Documentación y Evidencia de la prueba tecnica suministrada por Matrixtech para postulación a nuevo cargo.

Prueba Realizada por C.C. 1.036.657.518 Hamilton Ferney Mesa Restrepo.

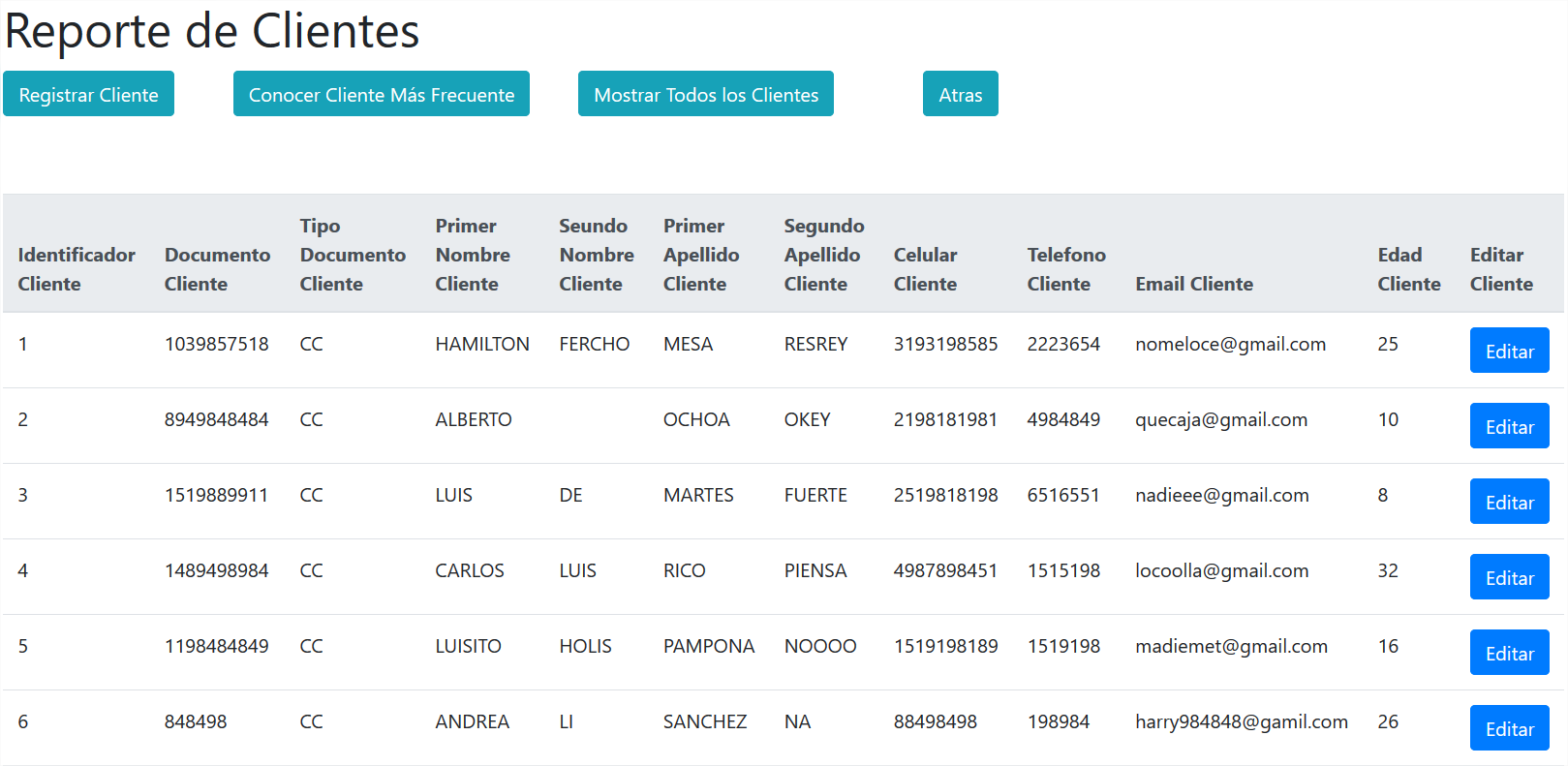
**Puntos del taller:**

1. **Almacenar la información básica de los clientes para poder conocer quien tiene alquilado un juego y poder reclamarlo cuando se venza el periodo de alquiler**.

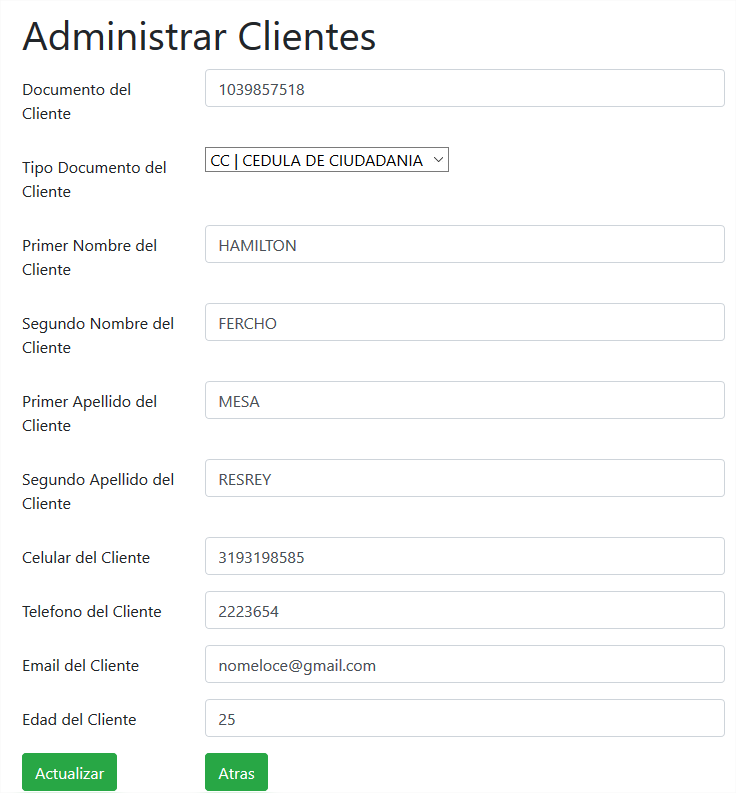
En el reporte de Alquileres, sale cada registro realizado, el juego que se presto, la plataforma y el cliente al que se le presento, con documento del cliente, podremos ir al Reporte de Clientes y obtener información mas detallada del mismo.



En el reporte de clientes encontraremos el nombre completo del cliente, el teléfono fijo, el celular, el correo y podremos actualizar la información del cliente dando click en el botón [Editar].



Edición de un cliente:



1. **Poder definir el precio de los alquileres que cambian periódicamente por título de juego.**

En la pantalla de ‘Reporte de Juegos’, podremos seleccionar el juego que deseemos modificar, le damos click al botón [Editar]. y nos llevara al la pantalla de ‘Administrar Juegos’.



En la pantalla de ‘Administrar Juegos’, se podrá modificar la información del juego, incluyendo el precio de alquiler, marca del juego (se agrego botón para crear nuevas marcas en caso de que se desee).



1. **Conocer el cliente más frecuente.**

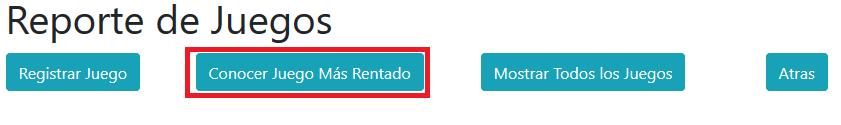
En la pantalla de ‘Reporte de Clientes’, hay un botón llamado (Conocer Cliente Más Frecuente), al dar click en el, el sistema consultara todas las transacciones y calculara cual es el cliente con mas transacciones, y lo mostrar en pantalla:

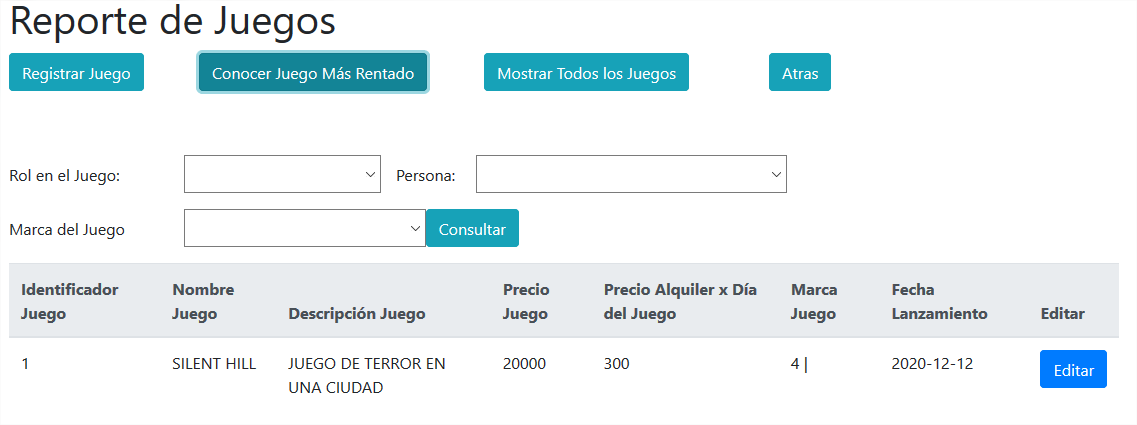




1. **Conocer el título de juego más rentado.**

En la pantalla de ‘Reporte de Juegos’, hay un botón llamado (Conocer Juego Más Rentado), al dar click en el, el sistema consultara todas las transacciones y calculara cual es el juego mas rentado, y lo mostrar en pantalla:





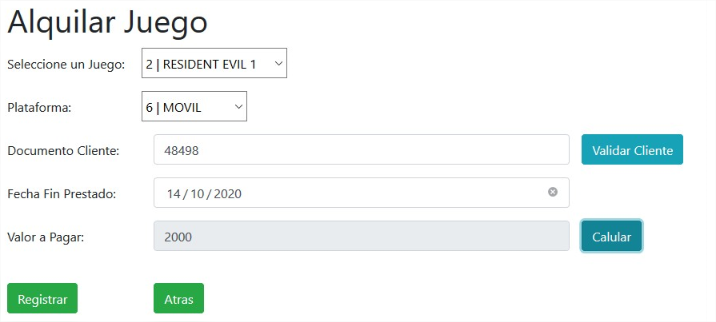
1. **Permitir registrar todos los alquileres hechos y generar prueba de compra.**

En la pantalla de ‘Reporte de Alquileres’ se mostrara la información de las transacciones realizadas en el sistema:



Ademas en esta opción hay un botón llamado (Alquilar Juego), donde al darle click en el, se presenta la pantalla de registro de Alquileres.

Esta pantalla que el cliente este registrado, que el juego tenga plataformas disponibles y calcula el valor a pagar de acuerdo a la cantidad de días de prestamo \* valorAlquilerJuego.



1. **Poder consultar las ventas del día.**

En la pantalla de ‘Reporte de Alquileres’, hay un botón llamado (Ventas del Día), al dar click en el día, el sistema filtra y presentara solo las ventas realizadas en el día actual, ahora si el usuario desea volver a ver todas las ventas, puedes dar click en el botón (Todas las Ventas).



1. **Poder consultar por director de Juego.**

En la pantalla de ‘Reporte de Juegos’, hay unos filtros en los cuales podemos indicar el Rol, la Persona y la Marca para filtrar la información, en este caso filtraremos por Rol = ‘Director’ y Persona = ‘Jorge Rios’. Y el sistema nos mostrar los juegos en los que Jorge Rios ha tenido Rol de Director:



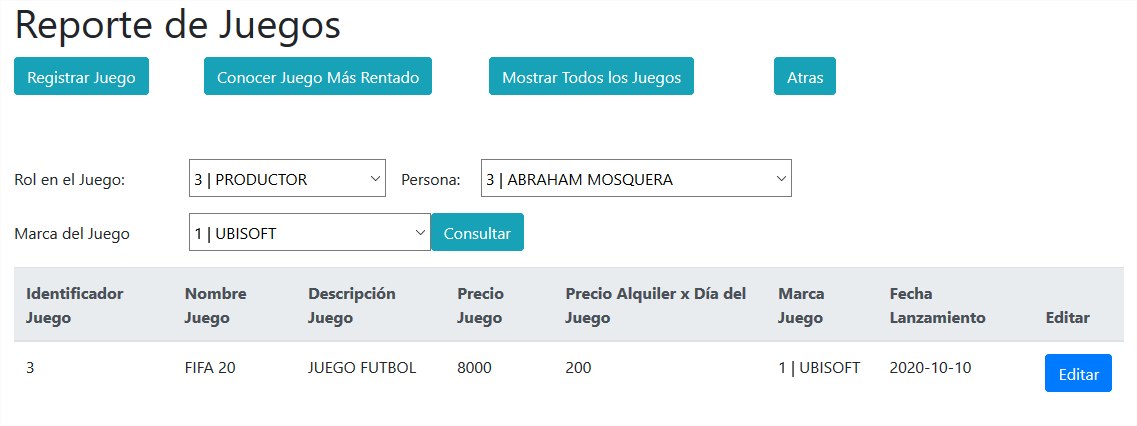
1. **Poder consultar por protagonista del juego.**

En la pantalla de ‘Reporte de Juegos’, hay unos filtros en los cuales podemos indicar el Rol, la Persona y la Marca para filtrar la información, en este caso filtraremos por Rol = ‘Protagonista’ y Persona = ‘Hamilton Mesa’. Y el sistema nos mostrar los juegos en los que Hamilton Mesa ha tenido Rol de Protagonista:



1. **Poder consultar por productor y&o marca del juego.**

En la pantalla de ‘Reporte de Juegos’, hay unos filtros en los cuales podemos indicar el Rol, la Persona y la Marca para filtrar la información, en este caso filtraremos por Rol = ‘Productor’, Persona = ‘Abraham’ y Marca = ‘UBISOFT’. Y el sistema nos mostrar los juegos en los que Abraham ha tenido Rol de Productor y la Marca del Juego es Ubisoft:



1. **Tener registrado de cada titulo, nombre, año, protagonistas, director, productor y tecnología (Xbox, PlayStation, Nintendo,PC,…)**

Con el siguiente SELECT obtendremos la repartición de los papales/roles que tuvieron x personas en los diferentes juegos:

SELECT

JUE.ID\_JUEGO, JUE.NOMBRE AS NOMBRE\_JUEGO, ROL.ID\_ROL, ROL.NOMBRE AS DESCIPCION\_ROL, PER.ID\_PERSONA, PER.NOMBRES

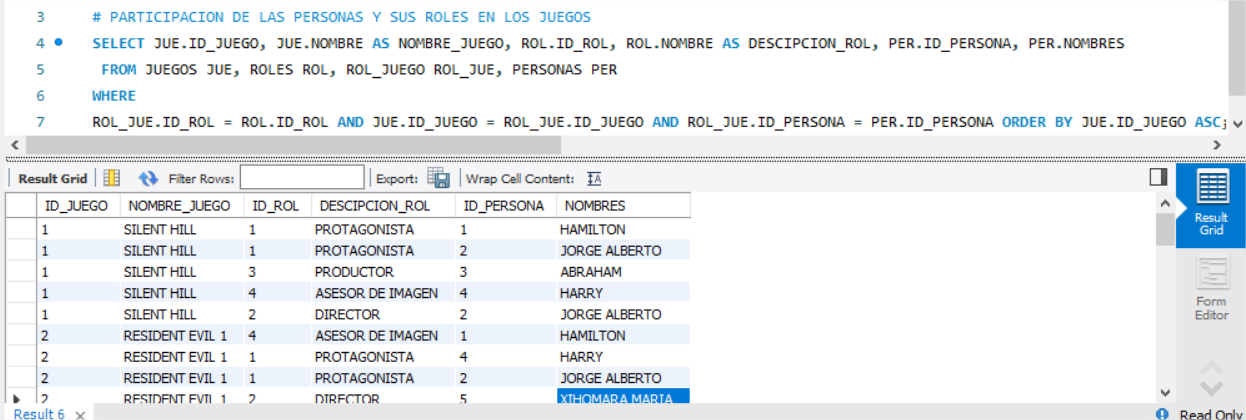
FROM

JUEGOS JUE, ROLES ROL, ROL\_JUEGO ROL\_JUE, PERSONAS PER

WHERE

ROL\_JUE.ID\_ROL = ROL.ID\_ROL AND JUE.ID\_JUEGO = ROL\_JUE.ID\_JUEGO

AND ROL\_JUE.ID\_PERSONA = PER.ID\_PERSONA ORDER BY JUE.ID\_JUEGO ASC;



Con el siguiente Select obtendremos las plataformas(tecnologías) en las que esta disponible cada juego.

SELECT

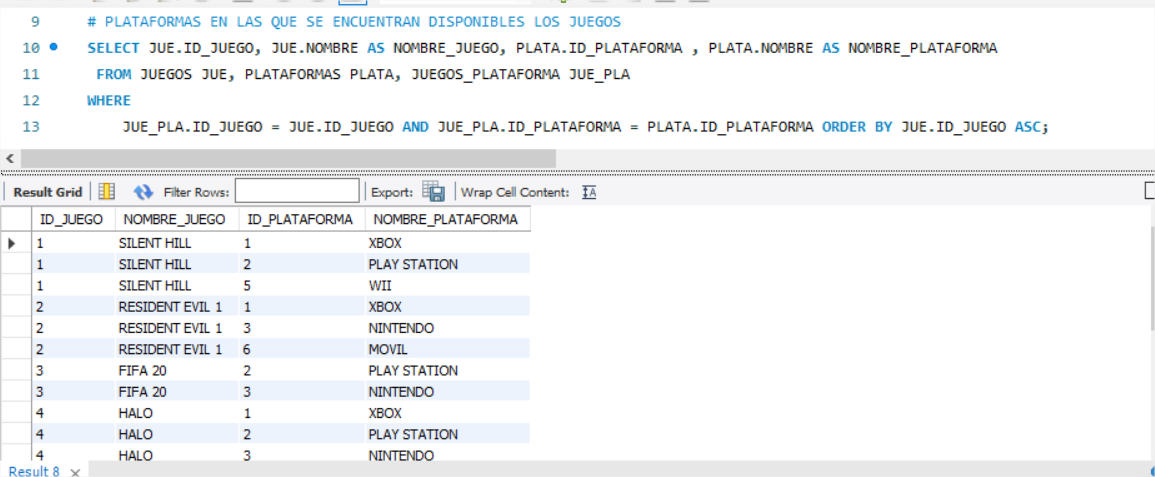
JUE.ID\_JUEGO, JUE.NOMBRE AS NOMBRE\_JUEGO, PLATA.ID\_PLATAFORMA , PLATA.NOMBRE AS NOMBRE\_PLATAFORMA

FROM

JUEGOS JUE, PLATAFORMAS PLATA, JUEGOS\_PLATAFORMA JUE\_PLA

WHERE

JUE\_PLA.ID\_JUEGO = JUE.ID\_JUEGO AND JUE\_PLA.ID\_PLATAFORMA = PLATA.ID\_PLATAFORMA ORDER BY JUE.ID\_JUEGO ASC;



1. El dueño quiere exponer un servicio web para que cualquier cliente consulte su balance, fecha de entraga y títulos alquilados.

Se crea un servicio el cual retornara todas los alquileres que han sido realizado por un usuario:

Documentación Técnica del servicio:

|  |  |
| --- | --- |
| Tipo de Servicio | REST |
| Formato Respuesta | JSON |
| Tipo de Petición HTTP | GET |
| Url | {IpServicio}:{puertoServicio} |
| recurso | /matrixtienda/getAlquileres/{documentoCliente} |
| Respuesta del Servicio | El servicio retornara una lista de objetos con la información de las transacciones realizadas por el usuario. |
| Content-Type | application/json; charset=utf8 |

A continuación se describe ejemplos de petición y recepción de la información, usando el aplicativo Postman, que nos permite simular dichas tranasacciones:

Ejemplo de Petición

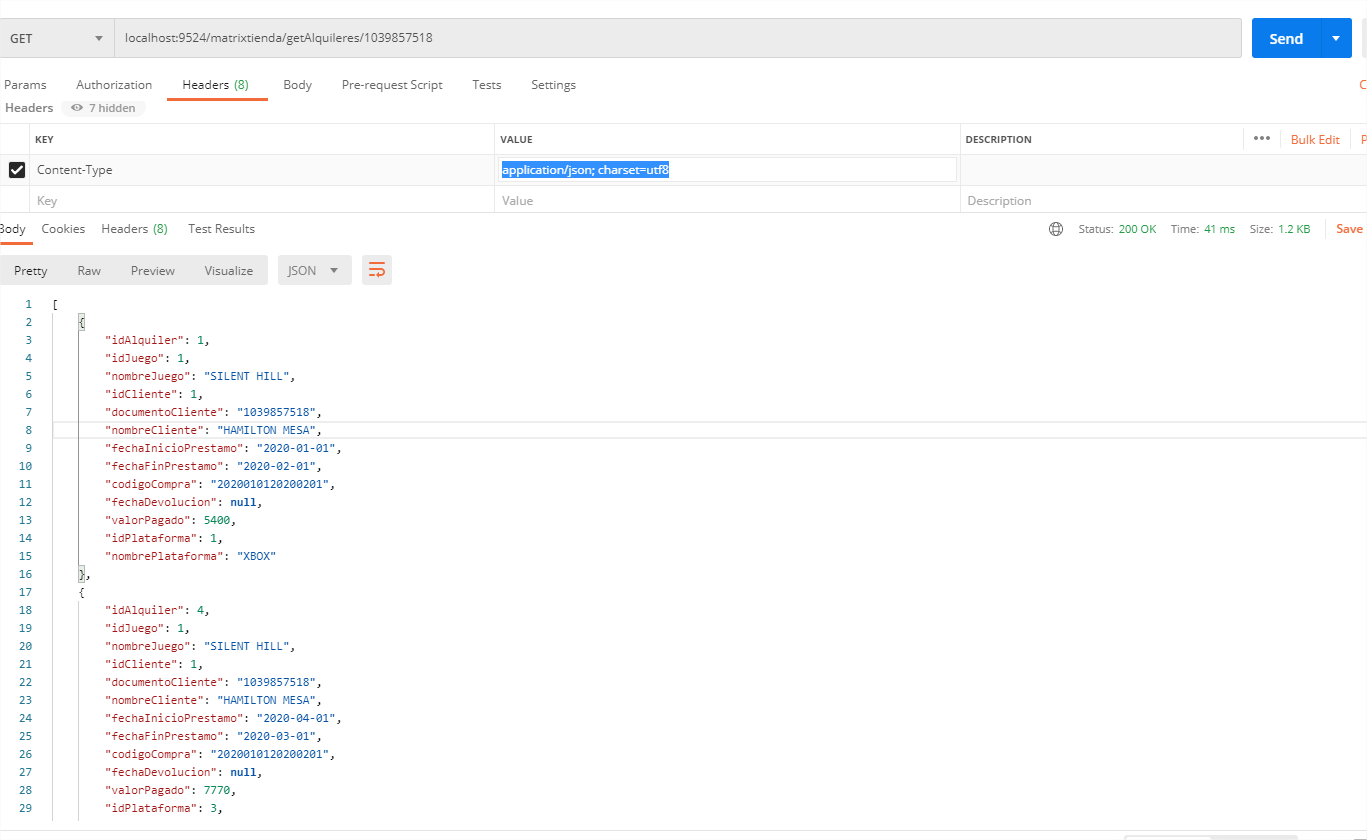
localhost:9524/matrixtienda/getAlquileres/1039857518



Ejemplo de Respuesta

El objeto de la respuesta tiene la siguiente estructura, por lo cual se describe campo a campo, y como debería ser definido:

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo de Dato** | **Descripción Campo** |
| idAlquiler | Numerico | Identificador unico del alquiler. |
| idJuego | Numerico | Identificador unico del juego alquilado. |
| nombreJuego | Cadena | Nombre del juego alquilado. |
| idCliente | Numerico | Identificador unico del cliente que alquilo el juego. |
| documentoCliente | Cadena | Número de documento del cliente que alquilo el juego. |
| nombreCliente | Cadena | Nombre del cliente que alquilo el juego. |
| fechaInicioPrestamo | Fecha | Fecha en que se alquilo el juego. |
| fechaFinPrestamo | Fecha | Fecha hasta donde se solicito el préstamo. |
| codigoCompra | Cadena | Código unico que genero por la compra. |
| fechaDevolucion | Fecha | Fecha en que se devolvió el juego. |
| valorPagado | Numerico | Valor pagado por el cliente por el préstamo. |
| idPlataforma | Numerico | Identificador unico de la plataforma del juego alquilado. |
| nombrePlatafotma | Cadena | Nombre de la plataforma del juego alquilado. |



**INFORMACIÓN DE LA ARQUITECTURA Y MONTAJE DEL PROYECTO:**

La construcción del proyecto esta distribuido en 3 secciones importantes, las cuales se proceden a describir.

**Base de Datos:**

Se uso el motor de base de datos Mysql.

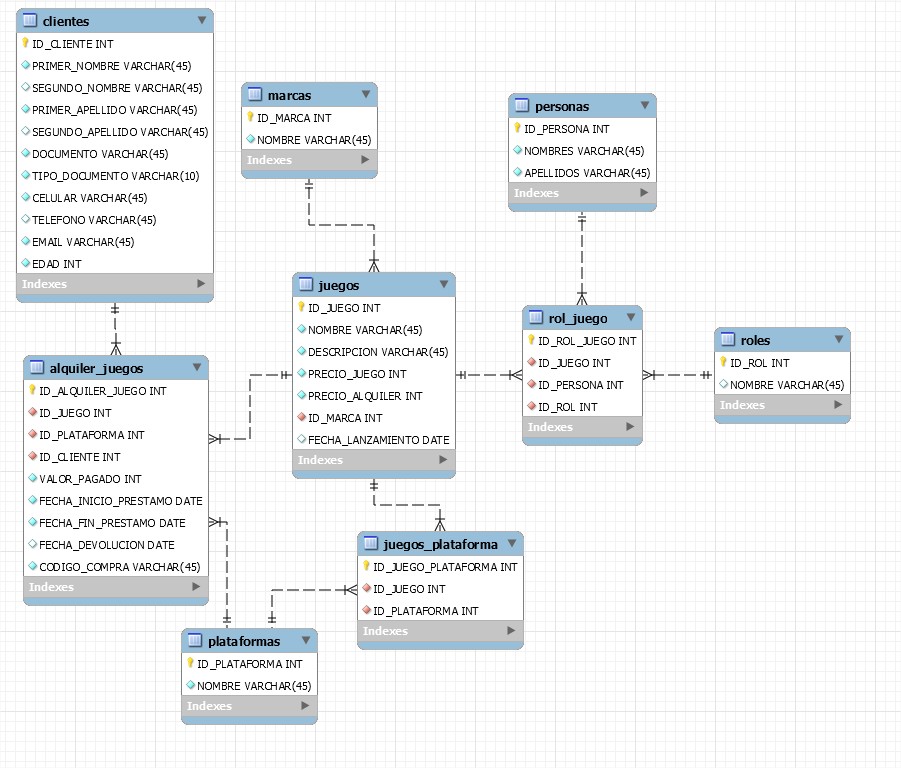
Se gestiono la base de datos desde la herramienta Mysql Workbetch 8.0

Se opto por Mysql ya que es una base de datos relacional, y ademas gratuita y cumplia con las necesidades manifestadas en el documento.

Para la creación de todos los objetos de base de datos y la población de los mismo, se adjunta el archivo: ScriptCreacionYPoblacionTablasPruebaTecnicaMatrix.sql

Y ademas se agrega este mismo al repositorio: TiendaMatrix\_api\matrixtienda\_api\src\main\java\com\matrix\matrixtienda\_api\docs\ScriptCreacionYPoblacionTablasPruebaTecnicaMatrix.sql

Se agrega el modelo relacional de la base de datos:



**Backend - Servicios - API:**

Se crearon los servicios usando el framework de Spring Boot en su versión 2.3.4.

Se uso el gestor de dependencias MAVEN para que orquestar y delegar dicha tarea a esta excelente herramienta.

Las servicios expuestos reciben y responden las peticiones sea GET o POST en formato JSON.

Para la conexión entre Spring y la Base de Datos, se uso la utilidad de jdbcTemplate del modelo de persistencia de jdbc.

Utilice el id de Eclipse para el desarrollo de los servicios y puesta en marcha de los mismo.

Se puede bajar el proyecto api desde el repositorio: <https://github.com/harry9524/TiendaMatrix_api.git>

**Frontend - APP**

Para la creación de las interfaces se uso el framework de Angular en su versión 10.1.6.

Se utilizo la dependencia de httpClient para realizar y recibir las peticiones al servicio.

Utilice el ide Visual Studio Code para el desarrollo y arranque de la app.

Se puede bajar el proyecto api desde el repositorio: <https://github.com/harry9524/TiendaMatrix_app.git>