

**Documento de recuperación de utu lab**

**por:**Lucas Diaz

**grupo:**3°MD

**Año:**2025

<b>situación problema.....</b>	<b>3</b>
<b>gestión de roles.....</b>	<b>4</b>
<b>Matriz foda.....</b>	<b>7</b>
<b>aplicación iso 9001 al proyecto.....</b>	<b>8</b>
<b>criterios de control de calidad.....</b>	<b>11</b>
<b>Planificación de mejora continua .....</b>	<b>13</b>
<b>diagrama de control de calidad.....</b>	<b>14</b>
<b>validación con usuarios.....</b>	<b>15</b>
<b>Link del GitHub.....</b>	<b>16</b>

### **situación problema**

La Intendencia de Montevideo desea lanzar una aplicación web sencilla para organizar jornadas de voluntariado ambiental, donde los ciudadanos puedan: Ver fechas y lugares de limpieza de playas y espacios públicos. Inscribirse como voluntarios. Ver materiales necesarios para cada jornada (guantes, bolsas, recipientes para reciclaje). Compartir fotos después del evento.

### Gestión de roles

Cordinador	
Responsabilidad	El coordinador es responsable de planificar, organizar, controlar, y guiar el desarrollo del software
supervisa a	todos
justificación	mayor autoridad dentro del equipo debido a ser el responsable por el desarrollo, la planificación y el control de los integrantes
Analista	
Responsabilidad	Es responsable de relevar la información y documentación de la situación problema y buscar las soluciones para tal situación, la creación de los requerimientos funcionales y no funcionales, estudio de factibilidad etc
supervisa a	diseñador de base de datos
justificación	para controlar que la base de datos sea acorde a la situación problema
Diseñador de base de datos	
Responsabilidad	realizar la planificación de la base de datos como el diagrama, las tablas en su tercer forma normal y la programación de los archivos html
supervisa a	Programador Frontend
justificación	certificar que la codificación utilice los mismos nombres que la base de datos

## Diseñador gráfico

Responsabilidad	realiza el prototipo y diseño del software
supervisa a	Programador Frontend
justificación	para controlar que el proceso y resultado sea acorde al diseño realizado

## Programador frontend

Responsabilidad	programar el estilo de la página(lo visual)
supervisa a	Programador Backend
justificación	controlar que las funcionalidades estén ubicadas correctamente en sus respectivas posiciones dentro del diseño

## Programador Backend

Responsabilidad	Programar la funcionalidad de la página
supervisa a	nadie
justificación	nulo

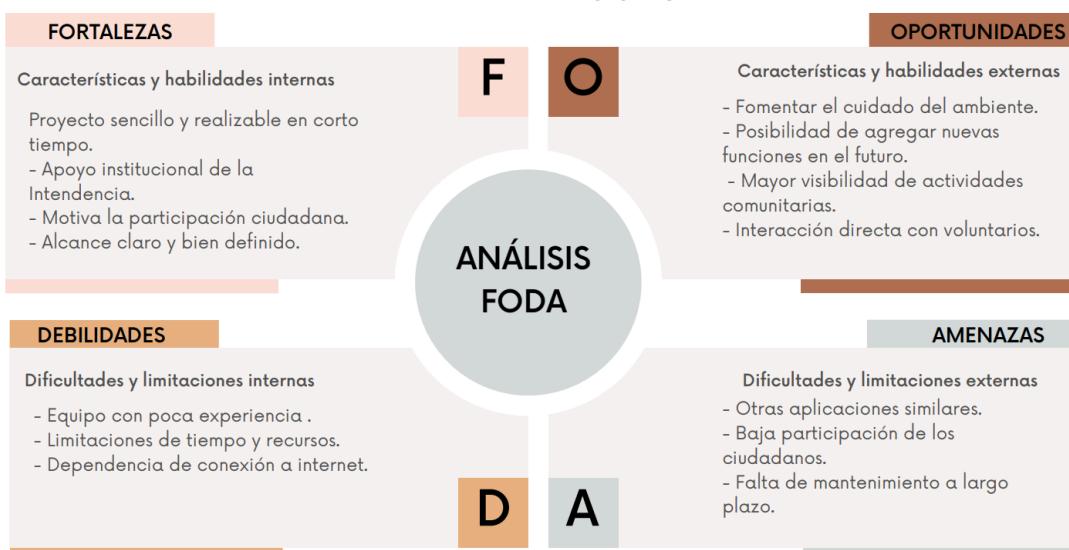
## Desarrollador de servidores

Responsabilidad	crea los servidores, baja sus dependencias, configura la red, implementa la base de datos, crea los contenedores, y desarrolla el sistema de backup
supervisa a	nadie
justificación	nulo

Tester	
Responsabilidad	realiza el testeo de la página con las estrategias correspondientes y lo documenta
supervisa a	Programador Backend
Justificación	informa los errores de funcionamiento o de lógica del código

división de roles	
Nombre	Rol
Lucas	Desarrollador de servidores

## MATRIZ DE ANÁLISIS FODA



## **Aplicación de ISO 9001 al proyecto**

### **1) Definición requisitos**

- Funcionalidades de la app
  - ver los detalles de los locales de limpieza
  - inscribirse a voluntariado
  - subir foto a la galería de la página
- Usuarios objetivos
  - voluntarios a la limpieza de los locales
- Fechas estimadas
  - 6 meses
- Restricciones técnicas
  - limitación de almacenamiento de fotos grandes
  - lenguajes de códigos php, html, java, java script
  - tiempo de desarrollo
  - app web

### **2) Planificación documentada**

- Plan del proyecto
  - roles
  - cronograma
  - reuniones
- Cronograma
  - 1°mes: relevamiento de datos
  - 2°mes: prototipado de la página, diseño de la base de datos
  - 3° - 4°mes: desarrollo del código frontend/Backend del programa
  - 5°mes: desarrollo del servidor
  - 6°mes: testing
- Entregables
  - documento de la información relevada
  - factibilidad del proyecto
  - diseño de la página y de la base de datos
  - archivos de código
  - métricas del proyecto
  - contenedores docker-compose
  - documentos de caja negra y blanca

- Roles y responsabilidades
  - coordinador: El coordinador es responsable de planificar, organizar, controlar, y guiar el desarrollo del software
  - Analista: Es responsable de relevar la información y documentación de la situación problema y buscar las soluciones para tal situación, la creación de los requerimientos funcionales y no funcionales, estudio de factibilidad etc
  - Diseñador de base de datos: realizar la planificación de la base de datos como el diagrama, las tablas en su tercer forma normal y la programación de los archivos html
  - Diseñador gráfico: realiza el prototipo y diseño del software
  - Programador frontend: programar el estilo de la página (lo visual)
  - Programador Backend: Programar la funcionalidad de la página
  - Desarrollador de servidores: crea los servidores, baja sus dependencias, configura la red, implementa la base de datos, crea los contenedores, y desarrolla el sistema de backup
  - Tester: realiza el testeo de la página con las estrategias correspondientes y lo documenta
  
- Recursos necesarios
  - Hardware:
    - Computadoras del laboratorio o PCs personales
    - Router / red local para pruebas
    - Servidor local (XAMPP, WAMP, Docker o Linux Server)
    - Almacenamiento disponible para la subida de fotos (mínimo 1 GB)
    - Riesgos identificados
  - software
    - Figma – Prototipo de la interfaz
    - GitHub – Control de versiones y documentación
    - MySQL – Base de datos
    - PHP o Node.js (según tu proyecto) – Backend
    - HTML, CSS, JavaScript – Frontend
    - Docker – Contenedores y ambiente de desarrollo (si lo usan)
    - Editor de código (VSCode recomendado)

### 3) Gestión de riesgos

- Fallos del servidor
- Retrasos en el prototipado
- Mala conexión entre frontend y backend
- Falta de tiempo para testeo

#### 4)Control de versiones y documentación

- Registrar avances
  - gant
- Controlar versiones
- Asegurar que nadie trabaje sobre archivos desactualizados
- Mantener historial claro

## **Criterios de control de calidad**

### 1) Calidad del código

- Código limpio y comentado
- Buenas prácticas de programación
- Nombres coherentes con la base de datos
- Validación de formularios

### 2) Calidad del diseño UX/UI

- Navegación simple
- Imágenes optimizadas
- Paleta de colores consistente
- Legibilidad del texto

### 3) Calidad funcional

- El registro de voluntarios funciona sin errores
- La carga de fotos después del evento funciona
- Los materiales de la jornada se muestran correctamente
- Los datos quedan guardados en la base sin duplicaciones

### 4) Calidad del rendimiento

- La web carga en menos de 3 segundos
- El servidor responde sin caídas

## 5) Calidad de documentación

- Manuales actualizados
- Requerimientos completos
- Cambios documentados

## **Planificación de mejora continua**

P – Planificar

Se planifica el proyecto:  
requerimientos, roles, riesgos, tiempos, calidad.

H – Hacer

Se desarrolla el prototipo y la aplicación.

V – Verificar

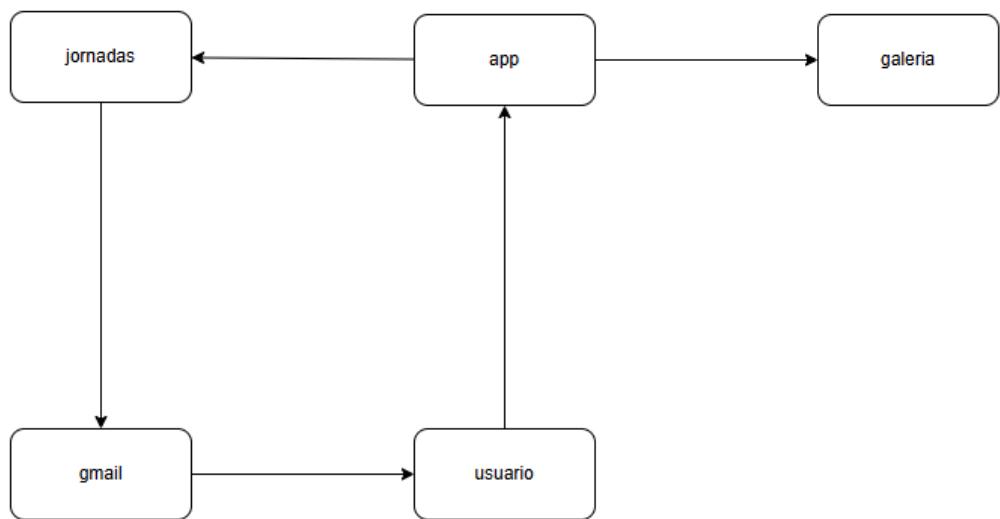
El tester revisa:

- fallas,
- errores de lógica,
- problemas de diseño,
- incumplimiento de requerimientos.

A – Actuar

Se corrigen errores y se mejora el proceso para que no vuelva a ocurrir.

## diagrama de flujo de datos



## **validación con usuarios**

- preguntas
  - ¿Pudiste encontrar rápido la jornada que buscabas?
  - ¿El botón de inscribirse es fácil de ver?
  - ¿Te fue sencillo subir una foto?
- Dos posibles cambios tras validar
  - Hacer más grande el botón de inscripción.
  - Simplificar el formulario a “nombre + correo”.

**GitHub:**<https://github.com/luhipaing/lucas->