**Pinzon Garcia Miguel Angel, Olaya Arias Pablo Olaya, Rojas Luna Juan Manuel, Quintero Castillo Carlos Fernando**

**No. de Equipo Trabajo: {*6*}**

# [[1]](#footnote-0) INTRODUCCIÓN

Desde factores como la pandemia, la guerra en ucrania, la crisis de los contenedores, la reactivación económica y las fluctuaciones en los mercados de valores, ha desencadenado una inflación y un encarecimiento del costo de vida en todo el mundo, generando una crisis, especialmente en el alza de los precios de la canasta familiar.

Una manera de contener la crisis económica de los hogares colombianos en el corto plazo en la adquisición de productos de la canasta familiar, es la búsqueda de ofertas que brinde al consumidor final el producto con el menor precio.

Por eso, desde una aplicación, se busca crear una herramienta de selección para los usuarios, donde tendrán al alcance las diferentes variedades en precios de una amplia gama de productos, destacando los almacenes que presentan la mejor oferta.

Gracias al desarrollo de software y las estructuras de datos, se facilitará la toma de decisiones en la búsqueda de los productos con bajo costo. Adicionalmente, los usuarios tendrán acceso a las diferentes ofertas que brindarán los almacenes de manera directa.

# DESCRIPCIÓN DEL PROBLEMA A RESOLVER

Se implementará una aplicación que reciba la información de los productos que el usuario desea cotizar, luego, ejecutará la búsqueda de los almacenes que ofertan el producto, almacenando estos datos, y luego organizando de menor a mayor precio, para que el cliente pueda ver los almacenes con las mejores ofertas en los productos solicitados, dando una respuesta completa y detallada con el mejor precio, todo desde la comodidad de un celular.

# usuarios DEL PRODUCTO DE SOFTWARE

La aplicación está dirigida para todas las personas que busquen reducir sus costos en la compra de productos con el mejor precio, especialmente las familias que se encuentren en crisis económica. También la aplicación está dirigida a las tiendas que quieran ofertar sus productos en esta aplicación.

# REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

## Ingreso del nombre del producto a comparar

## El usuario se encontrará con una interfaz en la cual se le pedirá que ingrese el nombre del producto o productos de los cuales quiere saber en qué tienda de las que están disponibles se encuentra el producto a un precio más económico.

*Acciones iniciadas y comportamiento esperado*:

*El usuario después de ingresar a la aplicación llegará al home donde habrá un espacio en el cual dirá”Ingrese el nombre del producto que desea buscar”, donde el usuario podrá añadir el producto, adicionalmente, en la parte inferior están los botones “Agregar” y “Eliminar” para poder adicionar o eliminar nuevos producto de la lista. Se espera que salga una lista en la que se mostrará el o los productos de las tiendas ordenados de menor a moyor precio.*

## Comparar productos entre tienda

*La aplicación buscará el producto que el usuario digitó*

*entre las diferentes tiendas disponibles.*

*Acciones iniciadas y comportamiento.*

*Cuando el usuario da clic en el botón “buscar”, la aplicación iniciará la búsqueda tienda por tienda, de cada unos de los productos agregados por el usuario; el orden de búsqueda se ejecutará en el mismo orden de ingreso de los productos.*

## Guardar productos

*En este proceso los usuarios pueden agregar productos al catálogo de su tienda.*

*Acciones iniciadas y comportamiento.*

*Los usuarios dispondrán de una ventana en la cual puedan ver la lista de productos de sus respectivas tiendas. En esta ventana encontrarán un botón que diga “Agregar producto”.*

## Eliminar productos

## El usuario puede eliminar productos de la lista que desea buscar.

*Acciones iniciadas y comportamiento.*

*Antes de realizar la búsqueda, el usuario tiene la opción para eliminar productos de la lista de productos a comparar, únicamente debe ingresar el nombre del producto en la ventana de texto y dar clic en la opción “Eliminar”.*

* *Buscar*

*Desde el botón “buscar” iniciará la búsqueda y comparación de los productos ingresados por el usuario*.

*Acciones iniciadas y comportamiento.*

*Desde el momento en que el usuario da clic en el botón “buscar”, la aplicación buscará las tiendas que ofertan cada unos de los productos ingresados por el usuario, la cual se realizará en cola, iniciando con el primer producto ingresado. Se fijará el precio de referencia con la primera tienda que contenga el producto a validar, y luego se actualizará el precio a comparar cuando se encuentre una tienda que contenga el producto y un precio inferior al de referencia.*

* *Mostrar lista de productos*

*La aplicación muestra los productos que el usuario buscó ordenado de menor a mayor precio con su respectiva tienda.*

*Acciones iniciadas y comportamiento.*

*El usuario previamente habrá digitado el nombre del producto que deseaba buscar y posteriormente pulsar el botón “Buscar”. Luego espera que salga una lista en la que se mostrará el o los productos de las tiendas ordenados de menor a moyor precio.*

# DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR

<https://www.figma.com/file/O8hch7GB0gnbtZIEVB0HN0/Economerk?type=design&node-id=0%3A1&t=cqRRzgzPnJheI7tu-1>

# Entornos de desarrollo y de operación

Para desarrollar la aplicación usamos el lenguaje python y replit para trabajar de forma colaborativa.

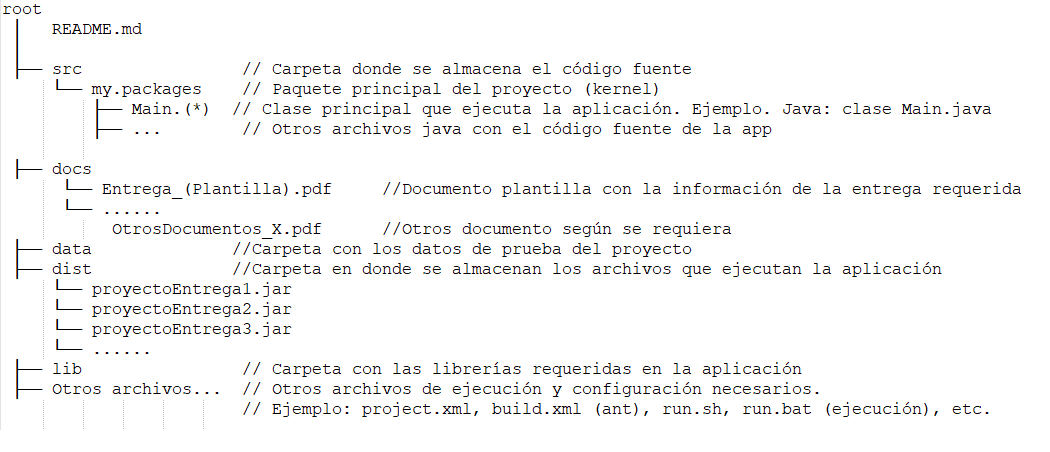
# PROTOTIPO DE SOFTWARE INICIAL

Para esta entrega de avance en el desarrollo del proyecto, se debe realizar una primera versión de un prototipo de software funcional de acuerdo con los requisitos que se explican a continuación.

El software desarrollado se debe registrar en un repositorio de software de Github. Para facilitar el uso de esta plataforma, se sugiere estudiar el tutorial disponible en

<https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud>

Se debe organizar el software en el repositorio de una manera estructurada como se ilustra en la siguiente figura:



Además, para mantener una versión gráfica de desarrollo del repositorio, ustedes se podrán apoyar en el uso de herramientas como Sourcetree, disponible en el siguiente URL:

<https://www.sourcetreeapp.com/>

En este prototipo se deben implementar y aplicar algunas estructuras de datos como se detalla en la siguiente sección.

# IMPLEMENTACIÓN Y aplicación de las estructuras de datos

En este prototipo se debe implementar por lo menos una instancia de cada una de las estructuras de datos secuenciales (llamadas también lineales o unidimensionales): arreglos, listas encadenadas, pilas y colas. Se reitera qué en este tipo de contenedores de datos, los datos se organizan de una manera secuencial o lineal (como en una hilera) de tal manera que cada dato puede tener a lo sumo un antecesor y a lo sumo un sucesor.

Para estas estructuras de datos se deben soportar por lo menos las siguientes operaciones funcionales:

* Creación
* Inserción de un solo dato
* Actualización de un solo dato
* Eliminación de un solo dato
* Búsqueda de un solo dato
* Consulta de todos los datos
* Almacenamiento de los datos

Para cada una de las diferentes estructuras de datos implementadas se debe describir brevemente el tipo de implementación realizada y su contribución para llevar a cabo las funcionalidades del prototipo de software.

# PRUEBAS DEL PROTOTIPO Y ANÁLISIS COMPARATIVO

Se deben realizar y documentar las pruebas del prototipo para algunos ejemplos (casos) de prueba para las funcionalidades que tomen más tiempo y realizar un análisis comparativo así:

* Escoger entre tres y cinco funcionalidades que sean las de mayor costo computacional en tiempo;
* Para cada funcionalidad se deben realizar pruebas para varios tamaños de datos de prueba (n), por lo menos para los siguientes valores de n:
  + 10 mil datos,
  + 100 mil datos,
  + 1 millón de datos,
  + 10 millones de datos, y
  + 100 millones de datos.
* Hacer una tabla comparativa de los tiempos que toma realizar cada una de las funcionalidades consideradas para los diferentes tamaños de los datos de prueba.
* Determine y grafique el correspondiente análisis asintótico comparativo entre las estructuras implementadas y su respectiva complejidad, de acuerdo con las pruebas realizadas. Para esto debe usar, por lo menos, la notación O grande (*Big O*).

# roles y actividades

| **INTEGRANTE** | **ROL(ES)** | **ACTIVIDADES REALIZADAS** |
| --- | --- | --- |
| Rojas Luna  Juan Manuel | Líder | Documento técnico  Mockups |
| Pinzón Garcia Miguel Angel | Técnico | Documento técnico  Mockups |
| Olaya Arias  Pablo Olaya | Técnico | Realización del software |
| Quintero Castillo  Carlos Fernando | Técnico | Realización del software |

# DIFICULTADES Y LECCIONES APRENDIDAS

Mencione las dificultades encontradas durante el desarrollo del proyecto. Además, haga alusión a las principales lecciones aprendidas durante el proceso.

# REFERENCIAS BIBLIOGRÁFICAS

1. Weiss, M.A.: *Data Structures and Algorithm Analysis in C++, 4th Edition*, Pearson/Addison Wesley, 2014.
2. Hernández, Z.J. y otros: *Fundamentos de Estructuras de Datos. Soluciones en Ada, Java y C++*, Thomson, 2005.
3. Shaffer, Clifford A.: *Data Structures and Algorithm Analysis in C++, Third Edition*, Dover Publications, 2013. ([En línea](http://people.cs.vt.edu/shaffer/Book/).)
4. Campos Laclaustra, J.: *Apuntes de Estructuras de Datos y Algoritmos*, segunda edición, 2018. ([En línea](https://webdiis.unizar.es/asignaturas/EDA/apuntes_EDA.pdf).)
5. Martí Oliet, N., Ortega Mallén, Y., Verdejo López, J.A.: Estructuras de datos y métodos algorítmicos: 213 ejercicios resueltos. 2ª Edición, Ed. Garceta, 2013.
6. Joyanes, L., Zahonero, I., Fernández, M. y Sánchez, L.: Estructura de datos. Libro de problemas, McGraw Hill, 1999.

1. [↑](#footnote-ref-0)