



DOCUMENTO FUNCIONAL DEL PROYECTO

ALUMNO/GRUPO: Stephany Gamito, Pau Cruz

1. Introducción y contexto

- Montaremos una infraestructura de red con todo lo necesario para poder utilizarla en un centro educativo o empresa. Incluiremos elementos como un dhcp, dns, etc.
 - Permitirá a los usuarios tener más privacidad al no depender de grandes empresas para gestionar sus archivos y su información. Lo vemos necesario ya que hoy en día no hay privacidad cuando se trata de utilizar aplicaciones que dependan de empresas externas, por lo que queremos promover el uso de aplicaciones de código abierto que uno mismo puede montar en su casa, centro educativo o empresa.
-

2. Análisis de requisitos

2.1. Requisitos funcionales (RF)

Qué debe hacer el sistema.

| Código | Descripción del requisito funcional |
|--------|--|
| RF1 | Proporcionar conexión a internet a los usuarios de la red. |
| RF2 | Proporcionar una serie de aplicaciones en la red que aseguran la privacidad de los datos de los usuarios, como servicios de mensajería o clouds. |

| | |
|-----|---|
| RF3 | Garantizar el buen funcionamiento de las máquinas de isard y las conexiones a internet. |
|-----|---|

2.2. Requisitos no funcionales (RNF)

Cómo debe comportarse el sistema.

| Código | Descripción del requisito no funcional |
|--------|---|
| RNF1 | El sistema debe asignar direcciones IP a los dispositivos que se conectan a la red. |
| RNF2 | El sistema debe soportar la carga de los usuarios que se conectan y utilizan sus servicios. |

2.3. Restricciones

Condiciones o limitaciones del proyecto.

- Lenguajes o tecnologías obligatorias: isard, apache2, aplicaciones web necesarias.
 - Recursos disponibles (tiempo, equipo, materiales): Disponemos de una cuenta de isardVDI que nos ha proporcionado nuestro magnífico profesor de servicios de red Victor Carceler. En ella, montaremos las diferentes máquinas virtuales que actuarán como servidor y cliente y donde montaremos las aplicaciones web, DNS y demás.
 - No disponemos de demasiado tiempo, ya que al principio tuvimos bastantes problemas para decidir que íbamos a hacer con nuestro proyecto, pero somos capaces de sacar el proyecto adelante y cumplir con todos los objetivos que nos hemos propuesto.
 - Dependencias o limitaciones técnicas: Dependemos mucho de isard, ya que si entra en mantenimiento o se cae, no podremos presentar o trabajar en nuestro proyecto.
-

3. Análisis de usuarios y roles

| Rol | Descripción | Permisos principales |
|-----|-------------|----------------------|
|-----|-------------|----------------------|

| | | |
|---------------|-------------------------------|--|
| Administrador | Gestiona usuarios y recursos. | root y usuario administrador en las aplicaciones web. |
| Usuario | Usuario común de la red. | Conexión a la red y acceso como usuario normal a las aplicaciones web. |

4. Casos de uso / Escenarios de uso

| Código | Nombre del caso de uso | Actor principal | Descripción | Resultado esperado |
|--------|------------------------|-----------------|---|-----------------------------------|
| CU1 | Registrar usuario | Administrador | El administrador, con las credenciales que le ha proporcionado la persona, registra como usuario a esa persona en las distintas aplicaciones web y en la red. | Usuario registrado correctamente. |
| CU2 | Iniciar sesión | Usuario | Introduce credenciales y accede a las aplicaciones web existentes. | Acceso concedido. |
| CU3 | Conectarse a internet | Usuario | El usuario enciende su máquina y el dhcp le asigna una IP | Conectado a internet. |

5. Modelo de datos o estructura de la información

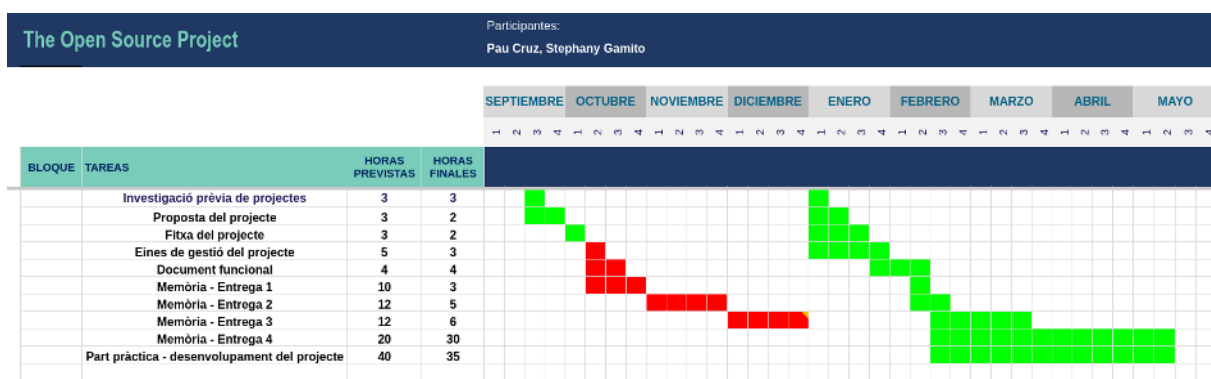
- Entidades: Usuarios y contraseñas
- Relaciones: los nombres de usuario y las contraseñas estarán almacenadas en sus respectivas bases de datos, conservándose así en un lugar seguro.

6. Diseño de la interfaz

- La interfaz será el CLI de Ubuntu en los servicios. Para los clientes de los usuarios, por ahora, utilizaremos ubuntu con GNOME, aunque les daremos la opción de usar KDE Plasma.
-

7. Planificación técnica

- Lenguajes y frameworks: HTML5, CSS, JavaScript
- Base de datos: mySQL, postgres
- Herramientas de diseño o edición: vi, neovim, krita
- Reparto de tareas (si es en grupo):
 - Stephany → Página web, memoria, configuración de aplicaciones web.
 - Pau → Instalación de servicios, aplicaciones y sistemas operativos, memoria.
- **Cronograma (puede incluir un diagrama de Gantt)**



8. Análisis de riesgos

8.1. Identificación de riesgos

- Falta de tiempo o mala planificación.
- Problemas técnicos con isard
- Pérdida de datos.
- Problemas de compatibilidad con servicios o aplicaciones

8.2. Valoración y respuesta

Clasifica cada riesgo según su probabilidad e impacto, e indica cómo se mitigará.

| Riesgo | Probabilidad | Impacto | Plan de prevención o contingencia |
|--|--------------|---------|---|
| Falta de tiempo | Alta | Alto | Dividir tareas y trabajar lo más rápido posible para dejar lo esencial hecho. |
| Problemas técnicos con isard | Media | Alto | Grabar de antemano un vídeo donde se vea que todo funciona. |
| Pérdida de datos | Media | Alto | Hacer copias de seguridad semanales. |
| Problemas de compatibilidad con servicios o aplicaciones | Media | Medio | Tener en cuenta alternativas a los servicios que vamos a utilizar. |

9. Validación y criterios de éxito

- Criterios de aceptación (qué debe cumplirse para darlo por válido)

Para dar por válido el proyecto, debe cumplir con los siguientes requisitos:

- Permitir a los usuarios conectarse a la red
- Asignar automáticamente una dirección IP al usuario
- Dar acceso a internet al usuario
- Dar acceso a los servicios y aplicaciones instalados, como matrix, owncloud, etc.

- Pruebas previstas (funcionales, de usuario, de rendimiento).

Realizaremos las siguientes pruebas una vez tengamos todo montado:

- Garantizar la conectividad a internet
- Garantizar la conexión a las aplicaciones instaladas, como owncloud y matrix
- Garantizar el buen funcionamiento de la red.
- Garantizar la conectividad y funcionamiento de la página web.

- Indicadores de calidad o resultados esperados.
 - Buena conectividad
 - Buen funcionamiento de las aplicaciones web
 - Buen funcionamiento de la página web
-

10. Conclusión

Resume las decisiones principales tomadas durante el análisis:

- Qué funciones tendrá el sistema.
 - 1. Proporcionar conexión a internet a los usuarios de la red.
 - 2. Ofrecer aplicaciones que aseguren la privacidad de los datos, como servicios de mensajería y almacenamiento en la nube.
 - 3. Garantizar el buen funcionamiento de las máquinas de isard y las conexiones a internet.

- Qué tecnologías se utilizarán.
 - Isard para la virtualización.
 - Apache2 como servidor web.
 - BASH para scripts.
 - MySQL para la gestión de base de datos.
 - Ubuntu con GNOME como interfaz de usuario.
 - Ubuntu server sin interfaz gráfica para el servidor.

- Qué valor aporta el proyecto.
 - Este proyecto no solo proporciona una solución de red segura y privada, sino que también promueve el uso de herramientas de código abierto, fomentando la independencia de grandes empresas en la gestión de datos y archivos.

Indica los próximos pasos:

- Preparar el entorno de desarrollo.
- Crear la estructura inicial del repositorio y las aplicaciones web y servicios.
- Empezar a añadir información a la memoria.

“El análisis funcional nos ha ayudado a ver lo que realmente necesitamos para el proyecto y la magnitud que tiene.”