#### **(Objeto Date): Funciones de manejo de fechas y tiempos**

1. **Programar Recordatorio de Evento**:
   * **Función**: programarRecordatorio(fechaConcierto, diasAntes)
   * **Descripción**: Establece recordatorios para fechas clave del concierto, como promoción, preventa, o recordatorio de último minuto.
   * **Entrada**: fechaConcierto (fecha del concierto en formato YYYY-MM-DD) y diasAntes (días de antelación para el recordatorio).
   * **Salida**: Fecha formateada para el recordatorio.
2. **Determinar Temporada de Conciertos**:
   * **Función**: determinarTemporada(fechaConcierto)
   * **Descripción**: Asigna cada concierto a una temporada (primavera, verano, otoño, invierno) para facilitar la programación y marketing.
   * **Entrada**: fechaConcierto
   * **Salida**: Temporada en la que cae el concierto (texto).
3. **Calcular Días de Antelación para Recordatorio de Venta de Tickets**:
   * **Función**: calcularDiasAntelacion(fechaConcierto, fechaVenta)
   * **Descripción**: Calcula cuántos días antes del concierto debe iniciar la venta de tickets.
   * **Entrada**: fechaConcierto y fechaVenta
   * **Salida**: Número de días de antelación.

#### **(Objeto Math): Funciones de cálculo y manipulación de ingresos**

1. **Calcular Precio de Tickets con Descuento**:
   * **Función**: calcularPrecioDescuento(precioBase, descuento)
   * **Descripción**: Calcula el precio del ticket después de aplicar un descuento para promociones especiales (porcentaje entre 5% y 30%).
   * **Entrada**: precioBase y descuento (número entre 5 y 30)
   * **Salida**: Precio final con descuento.
2. **Calcular Ingresos Esperados**:
   * **Función**: calcularIngresosEsperados(precioTicket, ticketsVendidos)
   * **Descripción**: Calcula el ingreso esperado multiplicando el precio del ticket por el número de tickets vendidos.
   * **Entrada**: precioTicket y ticketsVendidos
   * **Salida**: Ingreso total estimado.

#### **Objeto String): Funciones de manipulación de texto para nombres de artistas y conciertos**

1. **Formatear Nombre del Concierto**:
   * **Función**: formatearNombreConcierto(nombre)
   * **Descripción**: Formatea el nombre del concierto en estilo título.
   * **Entrada**: nombre
   * **Salida**: Nombre formateado en estilo título.
2. **Crear Descripción del Evento**:
   * **Función**: crearDescripcionEvento(nombreConcierto, nombreArtista, fecha)
   * **Descripción**: Genera una breve descripción para el concierto, combinando el nombre, el artista y la fecha.
   * **Entrada**: nombreConcierto, nombreArtista, fecha
   * **Salida**: Descripción completa (ej. "Concierto de Rock con Banda XYZ el 15 de Mayo").
3. **Validar Entrada de Texto para Nombre del Artista**:
   * **Función**: validarNombreArtista(nombre)
   * **Descripción**: Valida que el nombre del artista esté bien formateado, eliminando espacios extras.
   * **Entrada**: nombre
   * **Salida**: Nombre sin espacios adicionales.

#### **(Objeto Number): Funciones de manejo de entradas y costos por asistente**

1. **Validar Número de Tickets Disponibles**:
   * **Función**: validarTicketsDisponibles(ticketsIngresados, maxTickets)
   * **Descripción**: Verifica si el número de tickets ingresado no supera la capacidad máxima de la sala.
   * **Entrada**: ticketsIngresados y maxTickets
   * **Salida**: Booleano (true si es válido, false si no).

En conjunto se deberá desarrollar:

**Interfaz para el Usuario**: Crear una interfaz en HTML donde el usuario pueda:

* Ingresar detalles del concierto.
* Ver la descripción del evento, la fecha del recordatorio, la división de ingresos y el precio de los tickets.

**Integración de las funcionalidades dentro de un flujo principal.**

**Pruebas y Presentación Final**: Cada grupo probará su aplicación y presentará su funcionamiento, destacando cómo integraron todas las funcionalidades para gestionar de forma integral la sala de conciertos.