



Creación de una Plataforma Web para Generar y Firmar Documentos PDF con Django y Autofirma



Estudiante: Enrique Serrano Lendines

Centro Educativo: IES Medina Azahara

Licencia: Esta obra está sujeta a una licencia de Reconocimiento - No Comercial - Sin Obra Derivada 4.0 España de Creative Commons. [Creative Commons](https://creativecommons.org/licenses/by-nc-nd/4.0/es/)



ÍNDICE

1. DEFINICIÓN Y ANÁLISIS CONTEXTUAL.....	2
1.1 Contexto y Justificación.....	2
1.1.1 Antecedentes.....	2
1.1.2 Definición del Problema.....	2
1.1.3 Justificación.....	2
1.1.4 Clasificación de las Empresas del Sector:.....	2
1.2 Marco Legal.....	3
1.3 Alcance del Proyecto.....	4
1.4 Alternativas y Propuesta de Solución.....	5
2. ANÁLISIS DE REQUISITOS.....	8
2.1 Recursos Humanos:.....	8
2.2 Legales:.....	8
2.3 Hardware:.....	8
2.4 Software:.....	9
3. TEMPORALIZACIÓN.....	10
3.1 Identificación de Fases y Tareas.....	10
3.2 Secuenciación.....	11
4. DOCUMENTACIÓN TÉCNICA.....	12
4.1 Árbol de Directorios del Proyecto Web:.....	12
4.2 Manual de Implementación del Proyecto.....	13
4.2.1 Instalación Manual.....	13
4.2.2 Instalación Automática.....	24
5. ANÁLISIS ECONÓMICO.....	26
Presupuesto Detallado.....	26
6. SEGUIMIENTO Y CONTROL.....	27
6.1 Identificación de Fases y Tareas.....	28
7. FUENTES DE DOCUMENTACIÓN.....	29
8. ANEXOS.....	29
8.1 Guía de Usuario:.....	29
8.2 Referencias Adicionales:.....	29
8.3 Otros Materiales:.....	30
8.4 Mejoras Propuestas.....	31

1. DEFINICIÓN Y ANÁLISIS CONTEXTUAL



1.1 Contexto y Justificación

1.1.1 Antecedentes

En un mundo donde la digitalización está en constante avance, se ha vuelto crucial contar con herramientas que permitan administrar los documentos digitales de manera segura y eficiente. La demanda de soluciones que permitan generar documentos PDF con la opción de firmarlos rápidamente y de manera segura está aumentando, este proyecto busca satisfacer esa necesidad. Las empresas y los particulares están buscando herramientas que les permitan crear PDFs en un formato exclusivo y posteriormente firmarlos digitalmente.

1.1.2 Definición del Problema

Las herramientas actuales para la gestión de documentos digitales a menudo presentan problemas de accesibilidad, seguridad y facilidad de uso. Este proyecto propone una solución que utiliza Django para la gestión del backend y Autofirma para la firma digital, asegurando que los documentos puedan ser generados y firmados de manera segura y eficiente. Aunque el sistema de registro e inicio de sesión de usuarios no está completamente implementado, su inclusión futura está considerada para mejorar la funcionalidad de la plataforma.

1.1.3 Justificación

El desarrollo de esta plataforma está motivado por la necesidad de ofrecer una solución que no solo responda a las necesidades actuales de firma y generación de documentos, sino que también pueda adaptarse a futuros requerimientos. Empresas tecnológicas, firmas legales, instituciones educativas y organizaciones gubernamentales encontrarán en esta plataforma una herramienta valiosa para mejorar su eficiencia y seguridad documental.

1.1.4 Clasificación de las Empresas del Sector:

- **Empresas de tecnología:** Consumidores y proveedores de herramientas de firma digital.
- **Firmas legales:** Necesitan firmar contratos y documentos legales con frecuencia.
- **Instituciones educativas:** Gestión de documentos administrativos y académicos.
- **Organizaciones gubernamentales:** Gestión segura de documentos oficiales.

Recopilación de Información:

- Estudios sobre seguridad en la firma digital.



- Tecnologías para la generación de PDFs.
- Necesidades del mercado a través de encuestas y entrevistas.

1.2 Marco Legal

El marco legal aplicable al proyecto incluye una serie de leyes y regulaciones que aseguran el cumplimiento de estándares en cuanto a firma digital, protección de datos personales y normativa sobre transacciones electrónicas. Estas leyes y regulaciones son fundamentales para garantizar que la plataforma opere de manera legal y segura, protegiendo tanto a los usuarios como a las empresas que utilicen el servicio.

Leyes y Regulaciones Relevantes:

- **Firma Digital:**
 - Reglamento eIDAS (Reglamento (UE) N° 910/2014): Establece un marco para las firmas electrónicas en la Unión Europea, garantizando su validez y reconocimiento legal en todos los estados miembros.
 - Ley 59/2003, de 19 de diciembre, de firma electrónica (España): Regula el uso de la firma electrónica y sus efectos jurídicos, estableciendo los requisitos y procedimientos para su validez.
- **Protección de Datos Personales:**
 - Reglamento General de Protección de Datos (GDPR, Reglamento (UE) 2016/679): Proporciona directrices sobre la recolección, procesamiento y almacenamiento de datos personales dentro de la Unión Europea, asegurando los derechos de los individuos sobre su información personal.
 - Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales (España): Adaptación del GDPR en la legislación española, estableciendo medidas adicionales de protección de datos y derechos digitales.
- **Normativa sobre Transacciones Electrónicas:**
 - Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico (LSSI-CE): Regula las actividades comerciales electrónicas y los servicios de la sociedad de la información en España, estableciendo obligaciones para los proveedores de estos servicios.

Obligaciones Fiscales, Laborales y de Prevención de Riesgos:



- **Obligaciones fiscales:** Cumplimiento con las normativas fiscales aplicables, incluyendo el IVA y otros impuestos relacionados con la prestación de servicios digitales.
- **Obligaciones laborales:** Aseguramiento de que los empleados que trabajen en el desarrollo y mantenimiento de la plataforma cumplan con las leyes laborales vigentes, incluyendo contratos de trabajo adecuados y condiciones laborales seguras.
- **Prevención de riesgos:** Implementación de medidas de seguridad adecuadas para proteger tanto los datos personales como la integridad de los documentos firmados digitalmente. Esto incluye la utilización de tecnologías de cifrado, autenticación de usuarios y protocolos de seguridad en la transmisión de datos.

Ayudas y Subvenciones: El proyecto puede beneficiarse de diversas ayudas y subvenciones destinadas a fomentar la digitalización de PYMEs y la innovación tecnológica:

- **Subvenciones para la digitalización de PYMEs:** Programas nacionales y europeos que apoyan la adopción de nuevas tecnologías por parte de pequeñas y medianas empresas. Estos programas pueden ofrecer financiación para la implementación de soluciones de firma digital y gestión documental.
- **Programas de apoyo a la innovación tecnológica:** Iniciativas gubernamentales y de la Unión Europea que proporcionan fondos para proyectos innovadores que mejoren la eficiencia y seguridad de los procesos empresariales mediante el uso de tecnologías avanzadas.

1.3 Alcance del Proyecto

El proyecto se enfoca en desarrollar una plataforma web integral que ofrezca una solución eficiente y segura para la generación y firma de documentos PDF. Esta plataforma se diseñará para satisfacer las necesidades de individuos y empresas que requieren gestionar documentos de manera digital, asegurando la autenticidad y confidencialidad de la información.

Funcionalidades Principales:

- **Generación de Documentos PDF a partir de Formularios Web:** La plataforma permitirá a los usuarios crear documentos PDF rellenando formularios en línea. Estos formularios serán personalizables y podrán incluir campos de texto, opciones de selección, y otros elementos necesarios para la recopilación de datos.
- **Firma Digital de Documentos PDF utilizando Autofirma:** Implementación de Autofirma para la firma digital de documentos PDF. Esto garantizará que



los documentos firmados electrónicamente sean legalmente vinculantes y seguros.

- **Garantía de Seguridad y Confidencialidad de los Documentos:** La plataforma incorporará medidas de seguridad robustas, incluyendo cifrado de datos, autenticación de usuarios y controles de acceso. Se asegurará que los documentos sean accesibles solo para las personas autorizadas.
- **Colaboración entre Múltiples Usuarios:** La plataforma permitirá la colaboración entre varios usuarios, facilitando la revisión y firma de documentos por múltiples partes. Esto incluirá funciones de seguimiento de cambios y comentarios para mejorar la colaboración y comunicación.

Recopilación de Información: Para desarrollar la plataforma, se realizará un análisis exhaustivo de herramientas similares y tecnologías disponibles en el mercado. Esto incluirá:

- Evaluación de soluciones existentes de generación y firma de documentos PDF.
- Investigación sobre las mejores prácticas de seguridad en la gestión de documentos digitales.
- Identificación de tecnologías y bibliotecas que faciliten la integración de formularios web y firmas digitales.

Establecimiento de Objetivos: Los objetivos específicos del proyecto incluyen:

- Implementar una plataforma funcional y segura: Asegurar que la plataforma sea confiable y cumpla con los requisitos de seguridad y funcionalidad.
- Desarrollar una interfaz amigable para el usuario: Crear una experiencia de usuario intuitiva que facilite la generación y firma de documentos PDF.
- Soporte multiplataforma: Garantizar que la plataforma sea accesible desde diferentes dispositivos y sistemas operativos, permitiendo a los usuarios trabajar de manera flexible y eficiente.

1.4 Alternativas y Propuesta de Solución

Para abordar la necesidad de una solución eficiente y segura para la generación y firma de documentos PDF, se han considerado varias alternativas. Cada alternativa presenta diferentes ventajas y desafíos en términos de seguridad, costos y facilidad de uso.

Alternativas:

1. **Implementar Firma Digital utilizando Hardware Específico para Mayor Seguridad:**



- **Ventajas:** Proporciona un alto nivel de seguridad mediante el uso de dispositivos de hardware dedicados, como tokens de seguridad o tarjetas inteligentes.
 - **Desafíos:** Costos adicionales asociados con la adquisición y mantenimiento de hardware específico. Mayor complejidad en la implementación y posible dificultad para los usuarios no técnicos.
2. **Utilizar Soluciones Completamente Basadas en la Nube:**
- **Ventajas:** Facilita el acceso y la gestión de documentos desde cualquier lugar y dispositivo con conexión a internet. Puede ofrecer una escalabilidad flexible y una reducción de costos de infraestructura.
 - **Desafíos:** Dependencia de proveedores de servicios en la nube y posibles preocupaciones sobre la privacidad y seguridad de los datos almacenados en servidores externos.
3. **Desarrollar una Aplicación Móvil Complementaria:**
- **Ventajas:** Proporciona una solución accesible y conveniente para los usuarios que prefieren utilizar dispositivos móviles. Puede mejorar la movilidad y la flexibilidad en el manejo de documentos.
 - **Desafíos:** Necesidad de desarrollar y mantener versiones de la aplicación para diferentes sistemas operativos móviles. Mayor costo y tiempo de desarrollo.

Propuesta de Solución: Optar por una plataforma web que combine la generación y firma de documentos PDF utilizando Django para el backend y Autofirma para la firma digital. Esta solución ofrece un equilibrio entre costo, seguridad y facilidad de uso.

Justificación de la Propuesta de Solución:

- **Equilibrio entre Costo y Seguridad:** La implementación de una plataforma web utilizando Django y Autofirma permite aprovechar tecnologías robustas y seguras sin incurrir en los costos adicionales asociados con hardware específico.
- **Facilidad de Uso:** La plataforma web será accesible desde cualquier dispositivo con conexión a internet, ofreciendo una interfaz amigable y fácil de usar para todos los niveles de usuarios.
- **Escalabilidad y Flexibilidad:** La solución basada en Django permite una fácil escalabilidad y adaptabilidad a futuros requerimientos, mientras que Autofirma proporciona una firma digital segura y legalmente vinculante.

Planificación del Desarrollo, Diseño, Implementación y Pruebas:

- **Desarrollo de una Plataforma Web:**
 - **Backend:** Utilizar Django para manejar la lógica del servidor, la gestión de documentos y la generación de PDFs.



- **Firma Digital:** Integrar Autofirma para la firma digital de documentos, asegurando la autenticidad e integridad de los documentos firmados.

Elaboración del Guión de Trabajo:

- **Fase de Planificación:** Definición de requisitos y especificaciones del proyecto. Elaboración del plan de trabajo y cronograma.
- **Fase de Diseño:** Diseño de la arquitectura del sistema y la interfaz de usuario.
- **Fase de Implementación:** Desarrollo del backend en Django y la integración con Autofirma. Desarrollo del frontend para la interfaz de usuario.
- **Fase de Pruebas:** Pruebas unitarias, de integración y de aceptación para asegurar la funcionalidad y seguridad de la plataforma.

Estudio de Viabilidad Técnica:

- **Análisis de Tecnologías:** Evaluación de Django y Autofirma en términos de capacidades, seguridad y facilidad de integración.
- **Recursos Necesarios:** Identificación de los recursos humanos, tecnológicos y financieros necesarios para el desarrollo del proyecto.
- **Factibilidad del Proyecto:** Análisis de los posibles riesgos y mitigación de los mismos para asegurar la viabilidad técnica y financiera del proyecto.



2. ANÁLISIS DE REQUISITOS

Para el desarrollo de la plataforma web destinada a la generación y firma de documentos PDF, es crucial identificar y analizar los requisitos necesarios en términos de recursos humanos, legales, hardware y software. A continuación se detallan algunas características:

- La plataforma será accesible desde cualquier equipo que disponga de conexión a Internet.
- Altos estándares de seguridad.
- Compatibilidad multiplataforma.
- Interfaz de usuario intuitiva.

A continuación se detallan estos requisitos:

2.1 Recursos Humanos:

- **Desarrolladores Web:** Conocimientos en Django y JavaScript. Experiencia en desarrollo backend y frontend. Capacidad para integrar herramientas y bibliotecas de terceros.
- **Diseñadores UI/UX:** Habilidad en diseño de interfaces de usuario intuitivas y amigables. Experiencia en la creación de experiencias de usuario atractivas y funcionales.
- **Especialistas en Seguridad Informática:** Conocimientos en protocolos de seguridad y cifrado. Experiencia en la implementación de medidas de seguridad para proteger datos sensibles. Capacidad para realizar auditorías de seguridad y pruebas de penetración.

2.2 Legales:

- **Cumplimiento con la Normativa de Protección de Datos (GDPR):** Asegurar que el procesamiento y almacenamiento de datos personales cumplan con los requisitos del GDPR. Implementar políticas de privacidad y consentimiento informado para los usuarios.
- **Licencias de Software:** Adquisición de licencias necesarias para las herramientas y bibliotecas utilizadas en el desarrollo de la plataforma. Asegurar el cumplimiento con las licencias de código abierto y propietarias.

2.3 Hardware:

- **Servidores:** Servidores dedicados o en la nube para alojar la aplicación web. Capacidad suficiente para manejar el tráfico de usuarios y almacenamiento de documentos.



- **Dispositivos de Prueba:** PCs, tablets y móviles para realizar pruebas de compatibilidad y usabilidad. Diversos sistemas operativos y navegadores para asegurar la funcionalidad multiplataforma.

2.4 Software:

- **Django Framework:** Framework de desarrollo web basado en Python, conocido por su robustez y flexibilidad. Utilizado para el desarrollo del backend de la plataforma.
- **Autofirma:** Herramienta para la firma digital de documentos. Integración con Django para permitir la firma segura y legalmente vinculante de documentos PDF.
- **Herramientas de Desarrollo:** IDE (Entorno de Desarrollo Integrado) como PyCharm o Visual Studio Code. Sistema de control de versiones, como Git, para gestionar el código fuente.

Estudio de Viabilidad Técnica:

- **Análisis de Tecnologías:** Evaluación de Django y Autofirma en términos de capacidades, seguridad y facilidad de integración. Comparación con otras posibles tecnologías y herramientas.
- **Recursos Necesarios:** Identificación de los recursos humanos, tecnológicos y financieros necesarios para el desarrollo del proyecto. Planificación de la adquisición y gestión de estos recursos.
- **Factibilidad del Proyecto:** Análisis de los posibles riesgos y mitigación de los mismos para asegurar la viabilidad técnica y financiera del proyecto. Evaluación del tiempo y esfuerzo requerido para cada fase del desarrollo.

Recursos y Logística:

- **Permisos y Autorizaciones:** Identificación de las necesidades de permisos y autorizaciones para llevar a cabo las actividades del proyecto. Cumplimiento con las normativas legales y reglamentarias aplicables.
- **Plan de Prevención de Riesgos:** Identificación de los riesgos inherentes a la ejecución del proyecto. Definición de medidas preventivas y correctivas para mitigar los riesgos. Implementación de un sistema de seguimiento y control de riesgos.

Control de Calidad:

- **Establecimiento de Indicadores de Calidad:** Definición de métricas para evaluar la calidad del software desarrollado. Implementación de pruebas unitarias, de integración y de aceptación para asegurar la funcionalidad y seguridad de la plataforma.



- **Procedimientos de Evaluación:** Definición de procedimientos para la evaluación continua del proyecto. Implementación de revisiones periódicas y auditorías de calidad.

3. TEMPORALIZACIÓN

3.1 Identificación de Fases y Tareas

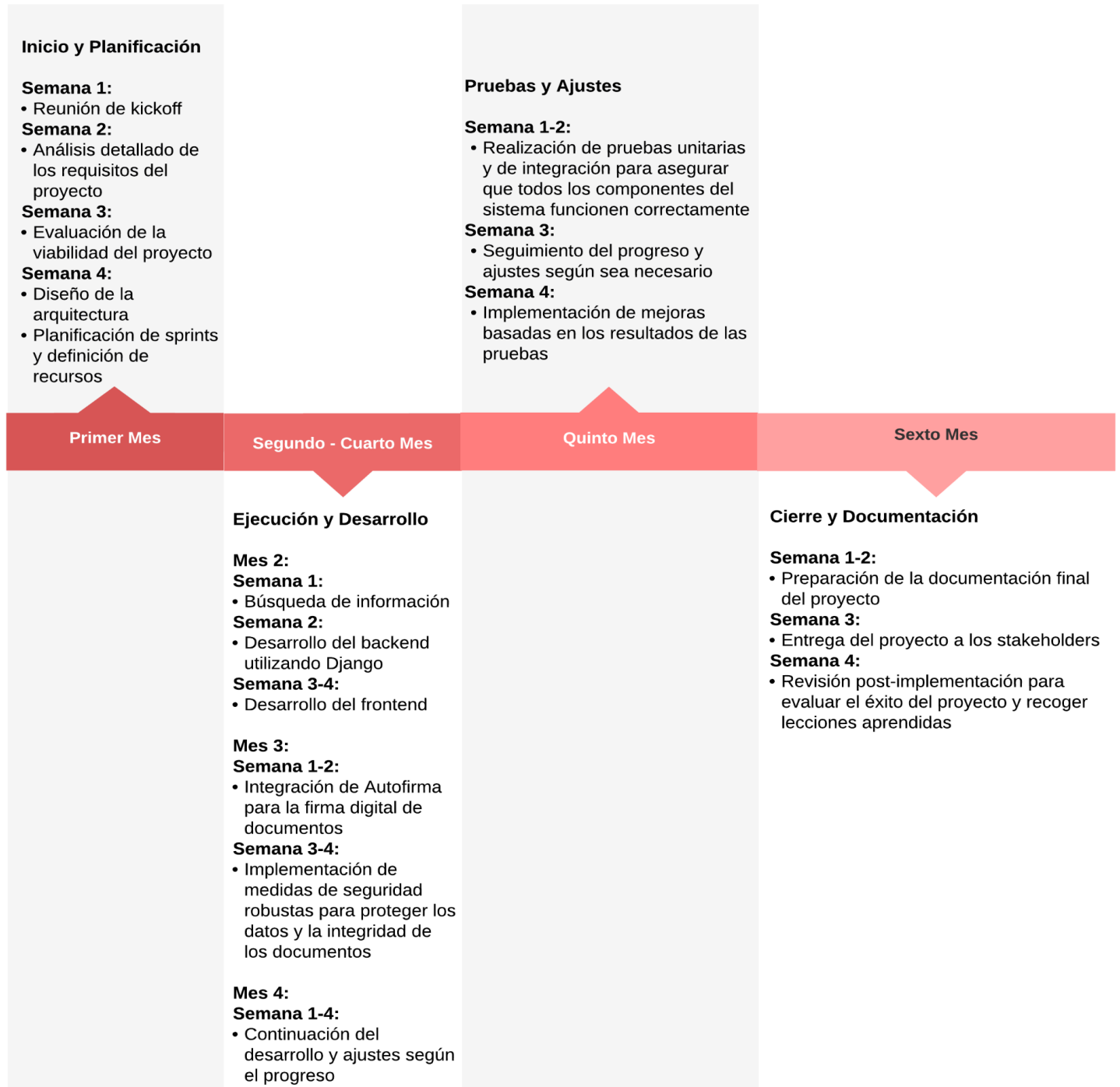
Para asegurar una gestión eficiente del proyecto, se identifican las siguientes fases y tareas:

Fases:

- **Inicio:**
 - **Objetivo:** Definir el objetivo del proyecto, identificar las necesidades y elaborar un plan preliminar. Establecer los recursos necesarios y evaluar la viabilidad del proyecto.
 - **Tareas:** Reunión de kickoff. Análisis de requisitos. Evaluación de viabilidad.
- **Planificación:**
 - **Objetivo:** Elaborar un plan detallado que incluya la definición de tareas, la asignación de responsabilidades, la estimación de tiempos y costos, y la elaboración de un cronograma.
 - **Tareas:** Diseño de la arquitectura. Planificación de sprints. Definición de recursos.
- **Ejecución:**
 - **Objetivo:** Implementar el proyecto de acuerdo al plan previamente establecido.
 - **Tareas:** Desarrollo del backend y frontend. Integración de Autofirma. Implementación de medidas de seguridad.
- **Control:**
 - **Objetivo:** Monitorear y controlar el progreso del proyecto.
 - **Tareas:** Pruebas unitarias y de integración. Seguimiento del progreso. Ajustes y mejoras.
- **Cierre:**
 - **Objetivo:** Finalizar el proyecto y evaluar su éxito.
 - **Tareas:** Documentación final. Entrega del proyecto. Revisión post-implementación.

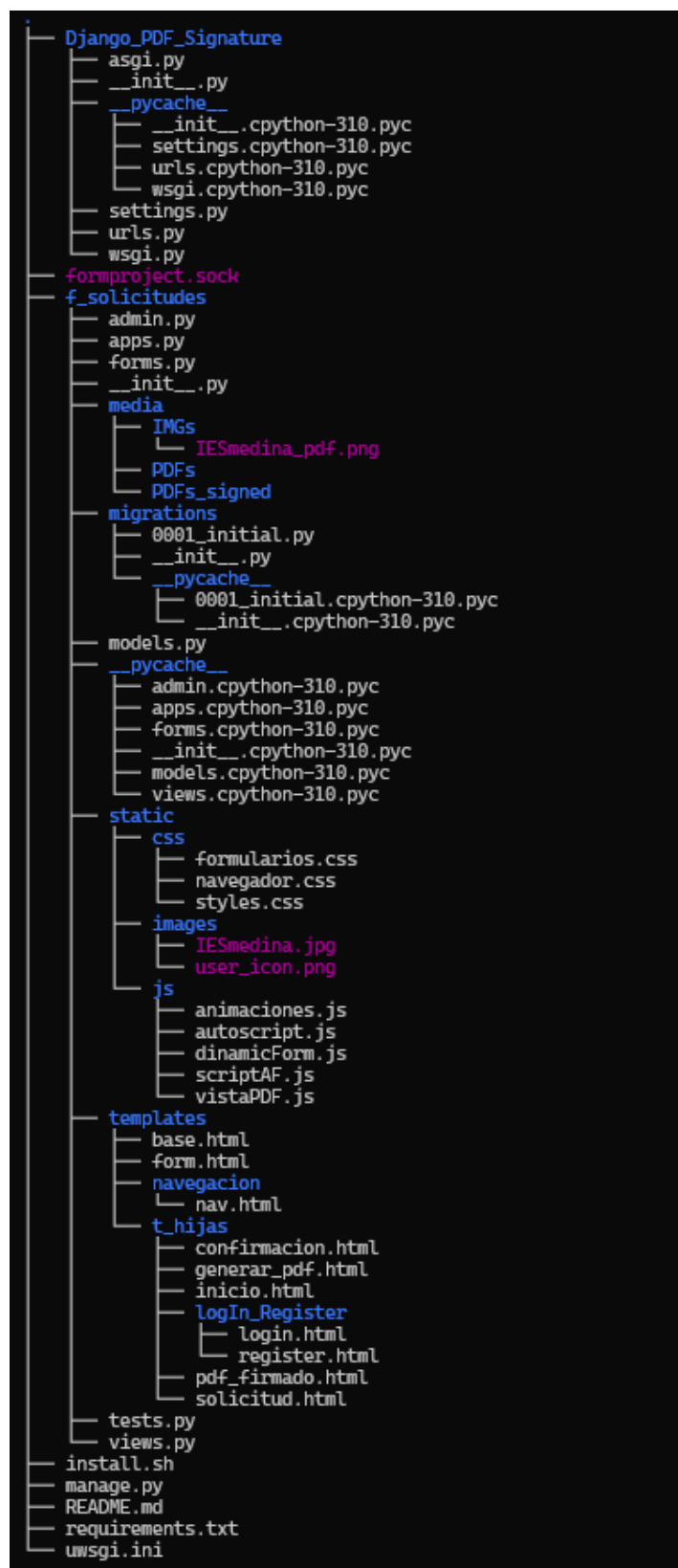
3.2 Secuenciación

Cronograma:



4. DOCUMENTACIÓN TÉCNICA

4.1 Árbol de Directorios del Proyecto Web:





4.2 Manual de Implementación del Proyecto

Antes de comenzar con la instalación, puedes usar esta propia guía o acceder directamente a mi repositorio:

Repositorio en GitHub: [Django_PDF_Signature](#)

Este manual proporciona dos métodos de instalación:

1. **Instalación Manual**
2. **Instalación Automática** (usando un script)

4.2.1 Instalación Manual

Paso 1: Instalación de Python 3, uWSGI, Nginx, PostgreSQL, unzip y dependencias Actualiza el gestor de paquetes apt:

```
sudo apt update
```

```
enrique@server:~$ sudo apt update
[sudo] password for enrique:
Des:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Obj:2 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Des:3 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Des:5 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1.517 kB]
Des:6 http://es.archive.ubuntu.com/ubuntu jammy/main Translation-es [332 kB]
Des:7 http://es.archive.ubuntu.com/ubuntu jammy/restricted Translation-es [964 B]
Des:8 http://es.archive.ubuntu.com/ubuntu jammy/universe Translation-es [1.356 kB]
Des:9 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [259 kB]
Des:10 http://es.archive.ubuntu.com/ubuntu jammy/multiverse Translation-es [68,2 kB]
Des:11 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1.731 kB]
Des:12 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1.933 kB]
Des:13 http://es.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [318 kB]
Des:14 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [329 kB]
```

Instalar Python 3, pip3, uWSGI, Nginx, PostgreSQL unzip y dependencias:

```
sudo apt install -y git python3 python3-pip uwsgi uwsgi-plugin-python3 nginx postgresql
postgresql-contrib libpq-dev
```

```
enrique@server:/var/www/html/PDjango$ sudo apt install -y git python3 python3-pip uwsgi uwsgi-plugin-python3 nginx postgresql postgresql-contrib libpq-dev
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
postgresql ya está en su versión más reciente (14+238).
postgresql-contrib ya está en su versión más reciente (14+238).
uwsgi ya está en su versión más reciente (2.0.20-4).
uwsgi-plugin-python3 ya está en su versión más reciente (2.0.20-4).
git ya está en su versión más reciente (1:2.34.1-1ubuntu1.11).
libpq-dev ya está en su versión más reciente (14.12-0ubuntu0.22.04.1).
```

Además es importante tener instalado en el equipo cliente autofirma 1.8 o superior:

Descargar → [aquí](#)



Paso 2: Descarga el contenido del repositorio y muévelo a su ruta específica.

- Clonar el Repositorio desde GitHub:

`git clone https://github.com/EnriqueSLsys/Django_PDF_Signature.git`

```
enrique@server:~$ git clone https://github.com/EnriqueSLsys/Django_PDF_Signature.git
Cloning into 'Django_PDF_Signature'...
remote: Enumerating objects: 101, done.
remote: Counting objects: 100% (101/101), done.
remote: Compressing objects: 100% (94/94), done.
remote: Total 101 (delta 25), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (101/101), 155.09 KiB | 1.80 MiB/s, done.
Resolving deltas: 100% (25/25), done.
```

- Crea el directorio `/var/www/html/PDjango`:

`sudo mkdir -p /var/www/html/PDjango`

```
enrique@server:~$ sudo mkdir -p /var/www/html/PDjango
```

- Mueve el archivo comprimido al directorio `/var/www/html/PDjango`:

`sudo mv Django_PDF_Signature /var/www/html/PDjango/`

```
enrique@server:~$ sudo mv Django_PDF_Signature/ /var/www/html/PDjango/
```



Paso 3: Instalación de Requisitos de la Aplicación Django Asegúrate de que estás dentro del directorio de la aplicación Django, por ejemplo, `Django_PDF_Signature`:

```
cd /var/www/html/PDjango/Django_PDF_Signature
```

```
enrique@server:/var/www/html/PDjango$ cd /var/www/html/PDjango/Django_PDF_Signature
enrique@server:/var/www/html/PDjango/Django_PDF_Signature$ |
```

- Instala los requisitos de la aplicación Django utilizando pip3:

```
sudo pip3 install -r requirements.txt
```

```
enrique@server:/var/www/html/PDjango/Django_PDF_Signature$ sudo pip3 install -r requirements.txt
Collecting Django==5.0.6
  Downloading Django-5.0.6-py3-none-any.whl (8.2 MB)
    8.2/8.2 MB 23.4 MB/s eta 0:00:00
Collecting psycopg2==2.9.9
  Downloading psycopg2-2.9.9.tar.gz (384 kB)
    384.9/384.9 KB 47.7 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting reportlab==3.6.0
  Downloading reportlab-3.6.0.tar.gz (4.5 MB)
    4.5/4.5 MB 6.5 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting babel==2.12.1
  Downloading Babel-2.12.1-py3-none-any.whl (10.1 MB)
    10.1/10.1 MB 25.1 MB/s eta 0:00:00
Collecting sqlparse>=0.3.1
  Downloading sqlparse-0.5.0-py3-none-any.whl (43 kB)
    44.0/44.0 KB 5.8 MB/s eta 0:00:00
Collecting asgiref<4,>=3.7.0
  Downloading asgiref-3.8.1-py3-none-any.whl (23 kB)
Collecting pillow>=4.0.0
  Downloading pillow-10.3.0-cp310-cp310-manylinux_2_28_x86_64.whl (4.5 MB)
    4.5/4.5 MB 18.7 MB/s eta 0:00:00
Collecting typing-extensions>=4
  Downloading typing_extensions-4.12.2-py3-none-any.whl (37 kB)
```




Paso 4: Configuración de PostgreSQL Cambia a la cuenta de usuario postgres:

```
sudo -i -u postgres
```

```
enrique@server:/var/www/html/PDjango$ sudo -i -u postgres
postgres@server:~$ |
```

- Accede a la consola de PostgreSQL:

```
psql
```

```
postgres@server:~$ psql
psql (14.12 (Ubuntu 14.12-0ubuntu0.22.04.1))
Type "help" for help.
```

- Cambia la contraseña del usuario postgres:

```
ALTER USER postgres PASSWORD 'usuario';
```

```
postgres=# ALTER USER postgres PASSWORD 'usuario'
```

- Crea la base de datos `forms_medinaazahara` y asigna permisos al usuario postgres:

```
CREATE DATABASE forms_medinaazahara;
```

```
postgres=# CREATE DATABASE forms_medinaazahara;
CREATE DATABASE
```

```
GRANT ALL PRIVILEGES ON DATABASE forms_medinaazahara TO postgres
```

```
postgres=# GRANT ALL PRIVILEGES ON DATABASE forms_medinaazahara TO postgres;
GRANT
```

- Sal de la consola de PostgreSQL:

```
\q
```

```
postgres=# \q
```



- Sal de la cuenta de usuario postgres:

`exit`

```
postgres@server:~$ exit
logout
```

- A continuación entra en el archivo de configuración de PostgreSQL, usa el comando `ctrl + w`, escribe [Connection Settings](#) y dale a enter.

`sudo nano /etc/postgresql/14/main/postgresql.conf`

```
GNU nano 6.2 /etc/postgresql/14/main/postgresql.conf
# If external_pid_file is not explicitly set, no extra PID file is written.
external_pid_file = '/var/run/postgresql/14-main.pid'           # write an extra PID file
                                                                # (change requires restart)

-----
# CONNECTIONS AND AUTHENTICATION
#-----

# - Connection Settings -
listen_addresses = '192.168.1.18'
```

- Ahora añade:

`listen_addresses = 'TU_IP'`

- Ahora añadir la siguiente línea al final del siguiente archivo [pg_hba.conf](#):

`sudo nano /etc/postgresql/14/main/pg_hba.conf`



- Al Final del archivo incluye:

host all all TU_IP/24 md5

```
GNU nano 6.2 /etc/postgresql/14/main/pg_hba.conf
# configuration parameter, or via the -i or -h command line switches.

# DO NOT DISABLE!
# If you change this first entry you will need to make sure that the
# database superuser can access the database using some other method.
# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket
local all postgres peer

# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all peer
# IPv4 local connections:
host all all 127.0.0.1/32 scram-sha-256
# IPv6 local connections:
host all all ::1/128 scram-sha-256
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all peer
host replication all 127.0.0.1/32 scram-sha-256
host replication all ::1/128 scram-sha-256
host all all 192.168.1.18/24 md5
```

- Recuerda cambiar TU_IP.

- Reiniciar postgresql:

sudo systemctl restart postgresql

```
enrique@server:/var/www/html/PDjango/Django_PDF_Signature$ sudo systemctl restart postgresql
```

Paso 5: Configuración de Nginx

Genera un certificado SSL autofirmado utilizando el siguiente comando:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/ssl-cert-snakeoil.key -out /etc/ssl/certs/ssl-cert-snakeoil.pem
```

[illegible]

- Crea un archivo de configuración para tu sitio web en Nginx, por ejemplo, `forms_medinaazahara.conf`, en el directorio `/etc/nginx/sites-available/`. Puedes usar el siguiente contenido, reemplazando `TU_IP` con la dirección IP de tu equipo:

```
sudo nano /etc/nginx/sites-available/forms_medinaazahara.conf
```



- Añadir dentro:

```
server {  
  
    listen 80;  
  
    server_name TU_IP;  
  
    return 301 https://$host$request_uri;  
  
}
```

```
server {  
  
    listen 443 ssl;  
  
    server_name TU_IP;  
  
  
    ssl_certificate /etc/ssl/certs/ssl-cert-snakeoil.pem;  
    ssl_certificate_key /etc/ssl/private/ssl-cert-snakeoil.key;  
  
  
    location / {  
  
        include uwsgi_params;  
  
        uwsgi_pass unix:/var/www/html/PDjango/Django_PDF_Signature/formproject.sock;  
  
        proxy_set_header Host $host;  
  
        proxy_set_header X-Real-IP $remote_addr;  
  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
  
        proxy_set_header X-Forwarded-Proto $scheme;  
  
        proxy_set_header Referer $http_referer;  
  
    }  
  
  
    location /static/ {
```



```
alias /var/www/html/PDjango/Django_PDF_Signature/f_solicitudes/static/;
```

```
}
```

```
}
```

```
server {
    listen 80;
    server_name TU_IP;
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name 192.168.1.18;

    ssl_certificate /etc/ssl/certs/ssl-cert-snakeoil.pem;
    ssl_certificate_key /etc/ssl/private/ssl-cert-snakeoil.key;

    location / {
        include uwsgi_params;
        uwsgi_pass unix:/var/www/html/PDjango/Django_PDF_Signature/formproject.sock;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header Referer $http_referer;
    }

    location /static/ {
        alias /var/www/html/PDjango/Django_PDF_Signature/f_solicitudes/static/;
    }
}
```

- Activa el sitio en Nginx:

```
sudo ln -s /etc/nginx/sites-available/forms_medinaazahara.conf /etc/nginx/sites-enabled/
```

```
enrique@server:/var/www/html/PDjango/Django_PDF_Signature$ sudo ln -s /etc/nginx/sites-available/
/forms_medinaazahara.conf /etc/nginx/sites-enabled/
```

- Reinicia Nginx:

```
sudo systemctl restart nginx
```

```
enrique@server:/var/www/html/PDjango/Django_PDF_Signature$ sudo systemctl restart nginx
enrique@server:/var/www/html/PDjango/Django_PDF_Signature$ |
```

Paso 6: Modificación de Configuraciones en settings.py En el archivo **settings.py**, tienes que modificar lo siguiente para adaptarlo a tu IP-Dominio tengas las siguientes modificaciones. Puedes buscar con el **ctrl + w** las líneas a cambiar:

```
sudo nano
```

```
/var/www/html/PDjango/Django_PDF_Signature/Django_PDF_Signature/settings.py
```

```
ALLOWED_HOSTS = ['localhost', '127.0.0.1', 'TU_IP']
```



```
CSRF_TRUSTED_ORIGINS = ['https://TU_IP']
```

```
ALLOWED_HOSTS = ['localhost', '127.0.0.1', '192.168.1.18']

# Configuraciones necesarias para que CSRF funcione correctamente con HTTPS en Nginx.
# CSRF_COOKIE_SECURE garantiza que las cookies CSRF se utilicen de manera segura.
# CSRF_TRUSTED_ORIGINS especifica los orígenes de confianza para CSRF, asegurando la autenticación adecuada.
CSRF_COOKIE_SECURE = False
CSRF_TRUSTED_ORIGINS = ['https://192.168.1.18']
```

```
DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.postgresql_psycopg2',

        'NAME': 'forms_medinaazahara',

        'USER': 'postgres',

        'PASSWORD': 'tu_nueva_contraseña',

        'HOST': 'TU_IP',

        'PORT': '5432',

    }

}
```

```
##Base de datos para Postgres
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'forms_medinaazahara',
        'USER': 'postgres',
        'PASSWORD': 'usuario',
        'HOST': '192.168.1.18',
        'PORT': '5432',
    }
}
```

Recuerda reemplazar **TU_IP** con la dirección IP de tu equipo.



Paso 7: Ahora hacemos las migraciones de la BD del proyecto

```
python3 /var/www/html/PDjango/Django_PDF_Signature/manage.py makemigrations
```

```
python3 /var/www/html/PDjango/Django_PDF_Signature/manage.py migrate
```

```
enrique@server:/var/www/html/PDjango$ python3 /var/www/html/PDjango/Django_PDF_Signature/manage.py makemigrations
python3 /var/www/html/PDjango/Django_PDF_Signature/manage.py migrate
No changes detected
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, f_solicitudes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying f_solicitudes.0001_initial... OK
  Applying sessions.0001_initial... OK
```

Paso 8: Iniciar uWSGI Después de configurar Nginx y uWSGI, puedes iniciar uWSGI con el siguiente comando:

```
sudo uwsgi --ini /var/www/html/PDjango/Django_PDF_Signature/uwsgi.ini --plugin python3
```




4.2.2 Instalación Automática

Para implementar la aplicación web automáticamente, puedes usar el siguiente script:

1. Clona el repositorio desde GitHub:

```
git clone https://github.com/EnriqueSLsys/Django_PDF_Signature.git
```

```
enrique@server:~$ git clone https://github.com/EnriqueSLsys/Django_PDF_Signature.git
Cloning into 'Django_PDF_Signature'...
remote: Enumerating objects: 124, done.
remote: Counting objects: 100% (124/124), done.
remote: Compressing objects: 100% (116/116), done.
remote: Total 124 (delta 35), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (124/124), 166.40 KiB | 1.81 MiB/s, done.
Resolving deltas: 100% (35/35), done.
```

2. Cambia al directorio del repositorio:

```
cd Django_PDF_Signature
```

```
enrique@server:~$ cd Django_PDF_Signature
```

3. Da permisos de ejecución al script de instalación:

```
chmod +x install.sh
```

```
enrique@server:~/Django_PDF_Signature$ chmod +x install.sh
```

4. Ejecuta el script de instalación:

```
sudo ./install.sh
```

```
enrique@server:~/Django_PDF_Signature$ sudo ./install.sh
^[[3~[sudo] password for enrique:
Sorry, try again.
[sudo] password for enrique:
Actualizando el gestor de paquetes e instalando dependencias...
Des:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Obj:2 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Des:3 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1.517 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Des:5 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [259 kB]
Des:6 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1.933 kB]
Des:7 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [329 kB]
Des:8 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [858 kB]
Des:9 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [166 kB]
Des:10 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37,2 kB]
```



Finalmente para comprobar su funcionamiento accede a cualquier navegador web y pon:

https://TU_IP

Al acceder darle enter te aparecerá la pantalla de inicio:



Para generar un PDF para posteriormente firmarlo debes darle al boton de inicio en el navegador y rellenar el formulario.

5. ANÁLISIS ECONÓMICO

Presupuesto Detallado

Concepto	Cantidad	Total
Desarrolladores	3 desarrolladores a 30€/hora, 400 horas de desarrollo	36,000
Diseñador UI/UX	1 diseñador a 25€/hora, 100 horas de diseño	2,500
Especialista en Seguridad	1 especialista a 40€/hora, 50 horas de asesoramiento	2,000
Servidores	1 servidor a 50€/mes, 12 meses	600
Dispositivos de Prueba	3 PCs a 500€ cada uno, 2 tablets a 300€ cada una	2,100
Licencias de Software	500€ anuales	500
Herramientas de Desarrollo	300€ anuales	300

Costo Total del Proyecto: 44,000€

Ayudas y Subvenciones: El proyecto puede beneficiarse de diversas ayudas y subvenciones para digitalización e innovación tecnológica.

Necesidades de Financiación: Si el costo total no se cubre con ayudas y subvenciones, se buscarán fuentes adicionales de financiación como préstamos o inversores privados.

Valoración Económica: Los beneficios a largo plazo incluyen:

- **Aumento de Eficiencia:** Reducción de tiempo y recursos para la gestión de documentos.
- **Mejora de Seguridad:** Protección de datos con firmas digitales seguras.
- **Satisfacción del Cliente:** Mejora de la experiencia del usuario.



- **Reducción de Costos Operativos:** Menores costos asociados a la gestión de documentos físicos.

6. SEGUIMIENTO Y CONTROL

Indicadores de Seguimiento:

- **Progreso del Desarrollo:** % de tareas completadas durante cada fase de trabajo.
- **Calidad del Código:** Número de bugs y cobertura de pruebas.

Frecuencia y Metodología:

- **Revisiones Semanales:** Reuniones para revisar el progreso del proyecto.
- **Pruebas Continuas:** Testing automático y manual durante todo el ciclo de desarrollo.

Responsables:

- **Líder de Proyecto:** Coordinación y seguimiento general.
- **Equipo de QA:** Realización de pruebas y documentación de problemas.

Acciones Correctivas:

- **Reuniones de Revisión:** Planificación de soluciones y acciones correctivas.
- **Ajustes de Planificación:** Reasignación de tareas y recursos.

Plan de Riesgos:

- **Identificación y Evaluación de Riesgos:** Listado y análisis de impacto.
- **Mitigación de Riesgos:** Definición de medidas preventivas y correctivas.
- **Seguimiento de Riesgos:** Monitoreo continuo y actualización del plan.

Procedimientos de Evaluación y Calidad:

- **Definición de Procedimientos de Evaluación:** Criterios y métricas de evaluación.
- **Documentación de Evaluación:** Registro de resultados y acciones correctivas.
- **Gestión de Cambios:** Procedimiento para gestionar cambios en recursos y actividades.

6.1 Identificación de Fases y Tareas

D

Debilidades

1. Limitaciones de presupuesto:

- El proyecto cuenta con un presupuesto ajustado, lo que puede limitar la capacidad de implementar todas las funcionalidades deseadas y realizar pruebas exhaustivas.

1. Dependencia de terceros (Autofirma):

- La integración con Autofirma es crucial para la funcionalidad de firma digital. Cualquier cambio o problema con esta herramienta puede afectar el proyecto.

A

Amenazas

1. Competencia de otras soluciones en el mercado:

- Existen otras soluciones de firma digital en el mercado que pueden ofrecer características similares o adicionales, lo que representa una competencia directa.

1. Cambios en la normativa:

- Las leyes y regulaciones sobre firma digital y protección de datos pueden cambiar, lo que podría requerir actualizaciones o cambios en la plataforma para cumplir con las nuevas normativas.

F

Fortalezas

1. Uso de tecnologías probadas (Django, Autofirma):

- La elección de Django como framework para el desarrollo y Autofirma para la firma digital garantiza el uso de tecnologías robustas y probadas.

1. Equipo de desarrollo capacitado:

- El proyecto cuenta con un equipo de desarrolladores, diseñadores y especialistas en seguridad con experiencia y habilidades adecuadas para llevar a cabo el proyecto con éxito.

O

Oportunidades

1. Aumento de la demanda de soluciones de firma digital:

- La creciente digitalización de procesos y el aumento del teletrabajo han incrementado la demanda de soluciones de firma digital, lo que representa una oportunidad de mercado.

1. Potencial de expansión a nuevos mercados:

- La plataforma tiene el potencial de expandirse a nuevos mercados y sectores, ofreciendo soluciones de firma digital adaptadas a diferentes necesidades y contextos.

7. FUENTES DE DOCUMENTACIÓN

Artículos Científicos sobre Firma Digital:

- Housley, R., & Polk, W. (2001). *Planning for PKI: Best practices guide for deploying public key infrastructure*. Wiley.

Documentación Oficial de Django y Autofirma:

- Django Documentation. (s.f.). Recuperado de <https://docs.djangoproject.com/en/5.0/>
- Portal Firma - Descargas. (s.f.). Recuperado de <https://firmaelectronica.gob.es/Home/Descargas.html>

Estudios de Mercado sobre Necesidades de Firma Digital:

- BeeDIGITAL. (2022, 8 de junio). Ventajas de la firma digital electrónica para las empresas. BeeDIGITAL. Recuperado de <https://www.beedigital.es/impulsa-negocio/ventajas-de-la-firma-digital-electronica-para-las-empresas/>

8. ANEXOS

8.1 Guía de Usuario:

- **Generar Documento PDF:**
 - **Acceso:** Diríjase a https://TU_IP.
 - **Rellenar Formulario:** Complete los campos del formulario en la sección "Generación de Documentos".
 - **Generar PDF:** Haga clic en "Generar PDF".
 - **Revisión:** Revise y edite el documento si es necesario.
- **Firmar Digitalmente Documento PDF:**
 - **Subir Documento:** Vaya a "Firma de Documentos" y cargue el PDF.
 - **Firma Digital:** Seleccione el área para la firma y use Autofirma.
 - **Descargue el Documento Firmado:** Descargue el documento firmado.

8.2 Referencias Adicionales:

- **Documentación Técnica:**
 - Django Documentation. (s.f.). Recuperado de <https://docs.djangoproject.com/en/5.0/>



- Portal Firma - Descargas. (s.f.). Recuperado de <https://firmaelectronica.gob.es/Home/Descargas.html>
- Cliente de Firma Electrónica de @Firma. (2010). Recuperado de <https://administracionelectronica.gob.es/ctt/clientefirma/descargas>

8.3 Otros Materiales:

- **Manual de Configuración y Administración:**
 - **Configuración Inicial:**
 - Instrucciones para instalar y configurar el servidor, PostgreSQL y Nginx:
 - Basic Installation | Ubuntu. (s.f.). Recuperado de <https://ubuntu.com/server/docs/basic-installation>
 - Uso de Python En Plataformas Unix. (s.f.). Recuperado de <https://docs.python.org/es/3/using/unix.html>
 - PostgreSQL: Documentation. (s.f.-b). Recuperado de <https://www.postgresql.org/docs/>
 - **Instalación de Django y Autofirma:**
 - Django Documentation. (s.f.). Recuperado de <https://docs.djangoproject.com/en/5.0/>
 - Portal Firma - Descargas. (s.f.). Recuperado de <https://firmaelectronica.gob.es/Home/Descargas.html>
 - **Mantenimiento del Sistema:**
 - Procedimientos de actualización y solución de problemas:
 - Actualizar Ubuntu. (s.f.). Recuperado de <https://www.softzone.es/linux/distros/actualizar-ubuntu/>
 - Migrations | Django Documentation. (s.f.). Recuperado de <https://docs.djangoproject.com/en/5.0/topics/migrations/>
 - The uWSGI Project — uWSGI 2.0 Documentation. (s.f.). Recuperado de <https://uwsgi-docs.readthedocs.io/en/latest/>
 - Chapter 26 PostgreSQL. Backup And Restore. (2024). Recuperado de <https://www.postgresql.org/docs/current/backup.html>
 - Pg_Dump. (2024). Recuperado de <https://www.postgresql.org/docs/current/app-pgdump.html>



- **Escalabilidad y Rendimiento:**
 - Optimización de Rendimiento:
 - Chapter 14 PostgreSQL. Performance Tips. (2024).
Recuperado de
<https://www.postgresql.org/docs/current/performance-tips.html>
- **Repositorio del Proyecto:**
 - EnriqueSLsys. (s.f.). *Django_PDF_Signature Repository*. GitHub.
Recuperado de
https://github.com/EnriqueSLsys/Django_PDF_Signature
- **Otros:**
 - ChatGPT. (s.f.).
<https://chatgpt.com/>

8.4 Posibles Mejoras

En esta sección, propongo varias mejoras para ampliar y optimizar el proyecto de generación de documentos PDF, con un enfoque en la usabilidad, accesibilidad y seguridad. Estas mejoras no solo incrementarían la funcionalidad del sistema, sino que también mejorarían la experiencia del usuario y la seguridad de los datos manejados.

1. Formularios de Inscripción a un Centro

Actualmente, el proyecto genera documentos PDF simples. Propongo una mejora significativa que permitiría la creación de formularios de inscripción a centros educativos. Esta funcionalidad permitiría a los usuarios llenar datos específicos y generar un documento formal de inscripción.

- **Función a Mejorar:** La función de generación de PDF debe adaptarse para trabajar con plantillas de inscripción y rellenar automáticamente los datos proporcionados por el usuario.
- **Pasos para Implementar:**
 1. **Identificación de Datos Necesarios:** Determinar los campos necesarios para la inscripción (nombre del estudiante, fecha de nacimiento, curso, etc.).
 2. **Creación de una Plantilla de PDF:** Diseñar una plantilla de PDF que incluya estos campos, utilizando herramientas como ReportLab o un editor de PDF.



3. **Modificación de la Función de Generación de PDF:** Ajustar la lógica de generación de PDF para que use la plantilla y llene los campos con los datos del formulario.
4. **Interfaz de Usuario:** Adaptar el formulario de la interfaz web para recoger los datos necesarios para la inscripción.

2. Plantillas Avanzadas de PDF

Propuesta para incluir plantillas predefinidas que los usuarios puedan seleccionar según sus necesidades, mejorando así la flexibilidad y utilidad de la generación de PDFs.

- **Función a Mejorar:** Implementar una función de selección de plantillas.
- **Pasos para Implementar:**
 1. Diseñar varias plantillas de PDF para diferentes propósitos.
 2. Añadir una interfaz de usuario que permita seleccionar y personalizar las plantillas.

3. Acceso Global a través de DDNS

Para hacer el proyecto accesible desde cualquier lugar, propongo configurar un servicio de Dynamic DNS (DDNS) como NOIP. Esto permitiría asignar un dominio a la IP pública del servidor, facilitando el acceso remoto.

- **Pasos para Implementar:**
 1. Registrar un dominio en un servicio DDNS como NOIP.
 2. Configurar el servidor para actualizar la IP en el servicio DDNS automáticamente.

4. Completar el Backend de Inicio y Registro de Usuarios

Actualmente, el sistema cuenta con una interfaz básica de inicio y registro de usuarios. Propongo mejorar y completar esta funcionalidad para asegurar una gestión de usuarios más robusta y segura.

- **Función a Mejorar:** Completar las funciones backend para manejar inicio de sesión, registro, recuperación de contraseñas y gestión de perfiles de usuario.
- **Pasos para Implementar:**
 1. **Inicio de Sesión y Registro:** Asegurar que las rutas de inicio de sesión y registro funcionen correctamente con validaciones de formularios.
 2. **Recuperación de Contraseña:** Implementar la funcionalidad de recuperación de contraseña vía email.



3. **Gestión de Perfiles:** Añadir la opción para que los usuarios editen sus perfiles y cambien su información personal.
4. **Seguridad:** Asegurar todas las rutas y datos sensibles utilizando prácticas de seguridad recomendadas (HTTPS, hashing de contraseñas, etc.).

5. Gestión de Documentos por Usuario

Para mejorar la usabilidad y la seguridad, propongo implementar un sistema que permita a cada usuario ver y gestionar sus propios documentos PDF.

- **Función a Mejorar:** Crear un sistema de gestión de documentos vinculado a cada cuenta de usuario.
- **Pasos para Implementar:**
 1. **Modelos de Base de Datos:** Añadir un modelo de base de datos que relacione usuarios con sus documentos.
 2. **Autorización y Autenticación:** Asegurar que cada usuario solo pueda acceder a sus propios documentos.
 3. **Interfaz de Usuario:** Crear una sección en el perfil del usuario donde pueda ver, descargar y gestionar sus documentos.