

## Objetivo general:

Desarrollar una plataforma web llamada Workspace que funcione como un intermediario digital eficiente entre empresas y usuarios que buscan proyectos temporales, permitiendo la publicación, búsqueda y gestión de oportunidades laborales de corta duración sin necesidad de establecer relaciones contractuales formales. El objetivo es ofrecer un sistema accesible, intuitivo y seguro que facilite la conexión entre oferta y demanda de proyectos, mejore la visibilidad de oportunidades para los usuarios, optimice los procesos de búsqueda para las empresas y garantice una experiencia confiable mediante herramientas de comunicación, perfiles profesionales y filtros de coincidencia basados en habilidades, duración y tipo de actividad.

Objetivos específicos:

Objetivos técnicos

1. Diseñar e implementar una interfaz web intuitiva y accesible que permita a los usuarios registrarse, buscar proyectos y postularse de manera sencilla.
2. Desarrollar el backend del sistema utilizando lenguajes y tecnologías apropiadas para garantizar eficiencia, escalabilidad y seguridad en el manejo de datos.
3. Configurar una base de datos que almacene de forma segura información de usuarios, empresas y proyectos publicados.
4. Integrar mecanismos de autenticación y verificación para garantizar la protección de datos y la confiabilidad de los perfiles dentro de la plataforma.
5. Implementar servicios de almacenamiento y despliegue web en proveedores de hosting compatibles con las tecnologías empleadas.

Objetivos funcionales

6. Permitir que las empresas publiquen proyectos temporales con descripciones detalladas, requisitos y tiempo estimado.
7. Brindar a los usuarios herramientas de filtrado y búsqueda avanzada para encontrar oportunidades acordes a sus habilidades y disponibilidad.

8. Facilitar un sistema de comunicación interna entre usuarios y empresas para aclarar dudas y gestionar acuerdos.
9. Integrar un sistema de reputación o evaluación que permita verificar el desempeño de los usuarios en proyectos anteriores.

#### Objetivos administrativos y operativos

10. Establecer un modelo de costos basado en técnicas de estimación paramétrica que permita proyectar el costo operativo mensual del servicio.
11. Analizar la viabilidad económica de Workspace mediante el cálculo de costos e ingresos potenciales según el número de usuarios y proyectos mensuales.
12. Definir políticas de uso, privacidad y términos del servicio para garantizar el cumplimiento de parámetros legales y éticos.

#### Objetivos de impacto y alcance

13. Fomentar el acceso a oportunidades laborales flexibles para estudiantes, trabajadores independientes y profesionales en general.
14. Reducir el tiempo y complejidad del proceso de conexión entre empresas y colaboradores temporales.
15. Crear un espacio digital confiable que impulse la colaboración, el trabajo temporal y la economía de proyectos.

#### Requisitos Funcionales:

##### RF1. Registro de usuarios

El sistema deberá permitir que los usuarios (freelancers y empresas) puedan registrarse mediante correo electrónico y contraseña.

##### RF2. Inicio de sesión

El sistema permitirá que los usuarios ingresen con sus credenciales para acceder a

sus perfiles y funciones.

#### RF3. Recuperación de contraseña

El usuario podrá recuperar su contraseña mediante correo electrónico.

#### RF4. Creación y gestión de perfiles

Los usuarios podrán crear y editar su perfil, incluyendo datos como nombre, habilidades, áreas profesionales e información de contacto.

#### RF5. Publicación de proyectos por parte de empresas

Las empresas podrán crear publicaciones donde especifiquen:

- nombre del proyecto,
- duración,
- pago (si aplica),
- tipo de actividad,
- habilidades requeridas.

#### RF6. Búsqueda de proyectos por parte de usuarios

Los usuarios podrán buscar proyectos mediante filtros como duración, categoría, habilidades requeridas o rango de pago.

#### RF7. Sistema de aplicación a proyectos

Los usuarios podrán enviar su solicitud a una vacante temporal o proyecto publicado.

#### RF8. Panel administrativo básico

Los administradores podrán gestionar usuarios, aprobar registros y revisar publicaciones sospechosas.

#### RF10. Visualización del historial de proyectos

Los usuarios podrán visualizar los proyectos a los que han aplicado o en los que han participado.

#### RF11. Notificaciones

El sistema enviará notificaciones sobre:

- nuevas publicaciones,
- respuesta a solicitudes,
- mensajes nuevos.

#### Requisitos no Funcionales:

##### RNF1. Escalabilidad

El sistema deberá ser capaz de soportar un crecimiento progresivo de usuarios sin degradar el rendimiento.

##### RNF2. Usabilidad

La interfaz deberá ser intuitiva, clara y fácil de usar para personas sin conocimientos técnicos.

##### RNF3. Disponibilidad

El sistema deberá estar disponible el 99% del tiempo, considerando posibles fallos de hosting o mantenimiento.

##### RNF4. Seguridad de datos

Los datos personales deberán encriptarse y mantenerse bajo políticas de privacidad que protejan la información del usuario.

##### RNF5. Rendimiento

Las páginas deberán cargar en un tiempo promedio menor a 3 segundos bajo condiciones normales de conexión.

#### RNF6. Compatibilidad multiplataforma

La plataforma deberá funcionar correctamente en:

- navegadores web modernos,
- sistemas Windows, Linux y macOS,
- dispositivos móviles mediante diseño responsive.

#### RNF7. Mantenibilidad

El código deberá desarrollarse siguiendo buenas prácticas para facilitar futuras actualizaciones.

#### RNF8. Confiabilidad

El sistema deberá manejar adecuadamente errores como solicitudes incompletas, formularios inválidos o caídas parciales del servicio.

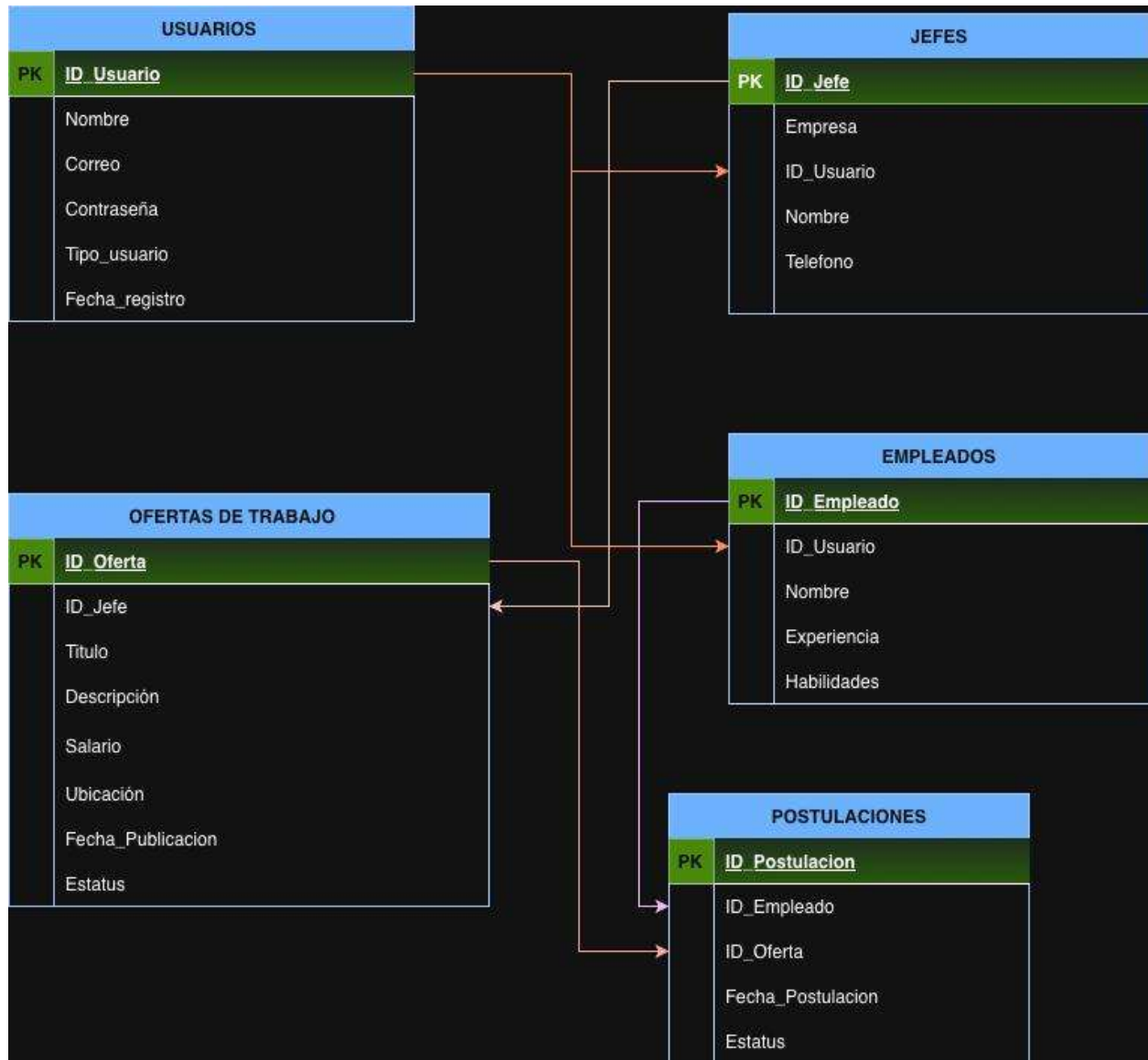
#### RNF9. Protección ante fallos externos

El sistema deberá soportar y recuperarse de fallos como:

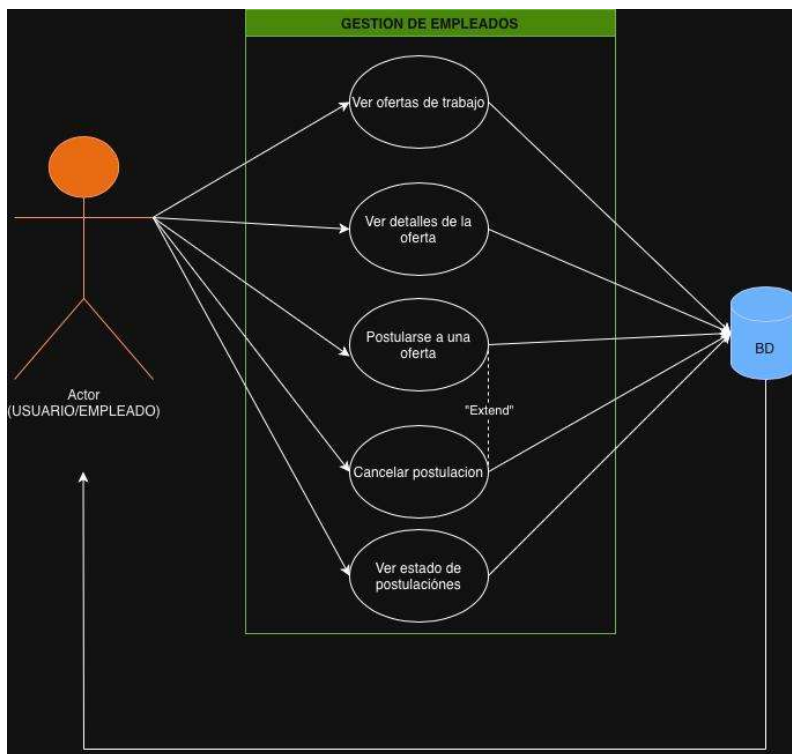
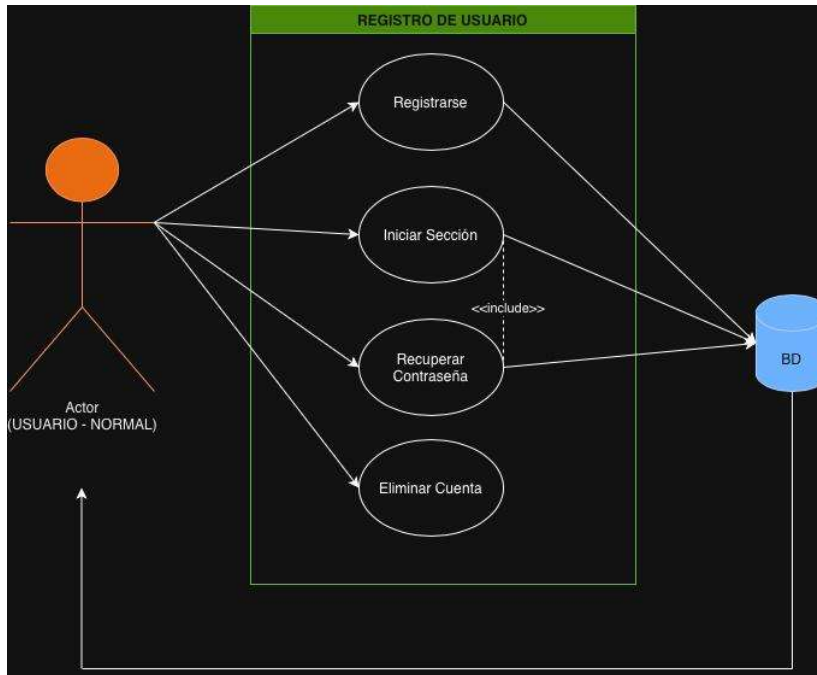
- caída del proveedor cloud,
- interrupción temporal del internet del usuario,
- sobrecarga de tráfico.

# B) MODELO LOGICO

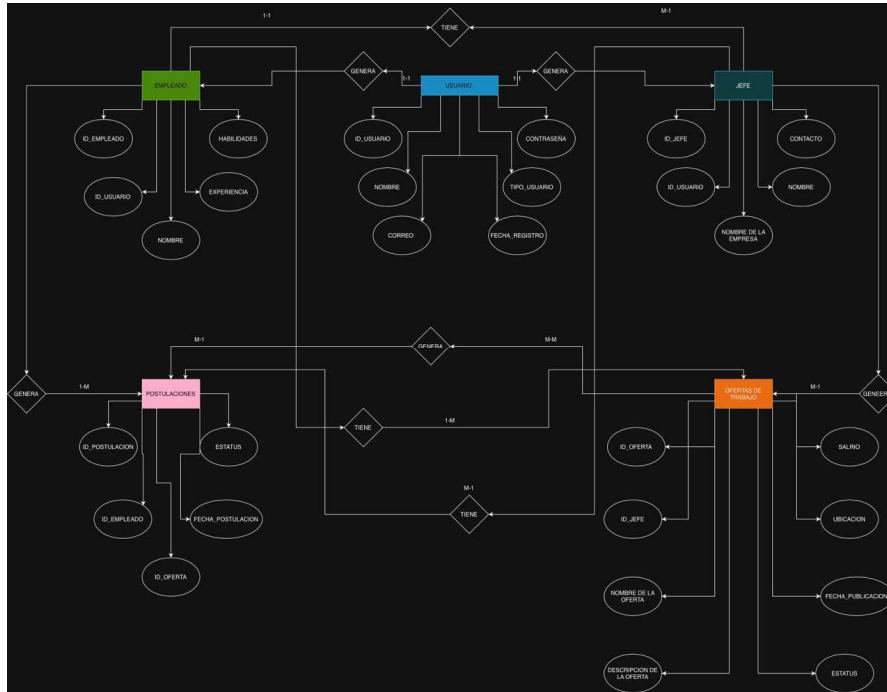
## MODELO LOGICO



## DIAGRAMA DE CASO DE USO



## DIAGRAMA ENTIDAD RELACION



## C) IMPLEMENTACIÓN DEL SISTEMA WORKSPACE

### 1. Introducción

La fase de implementación de **WORKSPACE** consistió en la transformación de los requisitos y diseños previos en un sistema de software funcional. Se utilizó una arquitectura Modelo-Vista-Controlador (MVC), aprovechando la robustez de **Python** y **Flask** para el *backend*, y la flexibilidad de **PostgreSQL** para la persistencia de datos. El desarrollo se centró en la seguridad, la escalabilidad y la experiencia de usuario, implementando roles diferenciados y flujos de trabajo estrictos.

### 2. Configuración del Entorno de Desarrollo

Para garantizar la reproducibilidad y el aislamiento de dependencias, se configuró el siguiente entorno técnico:

- **Lenguaje:** Python 3.13.
- **Gestor de Paquetes:** pip.



- **Dependencias Clave (requirements.txt):**
  - Flask: Framework web principal.
  - Flask-SQLAlchemy: ORM para manejo de base de datos.
  - Flask-Login: Gestión de sesiones de usuario.
  - psycopg2-binary: Conector para PostgreSQL.
  - Werkzeug: Utilidades de seguridad (*hashing*).
  - gunicorn: Servidor de aplicación para despliegue en Render.

### 3. Desarrollo del Backend (Lógica del Servidor)

El núcleo del sistema reside en `app.py`.

#### 3.1. Sistema de Autenticación

Se utilizó `Werkzeug.security` para no almacenar contraseñas en texto plano.

- **Registro:** Se genera un *hash* (`generate_password_hash`) antes de guardar en la BD.
- **Login:** Se compara el *hash* almacenado con la entrada del usuario (`check_password_hash`).

#### 3.2. Lógica de Negocio (Rutas y Flujos Clave)

Se desarrollaron controladores específicos para cada flujo, asegurando la integridad transaccional:

- **Publicación de Ofertas:** Validación de formularios y creación de registros en la tabla `job_offers`.
- **Postulación:** Verificación de duplicidad (un Worker no puede postularse dos veces al mismo proyecto) e inserción en la tabla `applications`.
- **Gestión de Estados:** Lógica para transicionar una solicitud de `pending` a `accepted`, `rejected` o `completed`. Se implementó el cierre automático de la oferta (`job.status = 'closed'`) al aceptar un candidato, eliminando la publicación del *marketplace* público.
- **Protección de Rutas:** Se implementaron los decoradores `@login_required`, `@boss_required` y `@worker_required` para aplicar seguridad a nivel de rol, evitando accesos no autorizados.

### 4. Implementación de la Base de Datos (ORM)

Se utilizó **SQLAlchemy** en el archivo `models.py` para mapear las clases de Python a tablas relacionales.

## 4.1. Estructura de Datos (Modelo Relacional)

Se implementó un modelo relacional normalizado:

- User: Tabla padre que contiene credenciales y el discriminador `user_type`.
- Boss / Employee: Tablas hijas vinculadas por `user_id` (Relación 1:1).
- JobOffer: Tabla de ofertas vinculada al Boss (Relación 1:N).
- Application: Tabla intermedia que vincula Employee y JobOffer (Relación N:M).

## 4.2. Integridad de Datos y Migración

Se añadieron restricciones para garantizar la coherencia:

- UniqueConstraint: Evita postulaciones duplicadas.
- CheckConstraint: Asegura que el campo `status` solo contenga valores válidos (`pending`, `accepted`, `rejected`, `completed`).
- **Despliegue Inicial (Solución Técnica):** Debido a las limitaciones de Render, la creación inicial y la migración de la base de datos se automatizó mediante un *script* (`fix_db.py`) que fue ejecutado a través del **Start Command** de Gunicorn, asegurando que el *CheckConstraint* incluyera el estado `'completed'`.

## 5. Desarrollo del Frontend (Interfaz de Usuario)

La implementación visual se realizó utilizando Jinja2 (motor de plantillas de Flask) y CSS3.

### 5.1. Diseño Adaptativo y Tematización

Se implementó un sistema visual basado en colores para mejorar la usabilidad y diferenciar los roles:

- **Tema Naranja:** Aplicado dinámicamente a las vistas del **Boss**.
- **Tema Azul:** Aplicado dinámicamente a las vistas del **Worker**.

Esto se logró mediante lógica condicional en las plantillas Jinja2 y CSS.

### 5.2. Feedback al Usuario

Se implementó el sistema de *flash messages* de Flask para notificar al usuario sobre el éxito o fracaso de sus acciones (ej. "Postulación enviada con éxito", "Credenciales incorrectas") sin necesidad de recargar la página completamente.

## 6. Pruebas Funcionales y Depuración (Ejecución Vía

## CMD)

Durante la implementación, la validación del sistema se realizó mediante pruebas funcionales de extremo a extremo, ejecutando el servidor de desarrollo directamente desde la **línea de comandos (CMD)** con `python app.py` y simulando las interacciones de los usuarios en el navegador.

Tipo de Prueba	Descripción del Procedimiento (Simulado en CMD/Navegador)	Validación de Éxito
Prueba de Roles y Redirección	Intentos de acceso directo a /perfilb siendo Worker, y acceso a /login estando ya autenticado.	El sistema forzó la redirección de los usuarios a su perfil correcto o a la página de inicio de sesión.
Prueba de Integridad Referencial	Simulación de la eliminación de un usuario Boss para verificar la respuesta de sus proyectos.	Los proyectos (job_offers) del Boss eliminado fueron gestionados según las reglas relacionales (ON DELETE SET NULL).
Prueba de Restricciones (Duplicidad)	Intento de postular dos veces al mismo proyecto con el mismo Worker.	El sistema lanzó el IntegrityError (manejado en app.py) y mostró el mensaje amigable "Ya te has postulado".
Prueba de Flujo de Estados	Se validó el ciclo completo: Creación (Boss) → Postulación (Worker) → Aceptación (Boss) → Finalización (Worker) [cite: app.py].	Se confirmó que el proyecto desapareciera de la búsqueda pública al ser aceptado, y que el estado final completed se guardara sin error, validando la corrección del <i>CheckConstraint</i> .

## 7. Conclusión de la Implementación

La implementación de **WORKSPACE** resultó en una aplicación web robusta y segura. La separación de responsabilidades entre el *Backend* (lógica y seguridad) y el *Frontend* (experiencia de usuario) permitió crear un sistema escalable. El uso de tecnologías estándar de la industria (Flask, PostgreSQL) y la superación de los retos de despliegue en la nube aseguran que el proyecto cumple con los requisitos de calidad esperados para un entorno de producción.

## d) Gestión del Proyecto

### Cronograma de Actividades (Diagrama de Gantt)

El desarrollo del proyecto **WORKSPACE** se ha planificado y ejecutado en un periodo de 8 semanas, siguiendo una metodología de desarrollo iterativa. A continuación, se detalla el cronograma de actividades dividido por fases.

Tabla de Planificación Temporal

ID	Fase	Tarea / Actividad	Duración	Semanas
1	Análisis y Requisitos	Definición del Alcance	1 semana	Semana 1
1.1		Identificación de roles (Boss/Worker) y Requisitos Funcionales	3 días	S1
1.2		Especificación	2 días	S1

		de Requisitos No Funcionales		
<b>2</b>	<b>Diseño del Sistema</b>	<b>Modelado de Datos y Arquitectura</b>	<b>1 semana</b>	<b>Semana 2</b>
2.1		Diseño del Diagrama Entidad-Relación (BD)	2 días	S2
2.2		Diseño de Diagramas de Casos de Uso y Flujo	3 días	S2
2.3		Diseño de Interfaz de Usuario (Mockups/HT ML Estático)	2 días	S2
<b>3</b>	<b>Implementación Backend</b>	<b>Desarrollo del Servidor (Flask)</b>	<b>2 semanas</b>	<b>Semana 3-4</b>
3.1		Configuración del entorno (Virtualenv, Config.py)	1 día	S3
3.2		Implementación de Modelos BD (SQLAlchemy)	3 días	S3
3.3		Desarrollo del sistema de Autenticación	3 días	S3

		(Login/Registro)		
3.4		Lógica de negocio: Rutas para Boss (Crear proyectos)	3 días	S4
3.5		Lógica de negocio: Rutas para Worker (Postulaciones)	2 días	S4
<b>4</b>	<b>Implementación Frontend</b>	<b>Integración y Vistas</b>	<b>2 semanas</b>	<b>Semana 5-6</b>
4.1		Desarrollo de Plantillas Base (Jinja2) y Estilos CSS	3 días	S5
4.2		Integración de formularios de Registro y Login	2 días	S5
4.3		Vistas de Perfiles (Boss/Worker) y Dashboards	3 días	S6
4.4		Vistas de Gestión: Detalles, Aceptar/Rechazar solicitudes	2 días	S6
<b>5</b>	<b>Pruebas y Depuración</b>	<b>Control de Calidad (QA)</b>	<b>1 semana</b>	<b>Semana 7</b>

5.1		Pruebas Unitarias (Modelos y Base de Datos)	2 días	S7
5.2		Pruebas de Integración (Flujo completo Boss -> Worker)	2 días	S7
5.3		Corrección de errores (Bugs de redirección, Constraints BD)	3 días	S7
<b>6</b>	<b>Despliegue y Cierre</b>	<b>Puesta en Producción</b>	<b>1 semana</b>	<b>Semana 8</b>
6.1		Configuración de Servidor (Gunicorn, Requirements.txt)	1 día	S8
6.2		Despliegue en Render.com y Configuración de PostgreSQL	1 día	S8
6.3		Elaboración de Manuales y Documentación Final	3 días	S8

# e) Documentación del Proyecto

## WORKSPACE

### 1. Manual de Usuario

#### 1.1. Introducción

**WORKSPACE** es una plataforma web diseñada para facilitar la conexión entre profesionales ("Workers") y gestores de proyectos ("Bosses"). Este manual guía al usuario a través de las funcionalidades principales del sistema, desde el registro hasta la finalización de un trabajo.

#### 1.2. Acceso y Registro

El sistema cuenta con un sistema de autenticación seguro.

##### 1. Ingreso al Sistema:

- Acceda a la página principal (Index).
- Haga clic en el botón "Login" o "Iniciar Sesión".
- Ingrese su correo electrónico y contraseña.

##### 2. Registro de Nuevo Usuario:

- En la pantalla de Login, seleccione "¿No tienes cuenta? Crear cuenta".
- **Selección de Rol:** Se le presentará una pantalla para elegir su perfil:
  - **Worker:** Para quienes buscan trabajo.



- **Boss:** Para quienes ofrecen proyectos.
- Complete el formulario correspondiente con sus datos personales y credenciales.

### 1.3. Perfil Boss (Jefe de Proyecto)

Al iniciar sesión como Boss, tendrá acceso a las siguientes herramientas:

- **Panel de Control (Perfil):** Visualización de sus datos de contacto y empresa.
- **Crear Proyecto:**
  1. Navegue a la opción "Crear Proyecto".
  2. Complete el formulario con el Título, Descripción, Ubicación y Presupuesto.
  3. Al guardar, el proyecto será visible inmediatamente para todos los Workers.
- **Gestión de Proyectos:**
  1. En "Mis Proyectos", verá una lista de sus ofertas publicadas.
  2. Haga clic en "**Ver / Administrar**" para ver los detalles.
  3. **Aceptar/Rechazar Candidatos:** En la vista de detalle, verá la lista de Workers que se han postulado. Puede "Aceptar" para contratar o "Rechazar" la solicitud.

### 1.4. Perfil Worker (Trabajador)

Al iniciar sesión como Worker, podrá realizar las siguientes acciones:

- **Buscar Proyectos:**
  1. Acceda a la sección "Buscar Proyectos".
  2. Verá todas las ofertas disponibles con estado "Abierto".
  3. Haga clic en "**Postularme**" para enviar su interés al Boss.
- **Mis Solicitudes:**
  - Permite ver el estado de sus postulaciones (Pendiente, Aceptado o Rechazado).
- **Trabajos Pendientes:**
  - Aquí aparecen únicamente los proyectos donde ha sido **Aceptado**.
  - Esta sección se divide en:
    - **En Curso:** Trabajos activos. Puede ver detalles y contactar al cliente.
    - **Historial:** Trabajos que usted ha marcado como "Completados".
- **Finalizar Trabajo:** Dentro de los detalles de un trabajo en curso, utilice el botón "Marcar como Completado" para notificar la finalización.

# 2. Manual del Sistema (Técnico)

## 2.1. Descripción General

WORKSPACE

Quiénes Somos

Por qué elegirnos

Login

Registro

### ¿Quiénes Somos?

WORKSPACE nació con la misión de conectar personas talentosas y facilitar la colaboración profesional en un entorno digital moderno y accesible. Creemos que cada individuo tiene algo único que ofrecer, y nuestra plataforma fue creada para ayudarte a mostrarlo al mundo.

Nuestro objetivo es crear un espacio donde la tecnología impulse el trabajo en equipo, la innovación y el crecimiento personal. Nos inspiramos en la necesidad de unir talento con oportunidad, y en la idea de que las grandes cosas nacen cuando las personas trabajan juntas.

El proyecto fue desarrollado por un equipo de estudiantes y profesionales apasionados por la tecnología, el diseño y la colaboración digital. WORKSPACE es más que una plataforma: es una comunidad en crecimiento.

Iniciar Sesión

Registrarse

© 2025 WORKSPACE - Inspirando Colaboración

WORKSPACE.

Quiénes Somos

Por qué elegirnos


Login

Registro

## Conecta con el Talento Ideal


WORKSPACE es la plataforma definitiva donde profesionales y empresas se encuentran. Publica proyectos, demuestra tus habilidades y colabora de manera eficiente y segura.

### Lo que nos hace diferentes




#### Conexión Directa

Eliminamos barreras. Encuentra personas con habilidades complementarias para colaborar al instante.



#### Gestión Ágil

Organiza tareas, define metas claras y haz un seguimiento del progreso de tus proyectos en tiempo real.



#### Comunicación Fluida

Mantén el contacto directo con clientes y colaboradores a través de nuestro sistema integrado.

## Explora Categorías

### Programación

Desarrollo web, apps móviles y software a medida.

### Diseño 3D

Modelado, renderizado, animación y visualización.

### Edición de Video

Postproducción, efectos visuales y montaje profesional.

### Fotografía

Sesiones, retoque digital y creación de contenido.

## ¿Listo para empezar a trabajar?

[Crear Cuenta Gratis](#)[Iniciar Sesión](#)

© 2025 WORKSPACE - Todos los derechos reservados.

### WORKSPACE.

Bienvenido de nuevo

Correo Electrónico

ejemplo@correo.com

Contraseña

\*\*\*\*\*

[Iniciar Sesión](#)

¿Aún no tienes cuenta?

[Crear cuenta nueva](#)

# WORKSPACE.

Elige cómo quieres unirte a nosotros



## Soy Worker

Busco oportunidades laborales. Quiero ofrecer mis habilidades, postularme a proyectos y construir mi carrera profesional.

[Crear cuenta Worker](#)



## Soy Boss

Tengo un proyecto en mente. Busco talento profesional, quiero publicar ofertas y gestionar equipos de trabajo.

[Crear cuenta Boss](#)

— Ya tengo cuenta, quiero iniciar sesión

## WORKSPACE.

### Crea tu cuenta Worker

Empieza a encontrar proyectos hoy mismo

Nombre

Tu nombre

Apellidos

Tus apellidos

Número Telefónico

+52 55 1234 5678

Correo Electrónico

ejemplo@correo.com

Nombre de Usuario

Usuario123

Contraseña

.....

[Registrarse](#)

¿Ya tienes una cuenta? [Inicia Sesión aquí](#)

WORKSPACE.

Crea tu cuenta Boss

Gestiona proyectos y encuentra talento ideal

Nombre

Tu nombre

Apellidos

Tus apellidos

Número Telefónico

+52 55...

Empresa

Nombre empresa

Cargo / Puesto

Ej. Gerente de Proyecto

Correo Electrónico

ejemplo@empresa.com

Nombre de Usuario

UsuarioBoss

Contraseña

.....

Registrarse

¿Ya tienes una cuenta? [Inicia Sesión aquí](#)

WORKSPACE.

Inicio 

Cerrar Sesión

Avatar

erick garcia

@erick garcia

BOSS / CLIENTE

EMPRESA / ORGANIZACIÓN

CISCO

CARGO

barrendero

CONTACTO

hola23@gmail.com

¡Bienvenido, erick! 🙌

¿Listo para liderar tu próximo gran proyecto?

Panel de Gestión

✦

Crear Nuevo Proyecto

Publica una nueva oferta para encontrar talento. Define requisitos y presupuesto.

📊

Mis Proyectos

Administra tus ofertas activas, revisa postulaciones y selecciona candidatos.

⚙️

Configuración

Actualiza los datos de tu empresa, información de contacto y preferencias.

© 2025 WORKSPACE - Potenciando el talento global.

Publicar Nuevo Proyecto

Define los detalles de tu oferta para encontrar al talento ideal.

Título del Proyecto

Ej. Rediseño de sitio web, App móvil...

Descripción Detallada

Describe los objetivos, requisitos técnicos y entregables...

Ubicación / Modalidad

Ej. Remoto, Híbrido, CDMX

Presupuesto (USD)

Ej. 1500.00

Publicar Proyecto

Cancelar y volver al perfil

Mis Proyectos

+ Nuevo Proyecto

chap

Publicado: 2025-11-30

necesito que chap me consiga trabajo

2 Postulaciones

Gestionar Candidatos →

← Volver a Mis Proyectos

chap

📍 mi casa 📅 Publicado: 2025-11-30 💰 Presupuesto: \$5000.00

necesito que chap me consiga trabajo

Postulantes 2

**damian ryes**  
Solicitud enviada el: 2025-12-01

**anais becerra camilo**  
Solicitud enviada el: 2025-12-01



**anais becerra camilo**  
Profesional

WORKER / FREELANCER

UBICACIÓN  
Remoto

SOBRE MÍ  
Sin descripción

CONTACTO  
hola@gmail.com

¡Hola, anais! 🖐️

¿Listo para encontrar tu próximo desafío?



⚡ Acciones Rápidas



Buscar Proyectos

Explora el mercado de ofertas disponibles y postúlate a las que encajen con tu perfil.



Mis Solicitudes

Revisa el estado de tus postulaciones enviadas (Pendiente, Aceptada o Rechazada).



Trabajos Activos

Gestiona los proyectos donde has sido aceptado y marca los completados.



Editar Perfil

Mantén tu información actualizada para atraer mejores oportunidades.

Ya te has postulado a este proyecto.

## Explora Oportunidades

Encuentra el proyecto perfecto para demostrar tus habilidades.

Limite: 2025-12-01

**trabajo para jisus**

estamos probando la practica

Postularme Ahora

Limite: 2025-12-01

**patas de pollo**

quiero que me metan un muslo en el culo

Postularme Ahora

Limite: 2025-11-30

**chap**

necesito que chap me consiga trabajo

Postularme Ahora

© 2025 WORKSPACE - Tu plataforma de crecimiento.

## Gestión de Trabajos

Controla tus proyectos activos y revisa tu historial de éxitos.

### En Curso

No tienes trabajos activos en este momento.

### Historial Finalizado

**chap**

PAGADO

Cliente: erick garcia

Pago: \$5000.00

Ver Resumen

© 2025 WORKSPACE - Tu historial profesional.



[← Volver a mis trabajos](#)

✓ FINALIZADO

chap

Ciente: erick garcia

FECHA DE ENTREGA  
📅 2025-11-30

PAGO ACORDADO  
💰 \$5000.00

ESTADO ACTUAL  
Completado

Descripción del Proyecto

necesito que chap me consiga trabajo

🎉 ¡Felicidades! Has completado este trabajo exitosamente.

## 2.2. Arquitectura Tecnológica

- **Backend:** Python 3.13 + Flask.
- **Base de Datos:** PostgreSQL (Producción) / SQLite (Desarrollo).
- **ORM:** SQLAlchemy para el mapeo objeto-relacional.
- **Frontend:** HTML5, CSS3, Jinja2 (Motor de plantillas).
- **Autenticación:** Flask-Login con hashing de contraseñas (Werkzeug Security).

## 2.3. Estructura del Proyecto

El código fuente se organiza de la siguiente manera:

- `app.py`: Controlador principal. Contiene la configuración de la app, inicialización de extensiones y todas las rutas (endpoints) del sistema.
- `models.py`: Define las clases de la base de datos (User, Employee, Boss, JobOffer, Application).
- `config.py`: Variables de configuración y conexión a base de datos.
- `templates/`: Contiene los archivos HTML dinámicos.

El sistema utiliza un modelo relacional con las siguientes tablas principales:

1. **Users:** Tabla maestra de credenciales (email, password hash, tipo de usuario).
2. **Employees / Bosses:** Tablas de perfil vinculadas 1:1 con Users mediante Foreign Key.
3. **Job\_Offers:** Almacena las ofertas publicadas por los Bosses.
4. **Applications:** Tabla intermedia que registra la relación N:M entre Employees y Job\_Offers, controlando el estado de la postulación (pending, accepted, rejected, completed).

## 2.5. Requisitos de Instalación y Despliegue

Para ejecutar el sistema en un entorno nuevo:

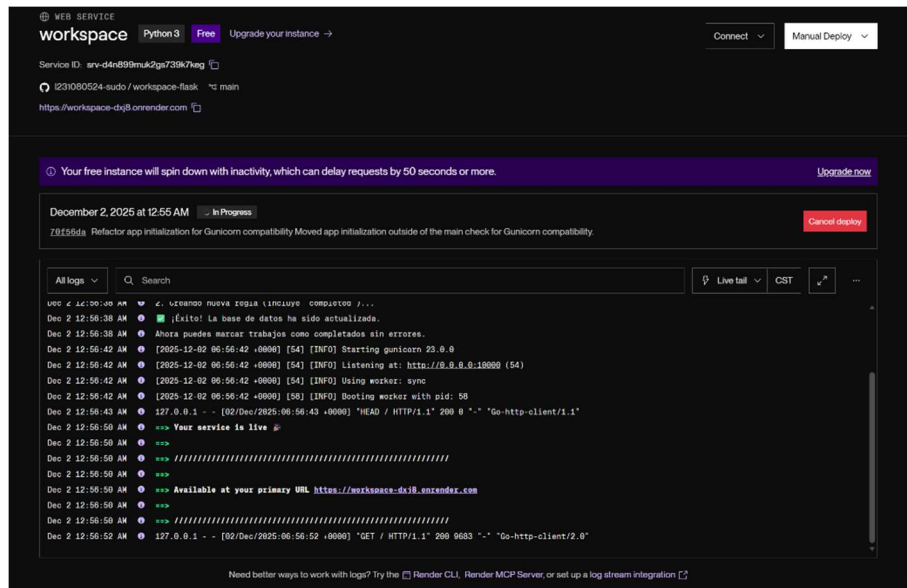
1. **Instalar Dependencias:**  
`pip install -r requirements.txt`
2. **Configurar Variables de Entorno:**
  - Definir `DATABASE_URL` y `SECRET_KEY`.

The screenshot shows the Heroku 'Deploy Phlogies' configuration page. It includes fields for Name, Database, User, Region, PostgreSQL Version, Deploy API Key, and Deploy Region. The 'Plan Options' section is expanded, showing the 'Free' plan selected, which includes 256 MB RAM, 1 GB CPU, and 1 GB Storage. The 'Monthly Total' is \$0/month. The 'Deploy Phlogies Instance' button is visible at the bottom right.

2.

The screenshot shows the Heroku 'Deploy Phlogies' configuration page. It includes fields for Name, Database, User, Region, PostgreSQL Version, Deploy API Key, and Deploy Region. The 'Plan Options' section is expanded, showing the 'Free' plan selected, which includes 256 MB RAM, 1 GB CPU, and 1 GB Storage. The 'Monthly Total' is \$0/month. The 'Deploy Phlogies Instance' button is visible at the bottom right.

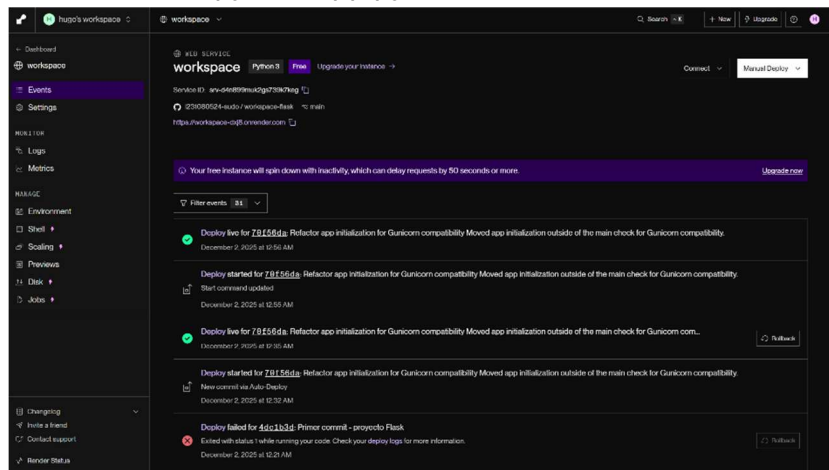
### 3. Inicializar Base de Datos: flask init-db



4. (Este comando crea automáticamente las tablas definidas en models.py).

### 5. Ejecución:

- Desarrollo: python app.py



- 2.
- Producción: gunicorn app:app

## 2.6. Seguridad

- **Contraseñas:** No se almacenan en texto plano. Se utiliza generate\_password\_hash con algoritmo pbkdf2:sha256.
- **Control de Acceso:** Uso de decoradores @login\_required, @boss\_required y @worker\_required para proteger rutas sensibles y evitar escalada de privilegios.
- **Protección SQL:** Uso de ORM (PostgreSQL) para prevenir inyecciones SQL.