**Herramientas Computacionales**

**Parcial – Punto 2**

*Isabella Victoria Henríquez, cód. 8961991*

*Manuel Cortés González, cód. 8962732*

**Desarrollo punto 2 del parcial**

1. **¿Qué tipos de errores se presentaron o se pueden presentar?**

Se presentaron pocos errores de sintaxis, como la falta de la función **len**() para llevar a cabo un ciclo for con rango que recorre una lista.

A screenshot of a computer

Description automatically generated with medium confidence

En cuanto a errores lógicos, se presentó la equivocación de dividir el precio del producto por el porcentaje de descuento y luego multiplicarlo por 100, lo que nos resultaba en un valor que excedía incluso el precio de los productos sin descuento y terminaba por mostrar un valor a pagar negativo. Adicional a esto posible que se pueda cometer alguno durante la operación del ciclo y sus condicionales si no se tiene una idea clara de estructuración del código.

Finalmente, y los más posibles, son los errores de ejecución; en primer lugar está la digitación del rol, esta se puede hacer en caracteres que no estén en minúscula, se puede cometer un “mistype” o incluso puede que se salte el proceso de digitar el rol y esto genere error. Por otro lado está la posibilidad de ingresar un número negativo en la sección de precios, digitar un número decimal y no un entero, no digitar o ingresar caracteres que no sean número e incluso digitar un precio erróneo y que este se agregue al total sin la posibilidad de revertir el proceso.

1. **¿Qué estrategias podrías utilizar para solucionar estos errores?**

Para los errores de sintaxis podemos identificar los errores en el depurador del IDE, eliminar la causa o corregir el código.

Por otra parte, para los errores de lógica podemos practicar la vuelta atrás, enfocarnos en la sección del código en donde creemos que hay un error e identificar las causas o crear un modelo computacional para el pedazo de código que falla.

Finalmente, para los errores de ejecución, podemos hacer más robusto el programa, implementar pequeñas funciones o pequeños algoritmos que impidan que el programa “crashee” y podemos modelar el sistema y sus posibles estados a través de una máquina de estado o una máquina de Turing.