



# Implementación de infraestructura TI para Nucleoo.

## Descripción del proyecto:

La empresa Nucleoo, situada en el corazón de Gran Vía en Granada, desarrolla fundamentalmente consultoría informática para otras empresas granadinas, y nos ha pedido diseñar una infraestructura TI para su sede, con 3 niveles de trabajadores según a los recursos físicos y de información a los que van a poder acceder. Técnicos, desarrolladores y administradores.

## Tareas a realizar:

Para ello vamos a utilizar nuestros conocimientos de los 2 primeros temas de SI, centrados en gestión de archivos y gestión de usuarios y procesos, además de lo nuevo que iremos aprendiendo haciendo scripts y conociendo la gestión de redes en Linux, para ofrecerles una solución fiable, escalable, fiable, duradera y sencilla para poder explicarla al cliente.

Recordar que este primer enunciado es solo la tarea inicial, la cual iremos modificando en distintos incrementos tal y como pasa en la industria real, con metodologías como la metodología ágil.

## Consideraciones IMPORTANTES:

1. El proyecto se entregará vía GitHub: <https://classroom.github.com/a/9UcZr8u->
2. La gestión de versiones es **obligatoria para que el proyecto sea evaluado**:
  - a. Debemos ver varios commits para comprobar la evolución del proyecto.
  - b. Proyectos realizados por arte de magia, con un único commit final, varios commits con el mismo contenido, pocos commits o commits en un espacio muy breve de tiempo, se entenderán como no realizados por el alumno.
  - c. Si no se cumple este requisito, la calificación del proyecto es de cero (0) puntos.
3. La defensa del proyecto es obligatoria. Si no se realiza la defensa, la calificación del proyecto es de cero (0) puntos.

## Tarea inicial:

### 1. Creación de usuarios:

- a. Habrá tres grupos de usuarios en el sistema. Técnicos, desarrolladores y administradores.



- b. De cada tipo se crearán 5 usuarios, menos el grupo de administradores que será únicamente administrador1 y administrador2.
- c. Cada usuario tendrá su carpeta home. Además, dentro de /opt, se crearán carpetas compartidas para depositar los repositorios.

### 2. Creación de estructura de archivos y carpetas.

- a. en /opt existirá una carpeta llamada compartido, y dentro la carpeta src donde estarán los códigos fuentes, la carpeta docs con la documentación, la carpeta config con los archivos de configuración y la carpeta recovery con los archivos sql para recuperar la base de datos en caso de caída o pérdida de datos.
- b. los únicos que deberían tener permisos de escritura sobre los archivos de configuración (config) deberán ser los administradores. Además, los demás usuarios podrán ver el contenido de la carpeta pero no leer sus archivos ni escribir sobre ellos.
- c. El código fuente solo deberían poder modificarlo los desarrolladores y los administradores, mientras que todos los demás deberían tener permiso de lectura para toda la carpeta compartida.

Los archivos de recovery deberían ser susceptibles de ser modificados solo por los administradores, pero leído por todos.

- d. la documentación deberá poderse modificar solo por técnicos, desarrolladores y administradores, pero si hay más usuarios en el sistema deberían solo poder leer el contenido de los archivos.

### 3. Automatización.

3.3. Genera las órdenes pertinentes para que, una vez al día, el sistema se asegure que no hay archivos fuera de sus carpetas. Debería quedar en /opt/compartido:

- a. En src los ejecutables y código fuente (.py, .c, .cpp, .js, .sh todos los que sepáis).
- b. En config los archivos de configuración (.cfg)
- c. En docs la documentación (.pdf, .md, .doc, .odt, .docx, .txt)
- d. En recovery los archivos de recuperación de la base de datos (.sql)