



SENATI

FORMACIÓN PRÁCTICA REMOTA S4

202120-PIAD-415-NRC_23944

PROYECTO FINAL DEL CURSO

PEA TAREA 4

AVANCE 01

Nro. de Grupo: 02

Integrantes:

- **Angel Felipe, Castilla Osnayo**
- **German Manuel, Salcedo Ccoyllo**
- **Luis Andres, Moloche Garcia**
- **Marycielo, Neciosup Botello**

INSTRUCTOR: Arturo Huapaya Huapaya

CENTRO DE FORMACIÓN: SENATI

2021

FORMACION PRÁCTICA REMOTA CUARTO SEMESTRE



PEA TAREA 4

AVANCE 01

1. ¿Qué es Python?

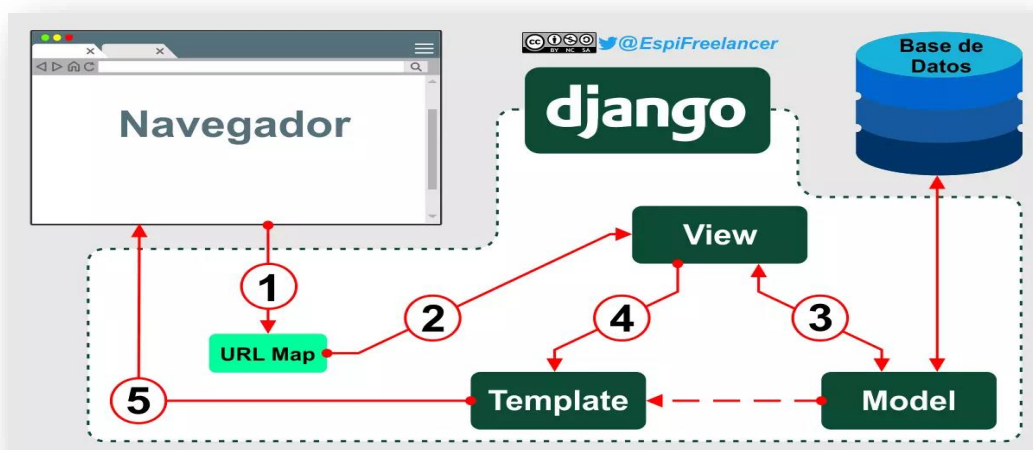
- Este es un lenguaje de programación de alto nivel, orientado a objetos, con una semántica dinámica integrada, principalmente para el desarrollo web y de aplicaciones informáticas. Siendo este un lenguaje sencillo de leer y escribir debido a su alta similitud con el lenguaje humano. Ciertamente se trata de un lenguaje multiplataforma de código abierto y, por lo tanto, gratuito, lo que permite desarrollar software sin límites. Trabajar con Python puede ser divertido cuando trabajamos en proyectos pequeños, pero si queremos sacar todo el provecho y potencial del lenguaje es inevitable pensar en usar un Framework.



2. ¿Qué es el patrón de diseño MTV?

- Estas siglas significan; **Model** (Modelo), **Template** (Plantilla) y **View** (Vista). Un patrón de diseño simplemente es son un conjunto de técnicas para resolver problemas comunes. Django implementa este patrón MVC (Modelo, Vista, Controlador) de una manera peculiar y con algunas variaciones que ellos llaman MTV. Estos tienen que implementar sus soluciones en cada sector u capa.
 - ❖ M significa “Model” (Modelo), donde sigue siendo, la capa que se encarga de comunicar e interactuar con la base de datos.
 - ❖ T significa “Template” (Plantilla), en este caso la plantilla cumple la función de nuestra vista, y es la que empleamos para mostrar los datos de nuestro usuario, generalmente la página HTML.

- ❖ V significa “View” (Vista), la “C” en el patrón de diseño MTV, es la capa que se encarga de manejar la lógica de negocios, a través de la que pasarán los datos del modelo a la plantilla.



3. ¿Qué es ORM?

- ORM procede de las siglas ORM, Object Relational Mapping. Es una pieza de softwares que nos permite interactuar con nuestra base de datos sin la necesidad de conoces SQL, utilizando el paradigma de programación orientada a objetos.
- Los ORM se encargan de traducir nuestras instrucciones en el lenguaje de programación que estemos utilizando a una sentencia SQL que el gestor de base de datos pueda comprender y ejecutar



- Algunas ventajas de Utilizar ORM:
 - Lo primero que el ORM funciona como una capa intermedia totalmente separada de la base de datos. Esto permite que puedas centrarte únicamente en el desarrollo de la aplicación.
 - Hace que una migración sea más sencilla. Si en algún momento se quiere cambiar la aplicación, no habría que reconstruir todas las consultas: podríamos migrar esa aplicación a otra base de datos sin problema.
 - El programador no necesita saber un lenguaje para cada base de datos: el ORM unifica el lenguaje.
 - Nos permite tener una mayor velocidad a la hora de hacer tareas básicas en cuanto al acceso a los datos.
 - Es un código más legible y con menos líneas.
 - Otra ventaja importante es la seguridad. Al ser una capa independiente a los datos, nos permite protegerlos de ciberataques al no estar en el mismo nivel.
 - Nos evita escribir a mano las consultas de SQL necesarias.
 - Facilita el trabajo con acciones simples y básicas de acceso a los datos. Como funciones básicas de bases de datos consideramos lo que se conoce como CRUD: Create, Read, Update y Delete.
 - Un ORM nos guarda y carga toda la información de una base de datos relacional automáticamente.



4. ¿Qué son las migraciones?

- Las migraciones son la forma en que Django propaga los cambios que realiza en sus modelos en el esquema de su base de datos. Están diseñados para ser en su mayoría automáticos, pero necesitará saber cuándo realizar migraciones, cuándo ejecutarlas y los problemas comunes con los que podría encontrarse.
- Hay varios comandos que usará para interactuar con las migraciones y el manejo de Django del esquema de la base de datos:
 - **MIGRATE:** que es responsable de aplicar y no aplicar las migraciones.
 - **MAKEMIGRATIONS:** Es responsable de crear nuevas migraciones basadas en los cambios que ha realizado en sus modelos.
 - **SQLMIGRATE:** Muestra las instrucciones SQL para una migración.
 - **SHOWMIGRATIONS:** que enumera las migraciones de un proyecto y su estado.



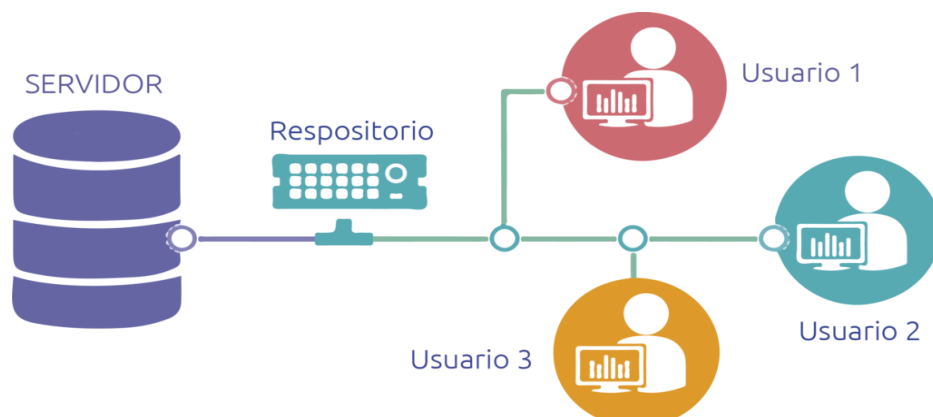
5. ¿Cómo se gestionan las rutas en Django?

- Django usa un sistema de rutas (Views) y un mapeo de URLs que permite controlar el despliegue de las Views, conocida como URLConf, su función es leer la URL que el usuario solicitó y encontrar la vista apropiada que será mostrada en el navegador del usuario. El URLConf está construido con expresiones regulares en Python, permitiendo que las rutas que maneje Django sean agradables y entendibles para el usuario.



6. ¿Qué es un sistema de control de versiones?

- Un sistema de control de versiones es una herramienta capaz de registrar todos los cambios que se realizan en uno o más proyectos, permitiéndonos volver en el tiempo a una versión anterior y, de este modo, salvar nuestro trabajo o recuperar una versión anterior de un archivo en concreto. Además de poder volver a un estado anterior, los actuales sistemas de versiones también ofrecen la posibilidad de comparar los cambios realizados a lo largo del tiempo.



7. ¿Cuál es la diferencia entre Git y Git HUB?

- Git se puede decir que es un control de versiones, es decir, nos ayuda a gestionar nuestros archivos de proyecto.
- Lo que hace principal Git es mantener un registro y gestión de nuestros proyectos, para luego mejorarlo y añadirle nuevas características, realizando un seguimiento y permite volver a la versión original del trabajo que estes realizando. Además, permite trabajar con personas en un mismo proyecto sin interferir en los archivos de las otras personas



- Y en cambio GitHub es una plataforma basada en la nube alrededor de Git, considerado como un servicio de redes sociales para desarrolladores donde las personas pueden ver los códigos del proyecto que estes realizando, que en esta misma pueden ayudar en mejorar el proyecto o ver los errores que esta tenga. También funciona como un servicio de repositorios de Git.

8. ¿Qué utilidad tendría utilizar Git HUB para el desarrollo de su proyecto?

- La utilidad va indicada en la forma de como almacenar nuestro proyecto dando así a la oportunidad de las personas que puedan ver nuestros errores en el código para nosotros corregirlos y mejorar en el proyecto deseado por nuestro grupo ya que de esta forma se evidenciaría nuestro aprendizaje en estos 4 ciclos.



9. Obtener una cuenta en Git HUB

Cuentas:

- <https://github.com/AngelCas07>
- <https://github.com/Cielo1202>
- <https://github.com/germanlml>
- <https://github.com/LuisAndresMoloché2003>

