**INFORME DESARROLLO DE BASE DE DATOS**

**Autor:**

Javier Nicolas Salas

**Entregado a**:

Ing. Brayan Arcos

**Fecha:** 11/09/2024

**Tabla de contenido**

[**INFORME DESARROLLO DE BASE DE DATOS** 1](#_Toc176939829)

[**Repositorio De GitHub:** 2](#_Toc176939830)

[**Resumen Ejecutivo** 2](#_Toc176939831)

[**Introducción** 3](#_Toc176939832)

[**Contexto y Motivación** 3](#_Toc176939833)

[**Alcance del Informe** 3](#_Toc176939834)

[**Objetivos** 3](#_Toc176939835)

[**Metodología** 3](#_Toc176939836)

[**Herramientas Utilizadas** 3](#_Toc176939837)

[**Procedimientos** 3](#_Toc176939838)

[**Desarrollo del Informe** 4](#_Toc176939839)

[**Descripción de la Base de Datos** 4](#_Toc176939840)

[**Consultas SQL y Resultado** 6](#_Toc176939841)

[**Diseño de Base de Datos** 10](#_Toc176939842)

[**Análisis y Discusión** 19](#_Toc176939843)

[**Interpretación de Resultados** 19](#_Toc176939844)

[**Conclusiones** 19](#_Toc176939845)

[**Recomendaciones** 20](#_Toc176939846)

[**Referencias** 20](#_Toc176939847)

**Repositorio De GitHub:**

Repositorio documentado con su estructura https://github.com/Nick0oo/DBD\_Proyecto

**Resumen Ejecutivo**

Este informe proporciona una visión detallada del proceso de desarrollo de una base de datos para el manejo total de una cancha complejo deportivo. Se aborda el diseño de la base de datos, la estructura de las tablas y las consultas realizadas, proceso de entidades fuertes y débiles, modelo entidad relación, normalización etc. El informe ofrece una explicación lógica de todo el proceso realizado.

**Introducción**

**Contexto y Motivación**

El principal motivador es el avance en el desarrollo de la base de datos de un proyecto personal que estoy realizando, para asi aplicar lo aprendido y poder realizar un trabajo a la altura y avanzar en mi proyecto

**Alcance del Informe**

El informe cubre los siguientes aspectos de SQL:

* Creación y diseño de la base de datos.
* Consultas SQL.
* Entidades, Atributos, Normalización etc.
* Modelo entidad relacion.

**Objetivos**

El desarrollo y documentación de todos los procesos realizados durante y para el proceso del desarrollo de mi base de datos, con el objetivo principal de que esta sea completa y funcional para el sistema el cual esta siendo desarrollada

**Metodología**

**Herramientas Utilizadas**

* **Sistema de Gestión de Bases de Datos (DBMS):** MySQL
* **Editor de SQL:** MySQL Workbench o similar

**Procedimientos**

1. **Creación de la Base de Datos:** Se construyeron las tablas necesarias con sus relaciones y claves principales.
2. **Inserción de Datos:** Se insertaron datos de ejemplo en las tablas para proporcionar un contexto realista.
3. **Desarrollo de Consultas:** Se diseñaron y ejecutaron consultas SQL para extraer y analizar datos según los requisitos.
4. **Procesos:** documentar procesos de simbolización, cardinalidad, normalización
5. **Modelo entidad relación:** desarrollo del modelo entidad relacion

**Desarrollo del Informe**

**Descripción de la Base de Datos**

Esta base de datos está diseñada para la gestión integral y eficiente de un complejo deportivo a través de una aplicación móvil. Su estructura está cuidadosamente elaborada para abarcar diversos aspectos clave de la administración del complejo, facilitando un manejo detallado y preciso de las operaciones diarias.

La base de datos incluye una serie de tablas interrelacionadas que cubren múltiples facetas del funcionamiento útil y necesario para la gestión adecuada y eficaz por lo que tiene mucho contenido en el cual profundizalemos ahora

**Esquema de la Base de Datos**

La base de datos proyecto está diseñada para gestionar información relacionada con un complejo deportivo que cuenta con bastante cantidad de sevicios

Tablas

type\_document: Almacena los tipos de documentos de identificación, con una columna para el identificador único y otra para el tipo de documento (por ejemplo, CC, PP, CE).

employees: Registra información sobre los empleados, incluyendo datos personales, detalles de contacto, ocupación y la fecha de contratación. Además, tiene columnas para rastrear la fecha de creación y actualización de los registros, así como los IDs de los usuarios que realizaron esas acciones.

customers: Contiene información sobre los clientes, similar a la tabla de empleados pero con datos específicos para los clientes. Incluye información de contacto, fecha de nacimiento, y el tipo de documento de identificación.

product\_categories: Define las categorías para los productos. Cada categoría tiene un identificador único, un nombre y una descripción.

products: Registra los productos disponibles para reservar. Incluye nombre del producto, descripción, precio y la categoría a la que pertenece.

reserve\_states: Enumera los posibles estados de una reserva (por ejemplo, Confirmada, Pendiente, Cancelada), con un identificador único para cada estado.

reserves: Almacena la información sobre las reservas realizadas por los clientes, incluyendo la fecha y hora de la reserva, el estado de la reserva, y el cliente asociado.

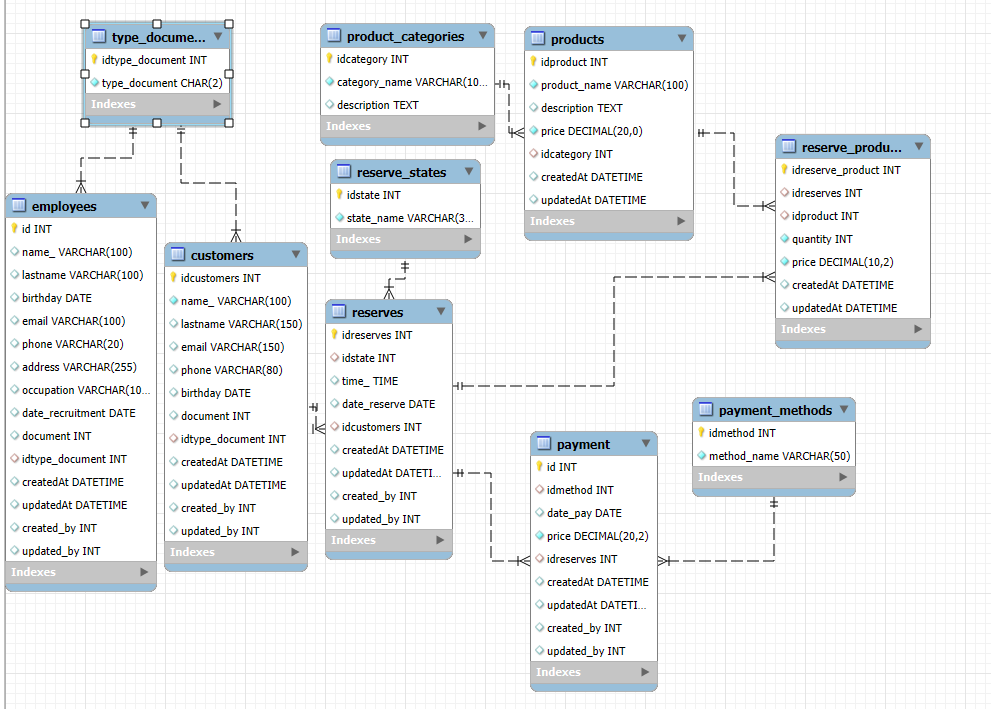
payment\_methods: Define los métodos de pago disponibles (por ejemplo, Tarjeta de Crédito, PayPal), con un identificador único y el nombre del método.

payment: Registra los pagos realizados para las reservas, incluyendo el método de pago utilizado, la fecha de pago, el monto y la reserva asociada.

reserve\_products: Detalla los productos incluidos en cada reserva, especificando la cantidad y el precio de cada producto reservado.

**El código relacionado puede observarse en el repositorio del github en la pate de “Backups”**

**GRAFICO DE LA BD**:



**Consultas SQL y Resultado**

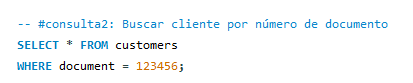
**Consulta 1:** Esta consulta busca empleados cuyo nombre sea 'Carlos' en la tabla de empleados.



El resultado esperado es una lista de todos los empleados que se llaman 'Carlos', con sus detalles completos.



**Consulta 2:** Esta consulta encuentra un cliente específico en la tabla de clientes basado en su número de documento.



El resultado esperado es la información del cliente cuyo número de documento es 123456, por l que en realidad no esperamos nada a no ser un numero valido.



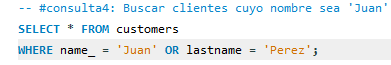
**Consulta 3**: Esta consulta selecciona empleados que tienen la ocupación de 'Analista' y cuyo nombre sea “carlos



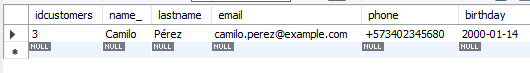
El resultado esperado es una lista de empleados que cumplen ambos criterios:



**Consulta 4:** Esta consulta busca clientes cuyo nombre sea 'Juan' o cuyo apellido sea 'Perez'.



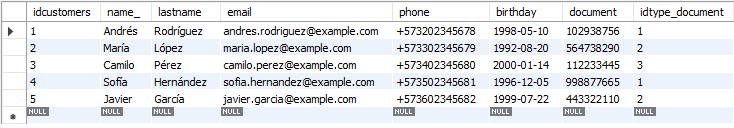
Se espera obtener una lista de clientes que cumplan al menos uno de estos criterios: nombre 'Juan' o apellido 'Perez'.



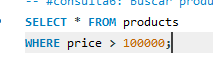
**Consulta 5:** Esta consulta selecciona clientes cuyo nombre no es 'Ana'.



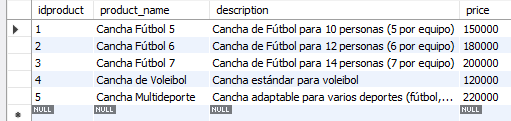
El resultado esperado es una lista de todos los clientes cuyo nombre no coincide con 'Ana'.



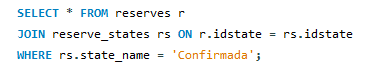
**Consulta 6:** Esta consulta busca productos cuyo precio sea mayor a 100,000.



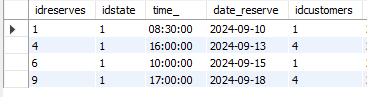
El resultado esperado es una lista de productos cuyo precio supera los 100,000.



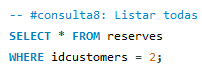
**Consulta 7:** Esta consulta busca reservas cuyo estado es 'Confirmada' al unir las tablas de reservas y estados de reserva.



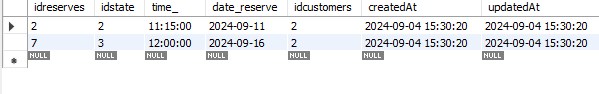
Se espera obtener una lista de todas las reservas que tienen el estado de 'Confirmada'.



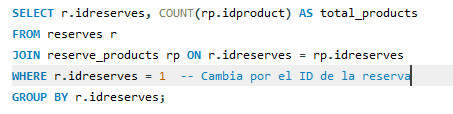
**Consulta 8**: Esta consulta lista todas las reservas realizadas por un cliente específico, identificado por un ID de cliente determinado.



El resultado esperado es un listado de todas las reservas asociadas con el cliente con el ID proporcionado.



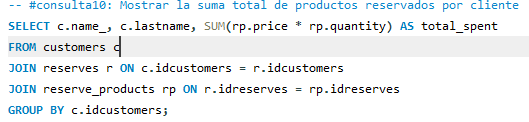
**Consulta 9**: Esta consulta cuenta el total de productos reservados en una reserva específica.



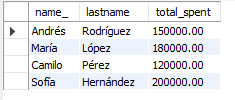
Se espera obtener el número total de productos asociados con una reserva dada, agrupados por el ID de la reserva.



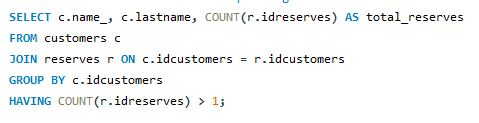
**Consulta 10**: Esta consulta muestra la suma total de dinero gastado por cada cliente en sus reservas.



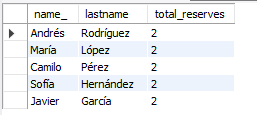
El resultado esperado es un listado de clientes con el total gastado en productos reservados, sumando el monto gastado en todas sus reservas.



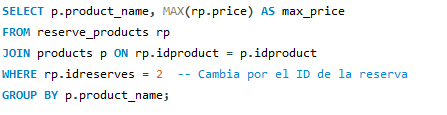
**Consulta 11:** Esta consulta encuentra clientes que tienen más de una reserva.



Se espera obtener una lista de clientes que han realizado más de una reserva, junto con el número total de reservas de cada uno.



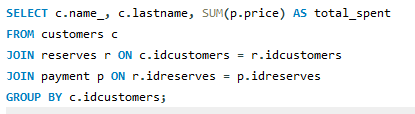
**Consulta 12:** Esta consulta muestra el producto más caro reservado en una reserva específica.



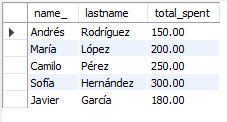
Se espera obtener el nombre del producto más caro y su precio dentro de una reserva dada.



**Consulta 13:** Esta consulta muestra el total de dinero gastado por cada cliente en sus reservas, sumando los pagos realizados.



El resultado esperado es una lista de clientes con el total gastado en todas sus reservas, según los pagos asociados.



**Diseño de Base de Datos**

**Modelo de Datos**

. El modelado de datos es una fase esencial en el diseño de la base de datos “Proyecto”, abordando aspectos fundamentales como el modelo Entidad-Relación, la normalización y la cardinalidad.

El modelo Entidad-Relación proporciona una representación visual de las entidades principales, como empleados, clientes, productos y reservas, así como de las relaciones entre ellas

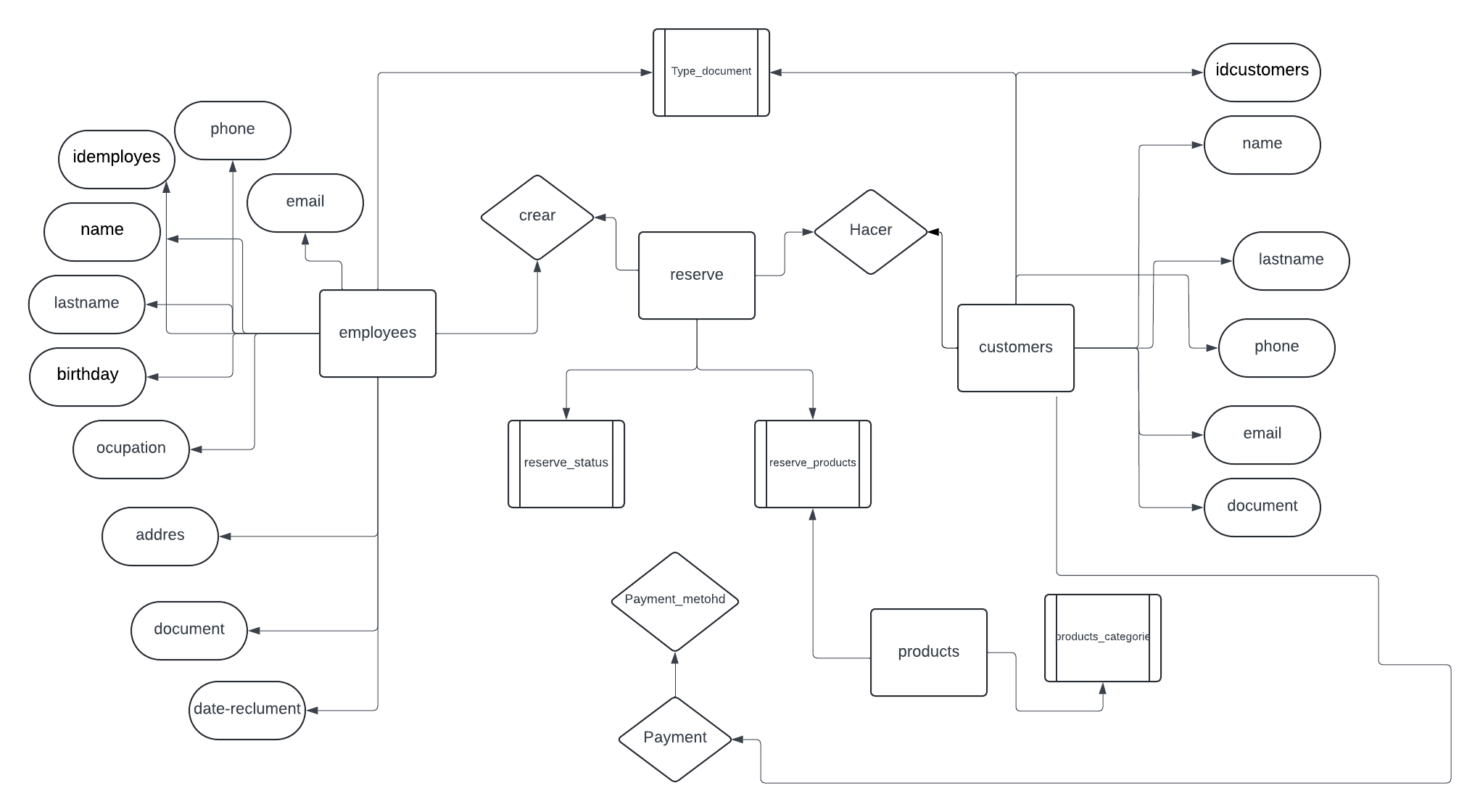
. La normalización asegura que los datos se organizan de manera eficiente, minimizando redundancias y mejorando la integridad de la información.

La cardinalidad define el número de instancias de una entidad que pueden estar asociadas con una instancia de otra entidad, estableciendo las restricciones necesarias para garantizar la precisión de las relaciones.

Asi que ahora vamos a profundizar en cada uno de ellos, ahí vamos.

#### Modelo entidad relación

Aquí se puede observar la primera parte de mi diagrama entidad relacion



#### Normalizacion

**1NF (Primera Forma Normal):** Todas las tablas contienen datos atómicos sin listas ni conjuntos.

**2NF (Segunda Forma Normal):** Todas las tablas están libres de dependencias parciales; los atributos no clave dependen completamente de la clave primaria.

**3NF (Tercera Forma Normal):** No hay dependencias transitivas; cada atributo depende directamente de la clave primaria.

#### Cardinalidad

**1. employees (1) ---- (N) type\_document (N) ---- (1) type\_document**

Un employee tiene un type\_document.

Un type\_document puede estar asociado con muchos employees.

**2. customers (1) ---- (N) type\_document (N) ---- (1) type\_document**

Un customer tiene un type\_document.

Un type\_document puede estar asociado con muchos customers.

**3. products (N) ---- (1) product\_categories (1) ---- (N) products**

Un product pertenece a una product\_category.

Una product\_category puede incluir muchos products.

**4. reserves (N) ---- (1) reserve\_states (1) ---- (N) reserves**

Una reserve tiene un reserve\_state.

Un reserve\_state puede aplicarse a muchas reserves.

**5. reserves (N) ---- (1) customers (1) ---- (N) reserves**

Una reserve está asociada a un customer.

Un customer puede tener muchas reserves.

**6. payment (N) ---- (1) payment\_methods (1) ---- (N) payment**

Un payment utiliza un payment\_method.

Un payment\_method puede ser usado en muchos payments.

**7. payment (N) ---- (1) reserves (1) ---- (N) payment**

Un payment está asociado a una reserve.

Una reserve puede tener muchos payments.

**8. reserve\_products (N) ---- (1) reserves (1) ---- (N) reserve\_products**

Un reserve\_product está asociado a una reserve.

Una reserve puede tener muchos reserve\_products.

**9. reserve\_products (N) ---- (1) products (1) ---- (N) reserve\_products**

Un reserve\_product se refiere a un product.

Un product puede estar en muchos reserve\_products.

**Consideraciones de Diseño**

**Elección de llaves y entidades**

* **type\_document** (Entidad débil)

idtype\_document (Llave primaria)

type\_document (No nulo, Único)

* **employees** (Entidad fuerte)

idemployes (Llave primaria)

name\_

lastname

birthday

email (Único)

phone

address

occupation

date\_recruitment

document (Único)

idtype\_document (Llave foránea, referencia a type\_document)

createdAt (Por defecto, CURRENT\_TIMESTAMP)

updatedAt (Por defecto, CURRENT\_TIMESTAMP con actualización automática)

created\_by

updated\_by

* **customers** (Entidad fuerte)

idcustomers (Llave primaria)

name\_ (No nulo)

lastname

email (Único)

phone

document (Único)

idtype\_document (Llave foránea, referencia a type\_document)

createdAt (Por defecto, CURRENT\_TIMESTAMP)

updatedAt (Por defecto, CURRENT\_TIMESTAMP con actualización automática)

created\_by

updated\_by

* **product\_categories** (Entidad débil depende de products)

idcategory (Llave primaria)

category\_name (No nulo, Único)

description

* **products** (Entidad fuerte)

idproduct (Llave primaria)

product\_name (No nulo)

description

price (No nulo)

idcategory (Llave foránea, referencia a product\_categories)

createdAt (Por defecto, CURRENT\_TIMESTAMP)

updatedAt (Por defecto, CURRENT\_TIMESTAMP con actualización automática)

* **reserve\_states** (Entidad fuerte)

idstate (Llave primaria)

state\_name (No nulo, Único)

* **reserves** (Entidad fuerte)

idreserves (Llave primaria)

idstate (Llave foránea, referencia a reserve\_states)

date\_reserve

idcustomers (Llave foránea, referencia a customers)

createdAt (Por defecto, CURRENT\_TIMESTAMP)

updatedAt (Por defecto, CURRENT\_TIMESTAMP con actualización automática)

created\_by

updated\_by

* **payment\_methods** (Entidad débil depende de payment)

idmethod (Llave primaria)

method\_name (No nulo, Único)

* **payment** (Entidad debil)

id (Llave primaria)

idmethod (Llave foránea, referencia a payment\_methods)

date\_pay

price (No nulo)

idreserves (Llave foránea, referencia a reserves)

createdAt (Por defecto, CURRENT\_TIMESTAMP)

updatedAt (Por defecto, CURRENT\_TIMESTAMP con actualización automática)

created\_by

updated\_by

* **reserve\_products** (Entidad Debil depende de productos y reserva)

idreserve\_product (Llave primaria)

idreserves (Llave foránea, referencia a reserves)

idproduct (Llave foránea, referencia a products)

quantity (No nulo)

price (No nulo)

createdAt (Por defecto, CURRENT\_TIMESTAMP)

updatedAt (Por defecto, CURRENT\_TIMESTAMP con actualización automática)

**Relaciones entre tablas**

* employees (Relación entre employees y type\_document)
* idtype\_document (Llave foránea en employees, referencia a type\_document(idtype\_document))
* customers (Relación entre customers y type\_document)
* idtype\_document (Llave foránea en customers, referencia a type\_document(idtype\_document))
* products (Relación entre products y product\_categories)
* idcategory (Llave foránea en products, referencia a product\_categories(idcategory))
* reserves (Relación entre reserves y customers)
* idcustomers (Llave foránea en reserves, referencia a customers(idcustomers))
* reserves (Relación entre reserves y reserve\_states)
* idstate (Llave foránea en reserves, referencia a reserve\_states(idstate))
* payment (Relación entre payment y payment\_methods)
* idmethod (Llave foránea en payment, referencia a payment\_methods(idmethod))
* payment (Relación entre payment y reserves)
* idreserves (Llave foránea en payment, referencia a reserves(idreserves))
* reserve\_products (Relación entre reserve\_products y reserves)
* idreserves (Llave foránea en reserve\_products, referencia a reserves(idreserves))
* reserve\_products (Relación entre reserve\_products y products)
* idproduct (Llave foránea en reserve\_products, referencia a products(idproduct))

**Nombre y descripción**

El sistema de base de datos denominado “Proyecto” ha sido desarrollado con el propósito de optimizar y sistematizar la gestión integral de un complejo deportivo. Este sistema está diseñado para proporcionar una solución eficiente y organizada para el manejo de diversas facetas del complejo, incluyendo la administración de empleados, clientes, canchas y otros aspectos esenciales.

El objetivo primordial de la base de datos es asegurar un funcionamiento rápido y eficiente del sistema, facilitando la organización y el acceso a la información clave relacionada con las operaciones del complejo deportivo. Para lograr esto, “Proyecto” incluye una serie de tablas interrelacionadas que abarcan las siguientes áreas:

**employees:** Contiene información detallada sobre el personal del complejo, incluyendo datos personales, tipo de documento de identificación y detalles sobre su ocupación y fecha de contratación.

**customers:** Mantiene un registro completo de los clientes del complejo, gestionando sus datos personales, información de contacto y detalles relacionados con las reservas y pagos.

**product\_categories**: Organiza los productos del complejo en distintas categorías, facilitando su gestión y comercialización.

**products:** Administra el inventario de productos disponibles, incluyendo información sobre su descripción, precio y categoría.

**reserve\_states:** Define los distintos estados posibles para las reservas realizadas, permitiendo un seguimiento más efectivo del estado de cada reserva.

**reserves:** Facilita la gestión de las reservas, registrando el estado de la reserva, la fecha y hora, y su asociación con el cliente correspondiente.

**payment\_methods:** Registra los métodos de pago disponibles para las transacciones realizadas por los clientes.

**payment:** Registra los pagos efectuados por los clientes, incluyendo el método de pago, la fecha y el monto total, además de la asociación con las reservas realizadas.

**reserve\_products:** Detalla los productos específicos incluidos en cada reserva, especificando la cantidad y el precio de cada producto.

**Análisis y Discusión**

El análisis general del proyecto de desarrollo la base de datos ha sido positivo y ha cumplido con mis expectativas. He realizado una evaluación exhaustiva y considero que todo lo implementado es funcional y adecuado. Aunque reconozco que siempre hay oportunidades para mejorar y ser más eficiente, estoy satisfecho con el trabajo realizado.

Este proyecto representa un importante primer paso en mi desarrollo profesional y personal, permitiéndome abordar mi proyecto final de manera seria y meticulosa. Estoy contento con los logros alcanzados y con la calidad del trabajo desarrollado, que sienta una sólida base para futuros avances en mi carrera y mi desarrollo de proyecto de vida.

**Interpretación de Resultados**

Los resultados obtenidos de la base de datos confirman que esta es funcional y está estructurada de manera correcta y adecuada, siguiendo las instrucciones proporcionadas. Se realizaron diversas consultas y se atendieron todas las solicitudes planteadas, obteniendo resultados consistentes y precisos.

La estructura de la base de datos cumple con los requisitos establecidos y se ha verificado su correcto funcionamiento a través de pruebas exhaustivas. Bajo mi criterio, la base de datos es adecuada y está completa, incluyendo toda la información necesaria para cumplir con su propósito de manera efectiva. Esto asegura que la base de datos no solo es operativa, sino que también es útil y capaz de soportar las operaciones del sistema de manera eficiente.

**Conclusiones**

El desarrollo de esta base de datos representó un desafío significativo, pero sumamente enriquecedor. A lo largo de este proyecto, pude enfrentar y superar diversos retos, aplicando las mejores prácticas y enfoques para lograr una solución eficiente y bien estructurada. Este proceso no solo me permitió consolidar mis conocimientos, sino que también avanzó considerablemente mi proceso de desarrollo profesional.

El proyecto me brindó una valiosa experiencia en la creación y gestión de bases de datos, estableciendo una sólida base sobre la cual puedo construir y expandir en el futuro. Estoy convencido de que los aprendizajes adquiridos y las habilidades desarrolladas durante este proyecto me preparan adecuadamente para enfrentar futuros desafíos en el campo del desarrollo de bases de datos, permitiéndome avanzar en mi desarrollo y llevar a cabo proyectos de mayor envergadura con éxito, principalmente mi futuro proyecto de grado y proyecto personal en el cual esta basado este desarrollo.

**Recomendaciones**

Por el momento, se requiere un uso básico de la base de datos, lo que implica descargar los scripts necesarios para su funcionamiento y realizar pruebas. Los detalles específicos se encuentran en el repositorio de GitHub correspondiente. Para asegurar un funcionamiento adecuado, es importante descargar el último backup disponible y los scripts asociados, garantizando así que el sistema opere de manera efectiva y eficiente.

**Referencias**