**Universidad ORT Uruguay**

**Facultad de Ingeniería**

**ANTEL**

**NetOps**

**Sistema de registro y operación a nivel de red**

Entregado como requisito para la obtención del título de Analista Programador

**Matías Santos – 168085**

**Tutor: Mauro Castro**

**2024**Declaración de autoría

Quién suscribe, Matías Santos, declaro que el trabajo que se presenta en esta obra es de mi propia mano. Quiero asegurar que:

* La obra fue producida en su totalidad mientras realizaba el proyecto final para la obtención del título de Analista Programador.
* Cuando he consultado el trabajo publicado por otros, lo he atribuido con claridad;
* Cuando he citado obras de otros, he indicado las fuentes. Con excepción de estas citas, la obra es enteramente mía;
* En la obra, he acusado recibo de las ayudas recibidas;
* Cuando la obra se basa en trabajo realizado conjuntamente con otros, he explicado claramente qué fue contribuido por otros, y qué fue contribuido por mí;
* Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | Matías W. Santos |  |
|  | [Fecha] |  |

Abstract

El sistema NetOps de registro y operación es un sistema web desarrollado en Python con una base de datos SQLite, diseñado para facilitar el registro de información de servicios de VPN, acuerdos comerciales de roaming y tareas de los integrantes de la unidad de Operaciones de Red de Núcleo Móvil de Antel.

Se desarrolla a partir de un relevamiento de requisitos planteados por el jefe de Unidad de Operaciones, el sistema consta de cuatro aplicaciones, y una interfaz de administración. Es utilizado por los funcionarios de la unidad de Operaciones para el registro de las tareas, VPN y acuerdos de roaming, también posee un sistema de login y la funcionalidad de crear script con los comandos a ejecutar en los equipos de red necesarios para dar de alta servicios de VPN y lanzamientos comerciales de roaming.

El sistema se desarrolló para correr en un servidor Linux, usando el framework Django en el backend, y para el frontend HTML5/CSS/JavaScript, así como también el framework Bootstrap y SweetAlert para los pop up.

El acceso a los datos en la base de datos será a través del ORM de Django (Object Relational Mapping) el cuál es muy potente y facilita las consultas y obtención de datos.

El diseño y desarrollo están hechos a la medida de las necesidades, pero el sistema es escalable a futuro, lo que significa que en caso de requerirse pueden desarrollarse otras aplicaciones si surgen nuevas necesidades.

Del lado web se desarrolló implementando el patrón MVT en el cual trabaja Django, y se mantuvo en el frontend la imagen corporativa de la empresa.

Como metodología de trabajo y gestión de proyecto, se utilizó Scrum adaptado a mis posibilidades y como repositorio para todo el código se utilizó Github.

Palabras clave

* APN
* Bootstrap
* Django
* Javascript
* MVT
* Python
* Roaming
* SQLite
* VPN

Índice

[1. Introducción 6](#_Toc54939673)

2. Glosario 7

[3. Anteproyecto 8](#_Toc54939676) 3.1 Descripción del cliente 8  
 3.2 Descripción del problema 8  
 3.3 Alternativas y solución propuesta 10  
 3.4 Lista de necesidades 10  
 3.5 Objetivos del proyecto 11  
 3.6 Actores involucrados ……………………………………………………….12  
4. Requerimientos …………...…………………………………………………….12  
 4.1 Requerimientos funcionales ………..………………………………………12  
 4.2 Requerimientos no funcionales …………..………………………………...15

5. Procesos de negocio 16

5.1. Procesos de negocios a ser desarrollados 16

5.2. Riesgos que se pueden presentar 17

5.3. Pila de productos del proyectos 18  
 5.4. Diagrama de Gantt………………………………………………………… 19

5.5. Sprints…………………………………………………………………….. 20

6. Framework Django …………………………………………………………… 24  
 6.1. Cómo funciona Django …………………………………………………... 25  
 6.2. Django y SQLite …………………………………………………………. 28  
 6.3. ORM de Django ………………………………………………………….. 29

1. Introducción

En el presente documento se presenta el desarrollo de un sistema para gestionar el registro y visualización de los datos de las VPN, así como también la creación de scripts de comandos para una red de telefonía móvil para los servicios de VPN cuyo APN es de la forma nombre.vpnantel.

También se gestiona el registro y visualización de los datos de los operadores con acuerdos de roaming, y la posibilidad de descargar el script de comandos para su ejecución en la red de telefonía móvil, para ejecutar el lanzamiento correspondiente en una fecha asignada.

Se mantiene el registro de tareas de los integrantes de la unidad de Operaciones, posibilitando la creación de informes mensuales para dicha unidad.

Actualmente la unidad mencionada no cuenta con un sistema que brinde estas posibilidades, las configuraciones a nivel de red se hacen manualmente, y no existe un script que informe los comandos a utilizar. Tampoco se cuenta con un registro de tareas que posibilite la creación de informes mensuales.

Se espera que con el desarrollo de este sistema se posibilite la descarga de los comandos a utilizar en los nodos correspondientes para cada servicio de los mencionados, lo cual dará la posibilidad a todos los integrantes de la unidad de Operaciones de ejecutar dichos comandos en los nodos de red correspondiente. Disminuyendo de esta forma el tiempo de análisis y ejecución de estas tareas y minimizando la posibilidad de errores de configuración que podrían surgir por la ejecución manual de comandos en la red.

Con respecto al registro de tareas, las mismas estarán centralizadas, y contarán con un sistema de acceso para los integrantes de la unidad mencionada y se facilitará de esta forma la creación de los informes mensuales de toda la unidad.

El principal objetivo de este sistema es colaborar con la unidad Operaciones de Red de Núcleo Móvil de Antel para automatizar tareas de operación en la red de telefonía móvil, disminuir la posibilidad errores por la ejecución de comandos de forma manual en los nodos que componen la red y centralizar el registro de tareas brindando la posibilidad de descargar informes mensualmente.

1. Glosario

APN: Access Point Name, nombre del punto de acceso que utilizan los servicios de telefonía móvil para acceder a la red del operador.

APNID: Identificador único del APN

Bootstrap: es un framework CSS de frontend utilizado para desarrollo de aplicaciones, permite construir páginas web responsivas de forma rápida y sencilla.

CSS: Cascading Style Sheets, hojas de estilo en cascada, es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.

GSMA: Global System for Mobile Communications, es la organización mundial de operadores de telefonía móvil, su labor es apoyar la telefonía móvil y su constante evolución.

IR21: documento estándar de GSMA que contiene datos técnicos de los operadores de telefonía móvil para la implementación y operación de servicios de roaming.

Lanzamiento de roaming: acuerdo internacional entre Antel y cualquier otro operador de telefonía móvil a nivel mundial para permitir el registro y el uso de la red de telefonía móvil cuando un cliente Antel visita un determinado país, y viceversa.

MVT: patrón de diseño, modelo vista template, es una redefinición del patrón MVC.

ORM: Object Relational Mapping, mapeo relacional de objetos, es una pieza de software que se encarga de hacer consultas e interactuar con la base de datos, usando lenguaje de backend, sin necesidad de escribir consultas SQL. Hace el mapeo de las estructuras de las bases de datos a objetos.

PLMN: Public Land Mobile Network, red móvil pública terrestre, es el conjunto de servicios de comunicación inalámbrica ofrecidos por un operador específico en un determinado país.

VPN: Virtual Private Network (Red Privada Virtual)

VPS: Virtual Private Server (Servidor Privado Virtual)

1. Anteproyecto
   1. Descripción del cliente

El cliente que tiene la necesidad y aporta la idea es la Unidad Operaciones de Red de Núcleo Móvil de Antel a través de su jefe de Unidad, Fabián Romeo.

Antel es la empresa de comunicaciones de Uruguay, brinda servicios de comunicaciones a nivel nacional e internacional satisfaciendo las expectativas de sus clientes de forma eficiente y con calidad, contribuye al desarrollo del país y el bienestar de la sociedad.

La unidad Operaciones de Red de Núcleo Móvil, se encarga de la operación y mantenimiento de los nodos de la red de núcleo de telefonía móvil, dichos nodos son los que posibilitan brindar los servicios de voz, datos y mensajería a nivel celular.

Con el desarrollo del siguiente sistema se beneficiará a la unidad mencionada para la automatización de la generación de scripts de comandos a ejecutarse en la red de telefonía móvil de Antel.

La unidad contará con un registro de los servicios de VPN de las empresas clientes de Antel, cuyas VPN sean de la forma nombre.vpnantel, así como también un registro de todos los convenios de roaming a nivel internacional con los diferentes operadores de telefonía.

Por otra parte, se beneficiará con la centralización de un registro de tareas para todos sus integrantes que posibilitará la creación de un informe mensual.

Los usuarios del sistema a desarrollar será el personal técnico de la unidad de Operaciones, y serán también quienes darán mantenimiento al mismo.

* 1. Descripción del problema

El problema fue presentado por el cliente Antel, unidad Operaciones de Red de Núcleo a través del jefe de unidad, Fabián Romeo, afirmando los siguientes comentarios en las reuniones introductorias:

* No tienen un sistema centralizado que posibilite registrar las tareas que se ejecutan en la red de telefonía móvil.
* No cuentan con un sistema de registro centralizado para los servicios de VPN que presente todos sus datos tanto a nivel del cliente como a nivel de red.
* No cuentan con un registro centralizado de la ejecución de los lanzamientos de roaming y sus configuraciones (comandos) a nivel de red.
* No cuentan con un sistema que posibilite informar a las diferentes gerencias las tareas llevadas a cabo en la unidad.
* No cuentan con automatización de tareas para configuraciones a nivel de red, que posibilite minimizar los errores humanos en la ejecución de los comandos de configuración y permita disminuir los tiempos de desarrollo de dichas tareas.

Luego de instancias de reuniones presenciales con el cliente y con compañeros de la unidad, se verifico que las tareas las agenda cada integrante de forma diferente, algunos utilizando Outlook para recordatorios, otros utilizando un bloc de notas o Excel, otros simplemente en cuaderno. Las configuraciones se realizan manualmente accediendo a los equipos de red, se siguen los pasos que están definidos en un archivo con formato txt y se va ejecutando comando a comando. En ocasiones la ejecución de tareas de forma manual lleva mucho tiempo de análisis y preparación de los comandos a ejecutar, dependiendo del requerimiento. No hay registro de tareas, se basan principalmente en los correos enviados y recibidos y en algunos casos toman nota en documentos de texto o excel.

Frente a la necesidad de desarrollar un sistema que ayude a la ejecución de comandos a nivel de red, que centralice diferentes registros, VPN, lanzamientos de roaming, tareas, y ayude a disminuir la posibilidad de errores humanos en la ejecución de dichos comandos a nivel de red, y posibilite disminuir el tiempo de trabajo realizado para cada tarea, los integrantes de la unidad se mostraron sumamente agradecidos, colaborativos y a la orden para aportar lo que sea necesario para tal fin.

* 1. Alternativas y solución propuesta

Como solución, la propuesta fue desarrollar un sistema web que integre una aplicación para cada requerimiento, dado que cada técnico de la unidad trabaja con un mínimo de dos monitores, no hay necesidad de desarrollo de una aplicación móvil, además estas tareas se llevan a cabo en horario laboral y no en horas de guardia.

Se propuso que el sistema sea desarrollado en lenguaje Python dado que Antel brinda capacitación en dicho lenguaje, Fabián Romeo, realiza diferentes tareas utilizando Python por lo cual cuenta con experiencia y conocimientos del lenguaje, al igual que otros integrantes de la unidad que también han realizado la capacitación en dicho lenguaje.

* 1. Lista de necesidades

Luego de reuniones con el cliente, se determinan y confirman las necesidades del sistema que figuran a continuación:

* Registrar y mantener el inventario de las VPN cuyo APN es de la forma nombre.vpnantel creadas en la red móvil.
* Posibilitar la búsqueda de los datos de una VPN por su APN o por su APNId
* Posibilitar la descarga de un script de comandos pronto para su ejecución en la red móvil para cada VPN.
* Registrar y mantener el inventario de lanzamientos de roaming con los diferentes operadores a nivel mundial.
* Posibilitar la descarga de un script de comandos para cada lanzamiento teniendo en cuenta el carrier de señalización a utilizar.
* Posibilitar la carga del documento IR21 en formato pdf de los operadores de telefonía móvil cuando se agende un lanzamiento.
* Registrar y mantener el inventario de tareas realizadas por todos los integrantes de la unidad de Operaciones.
* Posibilitar la descarga de un informe mensual de las tareas registradas.
* Creación de una aplicación que posibilite el registro y login de usuarios al sistema, permitirá solo el ingreso de los usuarios de la unidad de Operaciones.
* Desarrollar el sistema manteniendo la imagen corporativa de la empresa en el diseño del frontend.
* Creación de un usuario administrador que será quien habilite los usuarios que se registren en el sistema.
* Listados de las VPN, tareas y lanzamientos de roaming registrados en el sistema.
  1. Objetivos del proyecto

El principal objetivo del proyecto es brindar una solución de software que posibilite la creación de scripts de comandos para ser ejecutados en la red de telefonía móvil, contar con un registro de los servicios de VPN móvil de la empresa, lanzamientos de roaming y tareas realizadas en la unidad de Operaciones.

Disminuir el tiempo de ejecución de las tareas relacionadas al sistema, ayudando a minimizar la ocurrencia de errores humanos en la ejecución de comandos a nivel de red. Brindar la posibilidad rápida de obtención de los datos de una VPN mediante una búsqueda por APN o APNId sin la necesidad de ingresar a los nodos que componen la red móvil.

* 1. Actores involucrados
* Jefe de la unidad Operaciones de Red de Núcleo Móvil de Antel.
* Técnicos de la unidad de Operaciones (usuarios del sistema).

1. Requerimientos
   1. Requerimientos funcionales

RF01: Registro de usuario

Permitir mediante un formulario web completar los datos para dar de alta un usuario en el sistema.

RF02: Habilitar usuario registrado

El sistema mediante una interfaz de administración debe permitir habilitar los usuarios que se registran en el sistema.

RF03: Login de usuario

Acceder al sistema mediante ingreso de email y contraseña, permitiendo el acceso a las aplicaciones del sistema.

RF04: Cerrar sesión

El sistema debe permitir cerrar la sesión del usuario activo.

RF05: Registro de una VPN

Permitir mediante un formulario web completar los datos para dar de alta una VPN en el sistema.

RF06: Buscar VPN por APN

Permitir mediante un formulario el ingreso de un APN para la búsqueda de los datos de la VPN asociada al APN que se desea buscar.

RF07: Buscar VPN por APNId

Permitir mediante un formulario el ingreso de un APNId para la búsqueda de los datos de la VPN asociada al APNId que se desea buscar.

RF08: Actualización de los datos de una VPN

Permitir actualizar los datos de una VPN registrada mediante formulario.

RF09: Creación de script de comandos de VPN

El sistema debe permitir descargar para cada VPN un script con los comandos necesarios para ejecutar en la red de telefonía móvil de Antel.

RF10: Listado de las VPN registradas

Listado que muestre todas las VPN que hay registradas en el sistema con sus datos más relevantes.

RF11: Registrar lanzamiento de roaming

Permitir mediante un formulario web el ingreso de los datos para realizar un lanzamiento de roaming con otro operador de telefonía móvil a nivel mundial.

RF12: Cambiar el estado de un lanzamiento de roaming

El sistema debe permitir cambiar el estado de un lanzamiento de roaming registrado, cambiar de estado pendiente a ejecutado o finalizado.

RF13: Listado de los lanzamientos de roaming registrados en el sistema.

Listado que muestre los lanzamientos de roaming registrados, importante posibilitar la descarga del documento IR21 para el operador del convenio.

RF13: Creación de script de comandos de lanzamiento de roaming

El sistema debe permitir descargar para cada lanzamiento de roaming un script con los comandos del lanzamiento de acuerdo al carrier de señalización a utilizar, para ser ejecutados en los nodos de la red.

RF14: Registrar tareas de los integrantes de la unidad Operaciones

El sistema debe permitir mediante formulario web el registro por parte de los integrantes de la unidad Operaciones de las tareas que realizan.

RF15: Listado de las tareas registradas por los integrantes de Operaciones

Listado con todas las tareas que están registradas en el sistema.

RF16: Informe mensual de tareas de la unidad Operaciones

El sistema debe permitir descargar un documento con las tareas realizadas mensualmente por los integrantes de la unidad Operaciones.

RF17: Diseño del frontend acorde a la imagen corporativa de la empresa

El diseño del frontend debe estar acorde a la imagen corporativa de la empresa, manteniendo sus colores y sus logos.

* 1. Requerimientos no funcionales

RNF01: Confidencialidad

El sistema en su desarrollo no debe utilizar los datos reales de las empresas clientes de Antel, ni tampoco de ANTEL.

RNF02: Ejecución

El sistema debe correr en computadoras con Windows y Linux, en los navegadores más comunes, por ejemplo, Firefox, Edge.

RNF03: Usabilidad

El sistema debe mantener las interfaces de usuario lo más similares posibles entre ellas, en caso de mensajes al usuario deben ser claros y de fácil visibilidad.

RNF04: Lenguaje y programación

El sistema debe ser diseñado e implementado utilizando el lenguaje de programación Python, en el frontend HTML5, CSS y Javascript, pudiendo utilizar frameworks como Bootstrap o similares.

1. Procesos de negocios
   1. Procesos de negocios a ser desarrollados

Los procesos de negocio son las tareas o conjunto de actividades realizadas por una empresa para alcanzar sus resultados y generar más valor al cliente.

Es una actividad que ofrece valor a los clientes y apoya, gestiona otros procesos.

Las funcionalidades mencionadas por el cliente seguirán el siguiente proceso para su desarrollo:



Los requerimientos mencionados por el cliente son los siguientes:

* Proceso de registro de usuarios en el sistema (pendientes de habilitación)
* Proceso de login para acceder al sistema.
* Proceso de registro de una VPN con sus datos.
* Proceso de actualización de los datos de una VPN
* Proceso de búsqueda de una VPN por su APN
* Proceso de búsqueda de una VPN por su APNId
* Listado de los registros de VPNs
* Proceso de creación de script con los comandos de red de las VPN
* Proceso de registro de un lanzamiento de roaming con otro operador, en estado pendiente.
* Proceso de actualización del estado del registro de lanzamiento, pasar de pendiente a ejecutado o finalizado.
* Listado de los lanzamientos de roaming registrados
* Proceso de creación del script de comandos del lanzamiento de roaming
* Proceso de registro de una tarea por parte de los integrantes de la unidad de Operaciones, que cuenten con acceso al sistema
* Listado de las tareas registrados por los integrantes de Operaciones
* Proceso de generación de informe mensual de las tareas registradas para la unidad Operaciones.
  1. Riesgos que se pueden presentar

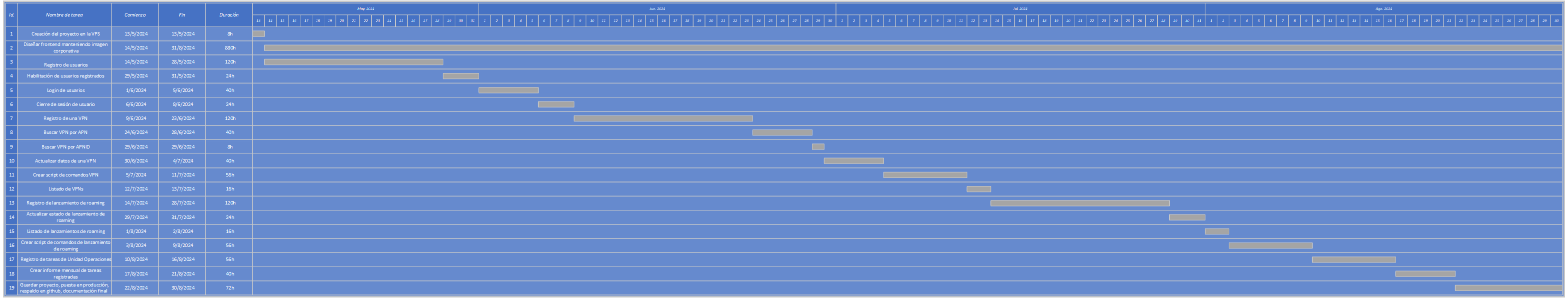
Un riesgo es un evento o condición incierta, que, de ocurrir, tendrá un impacto en el proyecto.

Riesgos:

* El desarrollo del sistema en las tecnologías necesarias (Django) requiere capacitación.
* La generación de los scripts de comandos para ser ejecutados en la red requiere capacitación.
  1. Pila de productos del proyecto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Prioridad** | **Descripción** | **Tiempo estimado** | **Responsable** |
| 1 | Alta | Creación del proyecto/ambiente en VPS | 8 hr | Matías Santos |
| 2 | Alta | Diseñar frontend manteniendo imagen corporativa | Todo el proyecto | Matías Santos |
| 3 | Alta | Registro de usuarios | 15 días | Matías Santos |
| 4 | Alta | Habilitar usuarios registrados | 3 días | Matías Santos |
| 5 | Alta | Login de usuarios | 5 días | Matías Santos |
| 6 | Media | Logout de usuarios | 3 días | Matías Santos |
| 7 | Alta | Registro de una VPN | 15 días | Matías Santos |
| 8 | Media | Buscar VPN por APN | 5 días | Matías Santos |
| 9 | Media | Buscar VPN por APNId | 8 hr | Matías Santos |
| 10 | Alta | Actualizar datos de VPN | 5 días | Matías Santos |
| 11 | Alta | Implementar script de comandos de VPN | 7 días | Matías Santos |
| 12 | Media | Listado de VPNs | 2 días | Matías Santos |
| 13 | Alta | Registro de lanzamiento de roaming | 15 días | Matías Santos |
| 14 | Media | Actualizar estado de lanzamiento de roaming | 3 días | Matías Santos |
| 15 | Media | Listado de lanzamientos de roaming | 2 días | Matías Santos |
| 16 | Media | Implementar script de comandos de lanzamientos de roaming | 7 días | Matías Santos |
| 17 | Alta | Registro de tareas de Unidad Operaciones | 7 días | Matías Santos |
| 18 | Media | Implementar informe mensual de tareas registradas | 5 días | Matías Santos |
| 19 | Alta | Respaldar proyecto, subir a Github, dejar operativo en VPS, finalizar documentación | 9 días | Matías Santos |

* 1. **Diagrama de Gantt (cronograma de trabajo)**



* 1. **Sprints**

**Sprint 1:** Creación del proyecto y ambiente de trabajo en VPS Linux.

Descripción de tareas a realizar: instalación de un entorno virtual Python, instalación de las librerías necesarias para utilizar Django, instalación de Git, instalación de Django.   
Creación del proyecto, configuración de su estructura de directorios y archivos. Creación de los templates base del proyecto con los archivos de Bootstrap, CSS y Javascript necesarios para comenzar a implementar las aplicaciones y mantener la imagen corporativa en sus diseños.

Duración estimada: 1 día

**Sprint 2:** Creación de aplicación de usuarios.

Descripción de tareas a realizar: creación de la aplicación de usuarios, creación del modelo de usuarios, migraciones utilizando los comandos que provee Django, creación y configuración de archivo de urls, vistas y templates necesarios para registrar usuarios.

Duración estimada: 15 días

**Sprint 3:** Habilitar usuarios registrados

Descripción de tareas a realizar: habilitar el sitio administrador de Django, crear el usuario administrador del sitio, migrar los modelos al sitio administrador para poder operar desde dicho sitio, el sitio administrador que provee Django nos dará la posibilidad de tener una interfaz amigable para administrar los datos de la base de datos, entre ellos, poder habilitar los usuarios que se registren.

Duración estimada: 3 días

**Sprint 4:** Login de usuarios

Descripción de tareas: diseñar e implementar la interfaz de login de usuarios, definir la url, la vista y los templates necesarios.

Duración estimada: 5 días

**Sprint 5:** Logout de usuarios

Descripción de tareas: diseñar e implementar la funcionalidad de logout para los usuarios logueados, definir las vistas, urls y templates que sean necesarios.

Duración estimada: 3 días

**Sprint 6:** Registro de VPN

Descripción de tareas: crear la aplicación para VPN, definir el modelo a utilizar, realizar las migraciones correspondientes con los comandos que provee Django. Diseñar e implementar la vista necesaria, así como también la url y los templates necesarios.

Duración estimada: 15 días

**Sprint 7:** Buscar VPN por APN

Descripción de tareas: diseñar e implementar la funcionalidad para poder realizar una búsqueda de los datos de una VPN a partir del campo APN, diseñar la url, vista y templates necesarios para tal fin.

Duración estimada: 5 días

**Sprint 8:** Buscar VPN por APNId

Descripción de tareas: diseñar e implementar la funcionalidad para poder realizar una búsqueda de los datos de una VPN a partir del campo APNID, diseñar la url, vista y templates necesarios para tal fin.

Duración estimada: 1 día

**Sprint 9:** Actualizar datos de una VPN

Descripción de tareas: diseñar e implementar la funcionalidad de actualizar los datos de una VPN, diseñar la url, vista y templates necesarios.

Duración estimada: 5 días

**Sprint 10:** Implementar script de comandos para VPN

Descripción de tareas: analizar la generación de un script con los comandos necesarios para configurar la VPN en la red móvil, crear el script con los comandos prontos para ejecutar.

Duración estimada: 7 días

**Sprint 11:** Listado de las VPN registradas

Descripción de tareas: diseñar e implementar el listado de las VPN registradas, así como también la url, vista y templates necesarios.

Duración estimada: 2 días

**Sprint 12:** Registro de lanzamiento de roaming

Descripción de tareas: crear la aplicación de lanzamientos, el modelo necesario, realizar la migración a la base de datos con los comandos que nos provee Django, diseñar e implementar la url, vista y templates necesarios para registrar un lanzamiento, en estado pendiente de ejecución.

Duración estimada: 15 días

**Sprint 13:** Actualizar el estado de un lanzamiento de roaming

Descripción de tareas: analizar e implementar la funcionalidad de actualizar el estado de un lanzamiento de roaming, de estado pendiente a ejecutado, crear la url, vista y templates necesarios para tal fin.

Duración estimada: 3 días

**Sprint 14:** Listado de lanzamientos de roaming

Descripción de tareas: diseñar e implementar el listado de lanzamientos de roaming registrados en la aplicación, crear la url, vista y templates necesarios para tal fin.

Duración estimada: 2 días

**Sprint 15:** Implementar script de comandos de lanzamiento de roaming

Descripción de tareas: analizar la generación de un script con los comandos necesarios para configurar un lanzamiento de roaming en la red móvil, crear el script con los comandos prontos para ejecutar.

Duración estimada: 7 días

**Sprint 16:** Registro de tareas de la Unidad Operaciones

Descripción de tareas: crear la aplicación de tareas, el modelo necesario y realizar la migración correspondiente con los comandos que nos provee Django. Diseñar e implementar la url, vista y templates necesarios para registrar una tarea por parte de un integrante de Operaciones.

Duración estimada: 7 días

**Sprint 17:** Implementar informe mensual de tareas

Descripción de tareas: diseñar e implementar la creación de un informe mensual de tareas, que se pueda descargar, crear la url, vista y templates necesarios para tal fin.

Duración estimada: 5 días

**Sprint 19:** Respaldar proyecto, dejarlo operativo en VPS Linux, probar acceso, subir a Github.

Descripción de tareas: respaldar todo el proyecto, dejarlo operativo en la VPS Linux, subir todo el proyecto a Github, documentar todo lo realizado.

Duración estimada: últimos días restantes (aproximadamente 9 días)

1. Framework Django

Django es un framework (marco de trabajo) web de alto nivel que permite el desarrollo de sitios web seguros, escalables y mantenibles. Es gratuito y de código abierto, esta desarrollado con el lenguaje Python, y consta de una gran comunidad y documentación.

Django sigue el principio DRY, “Don’t repeat yourself”, cuyo significado es no te repitas, es una filosofía de definición de procesos que promueve la reducción de la duplicación especialmente en computación, según este principio toda pieza de información nunca debería ser duplicada debido a que la duplicación incrementa la dificultad en los cambios y evolución posterior, puede perjudicar la claridad y crear un espacio para posibles inconsistencias.

Seguir dicho principio permite lograr una alta capacidad de mantenimiento del proyecto, facilidad para realizar cambios y pruebas de alta calidad. Este principio es clave para crear buenas aplicaciones, con la unificación y reutilización se mejora la calidad del código base y a su vez se mejora la capacidad de mantenimiento.

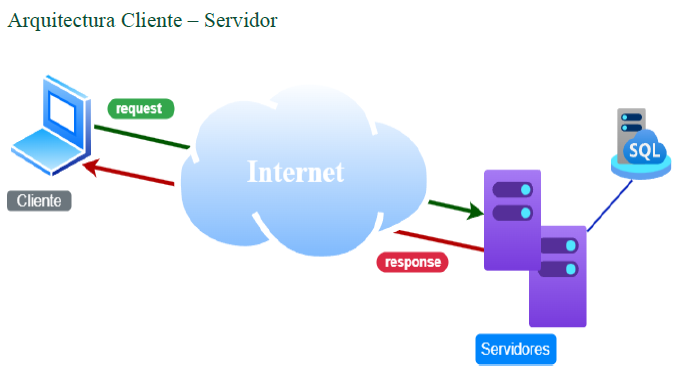
El uso de Django en el desarrollo web ofrece numerosas ventajas, entre ellas:

* Productividad; Django acelera el proceso de desarrollo web gracias a sus componentes reutilizables y su capacidad para automatizar tareas comunes.
* Seguridad; Django se enfoca en la seguridad, lo que lo convierte en una elección sólida para aplicaciones web sensibles que requieren protección contra amenazas comunes.
* Escalabilidad; Django es altamente escalable, lo que permite que las aplicaciones crezcan y se adapten a las necesidades cambiantes a medida que evolucionan.

**6.1. Como funciona Django**

En un sitio web tradicional basado en datos una aplicación web espera peticiones http del explorador web (o de otro cliente). Cuando se recibe una petición web la aplicación elabora lo que se necesita basándose en la URL y posiblemente en la información incluida en los datos POST o GET que le llegan desde el cliente. Dependiendo de que se necesita quizás pueda entonces leer o escribir información desde una base de datos o realizar tareas requeridas para satisfacer la petición recibida.

La aplicación devolverá a continuación una respuesta al explorador web, con frecuencia creando dinámicamente una página html para que el explorador la presente insertando los datos recuperados o procesados en marcadores de posición dentro de una plantilla html.



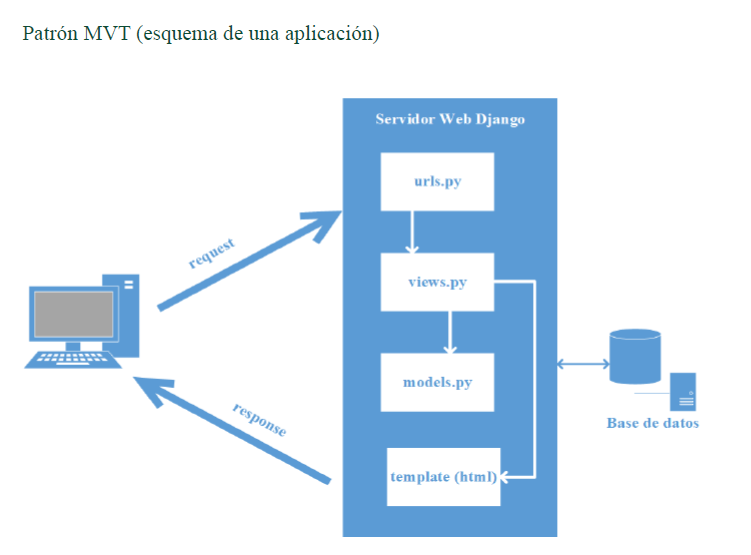
Django sigue el patrón de diseño **MVT** (modelo, vista, template), que es una redefinición del conocido patrón MVC (modelo, vista, controlador).

El **modelo** es la parte de la aplicación que maneja los datos, contiene todo lo relacionado a los datos, como acceder, como validarlos, cuál es su comportamiento y las relaciones entre ellos.

La **vista** describe los datos que se envían al cliente (usuario), pero no describe su presentación, es la capa lógica del negocio, contiene la lógica que accede al modelo y lo lleva al template apropiado.

El **template** (plantilla) es la parte que se encarga de mostrar la información con html, css, javascript, etc. Es la capa de presentación, contiene todas las decisiones de cómo debe mostrarse la información en la página web.

La configuración de Django posee un archivo de URLs que permite asociar las vistas con las diferentes URLs que los usuarios pueden visitar para acceder a las diferentes funciones de la aplicación.



Siguiendo ese patrón Django recibirá una petición desde un cliente, si la URL que se accede desde el cliente matchea con una función o clase en la vista, se pasará la petición a la vista correspondiente, en la vista se procesarán y/o recuperaran los datos desde el modelo (base de datos) que corresponda y finalmente se renderizará un template integrando los datos dinámicamente para luego enviar el html al cliente como respuesta a su petición.

**urls.py**: es un archivo Python que se utiliza para mapear una URL con su correspondiente función o clase de Python implementada en la vista (views.py), se usa un mapeador para redirigir las peticiones http a la vista apropiada basándose en la url de la petición. El mapeador llamado urlpatterns, puede también emparejar patrones de cadenas o dígitos específicos que aparecen en la URL y los pasan a la función o clase en la vista correspondiente como datos para ser procesados.

**views.py**: una vista es una función o clase de Python que gestiona las peticiones http que recibe y retorna una respuesta.

Las vistas acceden a los datos que necesitan para satisfacer las peticiones por medio de modelos, y delegan el formateo de la respuesta a los templates.

**models.py**: los modelos son objetos de Python que definen la estructura de los datos de una aplicación y proporcionan mecanismos para gestionar (agregar, modificar, eliminar, listar) y consultar registros en la base de datos.

**templates**: un template o plantilla es un archivo de texto que define la estructura o diseño de otro fichero (como una página html), con marcadores de posición que se utilizan para representar el contenido real. Django automáticamente buscara plantillas en un directorio llamado ‘templates’ de su aplicación. Una vista puede crear dinámicamente una página usando un template y rellenándolo con datos de un modelo.

**6.2. Django y SQLite**

Django, un marco de alto nivel para Python, incluye soporte integrado para usar SQLite como backend de base de datos. SQLite es un motor de base de datos liviano y sin servidor que almacena datos en un archivo local. Es una excelente opción para el desarrollo y aplicaciones de pequeño y mediano porte.

SQLite es una base de datos muy eficaz para cualquier desarrollo en ambientes embebidos, ofrece un alto rendimiento, eficacia, seguridad, estandarización e interoperabilidad.

SQLite no requiere una larga fase de aprendizaje, sino que puede integrarse directamente en una aplicación sin necesidad de utilizar software de servidor adicional. De este modo, la aplicación recibe automáticamente amplias funciones de base de datos sin necesidad de herramientas adicionales. Todas las tablas, referencias, enlaces y similares se alojan en **un único archivo que ahorra espacio**, por lo que también es posible almacenar opcionalmente este archivo en la memoria principal.

Algunas de sus ventajas son:

* **Multiplataforma;** SQLite funciona en cualquier plataforma por lo que es ideal para desarrollar aplicaciones, apps y webs que funcionen en Linux, Windows, Android o sistemas Apple.
* Es muy ligero; SQLite es muy ligero, integrándose con la propia aplicación. La base de datos se guarda en un solo archivo y su peso es muy pequeño en comparación al de otras alternativas existentes.  
  Este peso ligero de la base de datos la convierte en una alternativa ideal para utilizar en dispositivos que tienen capacidad limitada de almacenamiento como Smart TV, cámaras, celulares, tablets, etc.
* Reducción de costes; el código de SQLite es gratuito y público, pudiendo utilizarse de forma libre en cualquier tipo de proyecto. Además, se puede modificar para adaptarlo a las necesidades propias de cada proyecto.
* Simple de utilizar; utilizar SQLite es muy sencillo al no tener que montar un servidor para poder gestionar la base de datos. Esto facilita mucho el proceso de desarrollo al no requerir una compleja instalación y configuración. Además, está basado en el lenguaje de consultas SQL, por lo que sí lo domina se podrá utilizar SQLite sin ninguna dificultad.
* Alto rendimiento; SQLite se caracteriza por funcionar de forma muy rápida y eficiente. Cualquier consulta u operación que se realice se resuelve de manera eficiente y casi instantánea, consiguiendo que el funcionamiento de la app o web ofrezca un gran rendimiento.

**6.3. ORM de Django (Object Relational Mapping)**

**Object**; hace referencia a la programación orientada a objetos   
**Relational**; hace referencia a las bases de datos relacionales   
**Mapping**; es un proceso que consiste en emparejar valores

ORM significa mapeo relacional de objetos. Es una pieza de software que se encarga de hacer consultas e interactuar con la base de datos, usando lenguaje de backend, sin necesidad de escribir consultas SQL. Hace el mapeo de las estructuras de las bases de datos a objetos.

Un ORM convierte cada tabla de la base de datos en una clase, a cada fila o registro, en un objeto y a cada campo de esos registros, en una propiedad del objeto, y todo en el lenguaje de backend.

Django crea el mapeo de las tablas de la base de datos a estructuras y tipos de datos de Python usando el modelo de datos. Un modelo, es una clase que corresponde al nombre de una tabla en la base de datos, y cada variable de clase es un campo de dicha tabla, en esta clase también se especifican índices de la tabla, llaves foráneas, ordenamiento de los datos, reglas de validación, entre otras cosas.

Cada aplicación en nuestro sistema Django tiene un modelo asociado, y las tablas creadas son del tipo app\_nombre\_del\_modelo, por ejemplo, si tengo una aplicación “contabilidad” y mi modelo tiene la clase “Cliente”, la tabla se llamará “contabilidad\_cliente”.

Cuando el modelo está definido Django usa las migraciones para crear las tablas y campos necesarios en la base de datos, y si posteriormente hacemos modificaciones al modelo, las migraciones se encargan de replicar esos cambios.

El comando “python manage.py makemigrations” buscara cambios en todos los modelos de nuestro proyecto y creara las migraciones correspondientes. Para ejecutar las migraciones se utiliza el comando “python manage.py migrate”.

Hay que recordar que en los modelos habitualmente no definimos un campo ID, Django lo crea automáticamente, y es de tipo entero auto incremental, este campo es llave primaria de la tabla.