



Solución:

El objetivo de este análisis fue diseñar un modelo de base de datos que permita representar de manera eficiente y clara las relaciones entre las entidades principales identificadas en el documento del proyecto.

1. Análisis inicial del documento

Primero, analicé el documento del reto para extraer las ideas clave y los requisitos específicos del sistema. Como estrategia personal, utilicé un enfoque basado en la identificación de sustantivos dentro del texto. Esto me permitió delimitar las entidades más relevantes que serían fundamentales para el diseño de la base de datos.

Entre los sustantivos identificados, prioricé aquellos que:

- Se repetían frecuentemente en el texto.
- Tenían mayor relevancia para los objetivos funcionales del proyecto.

2. Identificación de entidades

Con los sustantivos seleccionados, definí las principales entidades que forman el núcleo de la base de datos. Estas entidades son:

- **Cliente:** Representa a los usuarios que adquieren los servicios.
- **Tiquete:** Registra la compra realizada por los clientes.
- **Atracción:** Representa las actividades o servicios ofrecidos.
- **Clasificación de Atracciones:** Agrupa y categoriza las atracciones.
- **Empleado:** Personas encargadas de las diferentes estaciones.
- **Estación:** Punto físico relacionado con una atracción específica.
- **Roles:** Definición de las funciones que desempeñan los empleados.
- **Tipos de Tiquete:** Clasificación de los tiquetes para definir sus características.

3. Diseño del modelo de relaciones

Basándome en las entidades identificadas, elaboré un diagrama entidad-relación (ER) para definir cómo se conectan entre sí. Algunos aspectos clave del diseño incluyen:

- **Relaciones principales:**
 - Los clientes **compran** tiquetes (relación 1:N).
 - Los tiquetes están **asociados** con atracciones mediante una tabla intermedia (**tiquetes_atracciones**) para evitar redundancias y problemas futuros (relación N:M).
 - Cada atracción está **clasificada** bajo una categoría específica (relación N:1).
 - Las estaciones están **asociadas** con los tiquetes adquiridos (relación N:1).
 - Los empleados están **asignados** a estaciones (relación 1:N), y su rol se gestiona a través de una tabla de roles.

4. Normalización

Implementé técnicas de normalización para mejorar la eficiencia y consistencia de la base de datos:

- Creé tablas auxiliares como **tipo_tiquete**, **roles**, y **clasificación_atracciones**, que permiten almacenar información categórica de manera separada, evitando redundancia y facilitando futuros cambios en la estructura de datos.
- La tabla intermedia **tiquetes_atracciones** evita una relación directa N:M entre tiquetes y atracciones, asegurando mayor flexibilidad y escalabilidad en el modelo.

5. Detalles adicionales:

El diseño visual se inspiró en la estética de Salitre Mágico, utilizando su paleta de colores para crear un entorno amigable y atractivo. Organicé los módulos del sistema de manera intuitiva para que los empleados del parque pudieran navegar fácilmente entre las funciones de gestión. Adicionalmente en el desarrollo del sistema implementé esto:

1. **Patrón de Diseño MVC:** Estructuré el sistema en Modelos, Vistas y Controladores para garantizar una organización clara y separación de responsabilidades. Esto mejora el mantenimiento y la ampliación futura del proyecto.
2. **Gestión de Excepciones:** Implementé un manejo robusto de excepciones para evitar caídas inesperadas del sistema, asegurando una experiencia fluida para el usuario.
3. **Uso de DAO y DTO:** Empleé estos patrones para separar la lógica de acceso a datos de la lógica de negocio, optimizando la transferencia de datos entre las capas del sistema.
4. **Base de Datos Relacional:** Elegí MySQL como gestor de base de datos, complementado con phpMyAdmin para monitorear las consultas. MySQL fue seleccionado por mi familiaridad y competencia con bases de datos relacionales.
5. **Conexión Configurable:** Utilicé un archivo `.properties` para centralizar las configuraciones de conexión con la base de datos, permitiendo ajustes rápidos sin modificar el código fuente.
6. **Interfaces Gráficas con Swing y AWT:** Decidí usar estas herramientas por su integración natural con POO en Java, demostrando herencia, abstracción y polimorfismo en su implementación.

En la siguiente pagina voy a mostrar el pdf generado por php my admin para mostrar mas a detalle la base de datos y sus atributos

Tabla de contenidos

1 atracciones	Número de página: 2
2 clasificaciones_atracciones	Número de página: 3
3 clientes	Número de página: 4
4 empleados	Número de página: 5
5 estaciones	Número de página: 6
6 roles_empleados	Número de página: 7
7 tipo_tiquete	Número de página: 8
8 tiquetes	Número de página: 9
9 tiquetes_atracciones	Número de página: 10
10 Esquema relacionado	Número de página: 11

1 atracciones

Creación: 03-01-2025 a las 11:54:17
Última actualización: 03-01-2025 a las 18:12:39

Columna	Tipo	Atributos	Null	Predefinido	Extra	Enlaces a	Comentarios	MIME
id	int(11)		No		auto_increment			
nombre	varchar(100)		No					
descripcion	text		No					
clasificacion_id	int(11)		No			-> clasificaciones_atracciones.id ON UPDATE RESTRICT ON DELETE RESTRICT		
estatura_minima	decimal(5,2)		Sí	NULL				
condiciones_uso	text		Sí	NULL				
estado	enum('Disponible', 'No Disponible', 'Mantenimiento')		No					
empleado_id	int(11)		No			-> empleados.id ON UPDATE RESTRICT ON DELETE RESTRICT		

2 clasificaciones_atracciones

Creación: 27-12-2024 a las 18:47:50
Última actualización: 31-12-2024 a las 21:06:54

Columna	Tipo	Atributos	Null	Predefinido	Extra	Enlaces a	Comentarios	MIME
id	int(11)		No		auto_increment			
nombre	varchar(50)		No					

3 clientes

Creación: 01-01-2025 a las 19:19:04
Última actualización: 03-01-2025 a las 23:09:47

Columna	Tipo	Atributos	Null	Predefinido	Extra	Enlaces a	Comentarios	MIME
id	int(11)		No		auto_increment			
nombre	varchar(100)		No					
N_Identificacion	varchar(20)		No					
telefono	varchar(15)		Sí	NULL				
correo	varchar(100)		No					
estatura	decimal(5,2)		No					
edad	int(11)		No					
nombre_familiar	varchar(100)		Sí	NULL				
telefono_familiar	varchar(15)		Sí	NULL				
visitas	int(11)		Sí	0				

4 empleados

Creación: 01-01-2025 a las 19:19:04
Última actualización: 01-01-2025 a las 20:20:42

Columna	Tipo	Atributos	Null	Predefinido	Extra	Enlaces a	Comentarios	MIME
id	int(11)		No		auto_increment			
nombre	varchar(100)		No					
cedula	varchar(20)		No					
telefono	varchar(15)		Sí	NULL				
correo	varchar(100)		No					
rol_id	int(11)		No			-> roles_empleados.id ON UPDATE RESTRICT ON DELETE RESTRICT		
horario_laboral	varchar(50)		No					

5 estaciones

Creación: 27-12-2024 a las 18:47:50
Última actualización: 01-01-2025 a las 23:10:36

Columna	Tipo	Atributos	Nul	Predefinido	Extra	Enlaces a	Comentarios	MIME
id	int(11)		No		auto_increment			
nombre	varchar(100)		No					
habilitada	tinyint(1)		Sí	1				
empleado_id	int(11)		No			-> empleados.id ON UPDATE RESTRICT ON DELETE RESTRICT		

6 roles_empleados

Creación: 27-12-2024 a las 18:47:50
Última actualización: 31-12-2024 a las 21:04:01

Columna	Tipo	Atributos	Null	Predefinido	Extra	Enlaces a	Comentarios	MIME
id	int(11)		No		auto_increment			
nombre	varchar(50)		No					

7 tipo_tiquete

Creación: 31-12-2024 a las 20:56:48
Última actualización: 31-12-2024 a las 20:59:59

Columna	Tipo	Atributos	Null	Predefinido	Extra	Enlaces a	Comentarios	MIME
id	int(11)		No		auto_increment			
nombre	varchar(50)		No					
descripcion	text		Sí	NULL				

8 tiquetes

Creación: 31-12-2024 a las 20:36:09
Última actualización: 03-01-2025 a las 23:09:47

Columna	Tipo	Atributos	Nul	Predefinido	Extra	Enlaces a	Comentarios	MIME
id	int(11)		No		auto_increment			
cliente_id	int(11)		No			-> clientes.id ON UPDATE RESTRICT ON DELETE RESTRICT		
estacion_id	int(11)		No			-> estaciones.id ON UPDATE RESTRICT ON DELETE RESTRICT		
fecha	date		No					
tipo_tiquete_id	int(11)		No			-> tipo_tiquete.id ON UPDATE RESTRICT ON DELETE RESTRICT		

9 tiquetes_atracciones

Creación: 31-12-2024 a las 20:50:53
Última actualización: 03-01-2025 a las 23:18:46

Columna	Tipo	Atributos	Null	Predefinido	Extra	Enlaces a	Comentarios	MIME
id	int(11)		No		auto_increment			
tickete_id	int(11)		No			-> tiquetes.id ON UPDATE RESTRICT ON DELETE RESTRICT		
atraccion_id	int(11)		No			-> atracciones.id ON UPDATE RESTRICT ON DELETE RESTRICT		

