



UNIVERSIDAD DE LAS FUERZAS ARMADAS

DESARROLLO WEB AVANZADO

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

---

## Prueba 1

---

*Estudiantes:*

Marlyn Almeida, Erick Andrade, Josué Merino

*Docente:*

Ing. Diego Saavedra

## 1. Objetivo

Crear y desplegar una aplicación Flask en Railway y desarrollar una aplicación Django que permita gestionar un inventario de productos utilizando Jinja2 para el manejo de plantillas.

## 2. Instrucciones

### 2.1. Laboratorio Ajax

#### 2.1.1. Descripción:

Implementar una aplicación web avanzada que permita buscar y mostrar detalles de Pokémon utilizando AJAX con JavaScript puro.

#### 2.1.2. Tecnologías:

HTML, CSS, JavaScript.

#### 2.1.3. Pasos:

1. Mejorar la aplicación básica de búsqueda de Pokémon para incluir paginación de resultados.
2. Implementar un sistema de filtros avanzados (por tipo, estadísticas, etc.).
3. Optimizar las solicitudes AJAX para mejorar el rendimiento y la experiencia del usuario.

### 2.2. Laboratorio Ninja 2 con Django

#### 2.2.1. Descripción:

Desarrollar una aplicación web avanzada utilizando Django y Jinja2 para gestionar un inventario de productos con características adicionales.

#### 2.2.2. Tecnologías:

Django (Python), Jinja2 (plantillas), HTML, CSS.

#### 2.2.3. Pasos:

1. Ampliar el CRUD básico para productos incluyendo funcionalidades avanzadas como la importación/exportación de datos en formato CSV.

2. Mejorar el diseño de la interfaz para hacerla más intuitiva y responsive.

## 2.3. Laboratorio de Railway con Flask

### 2.3.1. Descripción:

Realizar un despliegue avanzado de una aplicación en Flask utilizando la plataforma Railway con configuraciones personalizadas.

### 2.3.2. Tecnologías:

Flask (Python), Railway (plataforma de hosting).

## 3. Ajax

Se crea un archivo index.html donde se crea la estructura básica del proyecto, después se crea un archivo app.js, también un styles.css donde se agregan los estilos para una mejor interfaz visual

A continuación la codificación:

index.html:

```
1<!DOCTYPE html>
2<html lang="es">
3<head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6    <meta name="viewport" content="width=device-width, initial-scale
=1.0">
7    <title>Pokemon</title>
8    <link rel="stylesheet" href="styles.css">
9</head>
10<body>
11    <h1>Pokemon</h1>
12    <input type="text" id="search" placeholder="Buscar pokemon">
13    <button id="search-button">Buscar</button>
14    <button id="random-button">Buscar Pokemon Aleatorio</button>
15    <button id="clear-button">Limpiar</button>
16    <div id="spinner" style="display: none;">Cargando...</div>
17    <div id="error-message"></div>
18    <div id="pokemon"></div>
19    <div id="filters">
20        <select id="type-filter">
21            <option value="">Seleccionar Tipo</option>
22            <option value="normal">Normal</option>
23            <option value="fighting">Lucha</option>
```

# Prueba 1

---

```
24      <option value="flying">Volador</option>
25      <option value="poison">Veneno</option>
26      <option value="ground">Tierra</option>
27      <option value="rock">Roca</option>
28      <option value="bug">Bicho</option>
29      <option value="ghost">Fantasma</option>
30      <option value="steel">Acero</option>
31      <option value="fire">Fuego</option>
32      <option value="water">Agua</option>
33      <option value="grass">Planta</option>
34      <option value="electric">El ctrico </option>
35      <option value="psychic">Ps quico </option>
36      <option value="ice">Hielo</option>
37      <option value="dragon">Drag n </option>
38      <option value="dark">Oscuro</option>
39      <option value="fairy">Hada</option>
40      <option value="unknown">Desconocido</option>
41      <option value="shadow">Sombra</option>
42  </select>
43  <button id="filter-button">Filtrar</button>
44</div>
45<div id="advanced-filters">
46    <input type="number" id="min-height" placeholder="Altura
47      m nima">
48    <input type="number" id="max-height" placeholder="Altura
49      m xima">
50    <input type="number" id="min-weight" placeholder="Peso m nimo
51      ">
52    <input type="number" id="max-weight" placeholder="Peso m ximo
53      ">
54    <input type="number" id="min-base-exp" placeholder="Exp Base
55      m nima">
56    <input type="number" id="max-base-exp" placeholder="Exp Base
57      m xima">
58    <button id="advanced-filter-button">Filtrar Avanzado</button>
59</div>
60<table id="pokemon-table">
61  <thead>
62    <tr>
63      <th>Nombre</th>
64      <th>Imagen</th>
65      <th>Altura</th>
66      <th>Peso</th>
67    </tr>
68  </thead>
69  <tbody>
70  </tbody>
71</table>
72<div id="pagination">
```

## Prueba 1

---

```
67      <button id="prev-page">Anterior</button>
68      <span id="page-number">1</span>
69      <button id="next-page">Siguiente</button>
70  </div>
71  <script src="app.js"></script>
72</body>
73</html>
```

styles.css:

```
1 body {
2     font-family: Arial, sans-serif;
3     margin: 0;
4     padding: 0;
5     display: flex;
6     flex-direction: column;
7     align-items: center;
8     background-color: #f8f9fa;
9 }
10
11 h1 {
12     margin: 1rem 0;
13     color: #007bff;
14 }
15
16 input, select, button {
17     padding: 0.5rem;
18     margin: 0.5rem;
19     border: 1px solid #ddd;
20     border-radius: 0.25rem;
21 }
22
23 button {
24     background-color: #007bff;
25     color: white;
26     cursor: pointer;
27 }
28
29 button:hover {
30     background-color: #0056b3;
31 }
32
33 #spinner {
34     font-size: 1.5rem;
35     margin: 1rem 0;
36 }
```

## Prueba 1

---

```
38 #pokemon, #filters, #advanced-filters, #pagination {
39     margin: 1rem 0;
40 }
41
42 #error-message {
43     color: red;
44 }
45
46 #pokemon-table {
47     width: 80%;
48     margin: 1rem 0;
49     border-collapse: collapse;
50     box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
51 }
52
53 #pokemon-table th, #pokemon-table td {
54     border: 1px solid #ddd;
55     padding: 0.5rem;
56     text-align: center;
57 }
58
59 #pokemon-table th {
60     background-color: #007bff;
61     color: white;
62 }
63
64 #pokemon-table tr:nth-child(even) {
65     background-color: #f2f2f2;
66 }
67
68 img {
69     width: 100px;
70     height: 100px;
71     margin: 0.5rem 0;
72 }
73
74 #pagination {
75     display: flex;
76     justify-content: center;
77     align-items: center;
78 }
79
80 #pagination button {
81     margin: 0 0.5rem;
82 }
```

script.js:

## Prueba 1

---

```
1 let currentPage = 1;
2 const pageSize = 20;
3 const typeFilter = document.getElementById('type-filter');
4 const pageNumber = document.getElementById('page-number');
5
6 const addPokemonToTable = (pokemonData) => {
7     const tableBody = document.querySelector('#pokemon-table tbody');
8     const row = document.createElement('tr');
9     row.innerHTML =
10         <td>${pokemonData.name}</td>
11         <td></td>
13         <td>${pokemonData.height}</td>
14         <td>${pokemonData.weight}</td>
15     ;
16     tableBody.appendChild(row);
17 };
18
19 document.getElementById('search-button').addEventListener('click', () => {
20     const search = document.getElementById('search').value.toLowerCase();
21     const errorMessage = document.getElementById('error-message');
22     const pokemon = document.getElementById('pokemon');
23     const spinner = document.getElementById('spinner');
24     errorMessage.innerHTML = '';
25     pokemon.innerHTML = '';
26     spinner.style.display = 'block';
27
28     fetch(`https://pokeapi.co/api/v2/pokemon/${search}`)
29         .then(response => {
30             spinner.style.display = 'none';
31             if (!response.ok) {
32                 throw new Error('Pokémon no encontrado');
33             }
34             return response.json();
35         })
36         .then(data => {
37             pokemon.innerHTML =
38                 <h2>${data.name}</h2>
39                 
41                 <p>Altura: ${data.height}</p>
42                 <p>Peso: ${data.weight}</p>
43             ;
44             addPokemonToTable(data);
45         })
46         .catch(error => {
```

## Prueba 1

---

```
45     errorMessage.innerHTML = `<p style="color: red;">${error.
46         message}</p>`;
47   });
48
49 document.getElementById('random-button').addEventListener('click', () => {
50   const randomId = Math.floor(Math.random() * 898) + 1;
51   const errorMessage = document.getElementById('error-message');
52   const pokemon = document.getElementById('pokemon');
53   const spinner = document.getElementById('spinner');
54   errorMessage.innerHTML = ''; // Clear previous errors
55   pokemon.innerHTML = ''; // Clear previous results
56   spinner.style.display = 'block'; // Show spinner
57
58   fetch('https://pokeapi.co/api/v2/pokemon/${randomId}')
59     .then(response => {
60       spinner.style.display = 'none'; // Hide spinner
61       if (!response.ok) {
62         throw new Error('Pokémon no encontrado');
63       }
64       return response.json();
65     })
66     .then(data => {
67       pokemon.innerHTML =
68         `<h2>${data.name}</h2>
69         
71         <p>Altura: ${data.height}</p>
72         <p>Peso: ${data.weight}</p>
73         `;
74       addPokemonToTable(data);
75     })
76     .catch(error => {
77       errorMessage.innerHTML = `<p style="color: red;">${error.
78         message}</p>`;
79     });
80
81 document.getElementById('clear-button').addEventListener('click', () => {
82   document.getElementById('pokemon').innerHTML = '';
83   document.getElementById('error-message').innerHTML = '';
84   document.querySelector('#pokemon-table tbody').innerHTML = '';
85 });
86 document.getElementById('filter-button').addEventListener('click', () => {
87   currentPage = 1;
```

## Prueba 1

---

```
88     fetchPokemonByType(typeFilter.value);
89 });
90
91 document.getElementById('prev-page').addEventListener('click', () => {
92     if (currentPage > 1) {
93         currentPage--;
94         fetchPokemonByType(typeFilter.value);
95     }
96 });
97
98 document.getElementById('next-page').addEventListener('click', () => {
99     currentPage++;
100    fetchPokemonByType(typeFilter.value);
101 });
102
103 const fetchPokemonByType = (type) => {
104     const offset = (currentPage - 1) * pageSize;
105     const spinner = document.getElementById('spinner');
106     const tableBody = document.querySelector('#pokemon-table tbody');
107     const errorMessage = document.getElementById('error-message');
108     tableBody.innerHTML = ''; // Clear previous results
109     spinner.style.display = 'block'; // Show spinner
110     errorMessage.innerHTML = ''; // Clear previous errors
111
112     fetch(`https://pokeapi.co/api/v2/type/${type}`)
113         .then(response => {
114             if (!response.ok) {
115                 throw new Error('Error al obtener el tipo de
116                               Pok   mon');
117             }
118             return response.json();
119         })
120         .then(data => {
121             const pokemonList = data.pokemon.slice(offset, offset +
122                 pageSize);
123             if (pokemonList.length === 0) {
124                 throw new Error('No hay m   s Pok   mon de este tipo
125                               ');
126             }
127             return Promise.all(pokemonList.map(p => fetch(p.pokemon.
128                 url).then(res => res.json())));
129         })
130         .then(pokemonData => {
131             spinner.style.display = 'none'; // Hide spinner
132             pokemonData.forEach(data => addPokemonToTable(data));
133             pageNumber.innerText = currentPage;
134         })
135         .catch(error => {
136             spinner.style.display = 'none'; // Hide spinner
```

## Prueba 1

---

```
133     errorMessage.innerHTML = `<p style="color: red;">${error.
134         message}</p>`;
135     });
136 }
137 document.getElementById('advanced-filter-button').addEventListener('
138     click', () => {
139     currentPage = 1;
140     fetchFilteredPokemon();
141 });
142 const fetchFilteredPokemon = () => {
143     const minHeight = document.getElementById('min-height').value;
144     const maxHeight = document.getElementById('max-height').value;
145     const minWeight = document.getElementById('min-weight').value;
146     const maxWeight = document.getElementById('max-weight').value;
147     const minBaseExp = document.getElementById('min-base-exp').value;
148     const maxBaseExp = document.getElementById('max-base-exp').value;
149     const spinner = document.getElementById('spinner');
150     const tableBody = document.querySelector('#pokemon-table tbody');
151     const errorMessage = document.getElementById('error-message');
152     tableBody.innerHTML = ''; // Clear previous results
153     spinner.style.display = 'block'; // Show spinner
154     errorMessage.innerHTML = ''; // Clear previous errors
155
156     fetch('https://pokeapi.co/api/v2/pokemon?limit=1000')
157         .then(response => {
158             if (!response.ok) {
159                 throw new Error('Error al obtener la lista de
160                     Pok mon');
161             }
162             return response.json();
163         })
164         .then(data => {
165             const filteredPokemonPromises = data.results.map(pokemon
166                 => {
167                     return fetch(pokemon.url)
168                         .then(res => res.json())
169                         .then(details => {
170                             if (
171                                 (!minHeight || details.height >= minHeight
172                                     ) &&
173                                     (!maxHeight || details.height <= maxHeight
174                                         ) &&
175                                         (!minWeight || details.weight >= minWeight
176                                             ) &&
177                                             (!maxWeight || details.weight <= maxWeight
178                                                 ) &&
179                                                 (!minBaseExp || details.base_experience >=
```

## Prueba 1

---

```
174             minBaseExp) &&
175             (!maxBaseExp || details.base_experience <=
176                 maxBaseExp)
176         ) {
177             return details;
177         }
178     );
179 });
180
181     return Promise.all(filteredPokemonPromises);
182 }
183 .then(filteredData => {
184     spinner.style.display = 'none'; // Hide spinner
185     const validData = filteredData.filter(data => data !==
186         undefined);
186     if (validData.length === 0) {
187         throw new Error('No hay Pok mon que coincidan con
187             los filtros');
188     }
189     validData.forEach(data => addPokemonToTable(data));
190 })
191 .catch(error => {
192     spinner.style.display = 'none'; // Hide spinner
193     errorMessage.innerHTML = '<p style="color: red;">${error.
193         message}</p>';
194 });
195};
```

A continuación las capturas del aplicativo:

## Prueba 1

---

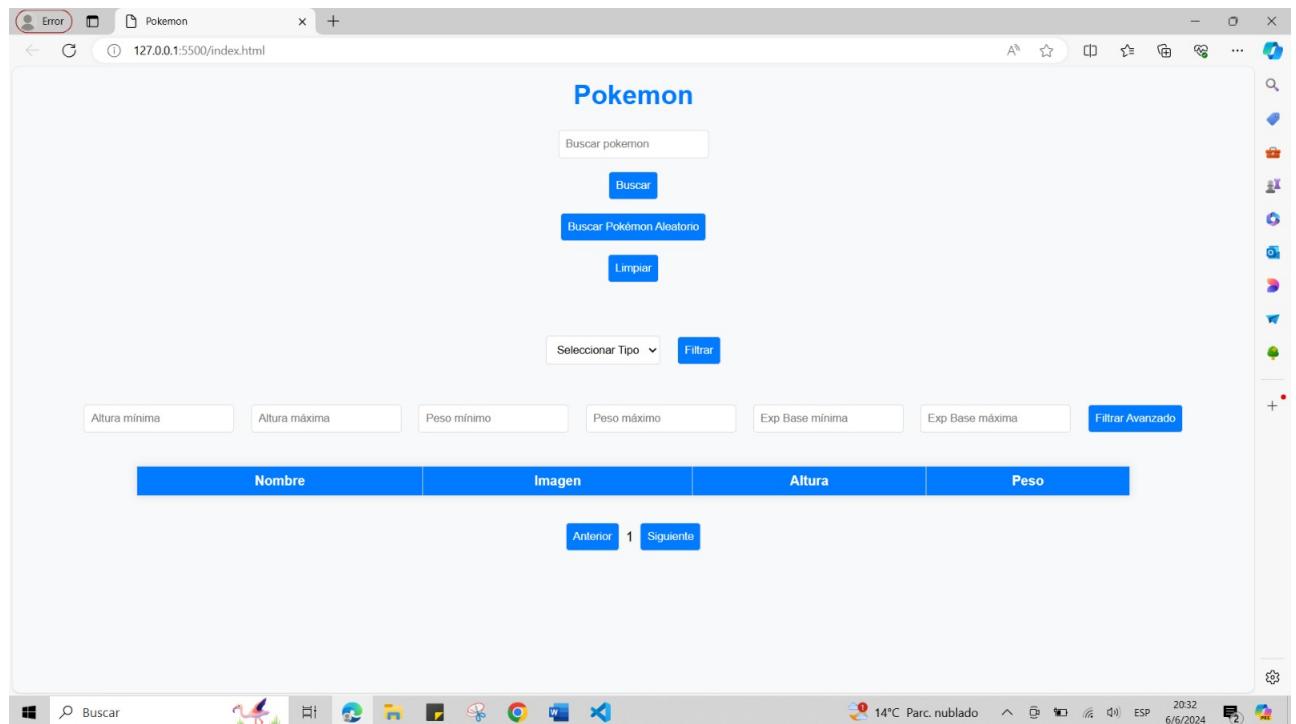


Figura 1: Laboratorio Ajax

## Prueba 1

---

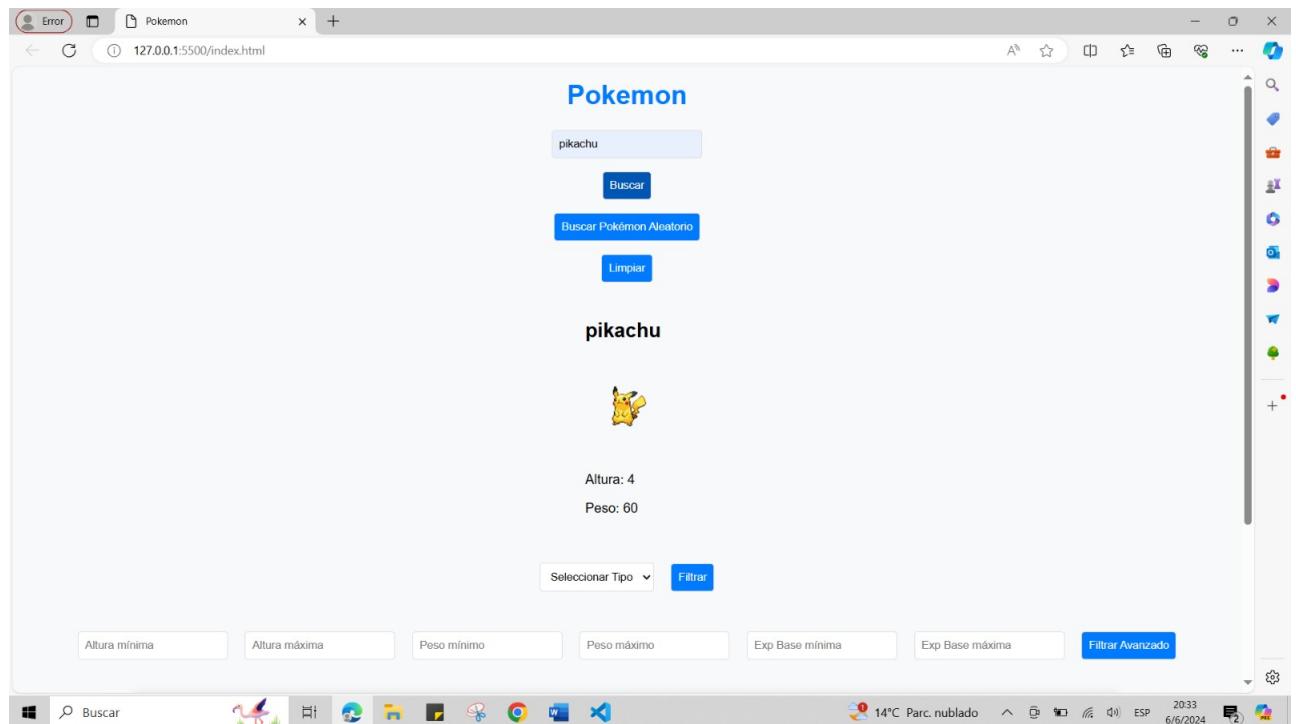
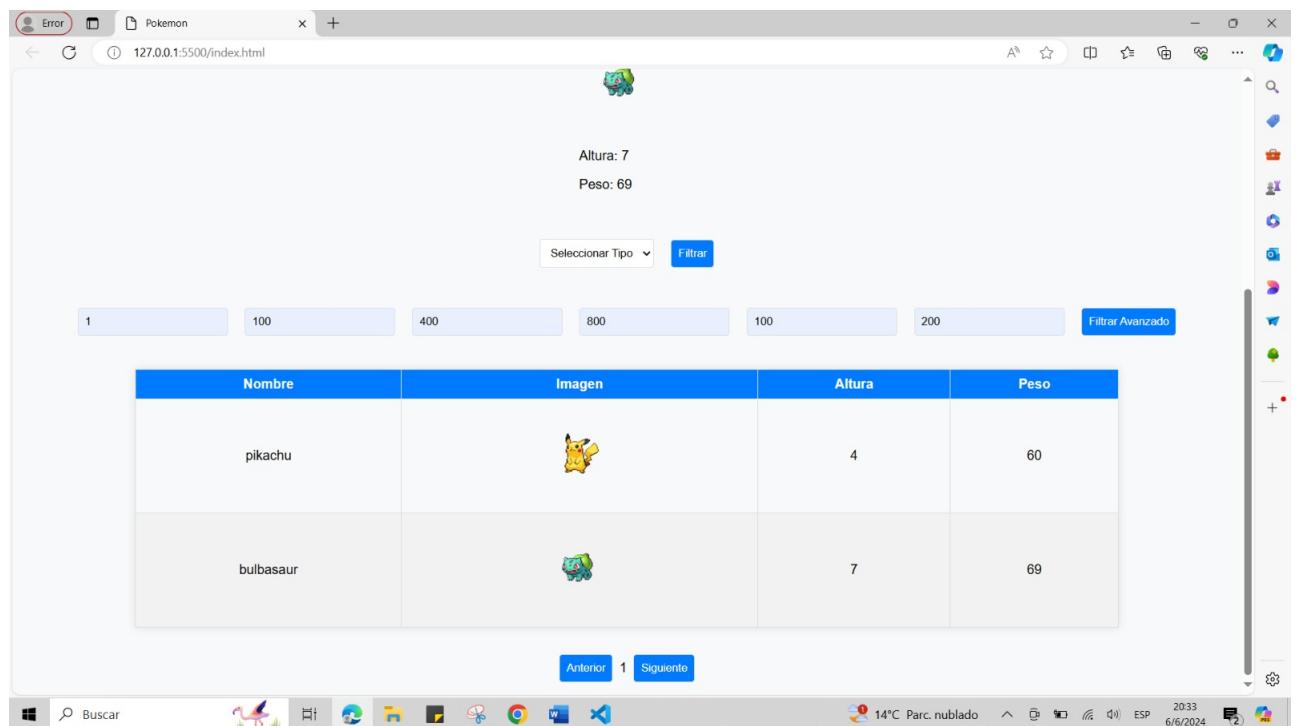


Figura 2: Laboratorio Ajax

## Prueba 1

---



The screenshot shows a web browser window titled "Pokemon" with the URL "127.0.0.1:5500/index.html". The page displays a table of Pokémon data. At the top, there are two status messages: "Altura: 7" and "Peso: 69". Below these are two buttons: "Seleccionar Tipo" and "Filtrar". Underneath the table are several input fields with dropdown menus containing values like 1, 100, 400, 800, 100, and 200. To the right of these is a blue button labeled "Filtrar Avanzado". The table has four columns: "Nombre", "Imagen", "Altura", and "Peso". It contains two rows of data:

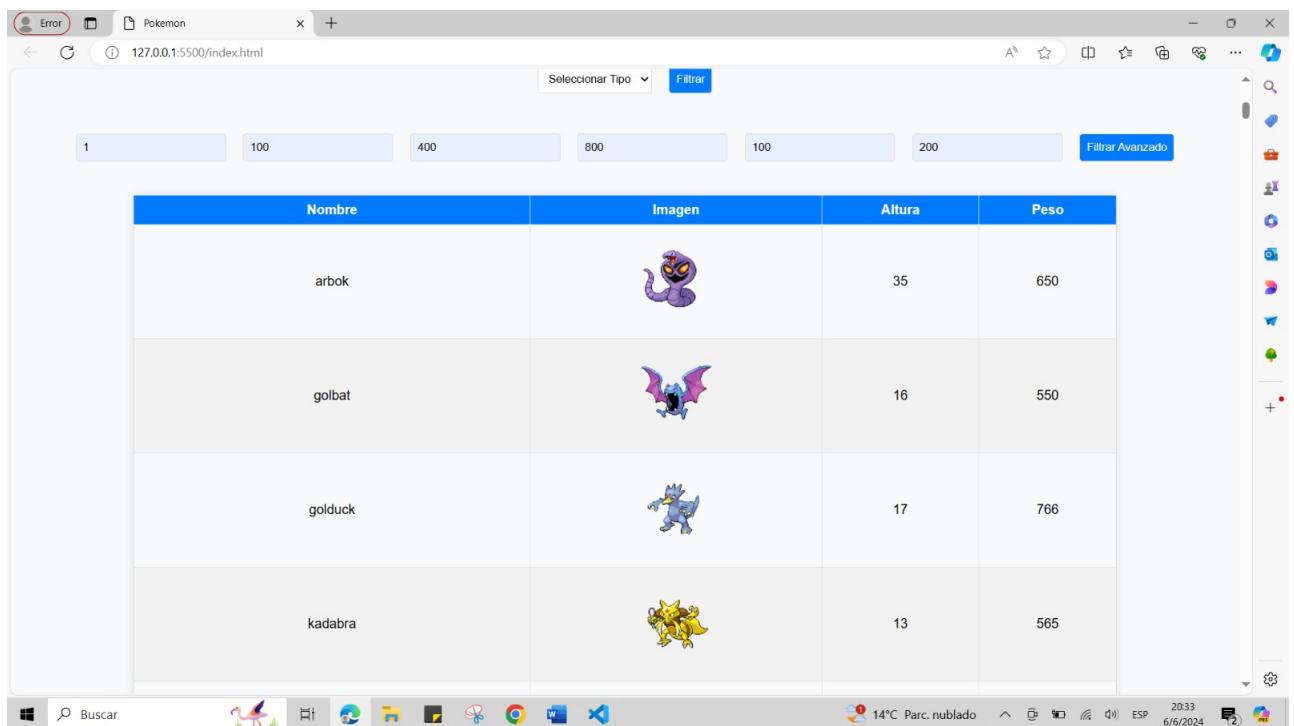
Nombre	Imagen	Altura	Peso
pikachu		4	60
bulbasaur		7	69

At the bottom of the table are navigation buttons: "Anterior", a page number "1", and "Siguiente". The browser's taskbar at the bottom shows various pinned icons and the system tray on the right.

Figura 3: Laboratorio Ajax

## Prueba 1

---



The screenshot shows a web browser window titled "Pokemon" with the URL "127.0.0.1:5500/index.html". The page displays a table of four Pokemon entries: arbok, golbat, golduck, and kadabra. Each entry includes the name, a small image of the Pokemon, its height (Altura), and its weight (Peso). The browser interface includes a search bar, filter buttons, and a toolbar at the bottom.

Nombre	Imagen	Altura	Peso
arbok		35	650
golbat		16	550
golduck		17	766
kadabra		13	565

Figura 4: Laboratorio Ajax

## Prueba 1

---

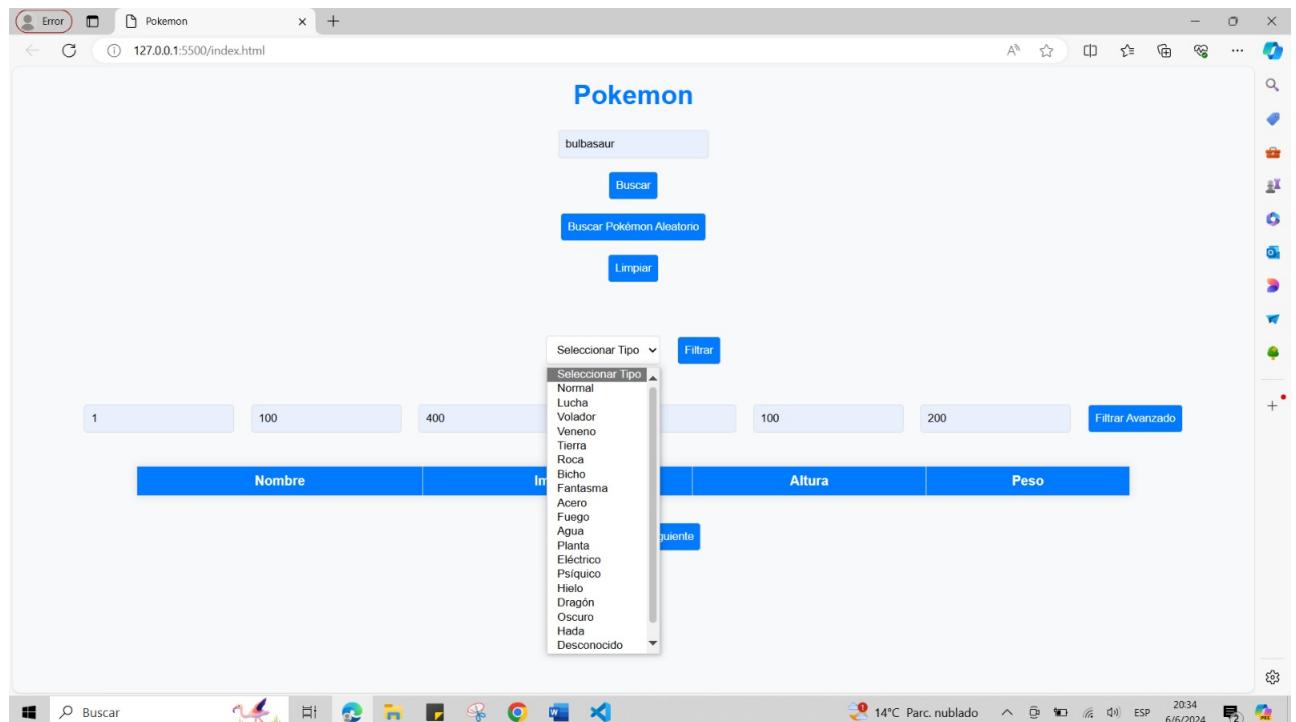


Figura 5: Laboratorio Ajax

Link del repositorio: [https://github.com/ejmerino/grupo1\\_webavanzada](https://github.com/ejmerino/grupo1_webavanzada)

## 4. Jinja 2 con Django

Se crea un proyecto, acto seguido se cree un entorno virtual para el proyecto con la instrucción por consola:

```
1 python3 -m venv venv
```

se activa el entorno virtual con:

```
1 venv\Scripts\activate
```

Se instala el entorno virtual con el siguiente comando:

```
1 pip install django==4.2
```

## Prueba 1

---

Se crea un proyecto llamado inventario con el siguiente comando:

```
1 django-admin startproject inventario .
```

Se crea una aplicación llamada productos con el siguiente comando:

```
1 python manage.py startapp productos
```

Se agrega productos al archivo settings.py del proyecto:

```
1 INSTALLED_APPS = [
2 ...
3 'productos',
4 ]
```

Crea un modelo llamado Producto en la aplicación productos con los siguientes campos:

```
1 from django.db import models
2
3 class Producto(models.Model):
4     nombre = models.CharField(max_length=100)
5     precio = models.DecimalField(max_digits=10, decimal_places=2)
6     cantidad = models.IntegerField()
7
8     def __str__(self):
9         return self.nombre
```

Se registra el modelo Producto en el archivo admin.py de la aplicación productos:

```
1 from django.contrib import admin
2 from .models import Producto
3
4 admin.site.register(Producto)
```

Se Crea una vista llamada listar\_productos en la aplicación productos que muestre todos los productos registrados en el sistema:

```
1 from django.shortcuts import render
2 from .models import Producto
3
```

## Prueba 1

---

```
4| productos = []
5|
6| def listar_productos(request):
7|     productos = Producto.objects.all()
8|     return render(request, 'listar.html', {'productos': productos})
```

Se crea el archivo base.html en la carpeta templates de la aplicación productos con el siguiente contenido:

```
1| <!DOCTYPE html>
2| <html lang="es">
3| <head>
4|     <meta charset="UTF-8">
5|     <meta name="viewport" content="width=device-width, initial-scale
6|         =1.0">
7|     <title>
8|         {% block title %} 
9|             Inventario
10|            {% endblock %}
11|     </title>
12|     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/
13|             bootstrap.min.css" rel="stylesheet" integrity="sha384-
14|             QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXhOJMhjY6hW+
15|             ALEWIH" crossorigin="anonymous">
16| </head>
17| <body>
18|     <div class="container">
19|         {% block content %} 
20|             {% endblock %}
21|     </div>
22|     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/
23|             bootstrap.bundle.min.js" integrity="sha384-
24|             YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDz0xhy9GkcIds1K1eN7N6jIeHz"
25|             crossorigin="anonymous"></script>
26| </body>
27| </html>
```

Se crea una plantilla llamada listar.html en la carpeta productos/templates con el siguiente contenido:

```
1| <!DOCTYPE html>
2| <html lang="es">
3| <head>
4|     <meta charset="UTF-8">
```

## Prueba 1

---

```
5      <meta name="viewport" content="width=device-width, initial-scale
6          =1.0">
7      <title>{% block title %}Productos{% endblock %}</title>
8      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/
9          bootstrap.min.css" rel="stylesheet" integrity="sha384-
10         QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+
11         ALEwIH" crossorigin="anonymous">
12     <style>
13         @import url('https://fonts.googleapis.com/css2?family=Roboto:
14             wght@400;500;700&display=swap');
15
16     body {
17         font-family: 'Roboto', sans-serif;
18         background-color: #f2f2f2;
19         color: #333;
20     }
21
22     .container {
23         margin-top: 50px;
24         background-color: #ffffff;
25         border-radius: 8px;
26         box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2);
27         padding: 20px;
28         text-align: center;
29     }
30
31     h1 {
32         color: #333333;
33         margin-bottom: 20px;
34         font-size: 28px;
35         font-weight: 700;
36         text-transform: uppercase;
37         letter-spacing: 2px;
38         text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);
39     }
40
41     table {
42         width: 80%;
43         margin: 0 auto;
44         border-collapse: collapse;
45         border-radius: 8px;
46         overflow: hidden;
47         box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
48         text-align: center; /* Centrar contenido de la tabla */
49     }
50
51     th, td {
52         padding: 12px 15px;
53         border-bottom: 1px solid #dddddd;
```

## Prueba 1

---

```
49         text-align: center; /* Centrar contenido de las celdas */
50     }
51
52     th {
53         background-color: #f2f2f2;
54         text-transform: uppercase;
55         cursor: pointer;
56         position: relative;
57     }
58
59     th:hover {
60         background-color: #e0e0e0;
61     }
62
63     tbody tr:nth-child(even) {
64         background-color: #f9f9f9;
65     }
66
67     tbody tr:hover {
68         background-color: #f1f1f1;
69         transition: background-color 0.3s;
70         transform: scale(1.02);
71     }
72
73     th:hover, th:hover ~ td {
74         background-color: #f1f1f1;
75     }
76
77     tbody tr td:hover {
78         background-color: #f1f1f1;
79         transition: background-color 0.3s;
80         transform: scale(1.02);
81     }
82
83     th:hover::after {
84         content: ' ';
85         display: block;
86         position: absolute;
87         top: 0;
88         left: 0;
89         width: 100%;
90         height: 100%;
91         background-color: #f1f1f1;
92         z-index: -1;
93         transition: background-color 0.3s;
94     }
95
96     th:hover, th:hover ~ td {
97         background-color: #f1f1f1;
```

## Prueba 1

---

```
98         transition: background-color 0.3s;
99     }
100
101    tbody tr td:hover, tbody tr th:hover {
102        background-color: #f1f1f1;
103        transition: background-color 0.3s;
104        transform: scale(1.02);
105    }
106
107    .btn-export {
108        margin-top: 20px;
109        margin-bottom: 20px;
110        background-color: #4CAF50;
111        color: white;
112        padding: 10px 20px;
113        border: none;
114        border-radius: 4px;
115        cursor: pointer;
116        transition: background-color 0.3s, box-shadow 0.3s;
117        position: relative;
118        font-weight: 500;
119        box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
120    }
121
122    .btn-export:hover {
123        background-color: #45a049;
124        box-shadow: 0 6px 8px rgba(0, 0, 0, 0.1);
125    }
126
127    .btn-export i {
128        position: absolute;
129        left: 10px;
130    }
131
132    footer {
133        background-color: #333;
134        color: white;
135        text-align: center;
136        padding: 20px 0;
137        position: fixed;
138        width: 100%;
139        bottom: 0;
140        left: 0;
141        font-weight: 500;
142    }
143    </style>
144</head>
145<body>
146    <div class="container">
```

## Prueba 1

---

```
147  {%- block content %}            
148  <h1>Productos</h1>            
149  <table id="productTable">        
150  <thead>                        
151  <tr>                            
152  <th>Nombre</th>                
153  <th>Precio</th>                
154  <th>Cantidad</th>               
155  </tr>                           
156  </thead>                       
157  <tbody>                        
158  {% for producto in productos %}   
159  <tr>                            
160  <td>{{ producto.nombre }}</td>   
161  <td>$ {{ producto.precio }}</td>   
162  <td>{{ producto.cantidad }}</td>   
163  </tr>                           
164  {% endfor %}                   
165  </tbody>                       
166  </table>                       
167  <button class="btn btn-primary btn-export" onclick="    
168    exportarCSV() "><i class="bi bi-download"></i> Descargar</    
169  button>                       
170  {% endblock %}                  
171  </div>                          
172  <footer>                        
173  Marlyn Almeida, Erick Andrade, Josu Merino - 2024. Todos    
174  los derechos reservados          
175  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/    
176  bootstrap.bundle.min.js" integrity="sha384-    
177  YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDz0xhy9GkcIds1K1eN7N6jIeHz    
178  " crossorigin="anonymous"></script>    
179  <script>                        
180  function exportarCSV() {         
181  var productos = [];             
182    
183  {% for producto in productos %}   
184  productos.push({              
185  nombre: "{{ producto.nombre }}",    
186  precio: "{{ producto.precio }}",    
187  cantidad: "{{ producto.cantidad }}"    
188  });                           
189  {% endfor %}                   
190    
191  var csvContent = "data:text/csv;charset=utf-8,";    
192  csvContent += "Nombre;Precio;Cantidad\n";    
193    
194  productos.forEach(function(producto) {    
195  
```

## Prueba 1

---

```
190     var dataString = producto.nombre + ";" + producto.
191         precio + ";" + producto.cantidad;
192     csvContent += dataString + "\n";
193   });
194
195   var encodedUri = encodeURI(csvContent);
196   var link = document.createElement("a");
197   link.setAttribute("href", encodedUri);
198   link.setAttribute("download", "productos.csv");
199   document.body.appendChild(link);
200   link.click();
201   document.body.removeChild(link);
202 }
203 </script>
204 </body>
205 </html>
```

Se crea una URL llamada listar\_productos en el archivo urls.py de la aplicación productos:

```
1 from django.urls import path
2 from . import views
3
4 urlpatterns = [
5     # URLs de vistas normales
6     path('', views.listar_productos, name='listar_productos'),
7 ]
```

Se agrega la URL de la aplicación productos al archivo urls.py del proyecto:

```
1 from django.contrib import admin
2 from django.urls import path, include
3
4 urlpatterns = [
5     path('admin/', admin.site.urls),
6     path('', include('productos.urls'))
7 ]
```

Se realizan las migraciones de la aplicación productos con los siguientes comandos:

```
1 python manage.py makemigrations productos
2 python manage.py migrate
```

## Prueba 1

---

Se crea un superusuario con el siguiente comando:

```
1 python manage.py createsuperuser
```

Por último se ejecuta el servidor con el siguiente comando:

```
1 python manage.py runserver
```

Capturas del aplicativo:

PRODUCTOS		
NOMBRE	PRECIO	CANTIDAD
Camisa	\$ 25,00	6
Pantalon	\$ 10,00	4
Gorra	\$ 7,00	9

[Descargar](#)

Figura 6: Django

Administración de Django

Sitio administrativo

AUTENTICACIÓN Y AUTORIZACIÓN

- Grupos [Añadir](#) [Modificar](#)
- Usuarios [Añadir](#) [Modificar](#)

PRODUCTOS

- Productos [Añadir](#) [Modificar](#)

Acciones recientes

Mis acciones

- + Gorra Producto
- + Pantalon Producto
- editar Camisa Producto
- + Camisa Producto

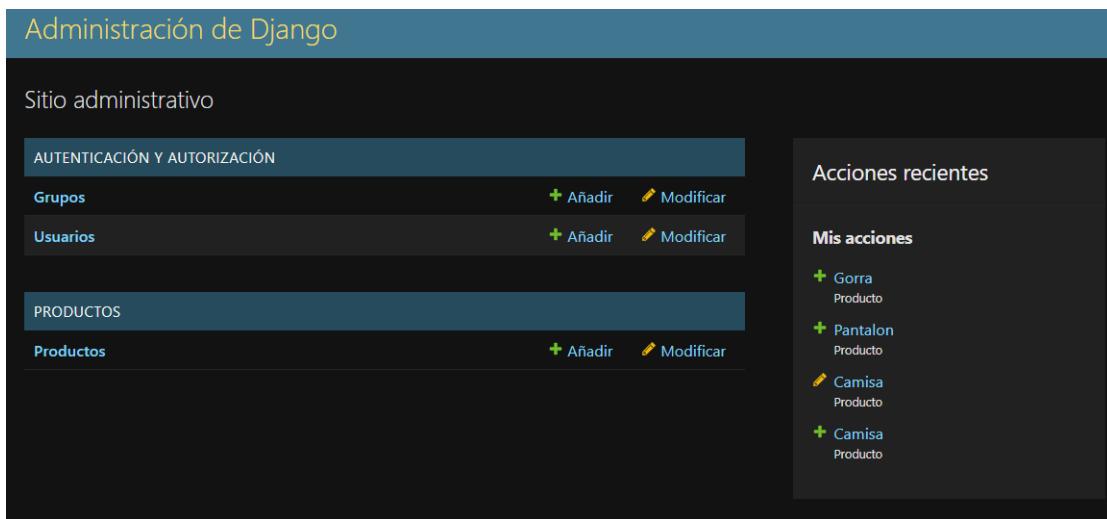


Figura 7: Django

## Prueba 1

---

Administración de Django

Inicio > Productos > Productos > Añadir producto

Empiece a escribir para filtrar...

AUTENTICACIÓN Y AUTORIZACIÓN

Grupos [+ Añadir](#)

Usuarios [+ Añadir](#)

PRODUCTOS

Productos [+ Añadir](#)

Añadir producto

Nombre: Zapatos

Precio: 65

Cantidad: 14

[GUARDAR](#) [Guardar y añadir otro](#) [Guardar y continuar editando](#)

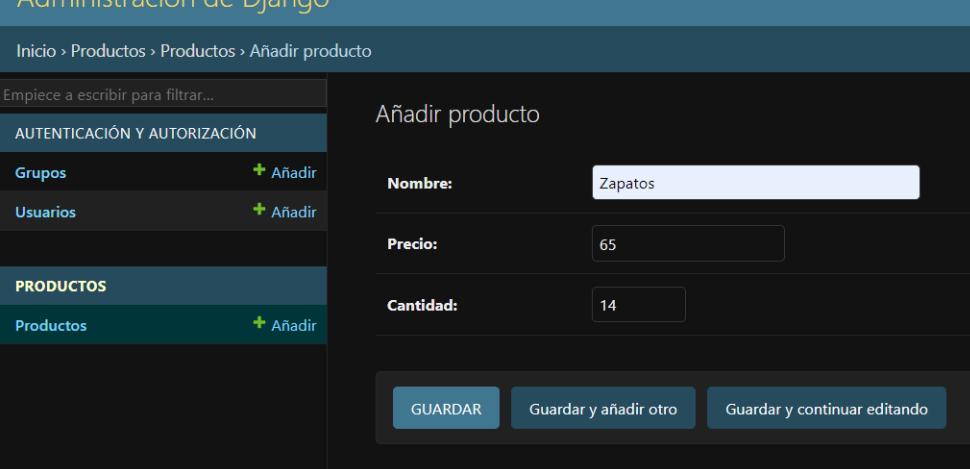


Figura 8: Django

**PRODUCTOS**

NOMBRE	PRECIO	CANTIDAD
Camisa	\$ 25,00	6
Pantalon	\$ 10,00	4
Gorra	\$ 7,00	9
Zapatos	\$ 65,00	14

[Descargar](#)



Figura 9: Django

	A	B	C
1	Nombre	Precio	Cantidad
2	Camisa	25	6
3	Pantalon	10	4
4	Gorra	7	9
5	Zapatos	65	14
6			
7			

Figura 10: Django

El repositorio: <https://github.com/Fuerzas-Armadas-ESPE/laboratorio-jinga2-group-01>

## 5. Railway con Flask

### 5.1. Paso 1: Creación de la Estructura de Directorios

Crea un directorio principal llamado "nintendo\_page". Dentro de este directorio, organiza los siguientes elementos:

- `main.py`: Contiene el código principal de la aplicación Flask.
- `static/`: Carpeta para las imágenes de los juegos.
- `templates/`: Carpeta para los archivos HTML de la aplicación.
- `railway.json`: Archivo de configuración para desplegar en Railway.
- `.gitignore`: Archivo para especificar archivos y directorios a ignorar en Git.
- `Readme.md`: Documentación del proyecto.

### 5.2. Paso 2: Iniciamos con la creación de la aplicación Flask.

Para ello vamos a crear un directorio llamado flask, vamos a seguir las buenas prácticas y vamos a crear un entorno virtual para instalar las dependencias de la aplicación.

```
1 python3 -m venv venv
2 source venv/bin/activate
```

### 5.3. Paso 3: Instalamos las dependencias de la aplicación.

```
1 pip install flask gunicorn
```

### 5.4. Paso 4: Creamos una carpeta llamada templates donde crearemos el archivo index.html con el siguiente contenido. .

```
1 from flask import Flask, render_template # Importa las funciones
    necesarias de Flask
2 import os # Importa el módulo os para manejar variables de entorno
3
```

## Prueba 1

---

```
4 app = Flask(__name__) # Crea una instancia de la aplicaci n Flask
5
6 @app.route('/')
7 def index():
8     # Historia de Nintendo que se mostrar en la p gina
9     nintendo_history = """
10     Nintendo Co., Ltd. es una empresa multinacional japonesa de
11         electr nica de consumo y videojuegos con sede en Kioto.
12     Nintendo fue fundada el 23 de septiembre de 1889 por Fusajiro
13         Yamauchi y comenz como una empresa que produc a naipes
14         japoneses.
15     Con el tiempo, Nintendo se transform en una de las compa as
16         m s influyentes en la industria de los videojuegos.
17     """
18
19     # Lista de los 6 juegos m s vendidos con su t tulo , ventas ,
20     # resumen y nombre de archivo de imagen
21     top_games = [
22         {"title": "Mario Kart 8 Deluxe", "sales": "33.41 million", "
23             "summary": "Juego de carreras de la serie Mario Kart.", "
24             "image": "mario_kart_8_deluxe.jpg"},,
25         {"title": "Animal Crossing: New Horizons", "sales": "31.18
26             million", "summary": "Simulaci n de vida en una isla
27             desierta.", "image": "animal_crossing_new_horizons.jpg"},,
28         {"title": "Super Smash Bros. Ultimate", "sales": "24.77
29             million", "summary": "Juego de lucha con personajes de
0             Nintendo.", "image": "super_smash_bros_ultimate.jpg"},,
1         {"title": "The Legend of Zelda: Breath of the Wild", "sales": "
2             "23.20 million", "summary": "Aventura en el mundo abierto
3             de Hyrule.", "image": "zelda_breath_of_the_wild.jpg"},,
4         {"title": "Pok mon Sword and Shield", "sales": "21.85 million
5             ", "summary": "Juego RPG de la serie Pok mon.", "image": "
6             pokemon_sword_shield.jpg"},,
7         {"title": "Super Mario Odyssey", "sales": "21.40 million", "
8             "summary": "Aventura de plataformas con Mario.", "image": "
9             super_mario_odyssey.jpg"},,
10     ]
11     # Renderiza la plantilla HTML y pasa las variables history y games
12         a la plantilla
13     return render_template('index.html', history=nintendo_history ,
14         games=top_games)
15
16     if __name__ == '__main__':
17         # Ejecuta la aplicaci n en modo debug en el puerto especificado
18         app.run(debug=True, port=os.getenv("PORT", default=5000))
```

## 5.5. Paso 5: Instalamos las dependencias de la aplicación.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale
6         =1.0">
7     <title>Nintendo Page</title>
8     <!-- Enlace a Google Fonts para la fuente "Press Start 2P" -->
9     <link href="https://fonts.googleapis.com/css2?family=Press+Start+2
10        P&display=swap" rel="stylesheet">
11 <style>
12     body {
13         font-family: 'Press Start 2P', cursive;
14         background-color: #ffffff; /* Fondo blanco */
15         color: #2d3748;
16         margin: 0;
17         padding: 0;
18     }
19     .header, .footer {
20         background-color: #b30000; /* Encabezado y pie de p gina
21             rojo intenso */
22         color: white;
23         text-align: center;
24         padding: 10px 0;
25         position: relative; /* Permite posicionar elementos dentro
26             del encabezado */
27     }
28     .header .logo {
29         position: absolute; /* Posiciona el logotipo de Nintendo
30             */
31         left: 20px;
32         top: 10px;
33         font-size: 1.5em;
34         color: white;
35     }
36     .container {
37         max-width: 1200px;
38         margin: 20px auto;
39         padding: 20px;
40     }
41     h1 {
42         text-align: center;
43         color: #b30000; /* T tulo rojo intenso */
44     }
45     h2 {
46         text-align: center; /* Centrar el t tulo */
47         color: #b30000; /* T tulo rojo intenso */
```

## Prueba 1

---

```
43         font-size: 1.5em; /* Aumentar el tamaño de la fuente */
44     }
45     p {
46         line-height: 1.6;
47     }
48     .catalog {
49         margin-top: 20px;
50         display: flex;
51         flex-wrap: wrap;
52         justify-content: space-around;
53     }
54     .game {
55         width: 30%; /* Ancho ajustado para tres juegos por fila */
56         margin-bottom: 20px;
57         position: relative;
58     }
59     .game img {
60         width: 100%;
61         border-radius: 10px;
62         box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
63         transition: transform 0.3s ease;
64     }
65     .game:hover img {
66         transform: scale(1.05); /* Efecto de zoom al pasar el
67         ratón */
68     }
69     .game-info {
70         display: none;
71         position: absolute;
72         top: 0;
73         left: 0;
74         width: 100%;
75         height: 100%;
76         background-color: rgba(0, 0, 0, 0.8); /* Fondo
77         semitransparente */
78         color: white;
79         border-radius: 10px;
80         padding: 10px;
81         box-sizing: border-box;
82         text-align: center;
83     }
84     .game:hover .game-info {
85         display: block; /* Mostrar información del juego al pasar
86         el ratón */
87     }
88     .game-title {
89         font-weight: bold;
90         margin-bottom: 10px;
91     }
```

## Prueba 1

---

```
89      .game-summary {
90          margin-bottom: 10px;
91      }
92  </style>
93 </head>
94 <body>
95     <div class="header">
96         <div class="logo">Nintendo</div>
97         <h1>Nintendo</h1>
98     </div>
99     <div class="container">
100        <h2>Historia</h2>
101        <p>{{ history }}</p>
102
103        <div class="catalog">
104            <h2>Los 6 juegos m s vendidos de Nintendo</h2>
105            {% for game in games %}
106            <div class="game">
107                
109                <div class="game-info">
110                    <div class="game-title">{{ game.title }}</div>
111                    <div class="game-summary">{{ game.summary }}</div>
112                    <div class="game-sales">Ventas: {{ game.sales }}</
113                        div>
114                </div>
115            {% endfor %}
116        </div>
117        <div class="footer">
118            <p>    2024 Nintendo </p>
119        </div>
120    </body>
121 </html>
```

- 5.6. **Paso 6:** Creamos una carpeta llamada static para poder crear dentro otra carpeta llamada images donde subiremos los archimos .jpg que necesitaremos .
- 5.7. **Paso 7:** Creamos una carpeta llamada static para poder crear dentro otra carpeta llamada images donde subiremos los archimos .jpg que necesitaremos .
- 5.8. **Paso 8:** Creamos el archivo .gitignore, para ello vamos a utilizar el servicio de gitignore.io.
- 5.9. **Paso 9:** Creamos el archivo Readme.md con el siguiente contenido.

```
1 # Página de Nintendo
2
3 Esta es una página web que muestra la historia de Nintendo y un
4 cat logo de juegos.
5
6 ## Estructura de Directorios
7
8 - 'app.py': Código principal de la aplicación Flask.
9 - 'static/': Carpeta para las imágenes de los juegos.
10 - 'templates/': Archivos HTML para las páginas de la aplicación.
11 - 'railway.json': Configuración para desplegar en Railway.
12 - '.gitignore': Archivo de ignorados para Git.
13 - 'Readme.md': Documentación del proyecto.
14
15 ## Ejecución
16
17 Para ejecutar la aplicación, asegúrate de tener Python y Flask
18 instalados. Luego, ejecuta el siguiente comando:
19
20 ````bash
21 python app.py
22 ## Características
23
24 - Python
25 - Flask
26
27 ## Cómo usar
28
29 - Instala los requisitos de Python 'pip install -r requirements.txt'
30 - Inicia el servidor para desarrollo 'python3 main.py'
```

**5.10. Paso 10: Creamos el archivo requirements.txt con el siguiente contenido..**

```
1 click==7.1.2
2 Flask==1.1.2
3 gunicorn==20.0.4
4 itsdangerous==1.1.0
5 Jinja2==2.11.3
6 jsonify==0.5
7 MarkupSafe==1.1.1
8 setuptools==70.0.0
9 Werkzeug==1.0.1
```

- 5.11. **Paso 11:** Subimos nuestro código a un repositorio en GitHub.
- 5.12. **Paso 12:** Iniciamos nuestra cuenta en Railway para poder subir nuestro proyecto.
- 5.13. **Paso 13:** Verificamos que la aplicación se despliegue correctamente.

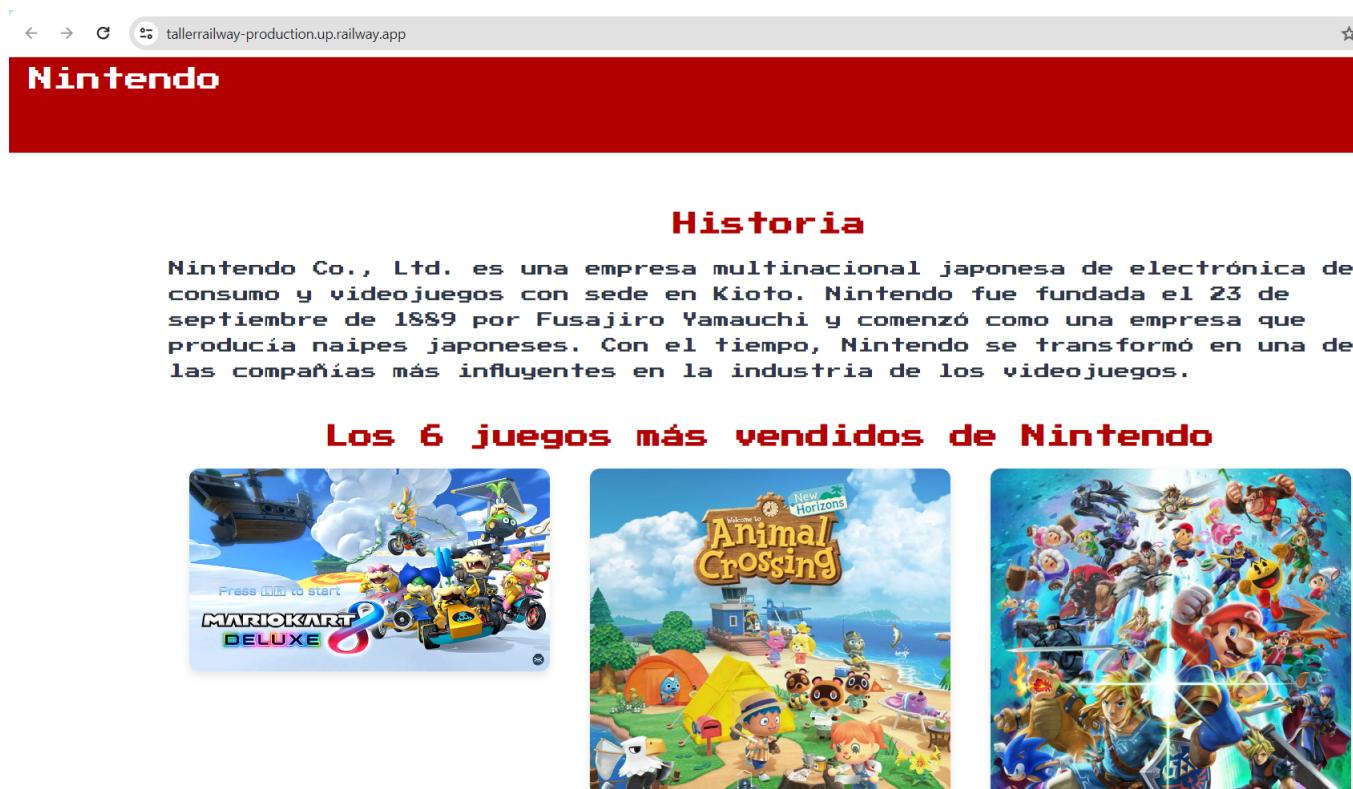


Figura 11: Proyecto desplegado

## Prueba 1

---

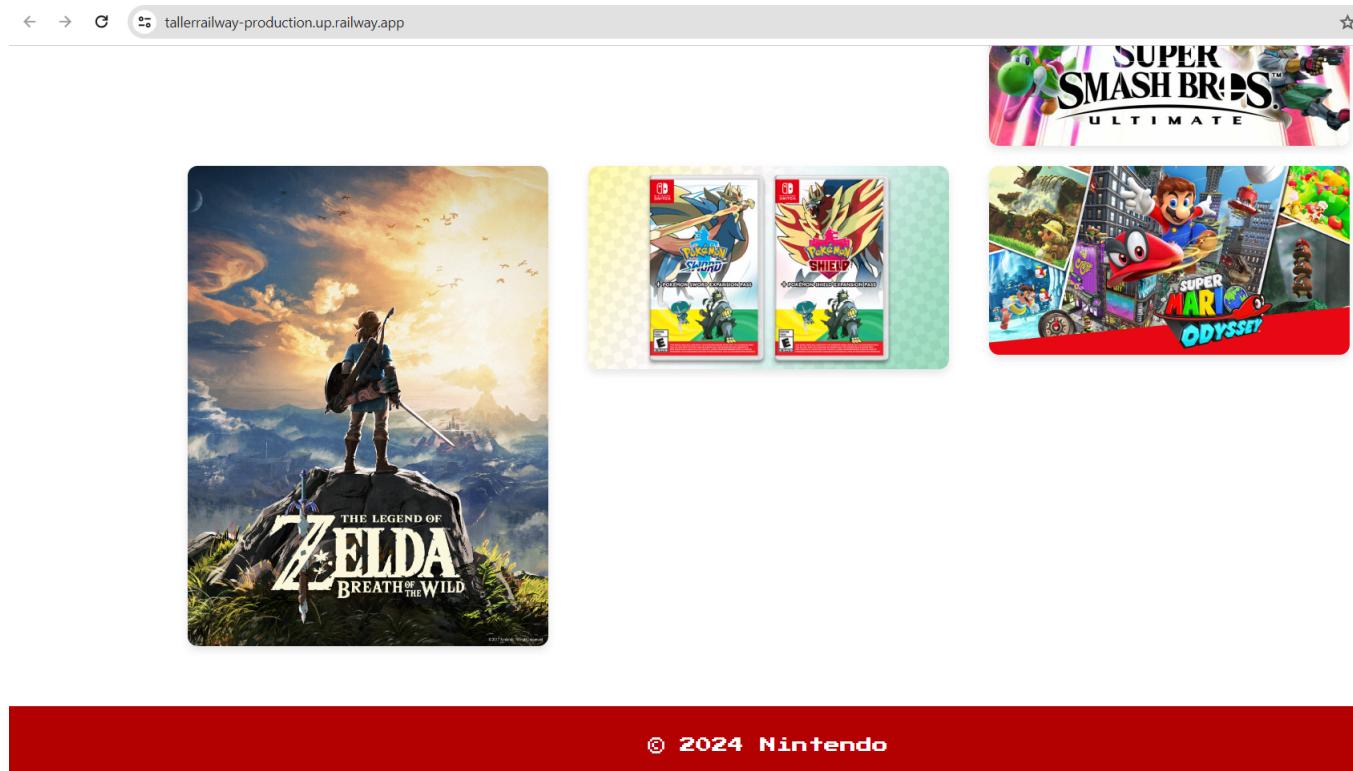


Figura 12: Proyecto desplegado



Figura 13: Proyecto desplegado

El repositorio: <https://github.com/marlyn-almeida/tallerrailway>

## 6. Conclusiones

- Implementación Exitosa de Funcionalidades CRUD con Django: El proyecto permitió implementar exitosamente un conjunto completo de operaciones CRUD (Crear, Leer, Actualizar y Eliminar) para la gestión de productos. Utilizando el framework Django, se logró desarrollar un sistema robusto y eficiente que facilita la administración de productos, demostrando la capacidad de Django para manejar aplicaciones web complejas de manera organizada y estructurada.
- Eficacia de Jinja2 en la Creación de Interfaces de Usuario: La utilización de Jinja2 para la creación de plantillas HTML resultó ser eficaz y eficiente. Jinja2 proporcionó una forma flexible y potente de integrar lógica en las plantillas HTML, permitiendo una experiencia de usuario dinámica y atractiva. La combinación de Django con Jinja2 facilitó la separación de la lógica de negocio y la presentación, promoviendo un desarrollo más limpio y mantenible.

- Integración Satisfactoria de Herramientas de Desarrollo y Colaboración: La subida del proyecto a GitHub representó un paso crucial en la integración de herramientas de desarrollo y colaboración. Utilizando Git para el control de versiones y GitHub para la gestión del repositorio, se estableció una base sólida para el desarrollo colaborativo y la gestión de cambios. Este proceso no solo permitió mantener un historial claro de las modificaciones realizadas, sino que también facilitó la colaboración futura y la revisión de código, subrayando la importancia de utilizar herramientas de control de versiones en proyectos de desarrollo de software.
- En este proyecto, hemos creado una página web utilizando Flask para mostrar la historia de Nintendo y un catálogo de sus juegos más vendidos. El uso de Railway para desplegar nuestra aplicación proporciona varios beneficios significativos, como la facilidad de despliegue al encargarse de la configuración del entorno y la gestión de dependencias, la automatización del proceso de construcción y despliegue mediante configuraciones simples en un archivo como `railway.json`, la escalabilidad que permite manejar un mayor tráfico sin ajustes manuales complejos, y el monitoreo y recuperación automáticos que aseguran que la aplicación se mantenga en funcionamiento con un tiempo de inactividad mínimo. Además, Railway se integra fácilmente con herramientas de control de versiones como GitHub, facilitando un flujo de trabajo de integración continua (CI/CD).

## 7. Referencias

1. Programación web avanzada - Laboratorio Jinga 2. (s.f.). [https://statick88.github.io/programacionWebAvanzada/unidades/laboratorios/3\\_laboratorio\\_jinga2.html](https://statick88.github.io/programacionWebAvanzada/unidades/laboratorios/3_laboratorio_jinga2.html)
2. Programación web avanzada - Laboratorio de Railway. (s.f.). [https://statick88.github.io/programacionWebAvanzada/unidades/laboratorios/2\\_laboratorio\\_railway.html](https://statick88.github.io/programacionWebAvanzada/unidades/laboratorios/2_laboratorio_railway.html)
3. Código comprimido [https://drive.google.com/file/d/1x1R1IE0z-hhy3z\\_fm1JjDz0M94FRw/view?usp=sharing](https://drive.google.com/file/d/1x1R1IE0z-hhy3z_fm1JjDz0M94FRw/view?usp=sharing)