|  |  |  |
| --- | --- | --- |
|  |  |  |

Firma

Apellidos

Nombres

**OBJETIVO:** Evaluar los conocimientos adquiridos por los alumnos durante el primer período.

**INSTRUCCIONES:** Leer, analizar y realizar correctamente cada una de las siguientes partes de su examen. Realizar cada ejercicio en Visual Studio Code.

**PARTE I:** realizar los siguientes ejercicios en JavaSwing

1. **JComboBox y JLabel:** crear una ventana con un JComboBox que contenga opciones de países ("México", "Argentina", "España", "Colombia") y un JLabel. Cuando el usuario seleccione un país, el JLabel debe mostrar el texto:

"Ha seleccionado: [país]".

1. **JSlider y JTextField:** diseñar una interfaz con un JSlider que tenga un rango de 0 a 100 y un JTextField. Cada vez que se mueva el slider, el valor debe aparecer automáticamente en el campo de texto.
2. **JTable simple:** crear una ventana con una tabla (JTable) que muestre productos con sus precios.  
   Ejemplo de datos iniciales:

"Laptop", 1200

"Mouse", 25

"Teclado", 45

1. **JList con JTextField y JButton:** crear una interfaz con un JTextField, un JButton y un JList. Cuando el usuario escriba un texto y presione el botón, ese texto debe agregarse a la lista.

**PARTE II:** realizar los siguientes ejercicios con arrays y cadenas.

1. **Eliminar e insertar en una lista de frutas.** Dado el array:

*let frutas = ["Manzana", "Pera", "Plátano", "Sandía", "Melón"];*

* Eliminar el segundo elemento.
* Insertar "Kiwi" en la tercera posición.
* Mostrar el array resultante en consola.

1. **Reemplazar elementos en un inventario.** Dado el array:

let inventario = ["Silla", "Mesa", "Sofá", "Cama", "Escritorio"];

Usar splice para **reemplazar "Sofá" por "Ropero"** y muestra el array final.

1. **Filtrar número pares.** Dado el array: *let numeros = [5, 12, 8, 130, 44, 7, 19];*

Crear un nuevo array solo con los números pares y muéstralo en consola.

**PARTE III:** realizar los siguientes ejercicios en base a manejo de objetos y arreglos.

1. **Buscar estudiante más joven.** A partir de un arreglo de estudiantes (nombre, edad, carrera), muestra al estudiante con la menor edad.
2. **Nota más alta por estudiante.** A partir de un arreglo de objetos estudiante que contiene (nombre, notas[]), muestra el nombre de cada estudiante junto con su nota más alta.
3. **Ordenar estudiantes por edad y nombre.** Ordenar un arreglo de estudiantes:

Primero por edad en orden ascendente.

Si dos estudiantes tienen la misma edad, ordenarlos por nombre en orden alfabético.

**Nota:** guardar cada ejercicio con el nombre: ejercicio1, ejercicio2, ejercicio3, y así sucesivamente hasta ejercicio10. Enviar todo su trabajo en una carpeta comprimida con el nombre: Examen Parcial I.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criterio** | **Excelente** | **Bueno** | **Regular** | **Deficiente** | **Ponderación** |
| **1. Comprensión del enunciado** | Interpreta correctamente lo solicitado y aplica el método adecuado (splice, filter, includes). | Interpreta bien, pero requiere pequeños ajustes. | Comprende parcialmente, con confusiones sobre qué método usar. | No interpreta correctamente lo solicitado. | **20%** |
| **2. Implementación correcta** | El código funciona sin errores y cumple exactamente con el resultado esperado. | El código funciona, pero con pequeños detalles de lógica o sintaxis. | El código presenta fallos importantes en la lógica, aunque intenta usar el método correcto. | El código no funciona o no aplica el método solicitado. | **40%** |
| **3. Claridad y estilo del código** | Código organizado, indentado, con variables descriptivas y consistentes. | Código entendible, con leves fallos de estilo o nombres poco claros. | Código funcional, pero desordenado o con nombres confusos. | Código desorganizado y difícil de leer. | **20%** |
| **4. Presentación de resultados en consola** | Usa console.log() con mensajes descriptivos y claros que muestran bien el resultado. | Muestra resultados, pero con mensajes básicos. | Resultados poco claros o incompletos. | No muestra resultados o son incorrectos. | **10%** |
| **5. Uso de comentarios (//)** | Incluye un título para cada ejercicio y comentarios explicativos claros dentro del código. | Incluye comentarios básicos, aunque comprensibles. | Incluye pocos comentarios o poco útiles para entender el código. | No incluye comentarios en el código. | **10%** |