**ALGORITMOS Y ESTRUCTURAS DE DATOS**

**TDSD**

|  |  |
| --- | --- |
|  | |
| ASIGNATURA: | ALGORITMOS Y ESTRUCTURAS DE DATOS |
| PROFESOR: | Ing. Lorena Chulde |
| PERÍODO ACADÉMICO: | 2023-B |
|  | |

**PRUEBA – BIMESTRE 2**

|  |  |
| --- | --- |
|  | |
| Nombre del estudiante:  Guerra Lovato Josué Eduard | |
|  | |
|  | |
| Algoritmos de Búsqueda en Python - Mi Diario Python |  |
|  | |

**2023-B**

**PARTE PRÁCTICA**

Actualmente “**POLICUPIDO**” es un emprendimiento dedicado a realizar cualquier tipo de detalles sorpresas para entregar a esa persona especial de manera anónima. No obstante, para armar un regalo hay que tomar en cuenta que el cliente debe seleccionar una opción:

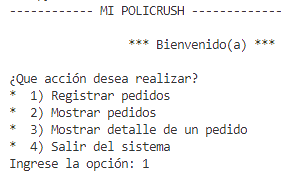
|  |  |
| --- | --- |
| **DETALLE** | **COSTO** |
| Poliflor + Polipeluche | $2,50 |
| Poliflor + Policarta | $1,50 |
| Poliflor + Polillavero | $2,00 |
| Poliflor + Polivaso | $2,75 |

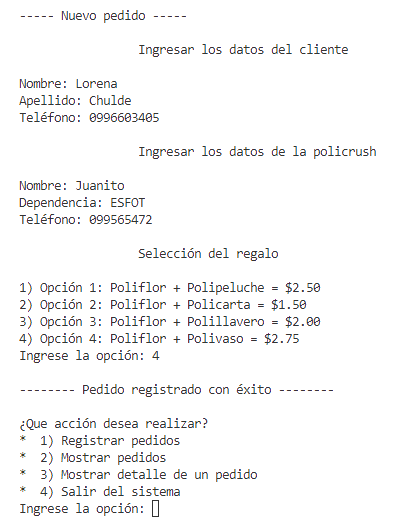
Recuerda que, el detalle que el cliente solicita lleva material extra que le permite a la sorpresa verse como un regalo premium, por lo cual se debe agregar al precio final un 10% extra.

Con base a los requisitos otorgados por el dueño del emprendimiento, la tarea consiste en desarrollar un sistema de gestión de pedidos (CRUD) de todos los clientes que van a requerir contratar un servicio de “**SE MI POLICRUSH**” en el cual se presente un menú de opciones con la siguiente información:

1. El momento de ingresar al sistema el mismo debe tener un menú con 3 opciones para que el administrador pueda realizar las acciones que requiera.

* Si el administrador ingresa la **opción 4** debe salir del sistema.

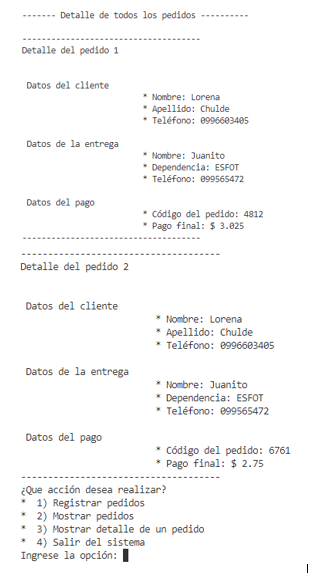


1. Si el administrador ingresa en la **opción 1** el sistema debe permitir:
   1. Registrar nuevos pedidos.

**0,5**

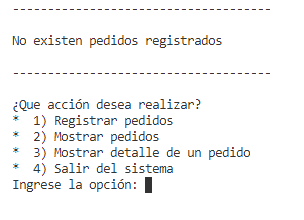
**0,5**

* 1. Si el administrador nuevamente ingresa a la opción 1, se deben registrar nuevos pedidos a los ya existentes.



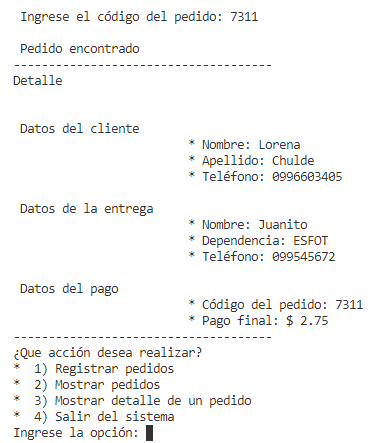
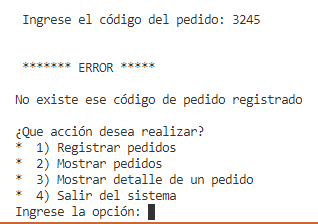
1. Si el administrador, ingresa la **opción 2** debe permitir:
   * + - 1. Visualizar todos los pedidos que se han registrado en la opción 1.

**0,5**



* + - * 1. Y en el caso de que no existan pedidos registrados, se debe mostrar un mensaje que mencione que no existen pedidos registrados y por supuesto le presente otra vez el menú.

**0,5**

1. Si el administrador, ingresa la **opción 3** debe visualizar el detalle de un pedido en particular en base a un código generado por el sistema, caso contrario mostrar un mensaje de error.

**1,0**



**Importante:**

Recuerda utilizar funciones.

Para generar números aleatorios

usa la librería random:

import random

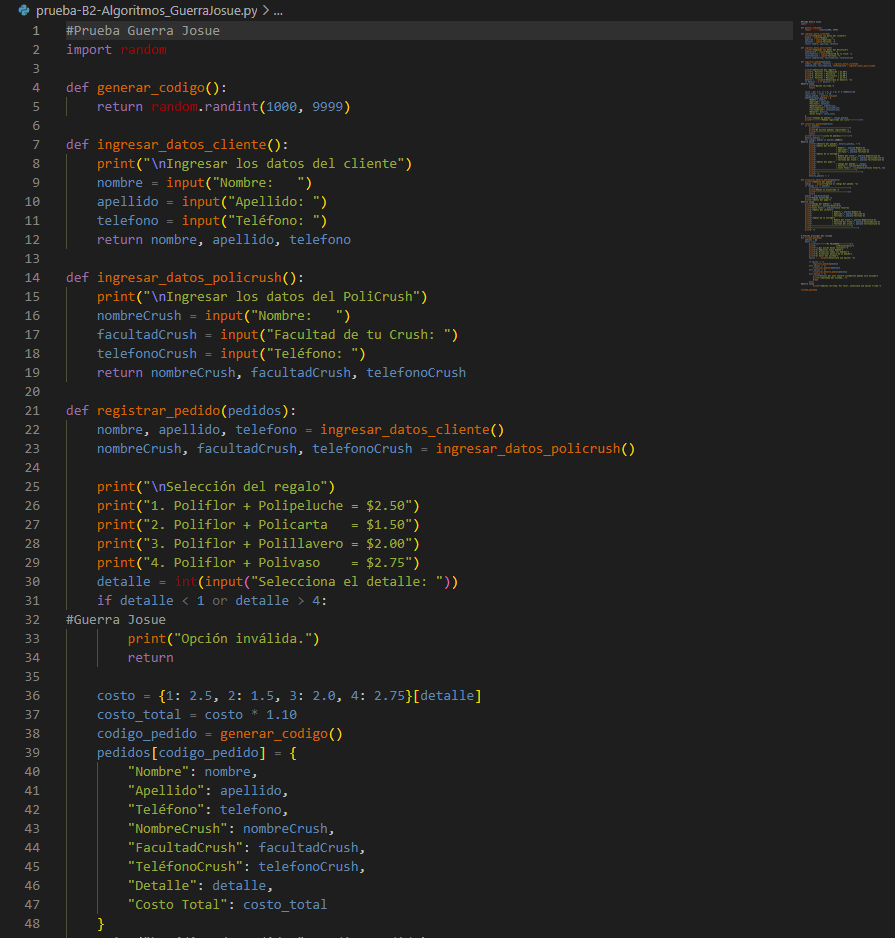
**ENTREGABLES:**

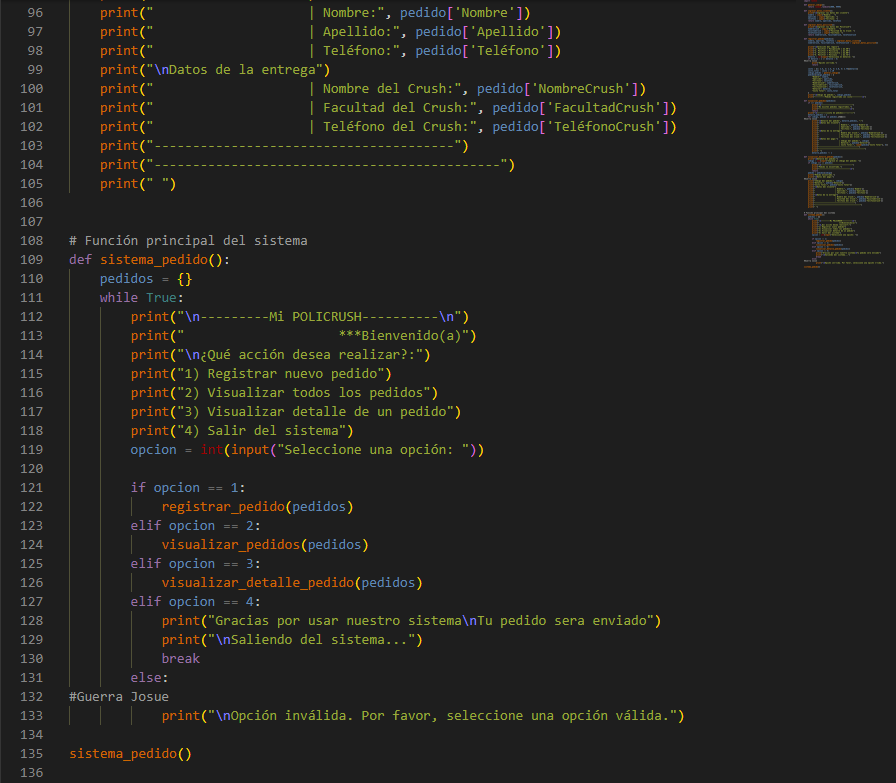
Al finalizar la prueba usted deberá entregar todos los archivos comprimidos en un paquete zip o rar. con lo siguiente:

1. Este formato en pdf, con las capturas de las pantallas de la ejecución del programa
2. El archivo .py del código fuente
3. Poner todo en una carpeta y comprimirla con su apellido y nombre

**Presentación del código**

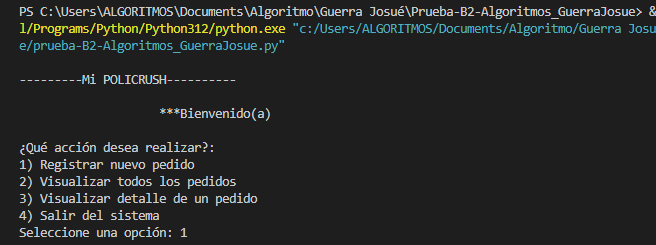
**CÓDIGO INICIAL**

****



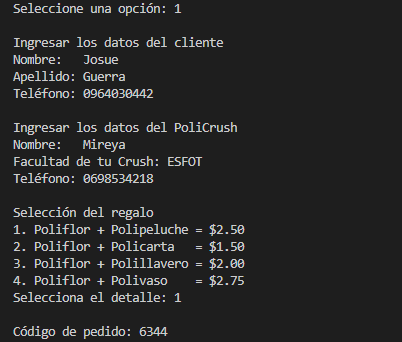
**Ejecución del código**

El momento de ingresar al sistema el mismo debe tener un menú con 3 opciones para que el administrador pueda realizar las acciones que requiera.

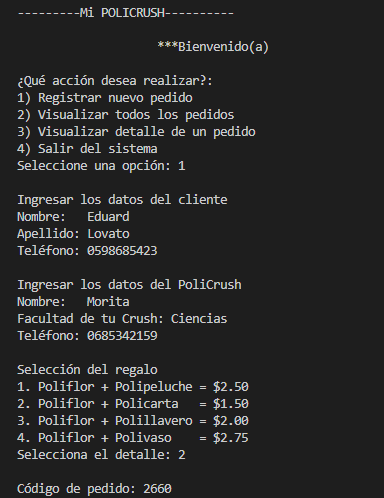
****

Si el administrador ingresa en la **opción 1** el sistema debe permitir:

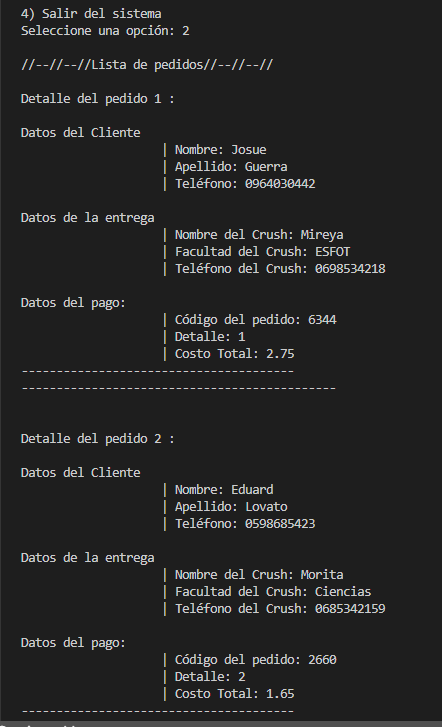
* 1. Registrar nuevos pedidos.

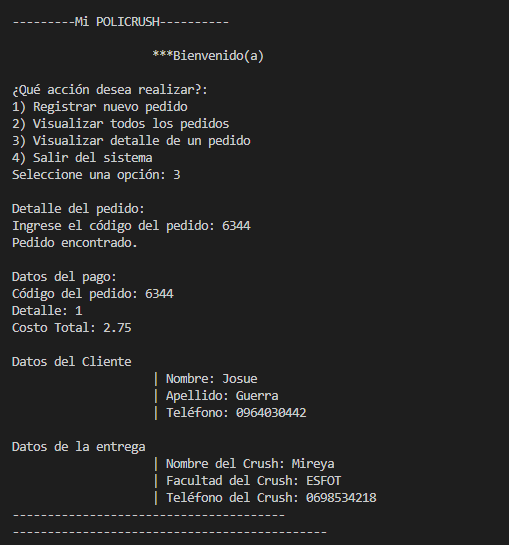


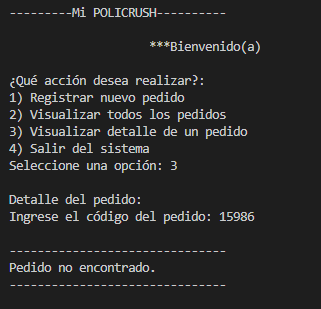
* 1. Si el administrador nuevamente ingresa a la opción 1, se deben registrar nuevos pedidos a los ya existentes.



* 1. Si el administrador, ingresa la **opción 2** debe permitir:
  2. Visualizar todos los pedidos que se han registrado en la opción 1.



Si el administrador, ingresa la **opción 3** debe visualizar el detalle de un pedido en particular en base a un código generado por el sistema, caso contrario mostrar un mensaje de error.



El usuario escoge la opción 4 el programa terminara automáticamente.

