

Reporte de la Aplicación de Envío de Datos de Inventario de Ventas a una Hoja de Cálculo de Google con App Inventor

1. Introducción

Este reporte detalla el desarrollo y funcionamiento de una aplicación creada con App Inventor, la cual permite enviar datos de inventario de ventas a una hoja de cálculo de Google. La aplicación está diseñada para facilitar la gestión y seguimiento de ventas, proporcionando una manera eficiente de actualizar los registros en tiempo real.

2. Objetivos

- Automatizar la actualización de inventarios de ventas en una hoja de cálculo de Google.
- Facilitar la entrada de datos de ventas a través de una interfaz amigable.
- Asegurar la integridad y precisión de los datos enviados.

3. Tecnologías Utilizadas

- **App Inventor:** Plataforma de desarrollo visual utilizada para crear la aplicación móvil.
- **Google Sheets API:** Servicio utilizado para interactuar con Google Sheets y actualizar los datos de forma programática.
- **Google Apps Script:** Script personalizado para manejar las solicitudes de la API y actualizar la hoja de cálculo.
- **Componente Web:** Utilizados para enviar datos desde la aplicación móvil a la hoja de cálculo.

4. Desarrollo de la Aplicación

4.1. Interfaz de Usuario

La interfaz de usuario de la aplicación fue diseñada para ser intuitiva y fácil de usar. Los principales componentes incluyen:

Campos de entrada: Para ingresar datos como nombre del fecha, concepto e importe.

Botón de envío: Para enviar los datos ingresados a la hoja de cálculo de Google.



4.2. Lógica de la Aplicación

La lógica de la aplicación en App Inventor incluye:

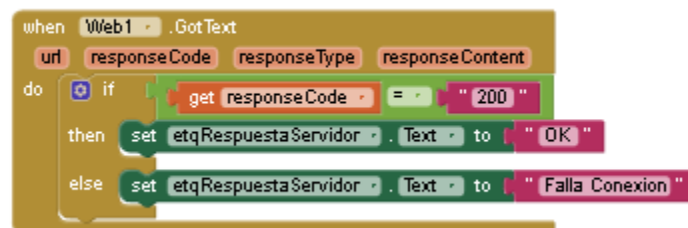
- Captura de datos: Los datos de ventas son capturados a través de campos de entrada.



- Validación de datos: Se realizan validaciones básicas para asegurar que los campos obligatorios no estén vacíos y que los datos sean del tipo correcto.



- Envío de datos: Los datos validados se envían a un servidor intermedio (Google Apps Script) mediante una llamada HTTP POST.



4.3. Google Apps Script

Un script de Google Apps Script fue desarrollado para recibir los datos enviados desde la aplicación y actualizar la hoja de cálculo de Google. Los pasos son los siguientes:

- Recepción de datos: El script recibe los datos a través de una solicitud HTTP POST.
- Actualización de la hoja de cálculo: Los datos recibidos se añaden a la siguiente fila disponible en la hoja de cálculo.

```
function action(e) {
  var fecha = e.values[1]; // Fecha
  var concepto = e.values[2]; // Concepto
  var action = e.values[4]; // Acción
  var sheet = SpreadsheetApp.getActiveSheet();
  var values = sheet.getDataRange().getValues();
  var dia=fecha.split("/")[0];
  var mes=fecha.split("/")[1];
  var anyo=fecha.split("/")[2];
  var fechaFormatoLargo = new Date(anyo+"/"+mes+"/"+dia);

  if (action == "DELETE"){
    del(fechaFormatoLargo, concepto, 2, 1, sheet, values);
  }
  if (action == "UPDATE") {
    upd(fechaFormatoLargo, concepto, 2, 1, sheet, values);
  }
}

function del(fechaFormatoLargo, concepto, columnaConcepto, columnaFecha, sheet, values){
  for(var row = values.length -1; row >= 0; --row){
    if (values[row][columnaConcepto] == concepto && +values[row][columnaFecha] === +fechaFormatoLargo){
      sheet.deleteRow(parseInt(row)+1); // loop is 0-indexed, deleteRow is 1-indexed
    }
  }
}

function upd(fechaFormatoLargo, concepto, columnaConcepto, columnaFecha, sheet, values){
  var lastRow = values.length - 1;
  sheet.getRange(parseInt(lastRow)+1, 5).clear(); // Elimina la acción de UPDATE
  for(var row = lastRow - 1; row >= 0; --row){
    Logger.log('values[row][columnaFecha] = ' + values[row][columnaFecha]);
    if (values[row][columnaConcepto] == concepto && +values[row][columnaFecha] === +fechaFormatoLargo)
    {
      sheet.deleteRow(parseInt(row)+1); // loop is 0-indexed, deleteRow is 1-indexed
    }
  }
}
```

5. Pruebas y Validación

Se realizaron pruebas para asegurar el correcto funcionamiento de la aplicación:

- Verificar la correcta comunicación entre App Inventor y Google Sheets.
- Asegurar que los datos enviados son correctos y se almacenan adecuadamente.

6. Despliegue

- La aplicación se exportó a formato apk y se instaló en un dispositivo móvil.
- El script se implementó y configuró con permisos adecuados para permitir la escritura en la hoja de cálculo.

7. Conclusiones

La aplicación desarrollada proporciona una solución para la gestión de inventarios de ventas. Utilizando App Inventor y Google Sheets API, se logró una integración efectiva que facilita la actualización en tiempo real de los datos de ventas, mejorando la gestión y seguimiento de las ventas.