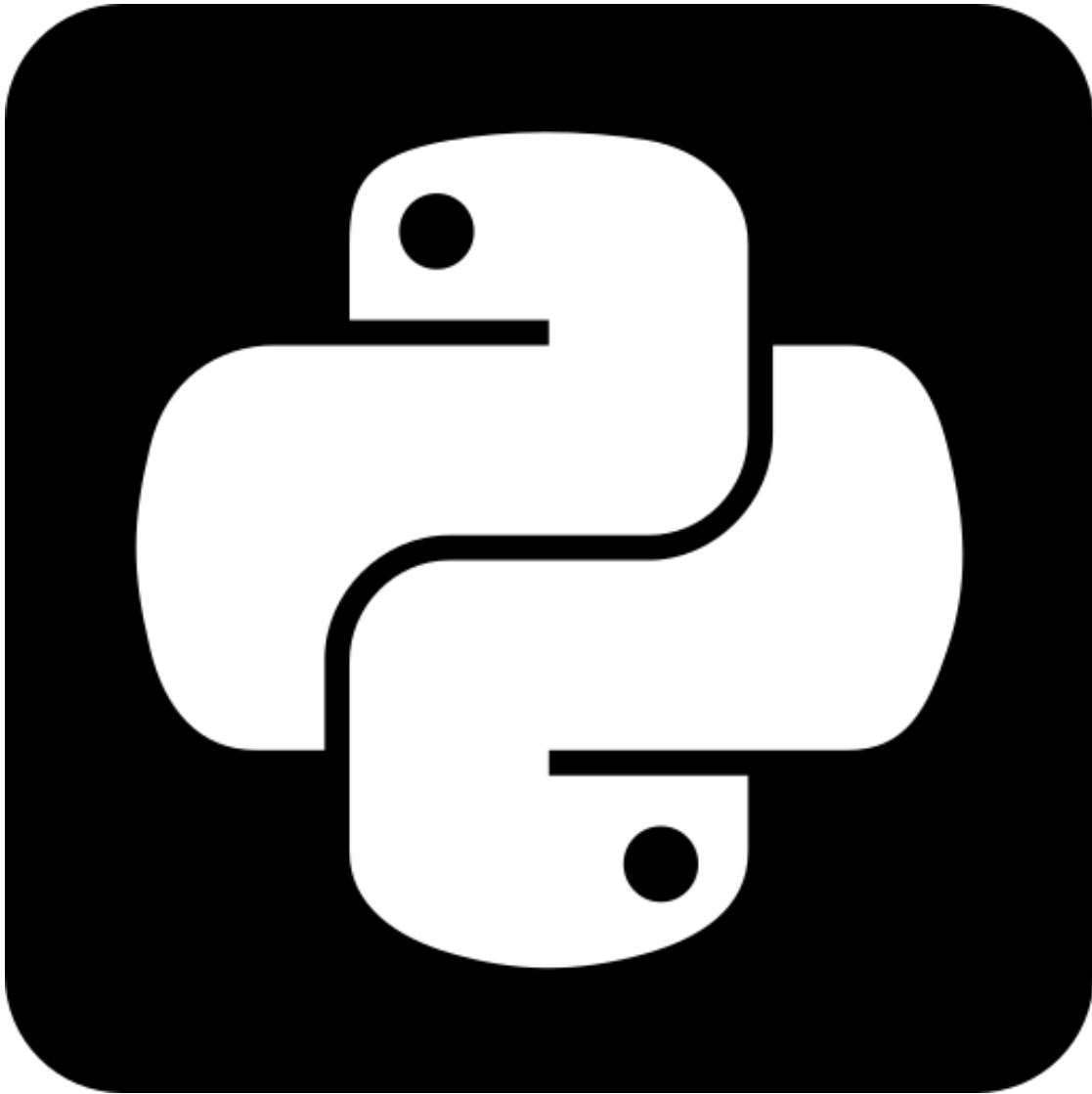


Proyecto - FDP



Equipo N. 2

Integrantes:

- **Ian Enrique Frausto Cervantes (Senior)**
- **Leonardo Pérez Molina (Junior)**
- **Nallely Maldonado Ramírez (Junior)**
- **Bryan Alexis Molina Loya (Mid)**
- **Oscar Gabriel Reyes Delgado (Mid)**

Contexto y problema:

Escenario:

El escenario planteado para nuestro proyecto es dentro de una empresa para la realización de un programa con el fin de que facilitemos las actividades relacionadas con la toma de un inventario, como facilitar los datos ingresados dentro de una lista para el conocimiento de los materiales disponibles, su cantidad actual, el costo de cada uno y el presupuesto total de todos y cada uno de ellos, con el fin de hacer todo de una manera fácil de entender, concisa y práctica.

Problemas a resolver:

Nuestra problemática a resolver son las siguientes:

- Captura de datos de manera fácil y optima para registrarlos dentro de una lista para un conocimiento de los mismos.
- Un costo total dentro de todos los materiales listados para una cotización y control de los mismos.
- Mostrar los datos capturados de forma clara para el mantenimiento de los mismos en el caso de ingresar más material a los ya existentes o una reducción de ellos.

Criterios de éxito:

El criterio de éxito de nuestro programa es que el mismo muestre todos los datos capturados de forma clara y concisa para facilitar la comprensión del lector, y optimizar procesos y el control del inventario registrado para disminuir perdidas y facilitar la captura de la nueva información entrante.

Reglas del negocio:

Las reglas dentro del negocio para un funcionamiento correcto del programa es dentro de las preguntas que realizara el programa es ingresar valores claves y claros, para no alterar los valores que puede interpretar el programa.

Requerimientos funcionales:

- Solicitar el nombre de los materiales
- Solicitar las especificaciones del material
- Solicitar las cantidades de los materiales ingresados
- Solicitar el costo de los materiales ingresados
- Validar los valores ingresados
- Guardar todos los datos ingresados
- Calcular el costo de todos los materiales
- Distribuir los materiales en cuestión de su tipo
- Opciones para cancelar el sistema

Diseño de entradas y salidas:

Entradas:

- Nombre del material
- Tipo que especifica el material
- Cantidad del material
- Costo del material
- Valores correctos de los materiales (Nombre, cantidad y costo)
- Guardar los datos ingresados
- Opciones de cancelar el programa

Salidas:

- Listado de nombre, tipo de material, cantidad y costo de cada uno
- Cotizar el costo total de todos los materiales
- Mostrar mensajes en caso de ingresar valores no permitidos
- Volver a solicitar valores en caso de errores

Diagrama de flujo:

<https://github.com/KianFrausto/Proyecto-FDP/blob/8227d230873adf4b8afd7b8cd459651c4b30ecf6/docs/Inventario.jpg>

Plan de trabajo:

Semana 1:

- [✓] Planteamiento del problema a realizar. (Todos)**
- [✓] Distribución de los roles. (Senior)**
- [✓] Realización del GitHub (main) para uso del equipo. (Senior)**
- [✓] Realización de los commits dentro del repositorio de GitHub (Todos)**

Semana 2:

- [-] Realización del pseudo-código para integración completa del proyecto a realizar. (Senior)**
- [-] Actualización a una versión principal del código con funciones completas. (Mids)**
- [-] Verificación del código base en busca de posibles errores y optimización de entradas y salidas. (Juniors)**
- [-] Validación de las ramas creadas para su integración dentro del código principal. (Senior)**

Semana 3:

- [-] Realización de actualizaciones dentro del programa para futuras versiones. (Mids)**
- [-] Ajustar variables y validación de las funciones nuevas del programa. (Juniors)**
- [-] Ajuste de ramas nuevas para validación de las funciones dentro de la rama principal y validar un funcionamiento correcto. (Senior)**

Semana 4:

- [-] Reajuste de nuevos comandos para parches dentro de las versiones anteriores. (Mids)**
- [-] Simplificar funciones dentro del código para un funcionamiento más optimo. (Junior)**
- [-] Integración para una versión final del código. (Senior)**

Semana 5:

- [-] Ultimas actualizaciones dentro del programa. (Mids)**
- [-] Revisión a detalle del código completo por posibles errores en el código. (Juniors)**
- [-] Entrega del código en una fase completa lista para su ejecución sin errores. (Senior)**