

# Guía de ejercicios de front-end (integración con Flask)

## Introducción

Esta guía presenta ejercicios prácticos para aprender HTML, CSS y cómo integrar dichos lenguajes con el framework Flask. También vemos el uso de Jinja.

Indicadores de dificultad:

- (Sin emoji) NORMAL
- 🦊 AVANZADO
- ☢️ PRO

Para los ejercicios 5a, 5b, 5c, 5d, 5e y 5f, revisar estos recursos:

- <https://playcode.io/html5>
- <https://fonts.google.com>

## Ejercicios

### Ejercicio 1:

Desarrollar una aplicación web de Flask que se levante en el puerto 5001 , con el modo "debug" en True .

### Ejercicio 2:

Desarrollar una aplicación web de Flask que se levante en el puerto 5002 y que al llamar a la ruta "/" desde el navegador web, devuelva un texto HTML que contenga la frase "Hola Flask!" dentro de una etiqueta h1 .

*Pista: `http://localhost:5002/` debe mostrar en el navegador web "HoLa FLask!"*

## Ejercicio 3:

A la aplicación del ejercicio 2 reemplazar el texto "Hola Flask!" por un template en el archivo `index.html` con el mismo texto.

*(Tarea: revisar como se crea un template en Flask)*

## Ejercicio 3a:

Agregar un enlace al sitio <https://www.google.com.ar/> el el texto "Hola Flask!" , de manera tal que al hacer click en el texto, el navegador Web redirija al buscador de Google

*(Pista: hay un elemento de HTML que nos permite hacer esto)*

## Ejercicio 4 🐱:

A la aplicación del ejercicio 2 agregar una función `datos_contacto()` que esté enlazada a la ruta `"/contacto"` . Dicha ruta, al ser invocada desde el navegador web debe devolver un "template" llamado `contacto.html` que contenga en su interior el siguiente texto:

Hola, soy (poné tu nombre acá) y mi padrón es (poné tu padrón acá) .

## Ejercicio 5 🐱:

A la aplicación del ejercicio 2 agregar un archivo CSS llamado `styles.css` *(tarea: revisar donde van los archivos CSS en Flask)* que contenga un selector de etiqueta `h1` que cambie el color del texto a rojo, y enlazarlo al archivo `index.html` .

## Ejercicio 5a:

Agregar un estilo de CSS que al hacer click en el texto, el texto cambie su color a verde.

## Ejercicio 5b:

Agregar un estilo de CSS que cambie el color del fondo de la página a `#36274a` (violeta azulado oscuro).

## Ejercicio 5c:

Agregar un estilo de CSS que subraye el texto con línea punteada.

## Ejercicio 5d 🦉:

Agregar un estilo de CSS que cambie la "font" (fuente) del texto a "Noto Sans" (*Pista: buscar como insertar fuentes externas en HTML*).

## Ejercicio 5e:

Agregar un estilo de CSS que al pasar el mouse por encima del texto, el texto se amplifique (*Pista: propiedad transform en CSS*).

## Ejercicio 5f:

Al estilo anterior agregar la propiedad de que la animación dure 1.5 segundos (*Pista: propiedad transition en CSS*).

## Ejercicio 6 🦉:

Desarrollar una segunda aplicación web de Flask que se levante en el puerto 5003 que contenga en el archivo principal ( index.html ) un elemento de tipo button que al hacerle click redirija el navegador a http://localhost:5002/contacto (la aplicación del ejercicio 4).

*Pista: hay que levantar las 2 aplicaciones al mismo tiempo.*

## Ejercicio 7 🦉:

Desarrollar una aplicación web en flask que se levante en el puerto 4856 y que en la ruta "/maps" muestre un mapa de Google Maps.

Recursos: <https://www.maps.ie/create-google-map/>

## Ejercicio 8:

A la aplicación del ejercicio 7 agregar la ruta "/" que redirija el navegador web automáticamente a la ruta "/maps" .

*Pista: buscar cómo se utilizan las funciones `redirect()` y `url_for()` en Flask.*

## Ejercicio 9 🚫 :

Crear una aplicación Web de Flask en el puerto 6772 que contenga 3 rutas:

- "/"
- "/productos"
- "/contacto"

La ruta "/" debe redirigir a la ruta "/productos". Desde dicha ruta se deben mostrar en una lista desordenada (elemento `<ul>`) los siguientes productos:

```
products: list[str] = ["PS5", "Switch 2", "PS5 Pro", "XBOX"]
```

Los productos deben mostrarse en color `rgb(144, 78, 199)` (lila) y con el texto en negrita.

Luego, la ruta "/contacto" debe contener un formulario (`<form>`) para sugerir nuevos productos, con los siguientes campos:

- Nombre (*campo de tipo texto*)
- Email (*campo de tipo email*)
- Comentarios (*campo de texto libre con un máximo de 256 caracteres*)
- Botón de "enviar" (`submit`)

De momento, el botón "Enviar" no hace nada. Eso lo veremos en las clases de *back-end*.

(Pista: [https://www.w3schools.com/html/html\\_forms.asp](https://www.w3schools.com/html/html_forms.asp))

## Ejercicio 10 🐱 :

A la aplicación del ejercicio anterior, reemplazar la lista de productos por la siguiente:

```
# cada línea es un producto con las propiedades "name" y "stock"
products = [
    {"name": "PS5", "stock": True},
    {"name": "Switch 2", "stock": False},
    {"name": "PS5 Pro", "stock": False},
    {"name": "XBOX", "stock": True},
]
```

Y mostrar en color gray (gris) aquellos productos que *no tengan stock*.