

Manual Técnico de la Aplicación de Registro de Compras de Acciones

1. Introducción

1.1 Propósito del documento

Este manual técnico está dirigido a desarrolladores y tiene como objetivo proporcionar una guía detallada sobre la arquitectura, funcionalidades, y configuración de la aplicación de registro de compras de acciones. El documento cubre tanto el frontend como la integración con la API de Alpha Vantage.

1.2 Alcance

El manual cubre:

- La estructura de carpetas y archivos del frontend.
- Las tecnologías utilizadas (HTML, CSS, JavaScript, Chart.js).
- La integración con la API de Alpha Vantage.
- Ejemplos de código y explicaciones de las funcionalidades principales.

1.2 Audiencia

Este documento está dirigido a:

- Desarrolladores frontend.
- Integradores de APIs.
- Mantenedores del proyecto.

2. Descripción General del Sistema

2.1 Resumen de la Aplicación

La aplicación permite a los usuarios registrar compras de acciones por fechas y cantidades, y visualizar el historial de compras, esta se puede ordenar alfabéticamente y ganancia/pérdida (ascendente o descendente). Adicionalmente tiene una página dedicada a presentar el resumen de sus inversiones a través de un dashboard. El dashboard muestra:

- Las 5 empresas con mayor cantidad de acciones compradas.
- Las empresas con mayor ganancia.
- Las empresas con mayor pérdida.

Y una tabla que muestra el registro de las acciones, pero de manera consolidada, es decir, solo va a mostrar una fila por cada acción comprada.

2.2 Arquitectura del Sistema

La aplicación sigue una arquitectura frontend basada en HTML, CSS y JavaScript, con integración de la librería Chart.js para la visualización de datos y la API de Alpha Vantage para obtener información en tiempo real sobre las acciones.

3. Estructura de Carpetas y Archivos del Frontend

3.1 Estructura de Carpetas

La estructura se forma de la siguiente manera

- Aseets: Contiene los recursos usados en la aplicación, como imágenes e iconos.
- Controllere: Contiene los controladore para manejar los cambios de las tablas.
- Pages: Contiene en si la web separada por pagina cada una tiene su archivo html, css y js.
- Servicios: Contiene la lógica de las solicitudes a la API.
- Test: Contiene las pruebas unitarias realizadas
- Utils: Contine lógica reutilizable en los puntos anteriores.

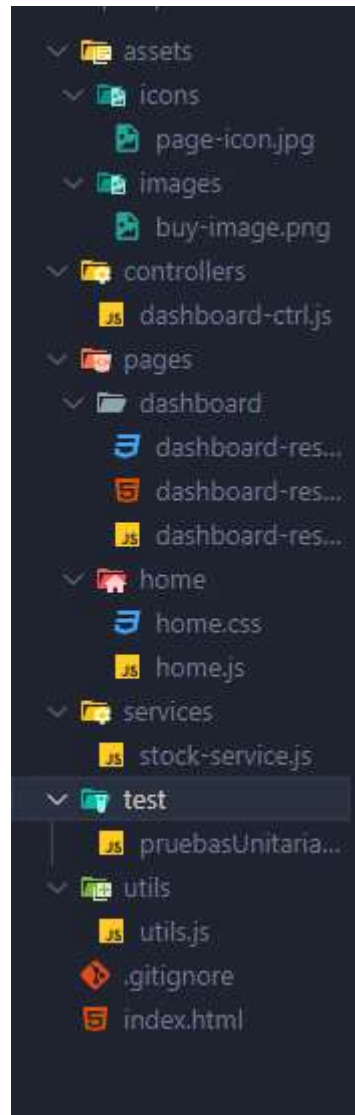


Ilustración 1 – Estructura del proyecto

4. Tecnologías Utilizadas

4.1 Frontend

- HTML: Para la estructura de la interfaz de usuario.
- CSS: Para los estilos y diseño visual.
- JavaScript: Para la lógica de la aplicación y la interacción con el usuario.

4.2 Librerías

- Chart.js: Para la visualización de datos en forma de gráficos.
- API de Alpha Vantage: Para obtener datos reales sobre precios de acciones.

5. Integración con la API de Alpha Vantage

5.1 Descripción de la API

La API de Alpha Vantage proporciona datos financieros en tiempo real, incluyendo precios de acciones, volúmenes de trading, y más. Se utiliza para calcular ganancias y pérdidas en el dashboard.

5.2 Uso de la API

- Endpoint utilizado: <https://www.alphavantage.co/query>
- Parámetros usados:
 - function=TIME_SERIES_DAILY: Para obtener datos diarios de acciones.
 - symbol={ticker}: Símbolo de la acción (ej: "AAPL" para Apple).
 - apikey={API_KEY}: Clave de API para autenticación.

5.2 Limitaciones para uso de la API

La aplicación está funcionando con el servicio gratuito lo que limita a una máximo de 25 peticiones por día. Tomando esas consideraciones se espera a futuro conseguir el servicio sin limitaciones y en tiempo real.

6. Funcionalidades Principales

6.1 Registro de Compras de Acciones

- Formulario HTML: Permite al usuario ingresar la fecha, cantidad, y símbolo de la acción comprada.
- Almacenamiento: Los datos se almacenan en el localStorage del navegador.

6.2 Dashboard de historial de compras

Tabla que presenta el historial de compra de las acciones (nombre, fecha, cantidad, precio de compra, precio actual, ganancias/perdidas)

6.3 Dashboard de Resumen

- Gráfico de las 5 empresas con más acciones: Utiliza Chart.js para mostrar un gráfico de barras.
- Gráfico de ganancias/pérdidas: Muestra un gráfico de líneas con la evolución de las inversiones.

6.4 Tabla de Resumen

La tabla presenta una consolidación de las acciones, es decir agrupa a todas las compras en una sola acción y esta muestra nombre, cantidad total, valor USD, precio de costo, ganancia/perdidas en dólares y porcentajes.

7. Ejemplos de Código

7.1 Búsqueda de una Acción

```
export async function fetchStockData(stockName, purchaseDate) {
  try {
    const response = await fetch(
      `https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol=${stockName}&apikey=${apiKey}&outputsize=full&datatype=csv`
    );

    const data = await response.text();
    const rows = data.split('\n');

    let purchasePrice = null;
    let currentPrice = null;

    for (let row of rows) {
      const [date, open, high, low, close, volume] = row.split(',');

      if (date && date === purchaseDate) {
        purchasePrice = parseFloat(close);
      }
      if (date && new Date(date) <= new Date() && !currentPrice) {
        currentPrice = parseFloat(close);
      }

      if (purchasePrice && currentPrice) {
        break;
      }
    }

    return { purchasePrice, currentPrice };
  } catch (error) {
    console.error('Error al realizar la solicitud:', error);
    throw new Error('Hubo un error al obtener los datos.');
```

7.2 Configuración de Chart.js

```
function loadChartData(stockNames, values, label) {  
    const canvas = document.querySelector('.myChart');  
  
    if (!canvas) {  
        console.error('Canvas con la clase "myChart" no encontrado.');        return;  
    }  
  
    if (currentChart) {  
        currentChart.destroy();  
    }  
  
    if (stockNames.length === 0) {  
        stockNames = ['No hay datos'];  
        values = [0];  
    }  
  
    currentChart = new Chart(canvas, {  
        type: 'doughnut',  
        data: {  
            labels: stockNames,  
            datasets: [  
                {  
                    data: values,  
                    label: label,  
                    borderWidth: 1,  
                },  
            ],  
        },  
    });  
}
```

8. Configuración y Despliegue

8.1 Requisitos

- Navegador web moderno (Chrome, Firefox, Edge).
- Conexión a Internet para acceder a la API de Alpha Vantage.

8.2 Despliegue Local

1. Clona el repositorio
2. Abre el archivo `index.html` en tu navegador.
3. Asegúrate de tener una clave de API válida para Alpha Vantage.

9. Pruebas

- Pruebas manuales: Verifica que el formulario de registro funcione correctamente y que los gráficos se actualicen.
- Pruebas de la API: Asegúrate de que los datos de Alpha Vantage se procesen correctamente.

10. Contacto y Soporte

- Equipo de desarrollo: Chileno Jefferson, Guerra Sebastián, Quille David, Tituaña Dorian.
- Repositorio del proyecto: <https://github.com/JeffersonDaviid/CE-Proyecto>