|  |
| --- |
| Actividades **Facultad:** Ingeniería. **Asignatura:** Administración y Gestión de Base de Datos  **Ciclo y Turno:** **Semestre Académico:** 2021-2 Docente: Ing. Daniel Yucra Fecha: |

**Actividades:**

Alumno MOZA REYES CESAR OSWALDO código: 2019002353

3.)

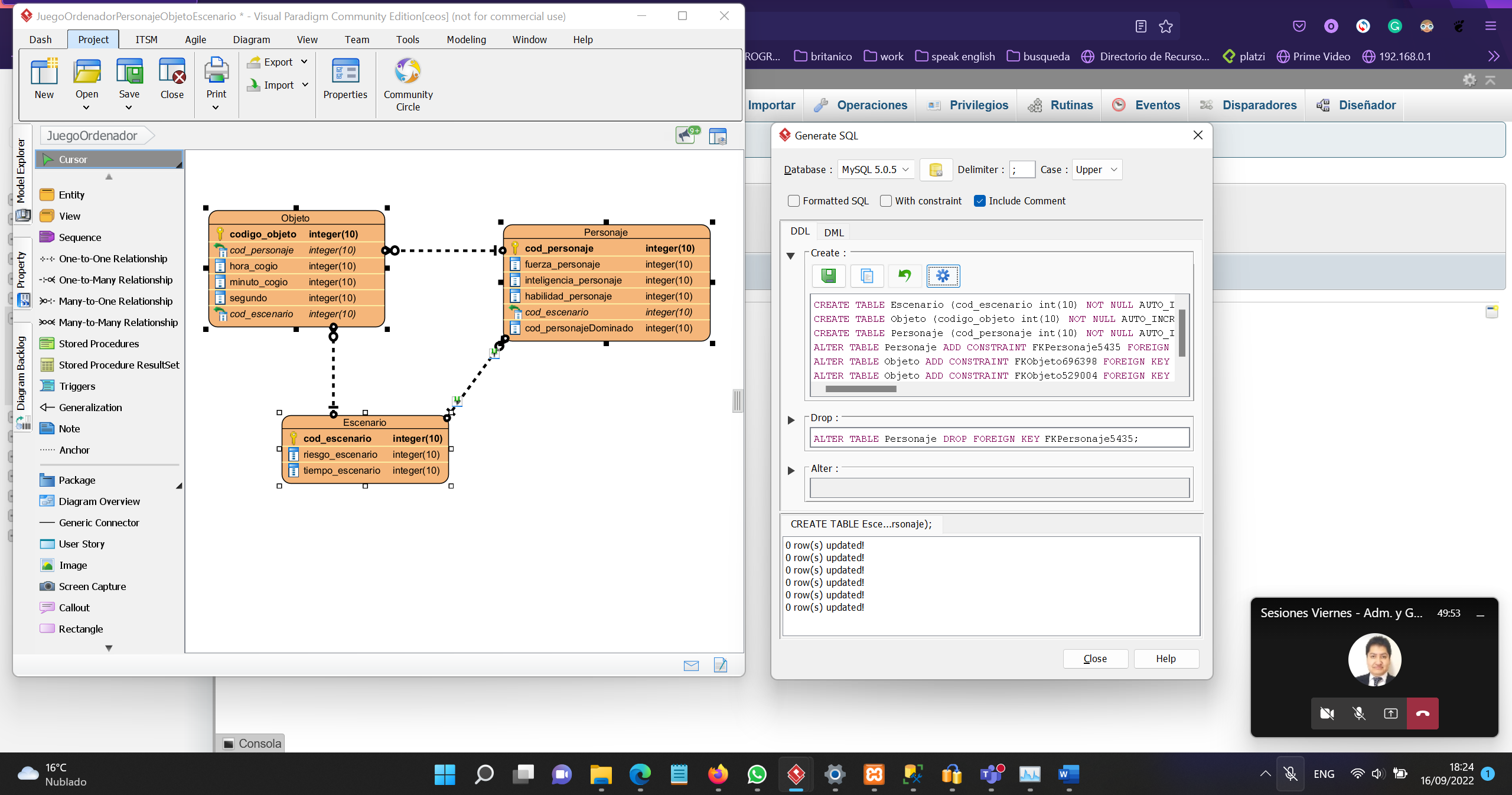
Definir el modelo E/R para un juego de ordenador. En el juego se utilizan conceptos como el de personaje, objeto y escenario. Sabemos que:

* Cada personaje tiene un nombre único y lleva asociados 3 valores numéricos indicando su nivel de fuerza, inteligencia y habilidad. Además, interesa saber en cada momento en que escenario (número de escenario) se encuentra.
* Cada personaje puede dominar a otros personajes (a 0 o mas), pero solo puede ser dominado por otro, o por ninguno.
* Los personajes tienen en cada momento algunos objetos en su poder. Además de saber que objetos tiene cada personaje hay que guardar información acerca de en qué momento (hora, minuto y segundo) cogió el personaje el objeto. Puede haber objetos que no estén en posesión de ningún personaje (no los ha encontrado nadie aun), pero ninguno puede pertenecer a dos personajes al mismo tiempo.
* Los objetos se distinguen entre sí por un código y tienen asociado un numero de escenario, el número del escenario en el que se encuentran situados al comienzo del juego. Cada escenario puede contener 0 o más objetos.
* Cada escenario tiene un número que lo identifica y 2 factores riesgo, tiempo que indican características particulares del escenario.

1. **Diseñar en Visual Paradig el modelo Entidad – Relación**

Diagrama

Descripción generada automáticamente



1. **Implementar el Modelo en MySQL, capturar pantalla del diagrama en phpMyadmin**

Interfaz de usuario gráfica, Diagrama, Aplicación

Descripción generada automáticamente

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

1. **Ingresar datos y registros al modelo en PhpMyadmin**

TABLA ESCENARIO



TABLA PERSONAJE

Tabla

Descripción generada automáticamente

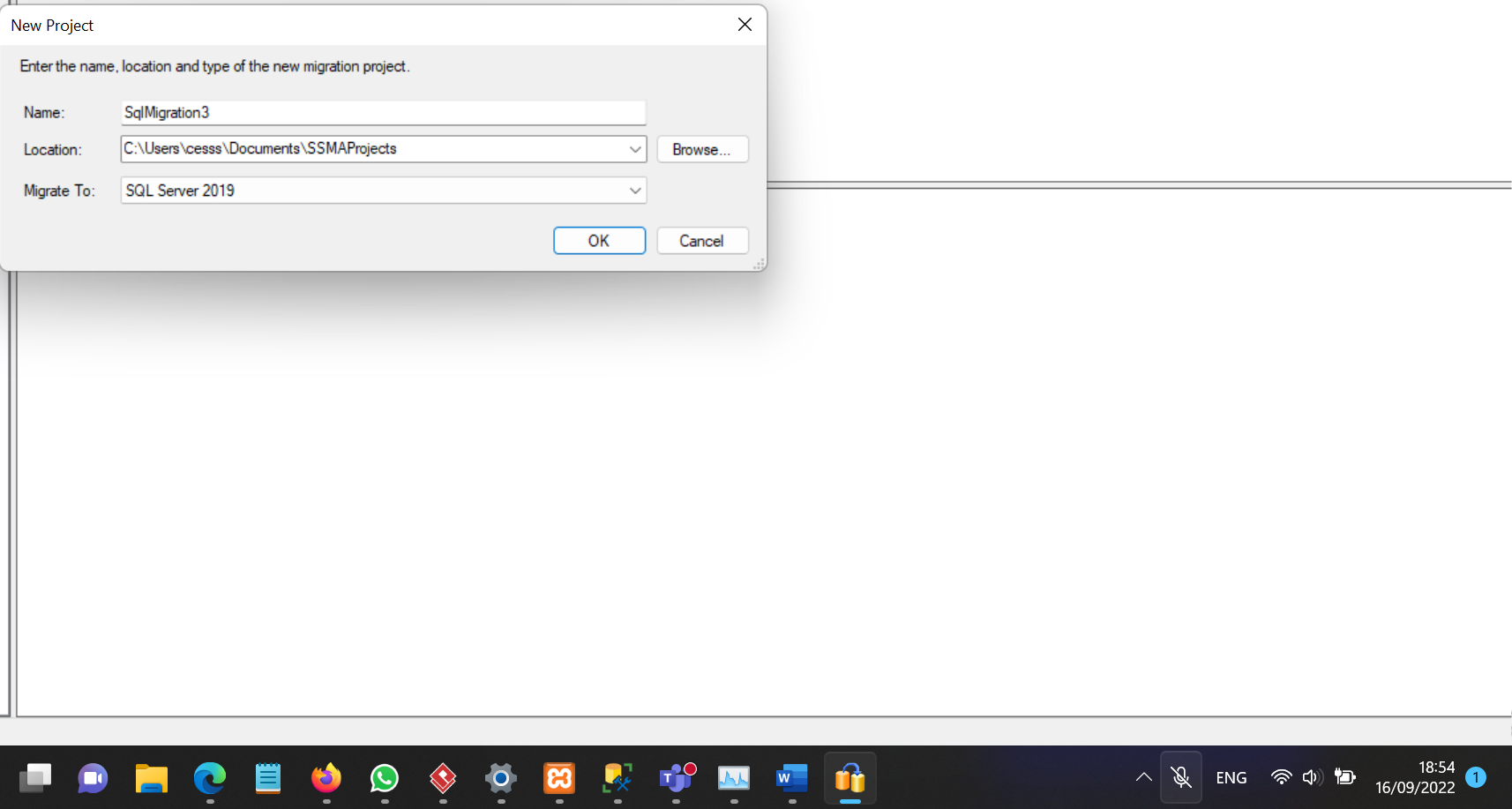
TABLA OBJETO

Imagen que contiene Tabla

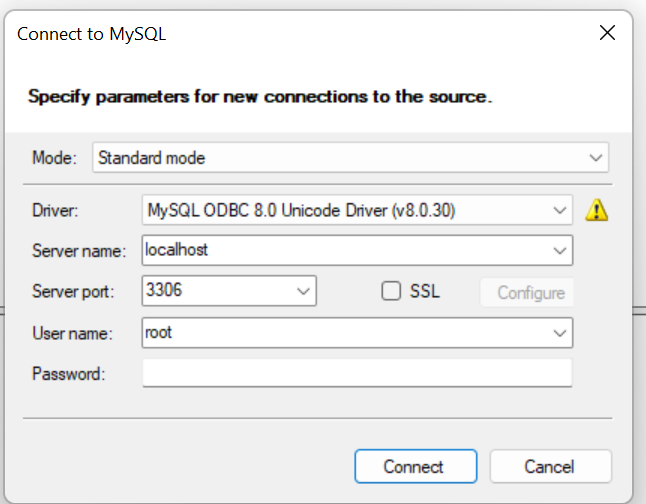
Descripción generada automáticamente

1. **Migrar la Base de Datos a SQL Server (Capturar las pantallas de la migración, el diagrama y los registros migrados en SQL Server)**

CREAMOS NUEVO PROYECTO DE MIGRACION



CONEXIÓN CON MYSQL



CONEXIÓN CON MYSQL

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

CONEXIÓN CON MYSQL

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

REPORTE DE LA BASE DE DATOS SIN ERRORES

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

**CREAMOS LA BASE DE DATOS EN SQL SERVER PARA PODER CONECTARNOS DESDE EL ASISTENTE**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

CREAMOS LA BASE DE DATOS EN SQL SERVER 2019

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

NOS CONECTAMOS CON SQL SERVER

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

CONEXIÓN EXITOSA CON SQL SERVER

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

SE GENERA EL ESQUEMA EN SQL SERVER 2019

Interfaz de usuario gráfica

Descripción generada automáticamente

SE SINCRONIZAN MYSQL Y SQL SERVER

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

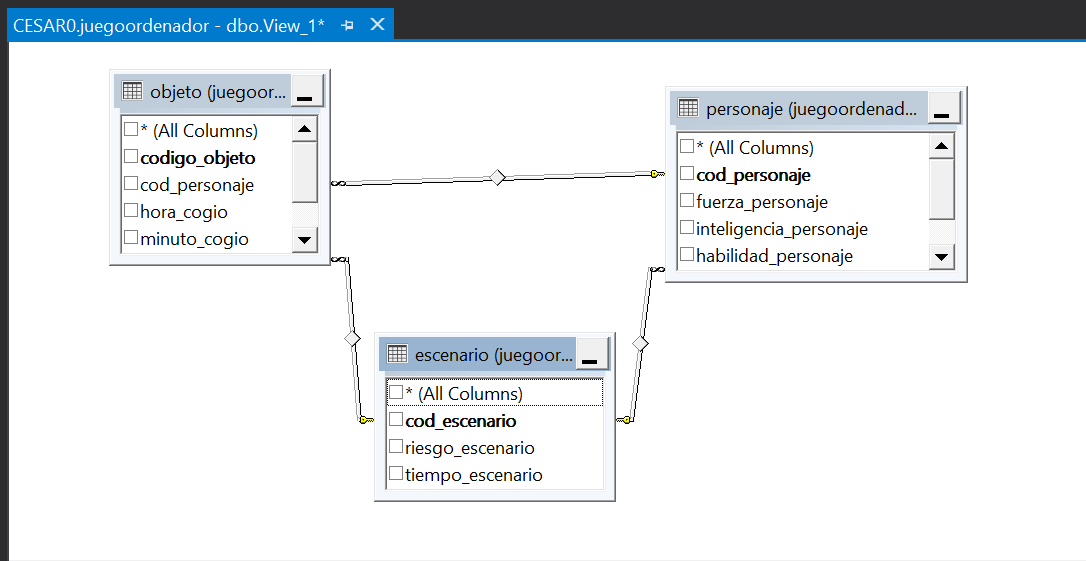
Descripción generada automáticamente

REVISAMOS SI SE CREARON LAS TABLAS CON LA SINCRONIZACION

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

GENERAMOS EL DIAGRAMA DE LA VISTA DE LAS TABLAS CON SUS RELACIONES



CONECTAMOS A MYSQL OTRA VEZ PARA MIGRACION DE LOS DATOS

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

CONECTAMOS A SQL SERVER OTRA VEZ PARA MIGRACION DE LOS DATOS

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

MIGRACION FINALIZADA DE LOS DATOS HACIA SQL SERVER

Tabla

Descripción generada automáticamente con confianza media

CONSULTANDO LA MIGRACION DE LA TABLA PERSONAJE EN SQL SERVER

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente con confianza media

CONSULTANDO LA MIGRACION DE LA TABLA OBJETO EN SQL SERVER

Tabla

Descripción generada automáticamente con confianza media

CONSULTANDO LA MIGRACION DE LA TABLA ESCENARIO EN SQL SERVER

Tabla

Descripción generada automáticamente

1. **Elaborar procedimientos almacenados para 2 tablas maestras (Insertar, Consultar, Actualizar y eliminar)**

**TABLA ESCENARIO**

**INSERTAR**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

**CONSULTAR**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente



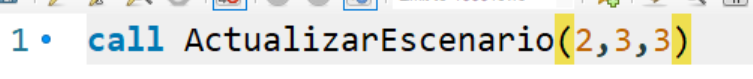
Tabla

Descripción generada automáticamente

**ACTUALIZAR**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente



Tabla

Descripción generada automáticamente con confianza baja

**ELIMINAR POR CODIGO**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



Tabla

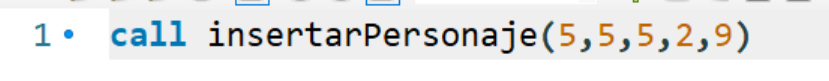
Descripción generada automáticamente

**TABLA PERSONAJE**

**INSERTAR**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente con confianza media



Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**CONSULTAR**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

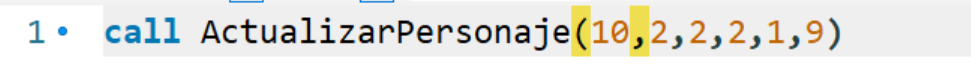
Tabla

Descripción generada automáticamente

ACTUALIZAR

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente



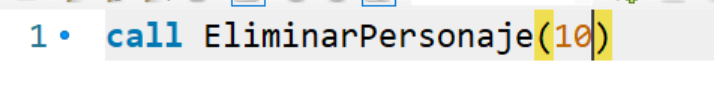
Tabla

Descripción generada automáticamente

**ELIMINAR POR CODIGO**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



Tabla

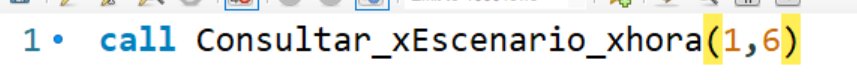
Descripción generada automáticamente

1. **Crear un procedimiento almacenado que permite realizar consultas en 3 tablas, el mismo debe ser con algún filtro de consulta (rangos), con una condición (Menor, igual o mayor que), y que te muestre un listado como respuesta a la consulta**

**CONSULTAMOS A LAS TABLAS OBJETO, PERSONAJE Y ESCENARIO USANDO JOIN, FILTRAMOS POR CODIGO DE ESCENARIO Y CONDICIONAMOS QUE SEAN LOS REGISTROS QUE SE DIERON DESPUES DE LAS 6 (HORA QUE SE COGIO)**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente



Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

1. Crear la BD proced.

**CREATE DATABASE juegoordenador;**

**USE juegoordenador;**

1. Crear las tablas del caso

**CREATE TABLE `auditoria` (**

`id` int(11) NOT NULL,

`accion` varchar(200) DEFAULT NULL,

`fecha` datetime DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

**INSERT INTO `auditoria**` (`id`, `accion`, `fecha`) VALUES

(0, 'se creo un registro en personaje', '2022-11-09 19:25:19');

**CREATE TABLE `escenario` (**

`cod\_escenario` int(10) NOT NULL,

`riesgo\_escenario` int(10) NOT NULL,

`tiempo\_escenario` int(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

**INSERT INTO `escenario` (`cod\_escenario`, `riesgo\_escenario`, `tiempo\_escenario`) VALUES**

(1, 1, 5),

(2, 3, 3),

(3, 5, 25),

(4, 7, 30);

**CREATE TABLE `objeto` (**

`codigo\_objeto` int(10) NOT NULL,

`cod\_personaje\_obj` int(10) NOT NULL,

`hora\_cogio` int(10) NOT NULL,

`minuto\_cogio` int(10) NOT NULL,

`segundo` int(10) NOT NULL,

`cod\_escenario\_obj` int(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

**INSERT INTO `objeto`** (`codigo\_objeto`, `cod\_personaje\_obj`, `hora\_cogio`, `minuto\_cogio`, `segundo`, `cod\_escenario\_obj`) VALUES

(1, 1, 8, 30, 25, 1),

(2, 2, 10, 45, 55, 1),

(3, 3, 11, 25, 55, 1),

(4, 4, 12, 30, 30, 2),

(5, 5, 7, 30, 25, 2),

(6, 6, 6, 34, 45, 2),

(7, 7, 6, 55, 34, 2),

(8, 8, 8, 34, 36, 2),

(9, 9, 6, 54, 23, 2);

**CREATE TABLE `personaje` (**

`cod\_personaje` int(10) NOT NULL,

`fuerza\_personaje` int(10) NOT NULL,

`inteligencia\_personaje` int(10) NOT NULL,

`habilidad\_personaje` int(10) NOT NULL,

`cod\_escenario\_per` int(10) NOT NULL,

`cod\_personajeDominado` int(10) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

**INSERT INTO `personaje`** (`cod\_personaje`, `fuerza\_personaje`, `inteligencia\_personaje`, `habilidad\_personaje`, `cod\_escenario\_per`, `cod\_personajeDominado`) VALUES

(1, 2, 4, 6, 1, NULL),

(2, 4, 4, 4, 1, 1),

(3, 5, 5, 5, 1, NULL),

(4, 5, 5, 5, 2, NULL),

(5, 6, 6, 6, 2, 4),

(6, 7, 7, 7, 2, 5),

(7, 8, 8, 8, 2, NULL),

(8, 9, 9, 9, 2, NULL),

(9, 5, 7, 9, 2, 8),

(10, 5, 7, 9, 1, 8);

1. Crear los procedimientos almacenados que permitan consultar, insertar y actualizar

--INSERTAR

use juegoordenador;

delimiter $

create procedure insertarEscenario( in riesgo int, in tiempo int)

begin

insert into escenario(riesgo\_escenario, tiempo\_escenario)values(riesgo, tiempo);

end$

call insertarEscenario(8,9)

Tabla

Descripción generada automáticamente

--CONSULTAR

use juegoordenador;

delimiter $

create procedure ConsultarEscenario()

begin

select \*from escenario;

end$

call consultarEscenario(8,9)

Tabla

Descripción generada automáticamente

-- ACTUALIZAR

use juegoordenador;

delimiter $

create procedure ActualizarEscenario(in id int, in riesgo int, in tiempo int)

begin

update escenario

set riesgo\_escenario=riesgo, tiempo\_escenario=tiempo where cod\_escenario=id;

end$

call ActualizarEscenario(2,3,3)

Tabla

Descripción generada automáticamente con confianza baja

**delimiter $**

**create procedure insertarPersonaje( in fza int, in inteli int,in habili int,in codEsce int,in codPersDomi int)**

begin

insert into personaje(fuerza\_personaje, inteligencia\_personaje,habilidad\_personaje,cod\_escenario, cod\_personajeDominado)

values( fza , inteli , habili,codEsce ,codPersDomi);

end$

call insertarPersonaje(5,5,5,2,9)

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

delimiter $

**create procedure consultarPersonaje()**

begin

select \*from personaje;

end$

delimiter $

**call consultarPersonaje()**

Tabla

Descripción generada automáticamente

**create procedure ActualizarPersonaje(in id int,in fza int, in inteli int,in habili int,in codEsce int,in codPersDomi int)**

begin

update personaje

set fuerza\_personaje=fza,

inteligencia\_personaje=inteli,

habilidad\_personaje=habili,

cod\_escenario=codEsce,

cod\_personajeDominado=codPersDomi

where cod\_personaje=id;

end$

**call ActualizarPersonaje(10,2,2,2,1,9)**

Tabla

Descripción generada automáticamente

1. Proponer 01 función y un disparador (trigger)

**FUNCIONES**

USE juegoordenador

delimiter //

**create function numero\_personaje\_xescenario(codEscenario int) returns int**

**begin**

declare numero int;

select count(\*) into numero from personaje where cod\_escenario\_per =codEscenario ;

return numero;

end//

delimiter ;

select numero\_personaje\_xescenario(2);



**TRIGGER**

**Creamos la tabla auditoria**

**CREATE TABLE `juegoordenador`.`auditoria` (**

`id` INT(11) NOT NULL,

`accion` VARCHAR(200) NULL DEFAULT NULL,

`fecha` DATETIME NULL DEFAULT CURRENT\_TIMESTAMP(),

PRIMARY KEY (`id`));

**Creamos el trigger**

DELIMITER //

create trigger log\_tabla\_personaje after insert on personaje

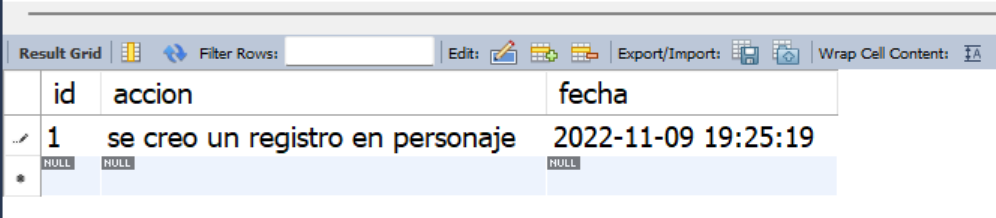
for each row begin

insert into auditoria(accion) value ('se creo un registro en personaje');

end //

delimiter ;

INSERT INTO `juegoordenador`.`personaje` (`cod\_personaje`, `fuerza\_personaje`, `inteligencia\_personaje`, `habilidad\_personaje`, `cod\_escenario\_per`, `cod\_personajeDominado`) VALUES ('10', '5', '7', '9', '1', '8');



1. Generar copia backup y restaura en otro BD, indicar el procedimiento y agregar el script SQL

Ingreso por cmd a mysql



C:\Users\cess>mysql -u root -p

Creamos la carpeta Backups donde guardaremos los respaldos.

Interfaz de usuario gráfica

Descripción generada automáticamente

Ejecuto el comando para crear mi base de datos en una carpeta que he creado en disco C:\Backups

Le asignamos el nombre que va tener nuestro respaldo.: copia\_juegoOrdenador.sql



C:\Users\cesss>mysqldump -u root -p juegoordenador >C :\Backups\copia\_juegoOrdenador.sql

Verificamos

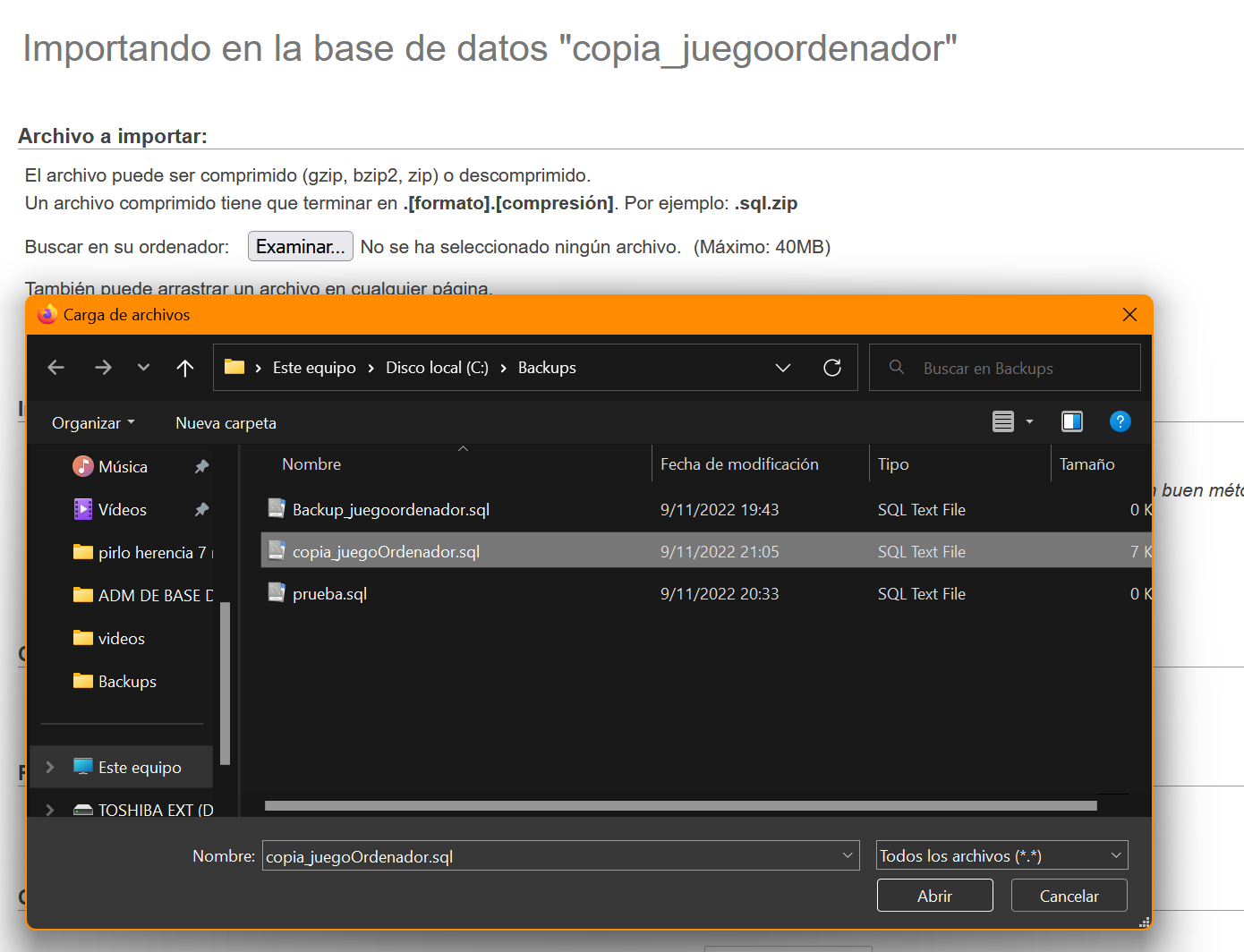
Captura de pantalla de un celular

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Importando en otra base de datos



Se creo correctamente

Diagrama

Descripción generada automáticamente con confianza baja

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

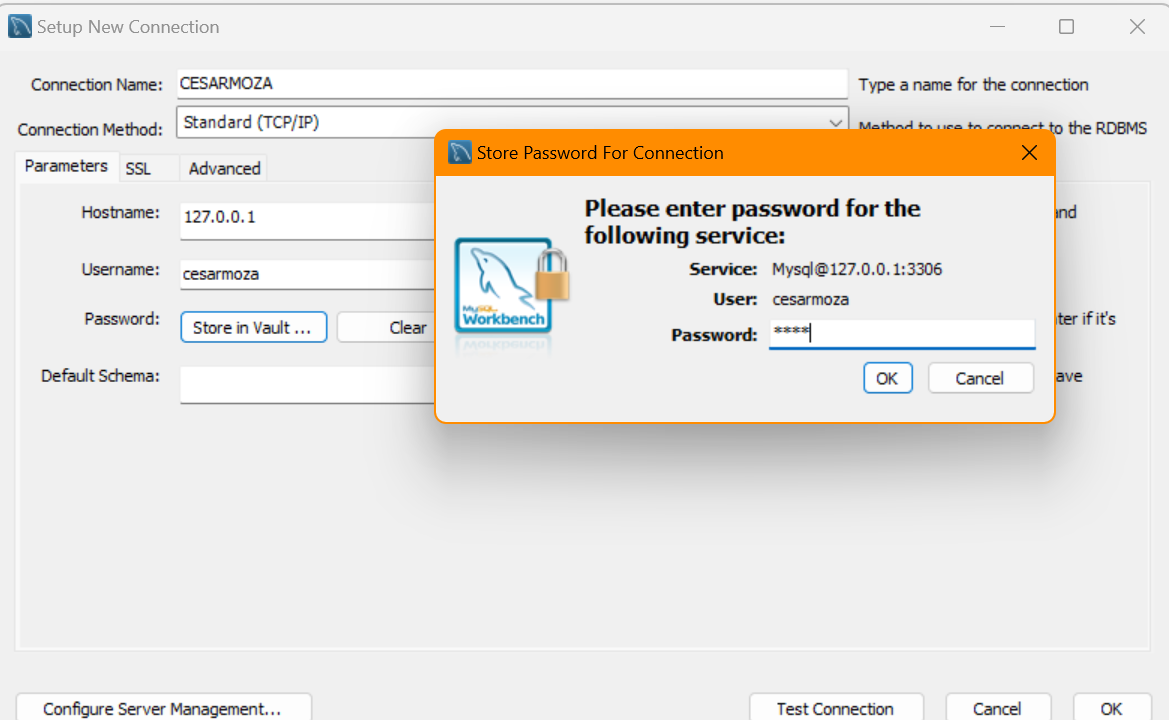
1. Crear usuarios que consulten y ejecuten solo procedimientos almacenados

CREATE USER 'cesarmoza'@'localhost' identified by 'aa11';

grant select on juegoordenador.auditoria to 'cesarmoza'@'localhost' ;

grant select on juegoordenador.personaje to 'cesarmoza'@'localhost' ;

GRANT EXECUTE ON PROCEDURE juegoordenador.insertarPersonaje TO 'cesarmozareyes'@'localhost';



Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

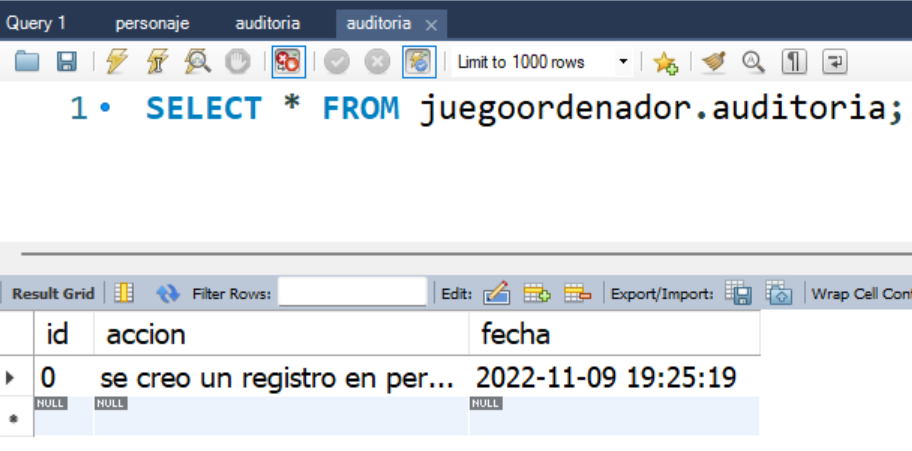
CON EL USARIO cesarmoza creado

SOLO TENGO ACCESO AL PROCEDIMIENTO ALMACENADO INSERTAR PERSONAJE Y A CONSULTAR EN LAS TABLAS AUDITORIA

Y PERSONAJE



SELECT TABLA AUDITORIA



SELECT TABLA PERSONAJE

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Si quiero ejecutar el procedimiento eliminar al que no tengo acceso no me deja

call EliminarPersonaje(10)

