|  |  |  |
| --- | --- | --- |
| | GUÍA 2.1.3:Ejercicios generales | | |
| Sigla | Asignatura | Experiencia de Aprendizaje |
| DSY | Desarrollo Orientado a Objetos | EA2: Herencia y Colecciones |
| Tiempo | Modalidad de Trabajo | Indicadores de logro |
| 1 h | Individual | IL2.1 , IL2.2 y IL2.3 |

|  |
| --- |
| **Código QR con relleno sólido**  **Antecedentes generales** |

## Esta guía tiene como objetivo identificar y reconocer los principales conceptos de herencia en programación orientada a objetos.

## Esta actividad tiene carácter formativo, es decir: es para visualizar lo que aprendes, en la directa medida que tú docente de asignatura te va retroalimentando constantemente, tanto a nivel individual como colectivo.

## Deberá dejar registro del trabajo realizado por medio de la plataforma de Blackboad

|  |
| --- |
| **Lista con relleno sólido Requerimientos para esta actividad** |

## En esta actividad, los estudiantes deberán identificar las entidades obtenidas del análisis del problema planteado, para definir las clases, sus atributos, métodos y un medio de almacenamiento temporal. Además, se debe implementar el código utilizando una colección para la solución del problema planteado

1. Realizar diagramar de clases
2. Programar en el lenguaje JAVA utilizando el IDE Netbeans el diagrama de clases generado
3. Mostrar los datos de la colección, a través de un menú que permita seleccionar la lista.
4. Agregar métodos necesarios para el funcionamiento de las clases
5. Realizar invocaciones desde main para todas las clases

|  |  |
| --- | --- |
| **Inteligencia artificial con relleno sólidoActividad** |  |

**Caso de Estudio: Mantenimiento Eficiente de Reservas de Eventos con Colecciones (****ArrayList) y Métodos de Mantenedor**

**Un pastel en una mesa

Descripción generada automáticamente con confianza mediaContexto:** Una empresa de eventos desea optimizar la gestión de sus reservas mediante el uso de ArrayList y métodos de mantenedor. Se busca una colaboración eficiente entre las clases de Evento, Cliente, Reserva y Lugar, aprovechando las capacidades de las colecciones y métodos específicos para simplificar las operaciones de mantenimiento.

**Requerimientos Iniciales:** El sistema debe permitir a los clientes buscar eventos, realizar reservas y gestionar transacciones. La utilización de ArrayList junto con métodos de mantenedor será fundamental para realizar operaciones de agregar, modificar y eliminar información de manera eficiente.

Modelo propuesto:

1. Evento:
   * Atributos:
     + idEvento (String): Identificador único del evento.
     + nombre (String): Nombre del evento.
     + fecha (Date): Fecha del evento.
     + reservas (ArrayList): Lista que contiene la información de las reservas asociadas al evento.
   * Métodos de Mantenedor:
     + agregarReserva(reserva: Reserva): void: Agrega una reserva a la lista de reservas asociadas al evento.
     + eliminarReserva(reserva: Reserva): void: Elimina una reserva de la lista de reservas asociadas al evento.
2. Cliente:
   * Atributos:
     + idCliente (String): Identificador único del cliente.
     + nombre (String): Nombre del cliente.
     + reservasRealizadas (ArrayList): Lista que contiene la información de las reservas realizadas por el cliente.
   * Métodos de Mantenedor:
     + agregarReserva(reserva: Reserva): void: Agrega una reserva a la lista de reservas realizadas por el cliente.
     + eliminarReserva(reserva: Reserva): void: Elimina una reserva de la lista de reservas realizadas por el cliente.
3. Reserva:
   * Atributos:
     + evento (Evento): Evento asociado a la reserva.
     + asientoReservado (String): Número de asiento reservado por el cliente.
4. Lugar:
   * Atributos:
     + idLugar (String): Identificador único del lugar.
     + nombre (String): Nombre del lugar.
     + eventosDisponibles (ArrayList): Lista que contiene información de eventos en el lugar.
   * Métodos de Mantenedor:
     + agregarEvento(evento: Evento): void: Agrega un evento a la lista de eventos disponibles en el lugar.
     + eliminarEvento(evento: Evento): void: Elimina un evento de la lista de eventos disponibles en el lugar.

Este enfoque de colaboración con ArrayList y métodos de mantenedor mejora la eficiencia y modularidad del sistema de reservas de eventos, facilitando operaciones clave de mantenimiento directamente en las clases correspondientes.