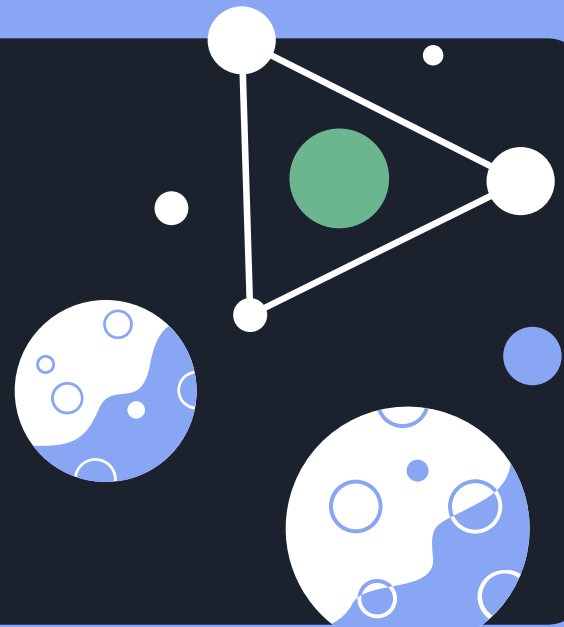


Ayudantía 4

IIC2413

2023-1





Resultados de aprendizaje

Al final de esta ayudantía, ustedes serán capaces de:

id	text
1	Crear y eliminar una sesión para la página de su proyecto
2	Conectar dos bases de datos distintas en su página a través de 2 métodos distintos
3	Comprender lo que es un procedimiento almacenado y su utilidad en su proyecto
4	Hacer sus propios procedimientos almacenados para su proyecto



Qué veremos hoy

01

Manejo de sesión

02

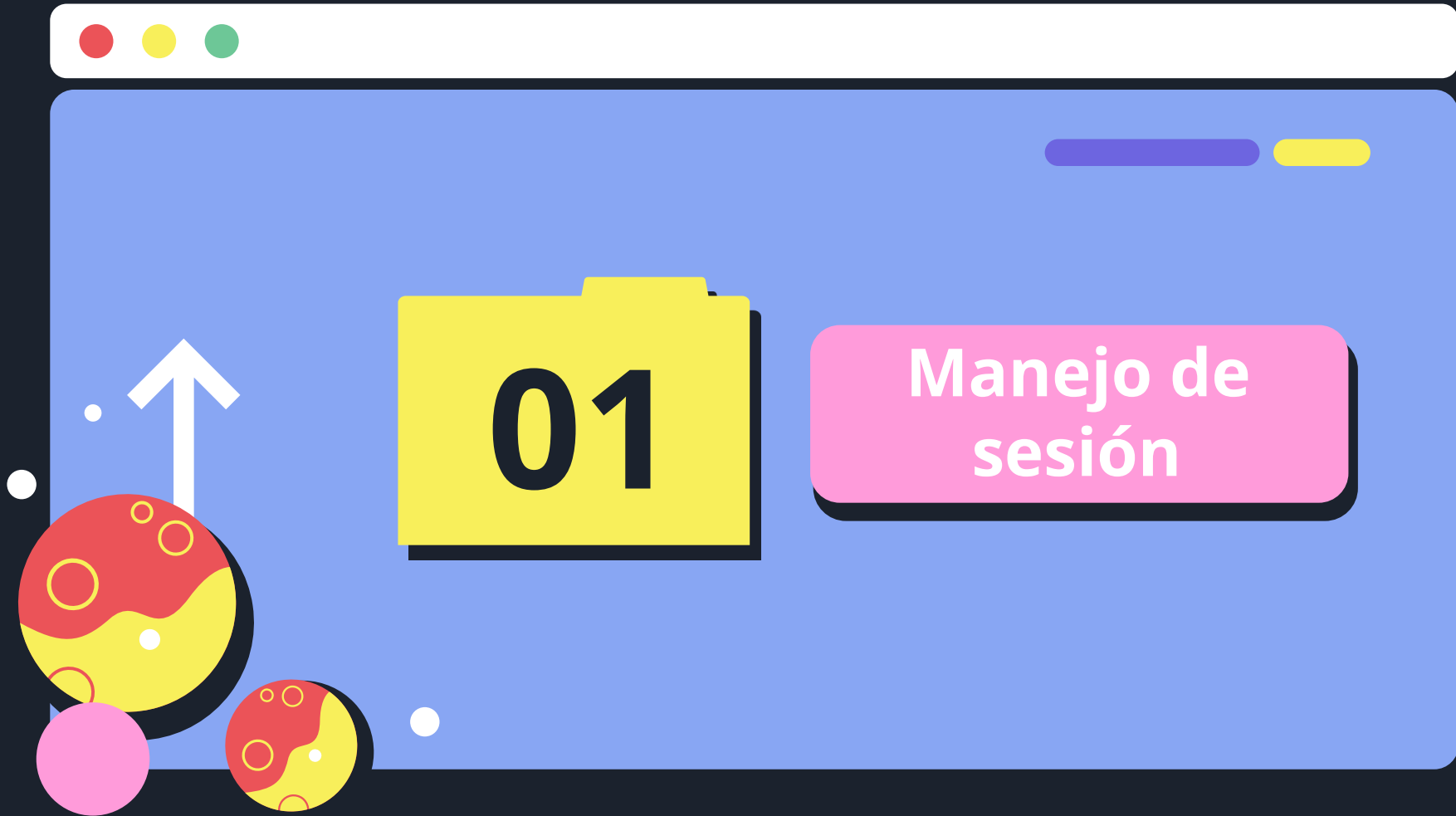
Procedimientos
almacenados

03

Ejemplos

04

Conexión entre
BDDs





¿Qué es una sesión?

Es un mecanismo que permite compartir información entre las diferentes páginas de un único sitio web o app, logrando que el servidor reconozca todas las peticiones que se originan desde un mismo usuario.



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 }?>
```

Crear sesión

```
<?php
// Llamar a la función al inicio del script
session_start();
// Almacenar variables de sesión en array global
$_SESSION

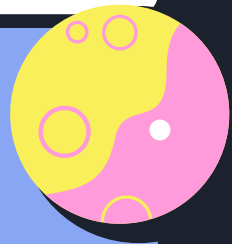
$_SESSION['ID'] = '123';
$_SESSION['username'] = $_POST['username']
...
```

Eliminar sesión

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

// Para destruir todo en esta sesión
session_destroy( ) ;

// Ahora $_SESSION estará vacía
?>
```





Visualización



```
<?php
    if (isset($_SESSION('username'))) { ?>
        <div>
            <h1> Hello <?php echo $_SESSION['username ']; ?></h1>
        </div>

        <form method="POST" >
            <button name="Logout">Logout</button>
        </form>
    <?php }?>
```


Logout

```
<?php
```

```
    if (isset($_POST['logout'])) {
```

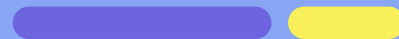
```
        session_destroy( );
```

```
        header("location: index. php");
```

```
    }
```

```
?>
```





Procedimientos
almacenados



¿Procedimientos alma qué? 🤯

Son **funciones en SQL** que se guardan en su DBMS para ejecutar lógica compleja. Resultan muy útiles y se pueden usar en **consultas**.





¿Cómo se ven?

Se trabajan en archivos con extensión **.sql**

My_procedure.sql

```
CREATE OR REPLACE FUNCTION  
get_name(int id)  
RETURNS varchar AS $$  
DECLARE  
Id INT  
  
BEGIN  
SELECT name FROM table WHERE id=  
END  
$$ language plpgsql
```

Sintaxis

CREATE OR REPLACE FUNCTION

<nombre_funcion> (<argumentos>)

RETURNS <tipo de dato> AS \$\$ -- declaramos lo que retorna

DECLARE

<variable_1>

<variable_2>

...

BEGIN

...

END

\$\$ language plpgsql

-- declaramos variables a utilizar

-- en caso de ser necesario

-- inicio de función

-- sentencias SQL

-- finalización de la función

Sintaxis

```
BEGIN
  IF condicion THEN
    ...
  ELSE IF otra condicion THEN
    ...
  ELSE
    ...
  END IF;

  FOR variable IN iterable
  LOOP
    ...
  END LOOP;

  RETURN valor -- ejemplo de valor→ SELECT * FROM TABLE;
END
```

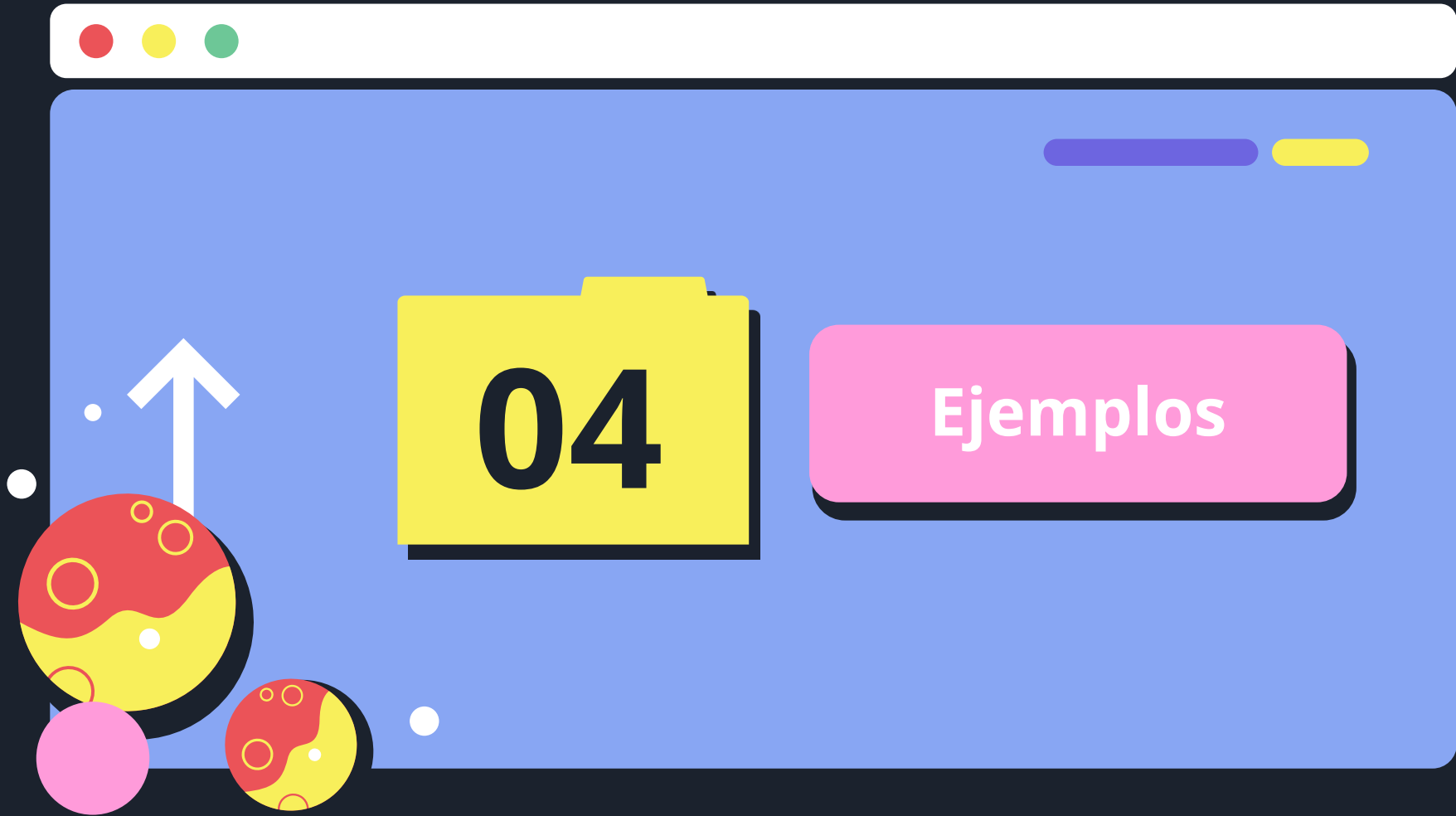




¿Cómo ejecutarlos?

1. Subir archivo .sql a servidor
2. Cargar la función con `\i nombre_procedimiento.sql`
3. Ejecutar `SELECT nombre_procedimiento(args);`
4. Si se desea eliminar, ejecutar `DROP FUNCTION nombre_procedimiento;`









Conexión entre BDDs

Método 1: Por php

Revisar codigo de ayudantia.

Code snippet en php para crear variables a partir de las bases de datos.






Conexión entre BDDs

Método 2: *dblink*

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



Conexión entre BDDs

Método 2: *dblink*

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

← estructura





Conexión entre BDDs

Método 2: *dblink*

Implementémoslo 

(El código estará disponible en el repo del curso)



Actividad práctica



Más recursos

- [Introduction to PostgreSQL PL/pgSQL Procedural Language](#)

