

## **Proyecto 2**

### **Segunda Fase**

### **Requisitos funcionales.**

Nuestro proyecto consiste en ser una ayuda para aquellas personas que busquen tener una mejor noción del funcionamiento de la bolsa de valores y de su buena implementación para toda toma de decisiones que sean necesarias. El programa será capaz de saber interpretar las gráficas y podrá sugerirle al usuario que decisión es considerada mejor con respecto a las ganancias. Para que esto sea funcional es necesario que

- Capacidad de llamado de Http Requests: enviar solicitudes a servidores web y recibir respuestas de ellos.
- Interfaz gráfica en web (React siendo el framework): usuarios pueden ver y con la que pueden interactuar visualmente.
- Capacidad de Obtener o buscar símbolos y tener los prompts para el AI: interactuar con el modelo AI para hacer preguntas y solicitar información.
- Mostrar gráfica de Stocks: representación visual del precio y el rendimiento de una acción.
- Mostrar mensajes del AI AKA Advisor: acción de mostrar los mensajes generados por un asistente de inteligencia artificial.

### **Priorización de los requisitos funcionales.**

- Funciones de primera prioridad:
  - Se necesitará que las API para todo lo que tenga que ver con los stocks sean llamadas y puedan ser llamadas
  - Que haya una forma de buscar los diferentes símbolos de los diferentes mercados.
  - Tener la posibilidad de mandar la información requerida a la AI

### **Identificación y descripción de clases.**

- Controlador:
  - Buscador de Símbolos
    - El propósito principal de esta clase es darle la capacidad al usuario de poder ingresar y pedir el símbolo que el usuario necesite.
      - AskApiData: El propósito de este método es llamar al API para que le de la información de todos los símbolos disponibles en

Pedro Avila # 23089

Javier Cifuentes #

Javier Benitez # 23405

Ernesto Ascencio #23009

---

- Alpha Vantage. Necesita el atributo de símbolo que lo pone el siguiente método
    - Select Symbol: Su propósito es por medio de Hooks de react se agarre el símbolo que el usuario asigna como el símbolo de toda la clase. Necesita el input que se pediría en JSX
    - Add Graph: es más que todo un setState para poder mostrar la gráfica o no poder mostrar la gráfica.
    - Render: Muestra el JSX.
  - Chat (Parte de inputs)
    - Su propósito es darle la habilidad al usuario de poder hablar con el AI por medio de los prompts que el mismo usuario quiera.
      - OpenAIPrompt: La información que es puesta en el JSX se usa para mandarla al AI.
      - ComponentMount: Técnicamente que hará al momento de instanciarse.
- Vista:
  - Gráfica
    - Principalmente su propósito es mostrar la información de los “stocks” de una manera más comprensible para el usuario, además de solo “raw data”.
      - AskForData: Le pide a la API la información de los stocks.
      - ComponentMount: Que hará al momento de instanciarse.
      - render: Mostrar el JSX
  - Chat (Respuesta de AI)
    - Aquí el usuario podrá ver las respuestas que la inteligencia artificial le devuelva
      - ReadStockData: hará que el AI llame por sí mismo a la API para ver la información del stock.
      - Render: Muestra el JSX
- Modelo:
  - API: Su propósito es poder dar toda la información requerida de las diferentes API de Alpha Vantage y Open AI de una forma más limpia y directa
    - SearchSymbol: Llamara a Información de stocks AKA Stocks para abreviar, el cual le dará la información de los Símbolos mencionados anteriormente
    - StockData: Llamara a la Stock para que le de toda la información de stocks de parte de Alpha Vantage
    - DelStock: Llamara a la función Del Stock en Stocks para borrar la instancia del objeto
    - AskAi: Llamara a AI para que le de la información de parte de chat gpt

Pedro Avila # 23089

Javier Cifuentes #

Javier Benitez # 23405

Ernesto Ascencio #23009

---

- CheckStockData: Llamara la función de AI para que revise los Stocks
  - AI
    - CheckStocks: pondrá la información de los stocks en el chat log de la conversación.
    - Ask: Llamará a la API de OPENAI para hacerle cualquier pregunta o conversación.
  - Información de stocks
    - Innit: el constructor, el cual solo necesitara el simbolo dado desde el controlador
    - URL: vera que tipo de request se necesita dependiendo la funcion, si es de Symbol data o de stock Data
    - SymbolSearcher: llamara a AlphaVantage para la informacion de los simbolos
    - Stock: Llamara a AlphaVantage para pedir informacion de los stocks.
    - delStock: Borrara la instancia de la clase

### **Atributos clase Controlador:**

- Symbol: Atributo para almacenar el símbolo que el usuario ingresa.
- Prompt: Atributo para almacenar el mensaje o pregunta que el usuario ingresa.
- Data: Atributo para almacenar datos relacionados con acciones.
- ChatgptPrompt: Atributo para almacenar la respuesta de la inteligencia artificial.

### **Atributos clase Vista (Gráfica):**

- Symbol: Atributo para almacenar el símbolo de acción relacionado con la gráfica.
- Data: Atributo para almacenar datos relacionados con acciones.

### **Atributos clase Información de stocks:**

- Symbol: Atributo para almacenar el símbolo de acción.
- URL: Atributo para determinar el tipo de solicitud que se debe realizar a la API.
- data: No se mencionó explícitamente en la descripción, pero se asume que podría ser un atributo para almacenar los datos relacionados con acciones.

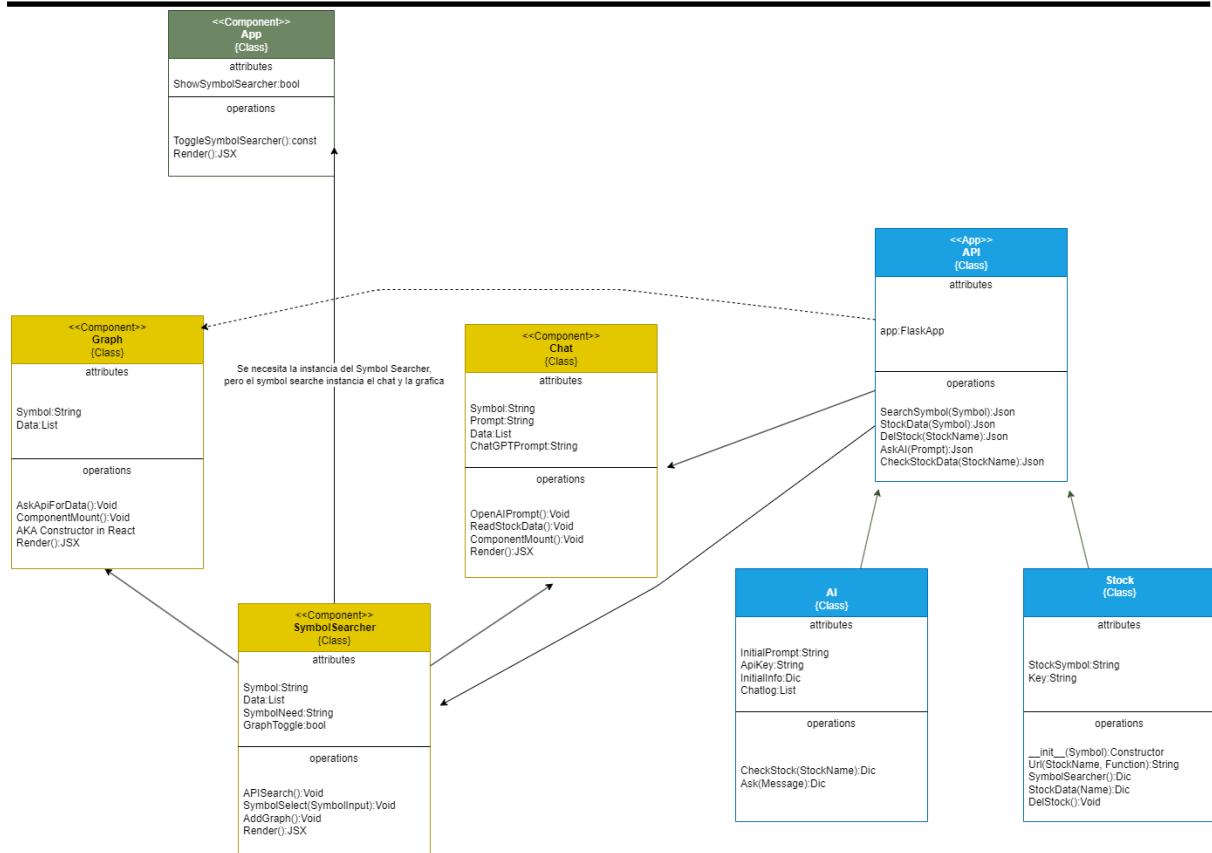
UML:

Pedro Avila # 23089

Javier Cifuentes #

Javier Benitez # 23405

Ernesto Ascencio #23009



Pedro Avila # 23089

Javier Cifuentes #

Javier Benitez # 23405

Ernesto Ascencio #23009

---

Se investigaron cinco plataformas que plantean una idea similar. Para estas se encontró el lenguaje en el que se programó, cómo se manejó su base de datos, qué se programó en el frontend y qué framework se utilizó:

1. **Investopedia:**

- **Lenguaje de Programación:** Investopedia está programada principalmente en PHP.
- **Base de Datos:** Utilizan MySQL para gestionar su base de datos.
- **Tecnologías Frontend:** Utilizan HTML, CSS y JavaScript para la interfaz de usuario.
- **Framework:** Utilizan su propio framework personalizado.

2. **Wall Street Survivor:**

- **Lenguaje de Programación:** Wall Street Survivor utiliza principalmente Ruby on Rails para el desarrollo web.
- **Base de Datos:** Utilizan PostgreSQL para la gestión de datos.
- **Tecnologías Frontend:** HTML, CSS, JavaScript, jQuery.
- **Framework:** Ruby on Rails es el framework principal utilizado.

3. **StockTrak:**

- **Lenguaje de Programación:** StockTrak utiliza una combinación de tecnologías, incluyendo PHP, Java y C#.
- **Base de Datos:** Utilizan una variedad de sistemas de gestión de bases de datos, como MySQL y SQL Server.
- **Tecnologías Frontend:** HTML, CSS, JavaScript.
- **Framework:** Utilizan múltiples frameworks y tecnologías según la parte de la plataforma.

4. **Morningstar Classroom:**

- **Lenguaje de Programación:** No se proporciona información específica sobre el lenguaje de programación utilizado, pero generalmente utilizan tecnologías web estándar como PHP o Java.
- **Base de Datos:** Probablemente utilizan bases de datos SQL estándar.
- **Tecnologías Frontend:** HTML, CSS, JavaScript.
- **Framework:** Es posible que utilicen un framework de desarrollo web.

5. **The Motley Fool:**

- **Lenguaje de Programación:** El sitio web de The Motley Fool está programado principalmente en Java y JavaScript.
- **Base de Datos:** Utilizan sistemas de bases de datos SQL.
- **Tecnologías Frontend:** HTML, CSS, JavaScript.
- **Framework:** Es posible que utilicen un framework de desarrollo web en Java.

GitHub Repo:

<https://github.com/PeDro0210/Ai-Stock-Advaisor>

Pedro Avila # 23089

Javier Cifuentes #

Javier Benitez # 23405

Ernesto Ascencio #23009

Formulario de Integrantes:

Nombre: Pedro Avila

Carné: 23089

Fecha	Inicio	Fin	Tiempo Interrupción (min)	Delta tiempo (min)	Tarea	Comentarios
1 de Sep	5:00 PM	8:30 PM	30	180	Planificar Grafica de Stocks	Intentarlo hacerlo lo mas rapido posible
4 de Sep	6:00 PM	8:00 PM	20	100	Planificar Chat para Advisor	Intentarlo hacerlo lo mas rapido posible
11 de Sep	6:45 PM	7:20 PM	20	115	Hacer Chat e Implementarlo con Symbol Searcher	Intentarlo hacerlo lo mas rapido posible
16 de Sep	8:00 PM	4:30 PM	30	240	Implementar Grafica con el resto	Intentarlo hacerlo lo mas rapido posible

Nombre: Ernesto Ascencio

Carné: 23009

Fecha	Inicio	Fin	Tiempo Interrupción (min)	Delta tiempo (min)	Tarea	Comentarios
1 de Sep	5:00 PM	8:30 PM	30	180	Investigación de las herramientas a utilizar.	Session larga para iniciar y pausa para cenar
4 de Sep	6:00 PM	8:00 PM	20	100	Definición de herramientas y inicio de codificación	Session regular con pausa para comer y descansar
11 de Sep	6:45 PM	7:20 PM	20	115	Utilizacion de React y Line Component para mostrar graficas	Semana de independencia, no hubo mucho avance
16 de Sep	8:00 PM	4:30 PM	30	240	Seguimiento de codificación (aún no se ha concluido)	Aun sigue conclusa

Pedro Avila # 23089

Javier Cifuentes #

Javier Benitez # 23405

Ernesto Ascencio #23009

Nombre: Javier Benitez

Carné: 23405

Fecha	Inicio	Fin	Tiempo Interrupción (min)	Delta tiempo (min)	Tarea	Comentarios
1 de Sep	5:00 PM	6:00 PM	10	40	Planificación del diseño y estilos del front end del app	La planificación quedó terminada y se comenzó a desarrollar un poco, pausa para distraerme y relajarme un poco
4 de Sep	6:00 PM	8:00 PM	20	100	Solucionar complicaciones en el desarrollo del diseño debido a conflictos en github, pero solucionados al final	Session regular con pausa para comer y descansar
11 de Sep	5:00 PM	10:00 PM	40	260	Diseño terminado y estilo aplicado al front end de la aplicación	Una larga sesión aprovechando el tiempo libre por la semana de independencia, pausa para comer y distraerme
16 de Sep	1:00 PM	4:30 PM	30	290	Debuggear la aplicación y mejorar detalles	Session larga para terminar detalles cortos.

Nombre: Javier Cifuentes

Carné:

Fecha	Inicio	Fin	Tiempo Interrupción (min)	Delta tiempo (min)	Tarea	Comentarios
1 de Sep	5:00 PM	8:30 PM	30	190	Investigar las plataformas y que herramientas usaron	hecho, se investigó 5 plataformas
4 de Sep	6:00 PM	8:00 PM	20	100	Realizar mi Parte del UML.	hecho, se terminó y se entregó de manera adecuada
11 de Sep	6:45 PM	7:20 PM	20	115	Ayudara a la programación de la interfaz gráfica en react	Por hacer
16 de Sep	1:00 PM	4:30 PM	30	290	Verificar la compatibilidad del código clases y su funcionalidad	Por hacer