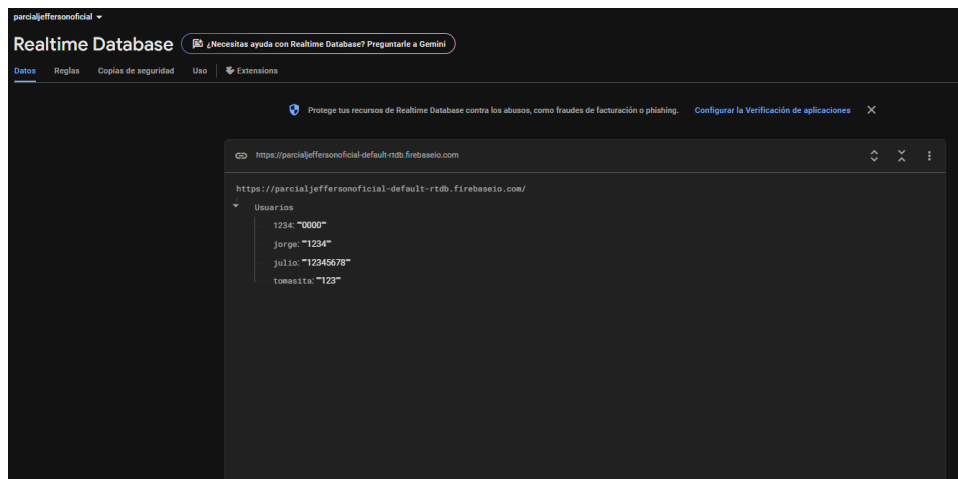


Figura 5: Captura de la base de datos en Firebase.

4.3. Menu principal

El menú principal de la aplicación actúa como el punto central de navegación, permitiendo a los usuarios acceder a las principales funcionalidades del sistema de manera clara y organizada. Este menú presenta tres módulos principales representados por botones interactivos:

1. **Actualizar:** Redirige al usuario a una interfaz dedicada donde puede modificar o eliminar registros financieros existentes, proporcionando opciones intuitivas para mantener actualizada su información.
2. **Consultar:** Permite acceder a una vista detallada de los registros almacenados, organizada de forma que sea fácil de interpretar, con opciones para filtrar y buscar datos específicos según las necesidades del usuario.
3. **Gráficos:** Dirige a una interfaz donde se presentan análisis visuales, como gráficos estadísticos e informes, que ayudan al usuario a comprender mejor sus patrones financieros y tomar decisiones informadas.

4.3.1. Interfaz del menu principal

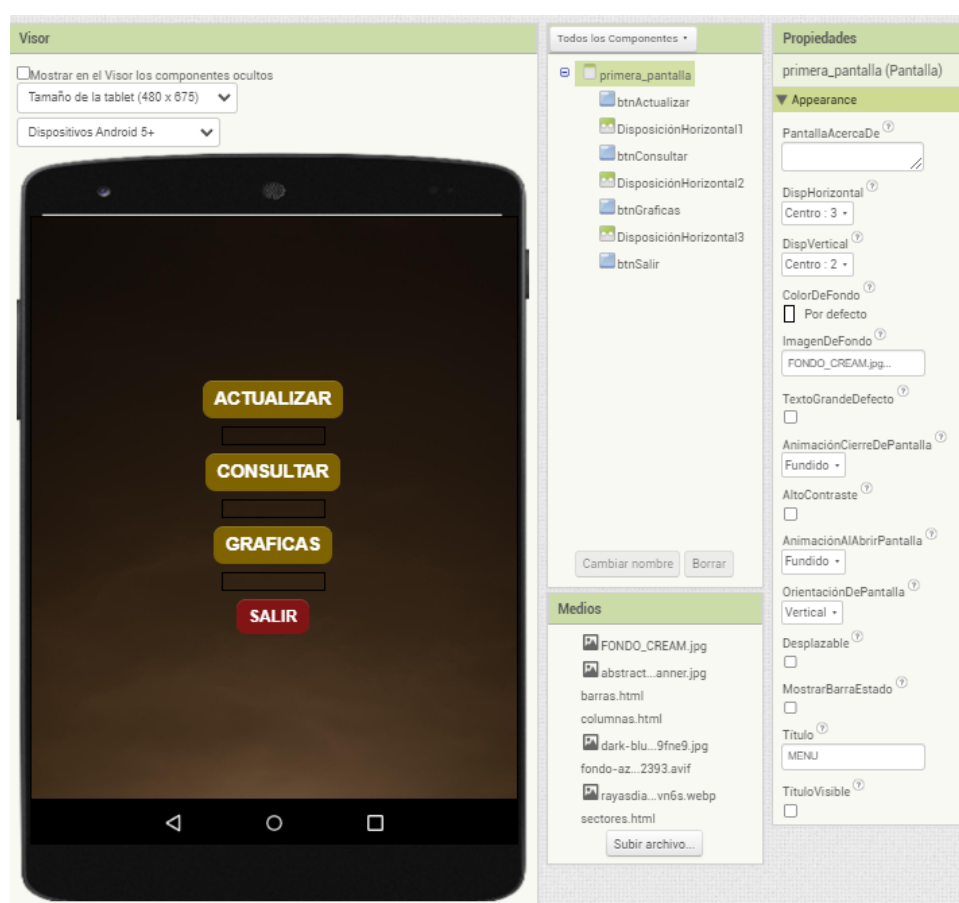


Figura 6: Captura de la interfaz del menu principal.

4.3.2. Logica del menu principal



Figura 7: Captura de los bloques del menu principal.

4.4. Sección de insertar - actualizar - borrar

El sistema utiliza un formulario de Google para registrar, actualizar y eliminar datos, integrándose directamente con una hoja de cálculo en **Google Sheets** que actúa como base de datos. A continuación, se describe el proceso y las funcionalidades principales:

1. **Registro de Datos:** Los datos ingresados en la aplicación (desarrollada en App Inventor) se envían al formulario de Google. El enlace de envío del formulario se copia y se adapta, dividiendo la URL en dos partes: una como dirección base y otra para identificar los campos. Esto permite enviar datos de manera similar a como se haría en sistemas como phpMyAdmin.
2. **Almacenamiento:** La información registrada se guarda automáticamente en una hoja de cálculo creada por el formulario o seleccionada previamente. Se agrega una columna adicional llamada *Acción*, que registra el tipo de operación realizada:
 - **POST:** Para registrar nuevos datos.
 - **DELETE:** Para eliminar datos existentes.
 - **UPDATE:** Para actualizar registros existentes.

3. **Gestión de Acciones:** Se utiliza **Apps Script**, una herramienta de Google para automatizar procesos, que permite gestionar las operaciones directamente desde la hoja de cálculo. Este script se activa cuando el formulario es utilizado, ejecutando el código necesario para actualizar o eliminar datos, según corresponda.
4. **Resultados:** Este enfoque permite implementar un sistema eficiente para registrar, actualizar y eliminar información, optimizando la interacción entre la aplicación y la hoja de cálculo. La automatización mediante Apps Script asegura que las operaciones sean precisas y rápidas.

4.4.1. Interfaz del Crud

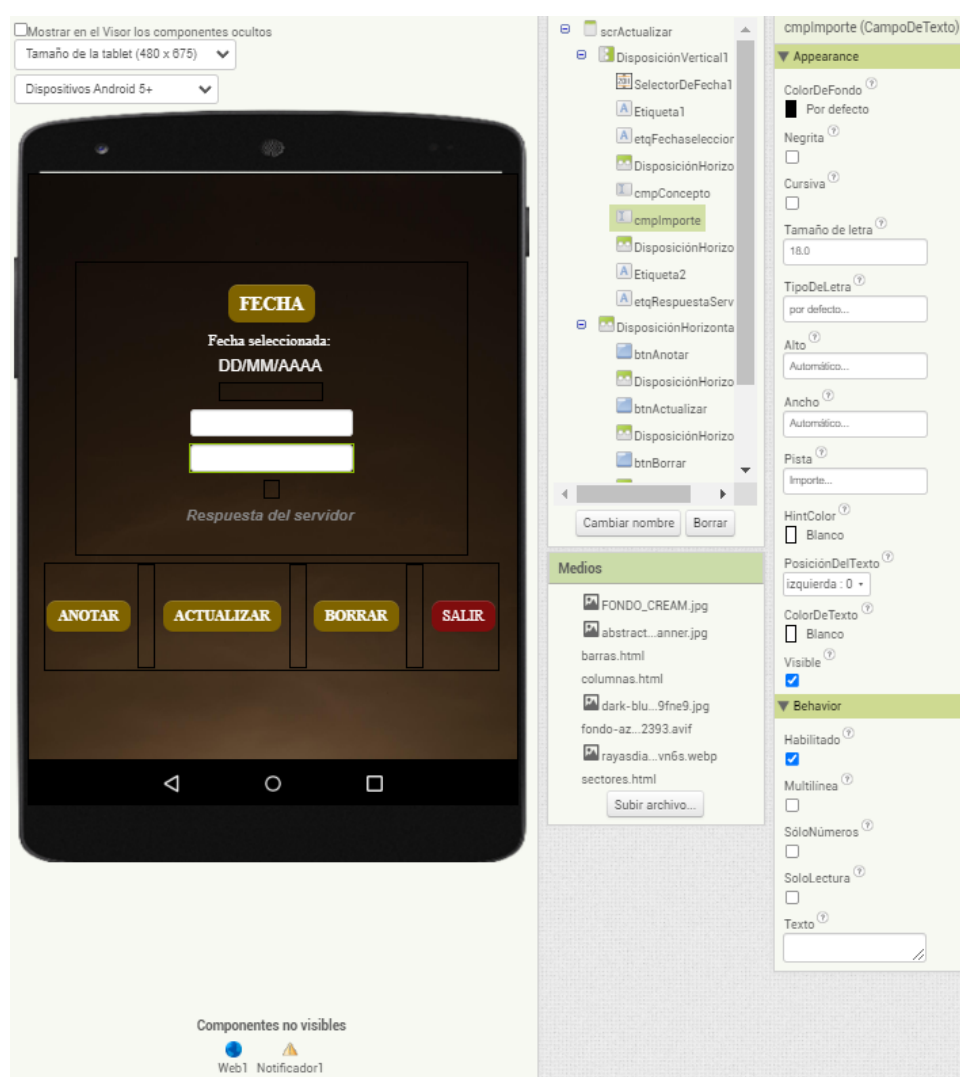


Figura 8: Captura de la interfaz del CRUD.

4.4.2. Logica del CRUD

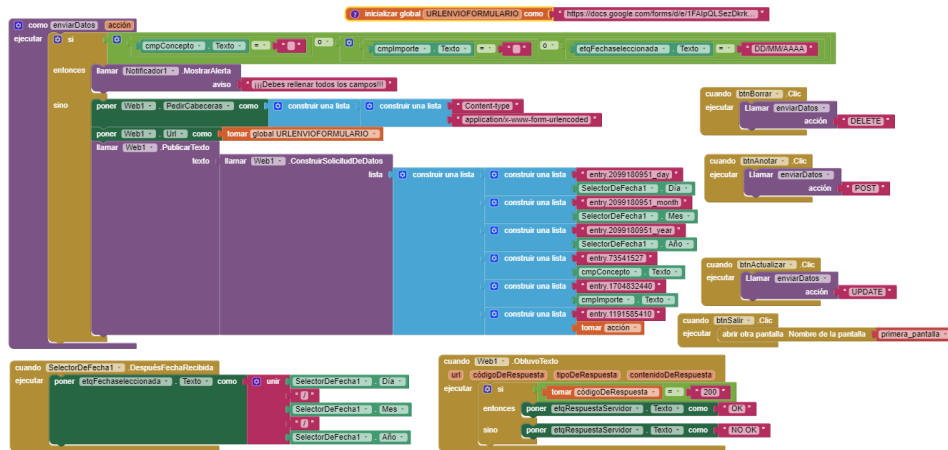


Figura 9: Captura de los bloques del CRUD.

4.4.3. Formulario

Figura 10: Captura del formulario de donde se recolecta los datos.

4.4.4. Almacenamiento de datos

	A	B	C	D	E
	Form_Responses2				
	Marca temporal	Fecha	Concepto	Importe	Acción
2	18/12/2024 18:15:50	13/12/2024	Ocio	20	
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					

Figura 11: Captura del excel utilizado para visualizar los datos.

4.4.5. Algoritmo para gestionar los datos

```

1 // Autor: José Luis Núñez para el Curso de App Inventor 2
2 // Basado en el código de puravidapps.com
3 // Partimos de que tenemos columna 0 = "Marca temporal", columna 1 = "Fecha", columna 2="Concepto", columna 3="Importe"
4
5 function action(e) {
6   var fecha = e.values[1]; // Fecha
7   var concepto = e.values[2]; // Concepto
8   var action = e.values[4]; // Acción
9   var sheet = SpreadsheetApp.getActiveSheet();
10  var values = sheet.getDataRange().getValues();
11  var dia = fecha.split("/") [0];
12  var mes = fecha.split("/") [1];
13  var anyo = fecha.split("/") [2];
14  var fechaFormatoLargo = new Date(anyo + "/" + mes + "/" + dia);
15
16  if (action == "DELETE") {
17    del(fechaFormatoLargo, concepto, 2, 1, sheet, values);
18  }
19  if (action == "UPDATE") {
20    upd(fechaFormatoLargo, concepto, 2, 1, sheet, values);
21  }
22 }
23
24 function del(fechaFormatoLargo, concepto, columnaConcepto, columnaFecha, sheet, values) {
25   for (var row = values.length - 1; row >= 0; --row) {
26     if (values[row][columnaConcepto] == concepto && values[row][columnaFecha] == +fechaFormatoLargo) {
27       sheet.deleteRow(parseInt(row) + 1); // loop is 0-indexed, deleteRow is 1-indexed
28     }
29   }
30 }
31
32 function upd(fechaFormatoLargo, concepto, columnaConcepto, columnaFecha, sheet, values) {
33   var lastRow = values.length - 1;
34   sheet.getRange(parseInt(lastRow) + 1, 5).clear(); // Elimina la acción de UPDATE
35   for (var row = lastRow - 1; row >= 0; --row) {
36     Logger.log(values[row][columnaFecha] + " + values[row][columnaFecha]);
37     if (values[row][columnaConcepto] == concepto && values[row][columnaFecha] == +fechaFormatoLargo) {
38       sheet.deleteRow(parseInt(row) + 1); // loop is 0-indexed, deleteRow is 1-indexed
39     }
40   }
41 }
42

```

Figura 12: Captura del Apps Script.

4.4.6. Creación del activador

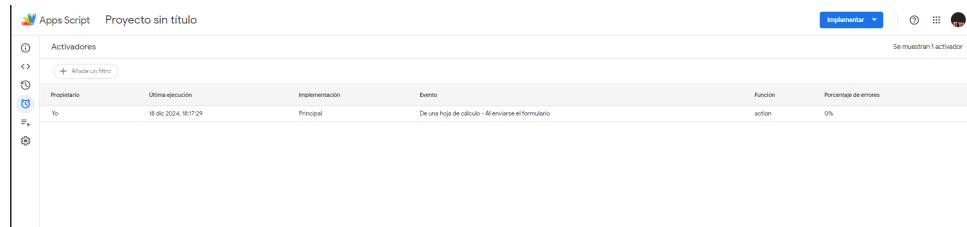


Figura 13: Captura del activador

4.5. Sección de consultar

En esta sección del sistema, se publica la hoja de cálculo utilizada como base de datos para que los usuarios puedan visualizar los datos de manera sencilla. Para ello, se realizan los siguientes pasos:

1. **Preparación de la Hoja de Cálculo:** Se elimina la columna de *Fecha y Hora* (que indica cuándo fue enviada la información) y la columna de *Acción*, ya que contienen datos internos que no son relevantes para el cliente final.
2. **Publicación de la Hoja de Cálculo:** La hoja de cálculo se publica en la web utilizando las opciones de compartición de **Google Sheets**. Esto permite generar un enlace público que puede ser accesible por cualquier usuario autorizado.
3. **Visualización en la Aplicación:** Para mostrar la hoja de cálculo publicada dentro de la aplicación, se utiliza un componente llamado **Visor Web de App Inventor**. Este componente permite integrar el enlace público de la hoja de cálculo directamente en la interfaz de usuario, asegurando que los clientes puedan consultar la información actualizada en tiempo real desde sus dispositivos móviles.
4. **Resultados:** Este enfoque garantiza una experiencia de usuario fluida, al ofrecer un método rápido y accesible para visualizar los datos relevantes, sin comprometer la seguridad ni exponer información sensible o interna.

4.5.1. Interfaz de la seccion de consultas

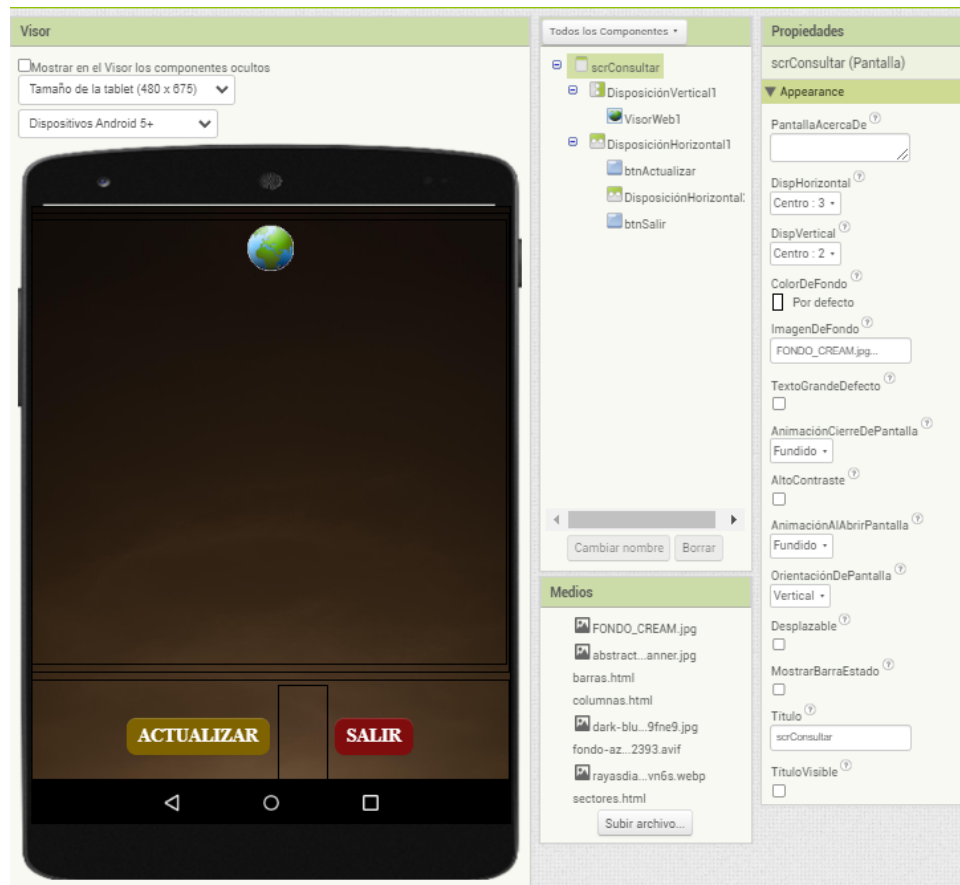


Figura 14: Captura de la interfaz de la seccion de consultas

4.5.2. Logica de la seccion de consultas



Figura 15: Captura de los bloques de la seccion de consultas

4.5.3. Publicacion del Exel

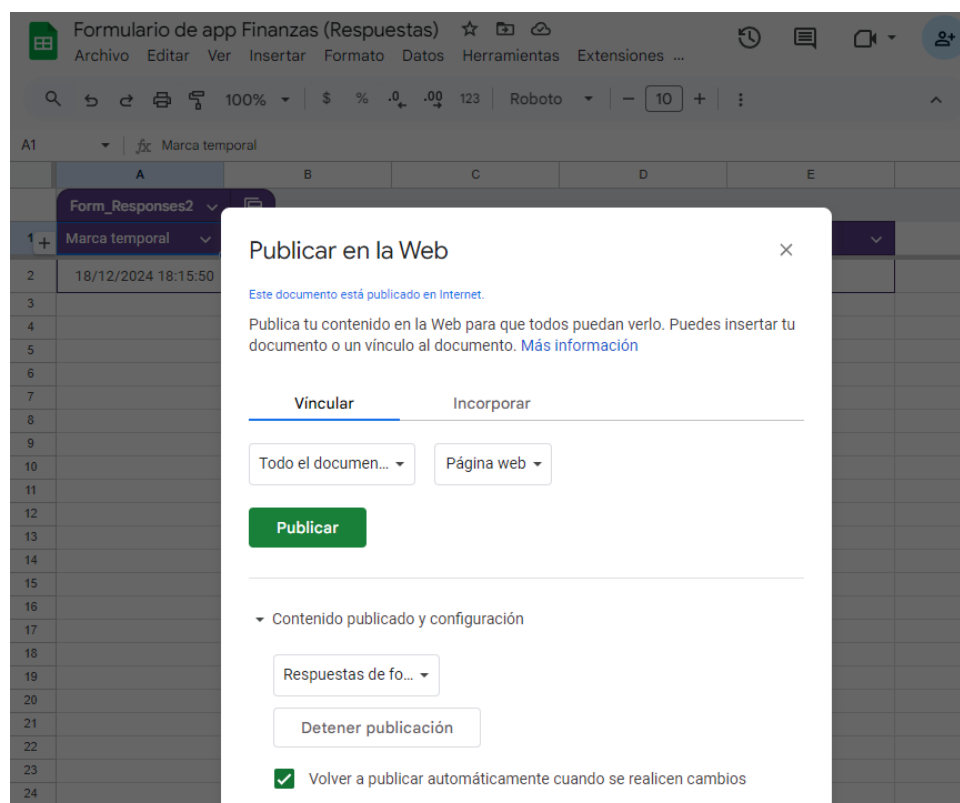


Figura 16: Captura de la publicacion del excel

4.6. Sección de graficos

La visualización de gráficos dentro de la aplicación se implementa utilizando una extensión para la generación de representaciones visuales, como gráficos de barras, columnas y tortas, basados en los datos de la hoja de cálculo vinculada al sistema. A continuación, se describen los detalles del proceso:

1. **Generación de Gráficos:** Los gráficos se generan mediante un código que utiliza una extensión especializada en crear representaciones visuales de datos, tomando como referencia la información almacenada en la hoja de cálculo vinculada al sistema.
2. **Integración en la Aplicación:** Los gráficos se presentan dentro de la aplicación mediante un componente llamado **Visor Web**, disponible en **App Inventor**. Este visor permite renderizar archivos HTML que contienen el código necesario para mostrar los gráficos dinámicos.

3. **Estructura de Archivos:** Es importante destacar que los archivos HTML utilizados para mostrar los gráficos deben estar incluidos directamente en el proyecto de **App Inventor**. Esto asegura que puedan ser accesibles por la aplicación y utilizados correctamente durante la ejecución.
4. **Resultados:** Este enfoque permite a los usuarios visualizar información financiera de manera clara e interactiva, mejorando la comprensión de los datos a través de gráficos intuitivos. Además, la integración con App Inventor garantiza un funcionamiento fluido y la disponibilidad de los gráficos dentro de la interfaz de la aplicación.

4.6.1. Interfaz de las seccion de graficos

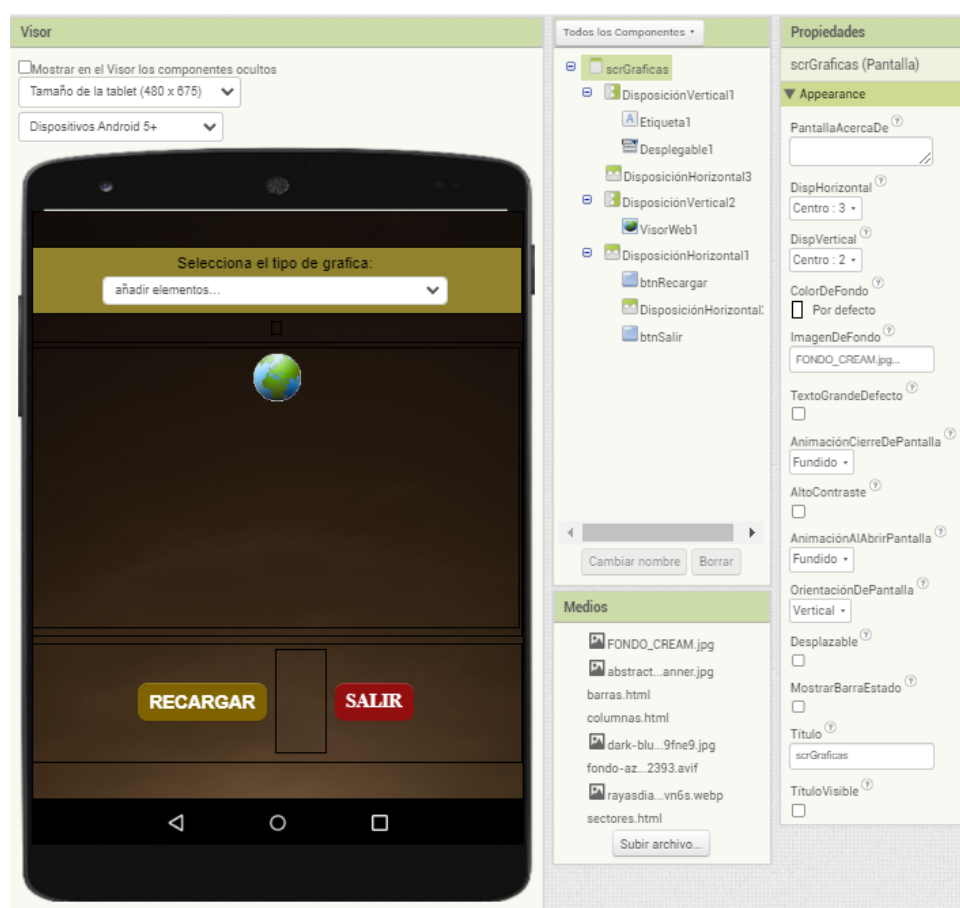


Figura 17: Captura de la interfaz de la seccion graficos

4.6.2. Logica de la seccion de graficos

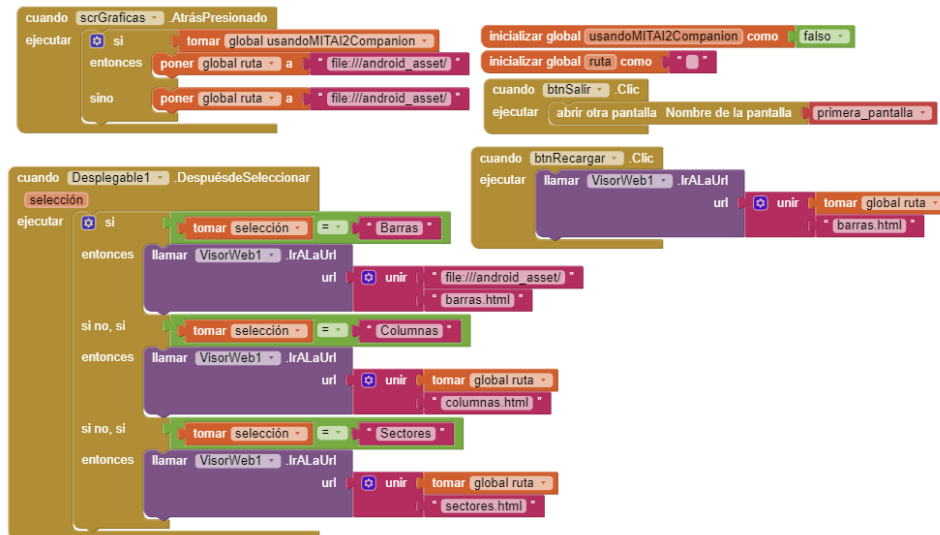


Figura 18: Captura de los bloques de seccion de graficos

4.6.3. Codigo para los graficos

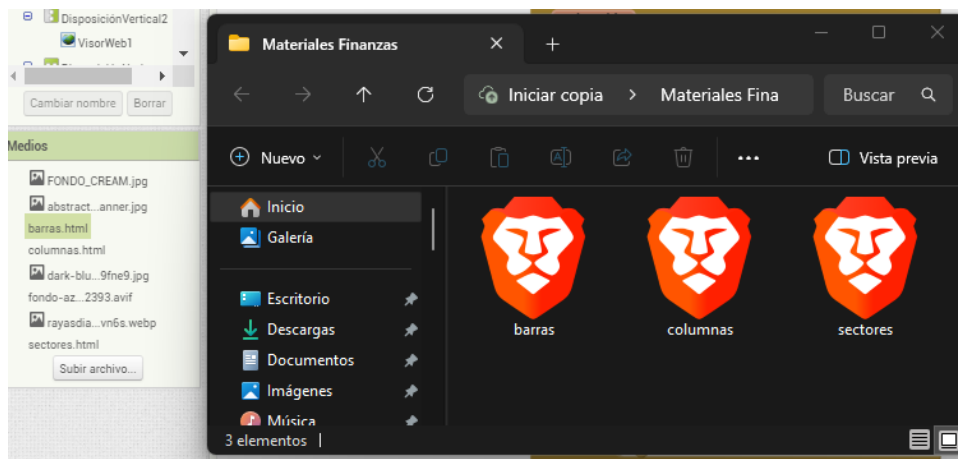
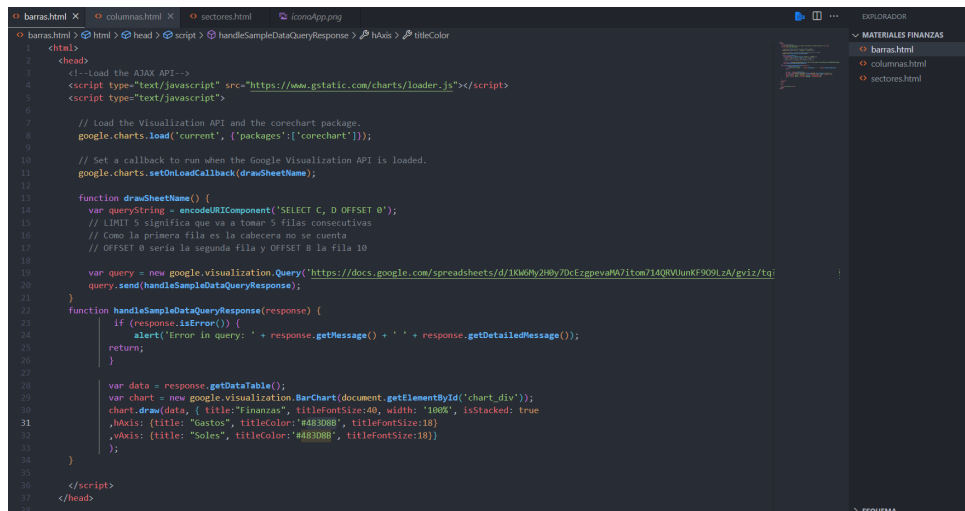


Figura 19: Captura de los archivos .html en el appInventor



```

1 <html>
2 <head>
3 <!--Load the AJAX API-->
4 <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
5 <script type="text/javascript">
6
7 // Load the Visualization API and the corechart package.
8 google.charts.load('current', {'packages':['corechart']});
9
10 // Set a callback to run when the Google Visualization API is loaded.
11 google.charts.setOnLoadCallback(drawSheetName);
12
13 function drawSheetName() {
14   var queryString = encodeURIComponent('SELECT C, D OFFSET 0');
15   // LIMIT 5 significa que va a tomar 5 filas consecutivas
16   // Como la primera fila es la cabecera no se cuenta
17   // OFFSET 0 seria la segunda fila y OFFSET 8 la fila 10
18
19   var query = new google.visualization.Query('https://docs.google.com/spreadsheets/d/3KW6WZ4dy7DcEzpevaWU7itcm714QRV1umKF909LzAJ/edit/tq?
20   query.send(handleSampleDataQueryResponse);
21 }
22 function handleSampleDataQueryResponse(response) {
23   if (response.isError()) {
24     alert('Error in query: ' + response.getMessage() + ' ' + response.getDetailedMessage());
25     return;
26   }
27
28   var data = response.getDataTable();
29   var chart = new google.visualization.BarChart(document.getElementById('chart_div'));
30   chart.draw(data, { title: 'Finanzas', titleFontSize: 40, width: '100%', isStacked: true
31   , hAxis: {title: 'Gastos', titleColor: '#483D8B', titleFontSize: 18}
32   , vAxis: {title: 'Sesiones', titleColor: '#483D8B', titleFontSize: 18}
33   });
34 }
35
36 </script>
37 </head>
38

```

Figura 20: Captura de los archivos .html

5. Ejecución de la aplicación

La ejecución de la aplicación se desarrolla en varios pasos clave, diseñados para garantizar un flujo de trabajo intuitivo y eficiente para los usuarios. A continuación, se describe el proceso:

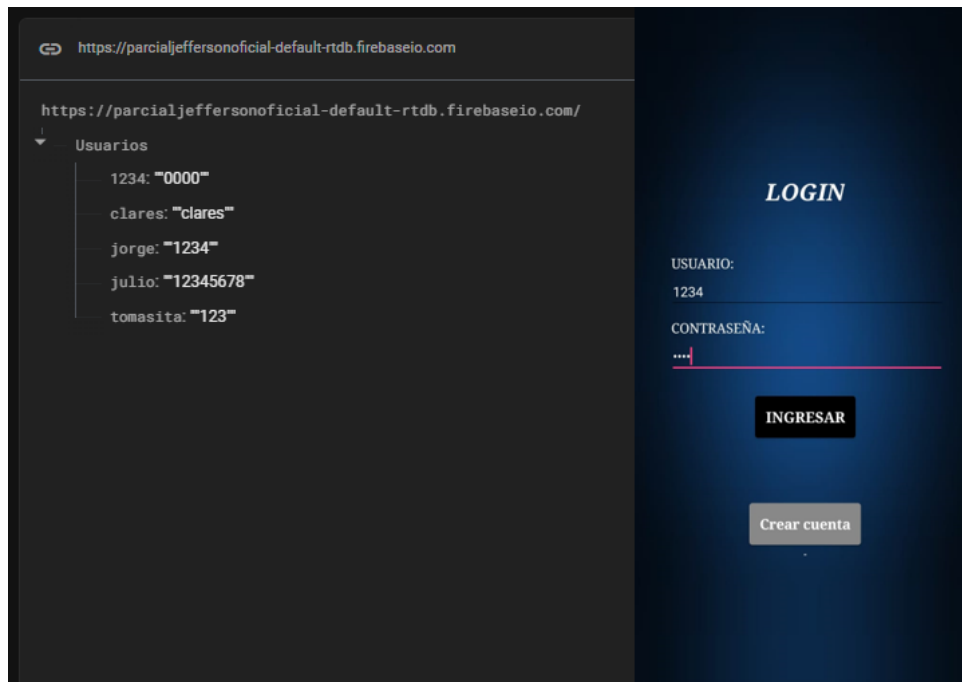


Figura 21: Prueba del uso del login

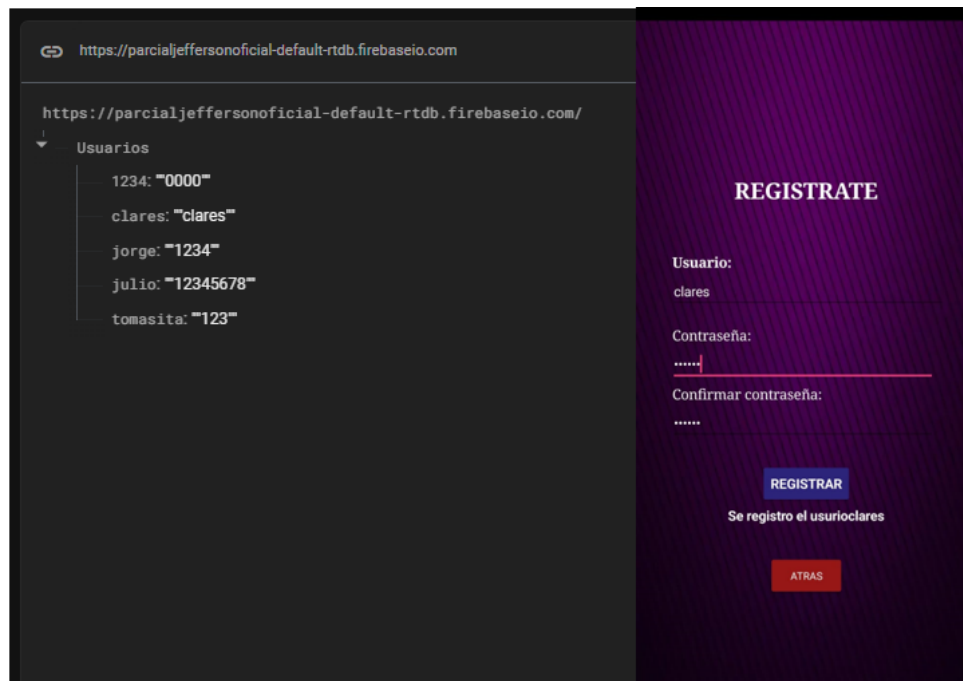


Figura 22: Prueba del uso para crear nuevos usuarios

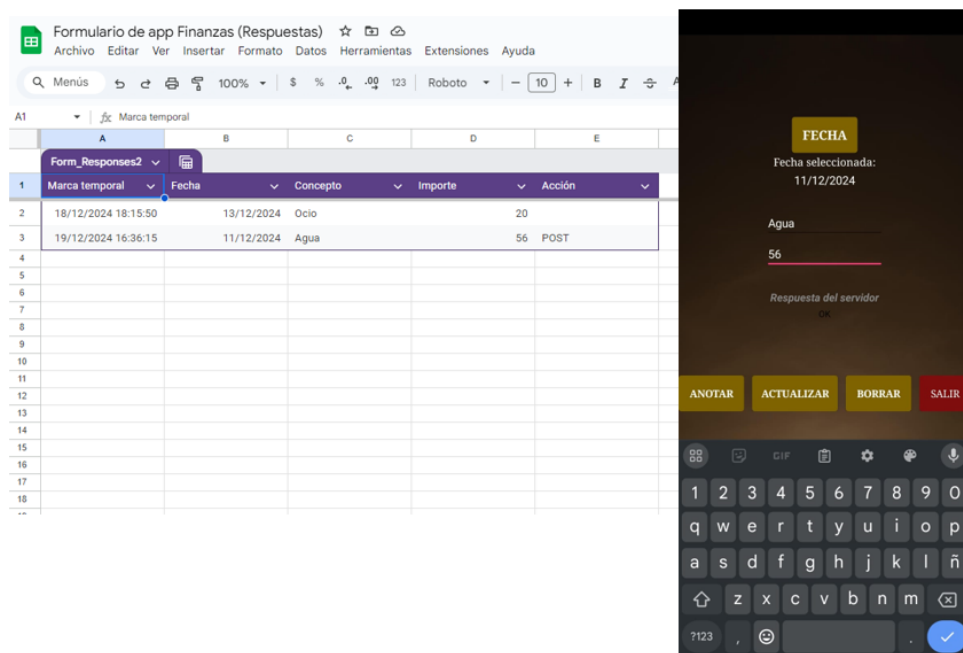


Figura 23: Prueba de la creacion de un nuevo dato

Fecha	Concepto	Importe	Acción
13/12/2024	Ocio	20	
11/12/2024	Agua	56	POST

Figura 24: Prueba de su visualizacion

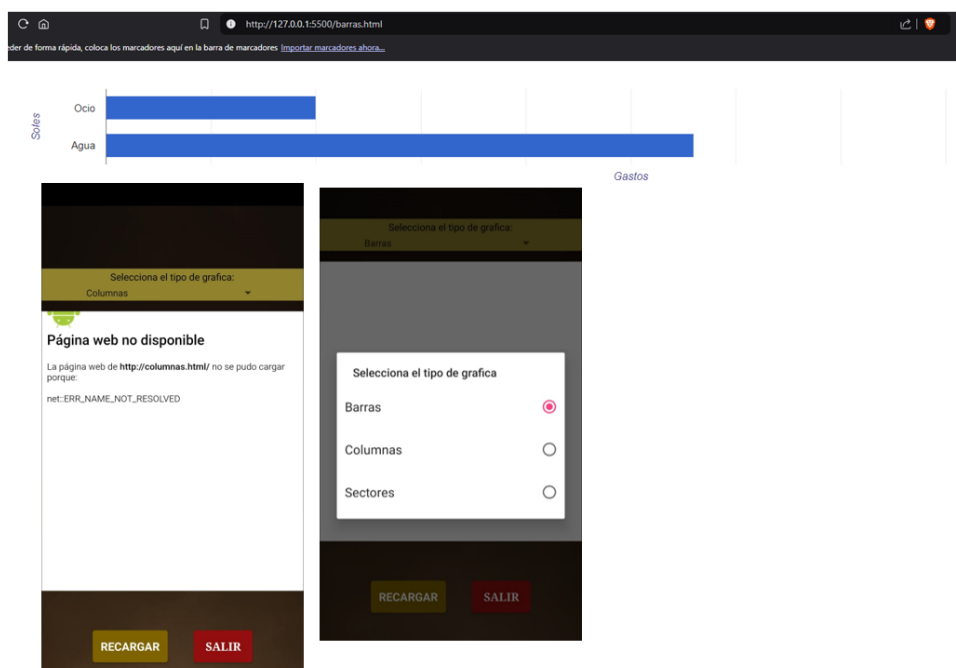


Figura 25: Prueba de los graficos