

Análisis de Cambio entre Versiones

El presente documento tiene como objetivo realizar un análisis comparativo entre dos versiones de un producto de software, identificando los cambios realizados en los artefactos y describiendo en detalle las modificaciones efectuadas. El producto de software en cuestión es un sistema de registro de compras de acciones, diseñado para facilitar el seguimiento y la gestión de transacciones financieras relacionadas con la compra de acciones en el mercado de valores.

Artefactos

1. Estructura HTML y CSS:

- En la versión antigua, se utiliza una estructura HTML más simple con elementos `<table>`, `<thead>`, `<tbody>`, y `<input>` para la entrada de datos.
- En la versión actual, se ha adoptado un enfoque más moderno y semántico utilizando etiquetas `<form>`, `<label>`, y `<input>`, junto con estilos CSS actualizados.

2. Funcionalidad JavaScript:

- En la versión antigua, el JavaScript se utiliza para agregar y eliminar filas de la tabla dinámicamente, así como para calcular el costo total de las acciones.
- En la versión actual, se ha reescrito la funcionalidad JavaScript para integrar la comunicación con el servidor a través de Fetch API, permitiendo la persistencia de datos en una base de datos MySQL.
- Se ha introducido una función `guardarAccion()` para enviar datos de acciones al servidor y actualizar la tabla de acciones después de guardar una nueva acción.

3. Servidor Express y Base de Datos:

- En la versión actual, se ha creado un servidor Express para manejar las solicitudes HTTP y las rutas relacionadas con las acciones de compra de acciones.
- Se ha integrado una base de datos MySQL para almacenar los datos de las acciones, lo que proporciona persistencia de datos entre sesiones de la aplicación.
- Se ha introducido un endpoint `/guardar-accion` para recibir los datos de las acciones del cliente y almacenarlos en la base de datos.
- Se ha añadido un endpoint `/acciones` para obtener todas las acciones almacenadas en la base de datos y mostrarlas en la tabla de la interfaz de usuario.

4. Integración de API para Obtener Precio de Acción en Tiempo Real:

- Se ha añadido una nueva funcionalidad para obtener el precio de la acción en tiempo real a partir del símbolo de la acción ingresado.
- Esta funcionalidad se ha implementado mediante una API externa que proporciona información actualizada sobre el valor de las acciones.

5. Configuración de Docker:

- Se ha introducido un archivo **docker-compose.yml** para orquestar los servicios de la aplicación web y la base de datos MySQL.
- Se ha creado un archivo **Dockerfile** para construir la imagen de la aplicación Node.js y exponer el puerto 3000.

Cambios Realizados

1. Estructura HTML y CSS:

- Se ha adoptado un enfoque más moderno y semántico en la versión actual, utilizando etiquetas HTML como `<form>` y `<label>` para mejorar la accesibilidad y la legibilidad del código.
- Los estilos CSS se han actualizado para mejorar la apariencia visual de la aplicación, utilizando colores más llamativos y un diseño más limpio.

2. Funcionalidad JavaScript:

- Se ha reescrito la funcionalidad JavaScript para integrar la comunicación con el servidor a través de Fetch API, permitiendo la persistencia de datos en una base de datos.
- Se ha introducido una función **guardarAcción()** para enviar datos de acciones al servidor y actualizar la tabla de acciones después de guardar una nueva acción.
- Se ha eliminado la funcionalidad de agregar y eliminar filas de la tabla dinámicamente, ya que ahora las acciones se agregan a través del formulario y se muestran automáticamente en la tabla.

3. Servidor Express y Base de Datos:

- Se ha creado un servidor Express para manejar las solicitudes HTTP y las rutas relacionadas con las acciones de compra de acciones.
- Se ha integrado una base de datos MySQL para almacenar los datos de las acciones, lo que proporciona persistencia de datos entre sesiones de la aplicación.

4. Integración de API para Obtener Precio de Acción en Tiempo Real:

- Se ha añadido una nueva funcionalidad para obtener el precio de la acción en tiempo real a partir del símbolo de la acción ingresado.
- Esta funcionalidad se ha implementado mediante una API externa que proporciona información actualizada sobre el valor de las acciones.
- Se ha creado una función en el archivo script.js para enviar una solicitud HTTP a la API y obtener el precio de la acción en tiempo real.

5. Configuración de Docker:

- Se ha introducido Docker para facilitar la implementación y el despliegue de la aplicación, utilizando un archivo `docker-compose.yml` para definir los servicios y un `Dockerfile` para construir la imagen de la aplicación.

Estos son los principales cambios entre las dos versiones del producto de software. La versión actual presenta una arquitectura más robusta y escalable, con una mejor experiencia de usuario y la capacidad de almacenar datos de manera persistente en una base de datos.