



BASE DE DATOS EN PHP

DESARROLLO WEB EN ENTORNO  
SERVIDOR

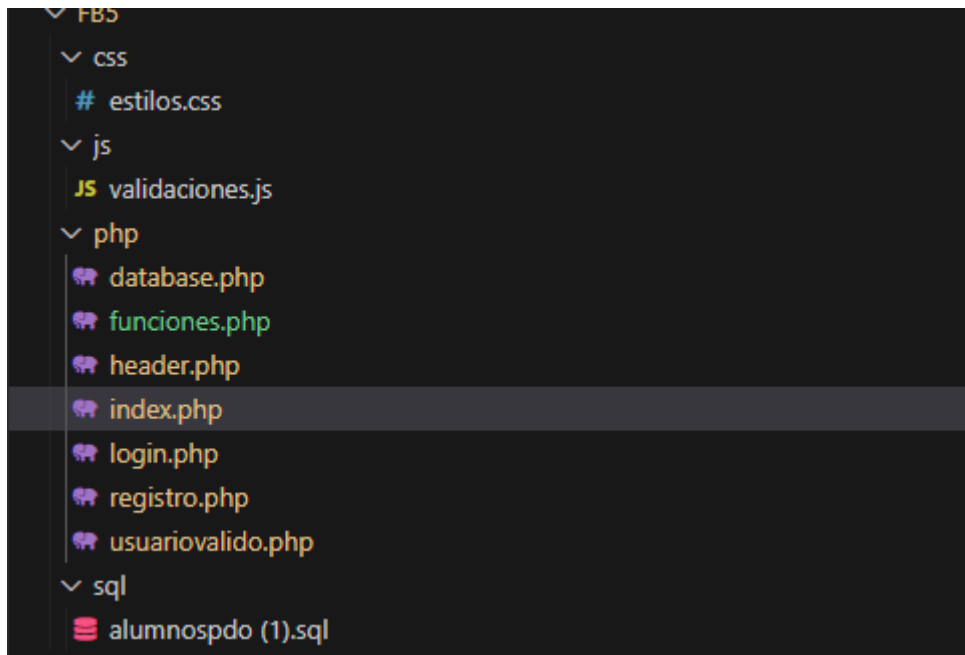
Feedback unidad 5

David Herrero Estévez

## Contenido

Estructura de directorios.....	3
Descripción de páginas php .....	3
Database.php .....	3
Header.php.....	3
Index.php.....	4
Login.php.....	6
Registro.php .....	8
Usuariovalido.php .....	10
Validaciones.js .....	11
Mensajes de error .....	16
Mensaje de éxito .....	17
Mensajes de error .....	19
En caso de introducir un usuario que no existe .....	19
En caso de introducir contraseña incorrecta .....	19
En caso de introducir usuario y contraseña válidos.....	19
Código fuente.....	19
Diagrama de flujo .....	20

## Estructura de directorios



## Descripción de páginas php

### Database.php

```
<?php
// Connect to database
    // $servername = "localhost";
    $servername="192.168.1.134";
    $username = "david";
    // $password = "aXVAcMv5k081LX6";
    $password="MBcaYoZE1G[cRok6";

    $dbname = "alumospdo";

?>
```

Contiene los datos de conexión de la base de datos.

### Header.php

```
<head>
<meta charset="utf-8">
<title>Feedback5</title>
```

```

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min
.css" rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
crossorigin="anonymous">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundl
e.min.js" integrity="sha384-
C6RzsynM9kWDrmNeT87bh950GNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
<script src="https://code.jquery.com/jquery-3.7.1.slim.js"
integrity="sha256-UgvvN8vBkg00luPSU12s8TI10SYRoGFAX4jlCIm9Adc="
crossorigin="anonymous"></script>
<link rel="stylesheet" href="../css/estilos.css">
<script src="../js/validaciones.js"></script>
</head>

<?php
    $debug=false;
    session_start();
?>

```

Contiene el contenido del header común a todo el “sitio” web. Básicamente contiene las librerías , css y ficheros js.

## Index.php

```

<html>
<?php
    include_once("header.php");
    ?>
<script>

</script>

<body>
    <?php

        //miramos si ha eliminado la sesión
        if (isset($_SESSION) && isset($_SESSION['success']))
        {
            $session_id = session_id();

            if(isset($_REQUEST['borrarsesion']) &&
$_REQUEST['borrarsesion']==$session_id){
                session_destroy();
                unset($_SESSION['success']);
                unset($_SESSION['borrarsesion']);
            }
        }
    }

```

```

        ?>

        <p>Ya está identificado con el siguiente email
[<?=$_SESSION["email"]?>] </p>
        <br>
        <hr>
        <a
href="<?=$_SERVER['PHP_SELF']?>?borrarsesion=<?=$session_id;?>">Borrar
sesión</a>
        <?php

}else{
    ?>
    <div class="container">
        <form action="login.php" method="post" onsubmit="return
validateLoginForm()">
            <div class="formulario">
                <label for="email" class="form-label">Email</label>
                <input type="email" class="form-control" id="email"
name="email" placeholder=''><br />
                <label for="password" class="form-label">Password</label>
                <input type="password" class="form-control" id="password"
name="password" placeholder=''><br />
                <input type="submit" class="btn btn-primary"
value="Login">
                <input type="reset" class="btn btn-primary"
value="Reset">
                <br />
                <a href="registro.php">No estoy registrado. Ir al
registro</a>
            </div>
        </form>
    </div>
    <?php
    }
    ?>
</body>
</html>

```

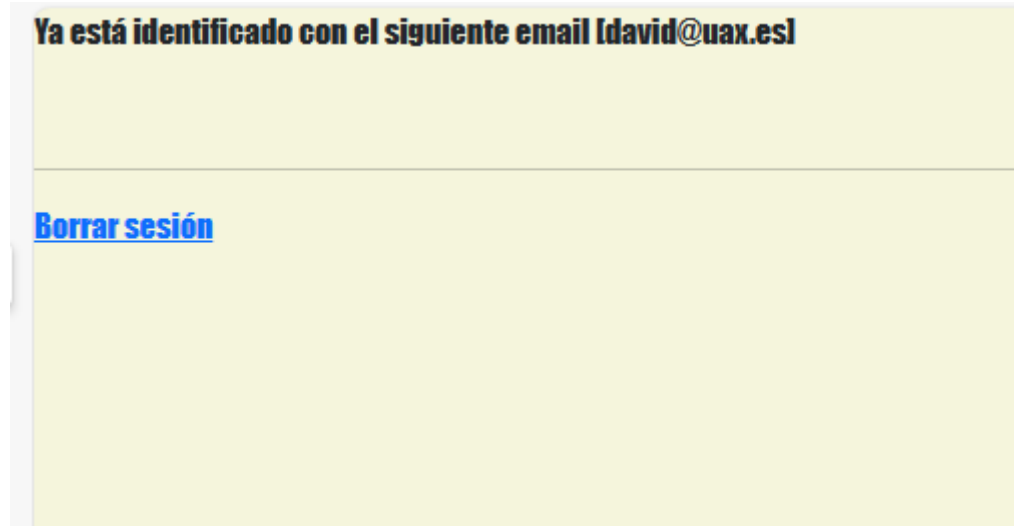
Contiene el formulario de login

El action del formulario va a la página de login, que es donde se ejecuta la lógica.

La usar la etiqueta de HTML5 email para el campo de texto, se está verificando que el email sea correcto.

En esta página tenemos una condición para ver si el usuario ya se ha identificado.

En ese caso mostramos un mensaje



El enlace a borrar sesión nos redirige a la misma página, pasando en la url el parámetro "borrarsesión"

Al inicio de la página comprobamos si queremos borrar la sesión y si la sesión coincide con la sesión actual.

En caso afirmativo borramos la sesión:

```
$session_id = session_id();

    if(isset($_REQUEST['borrarsesion']) &&
$_REQUEST['borrarsesion']==$session_id){
    session_destroy();
    unset($_SESSION['success']);
    unset($_SESSION['borrarsesion']);
}
```

Login.php

```
<html>
<?php
    include_once("header.php");
    include_once("funciones.php");
    ?>

<body>
    <?php
if ($_SERVER["REQUEST_METHOD"] == "POST" && $debug) {
    echo "<table>";
    foreach ($_POST as $key => $value) {
```

```

        echo "<tr><td><b>$key</b></td><td>$value</td></tr>";
    }
    echo "</table>";
}

// Check connection
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = $_POST["email"];
    $formpassword = htmlspecialchars($_POST["password"]);
    require_once ("database.php");
    // Connect to the database
    try {
        $conn = new PDO("mysql:host=$servername;dbname=$dbname",
$username, $password);
        $conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
    } catch(PDOException $e) {
        echo "<div class='error'>Falló la conexión " . $e-
>getMessage() . "</div>";
        exit;
    }

    // Prepare and execute the query
    $stmt = $conn->prepare("SELECT * FROM usuarios WHERE email = ?");
    // echo $email;
    $stmt->execute([$email]);
    $result = $stmt->fetch();

    // Check if the user exists
    if ($result) {
        $dbpassword=$result['password'];

        // Verify the password
        if (coincidenContrasenias($formpassword, $dbpassword)) {
            $_SESSION["success"] = "Login successful!";
            $_SESSION["email"]=$email;
            header("Location: usuariovalido.php"); // Redirect to
the dashboard page
            exit;
        } else {
            echo "<div class='error'>Contraseña no válida</div>";
            // Return an error message
        }

    } else {

```

```

        echo "<div class='error'>Usuario no encontrado</div>"; //
Return an error message
    }

    // Close the connection
    $conn = null;
}
?>
<a href="index.php">Volver</a>

</body>

</html>

```

La página de login primero comprueba que se haya enviado el formulario. En caso afirmativo realiza una consulta a la base de datos obteniendo usuario con ese mismo correo.

Una vez tenemos el usuario comprobamos si la contraseña encriptada es correcta o no.

La página cuenta con enlaces a la página de inicio.

En caso de que el usuario sea correcto, iniciamos la sesión y guardamos el par de clave valor success = "Login successful"

## Registro.php

```

<html>
<?php
    include_once("header.php");
    ?>
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    echo "<table>";
    foreach ($_POST as $key => $value) {
        echo "<tr><td><b>$key</b></td><td>$value</td></tr>";
    }
    echo "</table>";
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = $_POST["email"];
    $password = $_POST["password"];
    $passwordconfirm = $_POST["passwordconfirm"];

    // Validate input
    if (empty($email) || empty($password) ) {
        echo "<div class='error'>Error: Todos los campos son
obligatorios.</div>";
    }
}
}
?>

```



```

    } elseif ($password != $passwordconfirm) {
        echo "<div class='error'>Error: Las contraseñas no coinciden.</div>.";
    } else {
        require_once ("database.php");

        try {
            $conn = new PDO("mysql:host=$servername;dbname=$dbname",
$username, $password);
            // set the PDO error mode to exception
            $conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);

            // Check if user already exists
            $stmt = $conn->prepare("SELECT * FROM usuarios WHERE
email=:email");
            $stmt->bindParam(":email", $email);
            $stmt->execute();
            $result = $stmt->fetch(PDO::FETCH_ASSOC);

            if ($result) {
                echo "<div class='error'>Error: El usuario ya
existe.</div>.";
            } else {
                // Create user
                $stmt = $conn->prepare("INSERT INTO usuarios (email,
password) VALUES (:email, :password)");
                $stmt->bindParam(":email", $email);
                $hashed_password = password_hash($password,
PASSWORD_DEFAULT);
                $stmt->bindParam(":password", $hashed_password);

                $stmt->execute();
                echo "<div class='green'>Usuario creado
correctamente.</div>.";
            }
        } catch(PDOException $e) {
            echo "<div class='error'>Error: ". $e->getMessage()
."</div>" ;
        }
        $conn = null;
    }
}
?>
<body>
    <div class="container">
        <div class="titulo">Formulario de registro</div>

```

```

        <form action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post"
onsubmit="return validateRegisterForm()">
            <div class="formulario">
                <label for="email" class="form-label">Email</label>
                <input type="email" class="form-control" id="email"
name="email" placeholder=''><br />
                <label for="password" class="form-
label">Contraseña</label>
                <input type="password" class="form-control"
id="password" name="password" placeholder=''><br />
                <label for="passwordconfirm" class="form-
label">Confirmar Contraseña</label>
                <input type="password" class="form-control"
id="passwordconfirm" name="passwordconfirm" placeholder=''><br />
                <input type="submit" class="btn btn-primary"
value="Registro">
                <input type="reset" class="btn btn-primary"
value="Reset">
                <br />
                <a href="index.php">Volver</a>
            </div>
        </form>
    </div>
</body>
</html>

```

La página de registro contiene un formulario con validaciones HTML5 así como mediante la función de javascript validateRegisterForm.

El action del formulario va a la misma página mediante el uso de la supervariable `$_SERVER['PHP_SELF']`

Si la página detecta que estamos pasando el email y la contraseña + la contraseña repetida, trata de crear el usuario.

Si el email ya existe mostramos un mensaje de error diciendo que el usuario ya existe. En caso contrario creamos el usuario.

En caso de alguna excepción mostramos el mensaje devuelto por la excepción.

### Usuariovalido.php

```

<html>
<?php
    include_once("header.php");
    session_start();

    if (!isset($_SESSION['success'])) {
        header("Location: index.php");
        exit();
    }

```

```

    }
    ?>
<script>

</script>
<body>
    <div class="container">
        <h1>Usuario válido</h1>
        <a href="index.php">Volver</a>
    </div>
</body>

</html>

```

Esta página es a la que se accede cuando el login es correcto. La página comprueba que exista la sesión y contenga la clave "success". En caso contrario redirige a la página de login.

### Validationes.js

```

function validateLoginForm(){
    var email = document.getElementById("email").value;
    var password = document.getElementById("password").value;
    if (email == "" || password == ""){
        alert("Debe completar todos los campos");
        return false;
    }
    return true;
}

function validateRegisterForm() {

    var email = document.getElementById("email").value;
    var password = document.getElementById("password").value;
    var passwordconfirm =
document.getElementById("passwordconfirm").value;

    if (email == "") {
        alert("El email no puede estar vacío");
        return false;
    }
    if (password == "") {
        alert("La contraseña no puede estar vacía");
        return false;
    }
    if (passwordconfirm == "") {
        alert("La confirmación de la contraseña no puede estar vacía");
        return false;
    }
}

```

```

    if (password.length < 6) {
        alert("El tamaño de la contraseña no puede ser menor de 6 caracteres");
        return false;
    }
    if (password != passwordconfirm) {
        alert("Las contraseñas no coinciden");
        return false;
    }
    return true;
}

```

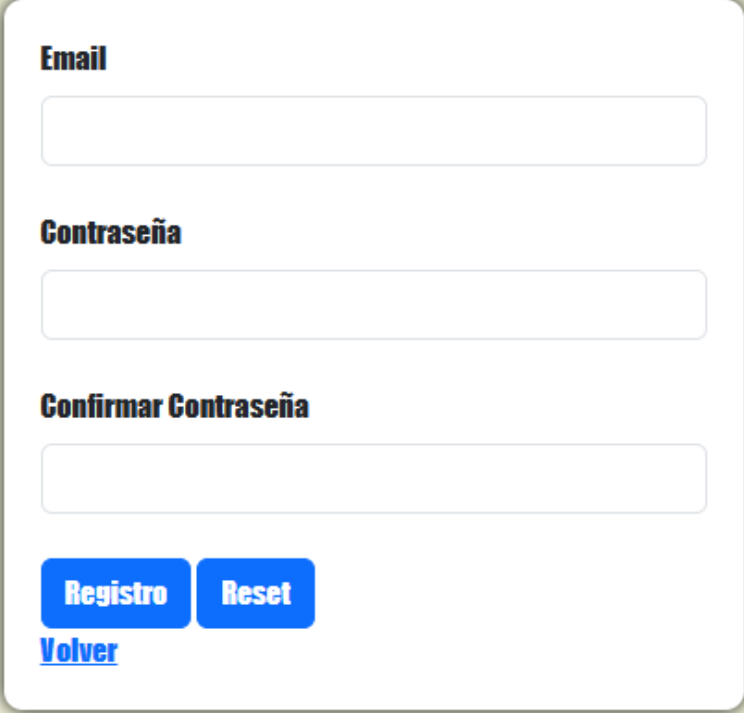
Fichero javascript que contiene dos funciones para validar los dos formularios.

En el validatelogin comprueba que el email y la password contenga un valor.

En el validateregister valida que el email no esté vacío, que la contraseña no esté vacía, que las contraseñas coincidan y que la longitud de la contraseña sea mayor que 6.

Funcionamiento de páginas

Registro



The image shows a registration form with a light green background. The form is a white rounded rectangle containing three input fields and three buttons. The first field is labeled 'Email', the second 'Contraseña', and the third 'Confirmar Contraseña'. Below the fields are two blue buttons labeled 'Registro' and 'Reset', and a blue link labeled 'Volver'.

Validaciones y errores

Convi

**localhost dice**

El email no puede estar vacío

Aceptar



PER



Quads

**Contraseña**

**Confirmar Contraseña**

**Registro**

**Reset**

[Volver](#)

Email

davidtfg



Incluye una '@' en la dirección de correo electrónico. A 'davidtfg' le falta una '@'.

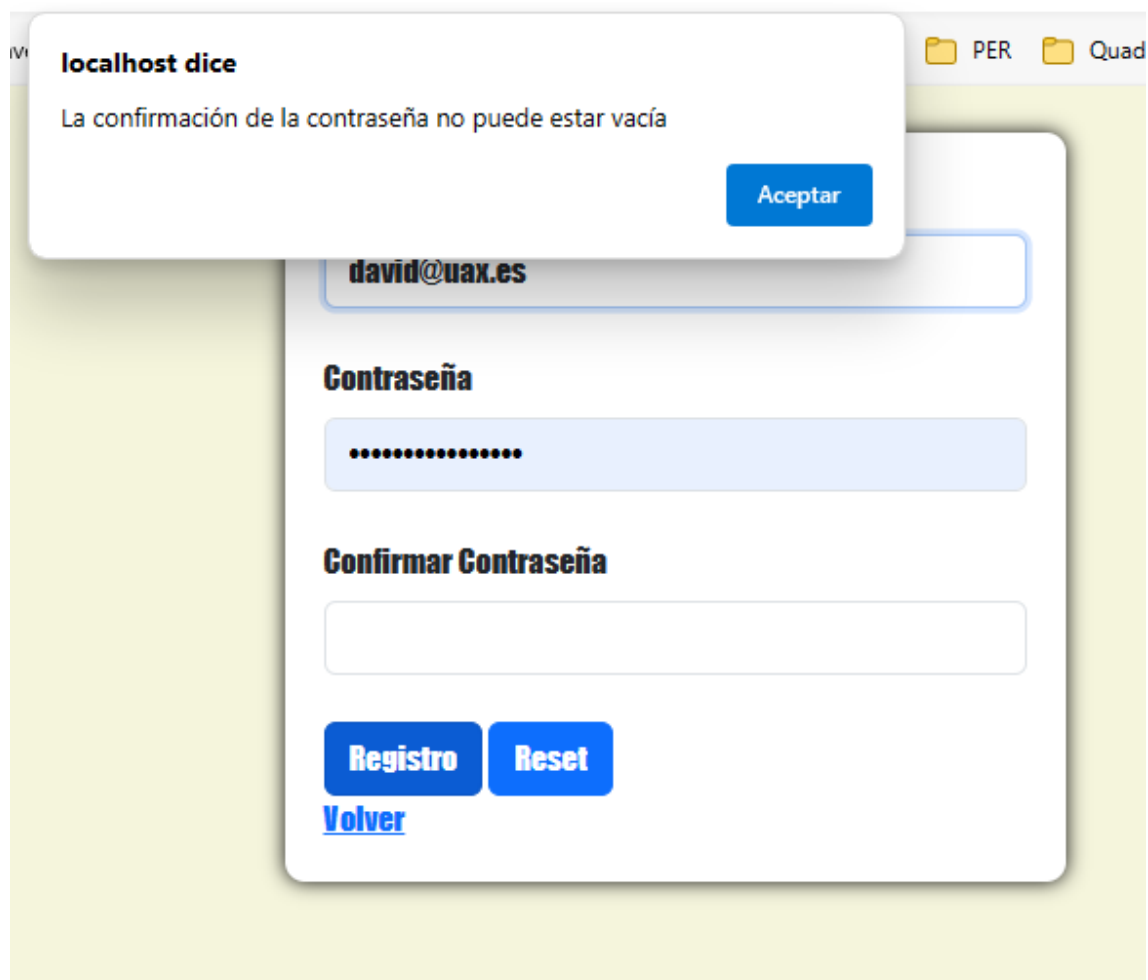
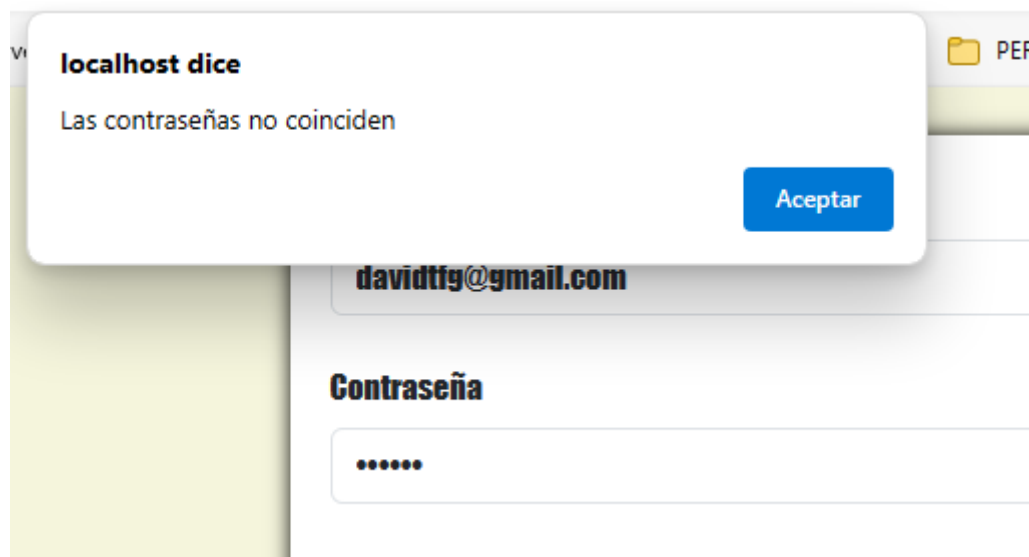
.....

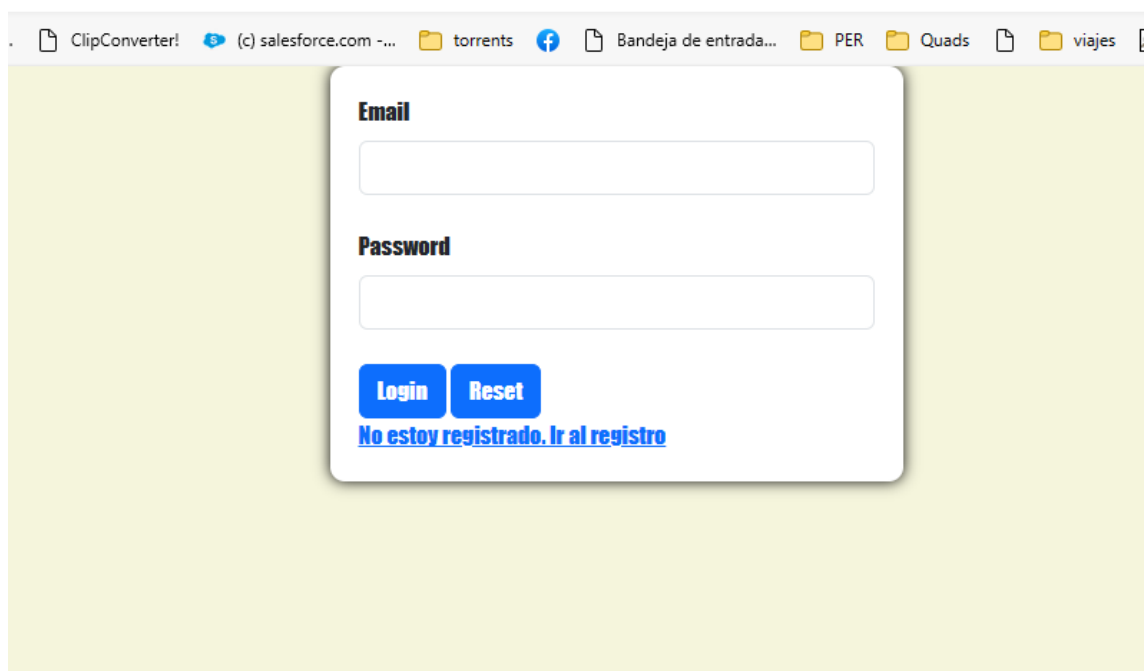
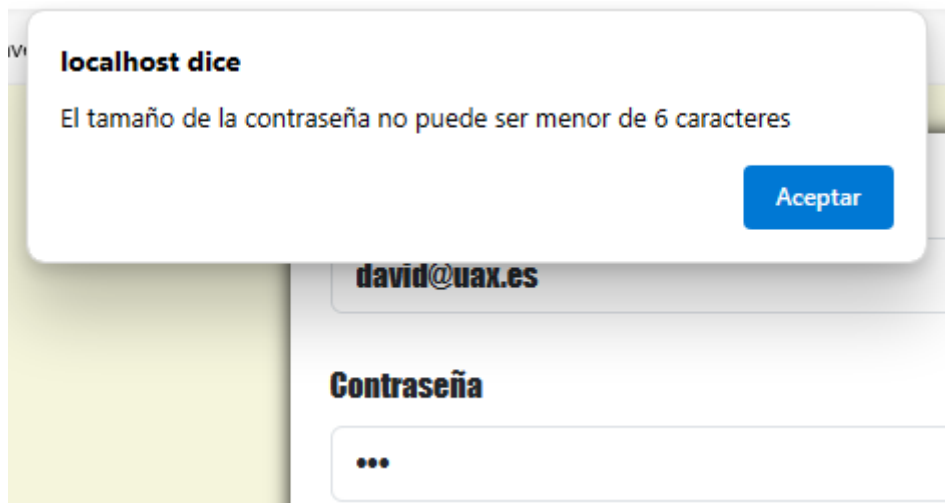
Confirmar Contraseña

Registro

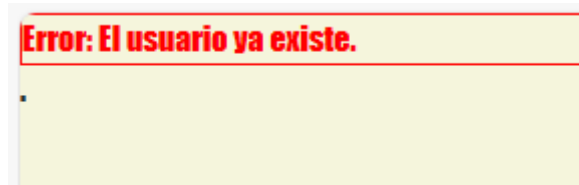
Reset

[Volver](#)





## Mensajes de error

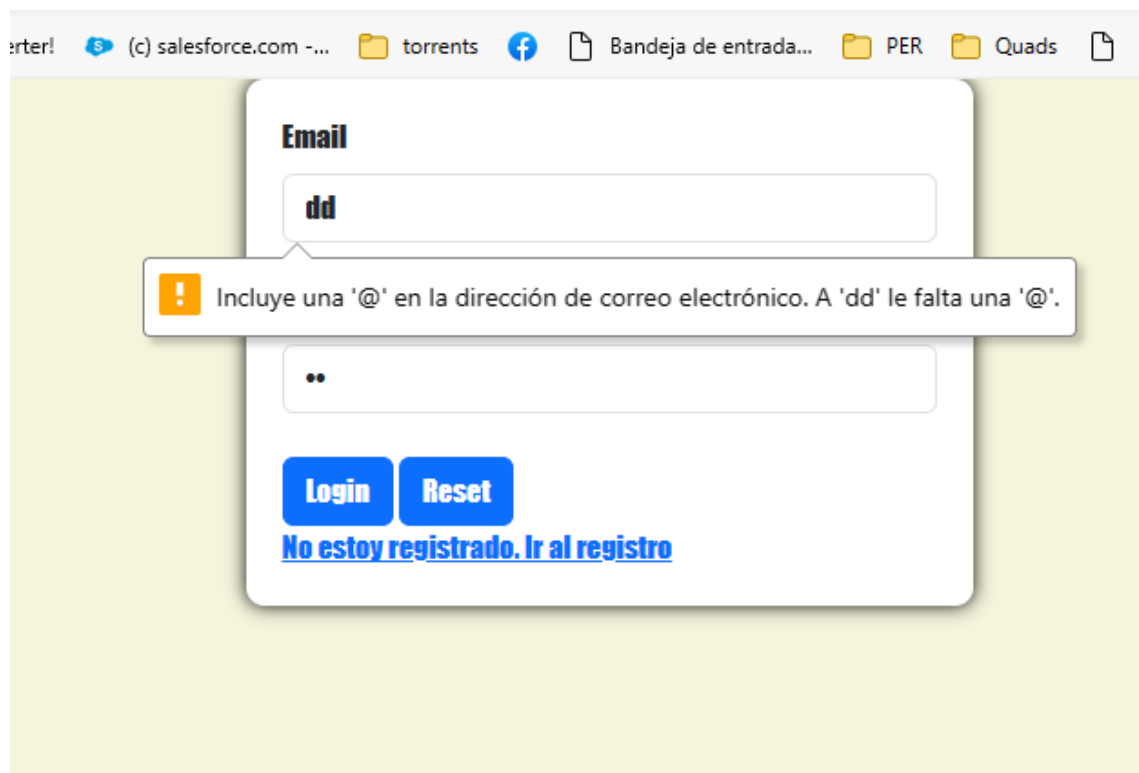
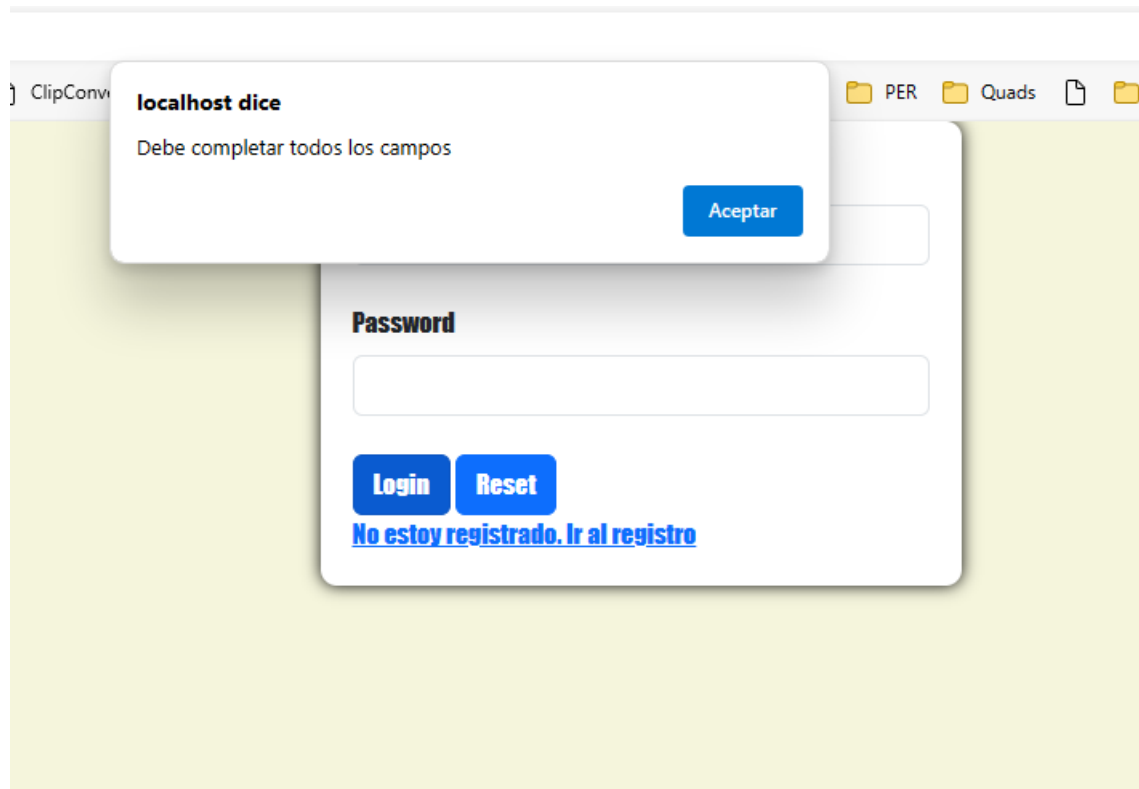




Mensaje de éxito

**Usuario creado correctamente.**

.



## Mensajes de error

En caso de introducir un usuario que no existe

**Usuario no encontrado**

[Volver](#)

En caso de introducir contraseña incorrecta

**Contraseña no válida**

[Volver](#)

En caso de introducir usuario y contraseña válidos

**Usuario válido**

[Volver](#)

## Código fuente

[EntornoServidor/FB5 at main · SFDCdavidesi/EntornoServidor \(github.com\)](https://github.com/SFDCdavidesi/EntornoServidor)

Diagrama de flujo

