
APLICACIÓN PARA ANÁLISIS DE INVERSIÓN

APPLICATION FOR INVESTMENT ANALYSIS



TRABAJO DE FIN DE GRADO

CURSO 2019-2020

AUTOR

Zhaoyan Ni

DIRECTOR

Adrián Riesco Rodríguez

Doble grado en Ingeniería Informática-Matemáticas
Facultad de Informática
Universidad Complutense de Madrid

ÍNDICE

APLICACIÓN PARA ANÁLISIS DE INVERSIÓN	1
ÍNDICE DE FIGURAS.....	4
PREFACIO	5
RESUMEN	6
PALABRA CLAVE	6
SUMMARY.....	7
KEYWORDS.....	7
SECCIÓN 1	8
INTRODUCCIÓN	8
1. Motivación	8
2. Objetivos.....	8
3. Estructura de la memoria.....	9
SECTION 1.....	10
INTRODUCTION.....	10
1. Motivation	10
2. Objectives.....	10
3. Memory structure.....	11
SECCIÓN 2	12
TECNOLOGÍAS USADAS.....	12
1. AngularJs	12
1.1 Arquitectura de aplicaciones de AngularJs.....	13
2. Visual Studio Code.....	14
3. Servicios Web.....	15
4. Otras tecnologías.....	19
SECCIÓN 3	20
FUNCIONALIDADES Y DISEÑO.....	20
1. Funcionalidades	20

2. Diseño.....	21
SECCIÓN 4.....	25
IMPLEMENTACIÓN	25
1. Arquitectura	25
2. Base de datos.....	29
3. Pantallas	31
4. Limitaciones	38
SECCIÓN 5	39
CONCLUSIONES Y TRABAJO FUTURO	39
1. Conclusiones	39
2. Trabajo futuro	40
SECTION 5.....	41
CONCLUSIONS AND FUTURE WORK.....	41
1. Conclusions	41
2. Future work.....	42
BIBLIOGRAFÍA	43

ÍNDICE DE FIGURAS

Figura 1: Componentes de servicios web	15
Figura 2: Operaciones de servicios web RESTful.....	17
Figura 3: Ejemplo de respuesta devuelta por Alpha Vantage	18
Figura 4: Home	21
Figura 5: Resultado de búsqueda.....	22
Figura 6: Página de log in.....	23
Figura 7: Área privada de un usuario.....	24
Figura 8: Arquitectura nivel 1	25
Figura 9: Arquitectura de la carpeta script	26
Figura 10: Arquitectura de la carpeta usuario	26
Figura 11: Arquitectura de la carpeta endpoint	27
Figura 12: Arquitectura de la carpeta components	28
Figura 13: Diagrama ER	29
Figura 14: Pantalla Home	31
Figura 15: Búsqueda de empresas.....	31
Figura 16: Ejemplo de filtro	32
Figura 17: Ejemplo de resultado	32
Figura 18: Elección entre Login y Register	33
Figura 19: Pantalla de Login	33
Figura 20: Pantalla de Register	34
Figura 21: Tabla de acciones	35
Figura 22: Total profit	35
Figura 23: Componente gráfico 1	36
Figura 24: Componente gráfico 2	36
Figura 25: Pantalla de Log Out.....	37
Figura 26: Versiones soportados	38

PREFACIO

Antes de empezar, me gustaría agradecer a aquellas personas que me apoyaron durante la realización de este trabajo.

En primer lugar, gracias a mi familia, sin sus apoyos y esfuerzos, no tendría la oportunidad de escribir este documento.

En segundo lugar, me gustaría agradecer a mi tutor, Adrián Riesco Rodríguez, por su esfuerzo que ha dedicado a este trabajo. Él siempre está ahí cuando le necesito, ya sea para proporcionarme los recursos necesarios, para declarar mis dudas o darme consejos.

Por último, gracias a mis amigos y compañeros que me animaron cuando pensaba que no podré terminar este trabajo.

Muchas gracias a todos ellos.

RESUMEN

Como es bien sabido, en el mundo de economía, la relación entre beneficio y el riesgo es directa, es decir, todas las inversiones están sujetas a la posibilidad de que la inversión inicial no se pueda recuperar. Sin embargo, las buenas decisiones de inversión pueden traer beneficios importantes. Por esta razón, es importante que los inversores analicen las inversiones antes de invertir en ellas para sufrir menos riesgo e intentar asegurar su beneficio.

Para facilitar que los inversores tomen buenas decisiones a la hora de comprar o vender acciones, se creó este proyecto que permite a los usuarios obtener en tiempo real las variaciones del valor de las acciones de las empresas. Para que esta aplicación pudiera ayudar a más usuarios con la mayor sencillez a la hora de obtener los datos, se implementa en versión web. Así, los usuarios pueden acceder a la aplicación y obtener datos interesantes siempre y cuando dispongan de un dispositivo móvil u ordenador con navegador y acceso a internet.

Para mejorar la visualización de los datos financieros, la aplicación los representa gráficamente. Además, dispone de un área privada para cada usuario registrado con el fin de visualizar los datos financieros de las empresas que le interesa con más rapidez y sencillez.

PALABRA CLAVE

Aplicación web, servicios web, accesibilidad, datos financieros, gráficas, acciones de empresas.

SUMMARY

As is well known, in the world of economics, the relationship between benefit and risk is direct, that is, all investments are subject to the possibility that the initial investment cannot be recovered. However, good investment decisions can bring significant benefits. For this reason, it is important that investors analyze investments before investing in them to suffer less risk and try to ensure their return.

To make it easier for investors to make good decisions when buying or selling shares, this project was created that allows users to obtain real-time variations in the value of company shares. So that this application could help more users with the greatest simplicity when obtaining data, it is implemented in a web version. Thus, users can access the application and obtain interesting data if they have a mobile device or computer with browsing and internet access.

To improve the visualization of financial data, the application represents them graphically. In addition, it has a private area for each registered user in order to view the financial data of the companies that interests them more quickly and easily.

KEYWORDS

Web application, web services, accessibility, financial data, graphs, company stocks.

SECCIÓN 1

INTRODUCCIÓN

En esta sección explicaremos la motivación de realizar presente trabajo y los objetivos que se pretenden cumplir. También listaremos las secciones que componen esta memoria.

1. Motivación

Una inversión es un desembolso de dinero y/o recurso a quien los necesitan a cambio de un beneficio de cualquier tipo en el futuro, a corto o a largo plazo dependiendo de interés de inversor. Por tanto, antes de hacer las inversiones, tenemos que analizarla para conocer sobre la conveniencia de realizarla o no.

Cada vez hay más gente que invierte en Bolsa porque cree que a largo plazo la Bolsa es la inversión más rentable que hay pues junto con las posibles plusvalías obtenidas por la revalorización de las acciones, están los dividendos: parte del beneficio social que se reparte entre los accionistas.

Sin embargo, invertir en la Bolsa es una decisión importante. Previamente hay que estudiar el mercado de inversión y los valores en los que se encuentra la Bolsa. Por tanto, la motivación de este proyecto es facilitar la toma de decisiones a los usuarios a la hora de comprar o vender las acciones.

2. Objetivos

El objetivo principal de este trabajo es permitir desarrollar una aplicación web que facilite a los usuarios tomar decisiones de inversiones. Para ello, permitirá a los usuarios visualizar en tiempo real las variaciones del valor de las acciones de las empresas mediante gráficas. La aplicación proporcionará filtros de búsqueda para que los usuarios puedan visualizar los datos financieros que les interesa con mayor eficacia y sencillez. Unos ejemplos de los filtros pueden ser el intervalo de tiempo entre dos datos y el tipo de datos financieros que le interesa. También cuenta con un área privada para los usuarios registrados para ofrecer servicios más personalizados.

3. Estructura de la memoria

Con el fin de que la memoria tenga estructura y organización coherente, presente documento está organizado en secciones.

- Sección 1: en esta sección están redactada tanto en español como en inglés las motivaciones de hacer presente proyecto y los objetivos que pretendemos alcanzar. También está redactada la estructura de la memoria.
- Sección 2: en esta sección se enumeran y explican brevemente las tecnologías usadas en la implementación del proyecto para que la aplicación cumpla los objetivos propuestos.
- Sección 3: en esta sección explicaremos las funcionalidades proporcionadas por la aplicación. También mostraremos el diseño de interfaces mediante *mockup*.
- Sección 4: en esta sección explicaremos la implementación de la aplicación. También mostraremos el funcionamiento de la aplicación mediante figuras y ejemplos.
- Sección 5: en esta sección concluimos presente trabajo y enumeramos las ideas para mejorar nuestra aplicación. Esta sección está redactada tanto en español como en inglés.

A continuación, se adjunta el repositorio del código :

<https://github.com/ZHAOYANNI/TFG-INFORMATIC>

El uso del código está bajo licencia MIT.

SECTION 1

INTRODUCTION

In this section we will expose the motivation for carrying out this present work and the objectives that are intended to be met. We will list the sections that make up this report too.

1. Motivation

An investment is a disbursement of money and / or resources to those who need them in exchange for a benefit of any kind in the future, in the short or long term depending on the investor's interest. Therefore, before making the investments, we have to analyze it to know about the convenience of making it or not.

There are more and more people who invest in the Stock Market because they believe that in the long term the Stock Market is the most profitable investment that there is because together with the possible capital gains obtained by the revaluation of the shares, there are the dividends: part of the social benefit that is distributed between shareholders.

However, investing in the Stock Market is an important decision. Previously, the investment market and the securities in which the Stock Market is located must be studied. Therefore, the motivation of this project is to facilitate decision-making for users when buying or selling the shares.

2. Objectives

The main objective of this work is to allow the development of a web application that facilitates users to make investment decisions. To do this, it will allow users to view in real time the variations in the value of company shares through graphs. The application will provide search filters so that users can view the financial data that interests them more efficiently and easily. Some examples of filters can be the time interval between two data and the type of financial data that interests you. It also has a private area for registered users to offer more personalized services.

3. Memory structure

For the report to have a coherent structure and organization, this document is organized into sections.

- Section 1: in this section the motivations for doing this project and the objectives we intend to achieve are written in both Spanish and English. The memory structure is also written.
- Section 2: in this section the technologies used in the implementation of the project are listed and briefly explained so that the application meets the proposed objectives.
- Section 3: in this section we will explain the functionalities provided by the application. We will also show the interface design through a *mockup*.
- Section 4: in this section we will explain the implementation of the application. We will also show the operation of the application through figures and examples.
- Section 5: in this section we conclude this work and list the ideas to improve our application. This section is written in both Spanish and English.

Next, the code repository is attached:

<https://github.com/ZHAOYANNI/TFG-INFORMATIC>

The use of the code is under the MIT license.

SECCIÓN 2

TECNOLOGÍAS USADAS

Vamos a explicar brevemente las tecnologías usadas para el desarrollo y la implementación de este trabajo en esta sección, exponiendo sus características principales.

1. AngularJs

AngularJs^{[1][2][3]} es un *framework* de *JavaScript* de código abierto desarrollado por Google. Se usa para la creación y el mantenimiento de aplicaciones web dinámicas, modernas de una sola página (SPA). Usa HTML como su plantilla principal y le permite extender la sintaxis de HTML con directivas y expresiones para conseguir un código más clara y sucinta.

AngularJs simplifica el desarrollo de aplicaciones al proporcionar a los desarrolladores un mayor nivel de abstracción a expensas de la flexibilidad para facilitar al desarrollo a la hora de hacer pruebas de las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC).

Las directivas de AngularJs son atributos de HTML con el prefijo *ng*. Los desarrolladores pueden crear sus propias directivas para facilitar la implementación de aplicaciones y proporcionar más funcionalidades. Unos ejemplos de las directivas pueden ser:

- La directiva *ng-app*: define el elemento raíz de una aplicación, guiará automáticamente (autoinicializará) la aplicación cuando se cargue la página web.
- La directiva *ng-controller*: define los controladores de la aplicación.

Las expresiones de AngularJs se escriben entre dos corchetes: `{{ expresión }}`, son muy parecidas a las expresiones de JavaScript: pueden contener literales, operadores y variables, ... Las expresiones enlazan datos a HTML y AngularJS "generará" datos donde se escribe la expresión.

1.1 Arquitectura de aplicaciones de AngularJs

Las aplicaciones hechas con AngularJs están definidos por módulos, por tanto, estos módulos tienen el código de la aplicación principal. Una aplicación puede contener varios módulos y cada módulo contiene un código que define funciones específicas.

Un módulo puede entenderse como un contenedor en el que se pueden colocar componentes de la aplicación, como controladores, servicios, filtros y directivas. Los módulos se conectan con la aplicación mediante la directiva *ng-app* a través de un valor.

Las aplicaciones de AngularJS están controladas por controladores, es decir, los controladores de AngularJS controlan los datos de las aplicaciones. Estos controladores son objetos de JavaScript. Un módulo puede tener varios controladores, y estos controladores se definen mediante la directiva *ng-controller*. Para vincular la parte de vista (HTML) con los controladores, usamos AngularJs scope, un objeto que dispone de propiedades y métodos.

2. Visual Studio Code

Visual Studio Code^{[4][5]} es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Fue lanzado bajo la licencia MIT y su código fuente fue publicado en GitHub.

Como todos los editores de código, Visual Studio Code tiene características diseñadas para simplificar y acelerar la escritura de código fuente, como marcas de errores y advertencias, autocomplección, colores de la sintaxis y pareo de llaves. Aparte de estas características, Visual Studio Code es extensible y personalizable, es decir, los programadores pueden instalar extensiones para agregar nuevas funcionalidades como depurar código fuente o para conectarse a servicios adicionales.

Un ejemplo de las extensiones es depurador para Chrome que permite al programador depura su código de JavaScript en el navegador Chrome o en cualquier otro navegador que admita el protocolo del depurador de Chrome.

3. Servicios Web

Un servicio web ^{[6][7]} es una vía de proporcionar comunicación, compartición de datos e intercambio de información entre máquinas conectadas en Red. Esta interoperabilidad se consigue a través de la adopción de estándares abiertos.

En la arquitectura de servicios web existen tres partes: el proveedor de servicios web, un cliente que pide el servicio web y el publicador.

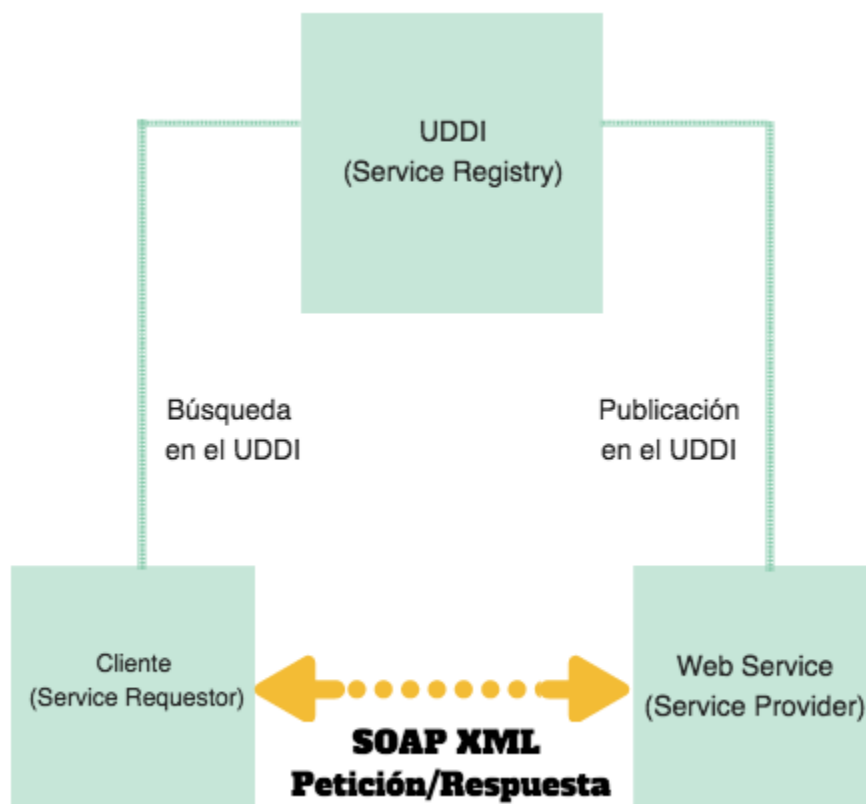


Figura 1: Componentes de servicios web

Fuente: <https://diego.com.es/introduccion-a-los-web-services>

Como se ve en la figura 1, son necesarios SOAP y UDDI para la correcta comunicación entre los tres componentes. SOAP^[8] es un protocolo estándar para el intercambio de información entre aplicaciones definiendo qué información se envía y cómo mediante XML. UDDI^[9] es un estándar XML para describir, descubrir y registrar servicios web con ayuda de WSDL siendo WSDL un lenguaje basado en XML para describir los servicios web y la forma de acceder a ellos.

El flujo entre los tres componentes es el siguiente: El proveedor de un servicio web genera el WSDL que describe el servicio web y lo registra en UDDI que actúa como un publicador. El cliente que requiere servicio web se pone en contacto con el UDDI mediante mensajes SOAP para localizar el servicio web. El UDDI manda al cliente el WSDL correspondiente al servicio web solicitado. Una vez recibido el WSDL, el cliente envía una petición al proveedor del servicios web teniendo en cuenta las informaciones proporcionadas por el WSDL. El proveedor envía la respuesta correspondiente a la petición del cliente al cliente mediante mensajes SOAP. Así, el cliente puede analizar el mensaje de respuesta SOAP y lo interpreta o genera un error si ha habido alguno.

Más tarde, REST^[11] logró gran impacto. En un principio, REST hace referencia a un conjunto de principios de arquitecturas que supuso un gran cambio al diseño de los servicios web haciendo foco en los recursos del sistema. Aparte de XML, JSON (JavaScript Object Notation) ^[10] es otro formato que se usa los servicios web RESTful en el intercambio de datos. JSON se representa usando texto y podemos convertir cualquier objeto de JavaScript en JSON y enviar JSON al servidor. También podemos convertir cualquier JSON recibido del servidor en objetos JavaScript. De esta manera podemos trabajar con los datos como objetos de JavaScript, sin complicados análisis ni traducciones.

Las características fundamentales de los servicios web RESTful son las siguientes:

- utiliza los métodos HTTP explícitamente,
- no mantiene el estado,
- expone URIs con forma de directorios, y
- el formato de los datos que se transfieren puede ser XML o JSON.

donde URI es un identificador uniforme de recurso y los métodos HTTP son GET, POST, PUT y DELETE.

Las operaciones típicas de los servicios web RESTful son: listar, crear, leer, actualizar y borrar. Cada operación requiere 2 parámetros: El URI y un método HTTP.

En la figura 2 podemos observar la relación entre las operaciones de los servicios web RESTful y los métodos HTTP.

Operación	Método HTTP	URI	Parámetros	Resultado
Listar	GET	/[recurso]	No aplica	Lista del tipo de recurso
Crear	POST	/[recurso]	Dentro del cuerpo en el POST	Se crea un nuevo recurso
Leer	GET	/[recurso]/[recurso_id]	No aplica	Recurso en función al id
Actualizar	PATCH/PUT	/[recurso]/[recurso_id]	Se pasan usando una cadena de consulta	Se actualiza/reemplaza el recurso
Borrar	DELETE	/[recurso]/[recurso_id]	No aplica	Se elimina el recurso en función al id

Figura 2: Operaciones de servicios web RESTful
Fuente: <https://blog.bi-geek.com/servicios-web-restful/>

En este presente trabajo, utilizamos los servicios web RESTful para obtener datos de APIs de Alpha Vantage^[12]. Alpha Vantage es una empresa que proporciona gratuitamente datos históricos y en tiempo real en formato .csv y .json de diferentes tipos de activos. Una vez registrado, los usuarios pueden acceder vía web para descargar el tipo de dato deseado con la api key que se obtiene al registrarse.

Las APIs de Alpha Vantage se agrupan en 4 categorías:

- APIs de acciones de series temporales (Time Series Stock).
- Datos fundamentales (Fundamental Data).
- Divisas entre monedas, tanto físicas como digitales (FOREX/Cryptocurrencies).
- Indicadores técnicos (Technical Indicators).

En su documentación se puede encontrar los parámetros necesarios y opcionales para la obtención de datos financieros.

```
{
  "Meta Data": {
    "1. Information": "Daily Prices (open, high, low, close) and Volumes",
    "2. Symbol": "IBM",
    "3. Last Refreshed": "2020-09-17",
    "4. Output Size": "Compact",
    "5. Time Zone": "US/Eastern"
  },
  "Time Series (Daily)": {
    "2020-09-17": {
      "1. open": "122.6200",
      "2. high": "125.5500",
      "3. low": "121.9700",
      "4. close": "124.9200",
      "5. volume": "3171262"
    },
    "2020-09-16": {
      "1. open": "122.7100",
      "2. high": "125.8200",
      "3. low": "122.7100",
      "4. close": "124.2200",
      "5. volume": "3789458"
    },
    "2020-09-15": {
      "1. open": "122.8200",
      "2. high": "123.4000",
      "3. low": "122.2400",
      "4. close": "122.4400",
      "5. volume": "2915221"
    },
    "2020-09-14": {
      "1. open": "122.3600",
      "2. high": "123.3800",
      "3. low": "121.7600",
      "4. close": "122.0900",
      "5. volume": "3641887"
    }
  }
}
```

Figura 3: Ejemplo de respuesta devuelta por Alpha Vantage

La figura anterior muestra un ejemplo de respuesta devuelta por APIs de acciones de series temporales de Alpha Vantage en formato .json. Como podemos observar, en "Meta Data" están las informaciones claves como el tipo de serie de tiempo, en este caso diario, el símbolo de la empresa y el tiempo de la última actualización. En "Time Serie (Daily)" están la información sobre precio diario y el volumen de las acciones de la empresa de cada día ordenado según la fecha, de más reciente a menos reciente.

4. Otras tecnologías

PHP (*Hypertext Preprocessor*)^{[13][14]} es un lenguaje de programación de servidores de código abierto que sirve para el desarrollo web y que puede estar incluido en archivos del tipo HTML. El código de PHP está encerrado entre las etiquetas especiales de comienzo y final `<?php` y `?>`. Utilizamos esta tecnología en este trabajo para conectar nuestra aplicación con la base de datos.

XAMPP^{[15][16]} es un entorno de desarrollo que consiste principalmente en el sistema de gestión de bases de datos MySQL mediante phpMyAdmin, el servidor web Apache y los intérpretes para lenguajes PHP. Usamos este entorno como un servidor web para hacer las depuraciones de los archivos de tipo PHP y observar el cambio de la base de dato durante la ejecución de los archivos de tipo PHP.

Angular-Chart.js^[17] es un paquete de software libre que sirve para hacer gráficos reactivos y receptivos para AngularJs basandose en Chart.js. Su código fuente está depositado en Github.

SECCIÓN 3

FUNCIONALIDADES Y DISEÑO

En esta sección vamos a ver las funcionalidades de nuestra aplicación web y su diseño.

1. Funcionalidades

Las funcionalidades básicas que están presentes en nuestra aplicación web son las siguientes:

- Búsqueda de empresas por su nombre o por símbolo de la empresa para poder visualizar sus datos.
- Visualización de los datos financieros e históricos de las acciones de una empresa, representados en gráficas o en tablas.
- Filtro de los datos financieros tales como elección de tipo de datos, intervalo de tiempos entre dos datos consecutivos.
- Descarga de archivos y gráficos en distintos formatos.
- Inicio de sesión, registro y fin de sesión de usuarios dentro de la aplicación.
- Acceso de área privada donde el usuario podrá visualizar sus datos personales.
- Modificación del número de acciones que el usuario posee.
- Visualización de los datos financieros más recientes de las empresas que el usuario posee acciones.

2. Diseño

En esta subsección, vamos a mostrar el diseño de la página web mediante *mockups*.

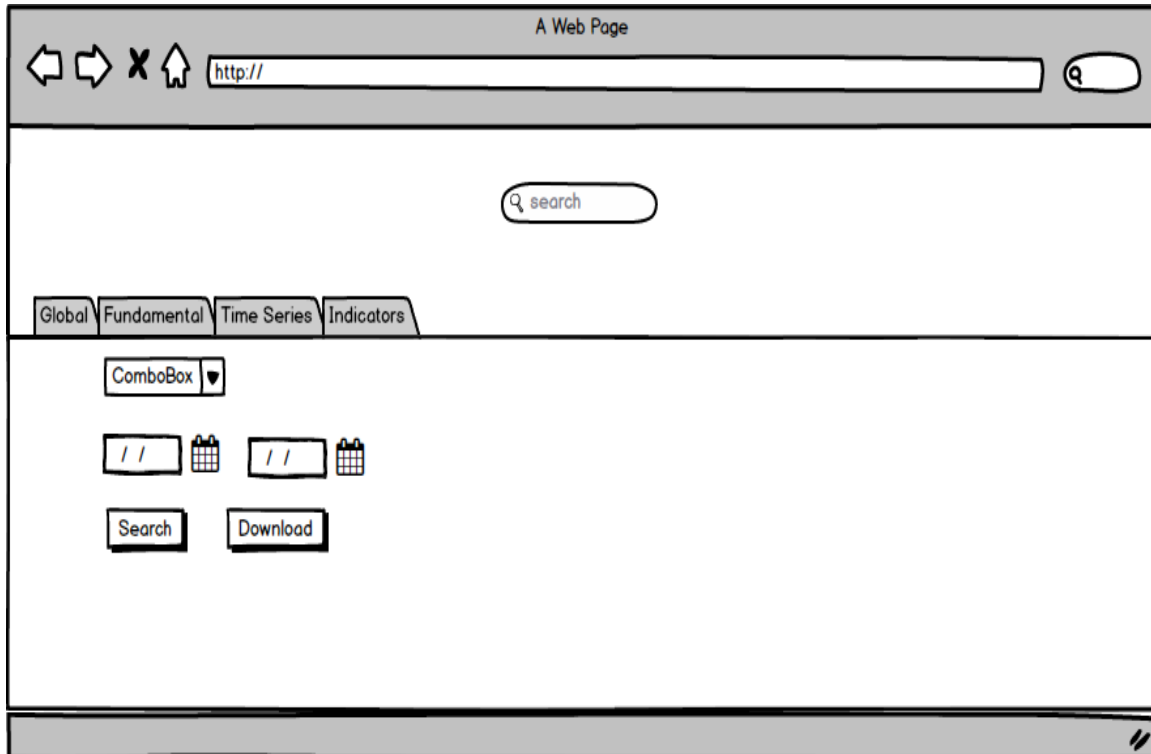


Figura 4: Home

En la figura 4 que ilustra la pantalla Home de la aplicación podemos ver: en el centro de la pantalla hay un campo de búsqueda. Este campo servirá para los usuarios introduzcan el nombre o símbolo de la empresa que quiere obtener informaciones. Además, pueden elegir la categoría de los datos financieros que quieren buscar. Las categorías son:

- Global, para mostrar un resumen sobre los precios y volumen de acciones de una empresa.
- Fundamental, para mostrar los datos fundamentales de una empresa como descripción de la empresa. Esta categoría corresponde a la categoría Fundamental Data de APIs de Alpha Vantage.
- Time Series, para mostrar informaciones de las acciones de series temporales. Esta categoría corresponde con la categoría Time Serie Stock de Alpha Vantage.
- Indicators, para mostrar informaciones relacionados con los indicadores

técnicos. Esta categoría corresponde a la categoría “Technical Indicator” de Alpha Vantage.

Como podemos observar en la figura, la aplicación permite a los usuarios poner filtros para los datos financieros como la fecha de inicio y la fecha final.

Una vez terminada las elecciones, los usuarios hay que pulsar en el botón “search” para visualizar el resultado de búsqueda. La aplicación mostrará el resultado de búsqueda mediante gráficas, como muestra la figura 5.

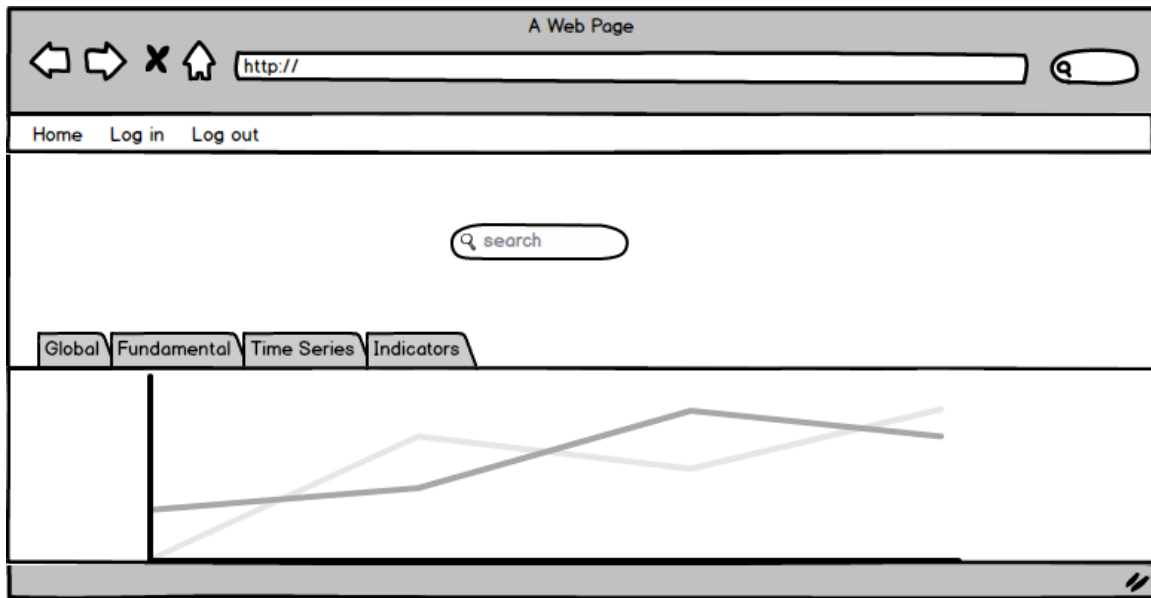


Figura 5: Resultado de búsqueda

Además, la aplicación dispone de una barra de navegación con 3 opciones: “Home”, “Log in” y “Log Out”. Si el usuario pulsa en “Log in” de la barra de menú, la aplicación saltará a la página de log in. Esta página está ilustrada en la figura 6. Podemos observar que, en esta página, el usuario tiene que introducir su nombre de usuario y la contraseña correspondiente para poder iniciar la sesión correctamente.

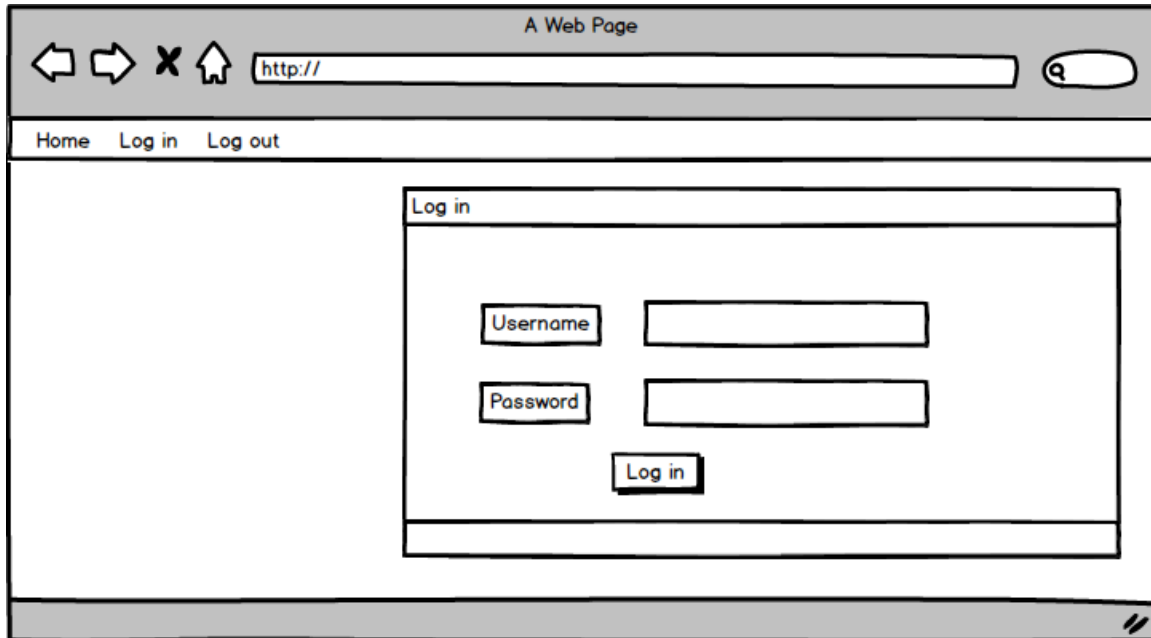


Figura 6: Página de log in

Una vez el usuario accede a su área privada, que tendrá la forma de la figura 7, puede ver una tabla de acciones. En esta tabla estarán las informaciones de las acciones que posee el usuario tales como el nombre de la empresa, el número de acciones que posee...

Aparte de la tabla de acciones, la aplicación también mostrará los datos financieros más recientes de las empresas que el usuario posee acciones.

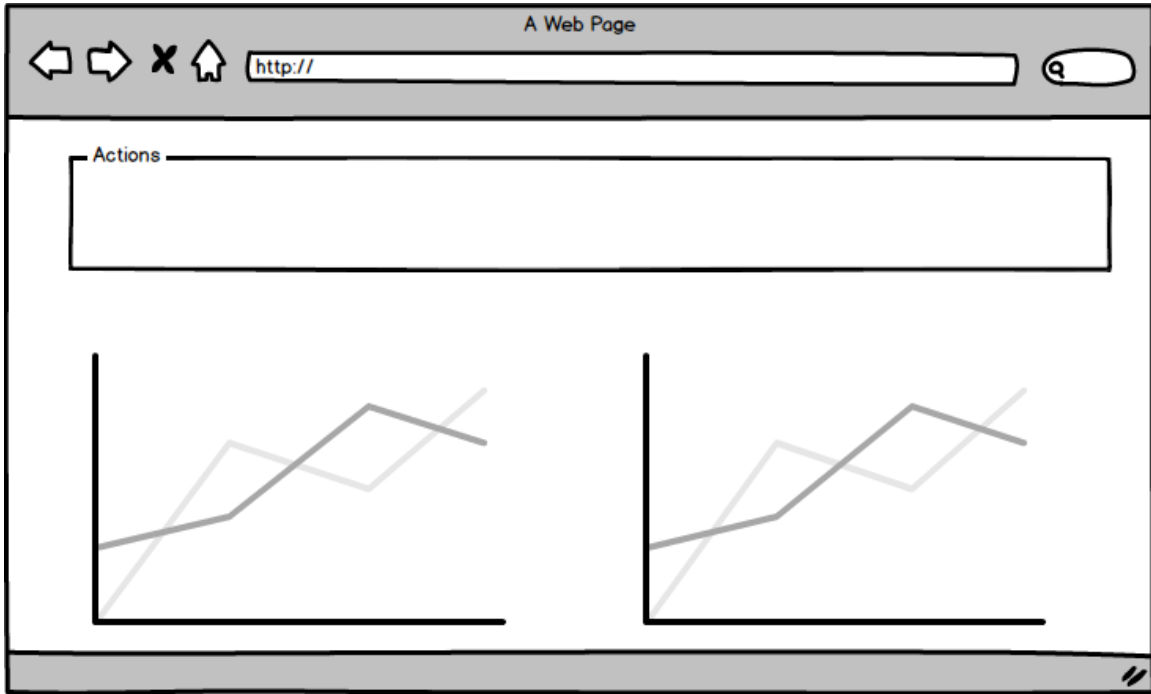


Figura 7: Área privada de un usuario

SECCIÓN 4

IMPLEMENTACIÓN

En esta sección se detallará la arquitectura de la aplicación, explicando las diferentes pantallas que ofrece la aplicación web y las interacciones que existen en estas pantallas.

1. Arquitectura

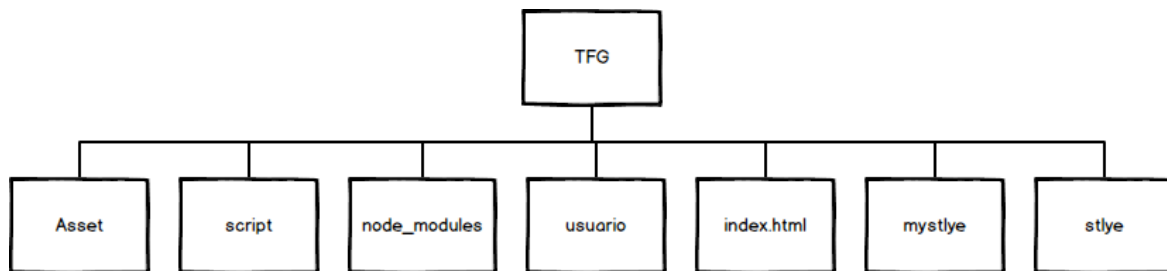


Figura 8: Arquitectura nivel 1

La figura 8 ilustra la organización del código fuente de la aplicación en archivos y en carpetas. El código fuente está organizado en cuatro carpetas y 3 archivos.

- El archivo “index.html” es el archivo raíz de la aplicación, está implementado según las figuras 4 y 5.
- En el archivo “mystlye.css” se escribe cómo queremos que sea el estilo de presentación de las etiquetas para el archivo raíz.
- El archivo “style.css” es la hoja de estilo para las interfaces de usuarios.
- En la carpeta *asset* están todas las imágenes usadas en la aplicación como el fondo de la aplicación. Todas ellas tienen licencia MIT.
- En la carpeta *script* están los controladores para la pantalla Home.
- En la carpeta *usuario* están los archivos para hacer registros, inicio de sesión, cierre de sesión y acceso al área privada.
- En la carpeta *node_module* están el código fuente de Angular-Chart.js que nos permite dibujar los gráficos.

Veamos ahora la estructura de las carpetas más importantes: *script* y *usuario*.

La carpeta *script* contiene cinco archivos como se muestra en la figura 9. Todos los archivos sirven para la búsqueda, representaciones gráficas y descargas de los datos financieros de APIs de Alpha Vantage menos el archivo “principal.js” que tiene otro tipo de funcionalidad. La diferencia entre ello es la categoría de datos que se maneja en los controladores.

- El archivo “controlador.js” maneja los datos financieros de la categoría de Time Series Stock.
- El controlador “fundamental.js” maneja los datos financieros de la categoría de Fundamental Data.
- En el archivo “global.js” se maneja datos financieros de Quote Endpoint de Alpha Vantage.
- El archivo “indicator.js” sirve para realizar búsquedas de los indicadores técnicos.
- El archivo “principal.js” sirve para realizar la búsqueda del nombre de empresas y su símbolo correspondiente.

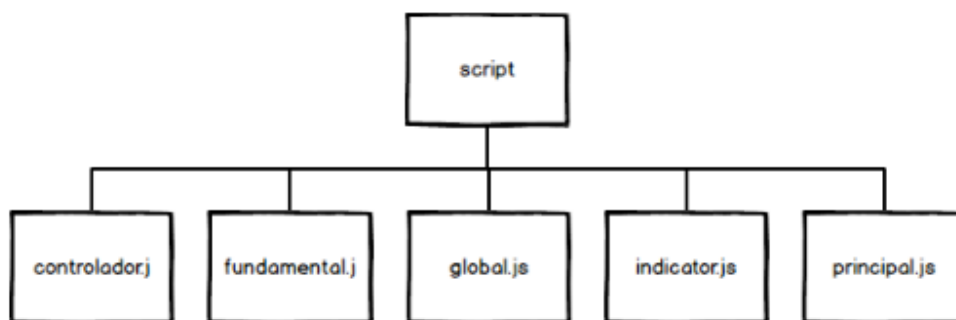


Figura 9: Arquitectura de la carpeta script

En la figura 10, está mostrada la arquitectura de la carpeta *usuario*.

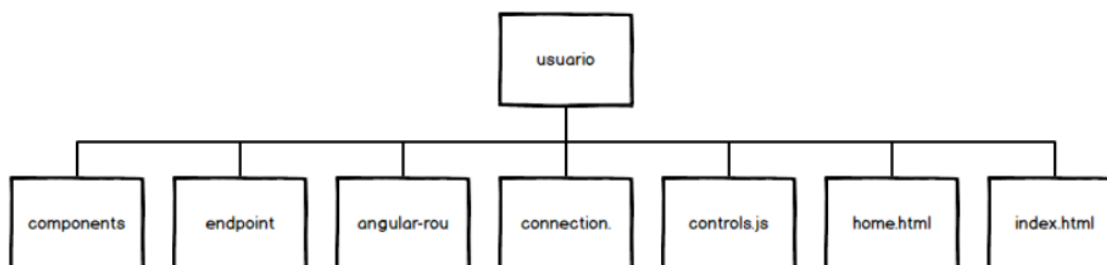


Figura 10: Arquitectura de la carpeta usuario

Como podemos ver, esta carpeta está compuesta por cinco archivos y dos carpetas:

- El archivo “index.html” es el archivo raíz para las interfaces de usuarios.
- El archivo “home.html” define la vista de componente *home* de las interfaces de usuarios. Este componente permite al usuario elegir entre iniciar sesión y hacer registro.
- En el archivo “control.js” está definido el controlador para controlar “home.html”. En este archivo está implementada las funciones de salto de pantallas según la opción que el usuario elige.
- El archivo “angular-route.min.js” sirve para hacer enrutamientos. Esta funcionalidad está implementada en el archivo “control.js”.
- El archivo “connection.php” sirve para conectar la base de datos a la aplicación.
- En la carpeta *endpoint* están los archivos de tipo PHP.
- En la carpeta *component* están las diferentes vistas y sus correspondientes controladores.

Ahora, explicaremos con más detalle las carpetas *endpoint* y *components*.

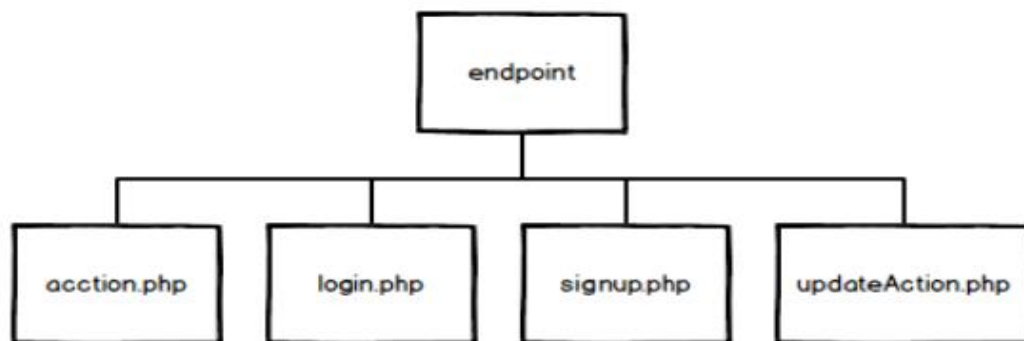


Figura 11: Arquitectura de la carpeta endpoint

La figura 11 muestra la estructura de la carpeta endpoint. En esta carpeta están cuatro archivos:

- El archivo “acction.php” sirva para buscar todas las acciones de un usuario en la base de datos.
- El archivo “login.php” sirve para iniciar una sesión consultando la base de datos la existencia de usuario y la corrección de la contraseña.
- El archivo “signup.php” sirve para registrar un usuario y actualiza la tabla *user_data* de la base de datos insertando informaciones del nuevo usuario.
- El archivo “updateAction.php” sirve para actualizar el número de acciones de un usuario y las informaciones relacionadas a las acciones.

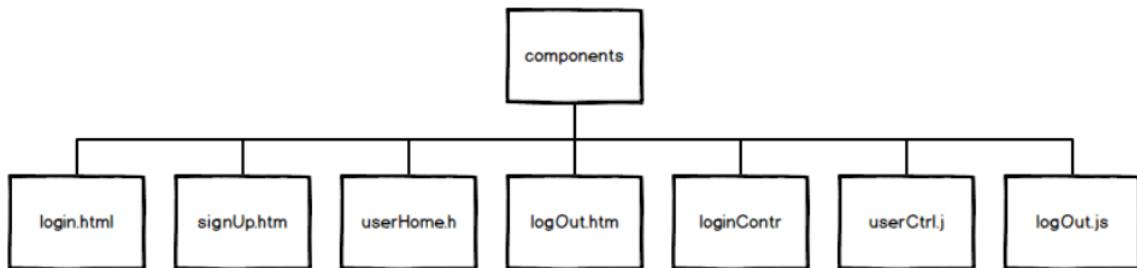


Figura 12: Arquitectura de la carpeta components

Por último, veamos la estructura de la carpeta *components* que está ilustrado con la figura 12:

- El archivo “login.html” define la vista de componente *login* que permite al usuario iniciar una sesión.
- El archivo “signUp.html” define la vista de componente *signUp* que permite al usuario hacer el registro.
- El archivo “loginController.js” es el controlador para los componentes *login* y *signUp*. En este archivo está implementada las funciones para que los componentes *login* y *signUp* funcione correctamente, así como solicitar los datos de la base de datos y las comprobaciones de datos solicitados.
- El archivo “userHome.html” define la vista de componente *user* que es el área privada del usuario.
- El archivo “userCtrl.js” es el controlador del componente *user*. Están implementadas en este archivo funciones como obtención de datos desde la base de datos con ayuda de los archivos de tipo PHP y mostración de datos en tablas.
- El archivo “logOut.html” define la vista de componente *logOut*.
- El archivo “logOut.js” es el controlador del componente *logOut*, su existencia permite al usuario cerrar una sesión.

2. Base de datos

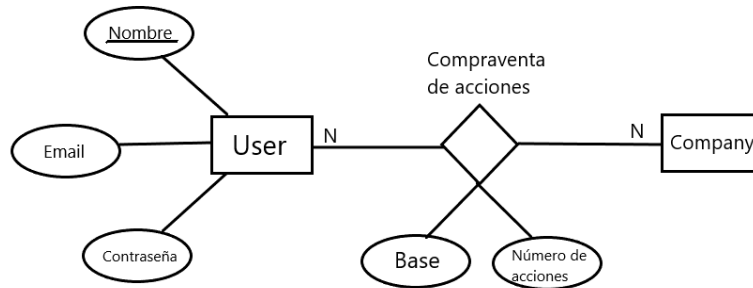


Figura 13: Diagrama ER

Antes de empezar a explicar la estructura de la base de datos, veamos el diagrama ER que está en la figura 13.

La figura 13 está representando una relación N:N entre los usuarios y las empresas. Los usuarios tienen tres atributos: nombre, email y contraseña, donde el atributo nombre actúa como identificador del usuario. Los usuarios están relacionados con las empresas mediante la relación hacer compraventa de acciones. Como un usuario puede tener acciones de varias empresas y una empresa puede tener varios accionistas, la relación es N:N. Al relacionar un usuario con la empresa, genera dos atributos: base y número de acciones. El atributo número de acciones indica el número de acciones que posee el usuario de la empresa relacionada. El atributo base indica el dinero total que el usuario ha invertido en las compras de acciones de una empresa.

Para implementar una base de datos según este diagrama ER, se necesita dos tablas: una para los usuarios y otra para la relación compraventa de acciones. Por tanto, nuestra base de datos está compuesta por las tablas *user_data* y *user_company_acction*.

La tabla *user_data* recoge las informaciones de los usuarios tales como el nombre de usuario, la contraseña y su email. Se identifica los usuarios por el nombre de usuario. Aseguramos que cada nombre de usuario es único cuando se registra un nuevo usuario, comprobando si se puede encontrar el nombre de usuario en la base de datos.

La tabla *user_company_acction* representa la relación compraventa de acciones,

es decir, recoge las informaciones relacionadas a las acciones de los usuarios. Esta tabla tiene cuatro claves: `user_name`, `company`, `number_acction`, `base`. `User_name` es un foreign key que está relacionado con la clave primaria de la tabla *user_data*. `Company` es la clave que identifica a una empresa por su símbolo. Estas claves juntas forman la clave primaria de esta tabla. La clave `number_acction` indica el número de acciones de la empresa `company` que el usuario `user_name` posee. La clave `base` indica la cantidad total de dinero ha invertido el usuario comprando acciones de una empresa. Tanto la clave `number_acction` como la clave `base` se actualiza cuando el usuario aumenta o disminuye el número de acciones en su área privada.

3. Pantallas

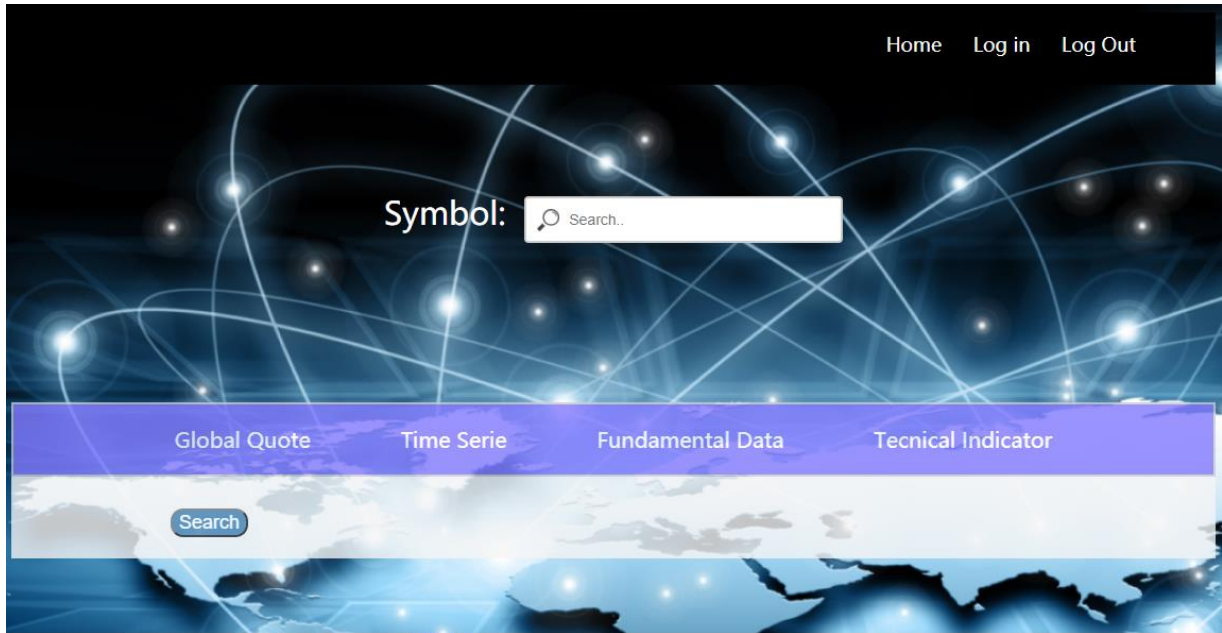


Figura 14: Pantalla Home

La figura 14 muestra la pantalla Home de la aplicación, que es la primera pantalla que el usuario visualiza al entrar a la aplicación. Observamos que esta pantalla tiene 4 elementos principales: la barra de navegación que está al principio de la pantalla, el campo de texto que está en el centro de la pantalla, la barra de menú que permita al usuario elegir la categoría de datos y la parte que permita al usuario configura filtros de datos de búsqueda y visualiza el resultado de búsqueda.

La búsqueda de empresas cuenta con la función de autocompletar a medida que el usuario va introduciendo caracteres como se muestra la figura 15.

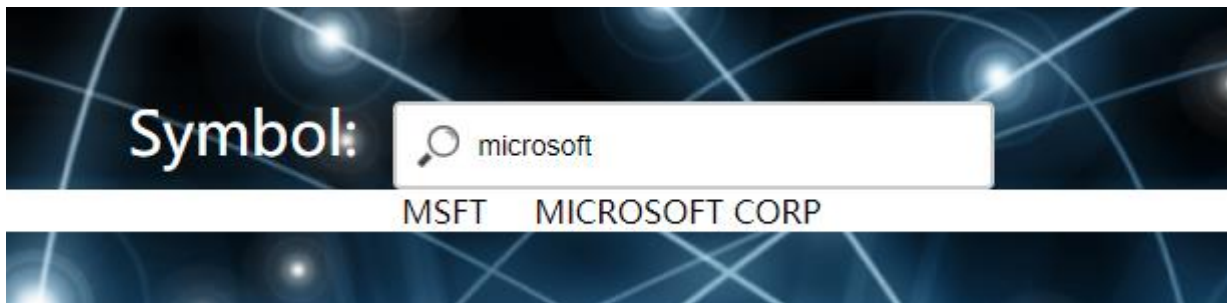


Figura 15: Búsqueda de empresas



Figura 16: Ejemplo de filtro

Una vez el usuario ha elegido la empresa, puede elegir la categoría de datos. Según el tipo de stock, la aplicación mostrará diferentes filtros. En la figura 16 está el ejemplo de elegir la categoría Time Series Stock y la función TIME_SERIES_INTRADAY. Cuando terminamos la elección de la función, la aplicación nos muestra el filtro “Interval” Este filtro sirve para elegir el intervalo de tiempo entre dos datos consecutivos. Los valores admitidos son: 1 minuto, 5 minutos, 15 minutos, 30 minutos y 60 minutos.

Si pulsamos en el botón *Search*, la aplicación nos mostrará los resultados gráficamente, como en la figura 17.

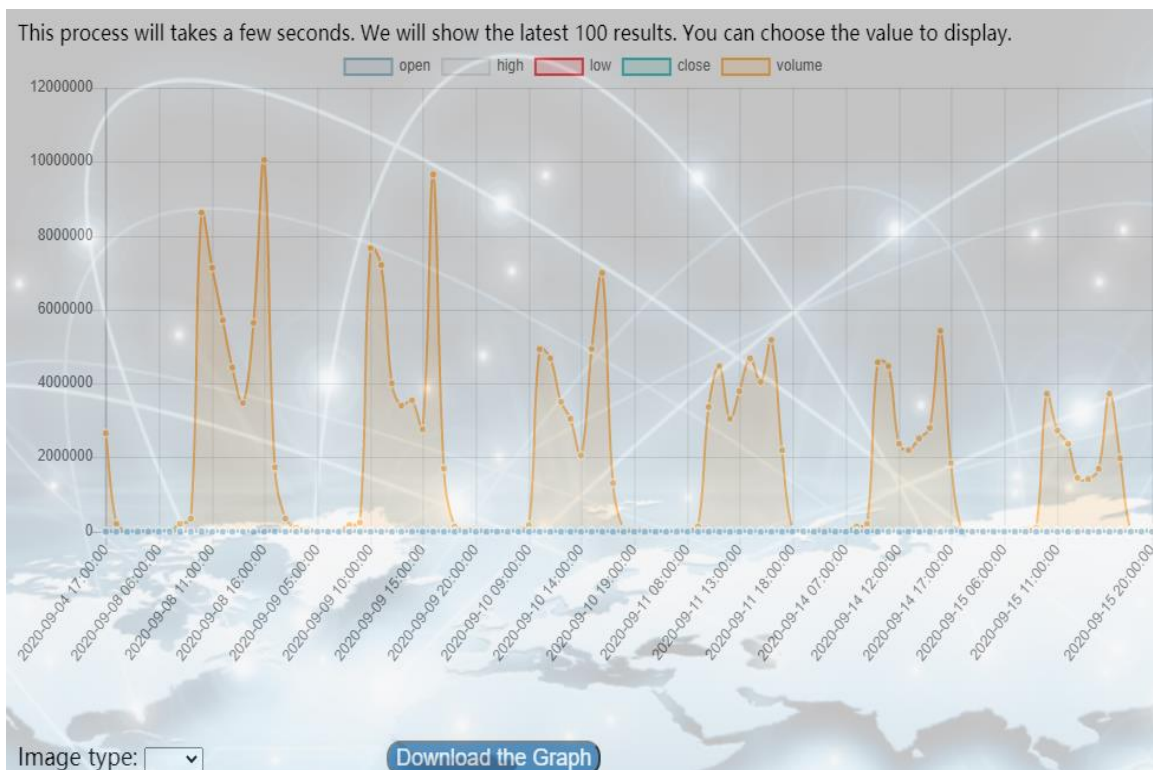


Figura 17: Ejemplo de resultado

Como podemos ver, además del gráfico, también nos muestra la opción de

descargar la gráfica permitiendo al usuario elegir el tipo de imagen. Observamos también que el usuario puede interactuar con el gráfico eligiendo qué datos no quiere visualizar haciendo clic en la parte superior del gráfico.

Si el usuario elige pulsa “Log in” de la barra de navegación, la aplicación saltará a la pantalla que está en la figura 18, la cual permite al usuario elegir entre iniciar sesión y registrar.

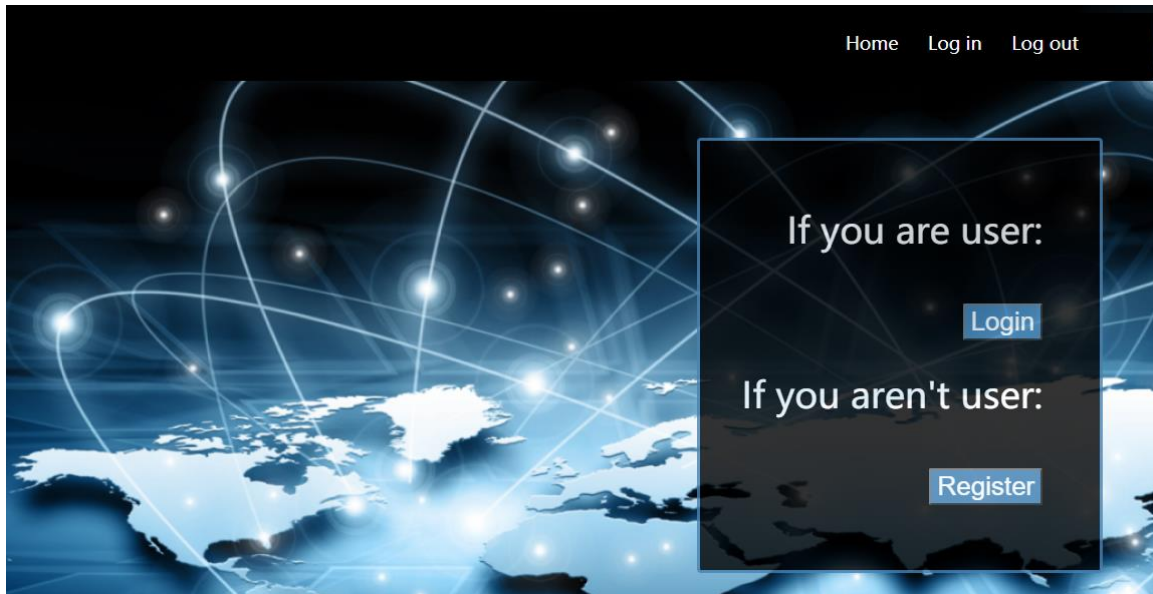


Figura 18: Elección entre Login y Register

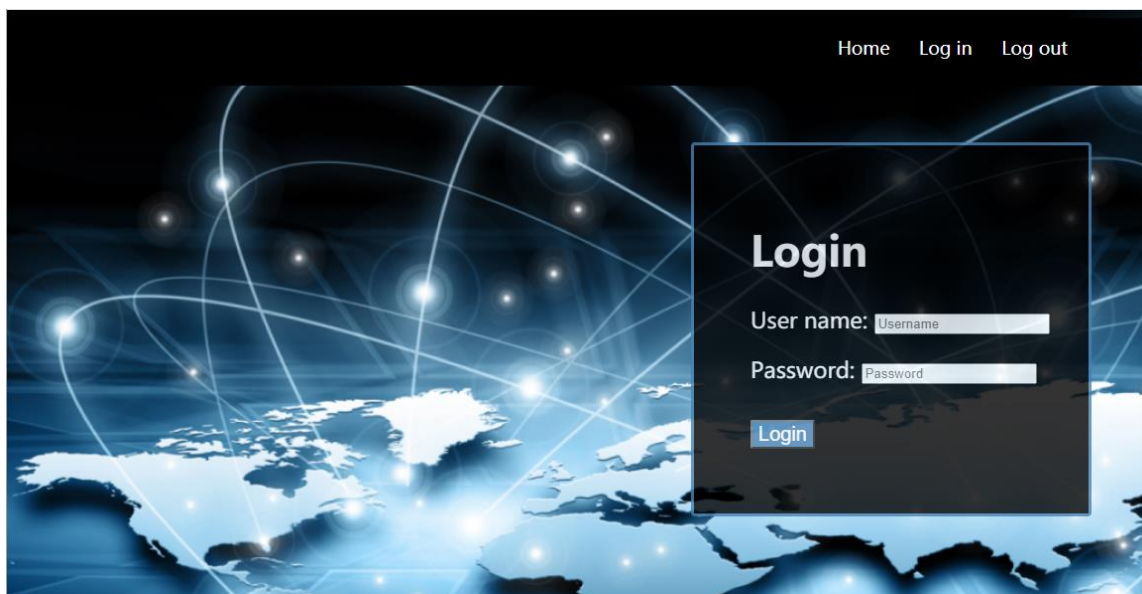


Figura 19: Pantalla de Login

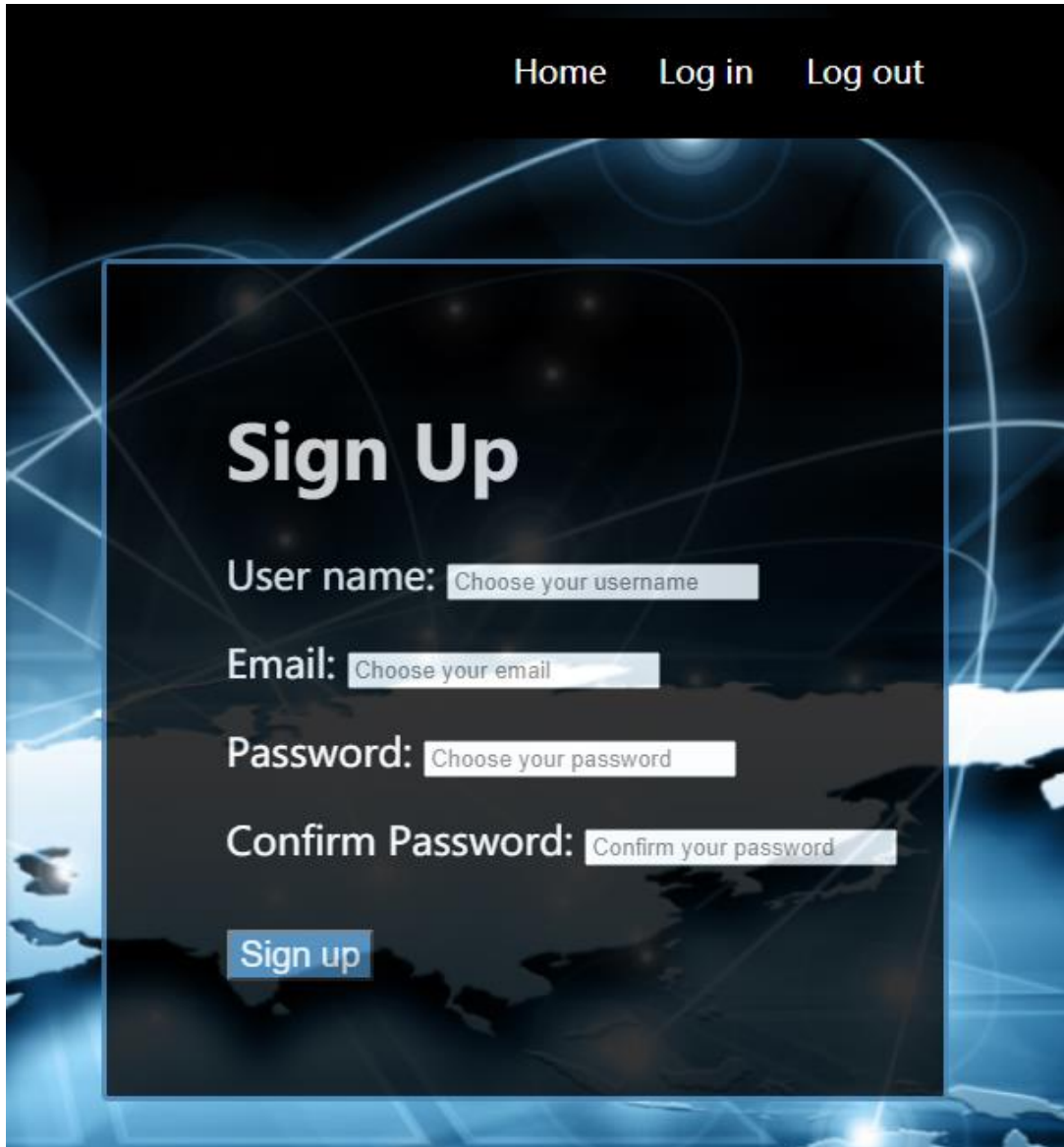


Figura 20: Pantalla de Register

La figura 19 nos muestra la pantalla de *Login* y la figura 20 nos muestra la pantalla de *Register*. La aplicación nos mostrará una de estas pantallas según la opción que hemos elegido antes. Como vemos, las dos pantallas son muy similares. En la pantalla de *Login* están los datos necesarios para poder iniciar una sesión y en la pantalla de *Register* están los datos necesarios para poder hacer el registro.

Una vez hemos iniciado la sesión correctamente, la aplicación nos mostrará la pantalla *userHome*, veremos esta pantalla por componentes:

En primer lugar, veamos la tabla de acciones del usuario que está en la figura 21.

Según la figura 21, la tabla de acciones muestra los datos relacionados con las acciones que el usuario posee tales como el símbolo de la empresa, el número de acciones, el beneficio de las acciones...Las columnas "Current/action" Y "Profit" se actualiza dinámicamente solicitando los datos de Alpha Vantage.

Además, ofrece dos funciones: añadir o disminuir el número de acciones de cada empresa.

Your actions:

Company	Number Acction	Base	Current/acction	Profit	Acction
IBM	10	1456.15	205.0500	594.3499999999999	<input type="button" value="add"/> <input type="button" value="subtract"/>
MSFT	5	1050.25	124.2200	-429.15	<input type="button" value="add"/> <input type="button" value="subtract"/>

« 1 »

Figura 21: Tabla de acciones

El siguiente componente es TotalProfit que nos indica el beneficio total de todas las acciones. Este componente está representado en la figura 22.

Total Profit:
165.19999999999993

Figura 22: Total profit

Por último, está el componente gráfico en las figuras 23 y 24.



Figura 23: Componente gráfico 1



Figura 24: Componente gráfico 2

Como vemos, la aplicación muestra tantos gráficos como el número de empresas que están en la tabla de acciones. Estos gráficos están basados en el stock Intraday de Alpha Vantage con un intervalo de 30 minutos. Siempre muestra los datos financieros más recientes. A diferencia con los gráficos de la pantalla Home, aquí no muestra la gráfica de volumen.

Si el usuario pulsa en “Log Out” de la barra de navegación, la aplicación comprueba si el usuario ha inicializado la sesión. En caso negativo, mostrará una

advertencia. En caso confirmativo, la aplicación mostrará la pantalla de Log Out que está en la figura 25:

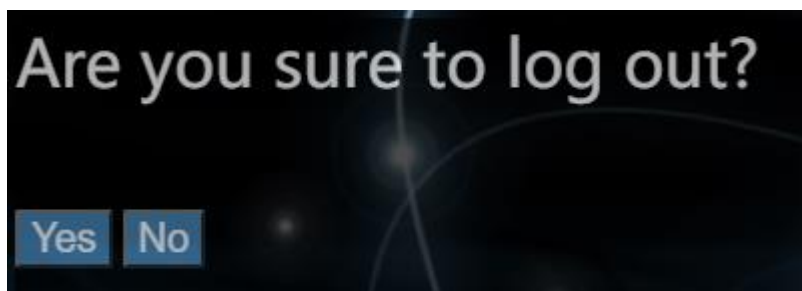


Figura 25: Pantalla de Log Out

Una vez el usuario ha cerrado la sesión, se salta a la pantalla Home.

4. Limitaciones

La aplicación cuenta con una serie de limitaciones que se han ido presentando durante el proceso de desarrollo y de implementación.

La principal limitación es el uso de la API de Alpha Vantage. Para obtener datos a través de la API es necesario el uso del api key que se puede conseguir gratuitamente al registrarse. El impedimento de esta clave es que tiene un número máximo de llamadas por minuto, siendo este número 5. Por lo tanto, si un usuario que esté navegando realiza más de 5 llamadas a la API en un minuto, la respuesta de la API es nula y como consecuencia, algunas funcionalidades de la aplicación no funcionan correctamente. Por ejemplo, en la tabla de acciones, si un usuario tiene acciones de más de 5 empresas, la columna *profit* y la columna *current/action* no aparecerán correctamente puesto que es necesario llamar a API para llenar estas columnas correctamente, además el componente *TotalProfit* tampoco aparecerá correctamente como consecuencia de la columna *profit* no aparece correctamente.

Otra limitación es descarga de archivos en formato .json. Si el navegador del usuario no es superior a la versión que está en la figura 26, la descarga de archivos en formato .json produce error.

Attribute					
download	14.0*	18.0	20.0*	10.1	15.0

* Chrome 65+ and Firefox only support same-origin download links.

Figura 26: Versiones soportados

Fuente: https://www.w3schools.com/tags/att_a_download.asp

SECCIÓN 5

CONCLUSIONES Y TRABAJO FUTURO

En esta última sección, describiremos las conclusiones obtenidas y proponemos una serie de mejoras que se pueden implementar.

1. Conclusiones

Antes de empezar a crear esta aplicación, hay que estudiar todo lo referente a las acciones, volatilidad del mercado de valores, etc.^{[16][17][18][19][20][21]}. Este proceso de aprendizaje nos permite identificar las funciones básicas que tienen que estar presentes en la aplicación para permitir a los usuarios valorar el progreso de los valores en los que actualmente se ha invertido en bolsa con más facilidad. El primer resultado que he obtenido de este trabajo es este conocimiento de Economía, que desconocía.

Después de este aprendizaje, empezamos con el aprendizaje de HTML para crear la aplicación en versión web. Esta elección de versión es debido a la independencia del sistema operativo de las aplicaciones web y a la facilidad para actualizar y mantenerlas. La mayor ventaja de esta versión para los usuarios es su facilidad de acceso y de uso sin instalar softwares. Los usuarios pueden acceder al contenido de la aplicación y usar los servicios teniendo un ordenador o un dispositivo móvil con navegador y acceso a Internet.

Una vez terminadas los procesos de aprendizaje, empezamos el proceso de desarrollo de la aplicación teniendo en cuenta nuestros objetivos. Debido a que no soy experta en los temas de finanzas, decidí implementar una aplicación que proporciona las funciones más importantes para facilitar a los usuarios a la hora de tomar decisiones.

2. Trabajo futuro

A pesar de que la aplicación tiene una funcionalidad completa, durante el proceso de implementación de la aplicación, hemos surgido algunas ideas para extender y mejorar nuestra aplicación, tales como:

- Implementar la comparación de empresas para dar más información a los usuarios.
- Añadir más opciones para visualización de gráficas y de descargas. Tales como el tipo de las gráficas, el tipo de archivos soportados, el formato de las imágenes soportados.
- Mejorar el área privada de los usuarios añadiendo más funciones por ejemplo de historial de beneficios para cada usuario.
- Añadir más informaciones de los usuarios tales como foto de perfil, fecha de nacimiento...
- Permitir a los usuarios modificar sus perfiles, por ejemplo, puedan cambiar su foto de perfil.
- Añadir la función de consulta de tasa de cambio entre las monedas, tanto físicas como digitales.
- Mejorar los aspectos estéticos de la aplicación.
- Intentar superar las limitaciones de la aplicación como solicitar varios api keys de Alpha Vantage y utilizarlas en diferentes partes para disminuir la probabilidad de producir errores.

SECTION 5

CONCLUSIONS AND FUTURE WORK

In this last section, we will describe the conclusions obtained and we propose a series of improvements that can be implemented..

1. Conclusions

Before starting to create this application, you should study everything related to stocks, stock market volatility, etc ^{[16] [17] [18] [19] [20] [21]}. This learning process allows us to identify the basic functions that must be present in the application to allow users to assess the progress of the securities in which they are currently invested in the stock market more easily. The first result that I have obtained from this work is this knowledge of Economics, which I did not know.

After this training, we started with the HTML training to create the application in web version. This choice of version is due to the independence of the operating system of the web applications, as well as the ease of updating and maintaining them. The greatest advantage of this version for users is its ease of access and use without installing software. Users can access the content of the application and use the services having a computer or mobile device with a browser and Internet access.

Once the learning processes are finished, we start the application development process considering our objectives. Since I am not an expert in finance, I decided to implement an application that provides the most important functions to facilitate users when making decisions.

2. Future work

Although the application has full functionality, during the application deployment process, we have come up with some ideas to extend and improve our application, such as:

- Implement the comparison of companies to give more information to users.
- Add more options for displaying graphics and downloads. Such as the type of the graphics, the type of files supported, the format of the images supported.
- Improve the users' private area by adding more functions.
- Add more information about users such as profile photo, date of birth ...
- Allow users to modify their profiles, for example, they can change their profile picture.
- Add the exchange rate query function between currencies, both physical and digital.
- Improve the aesthetic aspects of the application.
- Try to overcome the limitations of the application such as requesting several Alpha Vantage's api keys and using them in different part to reduce the probability of producing errors.

BIBLIOGRAFÍA

- [1] Google, “AngularJs”
<https://docs.angularjs.org/guide/>
- [2] Colaboradores de Wikipedia, “AngularJs” Wikipedia, La enciclopedia libre,
<https://es.wikipedia.org/wiki/AngularJS>
- [3] Colaboradores de w3schools, “AngularJs”
<https://www.w3schools.com/angular/default.asp>
- [4] Colaboradores de Wikipedia, “Visual Studio Code” Wikipedia, La enciclopedia libre
https://es.wikipedia.org/wiki/Visual_Studio_Code
- [5] Microsoft, “Visual Studio Code”
<https://code.visualstudio.com/>
- [6] Colaboradores de Wikipedia, “Servicio web” Wikipedia, La enciclopedia libre
https://es.wikipedia.org/wiki/Servicio_web
- [7] Diego Lázaro, “Introduccion a los Web Services”
<https://diego.com.es/introduccion-a-los-web-services>
- [8] Colaboradores de Wikipedia, “SOAP” Wikipedia, La enciclopedia libre
https://es.wikipedia.org/wiki/Simple_Object_Access_Protocol
- [9] Colaboradores de Wikipedia, “UDDI” Wikipedia, La enciclopedia libre
<https://es.wikipedia.org/wiki/UDDI#>
- [10] Colaboradores de w3schools, “JSON”
https://www.w3schools.com/js/js_json_intro.asp
- [11] Colaboradores de Wikipedia, “REST” Wikipedia, La enciclopedia libre
https://es.wikipedia.org/wiki/Transferencia_de_Estado_Representacional
- [12] Alpha Vantage Inc. “Alpha Vantage”
<https://www.alphavantage.co/>
- [13] The PHP Group, “PHP”

<https://www.php.net/manual/es/intro-what-is.php>

[14] Colaboradores de w3schools, "PHP"

<https://www.w3schools.com/php/default.asp>

[15] Creadores de XAMPP, "XAMPP"

<https://www.apachefriends.org/es/index.html>

[16] Colaboradores de Wikipedia, "XAMPP" Wikipedia, La enciclopedia libre,

<https://es.wikipedia.org/wiki/XAMPP>

[17] Jerome Touffe-Blin, "Angular-Chart.js"

<https://jtblin.github.io/angular-chart.js/>

[16] Grupo elEconomista.es, "Direccionario de economía"

<https://www.eleconomista.es/diccionario-de-economia/>

[17] Grupo Finanzas para todos y Asociación lectura fácil, "Finanzas para todos-
Lectura fácil"

http://www.finanzasparatodos.es/es/secciones/recursosrelacionados/pdf/100428Finanzas_para_todos-Lectura_Facil.pdf

[18] L.E. y M.F. Arturo Morales Castro, C.P. y M.F. Benjamín Sánchez Rodríguez,
L.A. y M.F. José Antonio Morales Castro y L.C. José Gabriel Figueroa Flores

"Apuntes para la asignatura FINANZAS I (Finanzas Básicas)"

<http://fcasua.contad.unam.mx/apuntes/interiores/docs/98/2/finanzas1.pdf>

[19] Natalia Matrínez Serra, "Derivados financieros: fundamentos teóricos y
lecciones recientes"

https://ruc.udc.es/dspace/bitstream/handle/2183/16380/MartinezSerra_Natalia_TFG_2015_01de2.pdf?sequence=2&isAllowed=y

[20] Álvaro Rodríguez, "Introducción a la Bolsa de Valores"

<https://labuenainversion.com/introduccion-a-la-bolsa-de-valores/>

[21] Organización Khan Academy, "Mercados financieros y de capitales Unidad:
Acciones y bonos"

<https://es.khanacademy.org/economics-finance-domain/core-finance/stock-and-bonds>