

Entregable #1 – Java Globant

Todos los ejercicios propuestos deben ser resueltos utilizando **Java** como lenguaje de programación. Para la estructura y organización de cada ejercicio, sigue estas pautas generales:

1. **Archivo individual:** Cada ejercicio debe resolverse en un archivo Java independiente. El archivo debe nombrarse de manera descriptiva según el ejercicio que contiene. Por ejemplo:
 - `Ejercicio1.java`
 - `Ejercicio6.java`
2. **Clase principal y método main:** Por ahora, todos los ejercicios deben implementarse dentro de una única clase con un método `main`. Utiliza solo el método `main` para la lógica del ejercicio. Posteriormente, se podrán modularizar las soluciones en métodos y clases adicionales si es necesario.
3. **Entrada y salida:** La interacción con el usuario debe realizarse a través de la consola, utilizando clases como `Scanner` para la entrada de datos y `System.out.println` para mostrar los resultados en pantalla.
4. **Manejo de errores básico:** No es necesario implementar manejo avanzado de errores (como excepciones), pero asegúrate de que el programa maneje entradas incorrectas de manera simple cuando sea necesario (por ejemplo, en elecciones de menú o reservas de asientos).
5. **Lógica dentro del main:** Toda la lógica del ejercicio (condicionales, bucles, arrays, `ArrayList`, etc.) debe estar dentro del método `main`. No es necesario separar la funcionalidad en métodos adicionales por el momento.

Ejercicio 1: Validación de Usuario Simple

Desarrolla una funcionalidad para verificar si un usuario es mayor de edad para acceder a una página con contenido restringido.

Instrucciones:

1. Solicita al usuario que ingrese su edad.



2. Usa un condicional para verificar si el usuario tiene 18 años o más.
3. Muestra un mensaje que indique si el acceso está permitido o denegado.

Ejercicio 2: Control de Entradas a un Evento

Crea un programa que gestione las entradas para un evento que solo permite un número limitado de asistentes.

Instrucciones:

1. Define un número máximo de asistentes (por ejemplo, 100).
2. Usa un bucle para permitir que los usuarios reserven su entrada.
3. Muestra cuántas entradas quedan disponibles después de cada reserva.
4. Si se agotan las entradas, muestra un mensaje indicando que el evento está lleno.

Ejercicio 3: Sistema de Gestión de Tareas Pendientes

Implementa una lista de tareas pendientes que los usuarios puedan gestionar en una aplicación.

Instrucciones:

1. Crea un arreglo para almacenar hasta 10 tareas.
2. Permite al usuario agregar nuevas tareas a la lista.
3. Implementa una opción para marcar una tarea como completada.
4. Muestra la lista de tareas al final del programa, indicando cuáles están completadas.

Ejercicio 4: Gestión de Inventario en una Tienda

Desarrolla un sistema para gestionar el inventario de una tienda pequeña.



Instrucciones:

1. Usa un ArrayList para almacenar productos disponibles en la tienda.
2. Permite al usuario agregar nuevos productos y eliminar productos que ya no están en stock.
3. Muestra el inventario completo al final, indicando si hay productos agotados.

Ejercicio 5: Sistema de Registro de Usuarios

Desarrolla un sistema de registro para nuevos usuarios de una aplicación.

Instrucciones:

1. Solicita al usuario que ingrese un nombre de usuario, una contraseña y su correo electrónico.
2. Valida que la contraseña tenga al menos 8 caracteres, contenga al menos una letra mayúscula y un número.
3. Almacena los usuarios registrados en un ArrayList y muestra la lista al final.

Ejercicio 6: Sistema de Reservas de Cine

Implementa un sistema para que los usuarios reserven asientos en un cine.

Instrucciones:

1. Crea una matriz de 5x5 para representar las sillas del cine, donde cada asiento puede estar disponible (O) u ocupado (X).
2. Permite al usuario seleccionar una fila y una columna para reservar un asiento.
3. Si el asiento ya está ocupado, informa al usuario y permite seleccionar otro.
4. Muestra la disposición de los asientos después de cada reserva.



Ejercicio 7: Gestión de Reservas en un Restaurante

Desarrolla un sistema para gestionar las reservas en un restaurante.

Instrucciones:

1. Usa un ArrayList para almacenar los nombres de los clientes que han hecho una reserva.
 2. Permite al usuario agregar y cancelar reservas.
 3. Al final, muestra la lista de reservas confirmadas.
-

Ejercicio 8: Sistema de Gestión de Estudiantes

Crea un programa para gestionar la información de los estudiantes en una universidad.

Instrucciones:

1. Usa un ArrayList para almacenar los nombres y promedios de los estudiantes.
 2. Permite al usuario agregar nuevos estudiantes y mostrar la lista de estudiantes con su promedio.
 3. Implementa una opción para buscar estudiantes por nombre y mostrar su promedio.
-

Ejercicio 9: Sistema de Evaluación de Empleados (Condicionales y Arrays)

Desarrolla un sistema de evaluación de desempeño para empleados.

Instrucciones:

1. Crea un arreglo para almacenar los nombres de 5 empleados.
 2. Permite al usuario asignar una calificación de desempeño (entre 1 y 10) a cada empleado.
-



3. Muestra el nombre del empleado con la calificación más alta y más baja al final.
-

Ejercicio 10: Aplicación de Encuestas de Satisfacción

Crea una aplicación para encuestar a los clientes sobre su nivel de satisfacción con un servicio.

Instrucciones:

1. Solicita a 10 clientes que califiquen su satisfacción del 1 al 5.
 2. Usa un arreglo para almacenar las respuestas.
 3. Calcula el promedio de satisfacción al final y muestra cuántos clientes dieron la calificación máxima (5).
-

Ejercicio 11: Sistema de Inventario de Productos con Cantidades

Desarrolla un sistema para gestionar el inventario de productos en una tienda, donde cada producto tiene una cantidad asociada.

Instrucciones:

1. Crea dos arreglos: uno para los nombres de los productos y otro para la cantidad en stock.
 2. Permite al usuario agregar nuevos productos y actualizar las cantidades.
 3. Muestra la lista de productos junto con las cantidades actualizadas.
-

Ejercicio 12: Sistema de Pago para Facturas

Desarrolla un sistema de pago para facturas de servicios (electricidad, agua, internet).



Instrucciones:

1. Crea un sistema que permita al usuario ingresar el monto de tres facturas.
 2. Calcula el total a pagar.
 3. Permite al usuario pagar cada factura individualmente, descontando el monto total.
-

Ejercicio 13: Sistema de Votación Electrónica

Implementa un sistema de votación electrónica para elegir un representante.

Instrucciones:

1. Crea un arreglo con los nombres de 3 candidatos.
 2. Permite a 10 usuarios votar por uno de los candidatos.
 3. Muestra los resultados de la votación al final, indicando quién ganó.
-

Ejercicio 14: Gestión de Empleados con Búsqueda

Crea un sistema para gestionar empleados en una empresa.

Instrucciones:

1. Usa un ArrayList para almacenar el nombre, puesto y salario de cada empleado.
 2. Permite al usuario buscar empleados por nombre o puesto.
 3. Muestra los detalles del empleado encontrado.
-

Ejercicio 15: Sistema de Reservas para un Hotel

Desarrolla un sistema para reservar habitaciones en un hotel.



Instrucciones:

1. Crea una matriz de 4x4 que represente las habitaciones del hotel (disponibles O y ocupadas X).
 2. Permite al usuario seleccionar una habitación para reservarla.
 3. Si la habitación está ocupada, informa al usuario y permite elegir otra.
-

Ejercicio 16: Control de Gastos Mensuales

Crea una aplicación para que el usuario registre sus gastos mensuales.

Instrucciones:

1. Usa un arreglo para almacenar los gastos de cada mes.
 2. Permite al usuario ingresar un gasto mensual durante 12 meses.
 3. Calcula el gasto total y el promedio al final del año.
-

Ejercicio 17: Sistema de Evaluación de Proyectos

Crea un programa que permita evaluar proyectos en una empresa.

Instrucciones:

1. Crea un arreglo para almacenar los nombres de 5 proyectos.
 2. Permite al usuario asignar una calificación a cada proyecto.
 3. Muestra los proyectos ordenados por su calificación, del más alto al más bajo.
-

Ejercicio 18: Sistema de Gestión de Vehículos

Desarrolla un sistema para gestionar la flota de vehículos de una empresa de transporte.



Instrucciones:

1. Usa un ArrayList para almacenar la matrícula, modelo y estado de cada vehículo (disponible u ocupado).
 2. Permite al usuario registrar nuevos vehículos y actualizar el estado de los existentes.
 3. Muestra la lista de vehículos disponibles.
-

Ejercicio 19: Sistema de Control de Inventario con Precios

Implementa un sistema para gestionar el inventario de productos y sus precios en una tienda.

Instrucciones:

1. Usa dos arreglos: uno para los nombres de los productos y otro para sus precios.
 2. Permite al usuario agregar productos y precios.
 3. Muestra el inventario junto con los precios al final.
-

Ejercicio 20: Sistema de Gestión de Reservas de Vuelos (Matriz y Condicionales)

Desarrolla un sistema para gestionar la reserva de asientos en un avión.

Instrucciones:

1. Crea una matriz de 6x6 para representar los asientos del avión.
 2. Permite al usuario seleccionar un asiento para reservar.
 3. Si el asiento ya está ocupado, muestra un mensaje y permite elegir otro.
-