Evaluación # 3 (35%)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ÁREA ACADÉMICA** | | Tecnologías de Información y Ciberseguridad | | | **CARRERA** | Análisis de Sistemas | |
| **ASIGNATURA** | Introducción a la Programación | | | | | **CÓDIGO** | TI2011/D-IEI-N1-P1-C1 |
| **SEDE** | Apoquindo | | | **DOCENTE** | Rebeca Cartagena Jessen | | |
| **Unidad de Aprendizaje** | | | N°3 | **Criterios a Evaluar** | | 3.1.1 al 3.1.4 | |
| **DURACIÓN** | 120 minutos | | | **FECHA** | | 14-06-2022 | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **NOMBRE ALUMNO:** |  | | | | | |  | | |  |
|  | | | Apellido Paterno | | | Apellido Materno | | | Nombres | |
| **RUT:** | | | **-** | | | | | | | |
| **PUNTAJE MÁXIMO** | | 100 | | | **NOTA:** | | | Firma conforme | | |
| **PUNTAJE OBTENIDO** | |  | | |
| **Solicita re-corrección** | | **Sí** | | **No** | **Motivo:** | | | | | |

|  |
| --- |
| **INSTRUCCIONES GENERALES:**   1. La nota 4.0 se obtiene logrando un 60% del puntaje total. |

|  |
| --- |
| **Aprendizaje esperado**  3.1.- Desarrolla Script que almacenen datos en memoria mediante el uso de arreglos unidimensional, tuplas y diccionarios, incluyendo su anidación recorrido y visualización. (Integrada Competencia Genérica Resolución de Problemas) |

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | **Criterios de evaluación**  3.1.1.- Identifica diferencias entre tuplas, arreglos y diccionarios en función de las capacidades de almacenamiento. | | 3.1.2.- Determina el tipo de dato óptimo para el almacenamiento de datos. | | 3.1.3.- Crea aplicación para búsqueda y ordenamiento de datos dentro de arreglos.  3.1.4.- Crea aplicación para la búsqueda de datos dentro de tuplas y diccionarios | |

**Ítem Desarrollo**

|  |  |
| --- | --- |
| Pregunta # 1 | PUNTAJE:40 puntos |
| Debe desarrollar un programa en Python que permita trabajar con LIST y realizar lo siguiente:  1.- Ingresar una lista de 5 números y guardarlos en una lista (realizarlo desde una función)  2.- Eliminar un número de la lista (Debe reconocer si existe el número, en caso contrario se envía un mensaje)  3.- Ordenar y mostrar la lista  4.- Salir | |

|  |  |
| --- | --- |
| Ítem | Puntaje |
| Ingresar | 15 |
| Eliminar | 15 |
| Ordenar | 10 |

|  |  |
| --- | --- |
| Pregunta # 2 | PUNTAJE: 60 puntos |
| Crear una lista para controlar los clientes de un banco:  La lista contiene la siguiente información de sus clientes   * Rut del cliente * Nombre del cliente * Tipo de cuenta (Corriente, Ahorro) * Saldo   El menú para el control de los clientes es:  1.- Ingresar los datos de los clientes (Rut, Nombre, tipo de cuenta, saldo)  2.- Consultar por tipo de cuenta  3.- Operación (Depositar / Girar)  4.- Salir  **Opción 1**: Ingresar al menos 5 clientes y los muestra una vez ingresados  **Opción 2**: Se ingresa tipo de cuenta y se guardan en una lista nueva los nombres de los clientes que tienen la cuenta indicada. Finalmente se muestra los clientes de la nueva lista y se entrega un total de dichos saldos(Acumulador)  **Opción 3**: Se solicita: Rut del cliente, tipo de operación y monto de la operación. Si la operación es Depósito se aumenta el saldo en dicho monto en caso contrario cuando es girar disminuye el saldo en el monto indicado. En caso que el saldo sea inferior a lo solicitado enviar un mensaje.  Ejemplo   |  |  |  |  | | --- | --- | --- | --- | | RUT | NOMBRE | TIPO CUENTA | SALDO | | 1111 | José | Corriente | 5000 | | 2222 | Alejandra | Ahorro | 1000 | | 3333 | Pedro | Corriente | 3000 |   Si la operación es **Depósito** al cliente **3333** por un monto de **100**, el monto actualizado es 3100  Si la operación es **Giro** al cliente **1111** por un monto de **1000**, el monto actualizado es 4000  Si la operación es **Giro** al cliente **2222** por un monto de **1500**, se indica que el saldo es insuficiente y el saldo no se modifica.  **Opción 4**: Salir | |