

# Sistema de Reservación de Asientos para un-Estadio de Baseball

**Fecha de Entrega:** miércoles, 27 de noviembre de 2024, 11:59 PM

**Grupos:** Hasta 3 personas, solamente un miembro del grupo entrega el trabajo

## Descripción:

Implementar un sistema de reservación de asientos para un estadio de baseball. El sistema maneja la disponibilidad de asientos, las reservaciones de los clientes y una lista de espera para las secciones que están llenas. El sistema debe permitir reservar asientos, cancelar reservaciones y ver la disponibilidad de asientos.

## El estadio

El estadio ofrece una variedad de opciones de asientos para los juegos de la temporada regular, cada uno con diferentes vistas, comodidades y precios. A continuación se muestran las principales secciones de asientos junto con el costo:

### 1. Field Level

- Asientos premium detrás del plato y a lo largo de las líneas.

Costo: \$300

Capacidad: 500

### 2. Main Level

- Asientos directamente sobre el nivel del terreno y proporcionan buenas líneas de visión detrás de los “dugouts” y a lo largo de las líneas.

Costo: \$120

Capacidad: 1,000

### 3. Grandstand Level

- Asientos ubicados en la cubierta superior detrás del plato y cerca de las líneas. El nivel más alto pero central para la acción.

Costo: \$45

Capacidad: 2,000

## Implementación

1. Debe escribir su programa en Java.
2. Utilice preferiblemente VS Code.
3. El sistema lo debe implementar desde el punto de vista de un operador, esta es la persona que hará las reservaciones para el cliente:
  - Obtendrá información del cliente:
    - Nombre
    - Correo electrónico (email)
    - Número de teléfono
  - Mostrará al cliente las secciones que están aún disponibles y el costo
    - Si hay secciones llenas, dar opción de lista de espera
    - Una vez el cliente decida:
      - Verificar que la cantidad de asientos está disponible
      - Informar al cliente el costo total – no tiene que implementar sistema de pago
      - Reservar los asientos
    - Si se cancela una reservación debe permitir que esta sea tomada por la persona en lista de espera

## **Diseño**

### **Componentes:**

Estos son las clases básicas que debe tener su programa. Puede crear clases adicionales si así lo estima.

1. Clase Asiento

Una clase "asiento" para representar cada asiento en el lugar con atributos como sección, fila y número de asiento.

2. Clase Cliente

Una clase "cliente" para representar cada cliente con atributos como nombre, email, teléfono.

3. Clase Estadio

La clase "Estadio" administra los asientos disponibles, las reservaciones de los clientes y la disponibilidad de asientos.

### **Requisitos:**

Su programa debe hacer uso de:

1. Set: Para almacenar los asientos disponibles en el estadio.
2. LinkedList: Para almacenar el historial de reservaciones o el registro de transacciones.
3. HashMap: Para parear clientes a los asientos que han reservado.
4. Stack: Para implementar la funcionalidad de deshacer, en la que se puede deshacer la última acción de reservación/cancelación.
5. Queue: Para implementar una lista de espera para los clientes cuando todos los asientos de una sección en particular están reservados.

Puede añadir estructuras adicionales en su trabajo si así lo estima necesario

## Documentación y comentarios

Debe documentar su programa.

- Clases – Incluya:
  - en la parte superior de la clase algunos comentarios que expliquen la estructura de su clase, en particular **qué** tipo de lista(s) está utilizando y **por qué**. Por ejemplo, *"Se usó una ArrayList para almacenar..."*.
- Métodos – Incluya:
  - Breve explicación de qué hace
  - Valores que recibe (los parámetros)
  - Valores que devuelve
  - A quién llama
  - Llamada por quien

## Entrega

La fecha límite para enviar su programa será **miércoles, 27 de noviembre a las 11:59 PM**.

Subirá su código a la plataforma github <https://classroom.github.com/a/vejrih7Q> y debe seguir el formato explicado en este documento. Se evaluará su proyecto con el último envío antes de la fecha límite. Cualquier código que se envíe al repositorio después de la fecha límite, así como cualquier código que no esté en el repositorio de GitHub Classroom (por ejemplo, por correo electrónico), no se tendrá en cuenta para la calificación.

## Integridad Académica

**No comparta su código!** Usted puede discutir estrategias de diseño e implementación pero, recuerde:

- si se encuentran proyectos que son muy similares para que sea coincidencia todas las partes involucradas obtendrán una calificación de 0.
- Igualmente si se identifica que el código fue implementado usando alguna herramienta de Inteligencia Artificial, como ChatGPT, también obtendrá una calificación de 0.