


# Definición y Alcance de Proyecto - Python Developer

## Objetivos

- ✓ **Planificación** (Definición de requerimientos, arquitectura)
  - ✓ **Desarrollo** (Backend, frontend, base de datos, API, seguridad, pruebas)
  - ✓ **Despliegue** (CI/CD, hosting, monitoreo)
- 

## Proyecto: "Plataforma de Gestión de Proyectos y Tareas"

 **Descripción:** Una aplicación web donde los equipos pueden gestionar proyectos, asignar tareas, hacer seguimiento del progreso y colaborar en tiempo real.

### ♦ Funcionalidades principales

- 1. Autenticación y gestión de usuarios**
    - Registro e inicio de sesión con JWT/OAuth2
    - Perfiles de usuario con roles (admin, usuario normal)
  - 2. Gestión de proyectos y tareas**
    - Crear, editar y eliminar proyectos
    - Asignar tareas a miembros del equipo
    - Estado de las tareas (pendiente, en progreso, finalizada)
    - Adjuntar archivos a tareas
  - 3. Dashboard con estadísticas**
    - Visualización de tareas completadas vs. pendientes
    - Gráficos con productividad por usuario/proyecto
  - 4. Chat en tiempo real** (Opcional)
    - Integración con WebSockets para comunicación en equipos
  - 5. Notificaciones**
    - Notificaciones por email o en la plataforma sobre cambios en tareas
  - 6. Seguridad y permisos**
    - Control de acceso basado en roles (RBAC)
    - Protección contra ataques comunes (CSRF, SQL Injection, etc.)
  - 7. Despliegue y monitoreo**
    - Docker + CI/CD para despliegue automático
    - Logs y monitoreo con herramientas como Prometheus/Grafana
- 

## Tecnologías a utilizar

### Backend:

- ♦ Python (Flask/Pyramid)
- ♦ PostgreSQL
- ♦ SQLAlchemy
- ♦ Celery + Redis (para tareas asíncronas, ej. notificaciones)

## Frontend:

- ♦ JavaScript con Bootstrap
- ♦ Consumo de API REST

## Infraestructura y despliegue:

- ♦ Docker + Docker Compose
  - ♦ GitHub Actions (para CI/CD)
  - ♦ AWS para hosting (opcional) o VPS en la nube o en railway
  - ♦ Nginx como proxy reverso
- 

## 17 **Ciclo de Vida del Proyecto**

### Fase 1: Planificación y Diseño (30h)

- Definir requerimientos
- Diseñar arquitectura y modelo de datos
- Wireframes del frontend

### Fase 2: Desarrollo Backend (70h)

- Configurar Flask/Pyramid y base de datos
- Implementar autenticación y API REST
- CRUD de proyectos y tareas
- Notificaciones y tareas asíncronas

### Fase 3: Desarrollo Frontend (50h)

- Configurar JavaScript con Bootstrap
- Componentes de UI (formularios, dashboard, chat)
- Integración con la API

### Fase 4: Pruebas y Seguridad (25h)

- Unit tests y pruebas de integración
- Seguridad en API y frontend

### Fase 5: Despliegue y Monitoreo (25h)

- Configuración Docker y CI/CD
  - Hosting en la nube
  - Configurar logs y monitoreo
- 

## **Entregables del Proyecto**

### 1 **Documentación Scrum (GitBook)**

- Product Backlog
- Sprint Backlog
- Historias de usuario
- Definición de Hecho (DoD)
- Actas de reuniones (Daily Scrum, Sprint Planning, Sprint Review, Sprint Retrospective)

- Reportes de avance

## 2 Código Fuente (GitHub)

- Repositorio en Git con versiones del código
- Código limpio y documentado
- Pruebas unitarias e integración
- Pipeline CI/CD configurado
- Despliegues automatizados

## 3 Manuales y Documentación Técnica (GitBook)

- **Manual Técnico** (Arquitectura del sistema, configuraciones, API docs)
- **Manual de Usuario** (Guía paso a paso para usuarios finales)
- **Guía de instalación y configuración** (Ambientación)
- **Matriz de pruebas y casos de prueba** (Allure Report/TestLink)

---

### ♦ Beneficios de este proyecto:

- ✓ Cobertura de **todo el ciclo de desarrollo**
- ✓ Usa tecnologías modernas
- ✓ Escalable y con buenas prácticas
- ✓ Se puede extender con nuevas funcionalidades

---

## Marco de Trabajo Scrum



- ♦ **Stakeholders** (Product Owner, Scrum Master, Scrum Developer)
- ♦ **Artefactos de Scrum** (Product Backlog, Sprint Backlog)
- ♦ **Eventos en Scrum** (Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective)
- ♦ **Herramientas recomendadas para Scrum** (Trello/Jira, GitHub, Slack, Zoom)

---

## Estimación aproximada de esfuerzo:

200 horas. 

## Desglose Recomendado (opcional)

Fase 	Horas Estimadas 
--	---

<b>Fase 1: Planificación y Diseño</b> <a href="#">↗</a>	<b>30h</b> <a href="#">↗</a>
• Definir requerimientos funcionales y no funcionales	<b>8h</b> <a href="#">↗</a>
• Diseñar arquitectura del sistema	<b>8h</b> <a href="#">↗</a>
• Modelado de base de datos	<b>6h</b> <a href="#">↗</a>
• Wireframes y prototipos del frontend	<b>8h</b> <a href="#">↗</a>
<b>Fase 2: Desarrollo Backend</b> <a href="#">↗</a>	<b>70h</b> <a href="#">↗</a>
• Configurar entorno de desarrollo	<b>6h</b> <a href="#">↗</a>
• Implementar autenticación y gestión de usuarios	<b>10h</b> <a href="#">↗</a>
• CRUD de proyectos y tareas	<b>15h</b> <a href="#">↗</a>
• Implementar control de permisos y roles	<b>8h</b> <a href="#">↗</a>
• Implementar sistema de notificaciones	<b>10h</b> <a href="#">↗</a>
• Implementar tareas asíncronas	<b>10h</b> <a href="#">↗</a>
• Optimización y validación de API	<b>6h</b> <a href="#">↗</a>
<b>Fase 3: Desarrollo Frontend</b> <a href="#">↗</a>	<b>50h</b> <a href="#">↗</a>
• Configurar entorno frontend	<b>5h</b> <a href="#">↗</a>
• Desarrollo de componentes de UI	<b>15h</b> <a href="#">↗</a>
• Implementación del tablero de tareas	<b>10h</b> <a href="#">↗</a>
• Integración con la API REST	<b>10h</b> <a href="#">↗</a>
• Implementación de chat interno	<b>10h</b> <a href="#">↗</a>
<b>Fase 4: Pruebas y Seguridad</b> <a href="#">↗</a>	<b>25h</b> <a href="#">↗</a>
• Implementación de pruebas unitarias y de integración	<b>10h</b> <a href="#">↗</a>
• Validación de seguridad en API y frontend	<b>10h</b> <a href="#">↗</a>
• Pruebas de carga y estrés	<b>5h</b> <a href="#">↗</a>
<b>Fase 5: Despliegue y Monitoreo</b> <a href="#">↗</a>	<b>25h</b> <a href="#">↗</a>
• Configurar contenedores con Docker	<b>5h</b> <a href="#">↗</a>
• Implementar CI/CD	<b>5h</b> <a href="#">↗</a>
• Despliegue en la nube	<b>5h</b> <a href="#">↗</a>
• Configurar logs y monitoreo	<b>5h</b> <a href="#">↗</a>
• Documentación final del proyecto	<b>5h</b> <a href="#">↗</a>
<b>Total de Horas Estimadas</b> <a href="#">↗</a>	<b>200h</b> <a href="#">↗</a>