

PROCEDIMIENTOS ALMACENADOS



Y

SESIÓN PHP



Daniel Sebastián, Andrés Pincheira y Sebastián Morales

HOY VEREMOS:

1

Manejo de sesión en php

3

Procedimientos almacenados

2

Conexión entre bases de datos

4

Temas adicionales para aportar
en su entrega

MANEJO DE SESIÓN EN PHP



¿QUÉ SON LAS SESIONES EN PHP?

"Las **sesiones**, en aplicaciones web realizadas con PHP y en el desarrollo de páginas web en general, nos sirven para **almacenar información** que se memorizará durante toda la visita de un usuario a una página web"



¿CÓMO **CREAR** UNA SESIÓN EN PHP?

```
<?php

// Llamar a la función al inicio del script
session_start();

// Almacenar variables de sesión en array global
$_SESSION
$_SESSION['id'] = '123';
$_SESSION['user_name'] = $_POST['user_name'];
?>
```

¿CÓMO **ELIMINAR** UNA SESIÓN EN PHP?

```
<?php
```

```
// Podemos desasignar variables con unset()  
unset($_SESSION['id'])
```

```
// Para destruir todo en esta sesión  
session_destroy();  
// Ahora $_SESSION estará vacía
```

```
?>
```

LOGIN

```
<?php

// Si no está asignada la variable mostrar form para ingresar
if(empty($_SESSION['username'])) {?>

<div>
  <form method="POST">
    <input type="text" name="username">
    <input type="password" name="password">
    <button type="submit" name="login">Login</button>
  </form>
</div>

<?php }?>
```

VISUALIZACIÓN

```
// Si ya está asignado el usuario, entonces se inicio sesión exitosamente
<?php if( isset($_SESSION['username'])) { ?>

<div>
    <h1>Hello <?php echo $_SESSION['username']; ?></h1>
</div>

// Desplegamos el botón de logout (cerrar sesión)
<form method="POST" >
    <button name="logout">Logout</button>
</form>

<?php }?>
```


LOGOUT

```
<?php

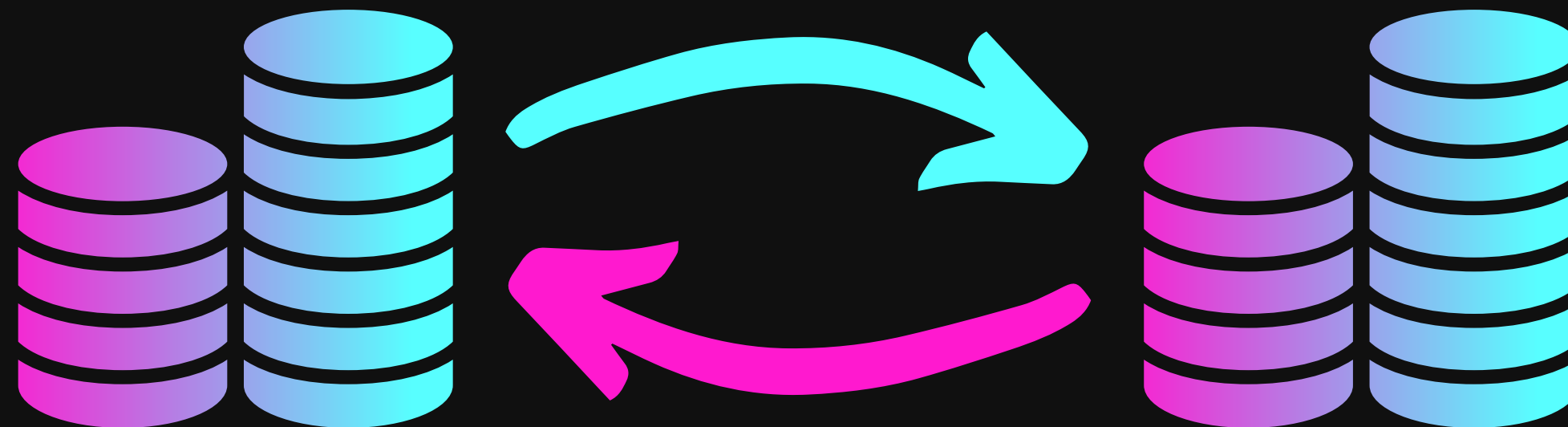
// Si se presiona el botón de logout...
if (isset($_POST['logout'])) {

    // ...destruimos la sesión...
    session_destroy();

    // ...y redireccionamos a la landing page.
    header("location: index.php");
}

?>
```

CONEXIÓN ENTRE BASES DE DATOS



CON: PHP



```
<?php

try {
    $user = 'grupoXX';
    $password = 'grupoXX';
    $databaseName = 'grupoXXe3';
    $db = new PDO("pgsql:dbname=$databaseName; host=localhost; port=5432; user=$user; password=$password");

    $user2 = 'grupoXX';
    $password2 = 'grupoXX';
    $databaseName2 = 'grupoXXe3';
    $db2 = new PDO("pgsql:dbname=$databaseName2; host=localhost; port=5432; user=$user2; password=$password2");
} catch (Exception $e) {
    echo "No se pudo conectar a la base de datos: $e";
}

?>
```

CON: DBLINK



```
SELECT * FROM dblink  
(  
  'dbname=grupoXXe3  
port=5432  
password=grupoXX  
user=grupoXX', 'SELECT * FROM TABLE'  
)  
AS name(col1 type1, col2 type2);
```

PROCEDIMIENTOS

ALMACENADOS

$$f(x)$$

¿QUÉ SON?

"Un **procedimiento almacenado** se puede definir como un **programa, procedimiento ó función**, el cual está almacenado en la base de datos y listo para ser usado."

¿QUÉ VENTAJAS TIENEN?

1) La ejecución del procedimiento ocurre en el **servidor
de bases de datos.**

¿QUÉ VENTAJAS TIENEN?

2) Al tener la lógica de la aplicación implementada en la base de datos **no tendremos que implementarla en los clientes.**

¿EN QUÉ LENGUAJE SE ESCRIBEN?

Un procedimiento almacenado en PostgreSQL se puede escribir en **múltiples lenguajes de programación**.

El único lenguaje que está disponible automáticamente es **PL/pgSQL** y es el que usaremos para esta ayudantía.

(<https://e-mc2.net/blog/procedimientos-almacenados-y-plpgsql/>)

SINTAXIS

```
CREATE OR REPLACE FUNCTION <nombre_funcion> (<argumentos>)  
  
-- declaramos lo que retorna  
RETURNS <tipo de dato> AS $$  
  
DECLARE  
-- declarar variables a utilizar si es necesario  
<variable1>;  
  
-- inicio función  
BEGIN  
  
... -- sentencias SQL  
  
-- término de función  
END  
$$ language plpgsql
```

SINTAXIS

```
BEGIN
```

```
-- consultas
```

```
SELECT * FROM TABLE;
```

```
END;
```

```
$$ language plpgsql;
```

```
BEGIN
```

```
-- condicionales
```

```
IF condicion THEN
```

```
ELSE
```

```
END IF;
```

```
END;
```

```
$$ language plpgsql;
```

```
BEGIN
```

```
-- ciclos
```

```
FOR condicion LOOP
```

```
END LOOP;
```

```
WHILE condicion LOOP
```

```
END LOOP;
```

```
END;
```

```
$$ language plpgsql;
```

¿CÓMO SE EJECUTA?

- 1) Subir archivo .sql al servidor
- 2) Cargar la función con `\i nombre_procedimiento.sql`
- 3) Ejecutar

SELECT nombre_procedimiento();

- 4) Si se desea eliminar,

DROP FUNCTION nombre_procedimiento;

EXTRAS

LIBRERÍAS PARA AGILIZAR EL ESTILO DE SUS PÁGINAS...

Bulma: <https://bulma.io/>

Bootstrap: <https://getbootstrap.com/>

