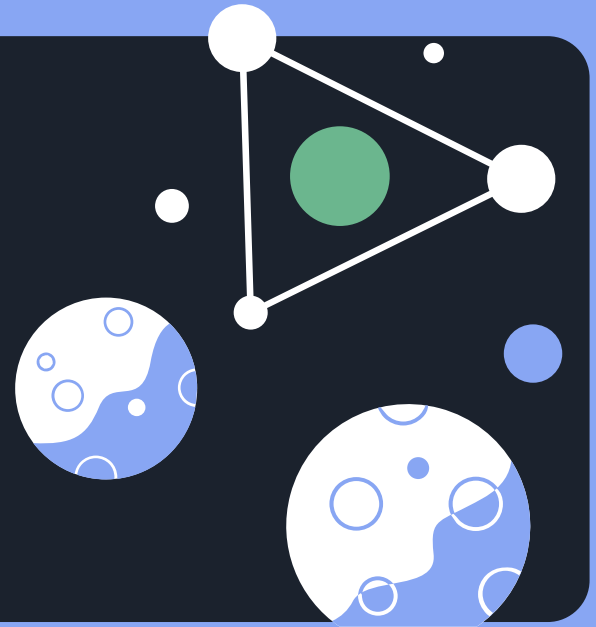


Ayudantía 4

IIC2413

2023-2





Qué veremos hoy

01

Manejo de sesión

02

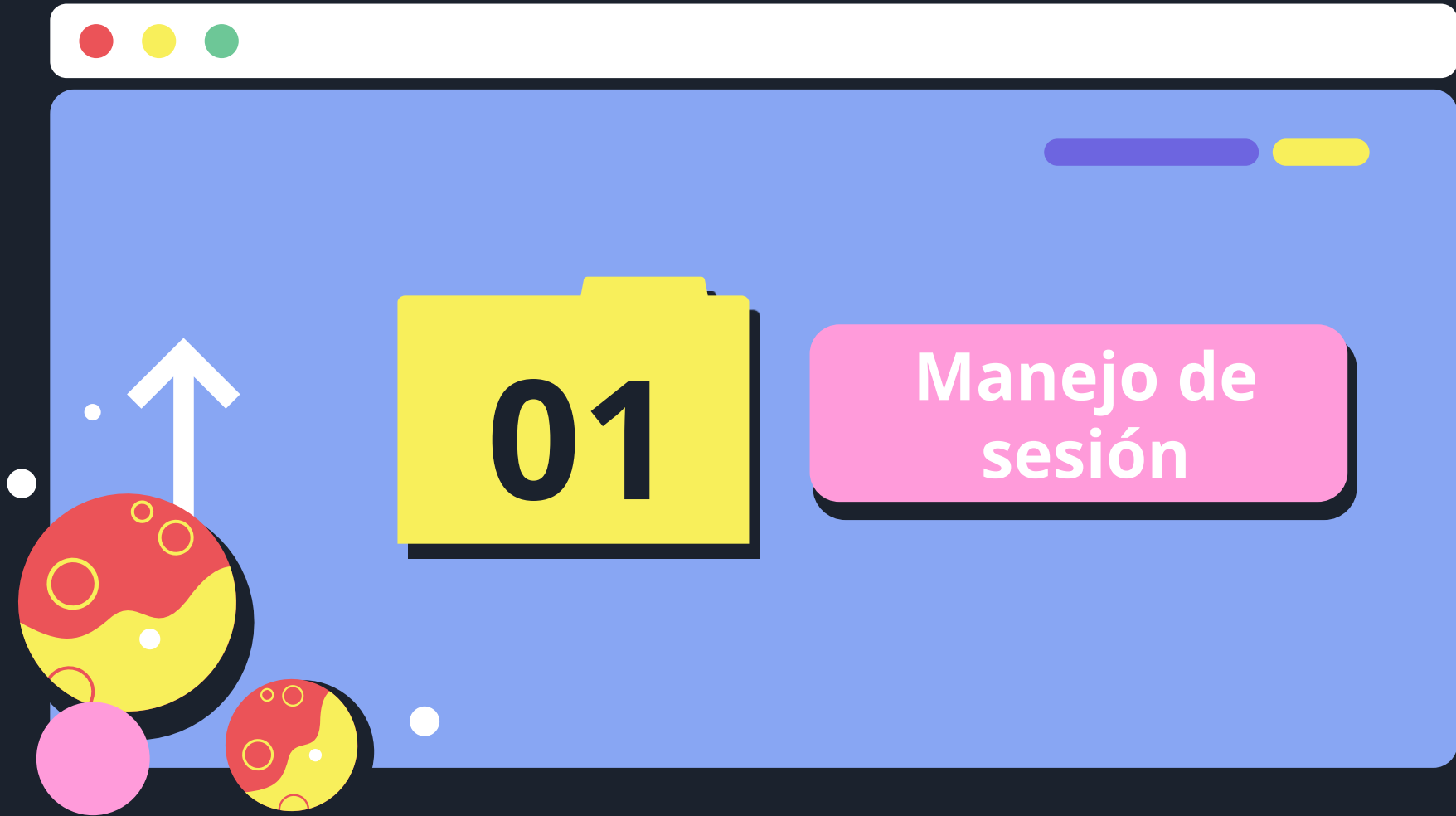
Procedimientos
almacenados

03

Ejemplos

04

Conexión entre
BDDs



Manejo de sesión



¿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.

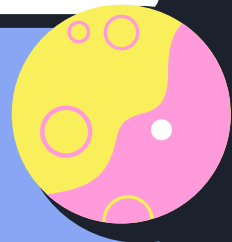




Login

```
<?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 Eliminar 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']
...
```

```
...
// 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_pro ced ure .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 - declaramos variables a utilizar

<variable_1> - en caso de ser necesario

<variable_2>

...

BEGIN - inicio de función

... - sentencias SQL

END - finalización de la función

\$\$ language plpgsql

Sintaxis

```
BEGIN  
IF condicion THEN
```

```
...
```

```
    ELSE IF otra condicion THEN
```

```
...
```

```
ELSE
```

```
...
```

```
END IF;
```

```
FOR variable IN iterable
```

```
LOOP
```

```
...
```

```
END LOOP;
```

```
RETURN valor
```

```
END
```

-- ejemplo de valor → SELECT * FROM TABLE;





¿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 código de ayudantía.
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



Actividad práctica

