**Evaluación Parcial N°3**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sigla** | **Nombre Asignatura** | **Tiempo Asignado** | **% Ponderación** |
| FPY1101 | Fundamentos de Programación | 3 bloques académicos | **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 de código (GitHub) en el desarrollo de una aplicación. | IE 4.4.1 Utiliza un sistema de control para administrar las versiones de código (GitHub) en el desarrollo de una aplicación. | 10% | 10% |
| **Total** | | | **100%** | **100%** |

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

|  |
| --- |
| Esta es una evaluación que corresponde a una prueba de ejecución práctica y tiene un **30%**deponderaciónsobre la nota final de la asignatura.  El **tiempo** para desarrollar esta evaluación es de **120 minutos** y se realiza de manera **individual** en **laboratorio**  **La evaluación consiste en:**   * Desarrollo de una aplicación en Python y en entorno de desarrollo Visual Studio Code * Uso de colecciones * Uso de archivos de texto * Uso de librerías estándar de Python * Uso de GitHub como repositorio del pryecto |

**5. Evaluación**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Desarrolle una aplicación en Python utilizando Visual Studio que permita resolver el siguiente caso:  La distribuidora de agua purificada “CleanWasser”, necesita desarrollar un sistema que permita registrar los pedidos para enviar la hoja de ruta al repartidor. Para el funcionamiento del sistema se requiere las siguientes funcionalidades:   1. Registrar pedido 2. Listar los todos los pedidos 3. Imprimir hoja de ruta 4. Buscar un pedido por ID 5. Salir del programa 6. **Registrar pedido**   Para registrar un pedido se requiere lo siguiente: Nombre y apellido del cliente, comuna, detalle del pedido. Por ejemplo, la empresa vende dispensadores de 6, 10 y 20 litros. Debe permitir ingresar la cantidad de cada cilindro a solicitar y considerar que la suma de las cantidades (de cada cilindro) deben sumar más de cero (para que tenga sentido el pedido).  Por lo tanto, un detalle de pedido podría verse registrado de la siguiente manera:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | ID pedido | Cliente | Dirección | Sector | Disp.6lts | Disp. 10lts | Disp 20lts | | 123456 | Miguel Cortés | Los abedules 742 | Hualpén | 1 | 1 | 0 |   Debe validar que todos los datos sean ingresados y que sean de los tipos de datos correspondientes.   1. **Listar Pedidos**   Debe mostrar en la pantalla la lista de todos los pedidos realizados en un formato similar al ejemplo anterior de registro de pedidos. Al listar se DEBE INCLUIR el encabezado tal y como se muestra en la imagen anterior.   1. **Imprimir hoja de ruta**   Para imprimir la hoja de ruta, el usuario debe seleccionar alguno de los sectores donde es posible realizar un pedido. Estos sectores deben estar previamente definidos en algún tipo de colección de Python en el código, los sectores son: Concepción, Chiguayante, Talcahuano, Hualpén, San Pedro.  Al seleccionar uno de los sectores, se generará un archivo de texto (.csv) con el detalle de los pedidos que se deberá llevar al sector. Este debe tener la misma forma del registro completo de las opciones anteriores, pero en archivo de texto. Este archivo debe contener el encabezado del detalle de pedido.   1. **Buscar un pedido por ID**   Debe pedir al usuario ingresar un id del pedido y mostrar su detalle   1. **Salir del programa**   El programa debe funcionar hasta que el usuario decida salir.  CONSIDERACIONES:   * Recuerde las validaciones implícitas en el problema. * Recuerde utilizar librería para limpiar pantalla haciendo más fácil su visualización y generar números aleatorios para generar el ID de pedido * Cada opción de la aplicación debe desarrollarse en una función que, pudiendo estar en archivos separados o en un mismo archivo, deben se llamadas desde el programa principal. |

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