



# GUÍA PRÁCTICA GRUPAL

LM - SI

13/02/2020

# Contenido

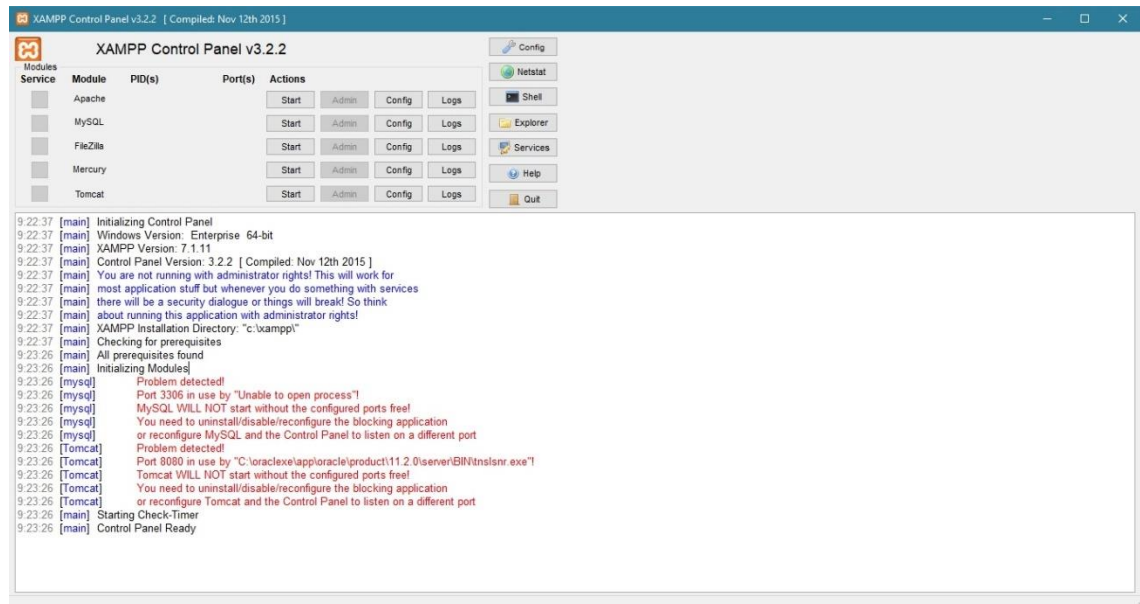
1.	Instalación de programas .....	1
1.1	Xampp .....	1
1.2	Oracle 11g Express Edition.....	6
1.3	SQL Developer.....	6
1.4	Comprobación.....	12
2.	Funcionamiento PL/SQL (para la semana del 17 al 23) .....	13
3.	Funcionamiento PHP (para la semana del 17 al 23) .....	17

# 1. Instalación de programas

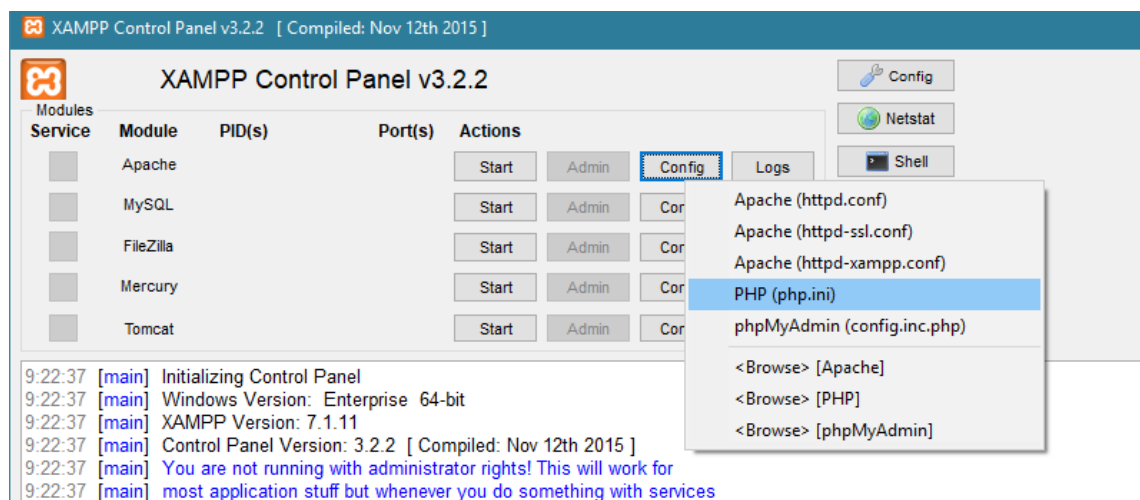
## 1.1 Xampp

Instalamos la última versión que haya en su web.

Una vez instalado debería aparecer esto:

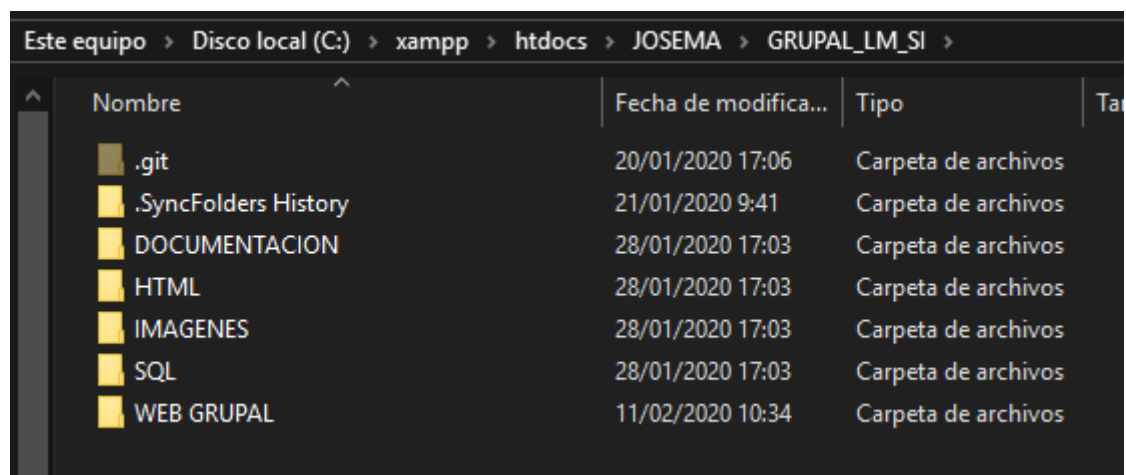
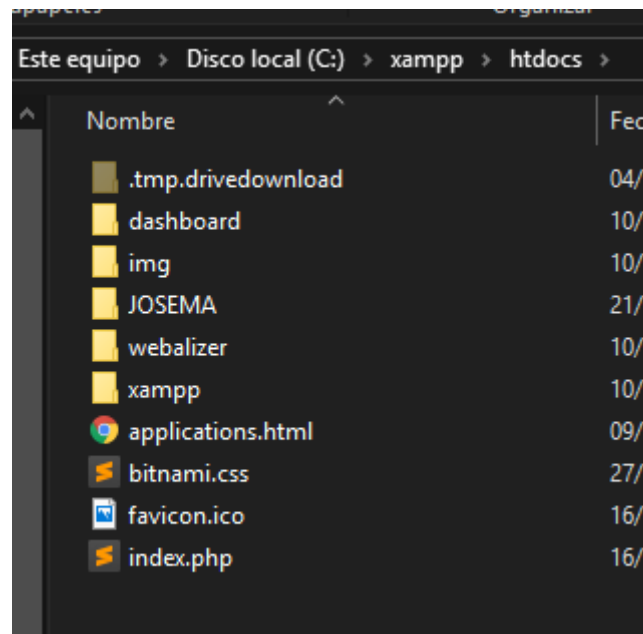


Ahora le damos a “config” y a “php.ini”





Si nos vamos a “C:” deberíamos tener una carpeta “xampp”. Dentro de “xampp” debemos tener una carpeta “htdocs”. Pues dentro de “htdocs” copiamos las carpetas que tengamos en el repositorio de GitKraken. **Cada vez que modifiquemos algo aquí hay que acordarse de copiarlo a la carpeta de GitKraken y hacer el PUSH (Primero hacedme el PULL mamones, que si no podemos machacar lo de otra persona).**

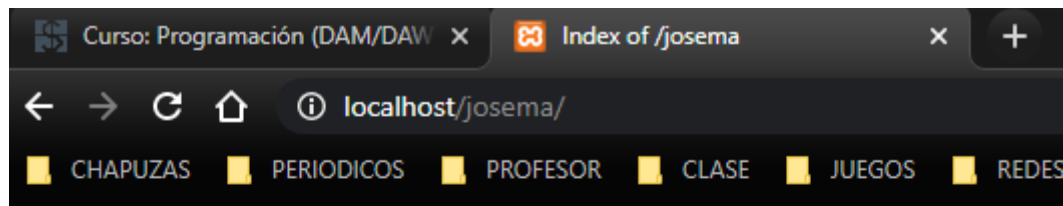


Fijaos en la ruta.

**Consejo:** Crearos un acceso directo desde el escritorio a esa carpeta antes de que queráis matar a alguien por tener que entrar cada vez hasta ahí.

Ahora en Chrome abrimos la dirección “http://localhost/...” donde “...” es la dirección de la carpeta una vez entrados en “htdocs”, para que os hagáis una idea, mi localhost es “http://localhost/josema/GRUPAL\_LM\_SI/WEB%20GRUPAL/”.

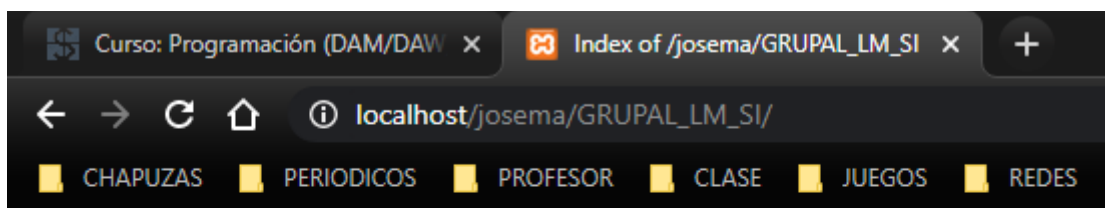
Si preferís, una vez que entras en la primera carpeta puedes ir navegando por las carpetas que tengas ahí guardadas.



## Index of /josema

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
<a href="#">Parent Directory</a>		-	
<a href="#">GRUPAL_LM_SI/</a>	2020-01-28 17:03	-	
<a href="#">IISSI/</a>	2018-09-12 20:00	-	
<a href="#">LIBROS/</a>	2017-03-01 11:22	-	

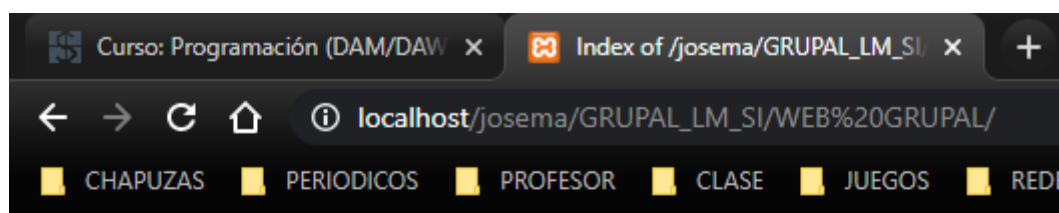
Apache/2.4.29 (Win32) OpenSSL/1.0.2l PHP/7.1.11 Server at localhost Port 80




















## Index of /josema/GRUPAL\_LM\_SI

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
<a href="#">Parent Directory</a>		-	
<a href="#">DOCUMENTACION/</a>	2020-01-28 17:03	-	
<a href="#">HTML/</a>	2020-01-28 17:03	-	
<a href="#">IMAGENES/</a>	2020-01-28 17:03	-	
<a href="#">SQL/</a>	2020-01-28 17:03	-	
<a href="#">WEB GRUPAL/</a>	2020-02-11 10:34	-	

Apache/2.4.29 (Win32) OpenSSL/1.0.2l PHP/7.1.11 Server at localhost Port 80



## Index of /josema/GRUPAL\_LM\_SI/W

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 <a href="#">Parent Directory</a>		-	
 <a href="#">accionAltaUsuario.php</a>	2020-02-11 09:45	6.4K	
 <a href="#">accionBorrarArticulo..&gt;</a>	2020-01-28 21:44	636	
 <a href="#">accionModificarArtic..&gt;</a>	2020-02-04 11:02	1.6K	
 <a href="#">consultaArticulos.php</a>	2020-01-29 13:00	4.7K	
 <a href="#">controladorArticulos..&gt;</a>	2020-01-29 12:32	678	
 <a href="#">css/</a>	2020-01-28 17:03	-	
 <a href="#">excepcion.php</a>	2020-01-28 17:47	1.1K	
 <a href="#">exitAltaUsuario.php</a>	2020-02-11 09:40	1.0K	
 <a href="#">formAltaUsuario.php</a>	2020-02-11 09:45	4.8K	
 <a href="#">gestionBD.php</a>	2020-01-23 14:18	643	
 <a href="#">gestionarArticulos.php</a>	2020-02-04 11:02	1.3K	
 <a href="#">gestionarUsuarios.php</a>	2020-02-11 09:47	2.3K	
 <a href="#">inicio.html</a>	2020-01-28 14:24	2.5K	
 <a href="#">js/</a>	2020-02-04 13:27	-	
 <a href="#">paginacion_consulta.php</a>	2018-06-03 20:41	1.0K	
 <a href="#">styles/</a>	2020-01-28 17:03	-	

Apache/2.4.29 (Win32) OpenSSL/1.0.2l PHP/7.1.11 Server at localhost Port 80

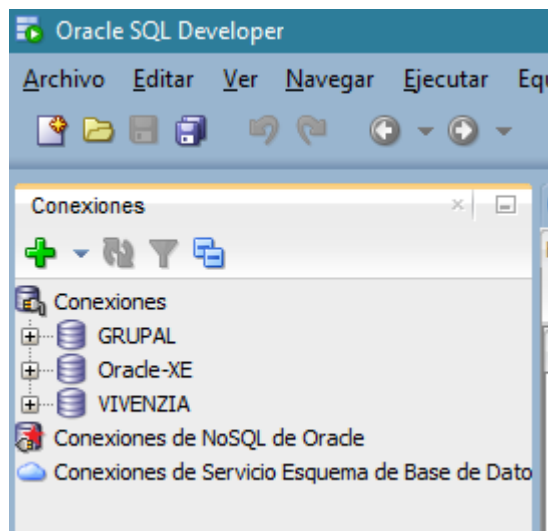
## 1.2 Oracle 11g Express Edition

Descargamos del enlace de Mega que hay en Discord.


De Oracle no tengo pantallazos (y no me voy a poner a desinstalarlo por lo que pueda pasar XD). Lo que si hay que tener claro al instalar es que **HAY QUE GUARDAR LA CONTRASEÑA QUE PONGAMOS**. Si la olvidas hay que desinstalar (se necesita para SQL Developer).

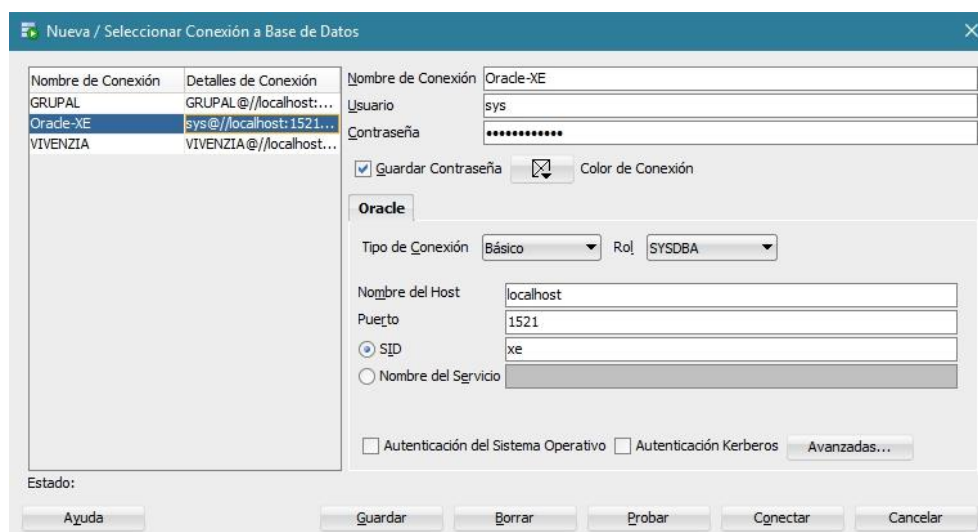
## 1.3 SQL Developer

En principio es un ejecutable, así que pasamos directamente a configurar.



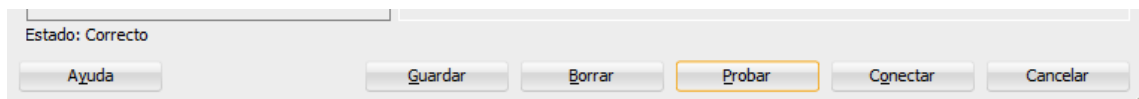
Debería apareceros esta pantalla, pero con la sección de “Conexiones” vacía.

Le damos al  y configuramos la siguiente pestaña que nos aparece (puede variar un poco según la versión. De nuevo deberíais tener vacía la columna de la izquierda. Si os aparece alguna opción que aquí no esté dejadla por defecto. **La contraseña debe ser la que se ha puesto al instalar Oracle 11g.**

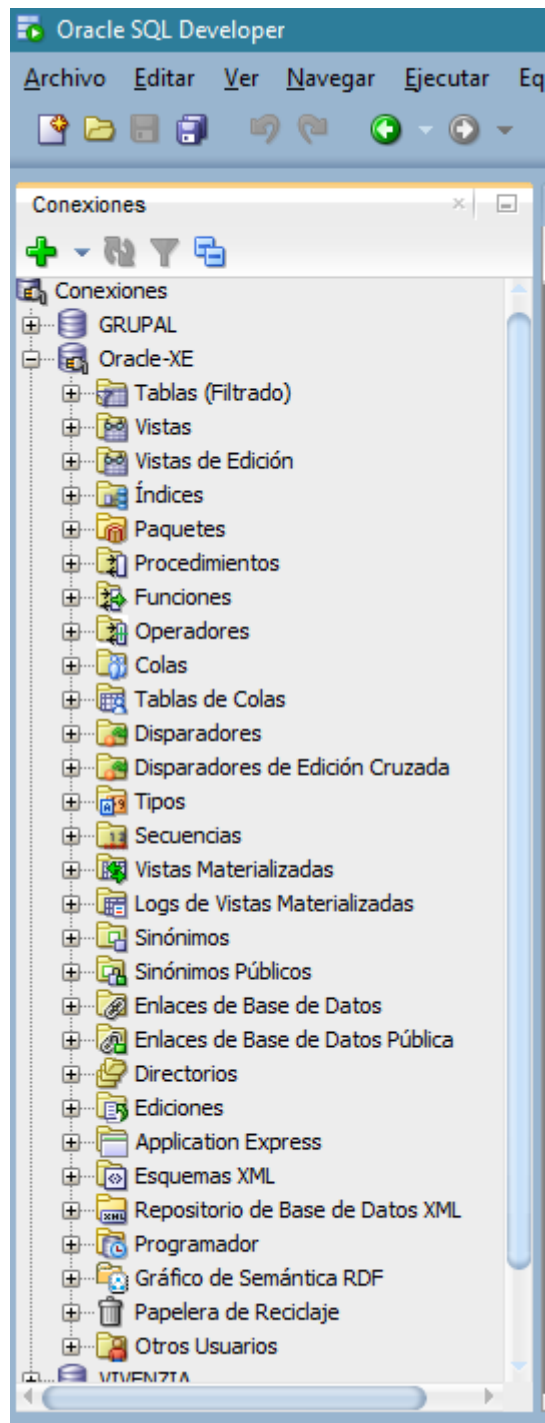




Antes de darle a nada mas le dais a probar.

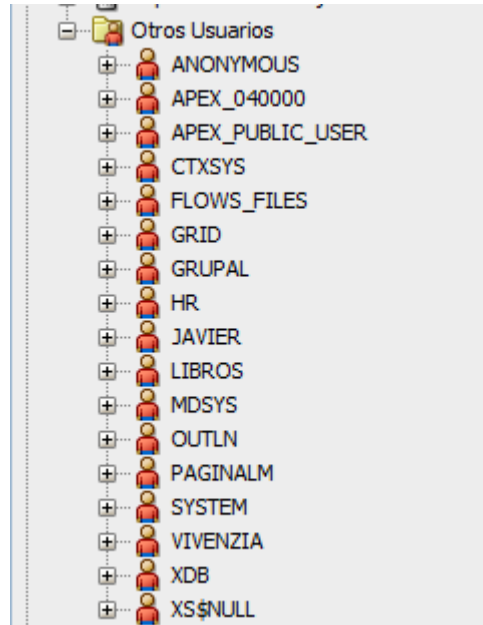


Si aparece “Estado: Correcto” perfecto, si no hay que mirar uno a uno (Que algo habréis hecho mal XD). Una vez correcto pulsamos en conectar. Y debería aparecernos la sección “Oracle-XE” dentro de conexiones (Sólo esa, paciencia).



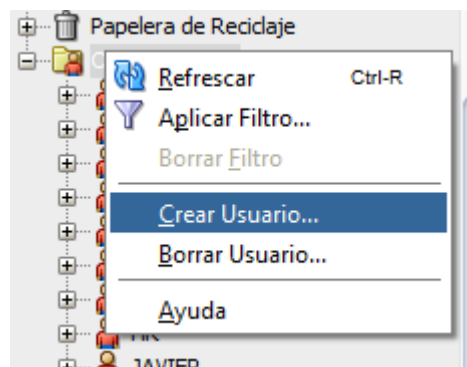
Ahora abrimos la sección de “Otros Usuarios” de dentro de “Oracle-XE”.

Esto habrá que hacerlo cada vez que quieras crear una base de datos nueva.

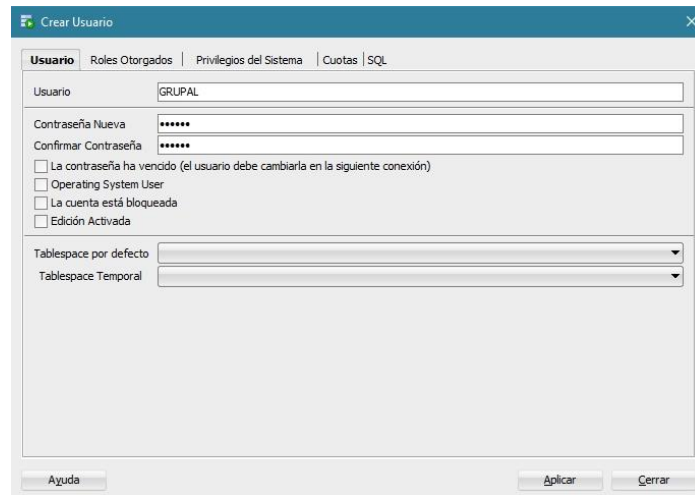


Si os aparecen mas o menos que a mi es por la versión. Si os fijáis tengo varias creadas por mi (“VIVENZIA”, “LIBROS”, “JAVIER”, “PAGINALM”, “GRUPAL”). Pues vamos a crear “GRUPAL”.

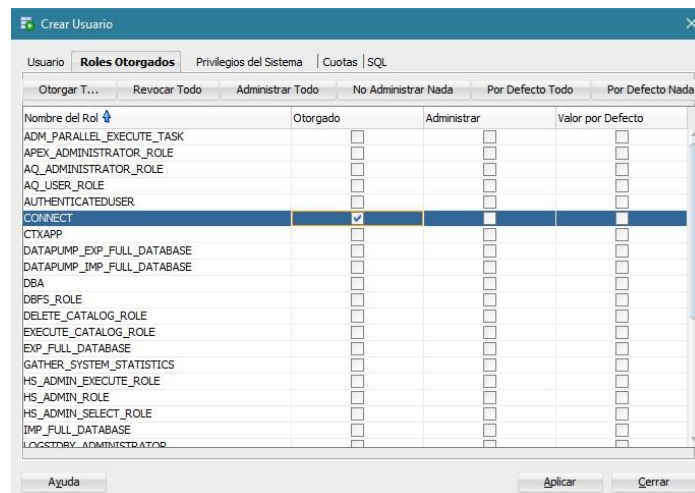
Hacemos click derecho sobre “Otros Usuarios” y a “Crear Usuario...”



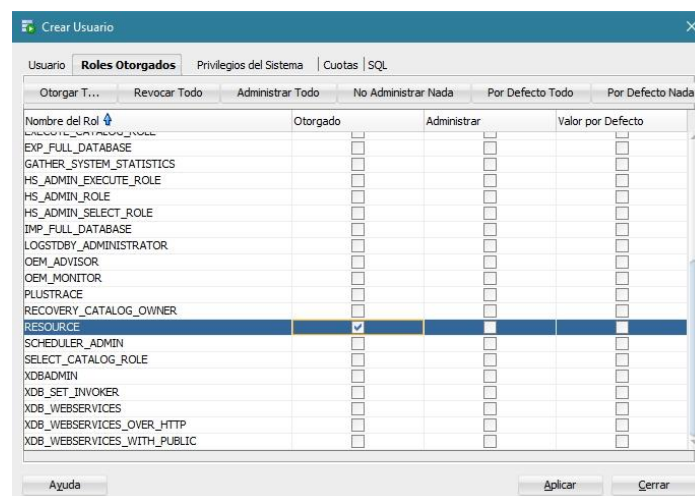
En usuario y en los 2 campos de contraseña ponemos “GRUPAL”, para no dar por culo con el PHP (así todos tenemos la misma contraseña y no hay que estar cambiándola con cada PULL que hagamos).




Antes de darle a “Aplicar” os vais a “Roles Otorgados” y se marca “Connect” y “Resource”. Una vez marcados le damos a “Aplicar”.

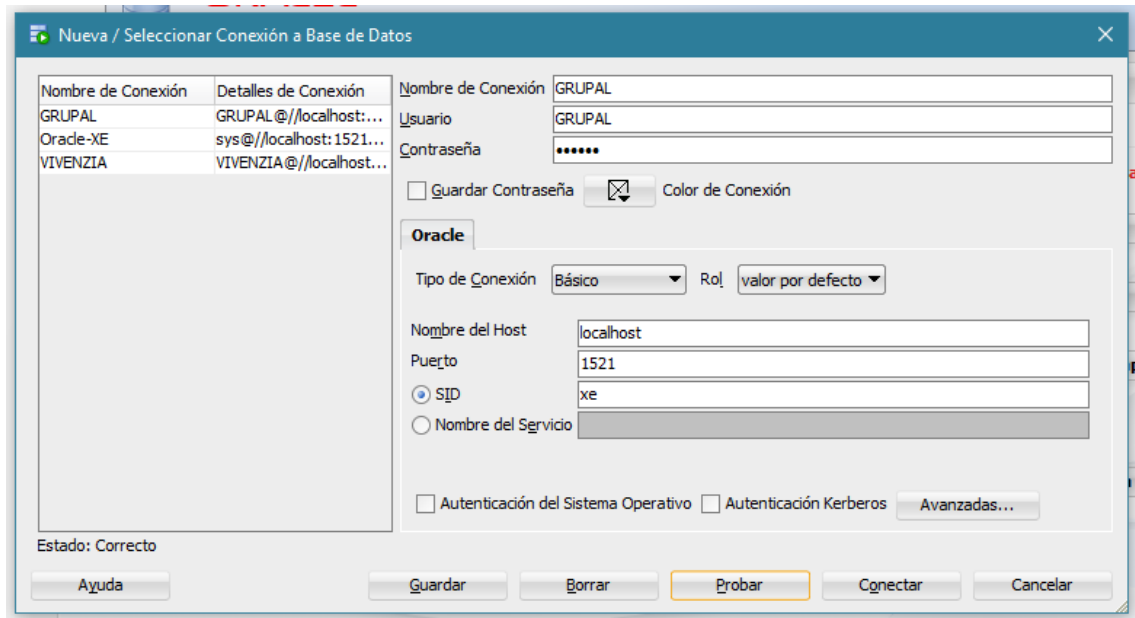


Nombre del Rol	Otorgado	Administrar	Valor por Defecto
ADM_PARALLEL_EXECUTE_TASK	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
APEX_ADMINISTRATOR_ROLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
AQ_ADMINISTRATOR_ROLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
AQ_USER_ROLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
AUTHENTICATEDUSER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>CONNECT</b>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CTXAPP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DATA PUMP_EXP_FULL_DATABASE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DATA PUMP_IMP_FULL_DATABASE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DBA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DBFS_ROLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DELETE_CATALOG_ROLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
EXECUTE_CATALOG_ROLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
EXP_FULL_DATABASE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
GATHER_SYSTEM_STATISTICS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
HS_ADMIN_EXECUTE_ROLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
HS_ADMIN_ROLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
HS_ADMIN_SELECT_ROLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
IMP_FULL_DATABASE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
LOGSTDBY_ADMINISTRATOR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

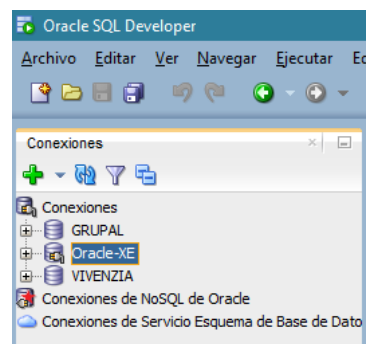


Nombre del Rol	Otorgado	Administrar	Valor por Defecto
EXP_FULL_DATABASE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
GATHER_SYSTEM_STATISTICS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
HS_ADMIN_EXECUTE_ROLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
HS_ADMIN_ROLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
HS_ADMIN_SELECT_ROLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
IMP_FULL_DATABASE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
LOGSTDBY_ADMINISTRATOR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
OEM_ADVISOR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
OEM_MONITOR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLUSTRACE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
RECOVERY_CATALOG_OWNER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>RESOURCE</b>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SCHEDULER_ADMIN	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SELECT_CATALOG_ROLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
XDBADMIN	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
XDB_SET_INVOKER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
XDB_WEBSERVICES	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
XDB_WEBSERVICES_OVER_HTTP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
XDB_WEBSERVICES_WITH_PUBLIC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

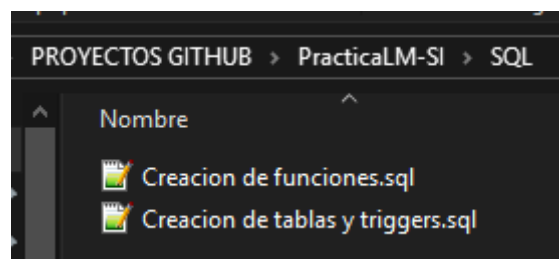
Le damos a  de nuevo. Rellenamos la pestaña, ahora con GRUPAL, GRUPAL, GRUPAL y dejamos los demás valores por defecto. Le damos a probar, y si aparece “Estado: Correcto” le damos a conectar.



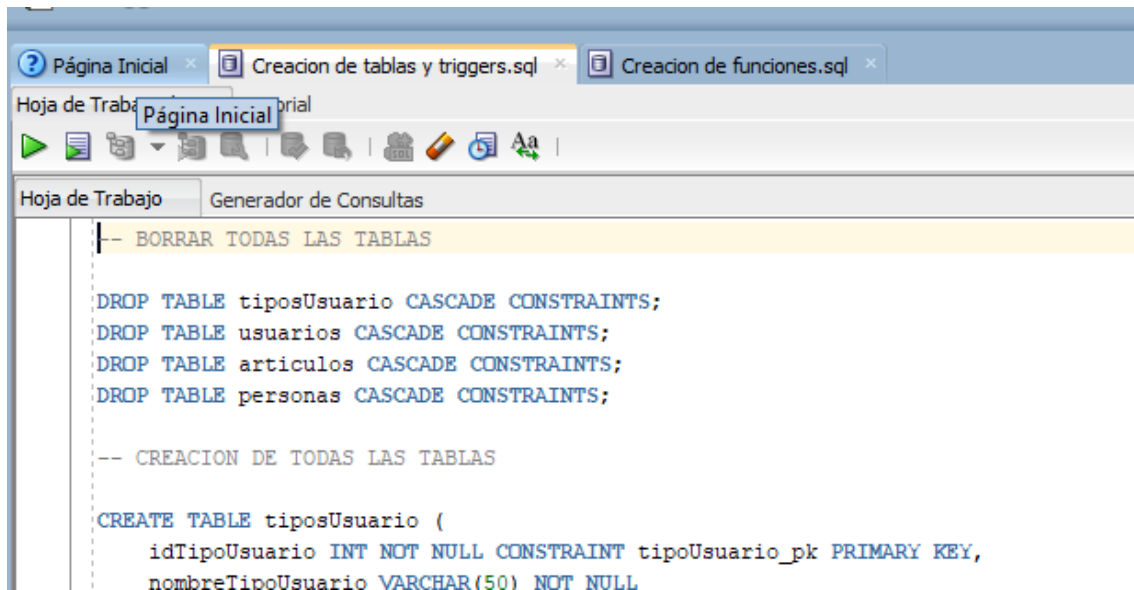
Ahora sí debería aparecer la sección conexiones con “Oracle-XE” y con “GRUPAL”.



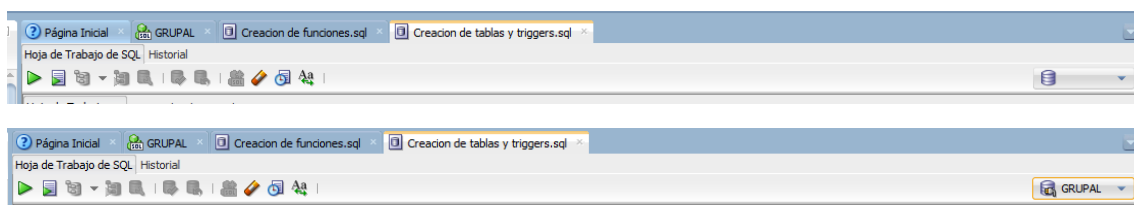
Ahora debemos irnos a la carpeta donde tengamos los archivos, a la carpeta SQL y arrastramos los archivos hasta SQL Developer.





Debería quedar así.



Antes de ejecutar nada debemos asegurarnos de que lo hacemos sobre la base de datos GRUPAL. Fijaos en el cambio de la opción de la derecha de la imagen.



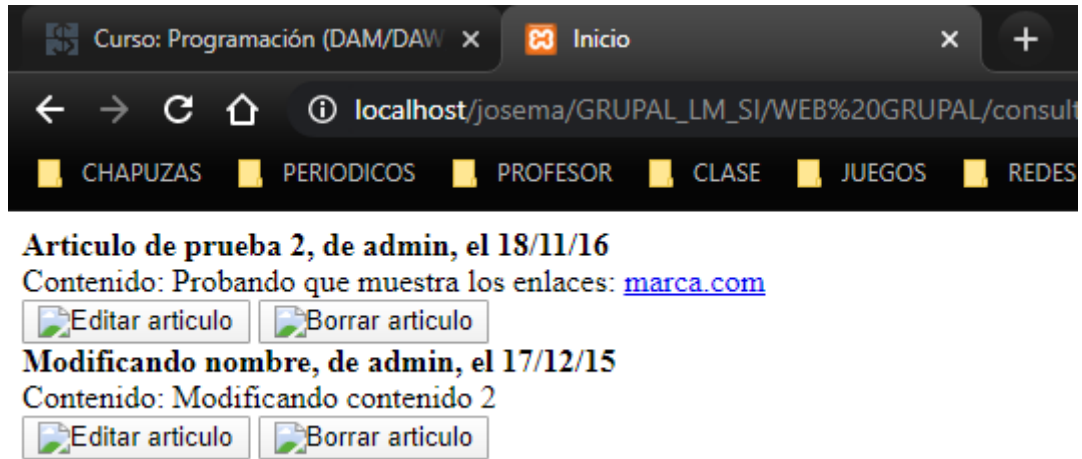
Para ejecutar miramos estos botones  . El play verde ejecuta solo la línea que tengamos marcada, la hoja con el miniplay ejecuta toda la página (**Al revés que en workbench**).

Primero ejecutamos 2 veces el de Creación de tablas y triggers, y luego 2 veces el de creación de funciones. Esto solo se hace la 1ª vez. Si no borraremos lo que insertemos desde la web. Si obtenéis fallos la segunda vez que le deis fijaos que en los mensajes de error debería haber mensajes escritos en los triggers (excepciones).

Es muy importante que si insertáis datos desde la web guardéis en un .txt todo lo que vayáis a meter, todos los campos del formulario, incluidos posibles enlaces o fotos con su estructura html (las fotos se van a guardar como parte del varchar del contenido, así que las podéis coger de internet sin descargar).

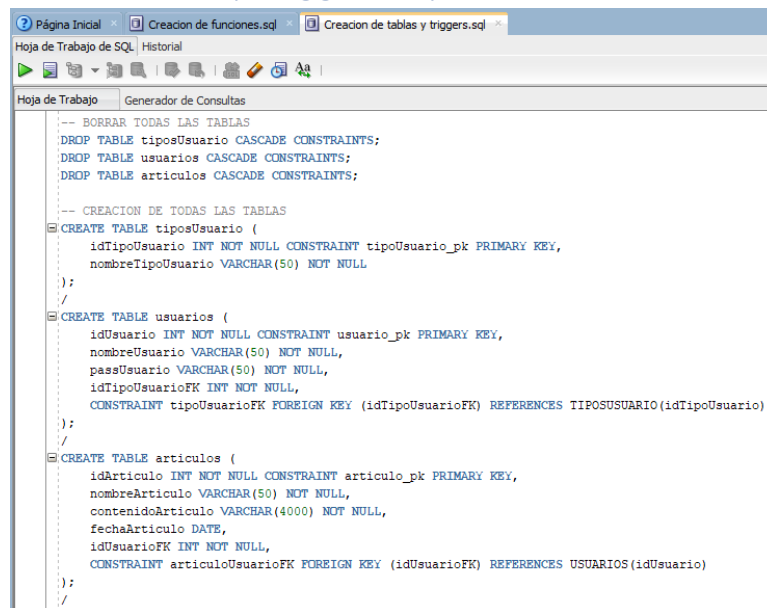
## 1.4 Comprobación

Si todo ha funcionado bien ahora debéis ir al localhost de XAMPP y entrar en "consultaArticulos.php" y que os aparezca esto. Si no sabes llegar vuelve al final de la instalación de XAMPP para ver cómo llegar.



## 2. Funcionamiento PL/SQL

### 2.1 Creación de tablas y triggers.sql

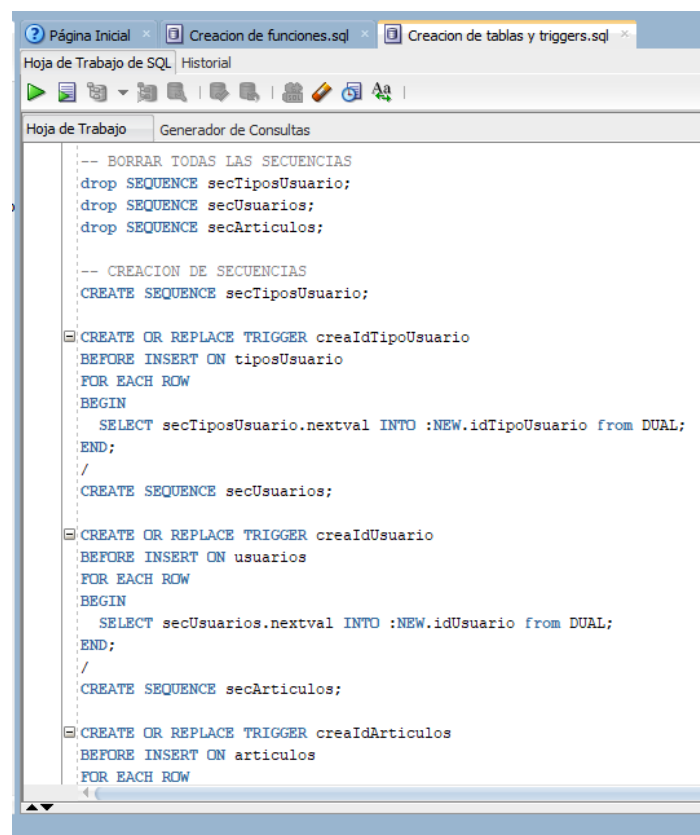


```

-- BORRAR TODAS LAS TABLAS
DROP TABLE tiposUsuario CASCADE CONSTRAINTS;
DROP TABLE usuarios CASCADE CONSTRAINTS;
DROP TABLE articulos CASCADE CONSTRAINTS;

-- CREACION DE TODAS LAS TABLAS
CREATE TABLE tiposUsuario (
  idTipoUsuario INT NOT NULL CONSTRAINT tipoUsuario_pk PRIMARY KEY,
  nombreTipoUsuario VARCHAR(50) NOT NULL
);
/
CREATE TABLE usuarios (
  idUsuario INT NOT NULL CONSTRAINT usuario_pk PRIMARY KEY,
  nombreUsuario VARCHAR(50) NOT NULL,
  passUsuario VARCHAR(50) NOT NULL,
  idTipoUsuarioFK INT NOT NULL,
  CONSTRAINT tipoUsuarioFK FOREIGN KEY (idTipoUsuarioFK) REFERENCES TIPOSUSUARIO(idTipoUsuario)
);
/
CREATE TABLE articulos (
  idArticulo INT NOT NULL CONSTRAINT articulo_pk PRIMARY KEY,
  nombreArticulo VARCHAR(50) NOT NULL,
  contenidoArticulo VARCHAR(4000) NOT NULL,
  fechaArticulo DATE,
  idUsuarioFK INT NOT NULL,
  CONSTRAINT articuloUsuarioFK FOREIGN KEY (idUsuarioFK) REFERENCES USUARIOS(idUsuario)
);
/
  
```

Si os fijáis se parece mucho a lo que hacemos en SQL Server, la diferencia es que a las primary key y las foreign key se les da un nombre.



```

-- BORRAR TODAS LAS SECUENCIAS
drop SEQUENCE secTiposUsuario;
drop SEQUENCE secUsuarios;
drop SEQUENCE secArticulos;

-- CREACION DE SECUENCIAS
CREATE SEQUENCE secTiposUsuario;

CREATE OR REPLACE TRIGGER creaIdTipoUsuario
BEFORE INSERT ON tiposUsuario
FOR EACH ROW
BEGIN
  SELECT secTiposUsuario.nextval INTO :NEW.idTipoUsuario from DUAL;
END;
/
CREATE SEQUENCE secUsuarios;

CREATE OR REPLACE TRIGGER creaIdUsuario
BEFORE INSERT ON usuarios
FOR EACH ROW
BEGIN
  SELECT secUsuarios.nextval INTO :NEW.idUsuario from DUAL;
END;
/
CREATE SEQUENCE secArticulos;

CREATE OR REPLACE TRIGGER creaIdArticulos
BEFORE INSERT ON articulos
FOR EACH ROW
  
```

Las secuencias son necesarias para que los “id...” sean autonuméricos. Así se evita que alguien con malas intenciones te machaque un “id” que ya estuviese creado.

```
-- TRIGGER PARA IDs
CREATE OR REPLACE
FUNCTION ASSERT_EQUALS (salida BOOLEAN, salida_esperada BOOLEAN) RETURN VARCHAR2 AS
BEGIN
    IF (salida = salida_esperada) THEN
        RETURN 'EXITO';
    ELSE
        RETURN 'FALLO';
    END IF;
END ASSERT_EQUALS;
/
```

Este otro trozo de código es para lo mismo que las secuencias. Evitamos duplicidad de id.

```
-- CREACION DE USUARIO ADMINISTRADOR INICIAL
insert into tiposUsuario values (null, 'Administrador Jefe');
insert into tiposUsuario values (null, 'Administrador');
insert into tiposUsuario values (null, 'Colaborador');
insert into tiposUsuario values (null, 'Registrado');

insert into usuarios values (null, 'admin', '12345', 1);
insert into usuarios values (null, 'admin2', '12345', 2);

insert into articulos values (null, 'Articulo de prueba', 'Probando que conecta con la base de datos y se ve correcto
<br><br>
Y continua el texto', TO_DATE('17/12/2015', 'DD/MM/YYYY'), 1);

insert into articulos values (null, 'Articulo de prueba 2', 'Probando que muestra los enlaces: <a href="http://marca.com"
target="_blank">marca.com</a>', TO_DATE('18/11/2016', 'DD/MM/YYYY'), 2);

SELECT * FROM ARTICULOS, USUARIOS, TIPOSUSUARIO where ARTICULOS.idUsuarioFK=USUARIOS.idUsuario and
USUARIOS.idTipoUsuarioFK=TIPOSUSUARIO.idTipoUsuario;
```

De los insert solo fijaros en que los de artículos incluyen todo el tema de los enlaces para cuando se muestren en el localhost.



## 2.2 Creación de funciones.sql

```
-- Funciones para la pagina web--
CREATE OR REPLACE PROCEDURE insertarArticulo
(w_tituloArticulo IN ARTICULOS.NOMBREARTICULO%TYPE,
w_contenidoArticulo IN ARTICULOS.CONTENIDOARTICULO%TYPE,
w_idAutor IN ARTICULOS.IDUSUARIOFK%TYPE)
IS
BEGIN
INSERT INTO ARTICULOS (IDARTICULO, NOMBREARTICULO, CONTENIDOARTICULO, FECHAARTICULO, IDUSUARIOFK)
VALUES (null, w_tituloArticulo, w_contenidoArticulo, current_date, w_idAutor);
COMMIT WORK;
END insertarArticulo;
/

CREATE OR REPLACE PROCEDURE eliminarArticulo
(w_IdArticulo IN ARTICULOS.IDARTICULO%TYPE)
IS
BEGIN
DELETE FROM ARTICULOS WHERE w_IdArticulo = IDARTICULO;
COMMIT WORK;
END;
/

CREATE OR REPLACE PROCEDURE modificarArticulo
(w_IdArticulo IN ARTICULOS.IDARTICULO%TYPE)
IS
BEGIN
DELETE FROM ARTICULOS WHERE w_IdArticulo = IDARTICULO;
COMMIT WORK;
END eliminarArticulo;
/
```

Las funciones y procedimientos se parecen a los métodos de JAVA. Se declara un nombre del procedimiento.

```
CREATE OR REPLACE PROCEDURE insertarArticulo
```

Se declaran las variables que vamos a insertar al llamar al procedimiento.

```
(w_tituloArticulo IN ARTICULOS.NOMBREARTICULO%TYPE,
w_contenidoArticulo IN ARTICULOS.CONTENIDOARTICULO%TYPE,
w_idAutor IN ARTICULOS.IDUSUARIOFK%TYPE)
```

Y luego se escribe que va a hacer en concreto.

```
IS
BEGIN
INSERT INTO ARTICULOS (IDARTICULO, NOMBREARTICULO, CONTENIDOARTICULO, FECHAARTICULO, IDUSUARIOFK)
VALUES (null, w_tituloArticulo, w_contenidoArticulo, current_date, w_idAutor);
COMMIT WORK;
```

Y se cierra.

```
END insertarArticulo;
```

Cada uno según lo que haga es diferente, pero manteniendo esta estructura.

```
/* Prueba Requisito funcional n°05 'alta oferta' */
BEGIN
  DBMS_OUTPUT.PUT_LINE('Prueba nuevo articulo:');
END;
/
EXECUTE insertarArticulo('Es una prueba', 'Probando que funcionan las funciones', 1);

SELECT * FROM ARTICULOS, USUARIOS, TIPOSUSUARIO where ARTICULOS.idUsuarioFK=USUARIOS.idUsuario and
USUARIOS.idTipoUsuarioFK=TIPOSUSUARIO.idTipoUsuario;

SELECT count(*) FROM ARTICULOS, USUARIOS, TIPOSUSUARIO where ARTICULOS.idUsuarioFK=USUARIOS.idUsuario and
USUARIOS.idTipoUsuarioFK=TIPOSUSUARIO.idTipoUsuario;

CALL eliminarArticulo(3);

SELECT count(*) FROM ARTICULOS, USUARIOS, TIPOSUSUARIO where ARTICULOS.idUsuarioFK=USUARIOS.idUsuario and
USUARIOS.idTipoUsuarioFK=TIPOSUSUARIO.idTipoUsuario;

EXECUTE insertarUsuario('Pepe', 'contraseña', 3);

SELECT * FROM USUARIOS;

SELECT count(*) FROM USUARIOS;

CALL eliminarUsuario(3);
```

Las 3 primeras líneas de “Begin...” es lo que en JAVA sería el “sysout”.

EXECUTE y CALL son llamadas a procedimientos y funciones. Mejor usar CALL siempre que podáis, en principio deberían funcionar igual de bien en SQL pero en PHP puede dar problemas EXECUTE.

Este código son pruebas de los procedimientos creados.

### 3. Funcionamiento PHP

PHP lo que va a hacer es mostrar por el HTML la información de SQL, también vamos a poder llamar a procedimientos (insertar, borrar, modificar). Va a estar “mezclado” con el HTML, así que atentos.

El orden no tiene que ser el que sigo por cojones. Puede haber variaciones. Si no veis claro cual es el siguiente PHP que tenéis que mirar buscad por el código.

```
require_once("gestionBD.php");
require_once("gestionarArticulos.php");
require_once ("paginacion_consulta.php");
```

```
<article class="articulo" id="articleArticulo">
  <form method="post" action="controladorArticulos.php">
```

Por poner 2 ejemplos. En “consultaArticulos.php” podemos tener que mirar cualquiera de estos 4 para buscar un fallo, aparte del propio código de “consultaArticulos.php”.

#### 3.1 gestionBD.php

```
gestionBD.php
1 <?php
2
3 function crearConexionBD()
4 {
5     $host="oci:dbname=localhost/XE;charset=UTF8";
6     $usuario="GRUPAL";
7     $password="GRUPAL";
8
9     try{
10         /* Indicar que las sucesivas conexiones se puedan reutilizar */
11         $conexion=new PDO($host,$usuario,$password,array(PDO::ATTR_PERSISTENT => true));
12         /* Indicar que se disparen excepciones cuando ocurra un error*/
13         $conexion->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
14         return $conexion;
15     }catch(PDOException $e){
16         $_SESSION['excepcion'] = $e->getMessage();
17         header("Location: excepcion.php");
18     }
19 }
20
21 function cerrarConexionBD($conexion){
22     $conexion=null;
23 }
24
25 ?>
```

De aquí solo nos interesa la parte donde pone:

```
$usuario="GRUPAL";
$password="GRUPAL";
```

Ese usuario y esa contraseña deben ser los que pongamos en SQL (por eso os insistí en que todos pusiéramos “GRUPAL” “GRUPAL”). Si alguna vez escribís un código para otra base de datos deberéis cambiar esto.

## 3.2 consultaArticulos.php

```

1 <?php
2 session_start();
3 require_once("gestionBD.php");
4 require_once("gestionArticulos.php");
5 require_once("paginacion_consulta.php");
6 if (isset($_SESSION["articulo"])) {
7     $articulo = $_SESSION["articulo"];
8     unset($_SESSION["articulo"]);
9 }
10 // ¿Venimos simplemente de cambiar página o de haber seleccionado un registro ?
11 // ¿Hay una sesión activa?
12 if (isset($_SESSION["paginacion"])) $paginacion = $_SESSION["paginacion"];
13 $pagina_seleccionada = isset($_GET["PAG_NUM"]) ? (int)$_GET["PAG_NUM"] : (isset($paginacion) ? (int)$paginacion["PAG_NUM"] : 1);
14 $pag_tam = isset($_GET["PAG_TAM"]) ? (int)$_GET["PAG_TAM"] : (isset($paginacion) ? (int)$paginacion["PAG_TAM"] : 5);
15 if ($pagina_seleccionada < 1) $pagina_seleccionada = 1;
16 if ($pag_tam < 1) $pag_tam = 5;
17 // Antes de seguir, borramos las variables de sesión para no confundirnos más adelante
18 unset($_SESSION["paginacion"]);
19 $conexion = crearConexionBD();
20 // La consulta que ha de paginarse
21 $query = "SELECT * FROM ARTICULOS, USUARIOS, TIPOSUSUARIO where ARTICULOS.idUsuarioFK=USUARIOS.idUsuario and USUARIOS.idTipoUsuarioFK=
TIPOSUSUARIO.idTipoUsuario";
22 // Se comprueba que el tamaño de página, página seleccionada y total de registros son conformes.
23 // En caso de que no, se asume el tamaño de página propuesto, pero desde la página 1
24 $total_registros = total_consulta($conexion, $query);
25 $total_paginas = (int)($total_registros / $pag_tam);
26 if ($total_registros % $pag_tam > 0) $total_paginas++;
27 if ($pagina_seleccionada > $total_paginas) $pagina_seleccionada = $total_paginas;
28 // Generamos los valores de sesión para página e intervalo para volver a ella después de una operación
29 $paginacion["PAG_NUM"] = $pagina_seleccionada;
30 $paginacion["PAG_TAM"] = $pag_tam;
31 $_SESSION["paginacion"] = $paginacion;
32 $filas = consulta_paginada($conexion, $query, $pagina_seleccionada, $pag_tam);
33 cerrarConexionBD($conexion);
34 >>

```

Primera parte del código.

Las primeras líneas son las que necesita la pagina para funcionar (llamadas a la base de datos para ver los artículos, mas adelante se detalla como mostrarlo).

A partir del primer "if..." es un código necesario para comprobar la sesión en la que estamos (información actual de la página tal como el usuario que estará logueado o los datos del formulario cuando hagamos un insert o un update).

A partir del segundo "if..." es mas especial. Su utilidad es poder mostrar "x" artículos a gusto. Esta parte es 100% copia pega, si os leéis los comentarios se explica su funcionamiento.

```

35 <!DOCTYPE html>
36 <html lang="es">
37 <head>
38   <meta charset="utf-8">
39   <title>Inicio</title>
40 </head>
41 <body>
42 <main>
43   <?php
44     foreach($filas as $fila) {
45       ?>
46 <article class="articulo" id="articleArticulo">
47   <form method="post" action="controladorArticulos.php">
48     <div class="filaArticulo">
49       <div class="datosArticulo">
50         <input id="IDARTICULO" name="IDARTICULO"
51           type="hidden" value="<?php echo $fila["IDARTICULO"]; ?>"/>
52         <input id="NOMBREARTICULO" name="NOMBREARTICULO"
53           type="hidden" value="<?php echo $fila["NOMBREARTICULO"]; ?>"/>
54         <input id="NOMBREUSUARIO" name="NOMBREUSUARIO"
55           type="hidden" value="<?php echo $fila["NOMBREUSUARIO"]; ?>"/>
56         <input id="FECHAARTICULO" name="FECHAARTICULO"
57           type="hidden" value="<?php echo $fila["FECHAARTICULO"]; ?>"/>

```

Aquí empieza el HTML que hemos visto en clase... más o menos.

Lo primero que necesitamos para mostrar la info del PHP es lo que está en el main. Si os fijáis cada vez que se usa el PHP se empieza con "<?php" y se termina con "?>".

Luego se crea la clase "articulo" y se establece un formulario. Lo del formulario es para poder modificar después. Si no vamos a modificar no haría falta.

Cada "input" es cada uno de los valores que vamos a trabajar después. Dentro de cada uno hay un trozo de PHP en la parte de "value", enlazando cada dato con su nuevo "id".

```

<?php if (isset($articulo) and ($articulo["IDARTICULO"] == $fila["IDARTICULO"])) { ?>
  <!-- Editando título -->
  <input id="NOMBREARTICULO" name="NOMBREARTICULO" type="hidden" value="<?php echo $fila["NOMBREARTICULO"]; ?>"/>
  <div class="Info">
    <em><input id="NOMBREARTICULO" name="NOMBREARTICULO" type="text" value="<?php echo $fila["NOMBREARTICULO"]; ?>"/></em>
    <b><?php echo " , de " . $fila["NOMBREUSUARIO"] . " , el " . $fila["FECHAARTICULO"]; ?></b>
  </div>
  <div class="Contenido">
    Contenido: <em><input id="CONTENIDOARTICULO" name="CONTENIDOARTICULO" type="text" value="<?php echo $fila['CONTENIDOARTICULO']; ?>"/></em>
  </div>
<?php } else { ?>
  <!-- mostrando título -->
  <input id="NOMBREARTICULO" name="NOMBREARTICULO" type="hidden" value="<?php echo $fila["NOMBREARTICULO"]; ?>"/>
  <div class="Info">
    <b><?php echo $fila["NOMBREARTICULO"] . " , de " . $fila["NOMBREUSUARIO"] . " , el " . $fila["FECHAARTICULO"]; ?></b>
  </div>
  <div class="Contenido">Contenido: <?php echo $fila['CONTENIDOARTICULO']; ?></div>
<?php } ?>
</div>

```

Aquí se muestra la información o se modifica. Se llama al objeto "\$fila" que es el que contiene cada apartado de lo que vamos a mostrar.

```

<div id="botones_fila">
  <?php if (isset($articulo) and ($articulo["IDARTICULO"] == $fila["IDARTICULO"])) { ?>
    <button id="grabar" name="grabar" type="submit" class="editar_fila">
      
    </button>
  <?php } else { ?>
    <button id="editar" name="editar" type="submit" class="editar_fila" >
      
    </button>
  <?php } ?>
  <button id="borrar" name="borrar" type="submit" class="editar_fila" onclick="return confirm('¿Está seguro de eliminar?')
  ;">
    
  </button>
</div>

```

Los botones para las distintas opciones, cuando pulsemos nos mandarán a controladorArticulos.php como pusimos al principio del formulario. Las funciones "onclick="return confirm(...)" son de JavaScript.

### 3.3 controladorArticulos.php

```

controladorArticulos.php x
1 <?php
2   session_start();
3
4   if (isset($_REQUEST["IDARTICULO"])) {
5     $articulo["IDARTICULO"] = $_REQUEST["IDARTICULO"];
6     $articulo["NOMBREARTICULO"] = $_REQUEST["NOMBREARTICULO"];
7     $articulo["CONTENIDOARTICULO"] = $_REQUEST["CONTENIDOARTICULO"];
8     $articulo["NOMBREUSUARIO"] = $_REQUEST["NOMBREUSUARIO"];
9
10    $_SESSION["articulo"] = $articulo;
11
12    if (isset($_REQUEST["editar"])) Header("Location: consultaArticulos.php");
13    else if (isset($_REQUEST["grabar"])) Header("Location: accionModificarArticulo.php");
14    else if (isset($_REQUEST["borrar"])) Header("Location: accionBorrarArticulo.php");
15  }
16  else
17    Header("Location: consultaArticulos.php");
18
19  ?>
20

```

Aquí no hay mucho que ver. Solo se identifica la información que se añade en \$articulo y se manda a otro de los php o se vuelve a consultaArticulos.

### 3.4 accionModificarArticulo.php

```

1 <?php
2 session_start();
3
4 if (isset($_SESSION["articulo"])) {
5     $modificarArticulo = $_SESSION["articulo"];
6     unset($_SESSION["articulo"]);
7
8     require_once("gestionBD.php");
9     require_once("gestionarArticulos.php");
10
11     try{
12         $conexion = crearConexionBD();
13         $errores = validarModificacion($conexion, $modificarArticulo);
14         cerrarConexionBD($conexion);
15     }catch(PDOException $e){
16         // Mensaje de depuración
17         $_SESSION["errores"] = "<p>ERROR en la validación: fallo en el acceso a la base de datos.</p><p>" . $e->getMessage() . "</p>";
18         Header('Location: consultaArticulos.php');
19     }
20
21     if (count($errores)>0) {
22         $_SESSION["errores"] = $errores;
23         Header('Location: consultaArticulos.php');
24     } else {
25         $conexion = crearConexionBD();
26         $excepcion = modificarArticulo($conexion, $modificarArticulo["IDARTICULO"], $modificarArticulo["NOMBREARTICULO"], $modificarArticulo["CONTENIDOARTICULO"]);
27         cerrarConexionBD($conexion);
28     } if ($excepcion<>"") {
29         $_SESSION["excepcion"] = $excepcion;
30         $_SESSION["destino"] = "consultaArticulos.php";
31         Header("Location: excepcion.php");
32     } else
33         Header("Location: consultaArticulos.php");
34 } else Header("Location: consultaArticulos.php"); // Se ha tratado de acceder directamente a este PHP

```

Aquí volvemos a necesitar conectar con la BBDD y probamos a llamar a función `modificarArticulo` en la línea que llamará a la BBDD en `gestionarArticulos.php` (más adelante)

```

} else Header("Location: consultaArticulos.php"); // Se ha tratado de acceder directamente a este PHP

function validarModificacion($conexion, $modificarArticulo){
    $errores=array();
    // Validación del NIF
    if($modificarArticulo["CONTENIDOARTICULO"]=="" && $modificarArticulo["NOMBREARTICULO"]=="") {
        $errores[] = "<p>No puede estar vacío</p>";
    }
    return $errores;
}
}

```

Aparte solo comprobamos que la modificación no venga vacía.

### 3.5 accionBorrarArticulo.php

```

1 <?php
2     session_start();
3
4     if (isset($_SESSION["articulo"])) {
5         $articulo = $_SESSION["articulo"];
6         unset($_SESSION["articulo"]);
7
8         require_once("gestionBD.php");
9         require_once("gestionarArticulos.php");
10
11         $conexion = crearConexionBD();
12         $excepcion = eliminarArticulo($conexion,$articulo["IDARTICULO"]);
13         cerrarConexionBD($conexion);
14
15         if ($excepcion<>"") {
16             $_SESSION["excepcion"] = $excepcion;
17             $_SESSION["destino"] = "consultaArticulos.php";
18             Header("Location: excepcion.php");
19         }
20         else Header("Location: consultaArticulos.php");
21     }
22     else Header("Location: consultaArticulos.php");
23 ?>
24

```

Lo mismo que en el anterior, pero llamando a eliminarArticulo.

### 3.6 gestionarArticulos.php

```

1 <?php
2
3 function consultaArticulos($conexion) {
4     $consulta = "SELECT * FROM ARTICULOS, USUARIOS, TIPOSUSUARIO where ARTICULOS.idUsuarioFK=USUARIOS.idUsuario and USUARIOS.idTipoUsuarioFK=
5     TIPOSUSUARIO.idTipoUsuario";
6     return $conexion->query($consulta);
7 }
8
9 function eliminarArticulo($conexion,$articulo) {
10     try {
11         $stmt=$conexion->prepare('CALL eliminarArticulo(:IDARTICULO)');
12         $stmt->bindParam(':IDARTICULO',$articulo);
13         $stmt->execute();
14         return "";
15     } catch(PDOException $e) {
16         return $e->getMessage();
17     }
18 }
19
20 function modificarArticulo($conexion,$IDARTICULO, $NOMBREARTICULO, $CONTENIDOARTICULO) {
21     try {
22         $stmt=$conexion->prepare('CALL modificarArticulo(:IDARTICULO,:NOMBREARTICULO,:CONTENIDOARTICULO)');
23         $stmt->bindParam(':IDARTICULO',$IDARTICULO);
24         $stmt->bindParam(':NOMBREARTICULO',$NOMBREARTICULO);
25         $stmt->bindParam(':CONTENIDOARTICULO',$CONTENIDOARTICULO);
26         $stmt->execute();
27         return "";
28     } catch(PDOException $e) {
29         return $e->getMessage();
30     }
31 }
32 ?>

```

Aquí es donde están las funciones a las que hemos estado llamando. Y esas funciones a su vez llaman a la BBDD con "CALL nombreProcedimiento (variablesProcedimiento)".



### 3.7 formAltaUsuario

```

1  <?php
2  session_start();
3  // Importar librerías necesarias para gestionar direcciones y géneros literarios
4  require_once("gestionBD.php");
5  require_once("gestionarUsuarios.php");
6  // Si no existen datos del formulario en la sesión, se crea una entrada con valores por defecto
7  if (!isset($_SESSION['formulario'])) {
8      $formulario['NOMBREUSUARIO'] = "";
9      $formulario['PASSUSUARIO'] = "";
10     $formulario['confirmpass'] = "";
11     $formulario['IDTIPOUSUARIO'] = "";
12     $formulario['DNIPERSONA'] = "";
13     $formulario['NOMBREPERSONA'] = "";
14     $formulario['APELLIDO1PERSONA'] = "";
15     $formulario['APELLIDO2PERSONA'] = "";
16     $formulario['CORREOPERSONA'] = "";
17
18     $_SESSION['formulario'] = $formulario;
19 }
20 // Si ya existían valores, los cogemos para inicializar el formulario
21 else
22     $formulario = $_SESSION['formulario'];
23
24 // Si hay errores de validación, hay que mostrarlos y marcar los campos (El estilo viene dado y ya se explicará)
25 if (isset($_SESSION["errores"])){
26     $errores = $_SESSION["errores"];
27     unset($_SESSION["errores"]);
28 }
29 // Creamos una conexión con la BD
30 $conexion = crearConexionBD();
31 >>

```

Parecido al inicio de consultaArticulos.php, pero en este caso declaramos la sesión “formulario” con todas las variables que vamos a necesitar. OJO que se llama a “gestionarUsuarios.php”, necesario para datos del formulario.

```

32 <!DOCTYPE html>
33 <html lang="es">
34 <head>
35     <meta charset="utf-8">
36     <link rel="stylesheet" type="text/css" href="css/biblio.css" />
37     <script src="https://code.jquery.com/jquery-3.1.1.min.js" type="text/javascript"></script>
38     <script src="js/validacion_cliente_alta_usuario.js" type="text/javascript"></script>
39     <title>Gestión de Usuarios: Nuevo usuario</title>
40 </head>
41 <body>
42     <script>
43         // Inicialización de elementos y eventos cuando el documento se carga completamente
44         $(document).ready(function() {
45
46             $("#PASSUSUARIO").on("keyup", function() {
47                 passwordColor();
48             });
49         });
50     </script>
51     <?php
52         // Mostrar los errores de validación (Si los hay)
53         if (isset($errores) && count($errores)>0) {
54             echo "<div id='div_errores' class='error'>";
55             echo "<h4> Errores en el formulario:</h4>";
56             foreach($errores as $error){
57                 echo $error;
58             }
59             echo "</div>";
60         }
61     >>

```

Inicio del HTML con las cosas de siempre y con un trozo de PHP que recopilará y mostrará los errores que cometamos al introducir la info en el formulario (como usuarios).

```

62 <form id="altaUsuario" method="get" action="accionAltaUsuario.php" onsubmit="return validateForm()">
63 <p><i>¡¡¡ TODOS LOS CAMPOS SON OBLIGATORIOS!!!</i></p>
64 <fieldset>
65 <div><label for="DNIPERSONA">DNI</label>
66 <input id="DNIPERSONA" name="DNIPERSONA" type="text" placeholder="12345678X" pattern="^[0-9]{8}[A-Z]" title="Ocho dígitos
seguidos de una letra mayúscula" value="{php echo $formulario['DNIPERSONA'];}" required>
67 </div>
68
69 <div><label for="NOMBREPERSONA">Nombre:</label>
70 <input id="NOMBREPERSONA" name="NOMBREPERSONA" type="text" size="40" value="{php echo $formulario['NOMBREPERSONA'];}" required
/>
71 </div>
72
73 <div><label for="APELLIDO1PERSONA">Primer apellido:</label>
74 <input id="APELLIDO1PERSONA" name="APELLIDO1PERSONA" type="text" size="80" value="{php echo $formulario['APELLIDO1PERSONA'];}"
/>
75 </div>
76
77 <div><label for="APELLIDO2PERSONA">Segundo apellido:</label>
78 <input id="APELLIDO2PERSONA" name="APELLIDO2PERSONA" type="text" size="80" value="{php echo $formulario['APELLIDO2PERSONA'];}"
/>
79 </div>
80
81 <div><label for="CORREOPERSONA">Email:</label>
82 <input id="CORREOPERSONA" name="CORREOPERSONA" type="email" placeholder="usuario@dominio.extension" value="{php echo $
formulario['CORREOPERSONA'];}" required/><br>
83 </div>
84
85 <div><label for="NOMBREUSUARIO">Usuario:</label>
86 <input id="NOMBREUSUARIO" name="NOMBREUSUARIO" type="text" size="40" value="{php echo $formulario['NOMBREUSUARIO'];}" />
87 </div>
88 <div><label for="PASSUSUARIO">Contraseña:</label>
89 <input type="password" name="PASSUSUARIO" id="PASSUSUARIO" placeholder="Mínimo 8 caracteres entre letras y dígitos" oninput=
"passwordValidation();" required/>

```

El formulario normal en HTML pero en cada value introducimos el correspondiente PHP.

```

<div><label for="IDTIPOUSUARIO">Rol:</label>
<select name="IDTIPOUSUARIO[]" id="IDTIPOUSUARIO" required>
  <?php $tipos = listarTipoUsuario($conexion);
  foreach ($tipos as $tipo) {
    if(in_array($tipo["IDTIPOUSUARIO"], $formulario['IDTIPOUSUARIO'])) {
      echo "<option value='" . $tipo["IDTIPOUSUARIO"] . "' label='" . $tipo["NOMBRETIPUSUARIO"] . "' selected/>";
    } else {
      echo "<option value='" . $tipo["IDTIPOUSUARIO"] . "' label='" . $tipo["NOMBRETIPUSUARIO"] . "' />";
    }
  }
  <?>
</select>
</div>
</fieldset>
<p><i>¡¡¡ TODOS LOS CAMPOS SON OBLIGATORIOS!!!</i></p>
<div><input type="submit" value="Enviar" /></div>
</form>
<?php
  cerrarConexionBD($conexion);
?>
</body>
</html>

```

Aquí creamos un select que traeremos de la base de datos y que a la vez será un dato para el formulario. Para terminar, se cierra la conexión con la base de datos (abierta hasta aquí por lo del select).

### 3.8 gestionarUsuarios.php

```

1 <?php
2 function altaUsuario($conexion,$usuario) {
3     $tipo = implode(',', $usuario["IDTIPOUSUARIO"]);
4     $hash = password_hash($usuario["PASSUSUARIO"], PASSWORD_DEFAULT);
5     try {
6         $consulta = "CALL insertarUsuario(:NOMBREUSUARIO, :PASSUSUARIO, :IDTIPOUSUARIO, :DNIPERSONA, :NOMBREPERSONA, :APELLIDO1PERSONA,
7             :APELLIDO2PERSONA, :CORREOPERSONA)";
8         $stmt=$conexion->prepare($consulta);
9         $stmt->bindParam(':NOMBREUSUARIO',$usuario["NOMBREUSUARIO"]);
10        $stmt->bindParam(':PASSUSUARIO',$hash);
11        $stmt->bindParam(':IDTIPOUSUARIO',$tipo);
12        $stmt->bindParam(':DNIPERSONA',$usuario["DNIPERSONA"]);
13        $stmt->bindParam(':NOMBREPERSONA',$usuario["NOMBREPERSONA"]);
14        $stmt->bindParam(':APELLIDO1PERSONA',$usuario["APELLIDO1PERSONA"]);
15        $stmt->bindParam(':APELLIDO2PERSONA',$usuario["APELLIDO2PERSONA"]);
16        $stmt->bindParam(':CORREOPERSONA',$usuario["CORREOPERSONA"]);
17
18        $stmt->execute();
19    } catch(PDOException $e) {
20        return $e->getMessage();
21        // Si queremos visualizar la excepción durante la depuración: $e->getMessage();
22    }
23 }
24
25 function listarTipoUsuario($conexion){
26     try{
27         $consulta = "SELECT * FROM TIPOUSUARIO";
28         $stmt = $conexion->query($consulta);
29         return $stmt;
30     }catch(PDOException $e) {
31         return $e->getMessage();
32     }
33 }
34
35 function listarUsuarios($conexion){

```

Este php esta lleno de las funciones necesarias para los Usuarios. Lo mismo que en gestionarArticulos.php. Hay CALL a procedimientos y SELECT que se dirigen a la BBDD.

### 3.9 accionAltaUsuario.php

```

1 <?php
2 session_start();
3 // Importar librerías necesarias para gestionar direcciones y géneros literarios
4 require_once("gestionBD.php");
5 require_once("gestionUsuarios.php");
6 // Comprobar que hemos llegado a esta página porque se ha rellenado el formulario
7 if (isset($_SESSION["formulario"])) {
8     // Recogemos los datos del formulario
9     $nuevoUsuario["DNIPERSONA"] = $_REQUEST["DNIPERSONA"];
10    $nuevoUsuario["NOMBREPERSONA"] = $_REQUEST["NOMBREPERSONA"];
11    $nuevoUsuario["APELLIDO1PERSONA"] = $_REQUEST["APELLIDO1PERSONA"];
12    $nuevoUsuario["APELLIDO2PERSONA"] = $_REQUEST["APELLIDO2PERSONA"];
13    $nuevoUsuario["CORREOPERSONA"] = $_REQUEST["CORREOPERSONA"];
14    $nuevoUsuario["NOMBREUSUARIO"] = $_REQUEST["NOMBREUSUARIO"];
15    $nuevoUsuario["PASSUSUARIO"] = $_REQUEST["PASSUSUARIO"];
16    $nuevoUsuario["confirmpass"] = $_REQUEST["confirmpass"];
17    $nuevoUsuario["IDTIPOUSUARIO"] = $_REQUEST["IDTIPOUSUARIO"];
18 }
19 } else // En caso contrario, vamos al formulario
20     Header("Location: formAltaUsuario.php");
21 // Guardar la variable local con los datos del formulario en la sesión.
22 $_SESSION["formulario"] = $nuevoUsuario;
23 // Validamos el formulario en servidor
24 // Si se produce alguna excepción PDO en la validación, volvemos al formulario informando al usuario
25 try{
26     $conexion = crearConexionBD();
27     $errores = validarDatosUsuario($conexion, $nuevoUsuario);
28     cerrarConexionBD($conexion);
29 }catch(PDOException $e){
30     // Mensaje de depuración
31     $_SESSION["errores"] = "<p>ERROR en la validación: fallo en el acceso a la base de datos.</p><p>" . $e->getMessage() . "</p>";
32     Header('Location: formAltaUsuario.php');
33 }

```

```

33     }
34     // Si se han detectado errores
35     if (count($errores)>0) {
36         // Guardo en la sesión los mensajes de error y volvemos al formulario
37         $_SESSION["errores"] = $errores;
38         Header('Location: formAltaUsuario.php');
39     } else
40         // Si todo va bien, vamos a la página de éxito (inserción del usuario en la base de datos)
41         Header('Location: exitoAltaUsuario.php');
42     //////////////////////////////////////
43     // Validación en servidor del formulario de alta de usuario
44     //////////////////////////////////////
45     function validarDatosUsuario($conexion, $nuevoUsuario){
46         $errores=array();
47         // Validación del NIF
48         if($nuevoUsuario["DNIPERSONA"]=="") {
49             $errores[] = "<p>El DNIPERSONA no puede estar vacío</p>";
50         } else if(!preg_match("/^[0-9]{8}[A-Z]$/", $nuevoUsuario["DNIPERSONA"])){
51             $errores[] = "<p>El DNIPERSONA debe contener 8 números y una letra mayúscula: " . $nuevoUsuario["DNIPERSONA"]. "</p>";
52         }
53
54         // Validación del Nick
55         if($nuevoUsuario["NOMBREUSUARIO"]=="")
56             $errores[] = "<p>El NOMBREPERSONA no puede estar vacío</p>";
57
58         // Validación del NOMBREPERSONA
59         if($nuevoUsuario["NOMBREPERSONA"]=="")
60             $errores[] = "<p>El NOMBREPERSONA no puede estar vacío</p>";
61
62         // Validación del Primer Apellido
63         if($nuevoUsuario["APELLIDO1PERSONA"]=="")
64             $errores[] = "<p>El primer apellido no puede estar vacío</p>";
65
66         // Validación del Segundo Apellido
67         if($nuevoUsuario["APELLIDO2PERSONA"]=="")
68             $errores[] = "<p>El segundo apellido no puede estar vacío</p>";
69
70         // Validación del CORREOPERSONA
71         if($nuevoUsuario["CORREOPERSONA"]==""){
72             $errores[] = "<p>El CORREOPERSONA no puede estar vacío</p>";
73         }else if(!filter_var($nuevoUsuario["CORREOPERSONA"], FILTER_VALIDATE_EMAIL)){
74             $errores[] = "<p>El CORREOPERSONA es incorrecto: " . $nuevoUsuario["CORREOPERSONA"]. "</p>";
75         }
76
77         // Validación de la contraseña
78         if(!isset($nuevoUsuario["PASSUSUARIO"]) || strlen($nuevoUsuario["PASSUSUARIO"])<8){
79             $errores[] = "<p>Contraseña no válida: debe tener al menos 8 caracteres</p>";
80         }else if(!preg_match("/[a-z]+/", $nuevoUsuario["PASSUSUARIO"]) ||
81             !preg_match("/[A-Z]+/", $nuevoUsuario["PASSUSUARIO"]) || !preg_match("/[0-9]+/", $nuevoUsuario["PASSUSUARIO"])){
82             $errores[] = "<p>Contraseña no válida: debe contener letras mayúsculas y minúsculas y dígitos</p>";
83         }else if($nuevoUsuario["PASSUSUARIO"] != $nuevoUsuario["confirmpass"]){
84             $errores[] = "<p>La confirmación de contraseña no coincide con la contraseña</p>";
85         }
86
87         // Validar Tipo de inmueble
88         $error = validarIDTIPOUSUARIO($conexion, $nuevoUsuario["IDTIPOUSUARIO"]);
89         if($error!="")
90             $errores[] = $error;
91
92         // Validar DNIPERSONA
93         $error = validarDNIPERSONA($conexion, $nuevoUsuario["DNIPERSONA"]);
94         if($error!="")
95             $errores[] = $error;
96
97         // Validar NOMBREPERSONA de usuario
98         $error = validarNOMBREUSUARIO($conexion, $nuevoUsuario["NOMBREUSUARIO"]);

```

```
97 // Validar NOMBREPERSONA de usuario
98 $error = validarNOMBREUSUARIO($conexion, $nuevoUsuario["NOMBREUSUARIO"]);
99 ▼ if($error!="")
100     $errores[] = $error;
101
102 // Validar CORREOPERSONA
103 $error = validarCORREOPERSONA($conexion, $nuevoUsuario["CORREOPERSONA"]);
104 if($error!="")
105     $errores[] = $error;
106
107 return $errores;
108 }
109
110 ▼ function validarIDTIPOUSUARIO($conexion, $tipos){
111     $error="";
112     $tipos_db = array();
113     $db = listarTipoUsuario($conexion);
114     foreach ($db as $tipo_db){
115         $tipos_db[] = $tipo_db["IDTIPOUSUARIO"];
116     }
117
118     if(count(array_intersect($tipos_db, $tipos)) < count($tipos)){
119         $error = $error . "<p>El tipo de usuario no es válido</p>";
120     }
121
122     return $error;
123 }
124
125 ▼ function validarDNIPERSONA($conexion, $DNIPERSONA) {
126     $error="";
127     $db = array();
128     $db = listarUsuarios($conexion);
129     $DNIPERSONA_php = explode(',', ' ', "".$DNIPERSONA." ");
130     foreach ($db as $DNIPERSONA_db){
131         $DNIPERSONAs db[] = $DNIPERSONA db["DNIPERSONA"];
```

```
134     if(count(array_unique(array_intersect($DNIPERSONAs_db, $DNIPERSONA_php))) > 0) {
135         $error = $error . "<p>El DNIPERSONA ya existe.</p>";
136     }
137     return $error;
138 }
139 function validarNOMBREUSUARIO($conexion, $usuario) {
140     $error="";
141     $db = array();
142     $db = listarUsuarios($conexion);
143     $DNIPERSONA_php = explode(' ', $usuario);
144     foreach ($db as $DNIPERSONA_db){
145         $DNIPERSONAs_db[] = $DNIPERSONA_db["NOMBREUSUARIO"];
146     }
147
148     if(count(array_unique(array_intersect($DNIPERSONAs_db, $DNIPERSONA_php))) > 0) {
149         $error = $error . "<p>El NOMBREPERSONA de usuario ya existe.</p>";
150     }
151     return $error;
152 }
153
154 function validarCORREOPERSONA($conexion, $usuario) {
155     $error="";
156     $db = array();
157     $db = listarUsuarios($conexion);
158     $DNIPERSONA_php = explode(' ', $usuario);
159     foreach ($db as $DNIPERSONA_db){
160         $DNIPERSONAs_db[] = $DNIPERSONA_db["CORREOPERSONA"];
161     }
162
163     if(count(array_unique(array_intersect($DNIPERSONAs_db, $DNIPERSONA_php))) > 0) {
164         $error = $error . "<p>El CORREOPERSONA ya existe.</p>";
165     }
166     return $error;
167 }
```

Este lo pongo entero porque una cosa tira de otra.

Es el mas lioso. Al final lo que se comprueba es que nada vaya vacío y que todo cumpla con los requisitos y no haya repeticiones de cosas que no pueda haberlas.

### 3.10 exitoAltaUsuario.php

```

1  <?php
2      session_start();
3      require_once("gestionBD.php");
4      require_once("gestionarUsuarios.php");
5      // Comprobar que hemos llegado a esta página porque se ha rellenado el formulario
6      if (isset($_SESSION["formulario"])) {
7          $nuevoUsuario = $_SESSION["formulario"];
8          $_SESSION["formulario"] = null;
9          $_SESSION["errores"] = null;
10     } else header("Location: formAltaUsuario.php");
11     $conexion = crearConexionBD();
12 >>
13 <!DOCTYPE html>
14 <html lang="es">
15 <head>
16     <meta charset="utf-8">
17     <link rel="stylesheet" type="text/css" href="css/biblio.css" />
18     <title>VIVENZia Inmobiliaria: Alta de Usuario realizada con éxito</title>
19 </head>
20 <body>
21     <main id="altaUsuario">
22         <?php if (altaUsuario($conexion, $nuevoUsuario)) { ?>
23             <h1>Ha ocurrido un error.</h1>
24             <div>
25                 Pulsa <a href="formAltaUsuario.php">aquí</a> para volver al formulario.
26             </div>
27         <?php } else {
28             header("Location: consultaArticulos.php");
29         } ?>
30     </main>
31 </body>
32 </html>
33 <?php
34     cerrarConexionBD($conexion);
35 >>

```

Line 10, Column 12; Saved C:\xampp\htdocs\JOSEMA\GRUPAL\_LM\_SI\WEB GRUPAL\exitoAltaUsuario.php (UTF-8)

En este php lo que se hace es que si la validación anterior ha ido bien se manda a la función altaUsuario (que está en gestionarUsuarios.php) y si no ha ido bien pues nos da la opción de volver al formulario.

### 3.11 excepcion.php y paginacionConsulta.php

Estas no se tocan, se copian y se pegan. Esto me lo dieron tal como está.

La idea principal es que en excepcion.php se mostrarían los errores en caso de error al conectar con la BBDD. Y paginacionConsulta es lo necesario para que el mostrar "x" artículos en consultaArticulos.php sea viable.