**Evaluación Parcial N°3**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sigla** | **Nombre Asignatura** | **Tiempo Asignado** | **% Ponderación** |
| FPY1101 | Fundamentos de Programación | 3 h | **30%** |

**1. Situación evaluativa**

|  |  |
| --- | --- |
| x | **Ejecución práctica** |

**2. Agente evaluativo**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| x | **Heteroevaluación** |  |  | **Coevaluación** |  |  | **Autoevaluación** |

**3. Tabla de Especificaciones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Resultado de Aprendizaje** | **Indicador de Logro (IL)** | **Indicador de Evaluación (IE)** | **Ponderación Indicador Logro** | **Ponderación Indicador de Evaluación[[1]](#footnote-1)** |
| RA3 Construye estructuras de datos temporales, utilizando arreglos para almacenar datos, con el fin de darle persistencia a éstos y trabajar con problemas que necesiten guardar información temporalmente. | IL 3.1 Identifica colecciones y arreglos que permitan el almacenamiento de datos según los requerimientos del problema planteado. | IE 3.1.1 Identifica colecciones y arreglos que permitan el almacenamiento de datos según los requerimientos del problema planteado. | 20% | 20% |
| IL 3.2 Utiliza arreglos y matrices para la inserción, eliminación, modificación y búsqueda de datos temporales para cumplir con los requerimientos del problema planteado. | IE 3.2.1 Utiliza arreglos y matrices para la inserción, eliminación, modificación y búsqueda de datos temporales para cumplir con los requerimientos del problema planteado. | 20% | 20% |
| IL 3.3 Utiliza archivos para logar la persistencia de los datos de la aplicación según el caso planteado. | IE 3.3.1 Utiliza archivos para logar la persistencia de los datos de la aplicación según el caso planteado. | 10% | 10% |
| RA4 Modulariza programas, mediante funciones con y sin parámetros, para simplificar y darle eficiencia a las soluciones construidas. | IL 4.1 Utiliza librerías de sistema para la optimización del código según el caso planteado. | IE 4.1.1 Utiliza librerías de sistema para la optimización del código según el caso planteado. | 15% | 15% |
| IL 4.2 Programa funciones que permitan la reutilización de código según el problema planteado. | IE 4.2.1 Programa funciones que permitan la reutilización de código según el problema planteado. | 15% | 15% |
| IL 4.3 Integra funciones invocadas desde el programa principal para dar solución al problema planteado. | IE 4.3.1 Integra funciones invocadas desde el programa principal para dar solución al problema planteado. | 10% | 10% |
| IL 4.4 Utiliza un sistema de control para administrar las versiones código (GitHub) en el desarrollo de una aplicación. | IE 4.4.1 Utiliza un sistema de control para administrar las versiones código (GitHub) en el desarrollo de una aplicación. | 10% | 10% |
| **Total** | | | **100%** | **100%** |

**4. Instrucciones generales para el/la estudiante**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Desarrolla un programa en Python para un sistema de ventas de pizzas en un local de comida en DUOC UC. El programa debe permitir a los usuarios realizar las siguientes acciones a través de un menú interactivo:   1. Registrar una venta. 2. Mostrar todas las ventas. 3. Buscar ventas por cliente. 4. Guardar las ventas en un archivo. 5. Cargar las ventas desde un archivo. 6. Generar Boleta 7. Salir del programa.   **Requisitos:**   * **Menú Interactivo:** El programa debe presentar un menú con las opciones mencionadas y permitir al usuario seleccionar una opción. * **Listas y Diccionarios:** Utilizar listas y diccionarios para almacenar la información de las ventas. * **Archivos:** Implementar la **funcionalidad para guardar l**as ventas en un archivo y cargar las ventas desde un archivo. * **Funciones:** Utilizar funciones para cada una de las funcionalidades del menú. * **Descuentos:** Aplicar descuentos según el tipo de usuario:   + Estudiante diurno: 15%   + Estudiante vespertino: 20%   + Administrativo: 10%  |  |  |  | | --- | --- | --- | | **Tipo de Pizza** | **Tamaño** | **Precio** | | Peperoni | Pequeña | 5.000 | | Peperoni | Mediana | 8.000 | | Peperoni | Familia | 10.000 | | Mediterranea | Pequeña | 6.000 | | Mediterranea | Mediana | 9.000 | | Mediterranea | Familia | 12.000 | | Vegetariana | Pequeña | 5.500 | | Vegetariana | Mediana | 8.500 | | Vegetariana | Familia | 11.000 |   El sistema debe también considerar el tipo de pizza (Peperoni, Mediterránea, Vegetariana) y el tamaño (Pequeña, Mediana, Familiar) al registrar una venta.   * **Boleta:** generar una boleta con fecha y hora en el sistema de ventas de pizzas   Guardar proyecto en github y enviar link por ava.  Enviar en archivo Python .py o texto .txt como Nombre\_Apellido\_ FPY1101\_seccion.py o .txt  Ejemplo( Josue\_Oteiza\_FPY1101\_003V.py o .txt) |

1. La ponderación del Indicador de Logro, debe ser distribuida como puntaje entre las preguntas que lo conformen. [↑](#footnote-ref-1)