



27-4-2023

# DOCUMENTACIÓN FINAL “THE GAME BAZAR”

Sprint 4 – Integración de  
Sistemas Informáticos (LAB)

JOSÉ LARA NAVARRO  
ALEJANDRO PANIAGUA RODRÍGUEZ  
AITOR MILLÁN TRUJILLO  
GRADO EN INGENIERÍA INFORMÁTICA (TI)

## ÍNDICE

1. INTRODUCCIÓN.....	3
2. PREPARÁNDOLO TODO .....	3
2.1. Crear un entorno virtual en Linux.....	3
2.2. Instalar los requerimientos.....	3
2.3. Arrancar el servidor Flask .....	4
2.4. Accediendo a la página .....	5
3. CARACTERÍSTICAS DE LA PÁGINA.....	6
3.1. Buscador de juegos .....	7
3.2. Top 10 Juegos .....	8
3.3. Inicio de sesión.....	9
3.4. Área de Administración.....	10
4. TESTING AUTOMÁTICO.....	13
5. AUTOMATIZACIÓN DEL DESPLIEGUE.....	14
5.1. Docker .....	14
5.2. Kubernetes .....	15

# 1. INTRODUCCIÓN

Bienvenido a la documentación de usuario de TheGameBazar. Esta página web ha sido desarrollada y probada en un sistema Ubuntu para arrancar el servidor Flask y hacer uso de la página web.

## 2. PREPARÁNDOLO TODO

### 2.1. Crear un entorno virtual en Linux

Antes de arrancar el servidor Flask, es recomendable crear un entorno virtual en Ubuntu para mantener separadas las dependencias del proyecto. Sigue los siguientes pasos para crear un entorno virtual en Ubuntu con el nombre *venvGB*

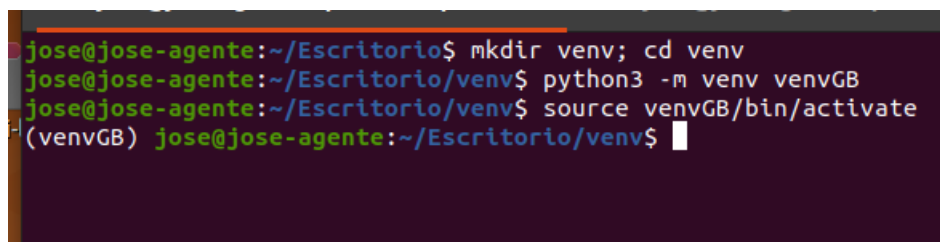
1. Abre una terminal
2. Navega hasta la carpeta donde quieres crear el entorno virtual.
3. Crea el entorno virtual con el comando:

***sudo python3 -m venv venvGB***

4. Activa el entorno virtual con el comando:

***source venvGB/bin/activate***

Verás que el prompt de la terminal cambia para indicar que estás trabajando dentro del entorno virtual



```
jose@jose-agente:~/Escritorio$ mkdir venv; cd venv
jose@jose-agente:~/Escritorio/venv$ python3 -m venv venvGB
jose@jose-agente:~/Escritorio/venv$ source venvGB/bin/activate
(venvGB) jose@jose-agente:~/Escritorio/venv$
```

*Ilustración 1. Creado y activado entorno virtual "venvGB"*

### 2.2. Instalar los requerimientos

En primer lugar, es necesario tener descargado y descomprimido el directorio con las herramientas necesarias para poder cargar la página web. Antes de arrancar el servidor Flask, es necesario instalar los requerimientos que se encuentran en el archivo *requirements.txt*.

En los pasos, nosotros instalamos los requerimientos en una máquina

Para ello, sigue los siguientes pasos:

1. Nos situamos en el directorio *theGameBazar*
2. Abre una terminal en la ubicación del archivo ***requirements.txt***.
3. Ejecuta el comando ***pip install -r requirements.txt*** para instalar los requerimientos necesarios.

## 2.3. Arrancar el servidor Flask<sup>1</sup>

Para arrancar el servidor Flask, abre una terminal en el directorio del repositorio con el entorno virtual creado en los pasos anteriores activado. A continuación, sigue los siguientes pasos:

1. Configura la variable de entorno `FLASK_APP` para que apunte al archivo de la aplicación Flask (en nuestro caso, `backend.py`) con el siguiente comando:

**`export FLASK_APP=backend`**

2. Ahora, puedes arrancar el servidor con el siguiente comando:

**`flask run --host=0.0.0.0`**

Este comando hará que el servidor escuche en todas las interfaces de red, permitiendo que cualquier host de la red se conecte a él<sup>2</sup>. Si solo quieres permitir que el servidor sea accesible desde el mismo host en el que se está ejecutando, utiliza el siguiente comando en su lugar: **`flask run`**. Cuando arranque el servidor, deberías ver un mensaje similar al siguiente:

```
(venvGB) jose@jose-agente:~/Escritorio/theGameBazar$ flask run --host=0.0.0.0
2023-04-12 13:28:13,338 - INFO - Preparando servidor...
2023-04-12 13:28:13,341 - DEBUG - Starting new HTTPS connection (1): id.twitch.tv:443
2023-04-12 13:28:14,509 - DEBUG - https://id.twitch.tv:443 "POST /oauth2/token?client_id=0c1ev74xgecluf5t8whx1k794x1tn&client_secret=tvlnhx3v1w6kenjb6xummy8l6vbr9u&grant_type=client_credentials HTTP/1.1" 200
93
2023-04-12 13:28:14,513 - DEBUG - Starting new HTTPS connection (1): api.twitch.tv:443
2023-04-12 13:28:15,115 - DEBUG - https://api.twitch.tv:443 "GET /helix/games/top?first=50 HTTP/1.1" 200 1455
2023-04-12 13:28:15,160 - DEBUG - Starting new HTTPS connection (1): www.cheapshark.com:443
2023-04-12 13:28:15,659 - DEBUG - https://www.cheapshark.com:443 "GET /api/1.0/deals?title=Just%20Chatting&exact=1 HTTP/1.1" 200 22
2023-04-12 13:28:15,662 - DEBUG - Starting new HTTPS connection (1): www.cheapshark.com:443
2023-04-12 13:28:16,096 - DEBUG - https://www.cheapshark.com:443 "GET /api/1.0/deals?title=Counter-Strike%20Global%20offensive&exact=1 HTTP/1.1" 200 22
2023-04-12 13:28:16,099 - DEBUG - Starting new HTTPS connection (1): www.cheapshark.com:443
2023-04-12 13:28:16,655 - DEBUG - https://www.cheapshark.com:443 "GET /api/1.0/deals?title=League%20of%20Legends&exact=1 HTTP/1.1" 200 22
2023-04-12 13:28:16,658 - DEBUG - Starting new HTTPS connection (1): www.cheapshark.com:443
2023-04-12 13:28:17,296 - DEBUG - https://www.cheapshark.com:443 "GET /api/1.0/deals?title=Dota%202&exact=1 HTTP/1.1" 200 22
2023-04-12 13:28:17,299 - DEBUG - Starting new HTTPS connection (1): www.cheapshark.com:443
2023-04-12 13:28:17,936 - DEBUG - https://www.cheapshark.com:443 "GET /api/1.0/deals?title=VALORANT&exact=1 HTTP/1.1" 200 22
2023-04-12 13:28:17,941 - DEBUG - Starting new HTTPS connection (1): www.cheapshark.com:443
2023-04-12 13:28:18,388 - DEBUG - https://www.cheapshark.com:443 "GET /api/1.0/deals?title=Grand%20Theft%20Auto%20V&exact=1 HTTP/1.1" 200 22
2023-04-12 13:28:18,391 - DEBUG - Starting new HTTPS connection (1): www.cheapshark.com:443
2023-04-12 13:28:18,896 - DEBUG - https://www.cheapshark.com:443 "GET /api/1.0/deals?title=Apex%20Legends&exact=1 HTTP/1.1" 200 22
2023-04-12 13:28:18,899 - DEBUG - Starting new HTTPS connection (1): www.cheapshark.com:443
2023-04-12 13:28:41,297 - INFO - Archivo de Twitch preparado
2023-04-12 13:28:41,297 - INFO - Base de datos preparada
2023-04-12 13:28:41,297 - INFO - ¡Servidor preparado!
* Serving Flask app 'backend.py'
* Debug mode: off
2023-04-12 13:28:51,383 - INFO - WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://10.0.2.5:5000
2023-04-12 13:28:51,383 - INFO - Press CTRL+C to quit
```

Ilustración 2. Servidor Flask arrancado y escuchando en todas las interfaces de red

3. Para detener el servidor, basta con pulsar `Ctrl+C` en la terminal.

<sup>1</sup> Se puede automatizar esta tarea si desde la terminal ejecutamos **`run_server`**. Es necesario haber seguidos los pasos anteriores para tener nuestro sistema preparado para arrancar el servidor. También es necesario, en caso de que no funcione, darle permisos al ejecutable. Para ello, introducimos en la terminal permisos de ejecución: **`sudo chmod +x run_server`**

<sup>2</sup> Puede ser útil si el servidor lo arrancamos desde una máquina virtual con Linux (o si estamos usando WSL) y queremos ejecutar el servidor en Windows.

## 2.4. Accediendo a la página

Una vez que ya hemos arrancado el servidor, accedemos desde el navegador al servidor introduciendo la ip y el puerto<sup>3</sup>. Si lo vamos a ejecutar en local, deberemos introducir la siguiente dirección: <http://127.0.0.1:5000>

Sin embargo, en nuestro caso, el servidor está corriendo en una máquina virtual con Ubuntu y queremos acceder desde Windows. Para ello, el servidor debe de estar corriendo en todas las interfaces de red, y, a continuación, debemos consultar la IP por la que podemos conectarnos al servidor de la máquina virtual. Podemos consultar esta dirección usando **ifconfig**.

```
jose@jose-agente:~/Escritorio/theGameBazar$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.5 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::f5ac:1ca2:825c:da6d prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:c5:63:d8 txqueuelen 1000 (Ethernet)
    RX packets 14357 bytes 17742204 (17.7 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8938 bytes 1146029 (1.1 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.79.144 netmask 255.255.255.0 broadcast 192.168.79.255
    inet6 fe80::7e97:9511:f60a:d1c3 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:9d:ef:d5 txqueuelen 1000 (Ethernet)
    RX packets 1595 bytes 183627 (183.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 967 bytes 94464 (94.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Bucle local)
    RX packets 788 bytes 84226 (84.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 788 bytes 84226 (84.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

*Ilustración 3. Consultado IP máquina virtual donde se está ejecutando Flask*

Como podemos ver en la imagen, nuestra IP de acceso en Ubuntu es 192.168.79.144. Por tanto, accedemos desde el navegador en Windows con la siguiente URL: <http://192.168.79.144:5000/>

---

<sup>3</sup> En Flask, el puerto por defecto en el que se ejecuta el servidor web es el 5000. Si deseas cambiar este valor, puedes hacerlo al iniciar el servidor con el parámetro `--port`, seguido del número del puerto deseado.

### 3. CARACTERÍSTICAS DE LA PÁGINA

Una vez, que hemos accedido a la página, podemos encontrar la página de bienvenida. En ella, podemos encontrar el logo de The Game Bazar, y sus 3 partes fundamentales: el buscador de juegos, el Top 10 de juegos y un inicio de sesión para administradores.

Como se puede ver, la página de inicio consta con un carrusel de imágenes de juegos populares en la actualidad.

Además, es preciso comentar que la página utiliza cookies (como el de sesión), por lo que nos aparece una alerta informativa en la parte inferior de la página. Para quitar la alerta, deberemos pulsar en el botón **Aceptar**.

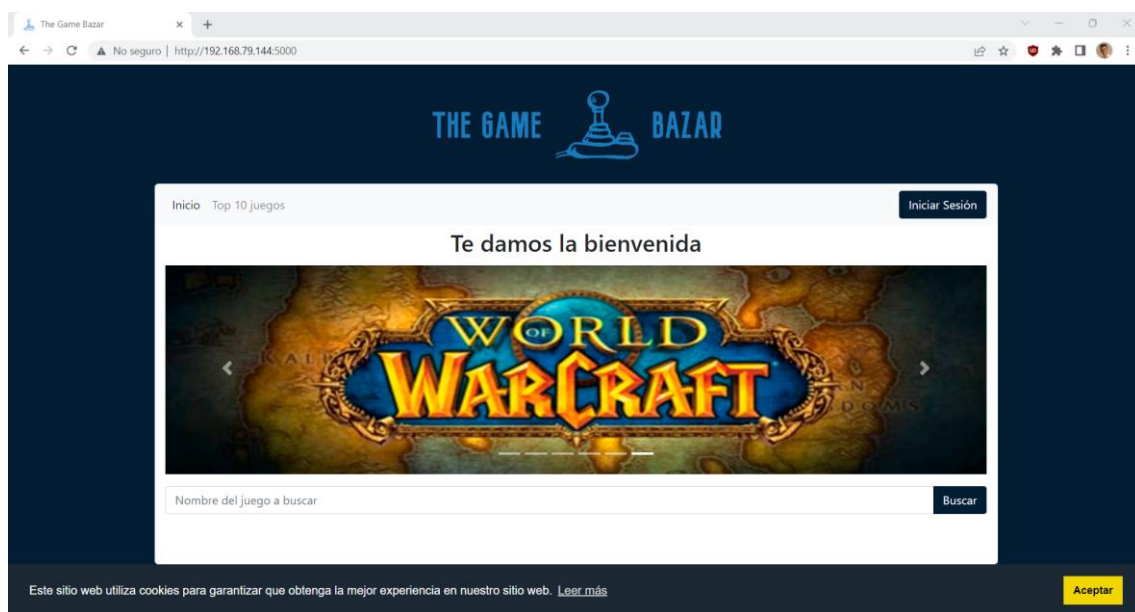


Ilustración 4. Página de bienvenida

### 3.1. Buscador de juegos

En el primer sitio de la página, se encuentra el buscador de juegos. Aquí puedes buscar el nombre de un juego y la página te mostrará las mejores ofertas que ha encontrado de ese juego, junto con la tienda a la que pertenece la oferta. Si haces clic en la tienda, te redirigirá a la oferta.

En la siguiente imagen, podemos ver qué ofertas aparece al buscar “Outlast”. Podemos comprobar que aparecen los siguientes campos: título del juego, tienda donde lo podemos comprar (pinchando en ella te redirige a la misma), precio del juego y descuento aplicado.

Outlast

Buscar

Título	Tienda / Enlace	Precio ▼	Descuento ▼
Outlast	<a href="#">GOG</a>	2.99€	85%
Outlast 2	<a href="#">GOG</a>	4.49€	85%

*Ilustración 5. Buscando juego "Outlast"*

Para realizar la búsqueda, introducimos el nombre en la caja de texto y pulsamos Enter o en el botón Buscar para realizar la búsqueda.

Además, podemos ordenar los resultados de manera ascendente o descendente por precio o por descuento. Para ello, debemos pulsar sobre la flecha que encontramos a la derecha de Precio o de Descuento.

### 3.2. Top 10 Juegos

La sección de Top 10 juegos muestra una lista de los juegos más populares en Twitch, basándose en el número de espectadores que tienen en la plataforma. La lista está limitada a los juegos que se encuentran en el buscador de juegos del sistema. En esta lista se muestran el título del juego y una imagen representativa del mismo.

Es importante mencionar que, aunque hay juegos muy populares en Twitch, no todos ellos están disponibles en el buscador de juegos del sistema, ya sea porque no son gratuitos o porque no están dentro del catálogo de juegos disponibles en la plataforma. Por esta razón, la lista se limita a los juegos que podemos encontrar en el buscador del sistema. De esta forma, nos aseguramos de que la lista sea representativa de los juegos más populares que los usuarios del sistema pueden disfrutar.

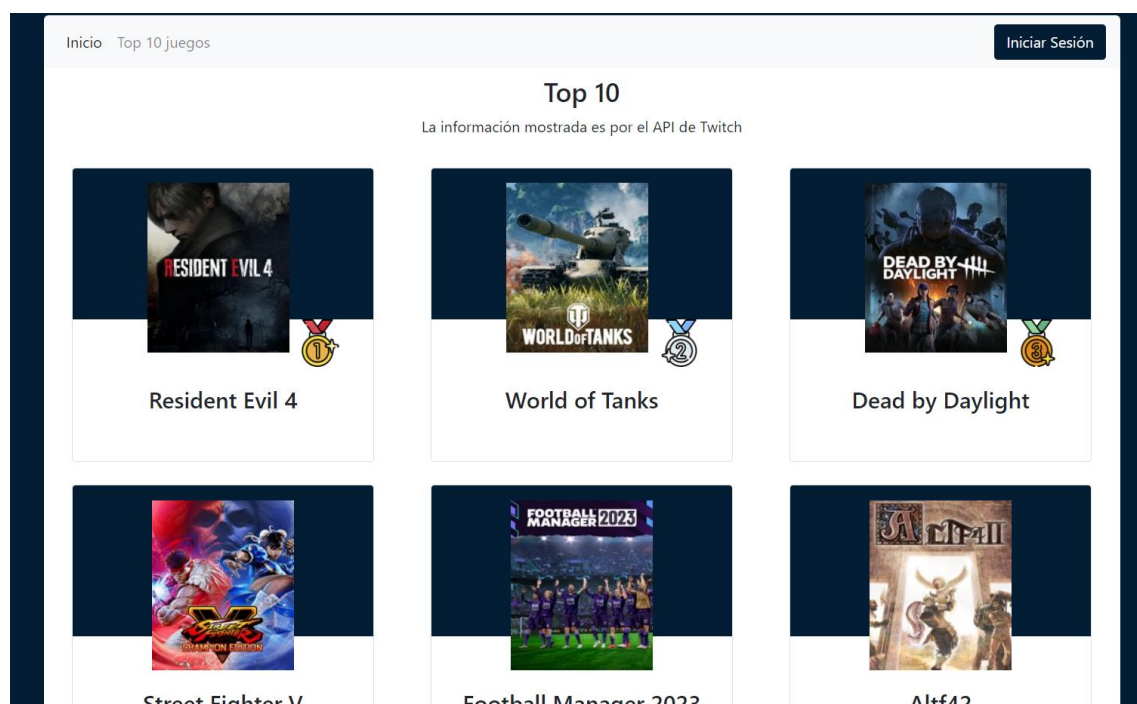


Ilustración 6. Sección Top 10 Juegos más populares

Como podemos ver, los 3 primeros juegos más populares están marcados con el símbolo de una medalla.



### 3.3. Inicio de sesión

El inicio de sesión es una funcionalidad importante en el sitio web de Game Bazar que permite a los administradores acceder al área de administración. El botón para iniciar sesión se encuentra tanto en la página de inicio (la del buscador de juegos) como en la página Top 10 Juegos. Cuando se pulsa en él, nos lleva al siguiente formulario:

Ilustración 7. Formulario de inicio de sesión. La imagen muestra una interfaz web con una barra superior que contiene los enlaces "Inicio" y "Top 10 juegos" a la izquierda, y un botón "Iniciar Sesión" a la derecha. El formulario principal está centrado y contiene dos campos de entrada: "\*Usuario:" y "\*Contraseña:". Debajo de estos campos hay un botón "Iniciar".

Ilustración 7. Formulario de inicio de sesión

Si el usuario se identifica con unos credenciales válidos, el sistema te redirige al área de administración donde podrá consultar diferentes estadísticas. Si en caso contrario, el usuario o la contraseña no son válidos, la página mostrará un mensaje de error y dejará intentarlo de nuevo.

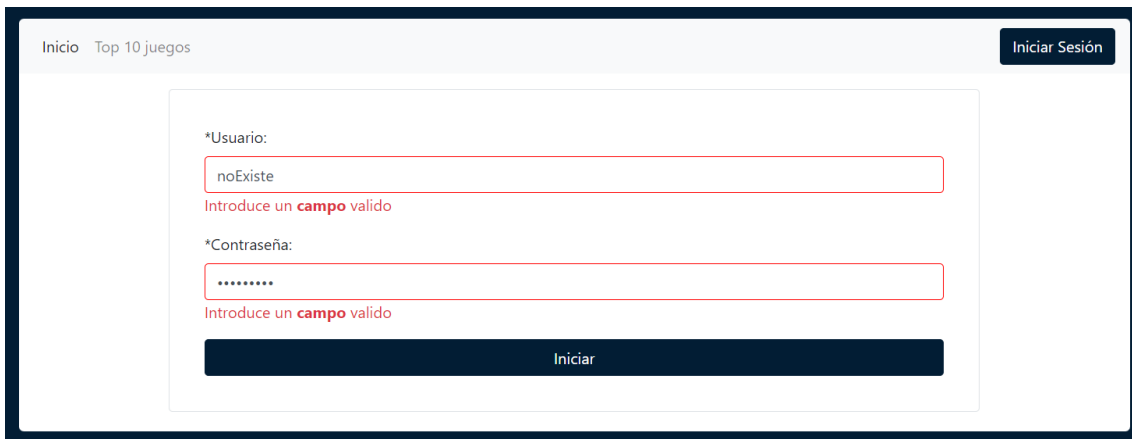
Ilustración 8. Inicio de sesión incorrecto. La imagen muestra la misma interfaz web que la Ilustración 7, pero con mensajes de error. El campo de usuario muestra el texto "noExiste" y el campo de contraseña muestra "\*\*\*\*\*". Debajo de cada campo hay un mensaje de error: "Introduce un campo valido". El botón "Iniciar" sigue presente.

Ilustración 8. Inicio de sesión incorrecto

Cuando el login es correcto<sup>4</sup>, la barra superior cambia y se muestran los botones de Administración y Cerrar Sesión, como se muestra en la siguiente imagen:

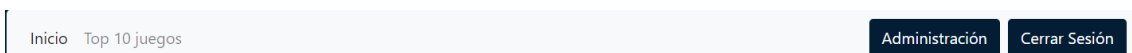
Ilustración 9. Barra superior una vez iniciado sesión. La imagen muestra la barra superior de la interfaz web después de iniciar sesión correctamente. Los enlaces "Inicio" y "Top 10 juegos" siguen presentes a la izquierda, pero el botón "Iniciar Sesión" ha sido reemplazado por dos botones: "Administración" y "Cerrar Sesión".

Ilustración 9. Barra superior una vez iniciado sesión.

<sup>4</sup> Para probar el área de administración, podemos acceder con las siguientes credenciales:  
**Usuario: Jose**  
**Contraseña: TGB123**

### 3.4. Área de Administración

El área de administración es una sección reservada para los usuarios con permisos de administrador. Desde aquí, se pueden consultar las estadísticas de los clics realizados en las tiendas y los clics totales por año. Además, se puede añadir nuevos administradores siempre y cuando no existan ya en la base de datos.

Para acceder al área de administración, es necesario haber iniciado sesión con una cuenta de administrador. Una vez dentro, se mostrará una pantalla principal con las diferentes opciones disponibles.

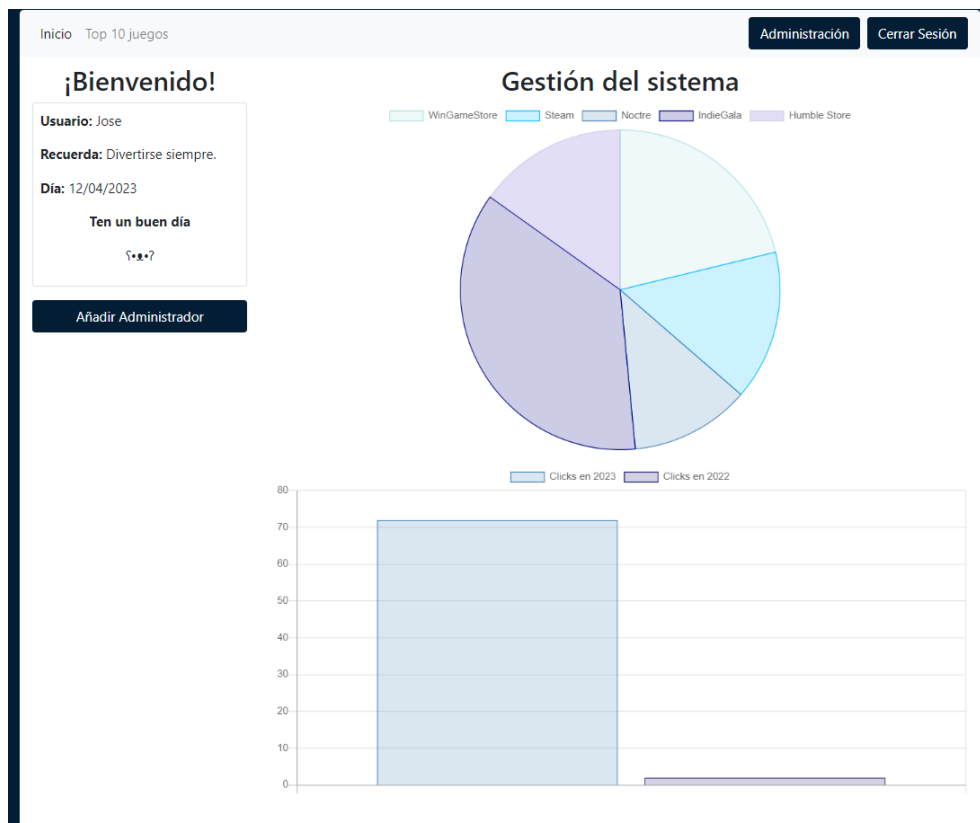


Ilustración 10. Página área de administración

Como se puede ver en la imagen anterior, tenemos una gráfica circular que muestra estadísticas de las tiendas sobre las que se ha hecho clic para consultar una oferta en la sección Buscador de juegos. Además, si pasamos el ratón por encima, podremos ver el número de clics totales, tal y como se ve en la siguiente imagen:

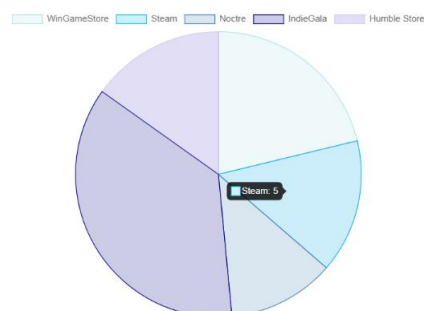


Ilustración 11. Número de clics en la tienda Steam

La siguiente gráfica nos muestra el número de clics registrados por año. Al igual que en la anterior, si pasamos el ratón por encima, se mostrarán los clics totales.

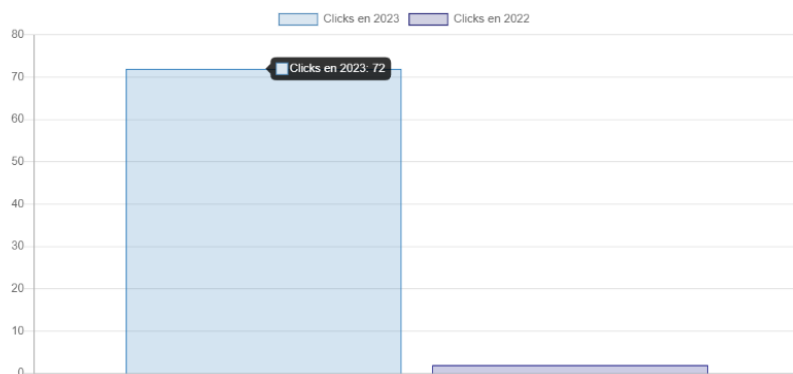


Ilustración 12. Gráfica de barras con el número de clics por año

Por último, podemos añadir un nuevo administrador pulsando sobre el botón Añadir Administrador. Una vez pulsado, se nos abrirá el siguiente formulario emergente:

**Añadir nuevo administrador** [X]

\*Usuario:

\*Contraseña:

**Registrar** **Cancelar**

Ilustración 13. Formulario para añadir nuevo administrador

Si introducimos un usuario existente, veremos un mensaje de error como el que aparece en la imagen siguiente. En este caso, solo coincide el usuario. No se especifica por temas de seguridad y privacidad.

**Añadir nuevo administrador** [X]

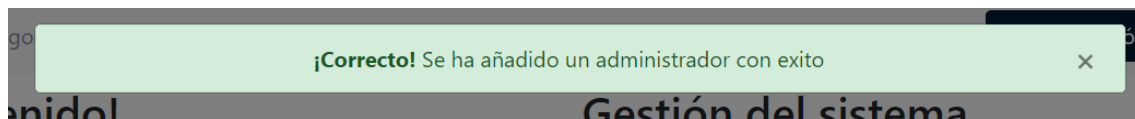
\*Usuario:  
  
Introduce un **campo** valido

\*Contraseña:  
  
Introduce un **campo** valido

**Registrar** **Cancelar**

Ilustración 14. Error al añadir nuevo administrador

Sin embargo, si se puede añadir el administrador, nos aparecerá un mensaje indicándolo.



*Ilustración 15. Mensaje confirmando que se ha añadido un administrador*

## 4. TESTING AUTOMÁTICO

Para garantizar la calidad y el correcto funcionamiento de nuestro backend, hemos implementado pruebas automáticas utilizando un script de Bash llamado *run\_tests*. Este script ejecuta un conjunto de pruebas unitarias y de integración que cubren las funcionalidades clave de nuestro servidor Flask y la conexión con las APIs. Para ejecutar las pruebas del backend, sigue los siguientes pasos:

1. Navega hasta el directorio del backend desde la terminal o línea de comandos.
2. Asegúrate de que todas las dependencias del proyecto estén instaladas (**usando *pip install -r requirements.txt***).
3. Si aún no lo has hecho, otorga permisos de ejecución al archivo *run\_tests* utilizando el siguiente comando: ***chmod +x run\_tests***.
4. Ejecuta el script *run\_tests* utilizando el siguiente comando: ***./run\_tests***.

Una vez ejecutado, podremos obtener resultados como lo que se pueden ver en la siguiente imagen:

```
(venvGB) jose@jose-agente:~/Escritorio/isi-lab/isi-TheGameBazar-Sprint-3$ ./run_tests
===== test session starts =====
platform linux -- Python 3.8.10, pytest-7.3.0, pluggy-1.0.0
rootdir: /home/jose/Escritorio/isi-lab/isi-TheGameBazar-Sprint-3/tests
configfile: pytest.ini
plugins: cov-4.0.0
collecting ...
----- live log collection -----
2023-04-22 12:57:36 [INFO] Preparando servidor... (backend.py:329)
2023-04-22 12:57:55 [INFO] Archivo de Twitch preparado (backend.py:331)
2023-04-22 12:57:55 [INFO] Base de datos preparada (backend.py:333)
2023-04-22 12:57:55 [INFO] ¡Servidor preparado! (backend.py:334)
collected 8 items

tests/test_app.py::test_obtener_juego
----- live log call -----
2023-04-22 12:57:55 [INFO] Iniciando test para obtener un juego (test_app.py:12)
PASSED [ 12%]
tests/test_app.py::test_juego_no_existente
----- live log call -----
2023-04-22 12:57:57 [INFO] Iniciando test para obtener un juego que no existe (test_app.py:18)
PASSED [ 25%]
tests/test_app.py::test_inicio_sesion_incorrecto
----- live log call -----
2023-04-22 12:57:57 [INFO] Iniciando test para un inicio de sesión incorrecto (test_app.py:24)
PASSED [ 37%]
tests/test_app.py::test_inicio_sesion_correcto
----- live log call -----
2023-04-22 12:57:57 [INFO] Iniciando test para un inicio de sesión correcto (test_app.py:30)
PASSED [ 50%]
```

*Ilustración 16. Ejecutando script para el testing*

## 5. AUTOMATIZACIÓN DEL DESPLIEGUE

### 5.1. Docker

Para facilitar y automatizar el despliegue de nuestra aplicación, hemos creado un Dockerfile que permite construir una imagen Docker de la misma. Esta imagen incluye todas las dependencias necesarias y permite ejecutar la aplicación en cualquier entorno que tenga Docker instalado. Para construir y desplegar la aplicación utilizando Docker, sigue los siguientes pasos:

1. Asegúrate de que Docker esté instalado en tu sistema.
2. Abre la terminal o línea de comandos y navega hasta el directorio del proyecto donde se encuentra el Dockerfile.
3. Construye la imagen Docker de la aplicación utilizando el siguiente comando:  
**`docker build -f Dockerfile -t the-game-bazar:latest .`**

Este comando indica a Docker que utilice el archivo Dockerfile especificado (**-f Dockerfile**) y que asigne un nombre y una etiqueta a la imagen construida (**-t the-game-bazar:latest**).

4. Una vez que la imagen se haya construido con éxito, puedes ejecutar un contenedor a partir de la imagen utilizando el siguiente comando: **`docker run -p 5000:5000 the-game-bazar:latest`**.

Este comando crea un contenedor a partir de la imagen **the-game-bazar:latest** y asigna el puerto 5000 del contenedor al puerto 5000 del host (**-p 5000:5000**).

5. Ahora, la aplicación debería estar ejecutándose en un contenedor Docker. Puedes acceder a ella a través de un navegador web navegando a **`http://localhost:5000`**.

Al seguir estos pasos, podrás automatizar el despliegue de la aplicación utilizando Docker. Esto facilita el proceso de implementación y garantiza que la aplicación se ejecute en un entorno aislado y controlado.

## 5.2. Kubernetes

Para facilitar y automatizar aún más el despliegue de nuestra aplicación, hemos creado un archivo de configuración de Kubernetes (**.yml**) que permite desplegar la aplicación en un clúster de Kubernetes usando Minikube. Este enfoque ofrece ventajas adicionales, como la escalabilidad y la gestión eficiente de recursos. Para desplegar la aplicación utilizando Kubernetes y Minikube, sigue los siguientes pasos:

1. Inicia Minikube utilizando el siguiente comando: ***minikube start***
2. Carga la imagen Docker the-game-bazar:latest en el registro de contenedores de Minikube utilizando el siguiente comando: ***minikube image load the-game-bazar:latest***
3. Ejecuta el siguiente comando para abrir el panel de control de Minikube: ***minikube dashboard***