



## **Informe: Proyecto Final Algorítmica I**

Integrantes: Pablo Acker Mancilla - 61203

Jhosias Laguna Noriega - 60948

Emanuel Lozano Leaño - 55678

Materia: Algorítmica I

Docente: Ing. Paul Landaeta

Fecha de entrega: 16/05/22

**La Paz - Bolivia**

## Índice:

1.Introducción.....	3
2.Planificación de proyecto.....	4
3.Proceso de instalación. ....	4
Requisitos de software .....	4
1.- Crear Carpeta.....	6
2.- Importar Archivos .....	6
3.- Ejecutar el programa .....	8
4.Definición del Problema. ....	8
5.Explicación del algoritmo.....	9
6.Conclusión.....	10

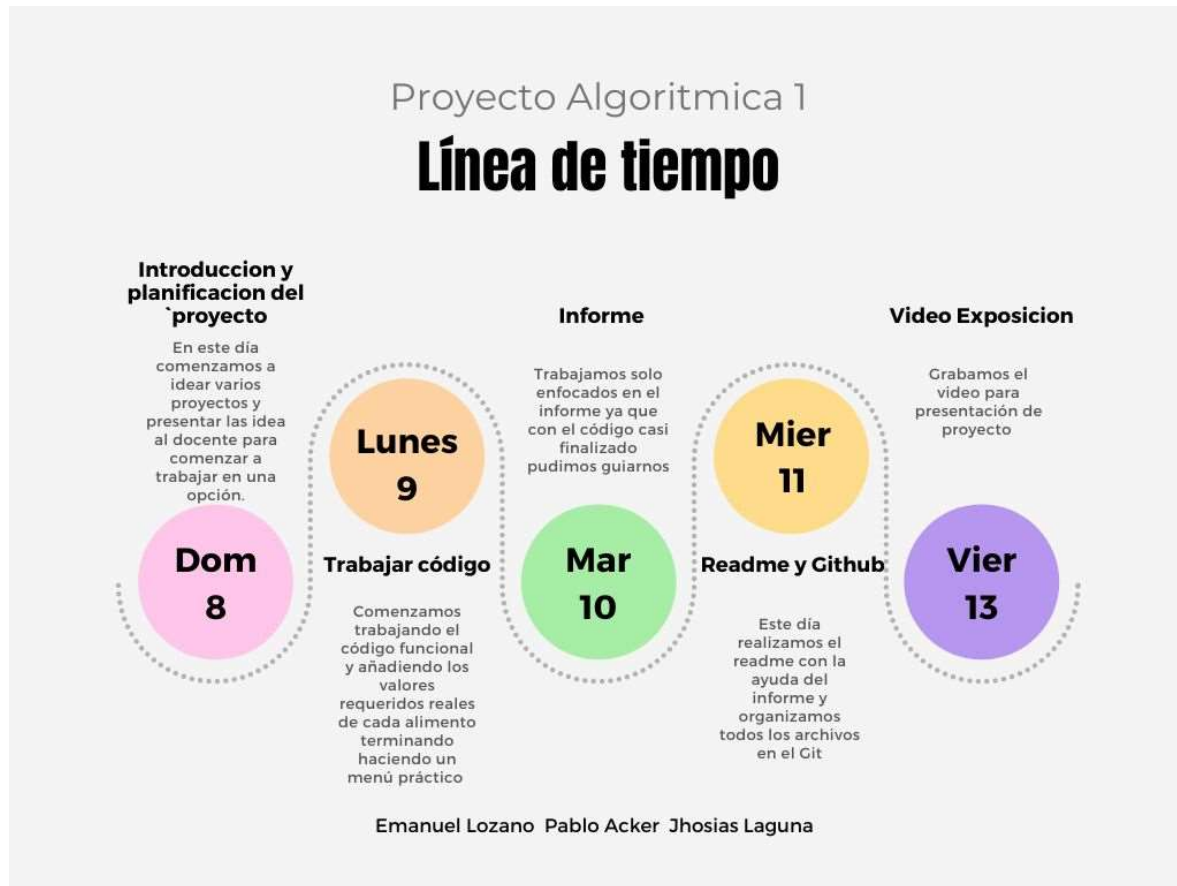
## **1.Introducción.**

El presente trabajo tiene como fin brindar orientación en el tema de las dietas de una persona y los diferentes costos que presenta un producto para encontrar la mejor relación ganancia-precio, para alcanzar la meta del usuario.

En este trabajo se seleccionó el lenguaje de programación C++. Este informe contiene un timeline de la planificación de como se distribuyó el trabajo a cada uno de los integrantes del grupo para su respectiva contribución en el mismo, también contiene el proceso de instalación del archivo del proyecto paso a paso, además de las instrucciones para ejecutar dicho programa (teniendo como datos, los requisitos de software para 32 y 64 bits).

Siguiendo con el contenido de este trabajo, se tiene la definición del problema más a detalle y la explicación del algoritmo utilizado para resolverlo.

## 2. Planificación de proyecto.



## 3. Proceso de instalación.

Primeramente, se debe tener en cuenta que el programa fue realizado en el lenguaje de programación C++, además este programa fue diseñado en el editor Visual Studio Code. Se recomienda utilizar este editor por la comodidad que ofrece y las distintas extensiones que tiene de por medio, además de la facilidad que otorga manejar distintos lenguajes de programación en un mismo editor. No obstante, el usuario puede manejar el editor de su preferencia.

### **Requisitos de software**

Los requisitos esenciales para poder ejecutar el programa son los siguientes:

#### **1.- Tener instalado un compilador para C++ (DEV C++)**

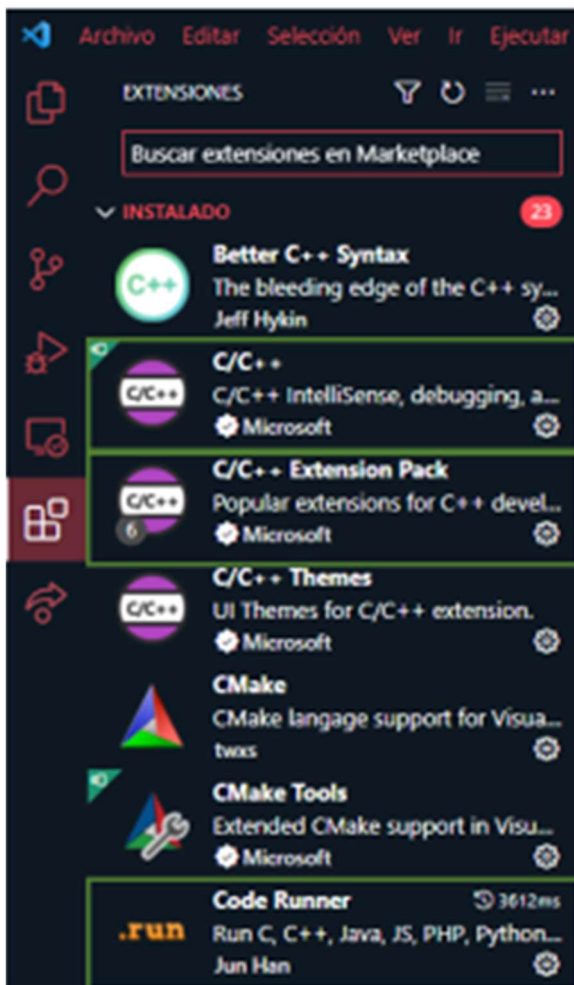
Si no tiene instalado el compilador puede observar el siguiente video para su correcta instalación: [https://www.youtube.com/watch?v=0B\\_Ly0A3RV0](https://www.youtube.com/watch?v=0B_Ly0A3RV0)

## 2.- Tener un editor funcional que acepte C++

En este caso Visual Studio Code, si no tiene instalado un editor puede observar el siguiente video para su correcta instalación: [https://www.youtube.com/watch?v=X\\_Z7d04x9-E](https://www.youtube.com/watch?v=X_Z7d04x9-E)

## 3.- Instalar extensiones

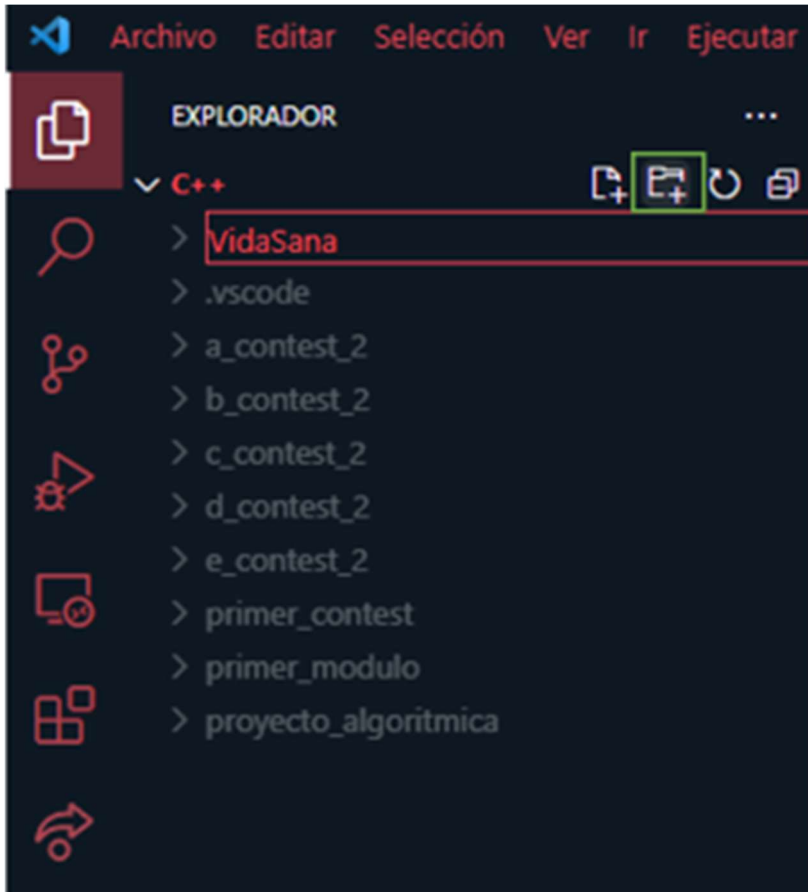
Una vez instalado VS Code debe instalar las siguientes extensiones: C/C++, C/C++ Extensión Pack, Code Runner. Estas extensiones se pueden instalar desde el menú de acciones del lado izquierdo del editor VS Code.



Continuando con el proceso de instalación debemos realizar las siguientes acciones:

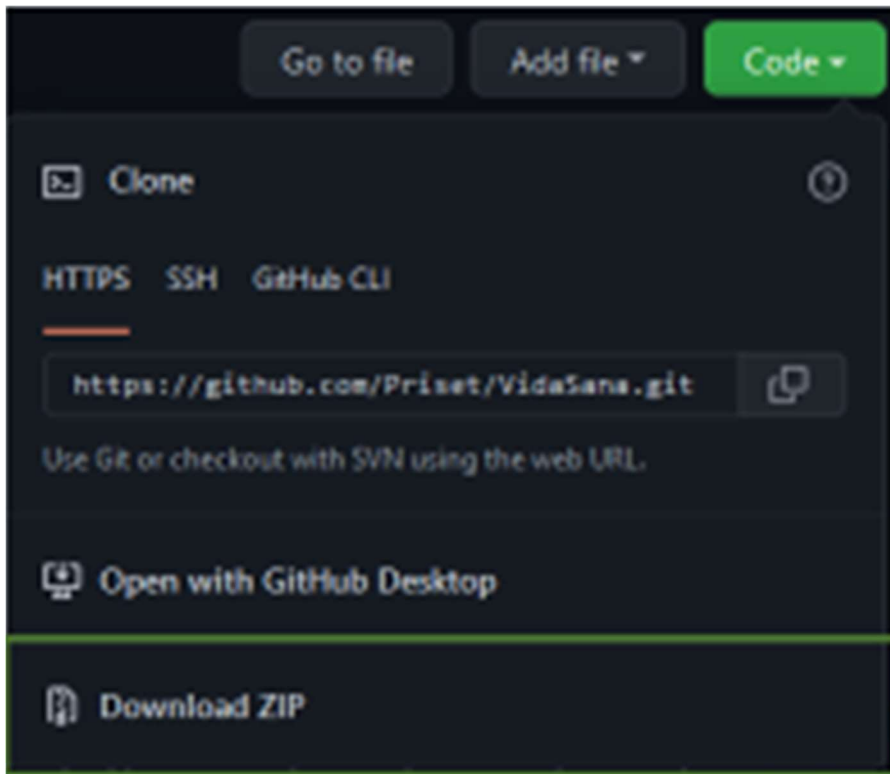
## 1.- Crear Carpeta

Se debe crear una carpeta en donde se almacenarán los archivos del programa asignándole el nombre que desee.



## 2.- Importar Archivos

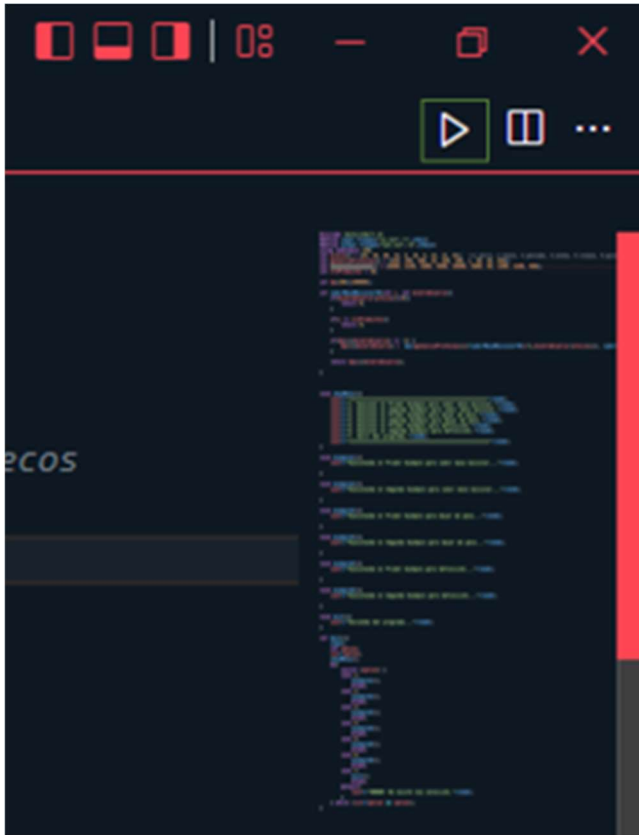
Se debe descargar el archivo ZIP del repositorio de GitHub, seguidamente en el explorador de archivos buscamos el directorio de nuestra carpeta ya creada y dentro de esta descomprimos el archivo ZIP. Terminados una vez estos pasos nos aparecerán los archivos del programa en el editor (VS Code).



Una vez terminados todos estos pasos tendremos listo el programa para ejecutarse. (Todas estas instrucciones fueron evaluadas para un Sistema Operativo de 64 bits, si usted tiene un Sistema operativo de 32 bits puede revisar los siguientes links: <https://www.youtube.com/watch?v=RfebFlu2CcM> , [https://www.youtube.com/watch?v=5grPpOZSr\\_c](https://www.youtube.com/watch?v=5grPpOZSr_c), para VS Code y el compilador de C++ respectivamente luego se podrían seguir los pasos de manera análoga).

### 3.- Ejecutar el programa

Una vez que se tenga todo funcional, se puede proceder a compilar el programa, primero se va al archivo vida\_sana.cpp y ejecutamos el programa en la flecha de compilación al lado superior derecho del editor (VS Code).



Finalmente podremos observar el resultado en la terminal del editor VS Code e introducir y podremos ingresar las opciones del 1 al 6 (el 7 es para salir del programa).

### 4. Definición del Problema.

Para el desarrollo del programa se planteó el siguiente problema, un usuario quiere alcanzar una meta física implementando una dieta, teniendo como opciones tener una ganancia de proteínas, carbohidratos o calorías, pero solo se tiene cierta cantidad de dinero para poder obtener los resultados deseados. Entonces lo que se busca resolver con el programa es poder hallar la mejor dieta que otorgue proteínas para subir masa muscular, alta en calorías para aumentar volumen y alta en



carbohidratos para subir de peso, así mismo, encontrar el precio más económico, teniendo una relación de ganancia-precio. Tenemos solamente 10 productos que estamos tomando en cuenta para realizar este análisis, los cuales son: Pollo, huevo, pescado, avena, linaza, quinoa, té verde, yogurt griego, palta, y frutos secos. Cada uno de estos productos tienen su respectiva valoración en los parámetros de calorías, proteínas y carbohidratos. Así mismo se desea calcular el mínimo de productos que se pueden comprar con el dinero que tenga el usuario, para su mejor comprensión.

## **5.Explicación del algoritmo.**

Primeramente, en este proyecto se ha usado el algoritmo Knapsack, que viene de uno de los varios algoritmos que se ven en programación dinámica, este código tiene la similitud del problema de la moneda (CoinChange) utilizando el mínimo entre las opciones dadas, y el problema de la mochila que utiliza el máximo entre sus opciones. En este caso, el algoritmo Knapsack fue implementado con Top Down. Explicando la idea en general de este algoritmo, se puede observar que es un problema de optimización combinatoria, es decir, que busca la mejor solución entre un conjunto finito de posibles soluciones a un problema (en este caso las máximas y mínimas posibilidades). Modela una situación análoga al llenar una mochila, incapaz de soportar más de un cierto peso límite, la idea es llenar dicha mochila con distintos objetos, cada uno con un peso y valor específicos. Los objetos colocados en la mochila deben ser los que tengan el mayor valor posible, obviamente sin exceder el peso máximo. Ahora bien, este algoritmo aplicado a nuestro código funcionaría de la siguiente manera: Con el algoritmo Knapsack se halla el máximo de proteínas, calorías y carbohidratos, cada uno con su propio código y con un propio arreglo dp, en cada uno de estos se tiene el dinero del usuario, que representaría el peso total de la mochila, el precio de los alimentos, los cuales se pueden comprar con dicho dinero del usuario, y la ganancia ya dicha, en proteínas, calorías y carbohidratos. También se tiene una cantidad N de productos que hacen referencia a la cantidad N de objetos que se pueden meter en la mochila, en este caso, N es igual a 10. Ahora, para el algoritmo CoinChange, saca el mínimo

de productos que se pueden comprar con el dinero del usuario, dicho código devuelve INF (un número muy grande simulando el Infinito) cuando no hay solución para el mínimo de productos.

## **6.Conclusión.**

Para finalizar este proyecto se puede decir que el precio de nuestra alimentación puede variar demasiado dependiendo nuestra meta, con este algoritmo podríamos escoger de acuerdo a nuestras necesidades y posibilidades lo más adecuado y accesible para una persona de cualquier estatus social. Por lo tanto, se ha llegado a la conclusión que este algoritmo nos ayuda a resolver problemas utilizando Knapsack siendo el proceso muy eficiente debido a la facilidad que nos otorga este, y por lo tanto es uno de los algoritmos más utilizados. Así mismo se usó programación dinámica para tener resultados más precisos y óptimos además de la obtención de datos adicionales que nos ayudan a desarrollar mejor la solución al problema. Obteniendo de este modo un resultado exitoso con el programa "vida\_sana" y resolviendo todo el problema planteado anteriormente.