

PROYECTO POKEMON SERVIDOR



Marcos Robador Mateos

En nuestra base de datos pokemon que acabamos de crear, vamos a crear la tabla pokemons con los siguientes campos:

- id [*]
- name
- type
- subtype
- region
- Acuérdate que [*] significa que es clave primaria y no olvides poner el tipo de datos de cada uno de los campos.
- Rellena la tabla pokemons con, al menos 5 registros, con los datos que tú quieras o si lo prefieres, puedes basarte en la página oficial del juego.

```
CREATE TABLE pokemons (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255),
    type ENUM('Fire', 'Water', 'Grass', 'Electric', 'Psychic', 'Normal', 'Ice', 'Dragon', 'Dark', 'Fairy', 'Fighting', 'Flying', 'Poison', 'Ground'),
    subtype ENUM('Fire', 'Water', 'Grass', 'Electric', 'Psychic', 'Normal', 'Ice', 'Dragon', 'Dark', 'Fairy', 'Fighting', 'Flying', 'Poison', 'Ground'),
    region ENUM('Kanto', 'Johto', 'Hoenn', 'Sinnoh', 'Unova', 'Kalos', 'Alola', 'Galar', 'Hisui')
);
```

- Crea el archivo 001pkms.php donde listes todos los pokemon que has metido en tu base de datos. Acuérdate que para ello deberás hacer una conexión con la base de datos y un foreach para cada Pokémon que tengas albergado en la tabla champ.

```
INSERT INTO pokemons (name, type, subtype, region) VALUES
('Bulbasaur', 'Grass', 'Poison', 'Kanto'),
('Charmander', 'Fire', '', 'Kanto'),
('Squirtle', 'Water', '', 'Kanto'),
('Pikachu', 'Electric', '', 'Kanto'),
('Jigglypuff', 'Normal', 'Fairy', 'Kanto'),
('Meowth', 'Normal', '', 'Kanto'),
('Psyduck', 'Water', '', 'Kanto'),
('Geodude', 'Rock', 'Ground', 'Kanto'),
('Eevee', 'Normal', '', 'Kanto'),
('Snorlax', 'Normal', '', 'Kanto');
```

Pokemones

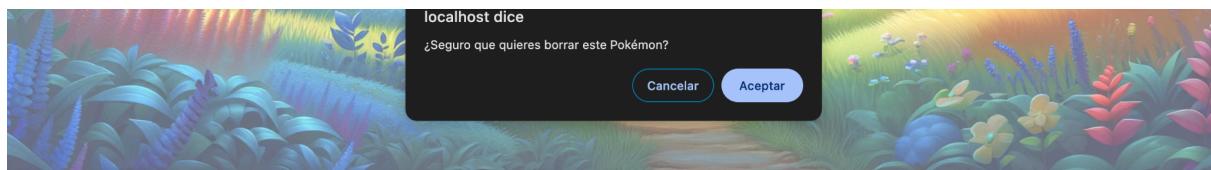
ID ▾	Nombre ▾	Tipo ▾	Subtipo ▾	Región ▾
1	Bulbasaur	Grass	Poison	Kanto
2	Charmander	Fire		Kanto
3	Squirtle	Water		Kanto
4	Pikachu	Electric		Kanto
5	Jigglypuff	Normal	Fairy	Kanto
6	Meowth	Normal		Kanto
7	Psyduck	Water		Kanto
8	Geodude	Rock	Ground	Kanto
9	Eevee	Normal		Kanto
10	Snorlax	Normal		Kanto

Añadir Pokémon

```
while ($pokemon = $result->fetch_assoc()) {
    echo "<tr>";
    echo "<td>" . htmlspecialchars($pokemon["id"]) . "</td>";
    echo "<td>" . htmlspecialchars($pokemon["name"]) . "</td>";
    echo "<td>" . htmlspecialchars($pokemon["type"]) . "</td>";
    echo "<td>" . htmlspecialchars($pokemon["subtype"]) . "</td>";
    echo "<td>" . htmlspecialchars($pokemon["region"]) . "</td>";
    echo "<td>";
```

Editar y borrar

ID ▾	Nombre ▾	Tipo ▾	Subtipo ▾	Región ▾	Acciones	
1	Bulbasaur	Grass	Poison	Kanto	<button>Editar</button>	<button>Borrar</button>
2	Charmander	Fire		Kanto	<button>Editar</button>	<button>Borrar</button>
9	Eevee	Normal		Kanto	<button>Editar</button>	<button>Borrar</button>
8	Geodude	Rock	Ground	Kanto	<button>Editar</button>	<button>Borrar</button>
5	Jigglypuff	Normal	Fairy	Kanto	<button>Editar</button>	<button>Borrar</button>
6	Meowth	Normal		Kanto	<button>Editar</button>	<button>Borrar</button>
4	Pikachu	Electric		Kanto	<button>Editar</button>	<button>Borrar</button>
7	Psyduck	Water		Kanto	<button>Editar</button>	<button>Borrar</button>
10	Snorlax	Normal		Kanto	<button>Editar</button>	<button>Borrar</button>
3	Squirtle	Water		Kanto	<button>Editar</button>	<button>Borrar</button>



Carlo Marcos ha sido introducido en el sistema con la contraseña Caracol.4

Pokemones

ID ▾	Nombre ▾	Tipo ▾	Subtipo ▾	Región ▾	Acciones	
1	Bulbasaur	Grass	Poison	Kanto	<button>Editar</button>	<button>Borrar</button>

EDITAR

```
include 'Conexion/conexion.php';

// Verifica si se recibe un ID y es numerico
if (isset($_GET['id']) && is_numeric($_GET['id'])) {
    $pokemonId = $_GET['id'];

    // Procesa el formulario de edicion
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        $name = $_POST['name'] ?? '';
        $type = $_POST['type'] ?? '';
        $subtype = $_POST['subtype'] ?? '';
        $region = $_POST['region'] ?? '';

        // Prepara consulta de actualizacion
        $stmt = $conexion->prepare("UPDATE pokemons SET name = ?, type = ?, subtype = ?, region = ? WHERE id = ?");
        $stmt->bind_param("ssss", $name, $type, $subtype, $region, $pokemonId);

        if ($stmt->execute()) {
            echo "Pokémon actualizado con éxito.";
            header("Location: 003pkms.php");
            exit;
        } else {
            echo "Error al actualizar el Pokémon.";
        }

        $stmt->close();
        $conexion->close();
        return;
    }
}
```

```

// obtiene datos del pokemon para el formulario, incluyendo la URL de la imagen
$stmt = $conexion->prepare("SELECT pokemons.*, imagenes_pokemon.url_imagen FROM pokemons LEFT JOIN imagenes_"
$stmt->bind_param("i", $pokemonId);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows === 1) {
    $pokemon = $result->fetch_assoc();
    ?>

<div class="form-editar" >
    <!-- Formulario de edicion -->
    <form action="editar-pokemon.php?id=<?php echo $pokemonId; ?>" method="post">

        <h3><?php echo htmlspecialchars($pokemon['name']); ?></h3>
        <input type="hidden" name="name" value="<?php echo htmlspecialchars($pokemon['name']); ?>">

        Tipo:
        <select name="type">
            <?php
                $tipos = ['Fire', 'Water', 'Grass', 'Electric', 'Psychic', 'Normal', 'Ice', 'Dragon', 'Dark', 'Fairy',
                foreach ($tipos as $tipo) {
                    echo "<option value=\"$tipo\"".($pokemon['type'] == $tipo ? ' selected' : '').">".ucfirst($tipo)."<
                }
            ?>
        </select><br>

        Subtipo:
        <select name="subtype">
            <option value="" <?php echo $pokemon['subtype'] == '' ? 'selected' : ''; ?>>Ninguno</option>
            <?php
                foreach ($tipos as $tipo) {
                    echo "<option value=\"$tipo\"".($pokemon['subtype'] == $tipo ? ' selected' : '').">".ucfirst($tipo)
                }
            ?>

```

BORRAR

```

<?php
include 'Conexion/conexion.php';
You, hace 5 días • archivos de editar y borrar y estilos
// Verifica si recibe un ID y es numérico
if (isset($_GET['id']) && is_numeric($_GET['id'])) {
    $pokemonId = $_GET['id'];

    // Primero, elimina registros relacionados en imagenes_pokemon
    $stmtImagenes = $conexion->prepare("DELETE FROM imagenes_pokemon WHERE pokemon_id = ?");
    $stmtImagenes->bind_param("i", $pokemonId);

    if (!$stmtImagenes->execute()) {
        echo "Error al borrar las imágenes asociadas al Pokémon.";
        $stmtImagenes->close();
        $conexion->close();
        exit;
    }

    $stmtImagenes->close();

    // Luego, elimina el registro en pokemons
    $stmtPokemon = $conexion->prepare("DELETE FROM pokemons WHERE id = ?");
    $stmtPokemon->bind_param("i", $pokemonId);

    if ($stmtPokemon->execute()) {
        echo "Pokémon borrado con éxito.";
        // Redirige después de borrar el Pokémon
        header("Location: 003pkms.php");
        exit;
    } else {
        echo "Error al borrar el Pokémon.";
    }

    $stmtPokemon->close();
} else {
    echo "ID no válido.";
    header("Location: 003pkms.php");
}
$conexion->close();
?>

```

- Modifica el archivo 001pkms.php y guárdalo como 003pkms.php para que se muestre como una tabla con las columnas de la propia tabla de la base de datos, es decir; id, nombre, etc. (Todos los campos)

ID ^v	Nombre ^v	Tipo ^v	Subtipo ^v	Región ^v	Acciones
1	Bulbasaur	Grass	Poison	Kanto	Editar Borrar

- Al lado de cada nombre de cada columna, pon 2 iconos que sean \wedge \vee y que cada uno de ellos ordene el listado en función de cuál se haya pinchado.
- Si se ha pulsado en Nombre el icono de \wedge , el listado debe aparecer ordenado por nombre ascendente. Si por el contrario se ha pulsado \vee tendrá que ordenarse por nombre descendente.
- Ten en cuenta que cada ícono debe llevar consigo un enlace al listado que contenga parámetros en la URL que satisfagan las opciones seleccionadas así que haced uso de `$_GET` para poder capturarlos y escribid las consultas SQL que sean necesarias para hacer cada uno de los filtros.



```
// Comprueba si 'ordenar_por' esta en la URL. Si esta presente, usa su valor
// Si no esta, establece 'name' como valor por defecto
$ordenar_por = isset($_GET['ordenar_por']) ? $_GET['ordenar_por'] : 'name';

// Comprueba si 'direccion' esta y es igual a 'desc'
// Si es asi, establece la variable $direccion a 'DESC'.
// Si no, establece 'ASC' (ascendente) como valor por defecto
$direccion = isset($_GET['direccion']) && $_GET['direccion'] == 'desc' ? 'DESC' : 'ASC';

// Crea la consulta SQL con los parametros de ordenacion
$sql = "SELECT id, name, type, subtype, region FROM pokemons ORDER BY $ordenar_por $direccion";
$result = $conexion->query($sql);

echo "<div class='pokemon-list'>";
echo "<h1>Pokemones</h1>";

if ($result->num_rows > 0) {
    echo "<div class='table-responsive'>";
    echo "<table class='table table-striped pokemon-table'>";
    echo "<thead>";
    echo "<tr>";
    echo "<th>ID <a href='?ordenar_por=id&direccion=asc'></a> <a href='?ordenar_por=id&direccion=desc'></a></th>";
    echo "<th>Nombre <a href='?ordenar_por=name&direccion=asc'></a> <a href='?ordenar_por=name&direccion=desc'></a></th>";
    echo "<th>Tipo <a href='?ordenar_por=type&direccion=asc'></a> <a href='?ordenar_por=type&direccion=desc'></a></th>";
    echo "<th>Subtipo <a href='?ordenar_por=subtype&direccion=asc'></a> <a href='?ordenar_por=subtype&direccion=desc'></a></th>";
    echo "<th>Region <a href='?ordenar_por=region&direccion=asc'></a> <a href='?ordenar_por=region&direccion=desc'></a></th>";
    echo "</th>Acciones</th>";
    echo "</tr>";
    echo "</thead>";
    echo "<tbody>";
```

Crea una tabla nueva dentro de la base de datos pokemon que ya tienes y crea un sistema de login con usuarios. Introduce en la base de datos al menos 3 usuarios diferentes con sus contraseñas distintas. Recuerda que:

La tabla nueva ha de llamarse users. Los campos por crear en la nueva tabla deben ser:

- id
- name
- username
- password
- email

```
CREATE TABLE usuarios (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(255),
    apellido VARCHAR(255),
    email VARCHAR(255),
    password VARCHAR(255)
);
```

Las contraseñas deben ser cifradas antes de guardar los datos en la base de datos.

```
$password_encriptada = password_hash($password, PASSWORD_DEFAULT);
```

Crea el formulario 005registro.php donde el usuario introduzca los datos de registro y vincúlalo con 006nuevoUsuario.php para que recoja los datos mediante POST y los inserte en la base de datos si todo ha ido bien.

Registro

Nombre:

Apellido:

Email:

Contraseña:

Escribe de nuevo la contraseña:

Registrar

¿Ya tienes una cuenta? [Iniciar sesión](#)

Iniciar Sesión

Email:

Contraseña:

Iniciar Sesión

¿No tienes una cuenta? [Regístrate](#)

```
<?php
include '../Conexion/conexion.php';

// Verifica si el formulario fue enviado
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Recopilación y saneamiento de los datos del formulario
    $nombre = $conexion->real_escape_string($_POST['nombre']);
    $email = $conexion->real_escape_string($_POST['email']);
    $password = $conexion->real_escape_string($_POST['password']);
    $apellido = $conexion->real_escape_string($_POST['apellido']);

    // Validar el formato del email
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        echo "Formato de correo electrónico inválido.";
        exit;
    }

    // Comprobar si el email ya está registrado
    $stmt = $conexion->prepare("SELECT COUNT(*) FROM usuarios WHERE email = ?");
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $stmt->bind_result($count);
    $stmt->fetch();
    $stmt->close();

    if ($count > 0) {
        echo "Este correo electrónico ya está registrado.<br>";
        echo "<button onclick=\"location.href='register.html'\">Registrar de Nuevo</button>";
        exit;
    } else { You, anteayer • Validacion de login. ...
        // El email es único, proceder con el registro
        $password_encriptada = password_hash($password, PASSWORD_DEFAULT);
        $consulta = "INSERT INTO usuarios (nombre, email, password, apellido) VALUES ('$nombre', '$email', '$password_encriptada', '$apellido')";

        // Ejecuta la consulta de inserción en la base de datos
        if (isset($consulta) && $consulta) {
            if ($conexion->query($consulta) === TRUE) {
                session_start();
                $_SESSION['nombre'] = $nombre;
                header('Location: login.html');
                exit;
            } else {
        }
    }
}
```

```

session_start();
include '../Conexion/conexion.php';

// Verificar si el formulario fue enviado
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = $conexion->real_escape_string($_POST['email']);
    $password = $conexion->real_escape_string($_POST['password']);

    // Consulta para buscar el usuario en la base de datos
    $consulta = "SELECT id, nombre, password FROM usuarios WHERE email = '$email'";
    $resultado = $conexion->query($consulta);

    if ($resultado->num_rows > 0) {
        // Usuario encontrado
        $fila = $resultado->fetch_assoc();
        if (password_verify($password, $fila['password'])) {
            // La contraseña coincide
            $_SESSION['usuario_id'] = $fila['id'];
            $_SESSION['nombre'] = $fila['nombre'];
            $_SESSION['password'] = $password; // Almacenar la contraseña introducida por el usuario en la sesión

            // Redireccionar a 003pkms.php
            header("Location: ../index.php");
            exit();
        }
        You, hace 3 días • 2 de php finalizado ...
    } else {
        // La contraseña no coincide
        echo "Contraseña incorrecta";
        echo "<button onclick=\"location.href='register.html'>Registrar</button>";
        echo "<button onclick=\"location.href='login.html'>Loguear de nuevo</button>";
    }
} else {
    // Usuario no encontrado
    echo "No se encontró una cuenta con ese email";
    echo "<button onclick=\"location.href='register.html'>Registrar</button>";
    echo "<button onclick=\"location.href='login.html'>Loguear de nuevo</button>";
}
$conexion->close();
?>

```

La sentencia de INSERT debe estar controlada para que no pueda introducirse ningún dato en blanco. Ten en cuenta que estás modificando la base de datos y no queremos campos mal llenados. Si todo ha ido bien, muestra un mensaje por pantalla diciendo El usuario XXX ha sido introducido en el sistema con la contraseña YYY.

El usuario Marcos ha sido introducido en el sistema con la contraseña Caracol.4

x

```

</div>
<div class="alert alert-info alert-dismissible fade show" role="alert">
    <?php
        // Verifica si existe la variable de sesión 'nombre'
        echo isset($_SESSION['nombre']) ? 'El usuario ' . $_SESSION['nombre'] : 'Invitado';
        echo isset($_SESSION['password']) ? ' ha sido introducido en el sistema con la contraseña ' . $_SESSION['password'] : ' ha sido intro
    ?>

```

Consigue que se registren/autentiquen y cierren sesión los usuarios.

Solo los usuarios autenticados pueden realizar operaciones de modificación (creación, actualización, borrado) en los datos.

SIN INICIAR SESIÓN, EL USUARIO, SOLAMENTE PUEDE REGISTRARSE O LOGUEARSE EN EL HEADER. ES LO ÚNICO FUNCIONAL

Invitado ha sido introducido en el sistema con la contraseña: No disponible. X

Pokemones

ID ▲▼	Nombre ▲▼	Tipo ▲▼	Subtipo ▲▼	Región ▲▼	Acciones
1	Bulbasaur	Grass	Poison	Kanto	
2	Charmander	Fire		Kanto	
9	Eevee	Normal		Kanto	
8	Geodude	Rock	Ground	Kanto	
5	Jigglypuff	Normal	Fairy	Kanto	
6	Meowth	Normal		Kanto	
4	Pikachu	Electric		Kanto	
7	Psyduck	Water		Kanto	
10	Snorlax	Normal		Kanto	
3	Squirtle	Water		Kanto	

```

<header>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="#">Pokémon</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbar" aria-controls="navbar" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbar">
            <ul class="navbar-nav">
                <?php if (isset($_SESSION['usuario_id'])): ?>
                    <!-- Usuario logueado -->
                    <li class="nav-item dropdown active">
                        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                            Perfil
                        </a>
                        <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
                            <a class="dropdown-item" href="login-register/login.html">Cerrar sesión</a>
                        </div>
                    </li>
                <?php else: ?>
                    <!-- Usuario no logueado -->
                    <li class="nav-item">
                        <a class="nav-link" href="login-register/login.html">Iniciar sesión</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="login-register/register.html">Registrar</a>
                    </li>
                <?php endif; ?>
            </ul>
        </div>
    </nav>
</header>

```

Mediante cookies utiliza de color principal en los detalles del diseño web el color que el usuario quiera.



```

<div class="container mt-5">
    <div class="colorizable p-5">
        <p>Elige tu color favorito:</p>
        <input type="color" id="colorPicker" class="form-control color-picker">
    </div>
</div>
You, hace 3 minutos • Uncommitted changes

```

```

window.onload = function() {
    var savedColor = getCookie("mainColor");
    if (savedColor) {
        changeColor(savedColor);
    }

    document.getElementById("colorPicker").addEventListener("input", function() {
        var selectedColor = this.value;
        changeColor(selectedColor);
        setCookie("mainColor", selectedColor, 30);
    });
};

function changeColor(color) {
    // Cambiar el color de fondo de los elementos con la clase 'colorizable'
    var elements = document.getElementsByClassName("colorizable");
    for (var i = 0; i < elements.length; i++) {
        elements[i].style.backgroundColor = color;
    }
}

function setCookie(name, value, days) {
    var expires = "";
    if (days) {
        var date = new Date();
        date.setTime(date.getTime() + (days*24*60*60*1000));
        expires = "; expires=" + date.toUTCString();
    }
    document.cookie = name + "=" + (value || "") + expires + "; path=/";
}

function getCookie(name) {
    var nameEQ = name + "=";
    var ca = document.cookie.split(';');
    for(var i=0;i < ca.length;i++) {
        var c = ca[i];
        while (c.charAt(0)==' ') c = c.substring(1,c.length);
        if (c.indexOf(nameEQ) == 0) return c.substring(nameEQ.length,c.length);
    }
    return null;
}

```

You, hace 1 segundo • Uncommitted changes

¿Serías capaz de trabajar con imágenes para cada uno de los pokémon?

Consigue crear a los campeones con un nuevo campo de imagen y poder editarlos también, en la visualización del pokémon debe aparecer la imagen.

Psyduck

Tipo: Water ▾

Subtipo: Ninguno ▾

Región: Kanto ▾

[Actualizar](#)



- **INSERT INTO** `imagenes_pokemon (pokemon_id, url_imagen) VALUES`
`(1, 'https://assets.pokemon.com/assets/cms2/img/pokedex/full/001.png'),`
`(2, 'https://assets.pokemon.com/assets/cms2/img/pokedex/full/004.png'),`
`(3, 'https://assets.pokemon.com/assets/cms2/img/pokedex/full/007.png'),`
`(4, 'https://assets.pokemon.com/assets/cms2/img/pokedex/full/025.png'),`
`(5, 'https://assets.pokemon.com/assets/cms2/img/pokedex/full/039.png'),`
`(6, 'https://assets.pokemon.com/assets/cms2/img/pokedex/full/052.png'),`
`(7, 'https://assets.pokemon.com/assets/cms2/img/pokedex/full/054.png'),`
`(8, 'https://assets.pokemon.com/assets/cms2/img/pokedex/full/074.png'),`
`(9, 'https://assets.pokemon.com/assets/cms2/img/pokedex/full/133.png'),`
`(10, 'https://assets.pokemon.com/assets/cms2/img/pokedex/full/143.png');`

```
<!-- Muestra la imagen del Pokémon -->
">
</div>

<?php
} else {
    echo "Pokémon no encontrado.";
}
```

- Adecuación de JavaScript con Bootstrap 5 (1 punto)



- Validación de todos los formularios presentes en la interfaz a nivel de cliente usando JavaScript (5 puntos)
 - Log in
 - Sign up
 - Crear Pokémon
 - Editar Pokémon
 - Eliminar Pokémon

localhost dice

El nombre solo debe contener letras.
El apellido solo debe contener letras.
La contraseña debe tener al menos 6 caracteres.
La contraseña debe contener al menos una letra mayúscula.
Las contraseñas no coinciden.

Aceptar

Registro

Nombre:

Apellido:

Email:

Contraseña:

Escribe de nuevo la contraseña:

Registrar

¿Ya tienes una cuenta? [Iniciar sesión](#)

```

You, hace 2 minutos | 1 author (You)
function validarFormulario() {
    let nombre = document.getElementById("nombre").value;
    let email = document.getElementById("email").value;
    let password = document.getElementById("password").value;
    let apellido = document.getElementById("apellido").value;
    let mensajeError = "";
    // Para verificar que solo contiene letras
    let soloLetras = /^[a-zA-Z]+$/;
    let emailForm = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6}$/;
    let confirma_password = document.getElementById("confirma-password").value;

    // Validación del nombre
    if (!soloLetras.test(nombre)) {
        mensajeError += "El nombre solo debe contener letras.\n";
    } else if (nombre.length < 3) {
        mensajeError += "El nombre debe tener al menos 3 caracteres.\n";
    } else if (nombre[0] !== nombre[0].toUpperCase()) {
        mensajeError += "El nombre debe comenzar con una letra mayúscula.\n";
    }

    // Validación del apellido
    if (!soloLetras.test(apellido)) {
        mensajeError += "El apellido solo debe contener letras.\n";
    } else if (apellido.length < 3) {
        mensajeError += "El apellido debe tener al menos 3 caracteres.\n";
    } else if (apellido[0] !== apellido[0].toUpperCase()) {
        mensajeError += "El apellido debe comenzar con una letra mayúscula.\n";
    }

    // Validación del formato de email
    if (!emailForm.test(email)) {
        mensajeError += "Por favor, ingresa un correo electrónico válido.\n";
    }

    // Validación de la contraseña
    if(password.length < 6) {
        mensajeError += "La contraseña debe tener al menos 6 caracteres.\n";
    }

    // Comprobar si la contraseña contiene al menos una letra mayúscula
    if(password && !/[A-Z]/.test(password)) {
        mensajeError += "La contraseña debe contener al menos una letra mayúscula.\n";
    }
}

0 Shared 0 You, hace 2 minutos Lín. 47, col. 45 Espacios: 4 UTF-8

```

- Creación de un footer con el año en el que estamos presentes " ©Copyright Proyecto Pokémon - 2024" , por ejemplo, donde el año se actualice al actual de forma automática. (1 punto)

```


!\[\]\(ed3644cb3771836dd964b84e042f63cf\_img.jpg\)

©Copyright Proyecto Pokémon <span id="year"></span>


```

```
+ // Año footer  
document.getElementById("year").textContent = new Date().getFullYear();
```

©Copyright Proyecto Pokémon 2024

- Crear una función JavaScript que cambie algunos aspectos visuales de la web dependiendo de la estación en la que estemos, por ejemplo: botones rosas si es primavera, naranjas si es otoño, verdes si es verano y azules si es invierno, pero pueden ser textos, imágenes etc. Cuánto más profesional quede mejor. (3 puntos).

Añadir Pokémon

```
function obtenerEstacion() {
    const mesActual = new Date().getMonth() + 1; // Enero = 1, Febrero = 2, etc.
    if (mesActual >= 3 && mesActual <= 5) {
        return 'primavera';
    } else if (mesActual >= 6 && mesActual <= 8) {
        return 'verano';
    } else if (mesActual >= 9 && mesActual <= 11) {
        return 'otoño';
    } else {
        return 'invierno';
    }
}

function aplicarEstilosPorEstacion() {
    const estacion = obtenerEstacion();
    let color;

    switch(estacion) {
        case 'primavera':
            color = 'pink';
            break;
        case 'verano':
            color = 'green';
            break;
        case 'otoño':
            color = 'orange';
            break;
        case 'invierno':
            color = 'blue';
            break;
    }

    // Cambiar el color del botón específico
    const botonCrearPokemon = document.querySelector('.btn-crear-pokemon');
    if (botonCrearPokemon) {
        botonCrearPokemon.style.backgroundColor = color;
    }
}

// Aplicar los estilos al cargar la página
window.onload = aplicarEstilosPorEstacion;
```