**REPORTE DE DESARROLLO DE SOFTWARE**

**Versión 1.0**

**Lima, abril 2023.**

**Historial de revisiones**

| **Fecha** | **Revisión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 30/04/2023 | 1.0 | Primera versión del reporte de desarrollo del software | Consultora HomeSkill S.A. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**ÍNDICE**

[**1. Introducción. 4**](#_q90lk01sy209)

[**1.1. Propósito del informe. 4**](#_9dw81asss0qp)

[**1.2. Alcance del informe. 4**](#_ibbmtx9bvnt5)

[**1.3. Definiciones, acrónimos y abreviaturas. 4**](#_45jr7rzh5bs2)

[**2. Descripción general del proyecto. 4**](#_e2innfqo5k7a)

[**2.1. Objetivos del proyecto. 4**](#_jnsn05tyqva3)

[**2.2. Alcance del proyecto. 4**](#_5b5btt471l1t)

[**2.3. Estructura del equipo y roles. 5**](#_5oajyx4s86h)

[**3. Trabajo realizado durante la primera iteración. 5**](#_d0mljzqatqeh)

[**3.1. Objetivos de la primera iteración. 5**](#_wp5avqioabcf)

[**3.2. Actividades realizadas durante la primera iteración. 5**](#_zgui8fmqs8a0)

[**3.2.1. Análisis de requisitos. 5**](#_4q4sgzly8a3n)

[**3.2.2. Diseño de arquitectura del software. 7**](#_g6i1sllju9n2)

[**3.2.3. Implementación de componentes clave. 8**](#_f6byo7sqo2hg)

[**4. Resultados de la primera iteración. 9**](#_hgu3hurx9jj)

[**4.1. Requisitos cumplidos durante la primera iteración. 9**](#_lxwk656y3i2z)

[**4.2. Estructura de directorios. 9**](#_x6ay3iltq458)

[**4.2.1. Pruebas unitarias realizadas con éxito. 15**](#_78sh5di98e44)

[**4.3. Evaluación de los resultados de la primera iteración. 15**](#_dhjdlrhv5wv)

[**4.3.1. Cumplimiento de los objetivos de la primera iteración. 15**](#_nfqhwxe5p6wi)

[**4.3.2. Rendimiento del equipo. 15**](#_ss6exe12rgkt)

[**5. Conclusiones. 16**](#_qtxdic48hnm)

[**6. Referencias externas. 16**](#_pvhvgotdf33j)

[**7. Referencias de documentación del proyecto. 16**](#_b3abvah16m93)

# Introducción.

# Propósito del informe.

El propósito principal de este informe es brindar una descripción detallada de nuestro software del proyecto GoShop. Todo esto con el fin de que los usuarios o desarrolladores puedan entender cómo funciona el programa.

# Alcance del informe.

Este documento proporciona información detallada del software incluyendo sus características y funcionalidades expuestas en el plan de iteración, una lista con los problemas y riesgos encontrados, una evaluación de los resultados de la primera iteración y las conclusiones sobre el proyecto.

# Definiciones, acrónimos y abreviaturas.

* **Diseño UI:** El diseño de interfaz de usuario se refiere al proceso de definir diversos aspectos que influyen en la apariencia de las interfaces de usuario, como la forma, la función, la utilidad, la ergonomía y la imagen de marca.
* **Casos de uso:** Un caso de uso se refiere a una descripción de una acción o actividad específica. Por otro lado, un diagrama de caso de uso describe las actividades que deben llevarse a cabo para completar un proceso determinado.
* **Base de datos:**  Una base de datos es una colección ordenada de información o datos estructurados que suelen almacenarse en un sistema informático en formato electrónico.
* **Django:** Es un framework desarrollado en Python y que se utiliza para la creación de páginas web.
* **GitHub:** Es una forja para alojar proyectos utilizando el sistema de control de versiones Git.
* **GS:** Acrónimo de GoShop.

# Descripción general del proyecto.

# Objetivos del proyecto.

El principal objetivo del proyecto es contar con los 3 requerimientos expuestos en el primer plan de iteración, donde cada funcionalidad deberá estar bien implementada y documentada, con sus respectivas pruebas unitarias.

# Alcance del proyecto.

El alcance del proyecto es desarrollar una aplicación web que permita a los usuarios registrarse y hacer login en la plataforma. La aplicación debe tener una página principal que muestre información relevante para el usuario, así como una vista de productos organizados por categorías. De las funcionalidades específicas incluyen:

* Registro y Login de usuario
* Vista de la página principal
* Vista de productos por categoría

El alcance del proyecto se limita a las funcionalidades mencionadas anteriormente y no incluirá otras características adicionales que no estén expresamente mencionadas en el plan de iteración. Además, presentar al equipo de desarrollo y a los clientes, la documentación del proyecto desarrollado hasta la fecha.

# Estructura del equipo y roles.

El equipo de desarrollo está conformado por 7 integrantes, los cuales se les asignó los siguientes roles:

* Ipanaque Pazo, Jorge Paul: **Jefe de Proyecto (JP), Arquitecto de software(AS)**
* Cjumo Chumbes, Jose Carlos : **Analista QA, Desarrollador Back-End (DB)**
* Ramírez Alvarado, Piero Jaime: **Designer UX / Analista Base de datos** (ABD)
* Castillo Bernal, Carlos Alberto: **Analista Junior (AJ) / Tester (T)**
* Ames Camayo, Daniel: **Desarrollador Front-End (DF) / Analista de Base de datos (ABD)**
* Saenz Chang, Jesus Angel: **Desarrollador Front-End (DF) / Tester (T)**

Hinostroza Quispe, Gianlucas Amed: **Desarrollador Back-End (DB)**

# Trabajo realizado durante la primera iteración.

# Objetivos de la primera iteración.

* Cumplir los requisitos establecidos:
  + Registro y Login de usuario.
  + Vista de la página principal.
  + Vista de productos por categoría.
* Cumplir la documentación establecida en el cronograma de proyecto:
  + Plan de Proyecto.
  + Plan de Fase (Inicio).
  + Plan de Iteración.
  + Documento de Arquitectura del software.
  + Documento de Especificación de UI.
  + Documento de Guía de Estilos.
  + Documento de Especificación de la Base de Datos.
  + Documentos de Especificación de Requisitos.
  + Documentos de Casos de Uso.
  + Documento de Informe de Pruebas.

# Actividades realizadas durante la primera iteración.

# Análisis de requisitos.

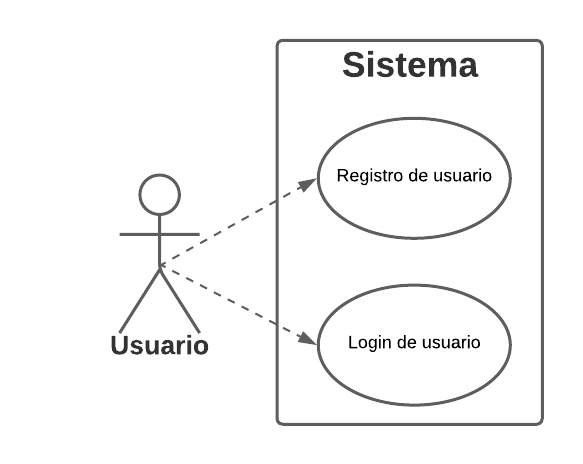
* + - 1. **Requisito 1: Registro y Login de usuario.**
* **Precondiciones.**

El usuario no debe estar con sesión activa, es decir, no debe haber iniciado sesión.

* **Poscondiciones.**

El usuario tendrá acceso a las funcionalidades del usuario.

* **Diagrama de casos de uso.**

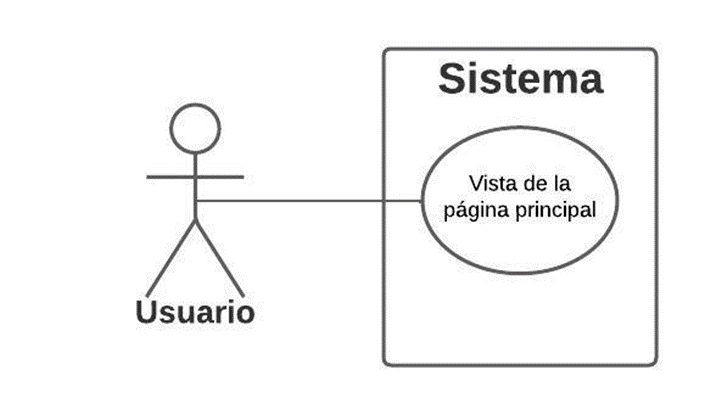


* + - 1. **Requisito 2: Vista de la página principal.**
* **Precondiciones.**

El sitio web debe funcionar y está a disposición de los usuarios, que la base de datos del sitio web cuente con un catálogo disponible y actualizado de bienes o servicios, que se haya desarrollado el diseño del sitio web,además los dispositivos conectados, Internet y los navegadores web deben estar a disposición de los usuarios.

* **Poscondiciones.**

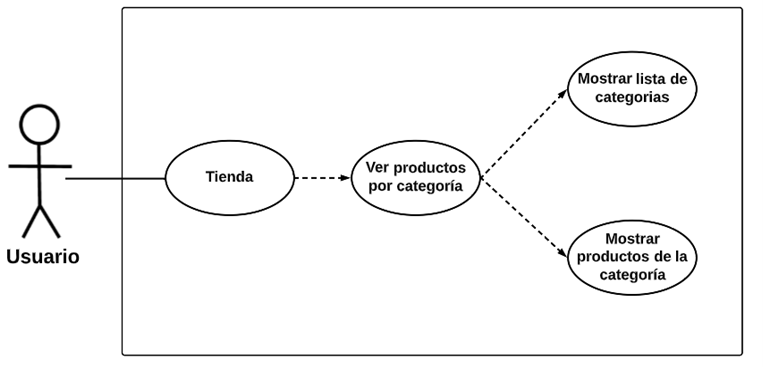
Si el usuario logra visualizar la página principal, la vista de la página de inicio debe actualizarse regularmente para mostrar nuevos productos o promociones y el diseño de la página debe actualizarse para mejorar la experiencia del usuario. En general, la funcionalidad de la página de inicio debe garantizar que los usuarios tengan una experiencia satisfactoria en el sitio y alentar el retorno futuro.

* **Diagrama de casos de uso.**
  + - 1. **Requisito 3: Vista de productos por categoría.**
* **Precondiciones.**

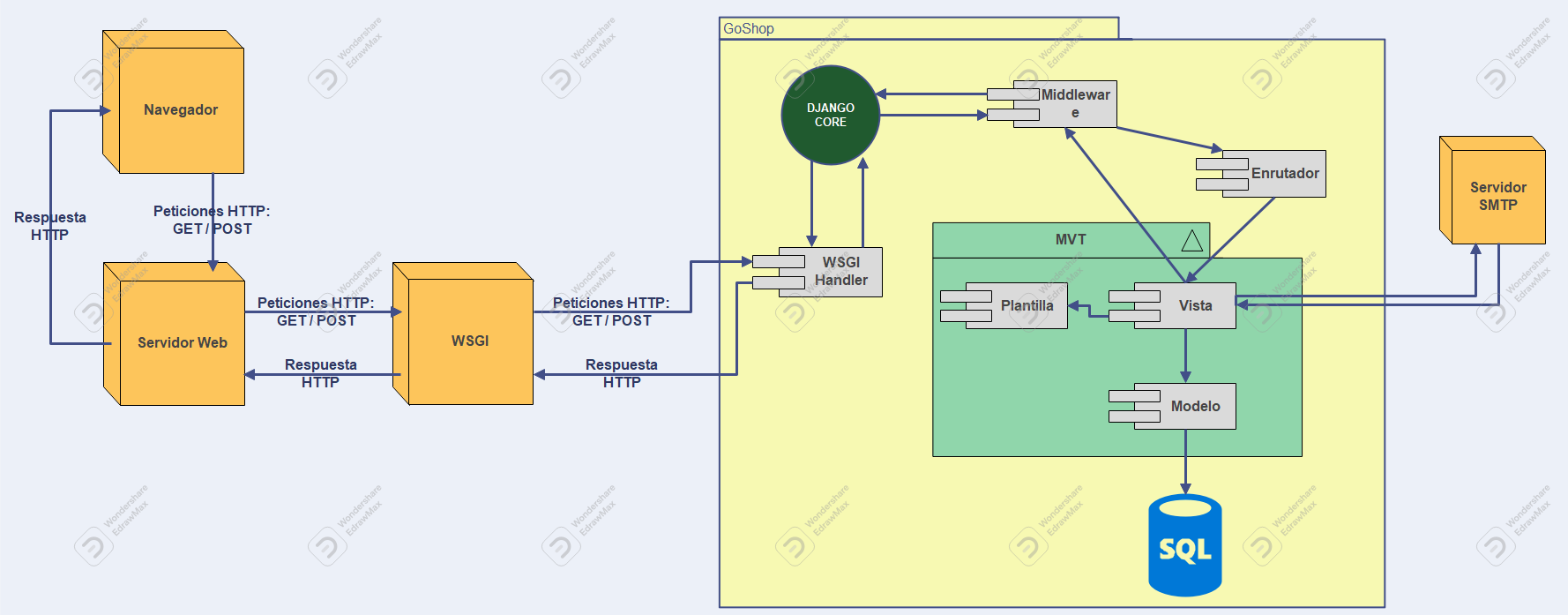
El usuario debe ingresar en la tienda en línea y tener acceso a internet mediante un dispositivo. Además, la tienda en línea debe tener los productos registrados y organizados en categorías.

* **Poscondiciones.**

El usuario podrá visualizar correctamente los productos por categoría.

* **Diagrama de casos de uso.**

# Diseño de arquitectura del software.



# Implementación de componentes clave.

Se implementaron los siguientes componentes en la primera iteración:

* **Configuración del entorno de desarrollo.** Se establecieron las configuraciones necesarias para el entorno de desarrollo, tales como la instalación del framework Django y la configuración de la base de datos.
* **Estructura de directorios.** La estructura de directorios ha sido definida gracias a Django, ya que este nos ofrece una estructura que sigue el patrón de diseño MVT, lo cual favorece a la escalabilidad y a la flexibilidad del código.
* **Diseño de la base de datos.** La base de datos ha sido diseñada con las tablas que ofrece Django por defecto y 2 tablas adicionales que han sido creadas para cumplir con los requerimientos establecidos.
* **Implementación de las vistas.** Se implementaron las vistas, las cuales son las responsables de conectar con la base de datos y mostrar los datos al usuario por medio de las plantillas, además de procesar las solicitudes HTTP y generar las respuestas correspondientes. Por lo tanto, las vistas son fundamentales para el correcto funcionamiento del software.
* **Implementación de las plantillas.** Se implementaron las plantillas HTML, las cuales representan el contenido que se mostrará al usuario.
* **Pruebas unitarias.** Se realizaron las respectivas pruebas unitarias a las vistas para comprobar el correcto funcionamiento de las vistas.
* **Documentación.** Se realizó la respectiva documentación para la presente iteración. Se incluyeron los siguientes documentos:
  + Plan de Proyecto.
  + Cronograma del Proyecto.
  + Plan de Fase (Inicio).
  + Plan de Iteración.
  + Documento de Arquitectura del software.
  + Documento de Especificación de UI.
  + Documento de Guía de Estilos.
  + Documento de Especificación de la Base de Datos.
  + Documentos de Especificación de Requisitos.
  + Documentos de Casos de Uso.
  + Documento de Informe de Pruebas.

Los documentos anteriores han sido realizados en la presente iteración, además del presente documento que es el último de esta iteración.

# Resultados de la primera iteración.

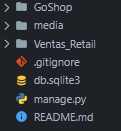
# Requisitos cumplidos durante la primera iteración.

En esta iteración se cumplieron los requisitos establecidos:

* Requisito 1: Registro y Login de usuario.
* Requisito 2: Vista de la página principal.
* Requisito 3: Vista de productos por categoría.

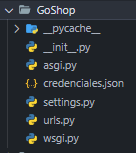
# Estructura de directorios.

Se tiene la siguiente estructura de directorios para el desarrollo del proyecto:



En la raíz del proyecto, se tienen las carpetas y archivos mostrados en la imagen anterior. A continuación se detallarán estas carpetas y archivos:

* **GoShop.**

****

En la carpeta del proyecto, la carpeta GoShop, se encuentran algunos archivos y carpetas importantes para el funcionamiento del proyecto. A continuación, se detallan estos archivos y carpetas:

* + **\_\_pycache\_\_**

El directorio \_\_pycache\_\_ puede contener archivos de caché compilados para módulos que se utilizan internamente en la aplicación, tales como vistas, modelos, entre otros.

* + **\_\_init\_\_.py**

El archivo \_\_init\_\_.py es un archivo especial en Python que indica que un directorio debe ser considerado como un paquete de Python. En el contexto de Django, se utiliza principalmente para configurar la aplicación, importar módulos y definir variables globales.

* + **asgi.py**

Es utilizado para configurar el servidor de aplicaciones que soporta la especificación ASGI (Asynchronous Server Gateway Interface). ASGI es un protocolo que permite a Django manejar conexiones asíncronas, lo que es particularmente útil para aplicaciones web en tiempo real y aplicaciones que necesitan procesamiento en tiempo real de solicitudes de clientes.

* + **credenciales.json**

Es un archivo que contiene los datos del correo electrónico que se ha utilizado en este proyecto para mandar correos a los usuarios.

* + **settings.py**

Es un archivo muy importante para la configuración del proyecto de Django, se pueden configurar las aplicaciones de Django, la base de datos utilizada, configuración de correo, idioma, zona horaria, entre otros.

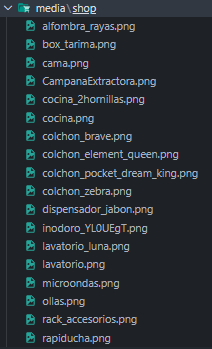
* + **urls.py**

Es un archivo que se utiliza para definir las rutas o URLs que se utilizarán en el proyecto. En este archivo se definen qué vistas son las que se mostrarán al usuario.

* + **wsgi.py**

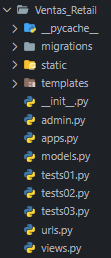
Es el archivo responsable de crear una instancia de la aplicación Django y exponerla a un servidor WSGI. El archivo se genera automáticamente cuando se crea un nuevo proyecto de Django y su función principal es configurar el entorno de la aplicación, incluyendo la carga de configuraciones desde settings.py para que pueda ser ejecutada por un servidor web.

* **media.**



En esta carpeta, se tiene otra llamada “shop”, la cual contiene todas las imágenes de los productos que se encuentran en la base de datos, estas imágenes se agregan automáticamente a esta carpeta cuando se crea un nuevo producto en la base de datos.

* **Ventas\_Retail.**

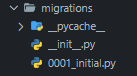


La carpeta Ventas\_Retail es la carpeta que contiene todo lo relacionado a la aplicación Django, que es diferente al proyecto Django, ya que el proyecto Django puede contener una o más aplicaciones. A continuación, se describen las carpetas y archivos de la aplicación:

* + **\_\_py\_\_cache**

Como ya se mencionó, el directorio \_\_pycache\_\_ puede contener archivos de caché compilados para módulos que se utilizan internamente en la aplicación, tales como vistas, modelos, entre otros.

* + **migrations**

****

En esta carpeta se encuentra la carpeta \_\_py\_\_cache\_\_, el archivo \_\_init\_\_.py y el archivo 0001\_initial.py.

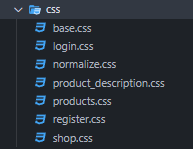
Los dos primeros ya se explicaron anteriormente. En cuanto al archivo 0001\_initial.py, el cual es generado por Django y contiene el código para la creación de las tablas. En este caso, este archivo contiene el código para crear las tablas ProductCategory y Product, las cuales se crearon mediante una migración.

* + **static**



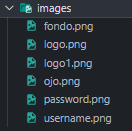
La carpeta *static*, como indica su nombre, contiene archivos estáticos que son servidos directamente por el servidor web, como lo son los archivos css, las imágenes y los archivos javascript. En este caso tenemos esas 3 carpetas para sus respectivos archivos.

* + - **css**

****

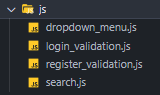
En esta carpeta se encuentran los archivos css, los cuales sirven para dar estilo a las plantillas html que se encuentran en otra carpeta llamada *templates*.

* + - **images**

****

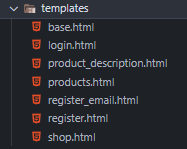
En esta carpeta se encuentran las imágenes que se utilizan en el sitio web, sin contar a las de la base de datos, que van en otra carpeta.

* + - **js**

****

En esta carpeta se encuentran los archivos javascript, los cuales sirven para dar dinamismo al sitio web gracias al manejo de eventos y el control de las interacciones del usuario con el sitio web en tiempo real.

* + **templates**



En esta carpeta se encuentran las plantillas HTML, las cuales sirven para mostrar definir la estructura de la página web que verá el usuario al visitar el sitio web. Además, al usar Django, se puede utilizar la herencia de plantillas para favorecer la reutilización de código, además de cargar los contenidos dinámicamente gracias a la conexión con las vistas. Representa la parte “Templates” del patrón de diseño MVT.

* + **\_\_init\_\_.py**

Como se mencionó anteriormente, el archivo \_\_init\_\_.py es un archivo especial en Python que indica que un directorio debe ser considerado como un paquete de Python.

* + **admin.py**

Se utiliza para definir la configuración de la interfaz de administración de la aplicación web. Permite que los administradores del sitio puedan gestionar los datos de la aplicación a través de una interfaz gráfica de usuario. En este archivo se registran los modelos de la base de datos para crear las tablas.

* + **apps.py**

Es un archivo de configuración utilizado en Django para definir la configuración de una aplicación específica. En este caso, se ha dejado el archivo como viene por defecto.

* + **models.py**

Este archivo sirve para crear las tablas en la base de datos mediante la creación de clases en este archivo. Estas clases deben heredar de la clase Model de Django, además, también se puede modificar el nombre de la tabla que deseemos y el nombre de la tabla cuando aparezca en la interfaz del administrador. Representa la parte de “Model” del patrón de diseño MVT.

* + **tests01.py**

Es un archivo que contiene las pruebas unitarias del primer requisito: Login y Registro.

* + **tests02.py**

Es un archivo que contiene las pruebas unitarias del primer requisito: Vista de la página principal.

* + **tests03.py**

Es un archivo que contiene las pruebas unitarias del primer requisito: Vista de productos por categoría.

* + **urls.py**

Es un archivo que se utiliza para definir las rutas o URLs que se utilizarán en la aplicación. No debe confundirse con el archivo *urls.py* del proyecto, ya que este solo controla las URLs de la aplicación.

* + **views.py**

Es el archivo que define la lógica de negocio de las vistas. Es el archivo que sirve de conexión entre las plantillas y los modelos de la base de datos. Las funciones que se encuentran en este archivo procesan una petición HTTP y devuelven una respuesta, que por lo general son plantillas HTML. Representa la parte “View” del patrón de diseño MVT.

* **.gitignore.**

Este archivo se utiliza para especificar los directorios y los archivos que se van a ignorar a la hora de subir los cambios al repositorio remoto. Hay muchos casos en los que no se requiere que se suban ciertos archivos porque son irrelevantes para el correcto funcionamiento del software, por lo tanto, lo mejor es ignorarlos. Otro caso para ignorar un archivo o carpeta es cuando puede contener datos que no se quieren mostrar a los otros desarrolladores, pueden ser datos confidenciales, por ejemplo.

* **db.sqlite3.**

Este es el archivo de la base de datos del sistema. Es una base de datos SQLite3, ya que esta viene con Django y es una base de datos ligera y apta para el desarrollo del software. Cuando el software pase a producción, se cambiará esta base de datos por una más grande como puede ser PostgreSQL.

* **manage.py.**

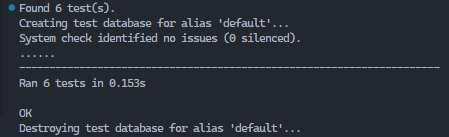
El archivo manage.py es una herramienta de línea de comandos proporcionada por Django para interactuar con el proyecto. Se utiliza para realizar varias tareas relacionadas con el proyecto de Django, como crear aplicaciones, realizar migraciones de bases de datos, ejecutar el servidor web de desarrollo, ejecutar pruebas, etc.

* **README.md**

Es un archivo de texto plano que sirve para proporcionar información general sobre el proyecto y puede servir como documentación. En esta iteración aún no se ha realizado el README.md, sin embargo, se estará realizando en las próximas iteraciones.

# Pruebas unitarias realizadas con éxito.

Las pruebas realizadas para los 3 requisitos se realizaron con éxito, tal y como se aprecia en la siguiente imagen:



Se realizaron 6 pruebas unitarias para un total de 3 requisitos, ya que un requisito no necesariamente implica una función en el archivo *views.py.*

# Evaluación de los resultados de la primera iteración.

# Cumplimiento de los objetivos de la primera iteración.

Los objetivos establecidos para la primera iteración se cumplieron con éxito, es decir, se logró la correcta codificación de los requisitos y la realización de la documentación para esta iteración.

# Rendimiento del equipo.

El equipo de desarrollo y los demás miembros del equipo de proyecto han realizado un correcto desempeño durante el desarrollo de esta primera iteración. Esto se ha visto reflejado en el cumplimiento de la codificación y la documentación del proyecto. Además, se muestra una gráfica de GitHub que muestra el aporte de cada uno de los miembros del equipo.



# 

# 

# 

# 

# 

# 

# 

Como se puede apreciar, todo el equipo ha aportado durante el desarrollo de esta primera iteración, por lo que se puede decir que el equipo ha sido eficiente y ha cumplido con sus actividades correctamente.

# Conclusiones.

* El uso del framework Django ha facilitado el desarrollo del sitio web en esta primera iteración, ya que su estructura de directorios sigue el patrón de diseño MVT, el cual está bien organizado y favorece a la mantenibilidad y a la escalabilidad.
* El diseño de la arquitectura del software ha permitido una mejor comprensión a los desarrolladores sobre el comportamiento del sistema y los componentes que interactúan interna y externamente con este.
* A lo largo de la iteración, el equipo de desarrollo pudo reforzar sus conocimientos y destacar la importancia de seguir las mejores prácticas y estándares de codificación para garantizar la calidad del código y facilitar el mantenimiento de la aplicación.
* La documentación referente a la especificación de requisitos ha sido importante para comprender detalladamente los requerimientos del sistema y su representación mediante el uso de diagramas.
* La documentación referente al diseño de la base de datos también fue importante para mejorar la eficiencia de las peticiones que realiza el sistema, además de reducir los tiempos de espera para el usuario y de esta manera favorecer a la experiencia de usuario.

# Referencias externas.

Django (s. f.). Meet Django. <https://www.djangoproject.com/>

# Referencias de documentación del proyecto.

* Plan de Proyecto (GS-PP.DOCX).
* Cronograma del Proyecto (GS-CP.XLSX).
* Plan de Fase (GS-PF-01.DOCX).
* Plan de Iteración (GS-PI-01.DOCX).
* Documento de Arquitectura del software (GS-AS-01.DOCX).
* Documento de Especificación de UI (GS-UI-01.DOCX).
* Documento de Guía de Estilos (GS-GE-01.DOCX).
* Documento de Especificación de la Base de Datos (GS-DEBD-01.DOCX).
* Documentos de Especificación de Requisitos.
  + GS-DER-01.DOCX
  + GS-DER-02.DOCX
  + GS-DER-03.DOCX
  + GS-DER-04.DOCX
  + GS-DER-05.DOCX
  + GS-DER-06.DOCX
  + GS-DER-07.DOCX
* Documentos de Casos de Uso (GS-CU-01.DOCX).
* Documento de Informe de Pruebas (GS-IP-01.DOCX).