

# Ejercicio Práctico N° 2: Front

**Fecha de entrega: 02/10/2025**

**Clase Presencial y Obligatoria**

**Individual o en Grupos de hasta 3 personas**

El Club Social y Deportivo Unidos por el Deporte, necesita realizar una página web como landing de un evento de ciclismo MTB rural del cual será sede.

El dirigente encargado de la organización del evento nos brindó las pautas y requisitos que necesita este sitio web.

Requerimientos:

- Utilizar un template gratuito (el club no quiere incurrir en gastos de diseño)
- Utilizar el framework Flask
- Crear únicamente las siguientes páginas dentro del sitio web.
  - index.html -> Toda la información del evento (ver información adicional)
  - registration.html-> Formulario de inscripción al evento y envío de mail
- Un script en bash que al ejecutarlo crea un proyecto en flask llamado “EjPractico2” con el ambiente virtual, y un archivo (vacío) app.py. El proyecto deberá tener las carpetas abajo mencionadas:

```
.
├── /TP2-IDS/
│   ├── /.env
│   ├── /static/
│   │   ├── /css
│   │   └── /images
│   └── /templates
└── app.py (vacío)
```

- Crear un archivo `base.html`, que reutilice todo el código posible (que se encuentra repetido en las distintas páginas html) con el uso de bloques de Flask.
- Reemplazar todas las urls por `url_for` según corresponda.
- Reparar la redirección de los archivos que sean invocados dentro del código HTML

Ejemplo:

```
<link rel="stylesheet" href="css/style.css">
```

deberá ser reemplazado por

```
<link rel="stylesheet" href="{{url_for('static',
filename='/css/style.css', _external=True)}}">
```

- Colocar imágenes, logos, etc, [obtenidas de la web en forma gratuita](#), reemplazando las existentes en el template.

Las mismas deberán ir en la carpeta **static/images/**

- Crear una variable tipo *diccionario* que contenga toda la información del evento

Ejemplo:

```
info_evento = {

    1: {    "nombre": "Rally MTB 2025",

            "organizador": "Club Social y Deportivo Unidos por el Deporte",

            "descripcion": "Carrera de MTB rural en dos modalidades 30km y 80km ...",

            "fecha": "24 de Octubre de 2025",

            "horario": "8am",

            "lugar": "Tandil, Buenos Aires",

            "tipo_carrera": "MTB rural",

            "modalidad_costo": {1: {"nombre": "Corta" ,"valor": "100"},

                                2: {"nombre": "Larga" ,"valor": "200"}},

            "Auspiciantes": ["ausp1","auspN"]

    }
```

```
}
```

Este diccionario deberá estar hardcodeado en el código principal del [app.py](#) con la información necesaria.

- Modificar el archivo `style.css` de forma que se adecúen al color del club (ver información adicional)
- Crear un form que contenga datos del competidor y la carrera que al clickear en el botón de envío de información, se envíe un correo electrónico con el cuerpo del formulario (el contacto del usuario y datos de la carrera para su inscripción).

Para implementar el envío de correos electrónicos, pueden utilizar el siguiente código como ejemplo:

```
from flask import Flask, render_template, request, redirect, flash
```

```
from flask_mail import Mail, Message
```

```
app = Flask(__name__)
```

```
app.config['MAIL_SERVER'] = 'smtp.gmail.com'
```

```
app.config['MAIL_PORT'] = 587
```

```
app.config['MAIL_USE_TLS'] = True
```

```
app.config['MAIL_USE_SSL'] = False
```

```
app.config['MAIL_USERNAME'] = 'your-email@gmail.com'
```

```
app.config['MAIL_PASSWORD'] = 'your-email-password' security
```

```
app.config['MAIL_DEFAULT_SENDER'] = 'your-email@gmail.com'
```

```
mail = Mail(app)
```

## **Información adicional:**

El evento se va a desarrollar en la ciudad de Tandil en Buenos Aires, organizado por nuestro club, el día 24 de Octubre de 2025 a las 8am. Existen dos modalidades corta y larga, la primera es de 30 km y la segunda de 80km. Los costos y auspiciantes se informarán más adelante, en principio el desarrollador puede poner auspiciantes e importes que desee.

### **¿Qué debe tener el index.html?**

Navbar con logo, distintos accesos, info importante etc.

- Footer con info relevante, contacto whatsapp, mail, links etc.
- Fotos relacionadas al evento (se pueden obtener gratuitas de la web)
- Mapa del recorrido generado con googlemaps (incluir el mismo)
- Punto de salida/llegada de la carrera
- Puntos de hidratación
- Listado con imágenes de logos de los auspiciantes
- Deslinde pdf para descargar
- Datos del kit de carrera
- Link al reglamento (link a un pdf)
- Link a inscripción

### **¿Qué debe tener el registration.html?**

- Navbar con logo, distintos accesos, info importante etc.
- Footer con info relevante, contacto whatsapp, mail, links etc.
- Fotos relacionadas al evento (se pueden obtener gratuitas de la web)
- Formulario con datos solicitados al corredor, y poder seleccionar la modalidad corta/larga de la carrera
- Botón enviar que termine enviando un mail a la organización con los datos del corredor.

## **Uso de un repositorio en Github**

- El desarrollo debe ser entregado al cliente en un repositorio de Github.
- Es deseable que se creen al menos dos ramas (develop, deploy...)