

Challenge Acciones (Stocks)

Origin

Ejercicio

El cliente necesita que se le desarrolle una interfaz web para graficar la cotización de una acción en tiempo real. El detalle del proyecto se describe a continuación.

Especificaciones

- El sitio cuenta con una página de login, donde se ingresa el usuario y password. En caso de ingresar algún dato invalido informar "usuario o clave invalida"

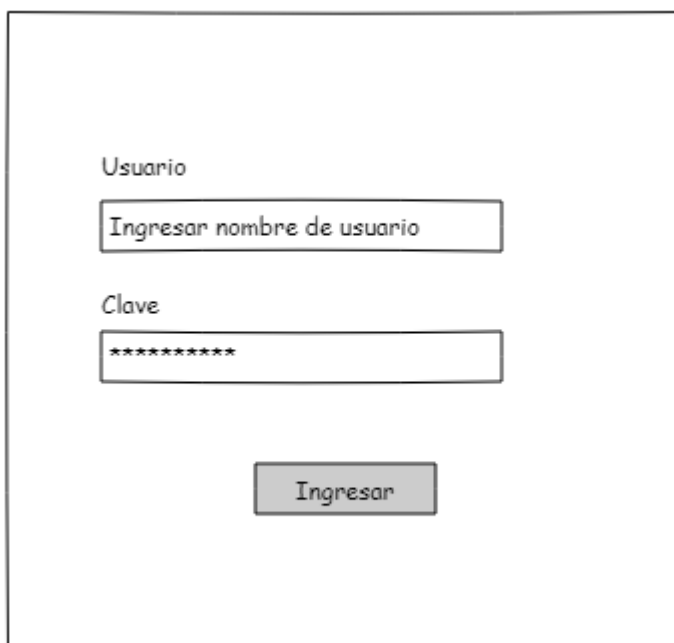


Diagrama de la interfaz de login:

- Etiqueta: Usuario
- Input: Ingresar nombre de usuario
- Etiqueta: Clave
- Input: *****
- Botón: Ingresar

- Luego de ingresar al sitio, nos redirecciona a la página de "Mis Acciones", esta página va a mostrar las acciones preferidas que haya seleccionado cada usuario. En la cabecera de la página, se debe visualizar el

nombre del usuario logueado. Para agregar una acción, hay que escribir en el "Autocomplete" para que muestre las acciones que coinciden con el texto buscado, luego seleccionar una y hacer click en "Agregar símbolo", esto agrega la acción a mis acciones preferidas y luego se refresca la grilla con los datos correspondientes. De cada acción se debe guardar el símbolo, nombre y moneda. Para eliminar un símbolo, hay que hacer click en el link de "Eliminar" en la fila correspondiente. Esto borra la acción de las acciones preferidas del usuario y luego se refresca la grilla.

Mis Acciones

Usuario: Juan

Símbolo

(Autocomplete) ▼

Agregar Símbolo

Símbolo	Nombre	Moneda	
TSLA	Tesla Inc	USD	Eliminar
AAPL	Apple Inc	USD	Eliminar
NFLX	Netflix Inc	USD	Eliminar

Al hacer click en el símbolo (ejemplo "TSLA"), nos llevara a la siguiente página como se describe a continuación.

- **Página de Detalle de Acción:** esta página, nos mostrará en la cabecera los datos de la acción seleccionada. Luego podremos seleccionar los parámetros para graficar la cotización de la acción.

Parámetro Tiempo real: Al seleccionar esta opción, vamos a tener que graficar la cotización en base a la fecha del día. Al graficar esta opción, el gráfico debe actualizarse en forma automática según el intervalo seleccionado (cada 1min/5min/15min, según corresponda). Es decir, luego de por ejemplo 1 minuto, se debería actualizar solo el gráfico.

Parámetro Histórico: Al seleccionar esta opción vamos a tener que graficar la cotización en base a los parámetros fecha y hora desde – fecha y hora hasta ingresados.

Parámetro Intervalo: Este parámetro define el intervalo de tiempo para graficar la cotización, los valores posibles son 1min / 5min/ 15min

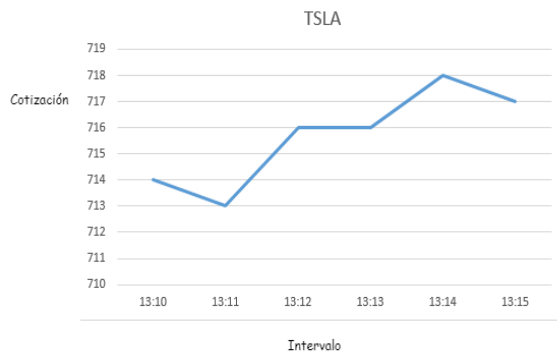
Para realizar el grafico se puede utilizar la librería de <https://www.highcharts.com/> o similar

☒ Tiempo Real (utiliza la fecha actual, al graficar esta opción, se debe actualizar el gráfico en forma automática según el intervalo seleccionado)

☐ Histórico Fecha hora desde Fecha hora hasta

Intervalo (opciones 1min / 5min / 15min)

Graficar (utilizar highcharts o similar para graficar)



API para obtener datos

1. Para obtener los datos de las acciones utilizar la siguiente API <https://api.twelvedata.com/> (es gratis, permite 800 request por día)
2. Para obtener los datos de una acción consultar de la siguiente forma (ejemplo para Netflix, NFLX) <https://api.twelvedata.com/stocks?symbol=NFLX&source=docs> . Para obtener un listado de acciones para llenar el Autocomplete se puede utilizar la siguiente llamada <https://api.twelvedata.com/stocks?source=docs&exchange=NYSE>
3. Para obtener la cotización de una acción consultar de la siguiente forma https://api.twelvedata.com/time_series?symbol=TSLA&interval=5min&start_date=2021-04-16%2009:48:00&end_date=2021-04-16%2019:48:00&apikey=*****

los parámetros start_date y end_date son opcionales, en el parámetro apikey colocar la apikey obtenida al crear una cuenta en <https://api.twelvedata.com/>

Tarea

1. Crear la base de datos y todos los objetos que creas necesarios para que la aplicación funcione.
2. Insertar una cantidad mínima de datos para poder probar la aplicación.
3. Desarrollar una aplicación web, en un repositorio público, utilizando una herramienta de versionado como github/bitbucket/etc, que responda a los requisitos descritos por el cliente.
4. Incluir un backup de la base de datos.
5. Incluir en el repositorio un readme.md con los pasos para levantar la aplicación.

Notas Técnicas

La arquitectura de la aplicación tiene que contener dos proyectos un Frontend y una API.

- Base de datos:
 - Relacional (Sql Server 2017, Mysql 8.0.15, PostgreSQL 13.2)

El stack tecnológico deberá ser

- Para el frontend (elegir una opción):
 - Angular ≥ 9
 - React ≥ 16.8 utilizando hooks
- Para la API (elegir una opción):
 - .Net Core ≥ 3.1
 - Node.js $\geq 14.16.1$
- Para acceso de base de datos:
 - Entity Framework (.net) o Express (Node)

No se evaluará el diseño ni el conocimiento sobre UI que posea el developer, sino la funcionalidad y la estabilidad de la solución, el modelo de datos, la atención a los requerimientos, seguridad, estructura del código y la actitud frente a los NFR (non-functional requirements), que son: mantenibilidad, extensibilidad y escalabilidad

