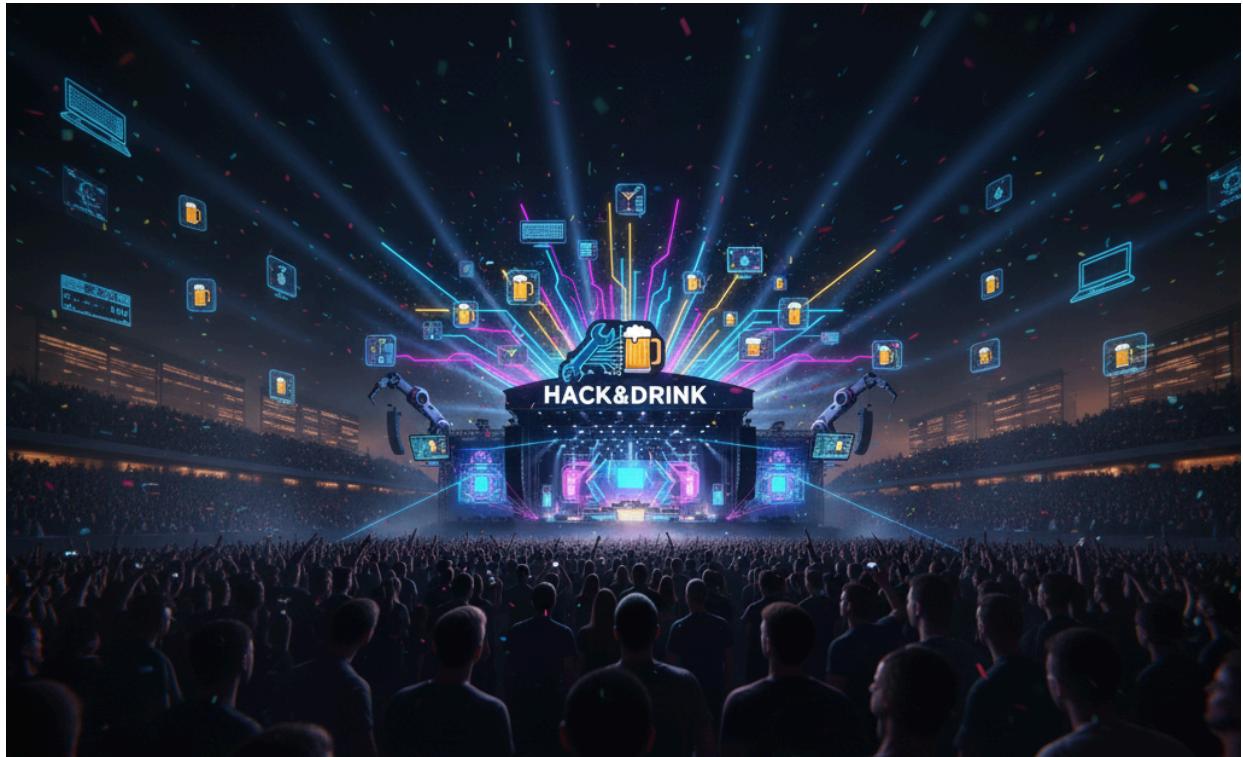


# **HACK&DRINK**

## *Recopilación y análisis de requisitos*



### *Datos del Grupo de Trabajo*

<b>Grupo Pequeño (prácticas y seminarios)</b>	B1
<b>Grupo de trabajo</b>	Hack&Drink
<b>Integrantes</b>	Mario Rasero Vaquero: Patrocinadores Alejandro Ortega Martín: Artistas Manuel Martín Rodríguez: Clientes Roberto Vera Ramos: Servicios Adrian Ros Moya: Infraestructura

# ***Índice***

1. Descripción del sistema.....	3
2. Gestión de clientes.....	6
3. Gestión de artistas.....	15
4. Gestión de patrocinadores.....	22
5. Gestión de servicios.....	29
6. Gestión de Infraestructura.....	36

# 1. Descripción del sistema

Una empresa de organización de festivales nos ha contratado para desarrollar un Sistema de Gestión de Festivales, mediante este sistema la empresa podrá optimizar la administración de los puntos clave para la organización de este tipo de eventos y consecuentemente la experiencia de todos los usuarios, artistas, patrocinadores y proveedores.

Actualmente nos enfrentamos a diversos problemas, como la falta de un registro centralizado de clientes y artistas, complicaciones en el control de entradas y pagos, así como una comunicación ineficiente entre las distintas partes. De esta forma con este sistema conseguirán mejorar la eficiencia, ofrecer al público un buen servicio y reducir errores.

El sistema software que se requiere debe permitir entre otras cosas:

- En el subsistema de clientes, los usuarios pueden registrarse en la plataforma creando una cuenta con datos personales como nombre, apellidos, correo electrónico, DNI/NIF, teléfono, dirección **y contraseña** o modificar dichos datos, pero no se permitirá la alta si ya existe otro cliente con el mismo DNI/NIF o correo electrónico, esta comprobación la realiza el SGBD. Una vez iniciada la sesión del cliente en el sistema, no es necesario que vuelva a iniciar cada vez que realice una operación pues el Sistema identifica la operación que haga con el cliente en cuestión. **Tras un mes de inactividad, el Sistema te pedirá que vuelvas a iniciar sesión con DNI y contraseña.** Posteriormente, el sistema permite solicitar la baja, aunque esta no podrá realizarse si el cliente tiene entradas compradas para eventos pendientes. Los clientes registrados pueden comprar entradas online seleccionando evento (nombre, localización, fecha), tipo y cantidad de entradas y método de pago (tarjeta (Número de tarjeta, Fecha de caducidad y Código de seguridad (CVC)), PayPal o transferencia), validando siempre la disponibilidad de las entradas, y la compra solo se confirma tras la validación del pago; en caso contrario, la reserva se cancela, todo esto comprobado por el SGBD. La entrada será enviada al correo electrónico del cliente y se le dará al cliente un localizador de la entrada para reclamar en caso de incidencia o si se le pierde. Tras asistir a un evento, únicamente los clientes con entrada válida en el evento podrán llenar encuestas de satisfacción sobre el evento identificado por su nombre, localización y fecha, cuyas respuestas quedan almacenadas para generar estadísticas. Además, los clientes pueden obtener abonos, descuentos, seguros y promociones, un abono sólo podrá adquirirse si se cumplen los requisitos establecidos (número mínimo de entradas). Un descuento o promoción solo será aplicable si se cumplen los requisitos establecidos (compra anticipada, cliente frecuente, fecha válida, documento válido de familia numerosa), todo esto será comprobado por el área de abonos, descuentos, promociones y seguros que le mandará una comprobación al gestión de abonos, descuentos, promociones y seguros. Por último, ofrece un servicio de atención al cliente para consultas, reclamaciones y devoluciones que el cliente debe indicar qué tipo de solicitud y la descripción de la misma. Una vez hecho eso, las solicitudes serán asignadas al área de la organización que se corresponda (departamento de incidencias, departamento de devoluciones..etc) y estas enviarán la respuesta a atención al cliente y se le notificará al cliente la respuesta final, pero las devoluciones sólo serán aceptadas si cumplen con la política establecida, como realizarse dentro del plazo permitido o si el evento ha sido

cancelado y solo se puede realizar una reclamación si el cliente ha asistido al evento. Para ello, se comprueba la validación de la entrada.

- En cuanto al subsistema de artistas, se realizará un registro de los mismos creando a su vez una cuenta con los datos personales como nombre, apellidos, correo electrónico, DNI/NIF, teléfono y dirección, además de otros datos relevantes como el caché del artista y el género al que pertenece. Posteriormente, se podrán realizar tanto modificaciones de cualquiera de los datos o bajas de artistas. También se llevará a cabo un registro de actuaciones que se identificarán por día, hora y número de escenario, pero no se permitirá el registro de otra actuación con las mismas identificaciones (día, hora y escenario iguales) de esta forma la organización tendrá todos los conciertos ubicados en distintas franjas horarias y escenarios, evitando que se pisen. Cada uno de los artistas tendrá un camerino privado, para gestionarlo se realizará una asignación a través de un identificador numérico, el sistema no permitirá asignar más de un artista al mismo camerino, también se le asociará la franja horaria en la que este camerino sera utilizado por dicho artista para así maximizar el uso de las instalaciones. Este subsistema también recoge la realización de los contratos, estos tendrán un ID único que asociara cada contrato a cada artista, además recogerá los datos personales y administrativos necesarios. Finalmente, se gestionan los pagos a los artistas, permitiendo registrar cada pago con su fecha y importe, tendrá una referencia al id del contrato del que forma parte el pago y además habrá un conteo del número de pago que representa con respecto al fraccionado.
- En el subsistema de patrocinadores, el responsable de patrocinadores realizará un registro del nombre del patrocinador y de la persona de contacto como representante de este patrocinador de la que se registrará DNI, teléfono, correo electrónico y el nombre de esta persona. Se podrán consultar los patrocinadores asociados al evento. Además, el sistema permitirá asociar y gestionar contratos entre los patrocinadores y el festival, registrando id(número único que representa a este contrato),fecha de inicio y fin, los datos bancarios y el importe acordado en cada contrato. El sistema permitirá gestionar los beneficios que se otorgan a cada uno de los patrocinadores del festival, indicando el tipo de beneficio con un máximo de 50 caracteres y una breve descripción de este mismo de hasta 150 caracteres, dentro de los beneficios posibles entrarían las menciones en las redes sociales o logo en la cartelería del festival. Finalmente, el sistema facilitará la gestión de los pagos realizados por los patrocinadores, permitiendo registrar cada pago con su fecha, importe y método, también incluirá un apartado para saber si el pago es fraccionado o completo, en cuantos meses se ha fraccionado(en caso de que lo esté será un número X que representa los meses en los que pagará la totalidad, en caso de que no tendrá por defecto 1) y tendrá una referencia al id del contrato del que forma parte el pago y además habrá un conteo del número de pago que representa con respecto al fraccionado.
- Para el subsistema de gestión de servicios, el responsable de la gestión de servicios debe poder asignar una serie de proveedores/empresas que serán las encargadas de proporcionar los servicios de seguridad, limpieza y catering, introduciendo de estas el NIF, el nombre, el servicio que van a cubrir y en el caso en que la función suponga la adhesión de nuevos empleados externos (como seguridad), el número de empleados, y el DNI de cada uno de estos, para poder identificarlos posteriormente, en caso de que el NIF introducido ya esté en el

sistema puede modificar los datos de ese proveedor. Como la salida es siempre una lista de los proveedores junto con su NIF, nombre y función, se puede utilizar esta operación a modo de consulta si no introduces nada, si introduces un NIF y no está en el sistema, saltará un error. Un punto clave en cualquier empresa, es asignar los turnos de trabajo, la coordinación es crucial para atender a los asistentes en todo momento. El responsable podrá asignar un turno de trabajo, haciendo uso del DNI del empleado y su función, pasando como parámetros el área que cubrirá, una hora de entrada y otra de salida para completar el turno. Cada función (seguridad, catering, limpieza) tendrá una serie de áreas distintas y, además, cada área tendrá establecido un número máximo de integrantes por turno, para evitar solapamientos innecesarios. Por otra parte, los empleados del catering deben hacer comprobaciones del nivel de suministros al final de su turno e indicarlo en el sistema para que otro empleado pueda reabastecer en caso de ser necesario y que todo se haga lo más rápido posible, sin tener que ir a comprobar el mismo el nivel de dicho suministro, para ello, el sistema dispondrá de un apartado, en el que el empleado en cuestión, podrá introducir el id del suministro, el área (por ejemplo, al haber varias barras de bebidas, puede que en un área haya una bebida pero en otra no haya, o podemos hablar de que no haya de esa bebida en el almacén tampoco) y por último el nivel que corresponderá a un entero (puede representar cajas, unidades...). En los festivales ocurren incidencias constantemente, un camarero borde, un lavabo roto... Los asistentes al festival deben poder notificar incidencias ocurridas en el festival, simplemente tendrán que poner como parámetros el DNI y su nombre, un email válido, el identificador de la entrada (opcional, pero sirve para confirmar que era un asistente al festival y tenerlo como prioridad) y un texto notificando la incidencia. Por último, a fin de mejorar el festival temporada tras temporada, el sistema dispone de un sistema de feedback/valoración, al que podrá acceder todo el mundo, incluso empleados, introduciendo un email válido (se validará si es real o no, es obligatorio) y a continuación el nombre (opcional), id de la entrada (opcional), un entero (será la valoración y es obligatorio) y un comentario de texto (opcional también).

- Por otro lado, en cuanto al subsistema de infraestructura se tienen en cuenta temas como el registro de permisos, ya sean de utilización de la zona para el festival, de distribución de alimentos y sobre construcción de escenarios si fuera necesario, dichos permisos son introducidos en la base de datos. También se controla el aforo de la siguiente forma: se tienen en cuenta 2 cifras, una representante del aforo actual durante el festival y otra representando el aforo máximo, cada vez que un cliente entra en el festival se registra actualizando la cifra actual (se aumenta en uno), si esta alcanza el aforo máximo no pueden entrar más clientes. La infraestructura también tiene en cuenta el registro y posicionamiento de puestos de comida y de baños, de forma que se introducen sus identificadores, y los nombres de los puestos de comida, así como los alimentos que estos venden, se distribuyen equitativamente alrededor del festival y se devuelve sus identificadores junto a la posición en la que se encuentran. Por último este subsistema controla averías de la siguiente forma: se registra en la base de datos el identificador de la avería (0,1,2...) y el tipo de avería que es, junto a la posición en la que ha ocurrido y si se ha solucionado o no y cómo.

En definitiva, se necesita desarrollar una aplicación que permita llevar a cabo toda la funcionalidad requerida para realizar la gestión integral de un festival, garantizando una comunicación eficiente

entre todas las partes involucradas, ofreciendo experiencias innovadoras y asegurando un servicio de calidad.

## 2. Gestión de clientes

Requisito funcional			
Número de referencia	RF1.1		
Nombre del requisito	Alta/Modificación de datos del cliente		
Descripción del requisito	Permite a un cliente registrarse en la plataforma proporcionando sus datos personales o modificar los datos de su cuenta		
Entrada	<b>Agente externo</b> <b>Requisito de datos</b>	<b>Cliente</b> <b>Número de referencia</b> <b>Composición</b>	<b>RDE1.1</b> <ul style="list-style-type: none"> <li>• Nombre: Cadena de caracteres (40)</li> <li>• Apellidos: Cadena de caracteres ( 60)</li> <li>• DNI/NIF: Cadena de alfanumericos (9)</li> <li>• Correo electrónico: Cadena de caracteres ( 50)</li> <li>• Teléfono: Cadena de numeros (20)</li> <li>• Dirección: Cadena de caracteres (100)</li> <li>• <b>Contraseña: Cadena de caracteres (12)</b></li> </ul>
Datos internos	<b>Requisito de datos</b>	<b>Número de referencia</b> <b>Descripción</b>	<b>RDW1.1</b> Almacenamiento y actualización de los datos del cliente en la base de datos

			que son RDE1.1.
Salida	Agente externo		
	Requisito de datos	Número de referencia	
		Composición	
Restricciones semánticas		Número de referencia	RS1.1
		Descripción	<ul style="list-style-type: none"> <li>No se podrá registrar un cliente si ya existe otro con el mismo DNI/NIF. Dicha comprobación lo hace el SGBD en RDW1.1</li> </ul>
		Número de referencia	RS1.2
		Descripción	<ul style="list-style-type: none"> <li>No se podrá registrar un cliente si ya existe otro con el mismo Correo electrónico. Dicha comprobación lo hace el SGBD en RDW1.1</li> </ul>

Requisito funcional			
Número de referencia	RF1.2		
Nombre del requisito	Solicitar baja del cliente		
Descripción del requisito	Permite a los clientes solicitar la baja de la plataforma		
Entrada	Agente externo	Cliente	
	Requisito de datos	<p>Número de referencia</p> <p>RDE1.2</p> <p>Composición</p> <ul style="list-style-type: none"> <li>• DNI/NIF: Cadena de alfanumericos (9)</li> <li>• Correo electrónico: Cadena de caracteres ( 50)</li> </ul>	
Datos internos	Requisito de datos	Número de referencia	RDW1.2
		Descripción	Eliminación de los datos del cliente en la base de datos que son los RDE1.1.
Salida	Agente externo		
	Requisito de datos	Número de referencia	
		Composición	
Restricciones		Número de referencia	RS1.3

<b>semánticas</b>		<b>Descripción</b>	<ul style="list-style-type: none"> <li>• La baja no será posible si el cliente tiene entradas de eventos que aún no han empezado. Dicha comprobación la hace el SGBD en RDW1.2.</li> </ul>
-------------------	--	--------------------	--

<b>Requisito funcional</b>			
<b>Número de referencia</b>	RF1.3		
<b>Nombre del requisito</b>	Compra de entrada online		
<b>Descripción del requisito</b>	Permite al cliente registrado adquirir entradas a través de la web seleccionando evento, número de entradas, tipo y método de pago.		
<b>Entrada</b>	<b>Agente externo</b>	Cliente	
	<b>Requisito de datos</b>	<b>Número de referencia</b>	RDE1.3
<b>Datos internos</b>		<b>Composición</b>	<ul style="list-style-type: none"> <li>• Nombre del evento: Cadena de caracteres (300)</li> <li>• Localización del evento: Cadena de caracteres (50)</li> <li>• Fecha del evento: Tipo Date (DD/MM/AA)</li> <li>• Número de entradas: Número</li> <li>• Tipo de entrada: Cadena de caracteres (50)</li> <li>• Método de pago (tarjeta, PayPal o transferencia): Cadena de caracteres (50)</li> <li>• Número de tarjeta: Número de 16 dígitos</li> <li>• Fecha de caducidad <b>Tarjeta</b>: Tipo Date (MM/AA).</li> <li>• Código de seguridad <b>Tarjeta</b> (CVC): Número de 3 o 4 dígitos</li> </ul>
	<b>Requisito de datos</b>	<b>Número de referencia</b>	RDW1.3

		<b>Descripción</b>	Registro de la compra en la base de datos (RDE1.3), <b>escritura del localizador generado de la entrada en la base de datos</b> , actualización de aforo disponible y validación del pago.
		<b>Número de referencia</b>	<b>RDR1.1</b>
		<b>Descripción</b>	Lectura en la Base de Datos del localizador de la entrada.
<b>Salida</b>	<b>Agente externo</b>	Cliente	
	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDS1.1</b>
<b>Restricciones semánticas</b>		<b>Composición</b>	<ul style="list-style-type: none"> <li>Localizador de la entrada: Cadena de alfanumericos (15)</li> </ul>
		<b>Número de referencia</b>	<b>RS1.4</b>
		<b>Descripción</b>	La compra no se confirmará si el número de entradas solicitadas supera el aforo disponible. Dicha comprobación la hace el SGBD en <b>RDW1.3</b>
		<b>Número de referencia</b>	<b>RS1.5</b>
		<b>Descripción</b>	Si el pago no se valida correctamente; en ese caso, la reserva se cancela. Dicha comprobación la hace el SGBD en <b>RDW1.3</b>

<b>Requisito funcional</b>	
<b>Número de referencia</b>	RF1.4
<b>Nombre del requisito</b>	Gestión de encuestas de satisfacción

<b>Descripción del requisito</b>	Permite a los clientes responder encuestas de satisfacción tras haber asistido a un evento.		
<b>Entrada</b>	<b>Agente externo</b>	Cliente	
	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDE1.4</b>
		<b>Composición</b>	<ul style="list-style-type: none"> <li>Nombre del evento: Cadena de caracteres (300)</li> <li>Localización del evento: Cadena de caracteres (50)</li> <li>Fecha del evento: Tipo Date (DD/MM/AA)</li> <li>Respuestas encuesta: Texto/Número según la pregunta</li> </ul>
<b>Datos internos</b>	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDW1.4</b>
		<b>Descripción</b>	Almacenamiento de las respuestas de los clientes para generar estadísticas de satisfacción y de la validación de la entrada del cliente
<b>Salida</b>	<b>Agente externo</b>	<b>Administrador del festival</b>	
	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDS1.4.2</b>
		<b>Composición</b>	<ul style="list-style-type: none"> <li>Estadísticas para la organización.</li> </ul>
<b>Restricciones semánticas</b>		<b>Número de referencia</b>	<b>RS1.6</b>
		<b>Descripción</b>	Solo podrán responder la encuesta los clientes cuya entrada haya sido validada en el evento. Dicha validación se guarda en RDW1.4 gracias al SGBD

<b>Requisito funcional</b>	
<b>Número de referencia</b>	RF1.5

<b>Nombre del requisito</b>	Gestión de abonos, descuentos, seguros y promociones personalizadas		
<b>Descripción del requisito</b>	Permite al cliente acceder a abonos, aplicar descuentos, contratar seguros adicionales y aprovechar promociones personalizadas.		
<b>Entrada</b>	<b>Agente externo</b>	Cliente	
	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDE1.5.1</b>
		<b>Composición</b>	<ul style="list-style-type: none"> <li>Selección de abono, descuento, promoción o seguro: cadena de caracteres (100)</li> </ul>
	<b>Agente externo</b>	<b>Administrador del festival</b>	
	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDE1.5.2</b>
		<b>Composición</b>	<ul style="list-style-type: none"> <li>Comprobación de abonos, descuento, promoción o seguro</li> </ul>
<b>Datos internos</b>	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDW1.5</b>
		<b>Descripción</b>	Registro de abonos, descuento, promoción o seguro en la base de datos.
<b>Salida</b>	<b>Agente externo</b>		
	<b>Requisito de datos</b>	<b>Número de referencia</b>	
		<b>Composición</b>	
<b>Restricciones semánticas</b>		<b>Número de referencia</b>	<b>RS1.7</b>
		<b>Descripción</b>	<ul style="list-style-type: none"> <li>Un abono sólo podrá adquirirse si se cumplen los requisitos establecidos (número mínimo de entradas ).</li> </ul>

		<b>Número de referencia</b>	<b>RS1.8</b>
		<b>Descripción</b>	<ul style="list-style-type: none"> <li>• Un descuento o promoción solo será aplicable si se cumplen los requisitos establecidos ( compra anticipada, cliente frecuente, fecha válida, documento válido de familia numerosa).</li> </ul>

<b>Requisito funcional</b>			
<b>Número de referencia</b>	RF1.6		
<b>Nombre del requisito</b>	Servicio de atención al cliente		
<b>Descripción del requisito</b>	Permite a los clientes registrar consultas, reclamaciones o solicitudes de devolución relacionadas con el servicio o eventos.		
<b>Entrada</b>	<b>Agente externo</b>	Cliente	
		<b>Número de referencia</b>	<b>RDE1.6.1</b>
<b>Requisito de datos</b>			

		<b>Composición</b>	<ul style="list-style-type: none"> <li>• Tipo de solicitud (Consulta/Reclamación/Devolución): Cadena de caracteres (12)</li> <li>• Descripción de la solicitud : Texto</li> </ul>	
	<b>Agente externo</b>	<b>Administrador del festival</b>		
	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDE1.6.2</b>	
		<b>Composición</b>	<ul style="list-style-type: none"> <li>• Comprobación de la Solicitud: Texto</li> </ul>	
<b>Datos internos</b>	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDW1.6</b>	
		<b>Descripción</b>	Registro de la solicitud en la base de datos y asignación de dicha solicitud al área de la organización correspondiente (al departamento de incidencias,al departamento de devoluciones...etc).	
	<b>Agente externo</b>			
	<b>Salida</b>	<b>Número de referencia</b>		
	<b>Requisito de datos</b>	<b>Composición</b>		
		<b>Número de referencia</b>	<b>RS1.9</b>	
		<b>Descripción</b>	Las devoluciones sólo serán aceptadas si cumplen la política establecida (plazo permitido o si el evento ha sido cancelado), de lo contrario, serán rechazadas. Esto es comprobado en el departamento de devoluciones que enviará la comprobación de la solicitud.	
		<b>Número de referencia</b>	<b>RS1.10</b>	
		<b>Descripción</b>	Solo se puede realizar una reclamación si el cliente ha asistido al evento. Para ello, se comprueba la validación de la entrada. Esto es comprobado en el departamento de Reclamaciones que enviará la comprobación de la solicitud.	



### 3. Gestión de artistas

Requisito funcional			
Número de referencia	RF2.1		
Nombre del requisito	Registro/Modificación de Artistas		
Descripción del requisito	Permite a la organización tener registrados a los artistas con sus datos personales, caché y género musical, o modificar los datos de su cuenta		
Entrada	Agente externo	Responsable del artista	
	Requisito de datos	Número de referencia Composición	RDE2.1 <ul style="list-style-type: none"> <li>● Nombre: Cadena de caracteres (40)</li> <li>● Apellidos: Cadena de caracteres (60)</li> <li>● DNI/NIF: Cadena de alfanumericos (9)</li> <li>● Correo electrónico: Cadena de caracteres ( 50)</li> <li>● Teléfono: Cadena de numeros (20)</li> <li>● Dirección: Cadena de caracteres (100)</li> <li>● Cache: Cadena de números (7)</li> <li>● Género musical: Cadena de caracteres (15)</li> </ul>
Datos internos	Requisito de datos	Número de referencia Descripción	RDW2.1 Almacenamiento y actualización de los datos del artista en la base de datos.

	<b>Agente externo</b>		
<b>Salida</b>	<b>Requisito de datos</b>	<b>Número de referencia</b>	
		<b>Composición</b>	•
<b>Restricciones semánticas</b>		<b>Número de referencia</b>	RS2.1
			<ul style="list-style-type: none"> <li>• No se podrá registrar un artista si ya existe otro con el mismo DNI/NIF o correo electrónico.</li> </ul>
		<b>Número de referencia</b>	RS2.2
		<b>Descripción</b>	<ul style="list-style-type: none"> <li>• No se podrá registrar un artista si ya existe otro con el mismo correo electrónico.</li> </ul>

<b>Requisito funcional</b>			
<b>Número de referencia</b>	RF2.2		
<b>Nombre del requisito</b>	Baja de artistas		
<b>Descripción del requisito</b>	Permite a los artistas solicitar la baja de la plataforma		
<b>Entrada</b>	<b>Agente externo</b>	Responsable del artista	
	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDE2.2</b>
		<b>Composición</b>	<ul style="list-style-type: none"> <li>• DNI/NIF: Cadena de alfanumericos (9)</li> <li>• Correo electrónico: Cadena de caracteres ( 50)</li> </ul>
<b>Datos internos</b>	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDW2.2</b>
		<b>Descripción</b>	Eliminación de los datos del artista en la base de datos.
<b>Salida</b>	<b>Agente externo</b>		
	<b>Requisito de datos</b>	<b>Número de referencia</b>	

		<b>Composición</b>	•
<b>Restricciones semánticas</b>		<b>Número de referencia</b>	<b>RS2.3</b>
		<b>Descripción</b>	<ul style="list-style-type: none"> <li>• No se podrá dar de baja un artista si tiene una actuación pendiente</li> </ul>

<b>Requisito funcional</b>			
<b>Número de referencia</b>	RF2.3		
<b>Nombre del requisito</b>	Registro de actuaciones		
<b>Descripción del requisito</b>	Permite a la organización tener todas las actuaciones encajadas en distintas franjas horarias y escenarios, de forma que no se pisen ni en horario ni en lugar.		
<b>Entrada</b>	<b>Agente externo</b>	Administrador del festival	
	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDE2.3</b>
		<b>Composición</b>	<ul style="list-style-type: none"> <li>• Dia: Tipo DATE</li> <li>• Hora: Tipo TIME</li> <li>• DNI: Cadena de alfanumericos (9)</li> <li>• Número escenario: Cadena de números (2)</li> </ul>
<b>Datos internos</b>	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDW2.3</b>
		<b>Descripción</b>	Almacenamiento en la base de datos de la asignación de cada uno de los escenarios
<b>Salida</b>	<b>Agente externo</b>		
	<b>Requisito de datos</b>	<b>Número de referencia</b>	

		<b>Composición</b>	•
<b>Restricciones semánticas</b>		<b>Número de referencia</b>	<b>RS2.4</b>
		<b>Descripción</b>	<ul style="list-style-type: none"> <li>El numero de escenario es único</li> </ul>
		<b>Número de referencia</b>	<b>RS2.5</b>
		<b>Descripción</b>	<ul style="list-style-type: none"> <li>No se podrá registrar una nueva actuación si ya existe un artista que tenga asociada la misma franja horaria y escenario que la que se pretende crear.</li> </ul>

<b>Requisito funcional</b>			
<b>Número de referencia</b>	RF2.4		
<b>Nombre del requisito</b>	Asignación de los camerinos		
<b>Descripción del requisito</b>	Permite a la organización realizar la distribución de los camerinos a cada uno de los artistas		
<b>Entrada</b>	<b>Agente externo</b>	Administrador del festival	
	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDE2.4</b>
		<b>Composición</b>	<ul style="list-style-type: none"> <li>Identificación: Cadena de números (2)</li> <li>DNI: Cadena de alfanumericos (9)</li> <li><b>• Dia: Tipo DATE</b></li> <li><b>• Hora: Tipo TIME</b></li> </ul>

<b>Datos internos</b>	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDW2.4</b>
		<b>Descripción</b>	Almacenamiento en la base de datos de la asignación de cada uno de los camerinos
<b>Salida</b>	<b>Agente externo</b>		
	<b>Requisito de datos</b>	<b>Número de referencia</b>	
		<b>Composición</b>	•
<b>Restricciones semánticas</b>		<b>Número de referencia</b>	<b>RS2.6</b>
		<b>Descripción</b>	<ul style="list-style-type: none"> <li>• La identificación es única</li> </ul>
		<b>Número de referencia</b>	<b>RS2.7</b>
		<b>Descripción</b>	<ul style="list-style-type: none"> <li>• No se podrá asociar un camerino a un artista durante una franja horaria si ya hay otro artista asociado a dicho camerino en esa franja horaria</li> </ul>

<b>Requisito funcional</b>			
<b>Número de referencia</b>	RF2.5		
<b>Nombre del requisito</b>	Gestión de contratos		
<b>Descripción del requisito</b>	Permite a la organización realizar un contrato que recoja los acuerdos con el artista.		
<b>Entrada</b>	<b>Agente externo</b>	Administrativo	
	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDE2.5</b>

		<b>Composición</b>	<ul style="list-style-type: none"> <li>• ID : number(3)</li> <li>• Fecha inicio : DATE</li> <li>• Fecha fin : DATE</li> <li>• Importe acordado : number(10,2)</li> <li>• Datos bancarios : cadena de alfanumericos (34)</li> </ul>
<b>Datos internos</b>	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDW2.5</b>
		<b>Descripción</b>	Se almacena el contrato en la base de datos
<b>Salida</b>	<b>Agente externo</b>		
	<b>Requisito de datos</b>	<b>Número de referencia</b>	
<b>Restricciones semánticas</b>		<b>Composición</b>	•
		<b>Número de referencia</b>	<b>RS2.7</b>
		<b>Descripción</b>	• El ID es único
		<b>Número de referencia</b>	<b>RS2.8</b>
		<b>Descripción</b>	• La fecha fin debe ser posterior a la fecha inicio

<b>Requisito funcional</b>		
<b>Número de referencia</b>	<b>RF2.6</b>	
<b>Nombre del requisito</b>	Gestión de pagos	
<b>Descripción del requisito</b>	Permite a la organización realizar los pagos acordados en los contratos a los artistas	
<b>Entrada</b>	<b>Agente externo</b>	Administrativo

	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDE2.6</b>
		<b>Composición</b>	<ul style="list-style-type: none"> <li>● ID : cadena de números (3)</li> <li>● NumPago: number(2)</li> <li>● Importe acordado : number (10,2)</li> <li>● Método de pago : cadena de caracteres (20)</li> <li>● Tipo de pago : cadena de caracteres (15)</li> <li>● Duración del fraccionado : number (2)</li> </ul>
<b>Datos internos</b>	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDR2.1</b>
		<b>Descripción</b>	Se lee el ID referente al contrato a pagar
		<b>Número de referencia</b>	<b>RDW2.6</b>
		<b>Descripción</b>	Se registra el pago en la base de datos
<b>Salida</b>	<b>Agente externo</b>		
	<b>Requisito de datos</b>	<b>Número de referencia</b>	
		<b>Composición</b>	●
<b>Restricciones semánticas</b>		<b>Número de referencia</b>	<b>RS2.9</b>
		<b>Descripción</b>	Duración del fraccionado es por defecto 1 y no podra ser mas de 1 ya que el tipo de pago de artista nunca será fraccionado
		<b>Número de referencia</b>	<b>RS2.10</b>
		<b>Descripción</b>	NumPago debe ser menor o igual que duración del fraccionado

## 4. Gestión de patrocinadores

Requisito funcional			
Número de referencia	RF3.1		
Nombre del requisito	Registro de patrocinadores		
Descripción del requisito	Registramos los datos de contacto de las empresas patrocinadoras en la base de datos.		
Entrada	Agente externo	Responsable de patrocinadores.	
	Requisito de datos	Número de referencia	RDE3.1
		Composición	<ul style="list-style-type: none"> <li>● Nombre Empresa: Cadena de caracteres (30)</li> <li>● Nombre persona de contacto : Cadena de caracteres (50)</li> <li>● Teléfono de contacto : Cadena de numeros (20)</li> <li>● Correo Electrónico de contacto : Cadena de caracteres(50)</li> <li>● <b>DNI persona de contacto</b> : Cadena de alfanumericos(9)</li> </ul>
Datos internos	Requisito de datos	Número de referencia	RDW3.1
		Descripción	Se registra al patrocinador en la base de datos
Salida	Agente externo		
	Requisito de datos	Número de referencia	
		Composición	
Restricciones semánticas		Número de referencia	RS3.1
		Descripción	<ul style="list-style-type: none"> <li>● El nombre de la empresa es único</li> </ul>

Requisito funcional			
Número de referencia	RF3.2		
Nombre del requisito	Consulta de patrocinadores		
Descripción del requisito	Muestra los datos de los patrocinadores registrados.		
Entrada	Agente externo		
	Requisito de datos	Número de referencia	
		Composición	•
Datos internos	Requisito de datos	Número de referencia	RDR3.1
		Descripción	Se leen los patrocinadores de la base de datos
Salida	Agente externo	Responsable de patrocinadores	
	Requisito de datos	Número de referencia	RDS3.1
		Composición	<ul style="list-style-type: none"> <li>• Nombre Empresa</li> <li>• Correo electrónico de contacto</li> </ul>
Restricciones semánticas		Número de referencia	
		Descripción	

Requisito funcional			
Número de referencia	RF3.3		
Nombre del requisito	Gestión de Contratos		
Descripción del requisito	Dada de alta el contrato en la base de datos del sistema. El ID asignado a cada contrato es elegido por el responsable de patrocinadores		
Entrada	Agente externo	Administrativo	
	Requisito de datos	Número de referencia	RDE3.2
Datos internos	Requisito de datos	Composición	<ul style="list-style-type: none"> <li>• ID : number(3)</li> <li>• Fecha inicio : DATE</li> <li>• Fecha fin : DATE</li> <li>• Importe acordado : number(10,2)</li> <li>• Datos bancarios : cadena de alfanumericos (34)</li> </ul>
		Número de referencia	RDW3.2
	Requisito de datos	Descripción	Se escribe el contrato en la base de datos
Salida	Agente externo		
	Requisito de datos	Número de referencia	
Restricciones		Composición	•
		Número de referencia	RS3.2

<b>semánticas</b>		<b>Descripción</b>	El ID es único
		<b>Número de referencia</b>	<b>RS3.3</b>
		<b>Descripción</b>	La fecha fin debe ser posterior a la fecha de inicio

<b>Requisito funcional</b>	
<b>Número de referencia</b>	RF3.4

<b>Nombre del requisito</b>	Control de beneficios a cada patrocinador		
<b>Descripción del requisito</b>	Se introduce en la base de datos los beneficios acordados con cada patrocinador y una breve descripción de ellos(número de menciones, lugar ocupado en los carteles,...). Se usa el nombre de la empresa para saber a qué empresa son acordados los beneficios descritos.		
<b>Entrada</b>	<b>Agente externo</b>	Responsable de patrocinadores	
	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDE3.3</b>
		<b>Composición</b>	<ul style="list-style-type: none"> <li>• Nombre Empresa : cadena de caracteres (30)</li> <li>• Tipo de beneficios : cadena de caracteres (50)</li> <li>• Descripción de los beneficios : cadena de caracteres (150)</li> </ul>
<b>Datos internos</b>	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDW3.3</b>
		<b>Descripción</b>	Se registra los beneficios en la base de datos
<b>Salida</b>	<b>Agente externo</b>		
	<b>Requisito de datos</b>	<b>Número de referencia</b>	
		<b>Composición</b>	<ul style="list-style-type: none"> <li>•</li> </ul>
<b>Restricciones semánticas</b>		<b>Número de referencia</b>	<b>RS3.1</b>
		<b>Descripción</b>	<ul style="list-style-type: none"> <li>• El nombre de la empresa es único</li> </ul>

#### Requisito funcional

<b>Número de referencia</b>	RF3.5		
<b>Nombre del requisito</b>	Gestión de pagos		
<b>Descripción del requisito</b>	Permite realizar pagos acordados en los contratos a los patrocinadores.		
<b>Entrada</b>	<b>Agente externo</b>	Administrativo	
	<b>Requisito de datos</b>	<b>Número de referencia</b> RDE3.4 <b>Composición</b> <ul style="list-style-type: none"> <li>● ID : cadena de números(3)</li> <li>● NumPago : number(2)</li> <li>● Importe acordado : number(10,2)</li> <li>● Método de pago : cadena de caracteres (20)</li> <li>● Tipo de pago : cadena de caracteres (15)</li> <li>● Duración del fraccionado : number (2)</li> </ul>	
<b>Datos internos</b>	<b>Requisito de datos</b>	<b>Número de referencia</b> RDW3.4	
		<b>Descripción</b>	Se registra el pago en la base de datos
<b>Salida</b>	<b>Agente externo</b>		
	<b>Requisito de datos</b>	<b>Número de referencia</b>	
		<b>Composición</b>	●
<b>Restricciones semánticas</b>		<b>Número de referencia</b> RS3.4	
		<b>Descripción</b>	Duración del fraccionado es por defecto 1 y no puede ser más de 1 si el tipo de pago no es fraccionado

		<b>Número de referencia</b>	RS3.5
		<b>Descripción</b>	Tipo de pago debe ser completo o fraccionado
		<b>Número de referencia</b>	RS3.6
		<b>Descripción</b>	NumPago debe ser menor o igual que duración del fraccionado

## 5. Gestión de servicios

Requisito funcional			
Número de referencia	RF4.1		
Nombre del requisito	Asignar proveedores a servicios		
Descripción del requisito	<p>Permite asignar un proveedor previamente dado de alta a un determinado servicio del festival. Para ello se introducirán su NIF (es único para cada uno), el servicio/función(puede ser también para una bebida concreta...) y si su servicio requiere de empleados, un entero con el número de empleados y un listado con el nombre y DNI de cada empleado. Como salida se mostrará un listado con los proveedores actuales a modo de comprobación. Si no se proporciona ningún parámetro el listado que pasará de salida será el de los proveedores actuales, a modo de consulta. <b>El servicio irá identificado por el nombre, y la relación/asignación se dará para únicamente para el área especificada.</b> Esto permite distinguir servicios entre áreas, podemos tener un escenario patrocinado por un proveedor, y otro patrocinado por otro proveedor que ofrece un servicio de bebidas también.</p>		
Entrada	<b>Agente externo</b> <b>Requisito de datos</b>	<b>Número de referencia</b> <b>Composición</b>	<b>RDE4.1</b> <ul style="list-style-type: none"> <li>• NIF del proveedor/empresa: cadena de 9 caracteres (8 dígitos y una letra)</li> <li>• Servicio: cadena de 60 caracteres (las bebidas tienen nombres largos)</li> <li>• Número de empleados: entero (por defecto será 0 si su servicio no ofrece empleados)</li> <li>• Listado de empleados: cada entrada estará compuesta por un nombre (cadena de máximo 12 caracteres), apellidos (cadena de caracteres), teléfono (número de 9 dígitos máx) y un DNI (cadena de 9 caracteres)</li> </ul>
Datos internos	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDR4.1</b>

		<b>Descripción</b>	<ul style="list-style-type: none"> <li>Lectura de los datos de cada una de las tuplas de la tabla proveedor -servicios para mostrarlo en la salida.</li> </ul>
		<b>Número de referencia</b>	<b>RDW4.1.1</b>
		<b>Descripción</b>	Adhesión proveedor-servicio y adhesión al sistema de los datos de los empleados dados por el proveedor en caso de haber dado algunos
		<b>Número de referencia</b>	<b>RDW4.1.2</b>
		<b>Descripción</b>	Adhesión al sistema de los datos de los empleados dados por el proveedor en caso de haber dado algunos
		<b>Número de referencia</b>	<b>RDW4.1.3</b>
		<b>Descripción</b>	Se almacena la relación entre el empleado y la relación proveedor-servicio para poder identificar que empleado está asignado a que servicio
<b>Salida</b>	<b>Agente externo</b>	Responsable de proveedores	
	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDS4.1</b>
		<b>Composición</b>	<ul style="list-style-type: none"> <li>Listado de proveedores, cada entrada se compone de:           <ul style="list-style-type: none"> <li>NIF del proveedor: cadena de 9 caracteres</li> <li>Nombre del proveedor: cadena de 100 caracteres</li> <li>Servicio: cadena de 60 caracteres (las bebidas tienen nombres largos)</li> <li>Número de empleados: entero (por defecto será 0 si su servicio no ofrece empleados)</li> </ul> </li> </ul>
<b>Restricciones semánticas</b>		<b>Número de referencia</b>	<b>RS4.1</b>
		<b>Descripción</b>	Si se introduce un CIF inválido o que no haya sido registrado previamente daña un error.

Requisito funcional			
Número de referencia	RF4.2		
Nombre del requisito	Asignar turnos del personal		
Descripción del requisito	Esta operación permite al responsable de servicios, asignar un turno a un determinado empleado así como su área de trabajo, introduciendo su DNI y su función, así como una hora de inicio, una de salida y el área en la que trabajará. Se mostrará una entrada compuesta por el DNI del empleado, su función , la hora de inicio, salida y el área de trabajo, a modo de comprobación. Se puede modificar cuantas veces queramos. <b>Los empleados provistos por los proveedores se consideran empleados externos no personal del festival.</b>		
Entrada	Agente externo	Responsable de personal	
	Requisito de datos	<p>Número de referencia</p> <p>Composición</p>	<p>RDE4.2</p> <ul style="list-style-type: none"> <li>• DNI del empleado: Cadena de 9 caracteres</li> <li>• Hora de inicio: tipo TIME</li> <li>• Hora de salida: tipo TIME</li> <li>• Área de trabajo: entero</li> <li>• Fecha del turno: tipo DATE</li> </ul>
Datos internos	Requisito de datos	Número de referencia	RDR4.2
		Descripción	<ul style="list-style-type: none"> <li>• Se leen los datos de cada turno de personal asignado para mostrarlo por salida</li> </ul>
		Número de referencia	RDW4.2
		Descripción	Se añadirá a la base de datos del sistema el turno con hora de inicio y salida,

			así como el área de trabajo del empleado indicado por DNI empleado y su función.
<b>Salida</b>	<b>Agente externo</b>	Responsable de personal	
	<b>Requisito de datos</b>	<b>Número de referencia</b> RDS4.2 <b>Composición</b> <ul style="list-style-type: none"> <li>• DNI del empleado: Cadena de 9 caracteres</li> <li>• Función: cadena de 60 caracteres</li> <li>• Hora de inicio: tipo TIME</li> <li>• Hora de salida: tipo TIME</li> <li>• Área de trabajo: entero</li> <li>• Fecha del turno: tipo DATE</li> </ul>	
<b>Restricciones semánticas</b>		<b>Número de referencia</b> RS4.2	
		<b>Descripción</b>	Si el DNI introducido no se encuentra en el sistema, o no se corresponde con la función introducida, dará error

<b>Requisito funcional</b>	
<b>Número de referencia</b>	RF4.3
<b>Nombre del requisito</b>	Reabastecer suministros
<b>Descripción del requisito</b>	Permite a los empleados notificar el nivel de cada suministro en una determinada área, pudiendo reabastecer un área y reducir el nivel de ese suministro en el almacén... Para ello basta con introducir el id del suministro previamente registrado, este lo generará el sistema

	mismo, el área y el nivel que corresponderá a un entero. De tal forma que así, cuando se gasta un suministro se puede notificar, para que otro lo reabastezca, o incluso tú mismo, notificando su extracción en el almacén también. Cada empleado también notificará los niveles al acabar el turno para que quién comience el siguiente turno lo tenga en cuenta.		
<b>Entrada</b>	<b>Agente externo</b>	Personal del festival	
	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDE4.3</b>
		<b>Composición</b>	<ul style="list-style-type: none"> <li>● id del suministro: entero</li> <li>● área: entero</li> <li>● nivel: entero</li> </ul>
<b>Datos internos</b>	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDW4.3</b>
		<b>Descripción</b>	Se modifica el dato <b>nivel</b> correspondiente al suministro de la base de datos del sistema con id y área introducidas.
<b>Salida</b>	<b>Agente externo</b>		
	<b>Requisito de datos</b>	<b>Número de referencia</b>	
		<b>Composición</b>	
<b>Restricciones semánticas</b>		<b>Número de referencia</b>	<b>RS4.3</b>
		<b>Descripción</b>	Si el id del suministro y el área (son una clave compuesta) no se corresponden con ningún suministro existente en esa área entonces dará error.

<b>Requisito funcional</b>	
<b>Número de referencia</b>	RF4.4
<b>Nombre del requisito</b>	Notificar incidencia.

<b>Descripción del requisito</b>	Permite a los clientes notificar cualquier tipo de incidencia ocurrida en una determinada área y servicio. Se introduce un comentario de texto exponiendo la incidencia. <b>Para poner una incidencia tienes que ser cliente registrado y haber iniciado sesión, y no es necesario introducir DNI o correo porque los datos se asociarán usando el identificador con el que hayas iniciado sesión.</b> La incidencia que se generará tendrá un identificador único generado aleatoriamente.		
<b>Entrada</b>	<b>Agente externo</b>	Cliente	
	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDE4.4</b>
		<b>Composición</b>	Comentario de texto: cadena de 500 caracteres <b>(es el único que el usuario ha de introducir, los demás ya están en la aplicación y en la base de datos)</b>
<b>Datos internos</b>	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDW4.4</b>
		<b>Descripción</b>	Se crea la incidencia con el comentario de texto, <b>tendrá como identificador uno generado aleatoriamente que será único.</b>
<b>Salida</b>	<b>Agente externo</b>		
	<b>Requisito de datos</b>	<b>Número de referencia</b>	
		<b>Composición</b>	•
<b>Restricciones semánticas</b>		<b>Número de referencia</b>	<b>RS4.4</b>
		<b>Descripción</b>	Esta operación se realizará exclusivamente si has iniciado sesión, luego las comprobaciones necesarias se realizarán en el inicio de sesión (se da por hecho que si ha iniciado sesión el identificador es correcto)

<b>Requisito funcional</b>	
<b>Número de referencia</b>	RF4.5
<b>Nombre del requisito</b>	Enviar valoración del festival

<b>Descripción del requisito</b>	Habilita a un cliente a valorar su asistencia al festival usando como referencia el id de la entrada. Para realizarse deben introducirse el id de la entrada, un número del 0 al 5 que será la valoración, y un comentario de texto (opcional). Los datos que se usarán para relacionar el cliente y la valoración al crear la valoración serán consultados haciendo uso del identificador que haya usado para iniciar sesión, de igual forma que con las incidencias. Se consultará si la entrada pertenece al cliente y se creará una nueva valoración en el sistema con dichos datos.		
<b>Entrada</b>	<b>Agente externo</b>	Cliente	
	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDE4.5</b>
		<b>Composición</b>	<ul style="list-style-type: none"> <li>• id de la entrada: entero</li> <li>• valoración: entero del 0 al 5</li> <li>• comentario de texto: cadena de 500 caracteres</li> </ul>
<b>Datos internos</b>	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDR4.5</b>
		<b>Descripción</b>	Se consulta en la base de datos si existe alguna entrada con ese id para el identificador del cliente que está realizando la operación, de ser así la valoración constará de un identificador compuesto por el de la entrada y un id generado aleatoriamente.
		<b>Número de referencia</b>	<b>RDW4.5</b>
		<b>Descripción</b>	Almacena la valoración así como todos los datos conjuntos en la base de datos del sistema.
<b>Salida</b>	<b>Agente externo</b>		
	<b>Requisito de datos</b>	<b>Número de referencia</b>	
		<b>Composición</b>	•
<b>Restricciones semánticas</b>		<b>Número de referencia</b>	<b>RS4.5</b>
		<b>Descripción</b>	Si el email introducido no es válido no se permite la valoración Si se introduce id de entrada y no ha sido generada no permite valoración

## 6. Gestión de Infraestructura

Requisito funcional			
Número de referencia	RF5.1		
Nombre del requisito	Registro de permisos sobre la zona del festival		
Descripción del requisito	Se registran los permisos necesarios referidos a la zona en la que se hará el festival (plazas, parques, recintos feriales, salas privadas...), los permisos son dados por el ayuntamiento de la zona, dichos permisos son sobre la distribución de alimentos, sobre la construcción de escenarios...		
Entrada	Agente externo	Administrador del festival	
	Requisito de datos	Número de referencia Composición	RDE5.1 <ul style="list-style-type: none"> <li>• identificador del permiso (entero)</li> <li>• descripción del permiso (char 5000)</li> <li>• tipo de permiso (char 30) (permiso de zona, de constructora, de distribución de alimentos...)</li> </ul>
Datos internos	Requisito de datos	Número de referencia Descripción	RDW5.1 Los permisos necesarios en la base de datos
Salida	Agente externo		
	Requisito de datos	Número de referencia Composición	

<b>Restricciones semánticas</b>		<b>Número de referencia</b>	
		<b>Descripción</b>	

<b>Requisito funcional</b>			
<b>Número de referencia</b>	RF5.2		
<b>Nombre del requisito</b>	Control de aforo		
<b>Descripción del requisito</b>	Controla el aforo máximo del festival, cada vez que entre una persona aumentará en 1 el aforo actual		
<b>Entrada</b>	<b>Agente externo</b>	Personal del festival	
	<b>Requisito de datos</b>	<b>Número de referencia</b>	RDE5.2
		<b>Composición</b>	<ul style="list-style-type: none"> <li>Una cifra representando el aforo actual (int)</li> </ul>
<b>Datos internos</b>	<b>Requisito de datos</b>	<b>Número de referencia</b>	RDW5.2
		<b>Descripción</b>	La cifra aumentada o disminuida en uno
<b>Datos internos</b>	<b>Requisito de datos Agente externo</b>	<b>Número de referencia</b>	RDL5.1
		<b>Descripción</b>	La cifra de aforo máximo correspondiente al festival
<b>Salida</b>	<b>Agente externo</b>	Personal del festival	
	<b>Requisito de datos</b>	<b>Número de referencia</b>	RDS5.1
		<b>Composición</b>	<ul style="list-style-type: none"> <li>Cada vez que se actualiza la cifra, devuelve la propia cifra actualizada (int)</li> </ul>

<b>Restricciones semánticas</b>		<b>Número de referencia</b>	<b>RS5.1</b>
		<b>Descripción</b>	Si el aforo actual ha alcanzado el aforo máximo no entrarán más personas, en caso de que se vuelva a llamar al requisito funcional lanzará un mensaje de error.

<b>Requisito funcional</b>			
<b>Número de referencia</b>	RF5.3		
<b>Nombre del requisito</b>	Asignación y registro de puestos de comida.		
<b>Descripción del requisito</b>	Al inicio del festival se asignan los puestos de comida dependiendo del espacio dado, intentando distribuirlos equitativamente.		
	<b>Agente externo</b>	Administrador del festival	
<b>Entrada</b>	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDE5.3</b>
		<b>Composición</b>	<ul style="list-style-type: none"> <li>• El identificador de cada puesto de alimentos (int)</li> <li>• El nombre del puesto (15 chars)</li> <li>• El nombre de la comida que distribuyen (60 chars)</li> </ul>
<b>Datos internos</b>	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDW5.3</b>
		<b>Descripción</b>	Dados los datos anteriores, se elige la distribución que tendrá el festival y se escriben las posiciones de cada puesto en la base de datos.
<b>Datos internos</b>	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDL5.2</b>
		<b>Descripción</b>	Se leen las posiciones y los identificadores de los puestos de la base de datos
<b>Salida</b>	<b>Agente externo</b>	Personal del festival	

<b>Requisito de datos</b>		<b>Número de referencia</b>	<b>RDS5.2</b>
		<b>Composición</b>	<ul style="list-style-type: none"> <li>• se devuelve el identificador del puesto junto con su posición en el festival. (int, pos)</li> </ul>
<b>Restricciones semánticas</b>		<b>Número de referencia</b>	<b>RS5.2</b>
		<b>Descripción</b>	Si se obtiene como entrada un identificador de puesto ya escrito en la base de datos, se devuelve un mensaje de error

<b>Requisito funcional</b>			
<b>Número de referencia</b>	RF5.4		
<b>Nombre del requisito</b>	Asignación y registro de baños.		
<b>Descripción del requisito</b>	Al inicio del festival se asignan los baños dependiendo del espacio dado, intentando distribuirlos equitativamente para evitar aglomeraciones.		
<b>Entrada</b>	<b>Agente externo</b>	Administrador del festival	
	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDE5.4</b>
		<b>Composición</b>	<ul style="list-style-type: none"> <li>• El identificador de cada baño (int)</li> </ul>
<b>Datos internos</b>	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDW5.4</b>
		<b>Descripción</b>	Dados los datos anteriores, se elige la distribución que tendrá el festival y se escriben las posiciones de cada baño en la base de datos.
<b>Datos internos</b>	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDL5.3</b>

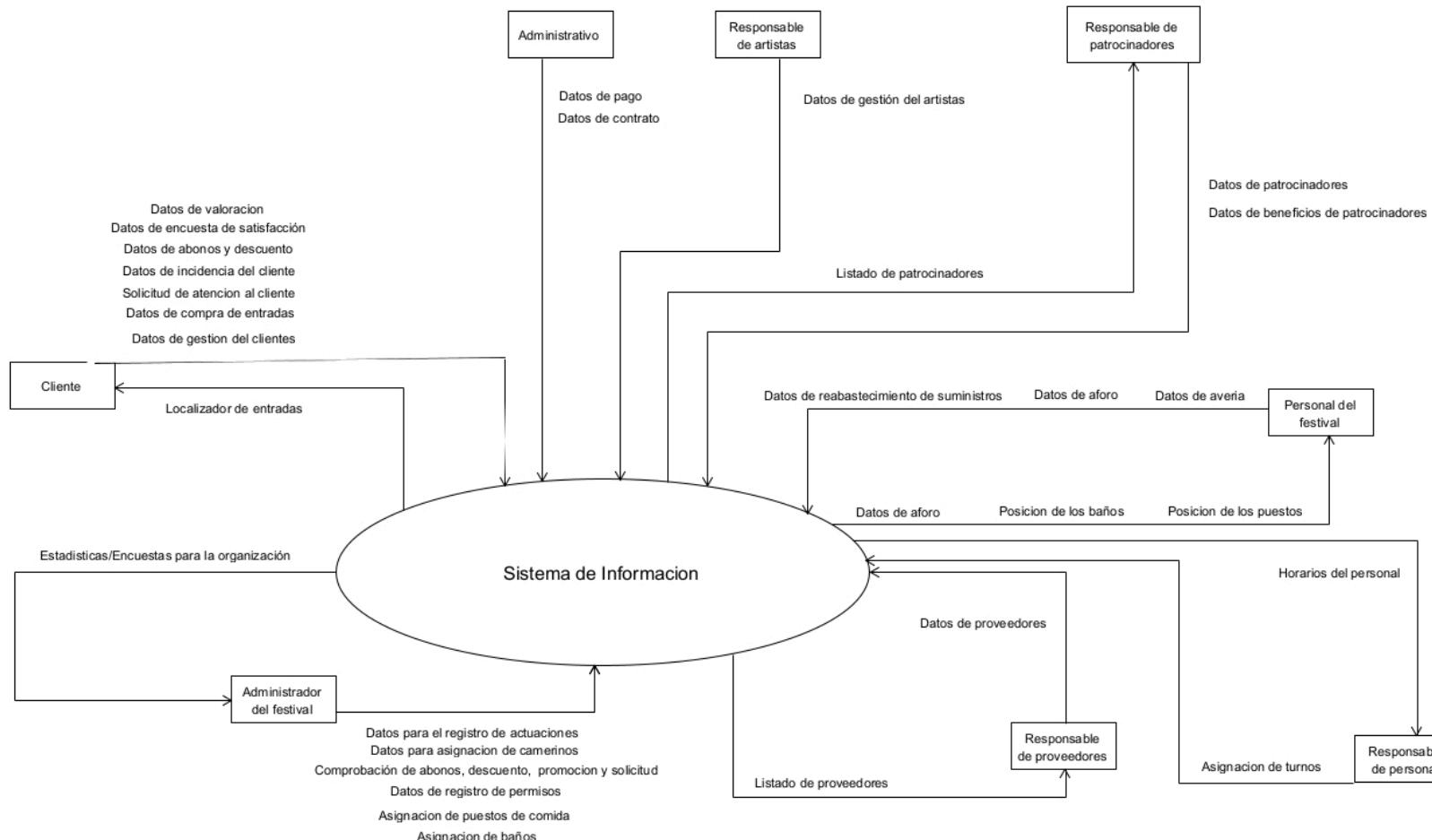
		<b>Descripción</b>	Se leen los identificadores y las posiciones de los baños del festival
<b>Salida</b>	<b>Agente externo</b>	Personal del festival	
	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDS5.3</b>
		<b>Composición</b>	<ul style="list-style-type: none"> <li>• Se devuelve la posición de cada baño junto con su identificador (int, pos)</li> </ul>
<b>Restricciones semánticas</b>		<b>Número de referencia</b>	<b>RS5.3</b>
		<b>Descripción</b>	Si se escribe un identificador que ya estaba en la base de datos, se devuelve un mensaje de error

Requisito funcional			
<b>Número de referencia</b>	RF5.5		
<b>Nombre del requisito</b>	Control de averías del festival.		
<b>Descripción del requisito</b>	Cada vez que se encuentre una avería en el festival, se registra el incidente en la base de datos, junto con los detalles de su solución (si se ha solucionado)		
<b>Entrada</b>	<b>Agente externo</b>	<b>Cliente</b>	
	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDE5.5</b>
		<b>Composición</b>	<ul style="list-style-type: none"> <li>• Identificador de avería (int)</li> <li>• Tipo de avería (15 caracteres)</li> <li>• Posición en la que ha ocurrido (pos)</li> <li>• Forma en que se ha solucionado (si lo ha hecho)</li> </ul>

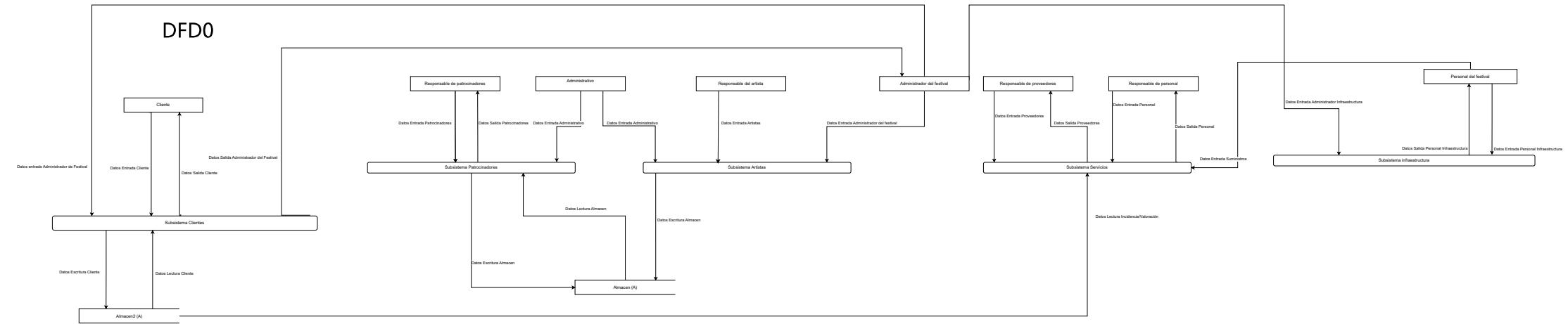
<b>Datos internos</b>	<b>Requisito de datos</b>	<b>Número de referencia</b>	<b>RDW5.5</b>
		<b>Descripción</b>	Se almacena en la base de datos la avería correspondiente
<b>Salida</b>	<b>Agente externo</b>		
	<b>Requisito de datos</b>	<b>Número de referencia</b>	
		<b>Composición</b>	
<b>Restricciones semánticas</b>		<b>Número de referencia</b>	
		<b>Descripción</b>	

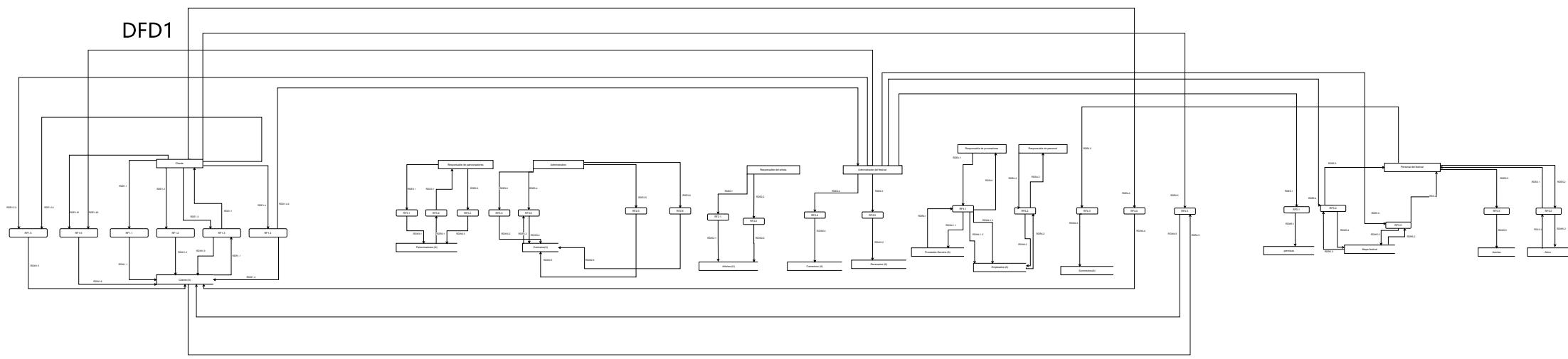
## CAJA NEGRA

venidad Granada))

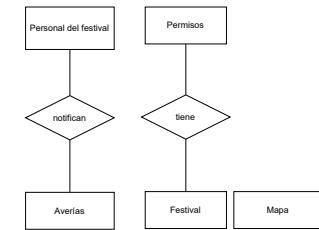
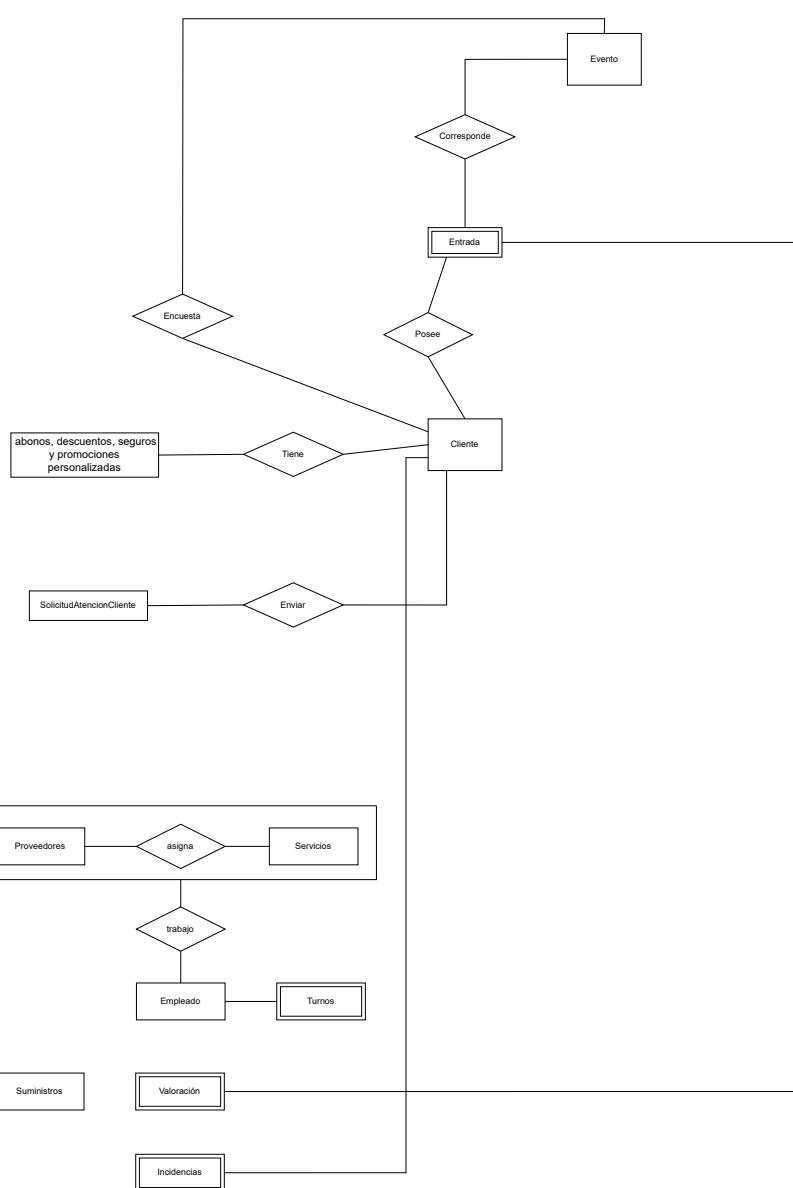
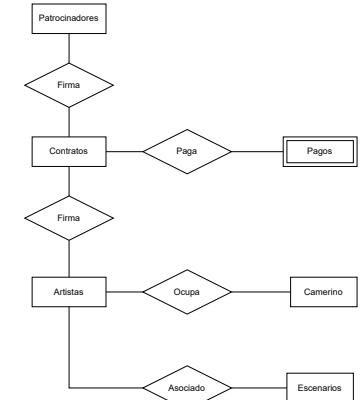


# DFD0

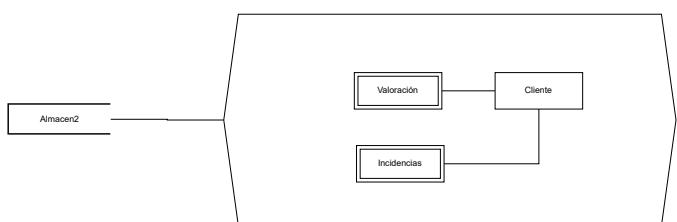
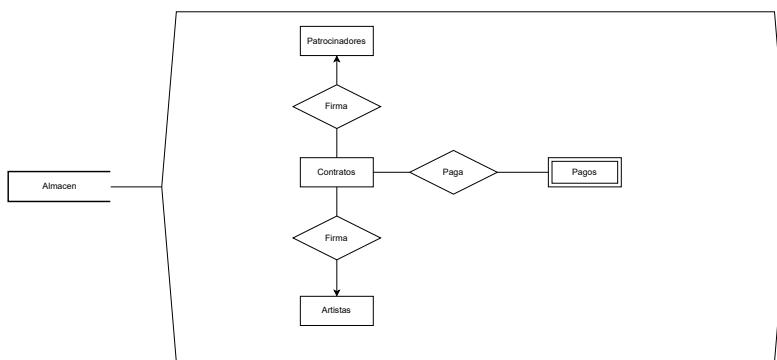
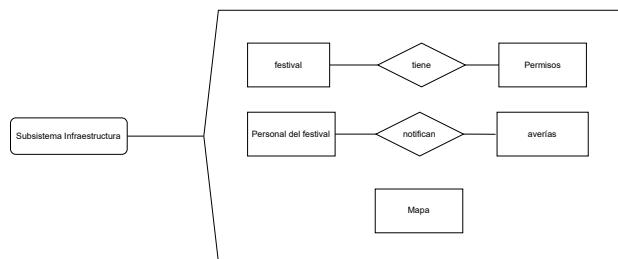
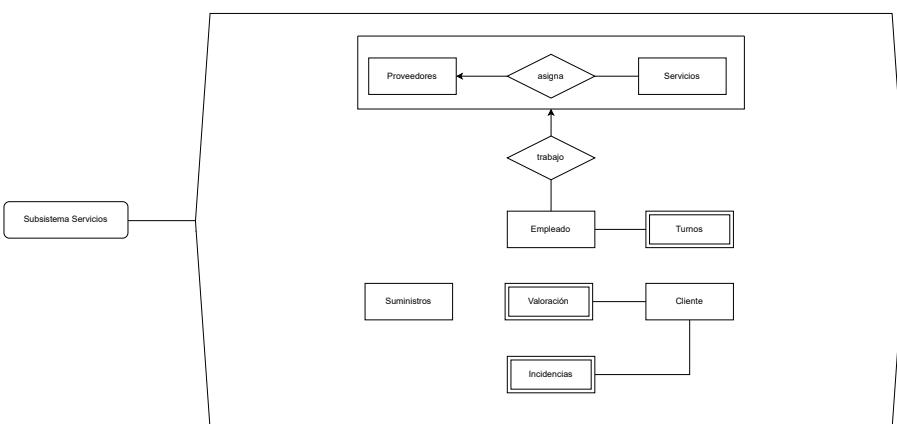
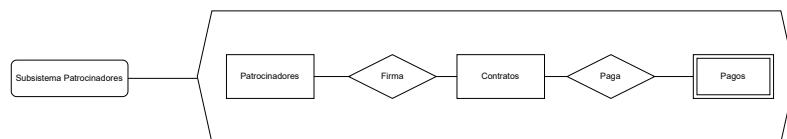
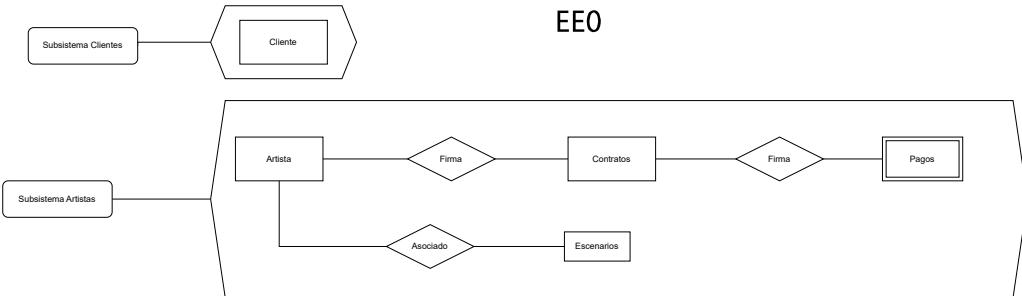




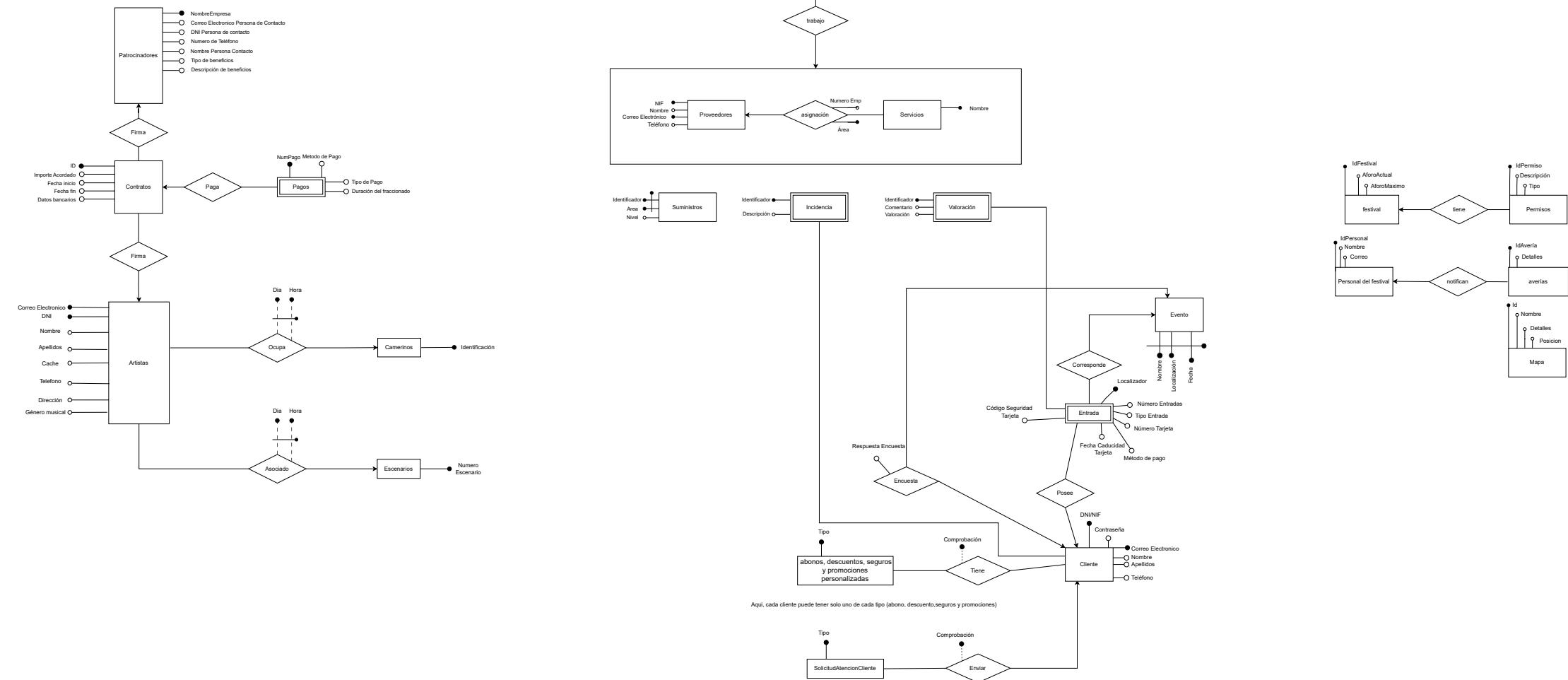
# ER0

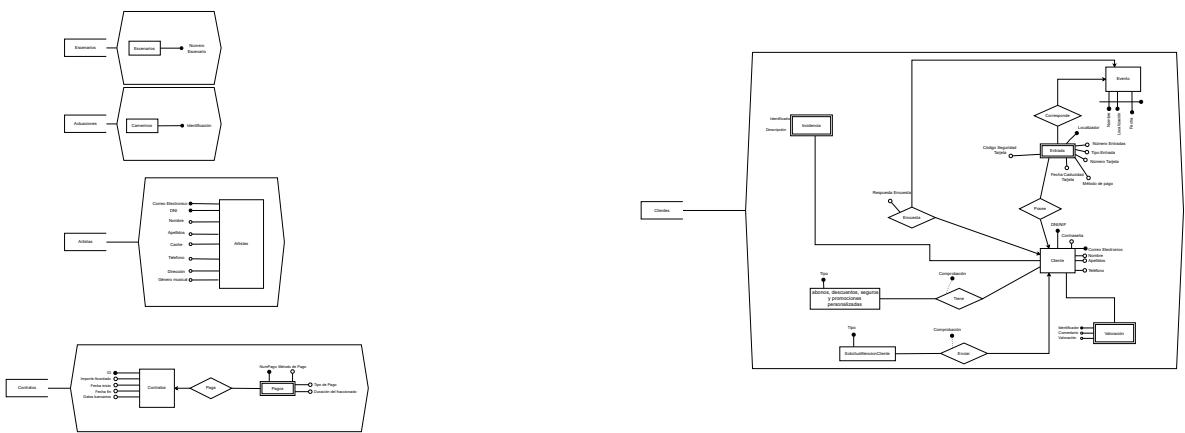
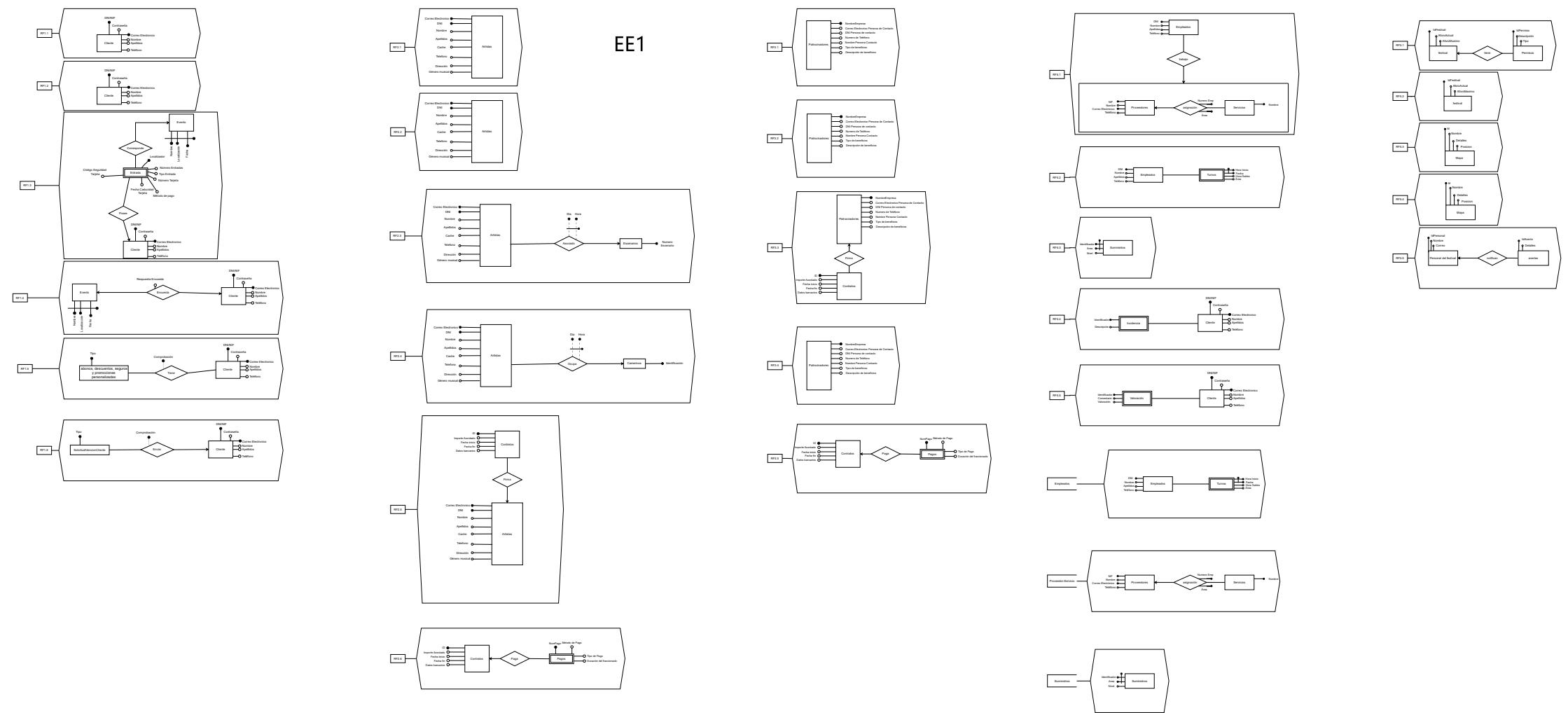


# EEO

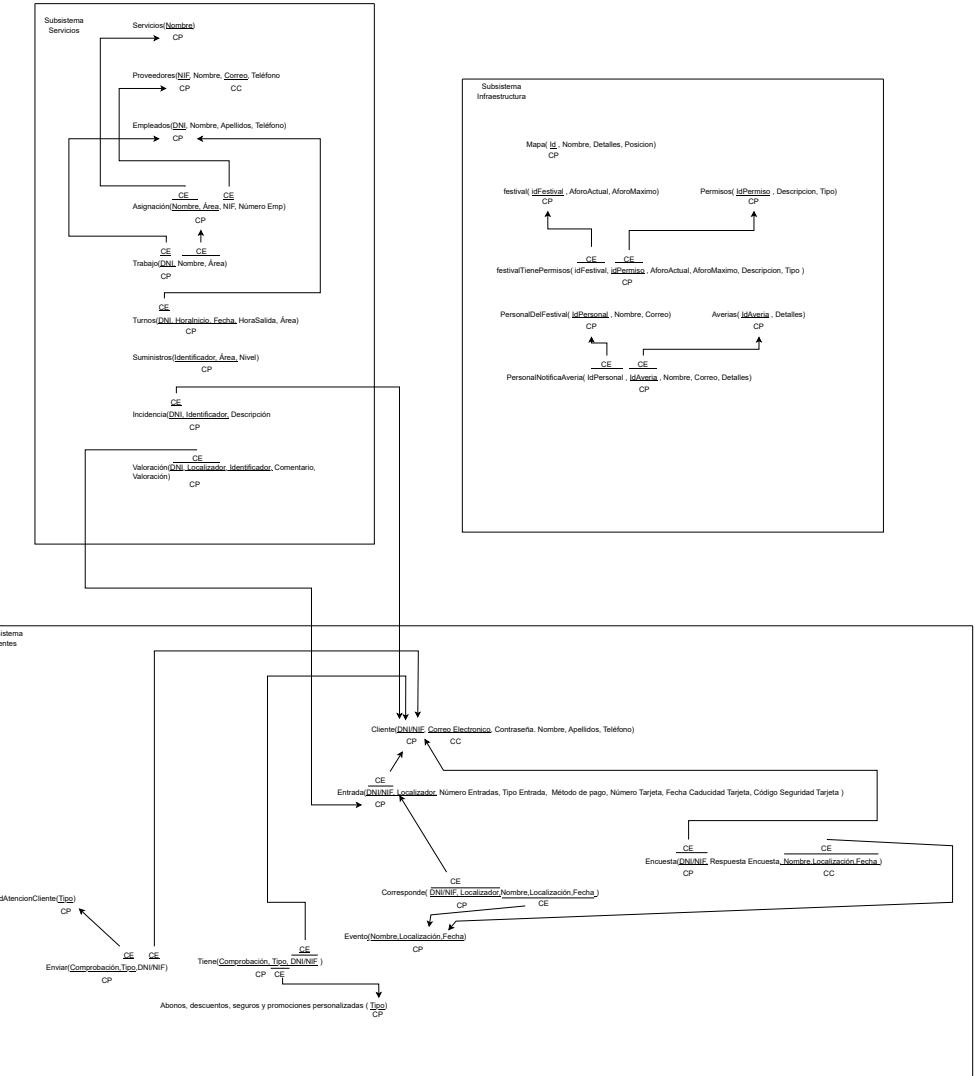
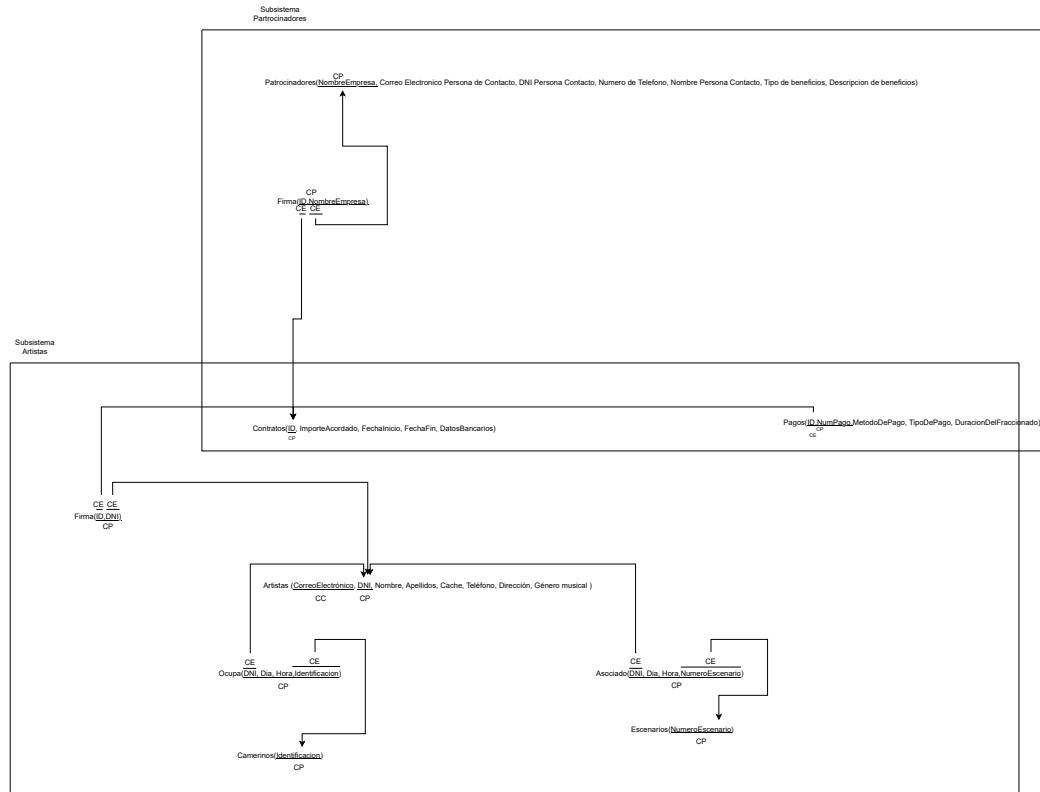


ER1



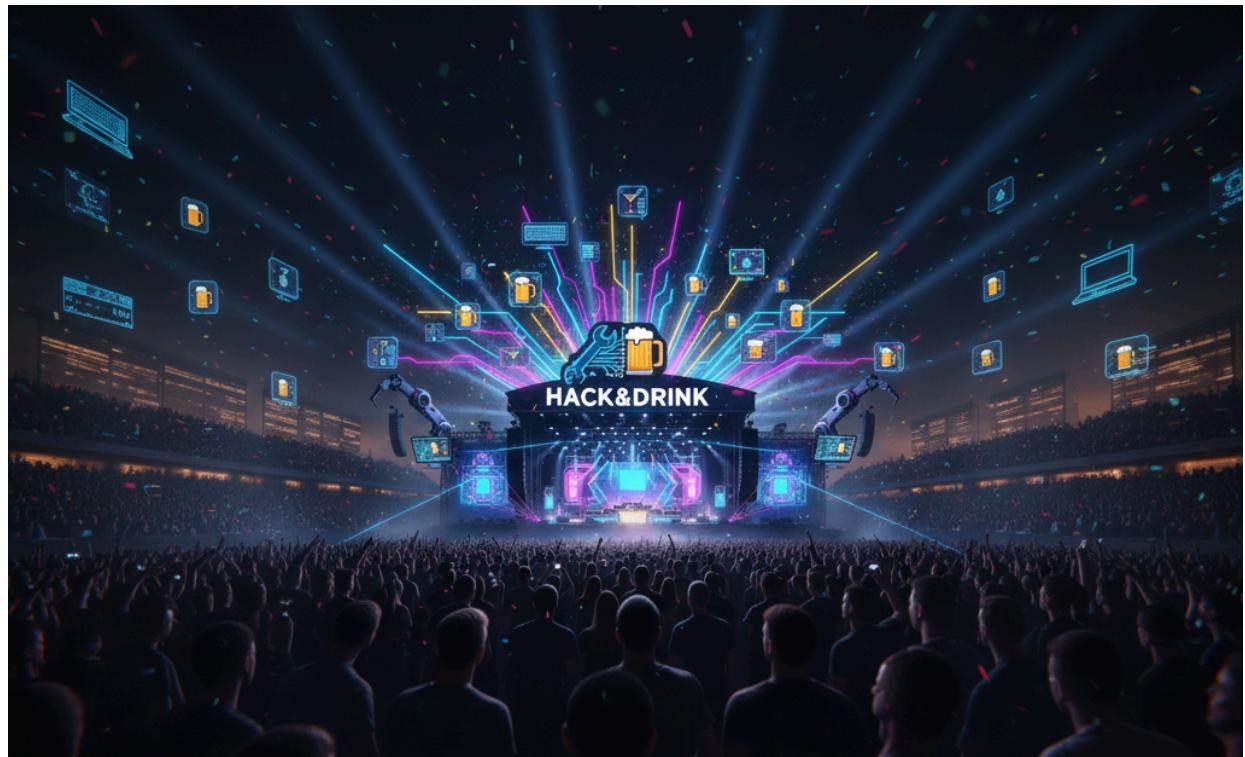


## Paso a Tablas



# **HACK&DRINK**

## *Implementación*



### *Datos del Grupo de Trabajo*

<b>Grupo Pequeño (prácticas y seminarios)</b>	B1
<b>Grupo de trabajo</b>	Hack&Drink
<b>Integrantes</b>	Mario Rasero Vaquero: Patrocinadores Alejandro Ortega Martín: Artistas Manuel Martín Rodríguez: Clientes Roberto Vera Ramos: Servicios Adrian Ros Moya: Infraestructura

# Indice

1. Subsistema Clientes.....	3
1.1 Crear Tablas.....	3
1.2 Transacciones.....	5
1.3 Disparador.....	7
1.4 Aspectos legales.....	7
1.5 Normalización de P2.....	7
2. Subsistema Artistas.....	11
2.1 Crear Tablas.....	11
2.2 Transacciones.....	12
2.3 Aspectos legales.....	13
2.4 Normalización.....	14
3. Subsistema Patrocinadores.....	14
3.1 Crear Tablas.....	14
3.2 Transacciones.....	15
3.3 Aspectos Legales.....	15
3.4 Normalización.....	16
4. Subsistema Servicios.....	17
4.1 Creación de tablas e inserción de tuplas predeterminadas:.....	17
4.2 Transacciones.....	19
4.3 Disparadores.....	20
4.4 Normalización de las tablas.....	21
4.5 Aspectos legales.....	21
5. Subsistema Infraestructura.....	21
5.1 Crear Tablas.....	21
5.2 Transacciones.....	23
5.3 Disparador.....	23
5.4 Normalización.....	24
5.4 Aspectos Legales.....	24

# 1. Subsistema Clientes

## 1.1 Crear Tablas

```
CREATE TABLE Cliente (
    DNI_NIF VARCHAR2(9) PRIMARY KEY,
    CorreoElectronico VARCHAR2(50) UNIQUE NOT NULL,
    Contrasena VARCHAR2(12) NOT NULL,
    Nombre VARCHAR2(40) NOT NULL,
    Apellidos VARCHAR2(60) NOT NULL,
    Telefono Number(20,0)
);
CREATE TABLE Tarjeta (
    NumeroTarjeta Number(16,0) NOT NULL,
    FechaCaducidad DATE NOT NULL,
    CodigoSeguridad Number(4,0) NOT NULL,
    PRIMARY KEY(NumeroTarjeta)
);
CREATE TABLE Entrada (
    DNI_NIF VARCHAR2(9),
    Localizador VARCHAR2(20) NOT NULL,
    NumeroEntradas NUMBER(5,0) NOT NULL,
    TipoEntrada VARCHAR2(50) NOT NULL,
    MetodoPago VARCHAR2(50) NOT NULL,
    NumeroTarjeta Number(16,0) NOT NULL,
    PRIMARY KEY(DNI_NIF, Localizador),
    FOREIGN KEY (DNI_NIF) REFERENCES Cliente(DNI_NIF),
    FOREIGN KEY (NumeroTarjeta) REFERENCES Tarjeta(NumeroTarjeta)
);
CREATE TABLE Evento (
    Nombre VARCHAR2(300) NOT NULL,
    Localizacion VARCHAR2(50) NOT NULL,
    Fecha DATE NOT NULL,
    PRIMARY KEY( Nombre, Localizacion, Fecha)
);
CREATE TABLE Corresponde (
    DNI_NIF VARCHAR2(9),
    Localizador VARCHAR2(20) NOT NULL,
    Nombre VARCHAR2(300) NOT NULL,
    Localizacion VARCHAR2(50) NOT NULL,
    Fecha DATE NOT NULL,
    PRIMARY KEY (DNI_NIF,Localizador),
    FOREIGN KEY (DNI_NIF,Localizador) REFERENCES Entrada(DNI_NIF,Localizador),
    FOREIGN KEY (Nombre,Localizacion, Fecha) REFERENCES Evento(Nombre,Localizacion,
    Fecha)
);
```

---

```

CREATE TABLE Encuesta (
    DNI_NIF VARCHAR2(9),
    RespuestaEncuesta VARCHAR2(200) NOT NULL,
    Nombre VARCHAR2(300) UNIQUE NOT NULL,
    Localizacion VARCHAR2(50) UNIQUE NOT NULL,
    Fecha DATE UNIQUE NOT NULL,
    PRIMARY KEY (DNI_NIF),
    FOREIGN KEY (Nombre,Localizacion, Fecha) REFERENCES Evento(Nombre,Localizacion,
    Fecha)
);
CREATE TABLE Abonosdescuentosseguros promociones_personalizadas(
    Tipo VARCHAR2(12) NOT NULL,
    PRIMARY KEY(Tipo)
);
CREATE TABLE Tiene (
    Tipo VARCHAR2(12) NOT NULL,
    Comprobacion VARCHAR2(300) NOT NULL,
    DNI_NIF VARCHAR2(9) NOT NULL,
    PRIMARY KEY(Tipo,Comprobacion,DNI_NIF),
    FOREIGN KEY (DNI_NIF) REFERENCES Cliente(DNI_NIF),
    FOREIGN KEY (Tipo) REFERENCES
    Abonosdescuentosseguros promociones_personalizadas(Tipo)
);

CREATE TABLE SolicitudAtencionCliente (
    Tipo VARCHAR2(12) NOT NULL,
    PRIMARY KEY(Tipo)
);
CREATE TABLE Enviar (
    Tipo VARCHAR2(12) NOT NULL,
    Comprobacion VARCHAR2(300) NOT NULL,
    detalleSolicitud VARCHAR2(300) NOT NULL,
    DNI_NIF VARCHAR2(9) NOT NULL,
    PRIMARY KEY(Tipo,Comprobacion),
    FOREIGN KEY (DNI_NIF) REFERENCES Cliente(DNI_NIF),
    FOREIGN KEY (Tipo) REFERENCES SolicitudAtencionCliente(Tipo)
);


```

---

## 1.2 Transacciones

Hay que identificar y controlar las siguientes transacciones:

- Alta/Modificación de cliente:
  - La transacción comienza cuando un cliente envía sus datos personales para registrarse o modificar su cuenta.
  - Primero, el sistema realiza un SELECT para verificar que no existe otro cliente con el mismo DNI/NIF o correo electrónico.
  - Si la validación es correcta, se ejecuta un INSERT en la tabla Cliente en el caso que sea Alta de Cliente o un UPDATE si es modificación de Cliente.
  - Si ocurre algún error, se ejecuta ROLLBACK, deshaciendo cualquier cambio.
  - Si todo se realiza correctamente, se ejecuta COMMIT, confirmando la transacción y liberando la tabla Cliente.
- Baja Cliente:
  - La transacción comienza cuando un cliente solicita la baja
  - El sistema realiza un SELECT en la tabla Entrada para comprobar si el cliente tiene entradas pendientes para eventos futuros
  - Si existen entradas pendientes, la transacción no permite la baja y se termina sin cambios.
  - Si no hay entradas pendientes, se ejecuta un DELETE en la tabla Cliente para eliminar los datos del cliente.
  - Si ocurre algún error durante la operación, se ejecuta ROLLBACK y se la operación de darse de baja.
  - Si la eliminación se realiza correctamente, se ejecuta COMMIT, confirmando la baja y liberando la tabla Cliente
- Compra de entrada online:
  - La transacción comienza cuando un cliente selecciona un evento (nombre, localización, Fecha), tipo y número de entradas, y método de pago.
  - El sistema valida la disponibilidad de entradas (SELECT sobre el aforo del evento).
  - Si hay entradas disponibles, se ejecuta un INSERT en la tabla Entrada registrando la compra, incluyendo el localizador generado.
  - Si el método de pago es tarjeta, se ejecuta un INSERT con la información de la tarjeta en la tabla Tarjeta.
  - El sistema procesa el pago.
    - Si el pago falla, se ejecuta ROLLBACK, eliminando los registros de la compra.
    - Si el pago es correcto, se ejecuta COMMIT, confirmando la compra, actualizando el aforo disponible del evento y liberando la tabla Entrada y Tarjeta.
  - Finalmente, se envía el localizador al cliente.
- Gestión de encuestas de satisfacción:
  - La transacción comienza cuando un cliente accede a la encuesta tras asistir al evento.
  - El sistema valida que el cliente tenga una entrada válida para el evento (SELECT en Entrada).
  - También verifica que no haya respondido la encuesta previamente para el mismo evento (SELECT en Encuesta).

- Si las validaciones son correctas, se ejecuta un INSERT en la tabla Encuesta registrando las respuestas del cliente y datos del cliente y evento.
  - Si ocurre algún error o ya se ha completado la encuesta, se ejecuta ROLLBACK, cancelando el proceso de llenar la encuesta.
  - Si todo es correcto, se ejecuta COMMIT, confirmando el registro de la encuesta y liberando la tabla Encuesta y Entrada.
- Gestión de abonos, descuentos, seguros y promociones
  - La transacción comienza cuando un cliente intenta adquirir un abono, descuento, promoción o seguro.
  - El sistema verifica las condiciones necesarias para dicho abono, descuento, promoción o seguro (tipo) con la comprobación que le manda el Administrador del festival y está en la tabla Tiene mediante SELECT en la tabla Tiene.
  - Si el cliente cumple los requisitos, se ejecuta un INSERT en la tabla Tiene, registrando que el cliente ha adquirido el abono, promoción, seguro o descuento.
  - Si el cliente no cumple los requisitos (Ponga en el campo comprobación INVALIDO) o ocurre algún error, se ejecuta ROLLBACK.
  - Si todo se realiza correctamente, se ejecuta COMMIT, confirmando la operación y liberando la tabla Tiene.
- Servicio de atención al cliente
  - La transacción comienza cuando un cliente registra una consulta, reclamación o solicitud de devolución.
  - Se ejecuta un INSERT en la tabla Enviar registrando el tipo de solicitud y DNI del cliente.
  - El sistema asigna automáticamente la solicitud al área de la organización correspondiente (incidencias, devoluciones, reclamaciones).
  - Para devoluciones, se valida que cumplan la política establecida (plazo permitido o evento cancelado) a través de la comprobación que le manda el departamento de devoluciones haciendo un SELECT en la tabla Enviar. Para reclamaciones, se verifica que el cliente haya asistido al evento mediante validación de entrada mediante SELECT en Entrada.
  - Si alguna de estas validaciones falla, se ejecuta ROLLBACK y no se registra la solicitud.
  - Si todo es correcto, se ejecuta COMMIT, confirmando la solicitud y liberando la tabla Enviar.

### 1.3 Disparador

```
CREATE OR REPLACE TRIGGER encuesta_por_evento
BEFORE INSERT ON Encuesta
FOR EACH ROW
DECLARE v_count NUMBER;
BEGIN
SELECT COUNT(*) INTO v_count FROM Encuesta WHERE DNI_NIF = :NEW.DNI_NIF AND
Nombre = :NEW.Nombre AND Localizacion = :NEW.Localizacion AND Fecha = :NEW.Fecha;
IF v_count > 0 THEN
    RAISE_APPLICATION_ERROR(-20002, 'El cliente ya ha respondido la encuesta de este
    evento.');
END IF;
END;
```

### 1.4 Aspectos legales

En el subsistema Clientes, se almacenan y procesan datos personales sensibles, como nombre, apellidos, DNI/NIF, correo electrónico, teléfono, dirección, contraseña, información de la tarjeta del cliente. Por ello, el sistema debe garantizar la seguridad, confidencialidad y uso responsable de esta información, cumpliendo con la normativa vigente, incluyendo la Ley Orgánica de Protección de Datos (LOPD) y el Reglamento General de Protección de Datos (RGPD). Antes de registrar o modificar cualquier dato, el interfaz de usuario debe informar al cliente sobre la finalidad del tratamiento y solicitar su consentimiento explícito, indicando sus derechos de acceso, rectificación, cancelación y oposición. Asimismo, el sistema debe permitir la eliminación de los datos personales cuando el cliente solicite la baja, respetando únicamente la conservación de información necesaria para obligaciones legales o transacciones pendientes. La organización debe designar un responsable del tratamiento de datos y registrar la base de datos ante la Agencia Española de Protección de Datos (AEPD).

### 1.5 Normalización de P2

Tablas:

- Cliente(DNI/NIF, Correo Electronico, Contrasena, Nombre, Apellidos, Teléfono)
- Entrada(DNI/NIF, Localizador, Número Entradas, Tipo Entrada, Método de pago, Número Tarjeta, Fecha Caducidad Tarjeta, Código Seguridad Tarjeta)
- Corresponde(DNI/NIF, Localizador, Nombre, Localización, Fecha)
- Encuesta(DNI/NIF, Respuesta Encuesta, Nombre, Localización, Fecha)
- Evento(Nombre, Localización, Fecha)
- Tiene(Comprobación, Tipo, DNI/NIF)

- Abonos, descuentos, seguros y promociones personalizadas (Tipo)
- SolicitudAtencionCliente(Tipo)
- Enviar(Comprobación,Tipo,DNI/NIF,detallesolicitud)

Dependencias funcionales:

DNI\_NIF → CorreoElectronico, Contrasena, Nombre, Apellidos, Teléfono

CorreoElectronico → DNI\_NIF, Contrasena, Nombre, Apellidos, Teléfono

DNI\_NIF, Localizador → NumeroEntradas, Tipo Entrada, Método de Pago, Número Tarjeta, Fecha Caducidad Tarjeta, Código Seguridad Tarjeta

Número Tarjeta → Fecha Caducidad Tarjeta, Código Seguridad Tarjeta

DNI\_NIF, Localizador → Nombre, Localizacion, Fecha

DNI\_NIF → RespuestaEncuesta,Nombre, Localizacion, Fecha

Nombre, Localizacion, Fecha → DNI\_NIF, RespuestaEncuesta

Comprobacion, Tipo→ DNI\_NIF,detallesolicitud

## 1FN

Todas las tablas están en 1FN: Todos los atributos son atómicos.

## 2FN

Todas las tablas están en 1FN y todos los atributos no primos dependen de forma completa de las claves candidatas:

- Cliente(DNI/NIF, Correo Electronico, Contrasena, Nombre, Apellidos, Teléfono)
  - Contrasena, Nombre, Apellidos, Teléfono dependen de forma completa de DNI/NIF y Correo Electronico
- Entrada(DNI/NIF, Localizador, Número Entradas, Tipo Entrada,Método de pago, Número Tarjeta, Fecha Caducidad Tarjeta, Código Seguridad Tarjeta)
  - Número Entradas, Tipo Entrada,Método de pago, Número Tarjeta, Fecha Caducidad Tarjeta, Código Seguridad Tarjeta dependen de forma completa de DNI/NIF,Localizador. Por ejemplo, no existe una dependencia que sea DNI/NIF → Numero Tarjeta. Que hay si que no se cumpliría la 2FN.

- Corresponde(DNI/NIF,Localizador,Nombre,Localización,Fecha)
  - Nombre, Localizacion, Fecha depende de forma completa de DNI\_NIF, Localizador
- Encuesta(DNI/NIF, Respuesta Encuesta,Nombre,Localización,Fecha)
  - RespuestaEncuesta,Nombre, Localizacion, Fecha depende de forma completa de DNI\_NIF
  - DNI\_NIF, RespuestaEncuesta depende de forma completa de Nombre, Localizacion, Fecha
- Enviar(Comprobación,Tipo,DNI/NIF,detallesolicitud)
  - DNI/NIF es un atributo primo.

Las demás tablas no tienen atributos no primos porque son todos claves candidatas (claves primarias)

### 3FN

Todas están en 3FN ya que están en 2FN y no presentan Dependencia transitiva problemática excepto la tabla Entrada:

- Entrada(DNI/NIF,Localizador, Número Entradas, Tipo Entrada,Método de pago, Número Tarjeta, Fecha Caducidad Tarjeta, Código Seguridad Tarjeta). Dada estas dos dependencias:
  - DNI\_NIF, Localizador → NumeroEntradas, Tipo Entrada, Método de Pago, Número Tarjeta, Fecha Caducidad Tarjeta, Código Seguridad Tarjeta
  - Número Tarjeta → Fecha Caducidad Tarjeta, Código Seguridad Tarjeta

Existe una dependencia cuya parte izquierda (Número Tarjeta) no es una superclave y la parte de la derecha (Fecha Caducidad Tarjeta, Código Seguridad Tarjeta) son atributos no primos.

Aplicando Teorema de Heath, normalizamos:

Entrada (DNI/NIF,Localizador, Número Entradas, Tipo Entrada,Método de pago, Número Tarjeta)

Clave Primaria: DNI/NIF, Localizador

Tarjeta (Número Tarjeta,Fecha Caducidad Tarjeta, Código Seguridad Tarjeta)

Clave Primaria:Número Tarjeta

Quedándonos ya una dependencia directa para ambas tablas.

## FNBC

Todas están en FNBC ya que están en 3FN y todo determinante es una clave candidata.

Determinantes:

- Cliente son DNI/NIF y CorreoElectronico que son claves candidatas
- Entrada es DNI/NIF,Localizador que es una clave candidata.
- Tarjeta es Numero Tarjeta que es una clave candidata
- Corresponde es DNI/NIF,Localizador que es una clave candidata.
- Encuesta son DNI/NIF y Nombre,Localización,Fecha que son claves candidatas
- Evento no hay dependencias funcionales es todo una clave candidata: Nombre,Localización,Fecha
- Tiene no hay dependencias funcionales es todo una clave candidata: Comprobación, Tipo, DNI/NIF
- Abonos, descuentos, seguros y promociones personalizadas no hay dependencias funcionales es todo una clave candidata: Tipo
- SolicitudAtencionCliente no hay dependencias funcionales es todo una clave candidata: Tipo
- Enviar es Comprobación,Tipo que es una clave candidata

## Resultado

- Cliente(DNI/NIF, Correo Electronico, Contrasena, Nombre, Apellidos, Teléfono)
- Entrada(DNI/NIF, Localizador, Número Entradas, Tipo Entrada,Método de pago, Número Tarjeta)
- Tarjeta (Número Tarjeta,Fecha Caducidad Tarjeta, Código Seguridad Tarjeta)
- Corresponde(DNI/NIF,Localizador,Nombre,Localización,Fecha)
- Encuesta(DNI/NIF, Respuesta Encuesta,Nombre,Localización,Fecha)
- Evento(Nombre,Localización,Fecha)
- Tiene(Comprobación, Tipo, DNI/NIF )

- Abonos, descuentos, seguros y promociones personalizadas (Tipo)
- SolicitudAtencionCliente(Tipo)
- Enviar(Comprobación,Tipo,DNI/NIF,detallesolicitud)

## 2. Subsistema Artistas

### 2.1 Crear Tablas

```
CREATE TABLE ARTISTAS (
    DNI_NIF VARCHAR2(9),
    CORREO_ELECTRONICO VARCHAR2(100),
    NOMBRE VARCHAR2(50),
    APELLIDOS VARCHAR2(100),
    CACHE NUMBER(10, 2),
    TELEFONO VARCHAR2(15),
    DIRECCION VARCHAR2(200),
    GENERO_MUSICAL VARCHAR2(50),
    CONSTRAINT PK_ARTISTAS PRIMARY KEY (DNI_NIF)
);
```

```
CREATE TABLE Contratos (
    ID NUMBER(3) PRIMARY KEY,
    ImporteAcordado NUMBER(10, 2),
    FechaInicio DATE,
    FechaFin DATE,
    MetodoDePago VARCHAR2(20),
    TipoDePago VARCHAR2(15),
    DuracionDelFraccionado NUMBER(2) DEFAULT 1,
    DatosBancarios VARCHAR2(34)
);
```

```
CREATE TABLE Camerinos (
    Identificacion VARCHAR2(2) PRIMARY KEY
);
```

---

```
CREATE TABLE Escenarios (
    NumeroEscenario NUMBER(2) PRIMARY KEY
);
```

```
CREATE TABLE Firma (
    ID NUMBER(3),
    DNI_NIF VARCHAR2(9),
    PRIMARY KEY (ID, DNI_NIF),
    FOREIGN KEY (ID) REFERENCES Contratos(ID),
    FOREIGN KEY (DNI_NIF) REFERENCES Artistas(DNI_NIF)
);
```

```
CREATE TABLE Pagos (
    ID NUMBER(3),
    NumPago NUMBER(2),
    PRIMARY KEY (ID, NumPago),
    FOREIGN KEY (ID) REFERENCES Contratos(ID)
);
```

```
CREATE TABLE Ocupa (
    DNI_NIF VARCHAR2(9),
    HoraInicio DATE,
    HoraFin DATE,
    Identificacion VARCHAR2(2),
    PRIMARY KEY (DNI_NIF, HoraInicio, HoraFin, Identificacion),
    FOREIGN KEY (DNI_NIF) REFERENCES Artistas(DNI_NIF),
    FOREIGN KEY (Identificacion) REFERENCES Camerinos(Identificacion)
);
```

```
CREATE TABLE Asociado (
    DNI_NIF VARCHAR2(9),
    HoraInicio2 DATE,
    HoraFin2 DATE,
    NumeroEscenario NUMBER(2),
    PRIMARY KEY (DNI_NIF, HoraInicio2, HoraFin2, NumeroEscenario),
    FOREIGN KEY (DNI_NIF) REFERENCES Artistas(DNI_NIF),
    FOREIGN KEY (NumeroEscenario) REFERENCES Escenarios(NumeroEscenario)
);
```

## 2.2 Transacciones

La gestión transaccional se articula de la siguiente manera

---

## **1. Desactivación del Autocommit**

Por defecto, JDBC opera en modo autocommit, ejecutando cada sentencia SQL de forma independiente.

```
conn.setAutoCommit(false);
```

Esto marca el inicio de una unidad de trabajo lógica. Hasta que no se dé una orden explícita, los cambios permanecen en un estado temporal.

## **B. Confirmación de Cambios**

Solo cuando todas las validaciones lógicas y las inserciones/actualizaciones se han completado con éxito, se invoca:

```
conn.commit();
```

Esto asegura que la información se grabe de forma permanente y sea visible para otros usuarios.

## **C. Reversión en caso de error (Rollback)**

Para evitar la corrupción de datos o registros, he implementado un bloque catch que captura cualquier SQLException. Si una sola operación falla, se ejecuta:

```
conn.rollback();
```

Esto devuelve la base de datos a su estado original antes de iniciar la transacción.

## **2.3 Aspectos legales**

El subsistema de Gestión de Artistas procesa datos sensibles: DNI/NIF, contacto, datos bancarios, cachés y contratos. El sistema garantiza la seguridad y confidencialidad de esta información.

Al procesar una baja el sistema elimina los datos personales del artista. No obstante, se mantendrá bloqueada la información necesaria para cumplir con las obligaciones tributarias y legales vigentes.

La organización designará un responsable del tratamiento y registrará la actividad ante la AEPD, asegurando que la base de datos sea íntegra y segura.

---

## 2.4 Normalización

Estaban todas en 3FN/FNBC menos Pagos y contratos que se explican en el punto 3.4

## 3. Subsistema Patrocinadores

### 3.1 Crear Tablas

-- Tabla PERSONA\_CONTACTO

```
CREATE TABLE PERSONA_CONTACTO (
    DNI VARCHAR2(9) PRIMARY KEY,
    NombrePersonaContacto VARCHAR2(50) NOT NULL,
    CorreoElectronicoPersonaContacto VARCHAR2(50),
    NumeroTelefono VARCHAR2(20)
);
```

-- Tabla PATROCINADORES

```
CREATE TABLE PATROCINADORES (
    NombreEmpresa VARCHAR2(30) PRIMARY KEY,
    DNI VARCHAR2(9) NOT NULL,
    TipoBeneficios VARCHAR2(100),
    DescripcionBeneficios VARCHAR2(500),
    CONSTRAINT FK_Patrocinador_Contacto FOREIGN KEY (DNI) REFERENCES
    PERSONA_CONTACTO(DNI)
);
```

-- Tabla CONTRATOS

```
CREATE TABLE CONTRATOS (
    ID NUMBER PRIMARY KEY,
    ImporteAcordado NUMBER(10,2) NOT NULL,
    FechaInicio DATE NOT NULL,
    FechaFin DATE NOT NULL,
    DatosBancarios VARCHAR2(24),
    MetodoDePago VARCHAR2(20),
    TipoDePago VARCHAR2(15),
    DuracionDelFraccionado NUMBER(2)
);
```

-- Tabla PAGOS

```
CREATE TABLE PAGOS (
    ID NUMBER NOT NULL,
    NumPago NUMBER NOT NULL,
    PRIMARY KEY (ID, NumPago),
    CONSTRAINT FK_Pagos_Contratos FOREIGN KEY (ID) REFERENCES CONTRATOS(ID)
);
```

---

```
-- Tabla FIRMA
CREATE TABLE FIRMA_P(
    ID NUMBER NOT NULL,
    NombreEmpresa VARCHAR2(30) NOT NULL,
    PRIMARY KEY (ID, NombreEmpresa),
    CONSTRAINT FK_Firma_Contratos FOREIGN KEY (ID) REFERENCES CONTRATOS(ID),
    CONSTRAINT FK_Firma_Patrocinadores FOREIGN KEY (NombreEmpresa) REFERENCES
    PATROCINADORES(NombreEmpresa)
);
```

## 3.2 Transacciones

Transacción en registrarPatrocinador()

Operaciones:

    INSERT en PERSONA\_CONTACTO

    INSERT en PATROCINADORES

Gestión: conn.setAutoCommit(false) + conn.commit()

Rollback: conn.rollback() en caso de error

Transacción en listarPatrocinadores()

Operación: Consulta JOIN entre tablas

Gestión: conn.setAutoCommit(false) + conn.commit()

Rollback: No implementado

Transacción en registrarContrato()

Operaciones:

    INSERT en CONTRATOS

    INSERT en FIRMA\_P

Gestión: conn.setAutoCommit(false) + Savepoint + conn.commit()

Rollback: conn.rollback(savepoint) en caso de error

Transacción en actualizarBeneficios()

Operación: UPDATE en PATROCINADORES

Gestión: conn.setAutoCommit(false) + conn.commit() o conn.rollback()

Rollback: Condicional (si no hay filas afectadas o error)

Transacción en registrarPago()

Operación: INSERT en PAGOS

Gestión: conn.setAutoCommit(false) + conn.commit()

Rollback: conn.rollback() en caso de error

## 3.3 Aspectos Legales

El sistema Hack-DrinkManager trata datos personales y financieros sensibles bajo el marco del RGPD y LOPDGDD. Se han implementado medidas básicas de seguridad (PreparedStatement, transacciones, validación de IBAN e información al usuario), pero faltan aspectos críticos como cifrado de datos bancarios, autenticación, y endpoints para ejercer derechos ARCO-PL. El sistema requiere refuerzos de seguridad antes de su uso en producción con datos reales.

### 3.4 Normalización

#### TABLAS QUE REQUERÍAN NORMALIZACIÓN

1) PAGOS (estaba en 1FN, no cumplía 2FN)

ANTES:

##### PAGOS

- ID (FK)
- NumPago
- PK(ID, NumPago)
- MetodoDePago
- TipoDePago
- DuracionDelFraccionado

DESPUÉS:

##### CONTRATOS (se añaden atributos)

- ID (PK)
- ImporteAcordado
- FechaInicio
- FechaFin
- DatosBancarios
- MetodoDePago
- TipoDePago
- DuracionDelFraccionado

##### PAGOS (solo quedan atributos del pago individual)

- ID (FK → CONTRATOS.ID)
- NumPago
- PK(ID, NumPago)

POR QUÉ: Había dependencias parciales  $ID \rightarrow \{MetodoDePago, TipoDePago, DuracionDelFraccionado\}$ . Como la PK es compuesta (ID, NumPago), esos atributos dependían solo de parte de la clave, violando 2FN. Se mueven a CONTRATOS porque son propiedades del contrato, no del pago individual.

2) PATROCINADORES (estaba en 2FN, no cumplía 3FN)

ANTES:

##### PATROCINADORES

- NombreEmpresa (PK)
- CorreoElectronicoPersonaContacto
- DNIPersonaContacto
- NumeroTelefono
- NombrePersonaContacto
- TipoBeneficios
- DescripcionBeneficios

DESPUÉS:

##### PERSONA\_CONTACTO (nueva tabla)

---

- DNI (PK)
- NombrePersonaContacto
- CorreoElectronicoPersonaContacto
- NumeroTelefono

#### PATROCINADORES

- NombreEmpresa (PK)
- DNI (FK → PERSONA\_CONTACTO.DNI)
- TipoBeneficios
- DescripcionBeneficios

Había dependencia transitiva: NombreEmpresa → DNI → {Nombre, Correo, Teléfono}. El DNI identifica únicamente a la persona de contacto y determina sus datos. Separar esta entidad elimina la dependencia transitiva y lleva PATROCINADORES a 3FN.

#### TABLAS QUE YA ESTABAN EN 3FN/BCNF

##### CONTRATOS

Ya estaba en 3FN/BCNF (todos los atributos dependen únicamente de ID).

##### FIRMA

Ya estaba en BCNF (tabla de relación sin atributos no-clave)

## 4. Subsistema Servicios

### 4.1 Creación de tablas e inserción de tuplas predeterminadas:

```
CREATE TABLE Servicios(
    Nombre VARCHAR2(20) PRIMARY KEY
);
```

```
CREATE TABLE Proveedores(
    NIF VARCHAR2(9) PRIMARY KEY,
    Nombre VARCHAR2(40),
    Correo VARCHAR2(40) UNIQUE,
    Telefono VARCHAR2(9)
);
```

```
CREATE TABLE Empleados(
    DNI VARCHAR2(9) PRIMARY KEY,
    Nombre VARCHAR2(29),
    Apellidos VARCHAR2(40),
    Telefono VARCHAR2(9)
);
```

---

```
CREATE TABLE Asignacion(  
    Nombre REFERENCES Servicios(Nombre),  
    Area NUMBER,  
    NIF REFERENCES Proveedores(NIF),  
    NumeroEmp NUMBER,  
    PRIMARY KEY (Nombre, Area)  
);
```

```
CREATE TABLE Trabajo(  
    DNI VARCHAR2(9) REFERENCES Empleados(DNI) PRIMARY KEY,  
    Nombre VARCHAR2(20),  
    Area NUMBER,  
    FOREIGN KEY (Nombre,Area) REFERENCES Asignacion(Nombre,Area)  
);
```

```
CREATE TABLE Turnos(  
    DNI VARCHAR2(9) REFERENCES Empleados(DNI),  
    FechaInicio DATE,  
    FechaSalida DATE,  
    Area NUMBER,  
    PRIMARY KEY (DNI, FechaInicio)  
);
```

```
CREATE TABLE Suministros (  
    Identificador VARCHAR2(4),  
    Area NUMBER,  
    Nivel NUMBER,  
    PRIMARY KEY (Identificador, Area)  
);
```

```
CREATE TABLE INCIDENCIA(  
    DNI VARCHAR2(9) REFERENCES Cliente(DNI_NIF),  
    Identificador VARCHAR2(9),  
    Descripcion VARCHAR2(200),  
    PRIMARY KEY (DNI, Identificador, Descripcion)  
);
```

---

```

CREATE TABLE VALORACION (
    DNI VARCHAR2(9),
    Localizador VARCHAR2(9),
    Identificador VARCHAR2(9),
    Comentario VARCHAR2(200),
    Valoracion NUMBER,
    FOREIGN KEY (DNI, Localizador) REFERENCES Entrada(DNI_NIF, Localizador),
    PRIMARY KEY (DNI, Localizador, Identificador)
);

INSERT INTO PROVEEDORES VALUES('12345678A', 'bebidas_manu', 'correo1@gmail.com',
'123456789');
INSERT INTO PROVEEDORES VALUES('12345678B', 'aguas_guadix', 'correo2@gmail.com',
'223456789');
INSERT INTO PROVEEDORES VALUES('12345678C', 'prosegur', 'correo3@gmail.com',
'323456789');
INSERT INTO PROVEEDORES VALUES('12345678D', 'aidalite', 'correo4@gmail.com',
'423456789');

INSERT INTO Servicios VALUES ('AGUA');
INSERT INTO Servicios VALUES ('REFRESCOS');
INSERT INTO Servicios VALUES ('SEGURIDAD');
INSERT INTO Servicios VALUES ('LIMPIEZA');

INSERT INTO Suministros VALUES('0000', 2, 13);
INSERT INTO Suministros VALUES('0001', 5, 100);
INSERT INTO Suministros VALUES('0002', 1, 122);
INSERT INTO Suministros VALUES('000A', 3, 4);

COMMIT

```

## 4.2 Transacciones

Respecto a las transacciones llevadas a cabo, voy a explicar cada una de las transacciones llevadas a cabo en cada una de las operaciones que he creado en el controller.java:

Lo primero que quiero aclarar es que para llevar a cabo las transacciones de forma manual en cada uno de los métodos post en los que se ha modificado de alguna forma la BD se ha desactivado lo primero de todo el autocommit, y haremos uso de commit para confirmar cambio y rollback para deshacer toda operaciones realizada sin confirmar hasta la última confirmación (no he usado

---

savepoint porque no había puntos medios, una vez introduce os datos se hace la operación completa, no hay nada que se pueda salvar o reutilizar si la operación fallara)

Asignación de Proveedores y Empleados (asignarProveedores): Inserta una relación en la tabla Asignacion y posteriormente inserta una lista de empleados en las tablas Empleados y Trabajo. Si falla la inserción de un solo empleado (por duplicado) o si la empresa no existe, se debe deshacer (rollback) toda la operación, incluyendo la asignación inicial para evitar dejar la

Creación de Turnos (asignarTurnos): Verifica solapamientos de fechas y, si es correcto y no hay solapamiento, inserta un nuevo turno en la tabla TURNOS. Antes de hacer commit, comprueba mediante un parámetro de salida configurado usando un ? de salida en prepared statement, que el empleado no tenga otro turno en ese mismo rango horario de forma que si esto ocurre cancelará la operación hará un rollback y saltará un error .

Reabastecimiento de Suministros (reabastecerSuministros): Actualiza con un UPDATE el nivel de stock de un suministro y devuelve el nuevo total. Si el UPDATE devuelve 0 filas afectadas (el suministro no existe), se hace rollback y se finaliza la operación devolviendo un error, si esta todo bien entonces hace commit.

Reporte de Incidencias (reportarIncidencia): Verifica la existencia un cliente y crea un registro en INCIDENCIA. Se comprueba primero si el DNI existe en la tabla CLIENTE. Si no existe, se cancela la operación y se hace un rollback para dejarlo todo como estaba en el anterior commit, para evitar violaciones de claves foráneas. Si todo va bien hace un commit y confirma la incidencia creada.

Envío de Valoraciones (procesarValoracion): Verifica la existencia de una entrada válida y crea un registro en VALORACION. Solo se permite valorar si existe una fila en la tabla ENTRADA que coincida con el DNI y el Localizador, asegurando que solo clientes reales con entrada pueden valorar. Si no se cumple hará un rollback y saltará un error. Si todo se cumple entonces almacenará la valoración y hará un commit.

### 4.3 Disparadores

No he implementado un trigger típico de Oracle SQL porque no he visto ninguna zona en la que fuera en extremo útil. Sin embargo sí que he llevado a cabo comprobaciones usando PL/SQL en las que si un no se cumplía una determinada condición para el parámetro introducido entonces no realizara un INSERT o un CREATE... Ejemplo:

DECLARE

```
inicio DATE;
salida DATE;
v_dni VARCHAR2(9);
area NUMBER;
contador NUMBER;
```

```
BEGIN
```

```
inicio := ?;
```

```

salida := ?;
v_dni := ?;
area := ?;

SELECT COUNT(*) INTO contador FROM TURNOS WHERE DNI = v_dni AND (FechaInicio < salida AND FechaSalida
> inicio);

? := contador;

IF contador = 0 THEN

    INSERT INTO TURNOS VALUES (v_dni, inicio, salida, area);

END IF;

END;

```

Donde luego uso registerOutParameter para sacar el valor de la variable contador fuera del programa y así determinar si hay error o no.

#### 4.4 Normalización de las tablas

En cuanto a la normalización de mis tablas, estás ya fueron construidas directamente en 3FN y BCNF (si está en BCNF ya está en 3F). Con lo que no tenido que llevar a cabo ninguna conversión nueva.

#### 4.5 Aspectos legales

En las operaciones implementadas del subsistema de servicios no se contempla el registro de datos sensibles. Es cierto que se usan, como podría ser el DNI de la persona... Pero estos ya están registrados en el sistema, se determina por tanto que han aceptado los acuerdos de seguridad establecidos por los sistemas anteriores para poder registrarse y que no se está vulnerando la privacidad de ninguna forma. Además no compartimos la información sensible para fines que no hayan sido previamente consultados con el cliente.

### 5. Subsistema Infraestructura

#### 5.1 Crear Tablas

---

```

CREATE TABLE Mapa(
    Id NUMBER(3) PRIMARY KEY,
    Nombre VARCHAR2(20),

```

```
Detalles VARCHAR2(256),  
Posicion NUMBER(3)  
);
```

```
CREATE TABLE Permisos(  
IdPermiso NUMBER(3) PRIMARY KEY,  
Descripcion VARCHAR2(256),  
Tipo VARCHAR2(64)  
);
```

```
CREATE TABLE Festival(  
IdFestival NUMBER(3) PRIMARY KEY,  
AforoActual NUMBER(10),  
AforoMaximo NUMBER(10)  
);
```

```
CREATE TABLE FestivalTienePermisos(  
IdFestival NUMBER(3) REFERENCES Festival(IdFestival),  
IdPermiso NUMBER(3) REFERENCES Permisos(IdPermiso) PRIMARY KEY  
);
```

```
CREATE TABLE PersonalDelFestival(  
IdPersonal NUMBER(3) PRIMARY KEY,  
Nombre VARCHAR2(32),  
Correo VARCHAR2(50)  
);
```

```
CREATE TABLE Averias(  
IdAveria NUMBER(4) PRIMARY KEY,  
Detalles VARCHAR2(256)  
);
```

```
CREATE TABLE PersonalNotificaAveria(  
IdPersonal NUMBER(3) REFERENCES PersonalDelFestival(IdPersonal),  
IdAveria NUMBER(4) REFERENCES Averias(IdAveria) PRIMARY KEY  
);
```

Debo hacer mención a la modificación de las tablas PersonalNotificaAveria y FestivaltienePermisos. Ambas heredaban todos los atributos de PersonalDelFestival, Averias, Festival y Permisos respectivamente (con foreign key). Sin embargo, los atributos no únicos no se pueden heredar con foreign key, por lo que he tenido que borrar todos los atributos no únicos, esto no ha producido ningún error.

---

## 5.2 Transacciones

He decidido implementar y controlar las siguientes transacciones:

1. Obtener los datos de Mapa y Permisos. En mi base de datos, se guarda la siguiente información sobre cada festival: el Mapa (El mapa correspondiente a cada festival) y los Permisos.

Para gestionar dicha transacción se realizan las siguientes operaciones:

```
SELECT * FROM MAPA WHERE Id = ?  
SELECT * FROM PERMISOS WHERE IdPermiso = ?
```

Antes de ejecutarlas he desactivado el autocommit, luego he hecho un commit.  
Para control de errores he manejado excepciones con try.

2. Obtener los datos del Personal. Para manejar esta transacción he ejecutado la siguiente sentencia:

```
SELECT * FROM PERSONALDELFESTIVAL WHERE IdPersonal = ?
```

También he manejado errores con try.

3. Insertar Avería. Para manejar esta transacción he ejecutado las siguientes sentencias:

```
SELECT COUNT(*) FROM AVERIAS  
INSERT INTO Averias VALUES (?,?)
```

Donde obtengo el número de averías que hay, le sumo uno e inserto en Averías el idAvería (la suma que he hecho), además del texto obtenido de una TextArea del html.  
Además he implementado el Trigger requerido sobre esta transacción, a continuación describo su funcionamiento.

## 5.3 Disparador

```
CREATE OR REPLACE TRIGGER validar_texto_averia  
BEFORE INSERT ON AVERIAS  
FOR EACH ROW  
BEGIN  
    -- Verificar si el texto (segundo parámetro) es NULL  
    IF :NEW.detalles IS NULL OR TRIM(:NEW.detalles) = " THEN  
        -- Detenemos la operación y lanzamos un error si es NULL o vacío  
        RAISE_APPLICATION_ERROR(-20001, 'El campo "texto" no puede ser nulo o estar vacío.');//  
    END IF;  
END;
```

---

El disparador implementado verifica que el texto de la avería pasado desde la página web no sea nulo, se ejecuta antes de la sentencia y si es nulo lanza un error a la web.

## 5.4 Normalización

Contando con PersonalNotificaAveria y FestivalTienePermisos ya modificados como se ha comentado anteriormente, todas las tablas implementadas ya se encuentran en tercera forma normal, no he tenido que hacer ninguna modificación aparte.

## 5.4 Aspectos Legales

El sistema maneja datos privados, por lo que es necesario tener en cuenta la legalidad en lo referente a el uso de los datos. El sistema ya ha implementado algunos de los aspectos legales necesarios, sin embargo, aún necesita un refuerzo en esta área antes de poder usarlo con datos reales.

### **Motivación del software seleccionado**

Nos hemos decantado con usar el framework de spring-boot para la práctica, este es un software gratuito con el que se pueden montar proyectos web muy conocido, su versión para páginas web te permite gestionar un modelo vista controlador.

Los html los hemos implementado con RocketCake, que es una herramienta especializada para este propósito, en la que puedes generar de manera interactiva una página web a tu gusto y obtener el html y css correspondiente (puede usarse en Mac y Windows).

Con RocketCake hemos podido construir páginas html con css sin tener mucha idea de los lenguajes.