

## Clase 06

---

# Rumbo al Proyecto final Full Stack Python con objetos y API REST

---

### Sexta etapa:

#### Implementar un frontend

En esta etapa veremos una serie de códigos mínimos que nos ayuden a desarrollar un frontend que utilice las funciones de la API desarrollada.

Haremos incapié en los temas relacionados con la API en sí, y no tanto en el apartado CSS o HTML.

Usaremos Vue3 para simplificar el código.

---

### Menú principal

---

El archivo de entrada al frontend es simplemente un menú con enlaces que llevan a cada una de las secciones implementadas:

#### index.html:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo de uso de la API</title>
  <link rel="stylesheet" href="estilos.css">
</head>

<body>
  <div>
    <h1>Ejemplos de uso de la API</h1>
    <h3>Codo a Codo 2023</h3>
    <table>
      <tr>
        <td class="contenedor-centrado"><a href="altas.html">Alta de productos</a></td>
      </tr>

      <tr>
        <td class="contenedor-centrado "><a href="listado.html">Listado de productos</a></td>
      </tr>

      <tr>
        <td class="contenedor-centrado "><a href="modificaciones.html">Modificar datos de productos</a></td>
      </tr>

      <tr>
        <td class="contenedor-centrado "><a href="listadoEliminar.html">Eliminar productos</a></td>
      </tr>

      <tr>
        <td class="contenedor-centrado "><a href="carrito1.html">Carrito de compras</a></td>
      </tr>
    </table>
  </div>
</body>

</html>
```

---

### Alta de productos

altas.html:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Agregar producto</title>
  <link rel="stylesheet" href="estilos.css">
</head>

<body>
  <h1>Agregar Productos al Inventario</h1>
  <h3>Codo a Codo 2023</h3>
  <form id="formulario">
    <label for="codigo">Código:</label>
    <input type="text" id="codigo" name="codigo" required><br>

    <label for="descripcion">Descripción:</label>
    <input type="text" id="descripcion" name="descripcion" required><br>

    <label for="cantidad">Cantidad:</label>
    <input type="number" id="cantidad" name="cantidad" required><br>

    <label for="precio">Precio:</label>
    <input type="number" step="0.01" id="precio" name="precio" required><br>

    <button type="submit">Agregar Producto</button>
    <a href="index.html">Menu principal</a>
  </form>

  <script>
    //const URL = "http://127.0.0.1:5000/"
    const URL = "https://xxxxx.pythonanywhere.com/"
    // Capturamos el evento de envío del formulario
    document.getElementById('formulario').addEventListener('submit', function (event) {
      event.preventDefault(); // Evitamos que se recargue la página

      // Obtenemos los valores del formulario
      var codigo = document.getElementById('codigo').value;
      var descripcion = document.getElementById('descripcion').value;
      var cantidad = document.getElementById('cantidad').value;
      var precio = document.getElementById('precio').value;

      // Creamos un objeto con los datos del producto
      var producto = {
        codigo: codigo,
        descripcion: descripcion,
        cantidad: cantidad,
        precio: precio
      };
      console.log(producto)
      // Realizamos la solicitud POST al servidor
      fetch(URL + 'productos', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json'
        },
        body: JSON.stringify(producto)
      })
        .then(function (response) {
          // Código para manejar la respuesta
          if (response.ok) {
            return response.json(); // Parseamos la respuesta JSON
          } else {
            // Si hubo un error, lanzar explícitamente una excepción
            // para ser "catcheada" más adelante
            throw new Error('Error al agregar el producto.');
```

```
</body>
</html>
```

Descripción del script utilizado:

Se comienza declarando una constante:

```
const URL = "https://xxxxx.pythonanywhere.com/";
```

Esta constante almacena la URL del servidor al que se enviarán las solicitudes POST. Puede ser una dirección local (como "http://127.0.0.1:5000/") o una dirección remota (como en este caso). Se debe cambiar la URL dependiendo de las necesidades del proyecto. A continuación, se utiliza el método **addEventListener** para agregar un evento de escucha al formulario con el identificador "formulario" cuando se envíe:

```
document.getElementById('formulario').addEventListener('submit', function (event) {
    event.preventDefault(); // Evitamos que se envíe el form por ahora
    // Código del evento
});
```

Esto permite capturar el evento de envío del formulario y ejecutar el código correspondiente cuando se envía el formulario. **event.preventDefault()**; se utiliza para evitar que la página se recargue cuando se envía el formulario. Esto se logra llamando al método **preventDefault()** en el objeto event pasado como parámetro en la función del evento. A continuación, se obtienen los valores de los campos de entrada de texto y se almacenan en variables:

```
// Obtenemos los valores del formulario
var codigo = document.getElementById('codigo').value;
var descripcion = document.getElementById('descripcion').value;
var cantidad = document.getElementById('cantidad').value;
var precio = document.getElementById('precio').value;
```

Esto se logra utilizando el método **getElementById()** para obtener los elementos del DOM correspondientes a los campos de entrada y accediendo a su propiedad value para obtener el valor ingresado por el usuario. Luego, se crea un objeto producto con los datos obtenidos del formulario:

```
// Creamos un objeto con los datos del producto
var producto = {
    codigo: codigo,
    descripcion: descripcion,
    cantidad: cantidad,
    precio: precio
};
```

Este objeto producto contiene las propiedades codigo, descripcion, cantidad y precio, con sus respectivos valores obtenidos del formulario. Se realiza una solicitud POST al servidor utilizando el método fetch():

```
fetch(URL + 'productos', {
    method: 'POST',
    headers: {
        'Content-Type': 'application/json'
    },
    body: JSON.stringify(producto)
})
.then(function (response) {
    // Código para manejar la respuesta
})
.catch(function (error) {
    // Código para manejar errores
});
```

El URL utilizado en la solicitud POST se construye concatenando la constante URL con la ruta específica, en este caso, 'productos'. Se especifica que el método de la solicitud es POST, se establece el encabezado 'Content-Type': 'application/json' para indicar que se está enviando datos en formato JSON, y se convierte el objeto producto a JSON utilizando JSON.stringify() antes de enviarlo como cuerpo de la solicitud.

Después de realizar la solicitud POST, se utiliza el método then() para manejar la respuesta del servidor:

```
.then(function (response) {
    if (response.ok) {
        return response.json(); // Parseamos la respuesta JSON
    } else {
```

```

        throw new Error('Error al agregar el producto.');
```

En este caso, se verifica si la respuesta del servidor es exitosa (response.ok). Si es así, se llama al método json() en la respuesta para obtener los datos de la respuesta en formato JSON. Si la respuesta no es exitosa, se lanza un error. Finalmente, se utiliza otro then() para manejar los datos de la respuesta:

```

.then(function (data) {
    alert('Producto agregado correctamente.');
```

En este caso, se muestra una alerta indicando que el producto se agregó correctamente y se restablecen los valores de los campos del formulario a través de la manipulación del DOM. En caso de producirse algún error durante el proceso, se captura en el bloque catch() y se muestra una alerta de error. Este script maneja el evento de envío del formulario, obtiene los valores ingresados por el usuario, realiza una solicitud POST al servidor con los datos del producto y maneja la respuesta del servidor.

## Listado de productos en el inventario (sin Vue3)

El siguiente documento HTML utiliza la API implementada para generar una tabla con el contenido de la tabla que contiene el inventario completo.

Se utilizan técnicas de manipulación del DOM "puras", sin Vue, para que el alumno pueda ver en acción lo aprendido antes:

**listado.html:**

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Listado de Productos</title>
    <link rel="stylesheet" href="estilos.css">
</head>

<body>
    <h1>Listado de Productos</h1>
    <h3>Codo a Codo 2023</h3>
    <table>
        <thead>
            <tr>
                <th>Código</th>
                <th>Descripción</th>
                <th>Cantidad</th>
                <th align="right">Precio</th>
            </tr>
        </thead>
        <tbody id="tablaProductos">
        </tbody>
    </table>

    <div class="contenedor-centrado">
        <a href="index.html">Menu principal</a>
    </div>

    <script>
        //const URL = "http://127.0.0.1:5000/"
        const URL = "https://xxxxx.pythonanywhere.com/"

        // Realizamos la solicitud GET al servidor para obtener todos los productos
        fetch(URL + 'productos')
```

```

        .then(function (response) {
            if (response.ok) {
                return response.json(); // Parseamos la respuesta JSON
            } else {
                // Si hubo un error, lanzar explícitamente una excepción
                // para ser "catcheada" más adelante
                throw new Error('Error al obtener los productos.');
```

## Eliminar productos del inventario (con Vue3)

El código siguiente muestra como generar un listado similar al anterior, pero empleando Vue3. Y se agrega un botón "Eliminar" en cada fila de la tabla, que utiliza la API desarrollada para quitar del inventario el producto seleccionado.

**listadoEliminat.html:**

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Listado de Productos</title>
    <link rel="stylesheet" href="estilos.css">
</head>

<body>
    <h1>Baja de Productos</h1>
    <h3>Codo a Codo 2023</h3>
    <table>
        <thead>
            <tr>
                <th>Código</th>
                <th>Descripción</th>
                <th>Cantidad</th>
                <th align="right">Precio</th>
                <th>Acciones</th>
            </tr>
        </thead>
        <tbody>
            <tr v-for="producto in productos">
                <td>{{ producto.codigo }}</td>
                <td>{{ producto.descripcion }}</td>
                <td>{{ producto.cantidad }}</td>
                <td align="right">{{ producto.precio }}</td>
                <td><button @click="eliminarProducto(producto.codigo)">Eliminar</button></td>
            </tr>
        </tbody>
    </table>

    <div class="contenedor-centrado">
        <a href="index.html">Menu principal</a>
    </div>

    <script src="https://unpkg.com/vue@next"></script>
    <script>
        //const URL = "http://127.0.0.1:5000/"
```

```

const URL = "https://arielcodo.pythonanywhere.com/";

const app = Vue.createApp({
  data() {
    return {
      productos: []
    }
  },
  methods: {
    obtenerProductos() {
      // Obtenemos el contenido del inventario
      fetch(URL + 'productos')
        .then(response => {
          if (response.ok) {
            return response.json(); // Parseamos la respuesta JSON
          } else {
            // Si hubo un error, lanzar explícitamente una excepción
            // para ser "catcheada" más adelante
            throw new Error('Error al obtener los productos.');
```

## Modificar productos en el inventario (sin Vue3)

El ejemplo siguiente nos muestra una manera de utilizar la API para modificar los datos de un producto del inventario. No se utiliza Vue3.

**modificaciones.html:**

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Modificar Producto</title>
  <link rel="stylesheet" href="estilos.css">
</head>

```

```

<body>
  <h1>Modificar Productos del Inventario</h1>
  <h3>Codo a Codo 2023</h3>
  <form id="formulario">
    <label for="codigo">Código:</label>
    <input type="text" id="codigo" name="codigo" required><br>
    <button type="submit">Modificar Producto</button> <a href="index.html">Menu principal</a>
  </form>

  <div id="datosProducto" style="display: none;">
    <h2>Datos del Producto</h2>
    <form id="formularioModificar">
      <label for="descripcionModificar">Descripción:</label>
      <input type="text" id="descripcionModificar" name="descripcionModificar" required><br>

      <label for="cantidadModificar">Cantidad:</label>
      <input type="number" id="cantidadModificar" name="cantidadModificar" required><br>

      <label for="precioModificar">Precio:</label>
      <input type="number" step="0.01" id="precioModificar" name="precioModificar" required><br>

      <button type="submit">Guardar Cambios</button>
      <a href="modificaciones.html">Cancelar</a>
    </form>
  </div>

  <script>
    //const URL = "http://127.0.0.1:5000/"
    const URL = "https://xxxxx.pythonanywhere.com/";

    // Capturamos el evento de envío del formulario para mostrar los datos del producto
    document.getElementById('formulario').addEventListener('submit', function (event) {
      event.preventDefault(); // Evitamos que se recargue la página

      // Obtenemos el código del producto
      var codigo = document.getElementById('codigo').value;

      // Realizamos la solicitud GET al servidor para obtener los datos del producto
      fetch(URL + 'productos/' + codigo)
        .then(function (response) {
          if (response.ok) {
            return response.json(); // Parseamos la respuesta JSON
          } else {
            // Si hubo un error, lanzar explícitamente una excepción
            // para ser "catcheada" más adelante
            throw new Error('Error al obtener los datos del producto.');
```

```

        .then(function (response) {
            if (response.ok) {
                return response.json(); // Parseamos la respuesta JSON
            } else {
                // Si hubo un error, lanzar explícitamente una excepción
                // para ser "catcheada" más adelante
                throw new Error('Error al guardar los cambios del producto.');
```

```

            }
        })
        .then(function (data) {
            alert('Cambios guardados correctamente.');
```

```

            location.reload(); // Recargamos la página para volver al formulario de consulta
        })
        .catch(function (error) {
            // Código para manejar errores
            alert('Error al guardar los cambios del producto.');
```

```

        })
    });
</script>
</body>
</html>

```

## Hoja de estilos:

Todos los archivos HTML mostrados utilizan el mismo código CSS:

**estilos.css:**

```

/* Estilos para todo el proyecto */
body {
    font-family: Arial, sans-serif;
    background-color: #f9f9f9;
}

.contenedor-centrado {
    display: flex;
    width: 100%;
    justify-content: center;
}

p {
    font-family: 'Times New Roman', Times, serif;
    background-color: #f9f9f9;
    max-width: 400px;
    margin: 0 auto;
    padding: 20px;
}

h1, h2, h3, h4, h5, h6 {
    text-align: center;
    color: #007bff;
}

form {
    max-width: 400px;
    margin: 0 auto;
    padding: 20px;
    background-color: white;
    border: 1px solid lightslategray;
    border-radius: 5px;
}

table {
    max-width: 90%;
    margin: 0 auto;
    padding: 20px;
    background-color: white;
    border: 1px solid lightslategray;
    border-radius: 5px;
}

label {
    display: block;

```



```

    margin-bottom: 5px;
}

input[type="text"],
input[type="number"],
textarea {
    width: 90%;
    padding: 10px;
    margin-bottom: 10px;
    border: 1px solid #cccccc;
    border-radius: 5px;
}

input[type="submit"] {
    padding: 10px;
    background-color: #007bff;
    color: #ffffff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}

input[type="submit"]:hover {
    background-color: #0056b3;
}

button {
    padding: 8px;
    margin: 4px;
    background-color: #007bff;
    color: #ffffff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}

button:hover {
    background-color: #0056b3;
}

a {
    padding: 8px;
    margin: 4px;
    background-color: #fdab13;
    color: #ffffff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    text-decoration: none;
    font-size: 85%;
}

a:hover {
    background-color: #cd7b00;
}

```

#### Breve descripción de cada estilo implementado:

- **body:** Aplica estilos al elemento <body> de la página. Establece la fuente del texto como Arial o una fuente genérica sin serifa (sans-serif) y establece el color de fondo como #f9f9f9.
- **.contenedor-centrado:** Aplica estilos a un contenedor específico utilizando la clase .contenedor-centrado. Hace uso de flexbox para centrar horizontalmente su contenido y establece su ancho como el 100% del contenedor padre.
- **p:** Aplica estilos a los elementos <p> (párrafos). Establece la fuente del texto como 'Times New Roman', Times, serif, establece el color de fondo como #f9f9f9, limita el ancho máximo a 400px y centra el elemento horizontalmente utilizando los márgenes automáticos. Además, se añade un relleno interno de 20px.
- **h1, h2, h3, h4, h5, h6:** Aplica estilos a los elementos de encabezado del nivel 1 al 6 (<h1> a <h6>). Los alinea al centro y establece el color del texto como #007bff (un tono de azul).
- **form:** Aplica estilos a los elementos <form> (formularios). Establece el ancho máximo a 400px, centra el formulario horizontalmente utilizando los márgenes automáticos y añade un relleno interno de 20px. Además, establece el color de fondo como blanco, agrega un borde de 1px sólido en lightslategray y un borde redondeado de 5px.
- **table:** Aplica estilos a los elementos <table> (tablas). Establece el ancho máximo al 90%, centra la tabla horizontalmente utilizando los márgenes automáticos y añade un relleno interno de 20px. Además, establece el color de fondo como blanco, agrega un borde de 1px sólido en lightslategray y un borde redondeado de 5px.
- **label:** Aplica estilos a los elementos <label> (etiquetas). Los muestra como bloques y agrega un margen inferior de 5px.
- **input[type="text"], input[type="number"], textarea:** Aplica estilos a los elementos de entrada de tipo texto, número y área de texto (<input type="text">, <input type="number"> y <textarea>). Establece el ancho al 90%, agrega un relleno de 10px, un margen inferior de 10px, un borde de 1px sólido en #cccccc y un borde redondeado de 5px.

- **input[type="submit"]:** Aplica estilos al botón de envío (<input type="submit">). Agrega un relleno de 10px, establece el color de fondo como #007bff, el color del texto como blanco, elimina el borde y añade un borde redondeado de 5px. Además, cambia el cursor del mouse a un puntero cuando se pasa sobre el botón.
- **input[type="submit"]:hover:** Aplica estilos al botón de envío cuando se pasa el cursor sobre él. Cambia el color de fondo a #0056b3.
- **button:** Aplica estilos a los elementos <button> (botones). Agrega un relleno de 8px, un margen de 4px, establece el color de fondo como #007bff, el color del texto como blanco, elimina el borde y añade un borde redondeado de 5px. Además, cambia el cursor del mouse a un puntero cuando se pasa sobre el botón.
- **button:hover:** Aplica estilos a los botones cuando se pasa el cursor sobre ellos. Cambia el color de fondo a #0056b3.
- **a:** Aplica estilos a los elementos <a> (enlaces). Agrega un relleno de 8px, un margen de 4px, establece el color de fondo como #fdab13, el color del texto como blanco, elimina el borde y añade un borde redondeado de 5px. Además, cambia el cursor del mouse a un puntero cuando se pasa sobre el enlace y establece el tamaño de fuente al 85%.
- **a:hover:** Aplica estilos a los enlaces cuando se pasa el cursor sobre ellos. Cambia el color de fondo a #cd7b00.