

Miembros del proyecto:
Marcos Morales Márquez
Mario Leal García

Documentación

Breve introducción al documento

En este documento se trata la documentación del proyecto, cabe destacar los apartados Manual de instalación, Manual de usuario y conclusiones cerca del final del documento.

Descripción del contexto y de los requisitos funcionales y no funcionales

Descripción del contexto:

El proyecto trata de una web local que se conecta a una base de datos PL/SQL hecha en SQLDeveloper, la web local está hecha en php y html usando XAMPP.

El usuario/jugador lanza la web local e introduce el nombre y rol del personaje que quiere crear, o puede cargar un personaje que ya existiese y este vivo introduciendo el nombre y continuar la partida, toda la información del juego está en la base de datos y la web local interactúa con la base de datos mediante queries. Una vez creado/cargado el personaje, el usuario puede ver su inventario, leer el manual de enemigos, objetos y escenarios, o ir de aventura a un escenario elegido por el usuario, se muestran en la web la descripción de una serie de eventos, los cuales pueden dañar/curar la salud del jugador, sin poder pasarse de la salud máxima, que es 20, o dar objetos al jugador, cada escenario y cada enemigo tiene 3 objetos como recompensa en un VARRAY, pero a la hora de añadir tal recompensa, solo se añade uno de esos objetos de forma aleatoria, mejorando la capacidad de jugar de nuevo del usuario. Si el jugador pierde toda su salud, pierde la partida y no puede seguir yendo de aventura, se hace visible un botón para volver a crear un personaje.

Requisitos funcionales:

El nombre del jugador es único, con el propósito de poder cargar partidas poniendo el nombre del jugador de tal partida a la hora de crear un personaje.

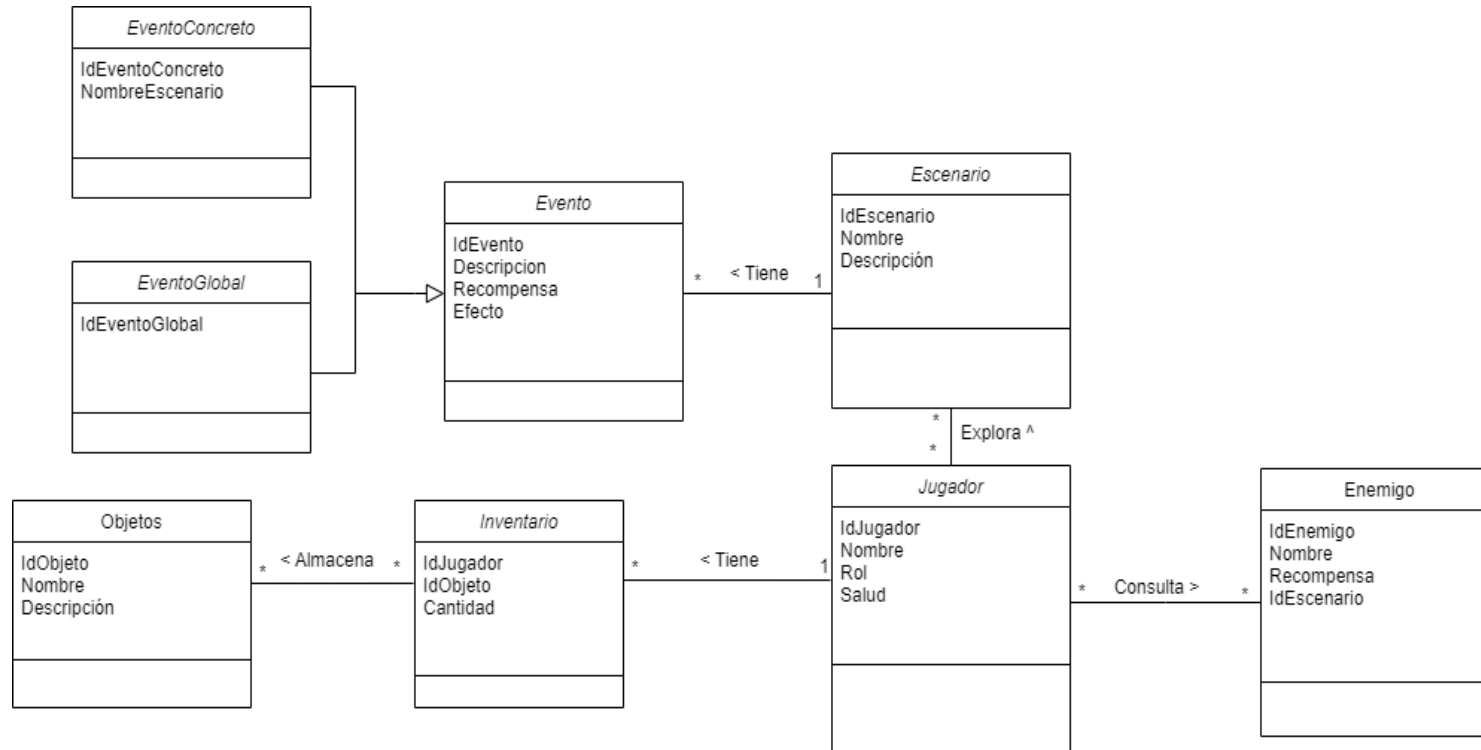
La tabla TablaInventario tiene dos claves foráneas, el IdObjeto de la TablaObjeto y el IdJugador de la TablaJugador.

La tabla TablaEnemigo tiene una clave foránea, IdEscenario de la TablaEscenario.

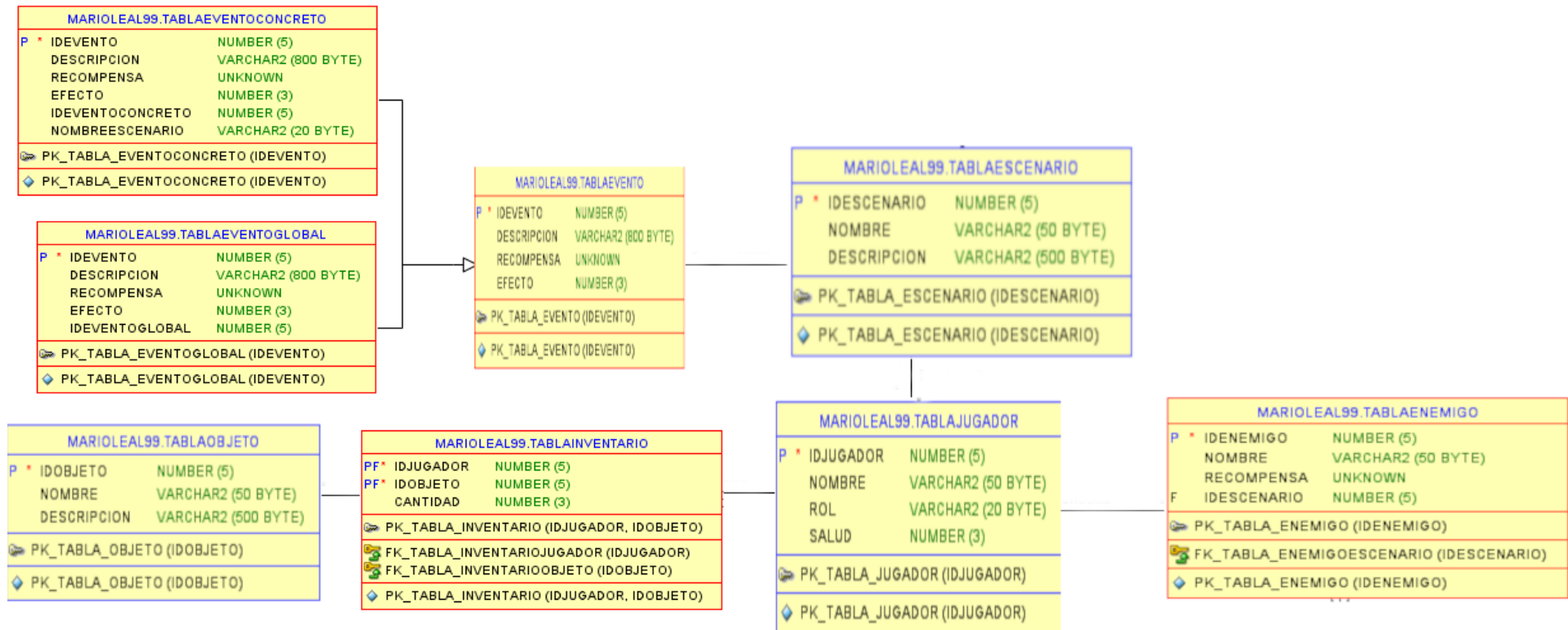
Requisitos no funcionales:

Diseño de la página web, por ejemplo, el color de la salud va cambiando dependiendo de la salud del jugador, el título de cada php dentro de la web muestra la acción que se está realizando, por ejemplo, al pulsar el botón para ver el manual, el título cambia a "Leyendo el manual...", cuando hay un error a la hora de poner el nombre del jugador, como poner un nombre muy largo, se muestra un mensaje de error en rojo. La información de los objetos, enemigos, escenarios, y los eventos al ir de aventura, se muestran dentro de una caja creada mediante <div> y aplicando un estilo a tal div.

Diagrama de clases persistentes



Esquema lógico específico O/



Diseño Físico

```
CREATE OR REPLACE TYPE TipoJugador AS OBJECT(  
  IdJugador NUMBER(5),  
  Nombre VARCHAR2(50),  
  Rol VARCHAR2(20),  
  Salud NUMBER(3)  
);  
/
```

```
CREATE OR REPLACE TYPE TipoObjeto AS OBJECT(  
  IdObjeto NUMBER(5),  
  Nombre VARCHAR2(50),  
  Descripcion VARCHAR2(500)  
);  
/
```

```
CREATE OR REPLACE TYPE TipoInventario AS OBJECT(  
  IdJugador NUMBER(5),  
  IdObjeto NUMBER(5),  
  Cantidad NUMBER(3)  
);  
/
```

```
CREATE OR REPLACE TYPE TipoEscenario AS OBJECT(  
  IdEscenario NUMBER(5),  
  Nombre VARCHAR2(50),  
  Descripcion VARCHAR2(500)  
);  
/
```

```
CREATE OR REPLACE TYPE Tipo_Recompensa AS VARRAY(3) OF NUMBER;  
/
```

```
CREATE OR REPLACE TYPE TipoEnemigo AS OBJECT(  
  IdEnemigo NUMBER(5),  
  Nombre VARCHAR2(50),  
  Recompensa Tipo_Recompensa,  
  IdEscenario NUMBER(5)  
);  
/
```

```
CREATE OR REPLACE TYPE TipoEvento AS OBJECT(  
  IdEvento NUMBER(5),  
  Descripcion VARCHAR2(800),  
  Recompensa Tipo_Recompensa,  
  Efecto NUMBER(3)
```

Documentación del proyecto de TABD.

```
) NOT FINAL;  
/
```

```
CREATE OR REPLACE TYPE TipoEventoConcreto UNDER TipoEvento (  
  IdEventoConcreto NUMBER(5),  
  NombreEscenario VARCHAR2(20)  
);  
/
```

```
CREATE OR REPLACE TYPE TipoEventoGlobal UNDER TipoEvento (  
  IdEventoGlobal NUMBER(5)  
);  
/
```

```
CREATE TABLE TablaJugador OF TipoJugador(  
  constraint PK_Tabla_Jugador PRIMARY KEY(IdJugador),  
  constraint UQ_Tabla_Jugador_Nombre UNIQUE(Nombre));  
/
```

```
CREATE TABLE TablaObjeto OF TipoObjeto(  
  constraint PK_Tabla_Objeto PRIMARY KEY(IdObjeto));
```

```
CREATE TABLE TablaEscenario OF TipoEscenario(  
  constraint PK_Tabla_Escenario PRIMARY KEY(IdEscenario));
```

```
CREATE TABLE TablaEnemigo OF TipoEnemigo(  
  constraint PK_Tabla_Enemigo PRIMARY KEY(IdEnemigo),  
  constraint FK_Tabla_EnemigoEscenario FOREIGN KEY(IdEscenario) REFERENCES  
  TablaEscenario(IdEscenario));
```

```
CREATE TABLE TablaEvento OF TipoEvento(  
  constraint PK_Tabla_Evento PRIMARY KEY(IdEvento));
```

```
CREATE TABLE TablaEventoConcreto OF TipoEventoConcreto(  
  constraint PK_Tabla_EventoConcreto PRIMARY KEY(IdEvento));
```

```
CREATE TABLE TablaEventoGlobal OF TipoEventoGlobal(  
  constraint PK_Tabla_EventoGlobal PRIMARY KEY(IdEvento));
```

```
CREATE TABLE TablaInventario OF TipoInventario(  
  constraint PK_Tabla_Inventario PRIMARY KEY(IdJugador, IdObjeto),  
  constraint FK_Tabla_InventarioJugador FOREIGN KEY(IdJugador) REFERENCES  
  TablaJugador(IdJugador),  
  constraint FK_Tabla_InventarioObjeto FOREIGN KEY(IdObjeto) REFERENCES  
  TablaObjeto(IdObjeto)  
)
```

Disparadores y paquete

Nota: en el código de la web, no siempre llamamos a una función para interactuar con la base de datos si la consulta es pequeña, por ejemplo, para ver la información de los escenarios en el manual, con una query que mandase un SELECT de los datos necesarios era suficiente.

```
/*Al actualizar el rol del jugador, añadirle el equipamiento correspondiente */
CREATE OR REPLACE TRIGGER Disparador_Rol
AFTER INSERT ON TablaJugador
FOR EACH ROW

BEGIN

IF (:NEW.rol = 'Caballero' ) THEN
    /* Items de caballero, espada de hierro, escudo de madera y armadura pesada */
    INSERT INTO TablaInventario (IdJugador, IdObjeto, Cantidad)
    VALUES (Seq_Jugador.CURRVAL, 2, 1);
    INSERT INTO TablaInventario (IdJugador, IdObjeto, Cantidad)
    VALUES (Seq_Jugador.CURRVAL, 5 , 1);
    INSERT INTO TablaInventario (IdJugador, IdObjeto, Cantidad)
    VALUES (Seq_Jugador.CURRVAL, 8 , 1);

ELSIF (:NEW.rol = 'Arquero' ) THEN
    /* Items de arquero, arco y armadura media */
    INSERT INTO TablaInventario (IdJugador, IdObjeto, Cantidad)
    VALUES (Seq_Jugador.CURRVAL, 3, 1);
    INSERT INTO TablaInventario (IdJugador, IdObjeto, Cantidad)
    VALUES (Seq_Jugador.CURRVAL, 6, 1);

ELSIF (:NEW.rol = 'Mago' ) THEN
    /* Items de mago, bastón y armadura ligera */
    INSERT INTO TablaInventario (IdJugador, IdObjeto, Cantidad)
    VALUES (Seq_Jugador.CURRVAL, 4, 1);
    INSERT INTO TablaInventario (IdJugador, IdObjeto, Cantidad)
    VALUES (Seq_Jugador.CURRVAL, 7, 1);
END IF;
END Disparador_Rol;

/* Paquete */

CREATE OR REPLACE PACKAGE PaqueteProyecto AS

    FUNCTION FuncionEvento(Nombre_ IN TablaEscenario.Nombre%TYPE, IdJugador_ IN
    TablaJugador.IdJugador%TYPE) RETURN TablaEvento.Descripcion%type;
    FUNCTION FuncionNombreRol(Nombre_ IN TablaJugador.Nombre%TYPE, Rol_ IN
    TablaJugador.Rol%TYPE ) RETURN TablaJugador.IdJugador%TYPE;
```

Documentación del proyecto de TABD.

```
END PaqueteProyecto;
```

```
/
```

```
/* Código */
```

```
CREATE OR REPLACE PACKAGE BODY PaqueteProyecto AS
```

```
FUNCTION FuncionEvento (
```

```
Nombre_ IN TablaEscenario.Nombre%TYPE,
```

```
IdJugador_ IN TablaJugador.IdJugador%TYPE)
```

```
RETURN TablaEvento.Descripcion%TYPE IS
```

```
Res VARCHAR2(800);
```

```
numero_aleatorio NUMBER;
```

```
total_filas NUMBER;
```

```
efecto_ NUMBER;
```

```
recompensa_ TablaEventoGlobal.Recompensa%TYPE;
```

```
existe_ NUMBER;
```

```
v_IdEvento NUMBER;
```

```
v_contador NUMBER;
```

```
saludActual_ NUMBER;
```

```
CURSOR eventosconcretos IS SELECT IdEventoConcreto FROM TablaEventoConcreto
```

```
WHERE NombreEscenario = Nombre_;
```

```
BEGIN
```

```
/* Generar un número aleatorio entre 0 y 100: */
```

```
numero_aleatorio := FLOOR(DBMS_RANDOM.VALUE(0, 101));
```

```
/* Si el número es menor que 75, entonces mostramos un evento concreto,
```

```
sino, uno global: */
```

```
IF (numero_aleatorio < 75) THEN
```

```
/* Obtener el número de filas: */
```

```
SELECT COUNT(*) INTO total_filas FROM TablaEventoConcreto
```

```
WHERE NombreEscenario = Nombre_;
```

```
/* Generar un número aleatorio entre 1 y total_filas: */
```

```
/* +1 para incluir la última fila */
```

```
numero_aleatorio := FLOOR(DBMS_RANDOM.VALUE(1, total_filas+1));
```

```
v_contador := 1;
```

```
OPEN eventosconcretos;
```

```
LOOP
```

```
FETCH eventosconcretos INTO v_IdEvento;
```

```
IF(v_contador=numero_aleatorio) THEN
```

```
EXIT;
```

```
END IF;
```

```
v_contador:=v_contador+1;
```

```
EXIT WHEN eventosconcretos%NOTFOUND;
```

```
END LOOP;
```

```
CLOSE eventosconcretos;
```

```
/* Obtener la descripcion del evento con ID = numero_aleatorio: */
```

```
SELECT Descripcion, Efecto, Recompensa INTO res, efecto_, recompensa_
```

```
FROM TablaEventoConcreto
WHERE IdEventoConcreto = v_IdEvento;
ELSE
  /* Obtener el número de filas: */
  SELECT COUNT(*) INTO total_filas FROM TablaEventoGlobal;
  /* Generar un número aleatorio entre 1 y total_filas: */
  /* +1 para incluir la última fila */
  numero_aleatorio := FLOOR(DBMS_RANDOM.VALUE(1, total_filas+1));

  /* Obtener la descripción del evento con ID = numero_aleatorio: */
  SELECT Descripción, Efecto, Recompensa INTO res, efecto_, recompensa_
  FROM TablaEventoGlobal
  WHERE IdEventoGlobal = numero_aleatorio;
END IF;

/* ##### */

/* Actualizar salud y añadir recompensa en caso de que haya: */
IF (efecto_ != 0) THEN
  SELECT Salud INTO saludActual_ FROM TablaJugador
  WHERE IdJugador = IdJugador_;
  IF saludActual_ + efecto_ < 20 THEN
    UPDATE TablaJugador SET Salud = Salud + efecto_
    WHERE IdJugador = IdJugador_;
  END IF;
END IF;

/* Generar un número aleatorio entre 1 y número de elementos del VARRAY: */
/* +1 para incluir la última fila */
numero_aleatorio := FLOOR(DBMS_RANDOM.VALUE(1, recompensa_.LAST+1));

/* Comprobamos si ya existe tal objeto en TablaInventario */
SELECT COUNT(*) INTO existe_ FROM TablaInventario
WHERE IdObjeto = recompensa_(numero_aleatorio) AND IdJugador = IdJugador_;
/* Si la recompensa no es 0, y no existe el objeto en la tabla inventario, insertar el objeto:
*/
IF (recompensa_(numero_aleatorio) != 0 AND existe_ = 0)
THEN
  INSERT INTO TablaInventario (IdJugador, IdObjeto, Cantidad)
  VALUES (IdJugador_, recompensa_(numero_aleatorio), 1);
/* Si ya existe, aumentar la cantidad en 1: */
ELSIF (recompensa_(numero_aleatorio) != 0 AND existe_ != 0) THEN
  UPDATE TablaInventario SET Cantidad = Cantidad + 1
  WHERE IdObjeto = recompensa_(numero_aleatorio);
END IF;
RETURN Res;
END;
```


Documentación del proyecto de TABD.

```
FUNCTION FuncionNombreRol (
  Nombre_ IN TablaJugador.Nombre%TYPE,
  Rol_ IN TablaJugador.Rol%TYPE )
RETURN TablaJugador.IdJugador%TYPE IS
  res TablaJugador.IdJugador%TYPE;
  salud_ TablaJugador.Salud%TYPE;
BEGIN
  BEGIN
    /* Obtenemos el IdJugador y la salud de la partida con nombre = nombre_ */
    SELECT IdJugador, Salud INTO res, salud_
    FROM TablaJugador
    WHERE Nombre = Nombre_;

    /* El nombre ya existe, comprobamos si el jugador esta vivo,
    si esta muerto, devolvemos 0, la web se encarga de tratarlo y mostrar
    un mensaje de error */
    IF salud_ <= 0 THEN
      res := 0;
    END IF;
  EXCEPTION
    /* El nombre no existía */
    WHEN NO_DATA_FOUND THEN
      INSERT INTO TablaJugador(IdJugador, Nombre, Rol, Salud)
      VALUES (Seq_Jugador.NEXTVAL, Nombre_, Rol_, 20);

      /* Guardar en res el valor de IdJugador */
      SELECT IdJugador INTO res
      FROM TablaJugador
      WHERE Nombre = Nombre_;
  END;

  RETURN res;
END;
END PaqueteProyecto;
/
```

Manual de instalación



Para poder probar el proyecto, hace falta descargar la versión 8.2.4 para windows de XAMPP e Instant Client para microsoft windows x64:

<https://www.apachefriends.org/download.html>

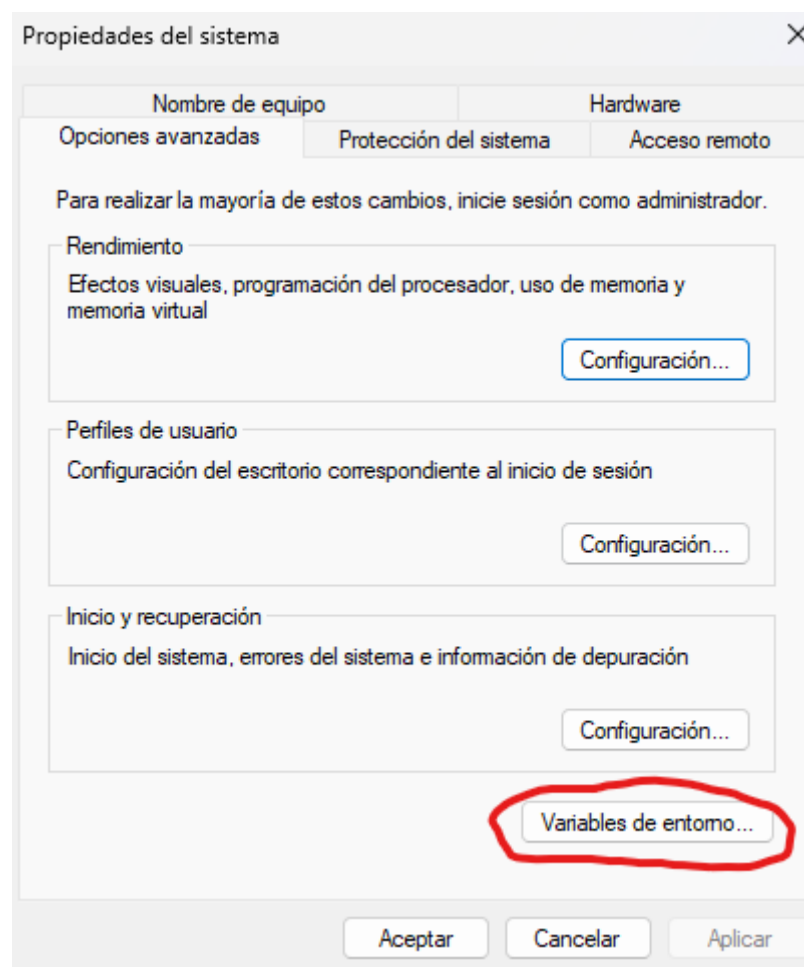
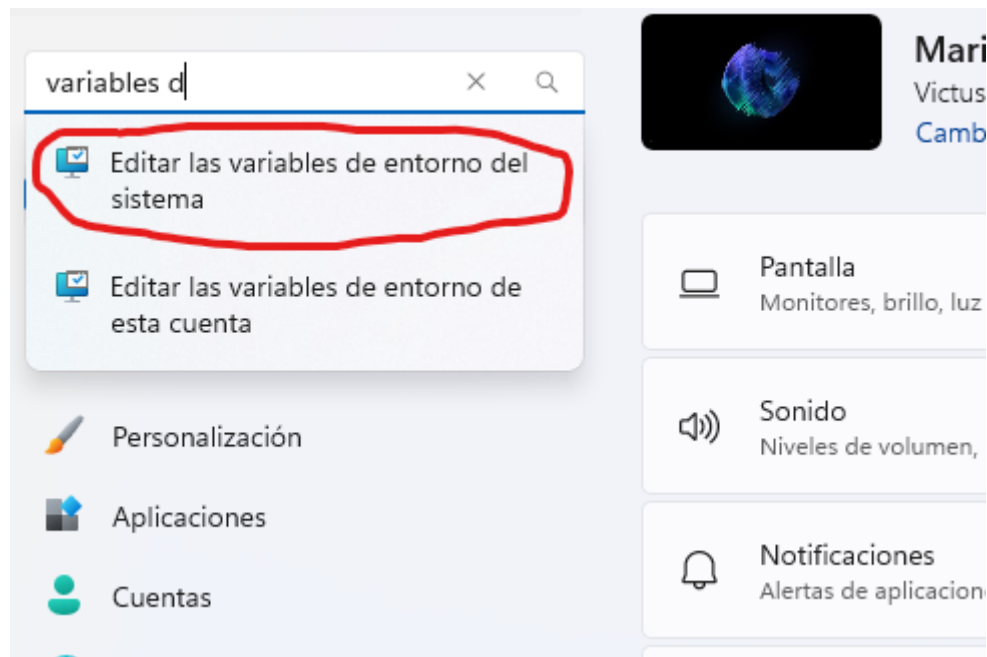
<https://www.oracle.com/database/technologies/instant-client/winx64-64-downloads.html>

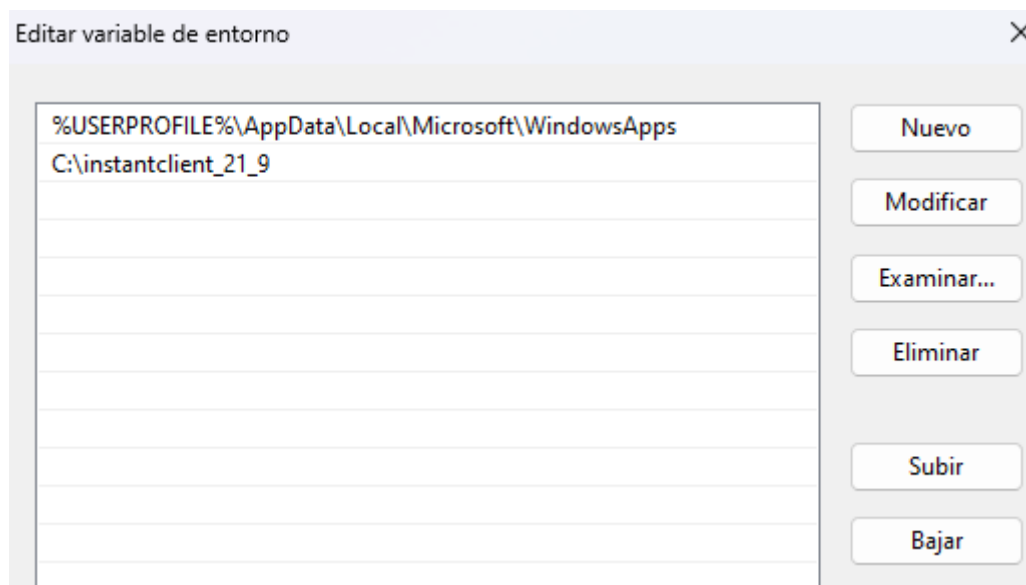
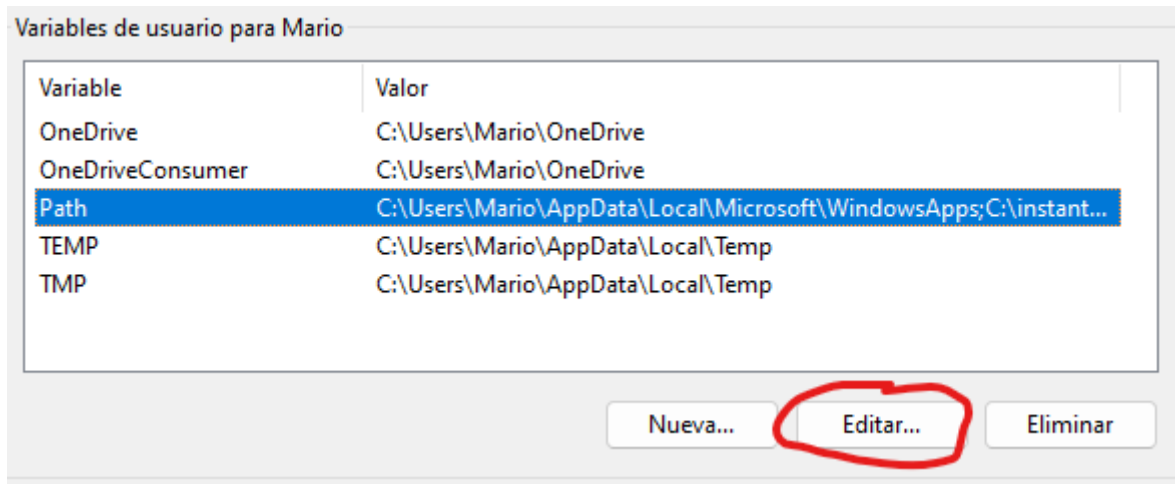
Una vez descargado XAMPP, ha de instalarse usando el wizard proporcionado por la descarga y dejar los valores de la instalación por defecto.

En el enlace de Instant Client, descargue el paquete básico y el paquete ODBC de la versión 21.9.0.0.0

Version 21.9.0.0.0		
Base - one of these packages is required		
Name	Download	Description
Basic Package	 instantclient-basic-windows.x64-21.9.0.0.0dbru.zip	<p>All files required to run OCI, OCCI, and JDBC:OCI applications (83,142,984 bytes) (cksum - 4043771063)</p> <p>Review the Operating System Checklist for Oracle Database Client Installation. Note Windows 7 is not supported.</p> <p>The 21c Basic package requires the Microsoft Visual Studio 2017 Redistributable.</p>
ODBC Package	 instantclient-odbc-windows.x64-21.9.0.0.0dbru.zip	<p>Additional libraries for enabling ODBC applications (1,029,085 bytes) (cksum - 902195586)</p>

Una vez descargado Instant Client, extraiga ambos zips en algún lugar del dispositivo, nosotros lo pusimos al inicio de nuestro disco C:, a continuación, ha de añadirse la ruta a la carpeta instantclient_21_9 al PATH del sistema, para hacer esto, en la configuración de windows, se busca editar las variables del entorno y saldrá una ventana, hay que pulsar el botón variables del entorno, pulsar PATH y editar, en la nueva ventana, pulsar new y ponerla ruta a instantclient_21_9, en nuestro caso, C:\instantclient_21_9.





Ahí no te aparecerá C:\instantclient_21_9 así que tienes que darle a Nuevo y escribir la ruta.












En la carpeta de XAMPP, dentro de la carpeta php, para habilitar la extensión oci8 que permite la conexión a una base de datos Oracle PL/SQL desde PHP, ha de descomentar o activar las siguientes líneas en el archivo php.ini:

Busca la línea ;extension=oci8 y elimina el punto y coma ; al inicio para descomentarla.

Asegúrate de que también está habilitada la extensión extension=php_oci8.dll para sistemas Window. Guarde los cambios en el archivo php.ini y reinicie el servidor Apache en XAMPP para que los cambios surtan efecto, en caso de que estuviese activo.

Documentación del proyecto de TABD.

Este equipo > Windows (C:) > xampp > php >

Nombre	Fecha de modificación	Tipo	Tamaño
 pci	27/01/2016 12:02	JetBrains PhpStorm	2 KB
 pciconf	27/04/2023 14:18	Archivo	18 KB
 pciconf	27/04/2023 14:18	Archivo por lotes ...	1 KB
 pear	27/04/2023 14:18	Archivo por lotes ...	5 KB
 peardev	27/04/2023 14:18	Archivo por lotes ...	5 KB
 pecl	27/04/2023 14:18	Archivo por lotes ...	5 KB
 phar.phar	14/03/2023 19:30	Archivo por lotes ...	1 KB
 pharcommand.phar	14/03/2023 19:30	Archivo PHAR	65 KB
 php	14/03/2023 19:30	Aplicación	139 KB
 php	27/04/2023 14:22	Opciones de confi...	76 KB
 php.ini-development	14/03/2023 19:30	Archivo INI-DEVEL...	74 KB

```
php: Bloc de notas
Archivo  Editar  Ver

; otherwise it results in segfault when unloading after using SASL.
; See https://github.com/php/php-src/issues/8620 for more info.
;extension=ldap

extension=curl
;extension=ffi
;extension=ftp
extension=fileinfo
;extension=gd
extension=gettext
;extension=gmp
;extension=intl
;extension=imap
extension=mbstring
extension=exif      ; Must be after mbstring as it depends on it
extension=mysqli
extension=oci8_12c  ; Use with Oracle Database 12c Instant Client
extension=oci8_19  ; Use with Oracle Database 19 Instant Client
extension=odbc
;extension=openssl
extension=pdo_firebird
extension=pdo_mysql
;extension=pdo_oci
;extension=pdo_odbc
;extension=pdo_pgsql
extension=pdo_sqlite
;extension=pgsql
;extension=shmop

; The MIBS data available in the PHP distribution must be installed.
; See https://www.php.net/manual/en/snmp.installation.php
;extension=snmp

;extension=soap
;extension=sockets
;extension=sodium
;extension=sqlite3
;extension=tidy
;extension=xml
;extension=zip

;zend_extension=opcache

; Module Settings ;
;::::::::::::::::::;
asp_tags=Off
display_startup_errors=On
track_errors=Off
zend.assertions=Off

Ln 941, Col 15
```

En la entrega del proyecto, hay un zip llamado CodigoFuenteBaseDatos con los .sql usados en la base de datos, y otro zip llamado CodigoFuenteWeb que contiene una carpeta llamada htdocs, la cual tiene dentro nuestro proyecto y los .php usados, al descomprimir, uno de esos .php es conexion.php, dentro tiene tres variables, que son el usuario,

contraseña y la cadena de acceso a su base de datos, sustituya el valor actual por el de su base de datos y guarde el fichero.

Mueva o copie la carpeta htdocs a su carpeta XAMPP, si le pide sobrescribir, confirme.

Dentro de la carpeta XAMPP, inicie xampp-control en modo administrador y empiece el módulo apache.

NOTA: Es posible que la primera vez que use Apache tenga que darle al recuadro de la izquierda(Debajo de Modules Service) para activarlo antes de darle a start.



A continuación, descomprima el zip llamadoCodigoFuenteBaseDatos que contiene los .sql, y ejecute el archivo 7Script.sql en SQLDeveloper, esto ejecutará el resto de sqls en su base de datos.

En su navegador web, escriba <http://localhost/ProyectoTABD>, debería poder ver un menú para la creación del personaje.

Si no es así, no dude en contactar con nosotros desde el campus de la asignatura.

Manual de usuario

Si se han seguido los pasos anteriores correctamente, en el último paso, debería poder ver un menú para la creación del personaje, puede escribir el nombre que quiera mientras no sea mayor de 50 caracteres, si escriba el nombre de un personaje existente que tiene salud mayor que 0, podrá seguir esa partida, si la salud es menor o igual que 0, o no se introduce un nombre, se mostrará un mensaje de error. Una vez creado o cargado un personaje, se mostrará un nuevo menú con varios botones, el botón Ver el manual mostrará un nuevo menú con varios botones, los cuales muestran información sobre los enemigos, los objetos y los escenarios, si pulsa el botón Volver, volverá al menú posterior a la creación del personaje, si pulsa el botón "Ver tu inventario" se mostrarán los objetos que estén en su inventario, debería ver al menos dos, dependiendo del rol que seleccionaste al principio, al seleccionar rol, un disparador inserta objetos en el inventario dependiendo del rol. En el menú anterior, puede seleccionar entre 4 zonas a explorar y entonces pulsar el botón Ir de aventura, esto mostrará la descripción de 5 eventos, los cuales dañan o sanan al jugador, o añaden un objeto a su inventario, o ambas, puede ver su salud en todo momento, dependiendo de la salud, el color del texto cambia, si su salud es 0, ha sido derrotado y no puede continuar explorando, esto también lo puede ver en el menú principal, donde además de no poder explorar más, ahora es visible un botón para empezar de nuevo.

Conclusiones

En conclusión, estamos bastante contentos con el resultado del trabajo. Hemos aprendido mucho sobre el diseño de las bases de datos, la forma de conectarlo a una aplicación o página web y cómo diseñar una página web. Pensamos que elegir un videojuego como enfoque del proyecto ha sido una idea bastante original y divertida, con un desarrollo un poco complicado pero un resultado satisfactorio.

Referencias bibliográficas

<https://www.ionos.es/digitalguide/servidores/herramientas/instala-tu-servidor-local-xampp-en-unos-pocos-pasos/>

<https://www.ionos.es/digitalguide/paginas-web/creacion-de-paginas-web/tutorial-de-php-fundamentos-basicos-para-principiantes/>

https://www.youtube.com/watch?v=ofKb3u4piHc&ab_channel=TelmoSampaio