# Ejercicio 1 ( 3,5 ptos)

En una prueba contrarreloj, los 25 ciclistas participantes deben salir con una cadencia de 5 segundos en función del dorsal que llevan (el dorsal 1 saldrá 5 segundos después de la hora de salida, el dorsal 2, 10 segundos después de la hora de salida… y así sucesivamente .

En un pequeño monitor de su moderna bicicleta, el participante debe ver un mensaje con una cuenta atrás que empieza 3 segundos antes de salir y la palabra ¡Salida! Después de la cuenta atrás para poder salir \*.

Crea la aplicación ***salidaEtapa*** que debe instalar cada ciclista.

Dorsal 23 3,2,1… ¡salida!

\*Simulación de la pantalla

Secuencia: Dorsal 23 3 --- Dorsal 23 2 --- Dorsal 23 1 --- Dorsal 23 ¡salida!

Valoración sobre 10:

* Funcionamiento correcto y según especificaciones

de la aplicación con uso de “template literals” 4 puntos

* Uso de promesas 2 puntos
* Utilización de async/await 2 puntos
* Cuenta atrás 2 puntos

# Ejercicio 2 ( 3,5 puntos)

Define una clase ***Animal*** con las propiedades ***nombre, edad, patas, especie*** y ***estado***.

Y el método ***introduce()*** que devuelva “Hola, mi nombre es [nombre] y tengo [edad] años.”

Define ahora las siguientes clases que heredan de ***Animal***.

1. ***Tiburon***. La función constructora para ***Tiburon*** debe aceptar 3 argumentos en total en el siguiente orden: nombre, edad, estado. Todos los tiburones pertenecen a la especie de ***Escualos***.
2. ***Gato***. La función constructora para ***Gato*** debe aceptar los mismos 3 argumentos que ***Tiburon***: nombre, edad, estado. Los gatos pertenecen a la especie ***Felinos***. Además, su método ***introduce()*** debe ser idéntico al original, excepto que debe haber exactamente 2 espacios en blanco antes de añadir las palabras "¡Miau, miau!".
3. ***Perro***. El constructor de la clase ***Perro*** debe aceptar 4 argumentos en el orden especificado: ***nombre, edad, estado, dueño***, donde ***dueño*** ( que es un string) es el nombre del propietario del perro. Además, los perros, que pertenecen a la especie ***Canidos***, tienen un método ***introduce()*** como cualquier otro animal, pero tienen su propio método llamado ***saludaDueño()*** que no acepta argumentos y devuelve "Hola [nombre del dueño]”
4. **Ejemplos de uso** .- Crea:

* Un tiburón llamado Chum de 5 años y estado hambriento
* Un gato llamado Silvestre de 3 años y estado juguetón
* El perro de Pepe, Cane de 2 años y estado feliz

Y muestra por consola sus métodos correspondientes

Valoración sobre 10:

* Funcionamiento correcto y según especificaciones

de la aplicación con uso de “template literals” 4 puntos

* Clases en JavaScript 6 3 puntos
* Utilización de herencia, encapsulamiento… 3 puntos

# Ejercicio 3 ( 3 puntos)

Con el código proporcionado, desarrolla un formulario de registro de usuarios. Debes implementar la validación del formulario para garantizar que los usuarios introduzcan la información de manera adecuada.

Requisitos del formulario:

1.Nombre:

* Debe contener solo letras (sin espacios).
* No puede estar en blanco.

2.Correo electrónico:

* Debe tener un formato de dirección de correo electrónico válido.

3.Contraseña:

* Debe tener al menos 8 caracteres de longitud.
* Debe incluir al menos una letra mayúscula, una letra minúscula y un número.

4.Confirmación de contraseña:

* Debe coincidir con la contraseña introducida en el campo anterior.

5.Número de teléfono:

* Debe contener solo números.
* Nueve dígitos

Tareas a realizar:

1. Implementa la validación en JavaScript:

* Valida **cada campo** del formulario según los requisitos mencionados anteriormente.
* Muestra mensajes de error **específicos** para cada campo si la validación falla.
* Deshabilita el envío del formulario si hay errores de validación.

2. Mejora la experiencia del usuario:

* Agrega un mensaje de éxito cuando el formulario se envíe correctamente.
* Realiza la validación en tiempo real mientras el usuario completa el formulario.
* Lleva el foco al elemento que provoca el error

Los mensajes deben aparecer en la zona final del formulario justo debajo del botón de registrarse.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Si no encuentras la manera de validar el correo y la contraseña, puedes utilizar el siguiente código:

**Validación de correo**

if (!/^\S+@\S+\.\S+$/.test(email)) {

displayError('Correo electrónico inválido');

}

**Validación de contraseña**

if (password.length < 8 || !/[a-z]/.test(password) || !/[A-Z]/.test(password) || !/\d/.test(password)) {

displayError('Contraseña inválida');

}

Valoración sobre 10:

* Funcionamiento correcto de la aplicación 4 puntos
* Creación e inclusión de elementos en el DOM 2 puntos
* Eventos, funciones, error, foco, preventDefault 2 puntos
* Validación HTML5, JavaScript y con API 2 puntos

# Instrucciones

El texto no tiene acentos en los nombres de campos, métodos …

El código JavaScript y el CSS deben ir embebidos en el HTML.

Añade los comentarios que creas oportuno para explicar tu solución

Añade tu nombre en el código (Author)

**Sube al aula únicamente tres ficheros HTML con formato de nombre** **siguiente:**

EV2A-1XXX.html, EV2A-2XXX.html y EV2A-3.html donde XXX son tus tres iniciales habituales

# Anexo: código

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <style>

    body {

        font-family: Arial, sans-serif;

        background-color: #f4f4f4;

        margin: 0;

        display: flex;

        justify-content: center;

        align-items: center;

        height: 100vh;

    }

    .container {

        background-color: #fff;

        padding: 20px;

        border-radius: 8px;

        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

        width: 300px;

    }

    form {

        display: flex;

        flex-direction: column;

    }

    label {

        margin-bottom: 5px;

    }

    input, button {

        margin-bottom: 10px;

        padding: 8px;

    }

    button {

        background-color: #620af0;

        color: #fff;

        border: none;

        cursor: pointer;

    }

    button:hover {

        background-color: #5445a0;

    }

    .error {

        color: #ff0000;

        font-size: 14px;

    }

    </style>

    <title>Formulario de Registro Examen </title>

</head>

<body>

    <div class="container">

        <h2>Formulario de Registro</h2>

        <form id="formRegistro">

            <label for="nombre">Nombre:</label>

            <input type="text" id="nombre" name="nombre" required>

            <label for="email">Correo electrónico:</label>

            <input type="email" id="email" name="email" required>

            //Resto de campos del formulario

</form>

        <div id="messages"></div>

    </div>

    <script>

    Aquí tu código

    </script>

</body>

</html>