

## Proyecto Hosting – Estructura del Computador II

### Plataforma de Hosting Basada en Contenedores

#### Descripción general

El propósito de este proyecto es desarrollar una plataforma de hosting de páginas web basada en contenedores. La solución permitirá que los usuarios, autenticados mediante Roble, puedan desplegar sus propios sitios web directamente desde un repositorio de GitHub.

Cada usuario podrá crear uno o varios proyectos web a partir de plantillas (templates) predefinidas. Una vez el usuario proporcione la URL de su repositorio y un nombre de proyecto, el sistema deberá levantar un contenedor Docker que aloje el sitio, accesible en la URL local:

`http://nombreProyecto.nombreUsuario.localhost`

#### Requerimientos funcionales

##### 1. Autenticación:

- ✓ La plataforma debe integrarse con Roble para gestionar la creación y autenticación de usuarios. *por ahora tenemos solo autenticación*
- Cada usuario dispondrá de un espacio personal de proyectos.

##### 2. Creación de proyectos

- Para crear un nuevo proyecto, el usuario deberá:
  - Seleccionar uno de los tres templates base. *estas templates son repos en github*
  - Clonar el template en su repositorio personal de GitHub. ✓
  - Modificar el contenido y registrar en la plataforma la URL del repositorio junto con el nombre del proyecto. *para modificar el contenido no necesariamente en la plataforma lo importante es la url del repositorio*
- La plataforma deberá desplegar automáticamente un contenedor Docker basado en la información recibida.

##### 3. Templates disponibles

- Los templates deben estar dockerizados y ser funcionales de manera independiente:
  - Template 1: Sitio estático (HTML + CSS + JS).
  - Template 2: Aplicación con React. *DECIDIR QUÉ TEMPLATES*
  - Template 3: Una opción adicional a elección del equipo, por ejemplo:
    - Flask (Python) con plantillas HTML.
    - Node.js + Express sirviendo contenido dinámico.

- Next.js, Vue.js o SvelteKit, si se desea explorar frameworks modernos.
- Cada template deberá incluir su propio Dockerfile
- 4. Despliegue y acceso
  - Cada proyecto activo deberá ejecutarse en su propio contenedor.
  - Los contenedores deberán ser accesibles mediante subdominios locales gestionados por un reverse proxy (p. ej. Nginx o Caddy): *Traefik 7to 8o*  
`http://nombreProyecto.nombreUsuario.localhost`
- 5. Gestión de recursos
  - Se deberá implementar control de uso de recursos y protección ante sobrecarga:
    - Límite de peticiones por minuto (rate limiting).
    - Restricciones de CPU y memoria por contenedor (`--cpus`, `--memory`).
  - Para optimizar el uso del sistema:
    - Los contenedores inactivos durante más de 30 minutos deberán apagarse automáticamente.
    - Si reciben una nueva solicitud, deberán reiniciarse de forma automática.

#### Requerimientos técnicos

- Uso obligatorio de Docker para todos los componentes.
- Arquitectura modular con servicios separados para autenticación, gestión de contenedores y proxy.

#### Condiciones del proyecto

- Trabajo en equipo:
  - El proyecto podrá desarrollarse en equipos de hasta 5 estudiantes (se pueden crear equipos con estudiantes de ambos NRC). Solo pueden participar estudiantes que obtuvieron por lo menos 4.0 en el corte de CPU.
  - Cada miembro deberá tener una participación claramente definida y documentada en el repositorio.
  - La participación en el proyecto no es obligatoria, es una alternativa para que el no quiera presentar el parcial. Eso sí, una vez comprometido con el proyecto, no se puede cambiar de opinión. Para participar se debe:
    - Formar el equipo e inscribirse en el grupo en el catálogo. ESTE PROCESO DE OBLIGATORIO PARA OBTENER NOTA.
    - Si un equipo está formado por participantes de dos NRC, se deben formar dos grupos, uno en cada NRC.
    - Inscribirse también en: [equiposProyectoHosting.xlsx](#)
- Autorización de uso:
 

Al presentar el proyecto, los integrantes del equipo autorizan expresamente que la solución (total o parcial) pueda ser utilizada por Roble como base para ofrecer

servicios de hosting académicos o institucionales en el futuro.

Se reconocerá públicamente la autoría del equipo en caso de que su trabajo sea adoptado.

### Entregable

Se debe entregar en la actividad por el catálogo (todos los miembros del equipo deben estar inscritos, si hay miembros de ambos NRC se debe enviar dos veces) el enlace al repositorio con código fuente completo de la plataforma (backend, frontend y configuración de contenedores). En el readme del repositorio se debe tener:

1. Enlace a los repositorios con los templates dockerizados funcionales.
2. Enlace al video en youtube de demostración mostrando:
  - a. Registro e inicio de sesión.
  - b. Creación y despliegue de un proyecto.
  - c. Funcionamiento de la gestión de recursos y apagado automático.
3. Documento técnico que incluya:
  - Descripción de la arquitectura y componentes.
  - Flujo de trabajo del sistema.
  - Estrategia de seguridad y optimización de recursos.

Fecha de entrega: Día del final según programación