

EVALUACIÓN	Obligatorio AED1	GRUPO	Todos	FECHA	Marzo de 2025
MATERIA					
CARRERA					
CONDICIONES	<p>Lectura de Obligatorio: 07/04/2025</p> <p>Fecha máxima de entrega 1: 12/05/2025 hasta las 21 hs.</p> <p>Puntaje: Máximo 20, Mínimo 0</p> <p>Fecha máxima de entrega 2: 18/06/2025 hasta las 21 hs.</p> <p>Puntaje: Máximo 35, Mínimo 0</p> <p>LA ENTREGA SE REALIZA EN FORMA ONLINE EN ARCHIVO NO MAYOR A 40MB EN FORMATO ZIP O PDF.</p> <p>Uso de material de apoyo y/o consulta</p> <p><u>Inteligencia Artificial Generativa</u></p> <ul style="list-style-type: none"> - Seguir las pautas de los docentes: Se deben seguir las instrucciones específicas de los docentes sobre cómo utilizar la IA en cada curso. - Citar correctamente las fuentes y usos de IA: Siempre que se utilice una herramienta de IA para generar contenido, se debe citar adecuadamente la fuente y la forma en que se utilizó. - Verificar el contenido generado por la IA: No todo el contenido generado por la IA es correcto o preciso. Es esencial que los estudiantes verifiquen la información antes de usarla. - Ser responsables con el uso de la IA: Conocer los riesgos y desafíos, como la creación de “alucinaciones”, los peligros para la privacidad, las cuestiones de propiedad intelectual, los sesgos inherentes y la producción de contenido falso - En caso de existir dudas sobre la autoría, plagio o uso no atribuido de IAG, el docente tendrá la opción de convocar al equipo de obligatorio a una defensa específica e individual sobre el tema <p>IMPORTANTE:</p> <ol style="list-style-type: none"> 1) Inscribirse 2) Formar grupos de hasta 2 personas del mismo dictado 3) Subir el trabajo a Gestión antes de la hora indicada (ver hoja al final del documento: “RECORDATORIO”) <p>Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor contactarse con el Coordinador o Coordinación adjunta antes de las 20:00hs. del día de la entrega, a través de los mails alamon@ort.edu.uy y fernandez_ma@ort.edu.uy, o telefónicamente al 29021505 - int 1156 u 1138</p>				

Introducción

Se desea implementar un sistema para la gestión de eventos presenciales. Cada evento se realiza en una fecha específica y en una sala determinada, la cual debe cumplir con los requerimientos de aforo del evento. Se deberá realizar la gestión de la compra de entradas de clientes a los eventos, las devoluciones de entradas, así como también las listas de esperas (en caso de no haber entradas disponibles). El sistema deberá brindar además un conjunto de reportes, fundamentales para la toma de decisiones.

El sistema permite entre otros:

- Registrar salas y eventos
- Asignar eventos a salas disponibles según capacidad y horario
- Registrar clientes
- Comprar y devolver entradas
- Generar reportes

Todas las estructuras deben resolverse utilizando TADs implementados por el estudiante. No se permite el uso de estructuras de Java.

Se proveen los siguientes tipos de datos que deberán ser respetados.

Sistema	<pre>public class Sistema{ /*Aquí introduzca la información que estime conveniente*/ }</pre>
Retorno	<pre>public class Retorno{ enum Resultado{OK,ERROR_1,ERROR_2,ERROR_3,ERROR_4, ERROR_5, NO_IMPLEMENTADA}; boolean valorBooleano int valorEntero; String valorString; Resultado resultado; }</pre>

Pueden definirse tipos de datos (clases) auxiliares.

1. Registro de Salas, Eventos y Clientes

1.1. Registro de Salas, Eventos y Clientes

Firma: Retorno `crearSistemaDeGestion()`;

Descripción: Crea la estructura necesaria para representar el sistema de Gestión.

Retornos posibles	
OK	Si pudo inicializar el sistema correctamente.
ERROR	
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

1.2. Registrar Sala

Firma: Retorno `registrarSala(String nombre, int capacidad)`;

Descripción: Registra una nueva sala en el sistema.

Retornos posibles	
OK	Si pudo registrar la sala
ERROR	1. Ya existe una sala con ese nombre 2. Si la capacidad es ≤ 0
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

1.3. Eliminar Sala

Firma: Retorno `eliminarSala(String nombre)`;

Descripción: Elimina la sala del sistema.

Retornos posibles	
OK	Si pudo eliminar la sala
ERROR	1. No existe una sala con ese nombre
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

1.4. Registrar evento

Firma: Retorno `registrarEvento(String codigo, String descripcion, int aforoNecesario, LocalDate fecha)`;

Descripción: Registra un nuevo evento. El sistema debe asignar automáticamente una sala disponible que tenga una capacidad mayor o igual al aforo necesario y no esté ocupada en el día (1 a 30) y mes (1 a 12). Una sala se asigna a un evento durante todo el día.

Retornos posibles	
OK	Si pudo registrar el evento
ERROR	1. Ya existe un evento con ese código 2. Si aforo necesario ≤ 0 3. No hay salas disponibles para esa fecha con aforo suficiente
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

1.5. Registrar cliente

Firma: Retorno Retorno registrarCliente(String cedula, String nombre);

Descripción: Registra un nuevo cliente. La cédula debe tener exactamente 8 dígitos.

Retornos posibles	
OK	Si pudo registrar el cliente
ERROR	1. Formato inválido de cédula 2. Cliente ya registrado (cédula ya registrada)
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

1.6. Comprar entrada

Firma: Retorno comprarEntrada(String cedula, String codigoEvento);

Descripción: Si existe disponibilidad de entradas, se asigna una al cliente. De lo contrario, se lo agrega a la lista de espera para el evento.

Retornos posibles	
OK	Si pudo comprar la entrada
ERROR	1. Cliente no existe 2. Evento no existe
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

1.7. Eliminar evento

Firma: Retorno eliminarEvento(String codigo);

Descripción: Elimina un evento, siempre y cuando no haya entradas vendidas. En dicho caso, se debe liberar la sala que tenga asignada.

Retornos posibles	
OK	Si pudo eliminar el evento
ERROR	1. No existe el evento 2. El evento tiene entradas vendidas
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

1.8. Devolver entrada

Firma: Retorno devolverEntrada(String cedula, String codigoEvento);

Descripción: Se devuelve una entrada. Si hay clientes esperando por ella, se reasigna automáticamente al primero en la lista, respetando el orden de llegada.

Retornos posibles	
OK	Si pudo devolver la entrada
ERROR	1. Cliente no existe 2. Evento no existe
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

1.9. Calificar evento

Firma: Retorno `calificarEvento(String cedula, String codigoEvento, int puntaje, String comentario);`

Descripción: Califica el evento al que asistió

Retornos posibles	
OK	Si pudo calificar el evento
ERROR	1. Cliente no existe 2. Evento no existe 3. Puntaje < 1 o puntaje > 10 4. El evento ya fue calificado por el cliente
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

2. Reportes

2.1 Listar salas

Firma: Retorno `listarSalas();`

Descripción: Lista todas las salas, en orden inverso al registro. Se deberá mostrar su nombre y capacidad.

Retornos posibles	
OK	Si pudo listar las salas
ERROR	
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

Formato:

Sala Verde-45#Sala Azul-20#Sala Violeta-67

2.2 Listar eventos

Firma: Retorno `listarEventos();`

Descripción: Muestra todos los eventos registrados ordenados alfabéticamente por el código del evento, incluyendo código, descripción, sala asignada y cantidad de entradas disponibles y cantidad de entradas vendidas.

Retornos posibles	
OK	Si pudo listar los eventos
ERROR	
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

Formato:

CUC22-Tango Azul-11-20-20#TEC43-Seminario de Tecnología-45-5-67#KAK34-Noche de Rock-45-10-30

2.3 Listar clientes

Firma: Retorno listarClientes();

Descripción: Lista todos los clientes registrados ordenados por cédula de identidad.

Retornos posibles	
OK	Si pudo listar los clientes
ERROR	
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

Formato:

2333111-Martina Rodríguez#35679992-Ramiro Perez#45678992-Micaela Ferrez

2.4 Sala óptima

Firma: Retorno esSalaOptima(String vistaSala[][]);

Descripción: Dado la vista de ocupación de una sala, en donde se identifican con: “#” -límites de la sala (no puede haber asientos), “X”- asiento libre, “O”-asiento ocupado; se debe indicar si su ocupación es óptima. Se considera que existe ocupación óptima si, existen al menos dos columnas de la sala, en donde la cantidad de asientos ocupados consecutivos de dichas columnas superan a los asientos libres (sin importar donde estén) de la misma columna. Se debe retornar: “Es óptimo” o “No es óptimo”

Retornos posibles	
OK	Si pudo indicar si es sala óptima
ERROR	
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

#	#	#	#	#	#	#
#	#	X	X	X	X	#
#	O	O	X	X	X	#
#	O	O	O	O	X	#
#	O	O	X	O	O	#
#	O	O	O	O	O	#
#	X	X	O	O	O	O
#	X	X	O	O	O	X
#	X	X	O	X	X	#
#	X	X	O	X	X	#
#	#	#	O	#	#	#
#	#	#	O	#	#	#

Retorno: “Es óptimo”

Explicación: la columna 4 cuenta con 7 asientos ocupados consecutivos vs. 3 asientos libres. La columna 5 cuenta con 5 asientos ocupados consecutivos vs. 4 asientos libres. Estas son las únicas columnas que cumplen con la condición.

2.5 Listar clientes de evento

Firma: Retorno `listarClientesDeEvento(String código, int n);`

Descripción: Lista los hasta n últimos clientes que compraron entradas para el evento (si hay menos de n clientes se listan todos)

Retornos posibles	
OK	Si pudo listar los clientes
ERROR	1. Evento no existe 2. $n < 1$
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

Formato:

35679992-Ramiro Perez#45678992-Micaela Ferrez

2.6 Lista de espera por evento

Firma: Retorno `listarEsperaEvento();`

Descripción: Muestra los clientes en lista de espera para cada evento. Los eventos deben estar ordenados en forma alfabética por código de evento y los clientes, ordenados por cédula dentro del mismo evento. Solo se deben listar aquellos eventos con clientes en espera.

Retornos posibles	
OK	Si pudo listar los clientes en espera
ERROR	
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

Formato:

KAK34-2333111# KAK34-45678992# TEC43-35679992

2.7 Deshacer compras de entradas

Firma: Retorno `deshacerUltimasCompras(int n);`

Descripción: Se deben deshacer las últimas n compras de entradas realizadas (tomando en cuenta todos eventos), devolviendo las mismas a sus respectivos eventos. Se deben mostrar las entradas, detallando: código del evento, cédula de identidad del cliente (ordenado por código de evento/cédula de cliente)

Retornos posibles	
OK	Si pudo deshacer las entradas.
ERROR	
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

Formato:

KAK34-2333111# KAK34-45678992

2.8 Evento mejor puntuado

Firma: Retorno eventoMejorPuntuado();

Descripción: Indica cual fue el evento que obtuvo el mejor puntaje promedio. En caso de que existan más de uno, mostrarlos todos, ordenados por código de evento, indicando el puntaje promedio que obtuvo.

Retornos posibles	
OK	Si muestra el mejor evento puntuado
ERROR	
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

Formato:

Ejemplo para puntaje promedio 9
KAK34-9#TEC43-9

2.9 Compras de cliente

Firma: Retorno comprasDeCliente(String cedula);

Descripción: Se deben mostrar las compras realizadas del cliente, indicando para cada una si fue devuelta (D) o si no fue devuelta (N). Las entradas deben mostrarse en el orden en el que fueron compradas (primero debe mostrarse la primera que fue comprada), incluyendo el código del evento.

Retornos posibles	
OK	Si pudo mostrar las compras del cliente
ERROR	1. Cliente no existe
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

Formato:

TEC43-N#CUC11-N# COP10-D

2.10 Cantidad de compras por día

Firma: Retorno comprasXDia(int mes);

Descripción: Se debe mostrar la cantidad de compras que se realizaron para cada uno de los días del mes, tomando en cuenta todos los eventos y clientes. Se debe mostrar: el número de día (ordenado en forma ascendente) y la cantidad. En caso de no haber compras en un determinado día, no debería aparecer.

Retornos posibles	
OK	Si pudo mostrar las compras del cliente
ERROR	1. Mes < 1 o Mes > 12
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

Formato:

1-10#2-6#4-67#7-5#28-6

PRIMERA ENTREGA

Para la primera entrega se solicita:

- 1) Representación gráfica de las estructuras seleccionadas **para resolver TODO el problema planteado** (ver ejemplo disponible en Aulas). Justificar las decisiones de diseño adoptadas (las mismas deben estar fundamentadas a partir de los requerimientos del problema).
- 2) Implementar las **operaciones 1.1, 1.2, 1.3, 1.4, 1.5, 2.1, 2.2, 2.3, 2.4**.
- 3) Realizar y entregar juego de pruebas que evidencien el correcto funcionamiento de dichas operaciones. Se deberá adjuntar una hoja (.pdf) con el resultado, detallando – en caso de existir – las funcionalidades que no cumplen con los requisitos planteados.

Se deberá entregar un .ZIP con el proyecto y un archivo .PDF con la representación de la arquitectura escogida y el resultado de las pruebas. Se valorará la eficiencia y adecuación de esta al contexto planteado.

SEGUNDA ENTREGA

Para la segunda entrega se solicita:

- 1) Representación gráfica - ajustada - final.
- 2) Implementar de TODAS las operaciones de la interfaz “Obligatorio”
- 3) Realizar y entregar juego de pruebas definitivo que evidencie el correcto funcionamiento de TODAS las operaciones.


Se deberá entregar un .ZIP con el proyecto y un archivo .PDF con la representación de la arquitectura final escogida. Se valorará la eficiencia y adecuación de esta al contexto planteado.

RECORDATORIO: IMPORTANTE PARA LA ENTREGA

- **Obligatorios**

La entrega de los obligatorios será en formato digital online, a excepción de algunas materias que se entregarán en Bedelía y en ese caso recibirá información específica en el dictado de la misma.

Los principales aspectos a destacar sobre la **entrega online de obligatorios** son:

1. Ingresá al sistema de Gestión.
2. En el menú, seleccioná el ítem "Evaluaciones" y la instancia de evaluación correspondiente, que figura bajo el título "Inscripto".
3. Para iniciar la entrega hacé clic en el ícono: 
4. Ingresá el número de estudiante de cada uno de los integrantes y hacé clic en "Agregar". El sistema confirmará que los integrantes estén inscriptos al obligatorio y, de ser así, mostrará el nombre y la fotografía de cada uno de ellos. Una vez agregados todos los integrantes, hacé clic en "Crear equipo".

Cualquier integrante podrá:

- **Modificar la integración del equipo.**
- **Subir el archivo de la entrega.**

5. Seleccioná el archivo que deseás entregar. Verificá el nombre del archivo que aparecerá en la pantalla y hacé clic en "Subir" para iniciar la entrega. Cada equipo (hasta 2 estudiantes) debe entregar **un único archivo en formato zip o rar** (los documentos de texto deben ser pdf, y deben ir dentro del zip o rar). El archivo a subir debe tener **un tamaño máximo de 40mb**

Cuando el archivo quede subido, se mostrará el nombre generado por el sistema (1), el tamaño y la fecha en que fue subido.

6. El sistema enviará un e-mail a todos los integrantes del equipo informando los detalles del archivo entregado y confirmando que la entrega fue realizada correctamente.
7. Podés cerrar la pestaña de entrega y continuar utilizando Gestión o salir del sistema.
8. La **hora tope para subir el archivo será las 21:00** del día fijado para la entrega.
9. La entrega se podrá realizar desde cualquier lugar (ej. hogar del estudiante, laboratorios de la Universidad, etc).
10. Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor contactarse con la Coordinadora o Coordinación adjunta antes de las 20:00hs. del día de la entrega, a través de los mails, alamon@ort.edu.uy y fernandez_ma@ort.edu.uy, o telefónicamente al 29021505 - int 1156 u 1138.