



**ESCUELA POLITÉCNICA NACIONAL**  
**FACULTAD DE INGENIERIA DE SISTEMAS**  
**SOFTWARE**  
**CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE**  
**PROYECTO [1B]**  
**GRUPO N°4**

**Integrantes:** Erick Cabezas, Kenny Pinchao, Kevin Cano

**Fecha:** 7/01/2024

**DOCUMENTACIÓN**

**I. OBJETIVOS:**

- Crear un artefacto funcional mediante la implementación de 2 iteraciones para entregar el producto al cliente.
- Elaborar la planificación del reléase con el objetivo de proporcionar una guía general para todo el equipo, delineando las acciones necesarias para el desarrollo del producto.
- Realizar la planificación de las 2 iteraciones, cada una con duración de 1 semana para agregar valor de manera significativa al usuario.

**II. DESARROLLO**

**Stakeholders:**

De acuerdo a lo especificado en este proyecto por el cliente, los tipos de usuario que interactúan con el sistema son:

- Usuario (+12)

**Alcance del proyecto:**

El proyecto nació gracias a la propuesta de un usuario para poder tener un registro de sus acciones compradas:

El sistema deberá permitir:

- Registrar una nueva acción.
- Buscar una acción ya registrada.
- Visualizar una lista con todas las acciones registradas.
- Calcular el costo total de compra = Precio de compra por acción\*cantidades de acciones.

A continuación, se presentan los requerimientos que cubren las necesidades del usuario:

*Tabla 1. Requerimientos funcionales para la aplicación.*

Código	Requerimiento
Requerimientos funcionales	
RFP1	La aplicación debe permitir al usuario registrar la compra de acciones
RFP2	La aplicación debe permitir al usuario observar una lista de las acciones compradas
RFP3	La aplicación debe permitir buscar con su nombre una acción registrada por el usuario.
RFP4	La información que debe mostrarse de las acciones es: el nombre de la acción, fecha de compra, precio de compra por acción, cantidades de acciones y costo total de compra.
RFP5	La información que ingresa el usuario al registrar una acción es el nombre de la acción, fecha de compra, precio de compra por acción y cantidades de acciones
RFP6	El costo total de compra cuando del usuario registra una compra de acción se calcula automáticamente.
RFP7	La interfaz de usuario debe poderse cambiar a modo oscuro.
RFP8	Las acciones registradas también se podrán eliminar del registro.

### **Configuración del proyecto:**

Para este proyecto se trabajará con la librería React de JavaScript creada por Facebook, react nos permite diseñar e implementar la interface de una aplicación web [1]. Se puede decir que es la V en el patrón MVC (Model, View, Controller) y C es nodesjs.

Para la base de datos se utilizará MySql en un servidor remoto, en el que se conectará la aplicación para obtener e ingresar datos a través de una conexión con la utilización del software Radmin VPN.

También se utilizará una arquitectura basada en componentes, ya que, se estará utilizando la librería de react de JavaScript la cual están basadas en componentes independientes, dinámicos y reutilizables.

### Arquitectura:

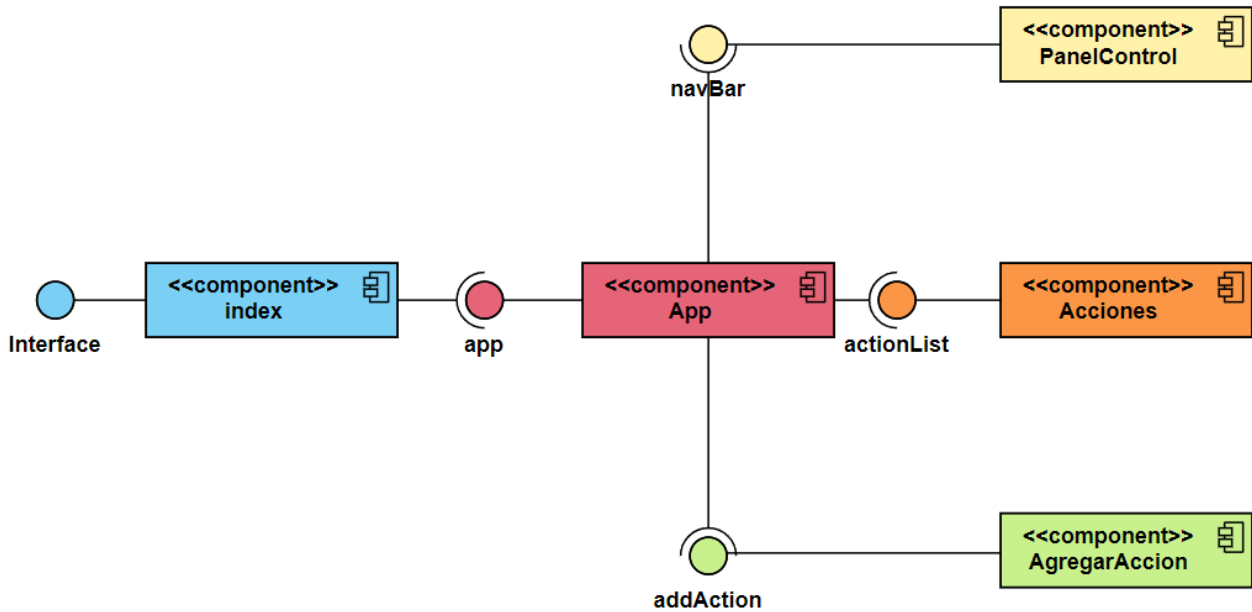


Ilustración 1. Arquitectura del software del sistema [2].

### Diseño de base de datos:

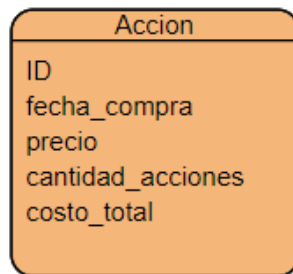


Ilustración 2. Modelo entidad relación de la base de datos del sistema.

### Artefactos XP

Para iniciar con la **primera parte “Artefactos XP”** lo que se hizo fue definir un tema, las épicas, y luego las historias de usuario a partir del proyecto.

**Tema:** Como usuario, quiero tener un registro de mis acciones compradas.

**Épica 1:** Como usuario, deseo tener un listado de mis acciones compradas.

<b>Título: Listado de acciones</b>	<b>Número: 1</b>
<b>Historia de usuario:</b> Como usuario, deseo visualizar una lista de mis acciones compradas para poder ver la información de cada una.	
<b>Criterios de aceptación:</b> Dado que ingreso a la aplicación, cuando inicie, Entonces se muestra las acciones compradas con la información necesaria.	

<b>Título: información de acciones</b>	<b>Número: 2</b>
<b>Historia de usuario:</b> Como usuario, deseo poder observar el nombre de la acción, fecha de compra, precio de compra por acción, cantidades de acciones y costo total de compra.	
<b>Criterios de aceptación:</b> Dado que se presente el listado de acciones adquiridas, cuando el usuario mire su información, Entonces el listado de acciones tiene que mostrar el nombre, fecha de compra, precio de compra por acción, cantidades de acciones y costo total de compra.	

**Épica 2:** Como usuario, deseo registra las acciones que compro para poder manejar de una mejor manera la información de cada una de mis acciones.

<b>Título: Registrar acción</b>	<b>Número: 3</b>
<b>Historia de usuario:</b> Como usuario, deseo ingresar el nombre de la acción, fecha de compra, precio de compra por acción y cantidades de acciones para que se realice su registro.	
<b>Criterios de aceptación:</b> Dado deseo agregar una acción, Cuando doy en el botón agregar, Entonces se presenta un formulario para registrar mi acción con el nombre de la acción, fecha de compra, precio de compra por acción y cantidades de acciones.	

<b>Título: Calculo de Costo total de compra</b>	<b>Número: 4</b>
<b>Historia de usuario:</b> Como usuario, deseo que el costo total de la compra se calcule automáticamente para solo ingresar el nombre de la acción, fecha de compra, precio de compra por acción y cantidades de acciones.	

**Criterios de aceptación:**

Dado que lleno el formulario de registro de acción, cuando le doy en agregar, entonces se mostrará en la lista de acciones con la siguiente información: El nombre de la acción, fecha de compra, precio de compra por acción, cantidades de acciones y costo total de compra.

**Épica 3:** Como usuario, deseo poder buscar y eliminar una acción registrada para tener una administración de mis acciones compradas.

Título: Búsqueda de acciones por su nombre	Número: 5
<b>Historia de usuario:</b> Como usuario, deseo ingresar el nombre de una acción que deseo para poderla filtrar y encontrarla más rápido.	
<b>Criterios de aceptación:</b> Dado que deseo una acción en específico, cuando ingrese su nombre en el buscador, entonces se mostrar la acción con ese nombre si existe.	

Título: Eliminar acciones	Número: 6
<b>Historia de usuario:</b> Como usuario, deseo poder eliminar una acción no deseada que se encuentra registrada para administrar de mejor manera las cuentas activas y registradas.	
<b>Criterios de aceptación:</b> Dado que voy a eliminar una acción, cuando selecciones las acciones a eliminar, Entonces cuando presione el botón eliminar ya no se mostrarán en el listado de acciones.	

Las historias de usuario fueron realizadas con los atributos **INVEST**. **Independiente** porque cada historia de usuario no depende de otras de otras historias o su dependencia es mínima. **Negociable** pues son flexibles a cambios. **Valiosa** porque cada historia brinda valor al usuario pues se alinea con el tema, la épica y como se visualizará de manera más clara también con el objetivo de cada iteración. **Estimable** porque se determinó el esfuerzo necesario para cada historia tanto con el valor real y el valor estimado. **Pequeña** porque las historias de usuario que fueron seleccionadas para las iteraciones son suficientemente pequeñas para completarlos en una iteración. **Testeable** porque cada historia de usuario tiene asociada al menos una prueba que permite verificar esa funcionalidad.

Muchos de estos atributos antes mencionados son más evidentes en cada iteración.

**Eventos XP****Planificación reléase**

Para la **segunda parte “Eventos XP”**, el equipo comenzó con la planificación del reléase de 2 semanas con 2 iteraciones, cada una con duración de una semana. Primero, se definió un objetivo para el reléase con la finalidad de que el equipo de trabajo se guíe acerca de lo que se va a hacer.

**Objetivo del reléase:** Automatizar en un 75% la administración de las acciones compradas por el usuario.

Luego, se realizó una selección de historias de usuario en función del objetivo para el que consideramos los criterios INVEST previamente mencionados.

El equipo de trabajo selecciono para este punto todas las historias de usuario, ya que el reléase en esta primera entrega es corto, pero se les dio un nuevo orden que fue definido por los desarrolladores.

<b>Título: Registrar acción</b>	<b>Número: 1</b>
<b>Historia de usuario:</b> Como usuario, deseo ingresar el nombre de la acción, fecha de compra, precio de compra por acción y cantidades de acciones para que se realice su registro.	

<b>Título: Calculo de Costo total de compra</b>	<b>Número: 2</b>
<b>Historia de usuario:</b> Como usuario, deseo que el costo total de la compra se calcule automáticamente para solo ingresar el nombre de la acción, fecha de compra, precio de compra por acción y cantidades de acciones.	

<b>Título: Eliminar acción registrada</b>	<b>Número: 3</b>
<b>Historia de usuario:</b> Como usuario, deseo poder eliminar una acción no deseada que se encuentra registrada para administrar de mejor manera las cuentas activas y registradas.	

<b>Título: Listado de acciones</b>	<b>Número: 4</b>
<b>Historia de usuario:</b> Como usuario, deseo visualizar una lista de mis acciones compradas para poder ver la información de cada una.	

<b>Título: información de acciones</b>	<b>Número: 5</b>
<b>Historia de usuario:</b> Como usuario, deseo poder observar el nombre de la acción, fecha de compra, precio de compra por acción, cantidades de acciones y costo total de compra.	

<b>Título: Búsqueda de acciones por su nombre</b>	<b>Número: 6</b>
<b>Historia de usuario:</b> Como usuario, deseo ingresar el nombre de una acción que deseo para poderla filtrar y encontrarla más rápido.	

<b>Título: Cambiar de modo de visualización a dark.</b>	<b>Número: 7</b>
<b>Historia de usuario:</b> Como usuario, deseo poder cambiar el modo de visualización de la app a modo dark para tener dos modos de visualización de la app.	

### Planificación iteraciones

Se definió un objetivo para cada iteración, se seleccionó las historias que más se acoplan al objetivo de la iteración (así se las agrupo), se definió las tareas, se asignó las tareas a los miembros del equipo, se realizó las estimaciones de tiempo y se encontró la velocidad del equipo. Hay que tener en cuenta que cada iteración es de una semana con 40 horas reales.

#### Iteración 1

**Objetivo:** Permitir al usuario registrar sus acciones compradas.

<b>Número</b>	<b>1</b>	<b>Estimado: 6 horas</b>
<b>Título</b>	Registrar acción	
<b>Historia de usuario</b>	Como usuario, deseo ingresar el nombre de la acción, fecha de compra, precio de compra por acción y cantidades de acciones para que se realice su registro.	
<b>Criterio de aceptación</b>	Dado deseo agregar una acción, Cuando doy en el botón agregar, Entonces se presenta un formulario para registrar mi acción con el nombre de la acción, fecha de compra, precio de compra por acción y cantidades de acciones.	
<b>Tareas</b>	T1. Crear la base de datos. (Estimado: 1 horas) T2. Conectar la base de datos. (Estimado: 1 horas) T3. Poner un botón “Agregar Acción”. (Estimado: 0.5 hora) T4. Crear el formulario de registro que permita agregar una acción. (Estimado: 2 horas) T5. Crear la sentencia en la base de datos para crear un nuevo registro (Estimado: 0.5 horas) T6. Vincular el formulario con la sentencia de la base de datos para crear la funcionalidad de crear un nuevo registro. (Estimado: 1 horas)	

<b>Número</b>	<b>2</b>	<b>Estimado: 3 horas</b>
<b>Título</b>	Cálculo de Costo total de compra	

<b>Historia de usuario</b>	Como usuario, deseo que el costo total de la compra se calcule automáticamente para solo ingresar el nombre de la acción, fecha de compra, precio de compra por acción y cantidades de acciones.
<b>Criterio de aceptación</b>	Dado que lleno el formulario de registro de acción, cuando le doy en agregar, entonces se mostrará en la lista de acciones con la siguiente información: El nombre de la acción, fecha de compra, precio de compra por acción, cantidades de acciones y costo total de compra.
<b>Tareas</b>	T1. Calcular el costo total de la compra. (Estimado: 1 hora) T2. Ingresar la acción a la base de datos. (Estimado: 2 horas)

<b>Número</b>	<b>3</b>	<b>Estimado: 3 horas</b>
<b>Título</b>	Eliminar acción registrada	
<b>Historia de usuario</b>	Como usuario, deseo poder eliminar una acción no deseada que se encuentra registrada para administrar de mejor manera las cuentas activas y registradas.	
<b>Criterio de aceptación</b>	Dado que voy a eliminar una acción, cuando selecciones las acciones a eliminar, Entonces cuando presione el botón eliminar ya no se mostrarán en el listado de acciones.	
<b>Tareas</b>	T1. Eliminar de la base de datos el registro de la acción eliminada. (Estimado: 2 hora) T2. Mostrar la nueva lista de acciones registradas (Estimado: 1 horas)	

#### Resumen de iteración:

No	Titulo	Estimado
1	Registrar acción	6
2	Cálculo de Costo total de compra	3
3	Eliminar acción registrada	3
Estimación Total:		12
Tiempo Real:		40
Velocidad Total:		0.3

#### Iteración 2

**Objetivo:** Permitir a el usuario observar las acciones registradas.

<b>Número</b>	<b>4</b>	<b>Estimado: 3 horas</b>
<b>Título</b>	Listado de acciones	



Historia de usuario	Como usuario, deseo visualizar una lista de mis acciones compradas para poder ver la información de cada una.
Criterio de aceptación	Dado que ingreso a la aplicación, cuando inicie, Entonces se muestra las acciones compradas con la información necesaria.
Tareas	T1. Crear una página para mostrar la lista de acciones compradas. (Estimado: 2 horas) T2. Actualizar la página al ingresar una nueva acción. (Estimado: 1 horas)

Número	5	Estimado: 2 horas
Título	información de acciones	
Historia de usuario	Como usuario, deseo poder observar el nombre de la acción, fecha de compra, precio de compra por acción, cantidades de acciones y costo total de compra.	
Criterio de aceptación	Dado que se presente el listado de acciones adquiridas, cuando el usuario mire su información, Entonces el listado de acciones tiene que mostrar el nombre, fecha de compra, precio de compra por acción, cantidades de acciones y costo total de compra.	
Tareas	T1. Codificar a la página de lista de acciones para que presente la información que se mostrara por cada acción. (Estimado: 2 hora)	

Número	6	Estimado: 4 horas
Título	Búsqueda de acciones por su nombre	
Historia de usuario	Como usuario, deseo ingresar el nombre de una acción que deseo para poderla filtrar y encontrarla más rápido.	
Criterio de aceptación	Dado que deseo una acción en específico, cuando ingrese su nombre en el buscador, entonces se mostrar la acción con ese nombre si existe.	
Tareas	T1. Agregar a la página un navBar y un espacio de texto para buscar una acción. (Estimado: 2 horas) T2. Agregar un botón en el navBar para poder presentar el listado de las acciones en modo tabla y modo cuadros. (Estimado: 2 horas)	

#### Resumen de iteración:

No	Titulo	Estimado
3	Listado de acciones	3

4	Información de acciones	2
5	Búsqueda de acciones por su nombre	4
		Estimación Total: 9
		Tiempo Real: 40
		Velocidad Total: 0.225

### Iteración 3

**Objetivo:** Permitir al usuario el modo de visualización de la app en claro o a dark.

Número	6	Estimado: 4 horas
Título	Cambiar de modo de visualización a dark.	
Historia de usuario	Como usuario, deseo poder cambiar el modo de visualización de la app a modo dark para tener dos modos de visualización de la app.	
Criterio de aceptación	Dado que la app se presenta con tema claro, cuando se presiona el botón para cambiar de modo, entonces la app se pone en modo dark.	
Tareas	T1. Agregar el botón para cambiar de modo claro a oscuro. (Estimado: 1 horas) T2. Cambiar el color del contenido de la app para que se al cambiar de modo se pueda visualizar. (Estimado: 2 horas)	

### Resumen de iteración:

No	Titulo	Estimado
3	Cambiar de modo de visualización a dark.	3
		Estimación Total: 3
		Tiempo Real: 40
		Velocidad Total: 0.075

### TDD

Se realizaron la prueba unitaria para la página principal de la aplicación:

Primer test unitario para que se renderice el programa y se muestre las acciones registradas con la información necesaria:

Caso de prueba:

Historia de usuario:

Como usuario, deseo visualizar una lista de mis acciones compradas para poder ver la información de cada una.

Criterios de aceptación:

Dado que ingreso a la aplicación, cuando inicie, Entonces se muestra las acciones compradas con la información necesaria.

```

*/
Run Test | Debug Test
it('renderiza y se muestra las acciones compradas con la informacion necesaria', async () => {
  // Mockear la función fetch para simular la respuesta de la API
  jest.spyOn(global, 'fetch').mockResolvedValue({
    ok: true,
    json: () => Promise.resolve([]),
  });

  render(<App />);

  // Verificar que algunos elementos específicos estén presentes
  expect(screen.getByText("Agregar Acción")).not.toBeNull();
  expect(screen.getByRole("button", { name: "Eliminar Acción" })).not.toBeNull();

  // Ejemplo de interacción con algún elemento
  userEvent.click(screen.getByText("Agregar Acción"));

  // Verificar que se llame a la función handlerTraerAcciones después de la interacción
  await waitFor(() => {
    expect(global.fetch).toHaveBeenCalled();
    expect(global.fetch).toHaveBeenCalledWith(
      "http://26.240.184.51:3000/api/v1/acciones",
      expect.objectContaining({ method: "GET" })
    );
  });

  global.fetch.mockRestore();
});

```

Ilustración 3. Test Mostrar acciones registradas con la información necesaria.

Segundo test unitario para que muestre toda la lista de acciones actuales:

Historia de usuario:

Como usuario, deseo poder observar el nombre de la acción, fecha de compra, precio de compra por acción, cantidades de acciones y costo total de compra.

Criterios de aceptación:

Dado que se presente el listado de acciones adquiridas, cuando el usuario mire su información, Entonces el listado de acciones tiene que mostrar el nombre, fecha de compra, precio de compra, cantidad de acciones y costo total de compra.

```

*/
Run Test | Debug Test
it('se renderiza toda la informacion de las acciones', async () => {
  // Simula datos de acciones
  jest.spyOn(global, 'fetch').mockResolvedValueOnce({
    ok: true,
    json: () => Promise.resolve([
      {
        id_accion: 1,
        siglas_accion: 'AAPL',
        fecha_compra: '2022-01-11',
        precio_compra: 150,
        cantidad_acciones: 10,
        costo_total: 1500,
      },
    ]),
  });

  // Renderiza el componente
  render(<App />);

  // Espera a que se resuelva la llamada asíncrona o realiza operaciones asíncronas necesarias

  // Realiza las verificaciones
  await waitFor(() => {
    expect(screen.queryByText('AAPL')).not.toBeNull();
    expect(screen.queryByText('2022-01-11')).not.toBeNull();
    expect(screen.queryByText('150')).not.toBeNull();
    expect(screen.queryByText('10')).not.toBeNull();
    expect(screen.queryByText('1500')).not.toBeNull();
  });

  global.fetch.mockRestore();
});

```

Ilustración 4. Test para mostrar la lista de acciones registradas actualmente.

Tercer test unitario para que se registre una acción comprada en el sistema.

```
Run Test | Debug Test
it('renderiza App y registra una accion', async () => {
  // Renderiza el componente
  render(<App />);

  // Simula hacer clic en el botón "Agregar Acción"
  fireEvent.click(screen.getByText('Agregar Acción'));

  // Espera un momento para que se muestre el formulario
  await waitFor(() => {
    expect(screen.getByText('Agregar')).not.toBeNull();
  });

  // Verifica que se presente el formulario de registro
  expect(screen.getByPlaceholderText('Nombre de la acción')).not.toBeNull();
  expect(screen.getByPlaceholderText('Fecha de compra')).not.toBeNull();
  expect(screen.getByPlaceholderText('Precio de compra')).not.toBeNull();
  expect(screen.getByPlaceholderText('Cantidad')).not.toBeNull();

  // Simula completar el formulario
  fireEvent.change(screen.getByPlaceholderText('Nombre de la acción'), { target: { value: 'PRUEBA' } });
  fireEvent.change(screen.getByPlaceholderText('Fecha de compra'), { target: { value: '2022-01-11' } });
  fireEvent.change(screen.getByPlaceholderText('Precio de compra'), { target: { value: '123' } });
  fireEvent.change(screen.getByPlaceholderText('Cantidad'), { target: { value: '6' } });

  // Simula enviar el formulario (puedes hacer clic en el botón "Guardar" si existe)
  fireEvent.click(screen.getByText('Agregar'));

  // Espera unos segundos para que se registre la acción con funcion de tiempo
  await waitFor(() => {
    expect(screen.getByText('PRUEBA')).not.toBeNull();
  }, { timeout: 5000 });
});
```

Ilustración 5. Test para registrar una acción.

A continuación, se presenta la ejecución de los tests unitarios.

```
PASS src/componentes/app.test.js
App
  ✓ renderiza y se muestra las acciones compradas con la informacion necesaria (339 ms)
  ✓ se renderiza toda la informacion de las acciones (34 ms)
  ✓ renderiza App y registra una accion (144 ms)

A worker process has failed to exit gracefully and has been force exited. This is likely caused by tests leaking due to improper teardown. Try running with --detectOpenHandles to find leaks. Active timers can also cause this, ensure that .unref() was called on them.
Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:  0 total
Time:        4.84 s
Ran all test suites related to changed files.
```

Ilustración 6. Ejecución de los tests unitarios.

Como se pudo observar en la imagen anterior los test pasaron exitosamente, por lo tanto, se puede continuar con la implementación del software.

```

console.log
[
  {
    id_accion: 52,
    siglas_accion: 'PRUEBA',
    fecha_compra: '2022-01-11',
    precio_compra: '123.00',
    cantidad_acciones: 6,
    costo_total: '738.00'
  },
  {
    id_accion: 5,
    siglas_accion: 'AAPL',
    fecha_compra: '2023-02-15',
    precio_compra: '250.00',
    cantidad_acciones: 5,
    costo_total: '1250.00'
  },
  {
    id_accion: 4,
    siglas_accion: 'TSLA',
    fecha_compra: '2023-03-20',
    precio_compra: '700.00',
    cantidad_acciones: 2,
    costo_total: '1400.00'
  },
  {
    id_accion: 3,
    siglas_accion: 'MSFT',
    fecha_compra: '2023-01-10',
    precio_compra: '150.00',
    cantidad_acciones: 10,
    costo_total: '1500.00'
  },
  {
    id_accion: 2,
    siglas_accion: 'AAPL',
    fecha_compra: '2023-02-15',
    precio_compra: '250.00',
    cantidad_acciones: 5,
    costo_total: '1250.00'
  },
  {
    id_accion: 1,
    siglas_accion: 'AMZN',
    fecha_compra: '2020-01-01',
    precio_compra: '1000.00',
    cantidad_acciones: 10,
    costo_total: '10000.00'
  }
]
at src/App.js:39:17

```

Ilustración 7. resultados de la ejecución de los test (lista de acciones actuales).

## Resultados:

### Ejecución de la aplicación:

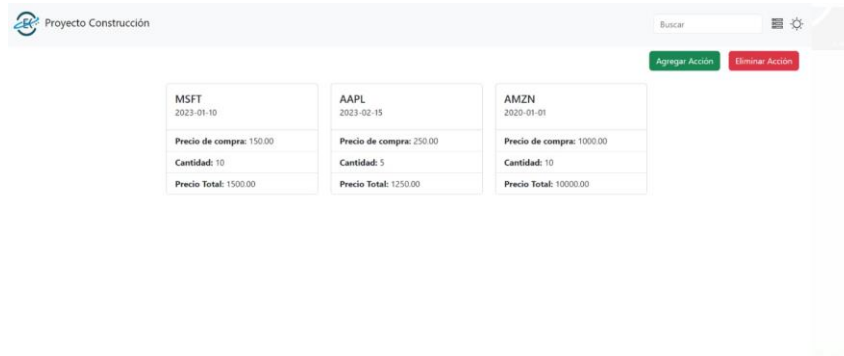


Ilustración 8. Pantalla principal de la app con la lista de acciones registradas.

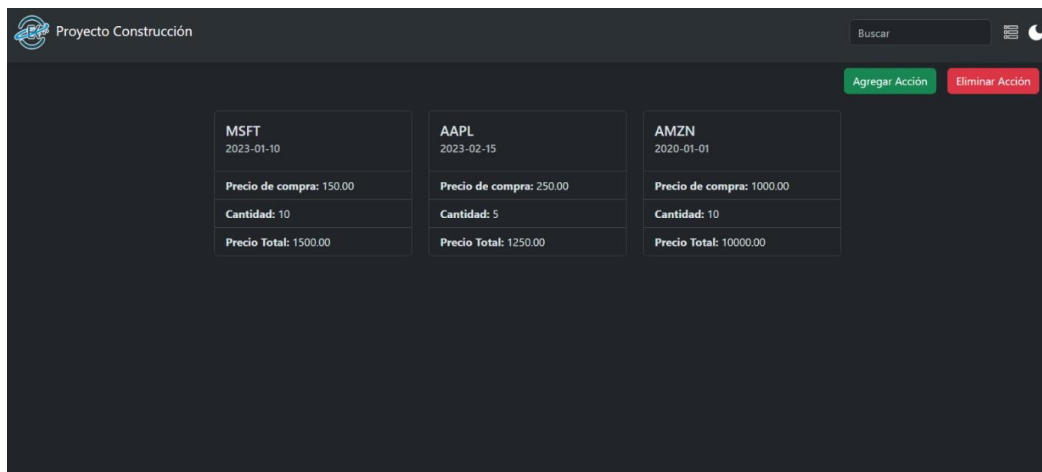


Ilustración 9. Primera pantalla modo oscuro.



Ilustración 10. Vista en modo tablas y claro.

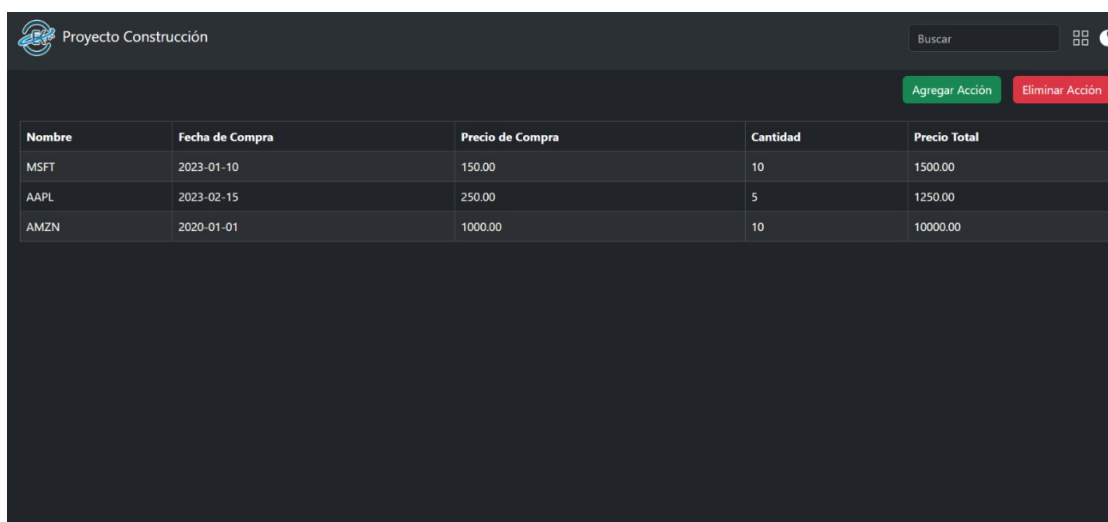


Ilustración 11. Vista en modo tablas y oscuro.

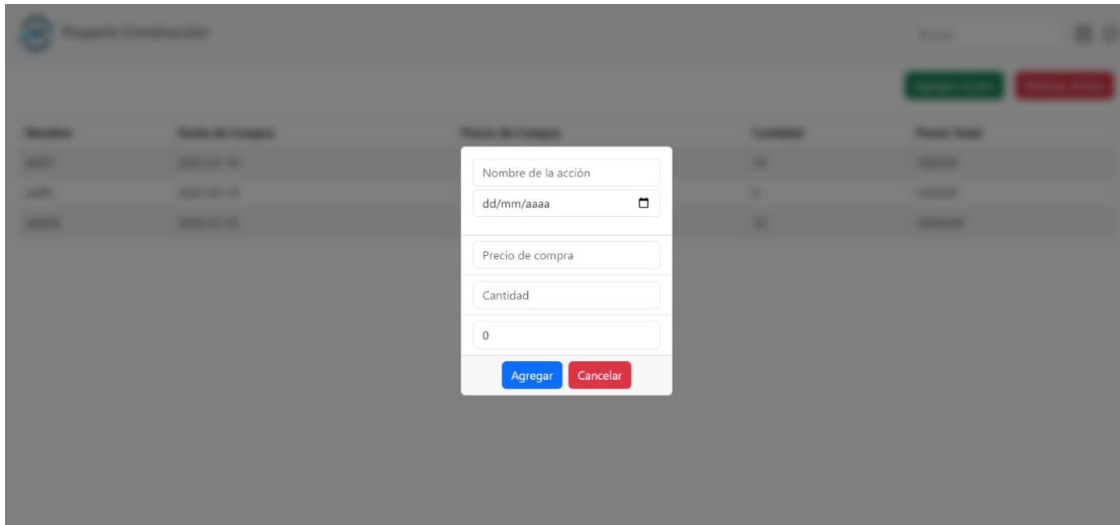


Ilustración 12. registro de una acción.

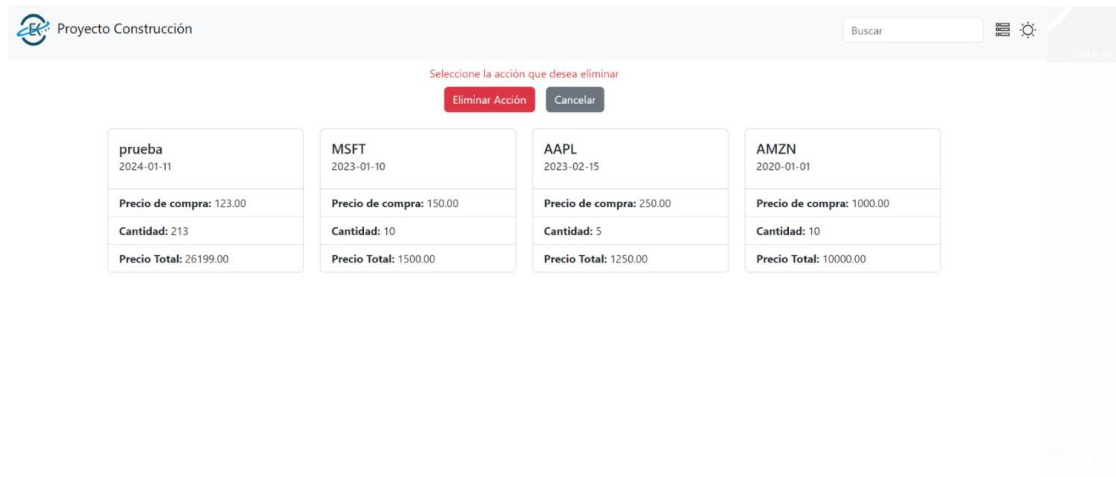


Ilustración 13. Eliminar una acción registrada.

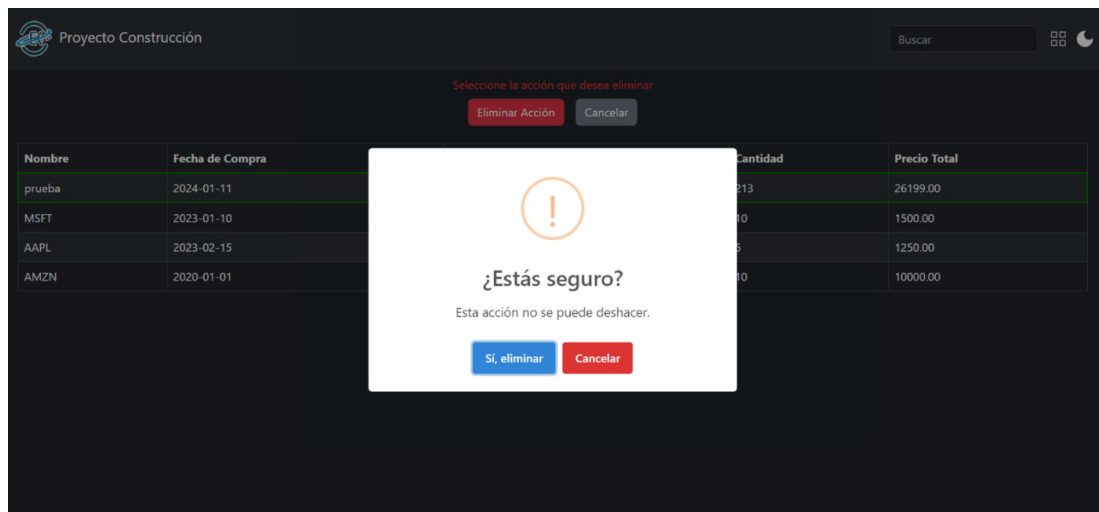


Ilustración 14. Alerta para ejecutar la función de eliminación.

## Conclusiones:

- En conclusión, la programación con paradigma en componentes ofrece varios beneficios, como la reutilización de código, el mantenimiento simplificado y la escalabilidad. Sin embargo, también impone desafíos, especialmente en términos de gestión de dependencias y versionamiento. La elección de este enfoque debe considerarse cuidadosamente en función de los requisitos del proyecto y las características específicas del sistema que se está desarrollando.
- La metodología XP puede ser una opción efectiva para el desarrollo de software en entornos que requieren flexibilidad, entrega rápida y colaboración cercana. Sin embargo, es importante evaluar cuidadosamente su idoneidad para el contexto y los requisitos específicos del proyecto antes de adoptarlo.

## Referencias:

- [1] «¿Qué es React?», MVP Cluster. Accedido: 8 de enero de 2024. [En línea]. Disponible en: <https://mvpcluster.com/react/>
- [2] *Diagrama de componentes*, (24 de septiembre de 2020). Accedido: 8 de enero de 2024. [En línea Video]. Disponible en: [https://www.youtube.com/watch?v=wSf\\_JGyP0Y8](https://www.youtube.com/watch?v=wSf_JGyP0Y8)