

# Laboratorio de administración y gestión de redes y sistemas

## Examen de teoría

11 de enero de 2024.

Grado en Ingeniería Telemática, Universidad Rey Juan Carlos

---

- Ejecuta intercambio para llevar el directorio `~/Escritorio/intercambio/lagrs` de tu cuenta habitual a esta cuenta en modo examen.
- Descarga el fichero `http://ortuno.es/examen_lagrs.zip` en cualquier lugar (p.e. la carpeta *descargas*) y extrae su contenido (gestor de ficheros, botón secundario, menú contextual, extraer aquí)
- Verás que acabas de descomprimir un programa llamado *prepara\_todo*. Ejecútalo (con doble clic).
- Esto ha copiado en tu puesto el fichero `~/teoria.txt`. Ábrelo, rellena tus datos personales y responde ahí las siguientes preguntas.

### Ejercicio 1. (2.5 puntos)

Define brevemente el concepto *devops*. Define brevemente el concepto *diseño en cascada* ¿qué relación hay entre ambos?

#### Respuesta:

Versión elaborada por el profesor de la asignatura:

- *Devops* son una serie de técnicas para optimizar el trabajo conjunto del equipo de desarrollo (programadores) y el equipo de operaciones (administradores). Se dividen en dos grandes grupos: técnicas para organizar el trabajo de las personas y técnicas para automatizar la puesta en producción del software.
- *Diseño en cascada* es una técnica de desarrollo de software propia de los años 80 donde se siguen una serie de pasos de forma rígida y secuencial: análisis de requerimientos, diseño, programación, prueba, despliegue y mantenimiento.
- Ambas técnicas buscan lo mismo: optimizar el desarrollo y puesta en producción del software. Pero el *diseño en cascada* es un enfoque muy rígido que normalmente hoy se considera obsoleto, mientras que *devops* son técnicas actuales, con un enfoque mucho más flexible.

Versión de ChatGPT

- DevOps es una cultura y conjunto de prácticas que busca integrar de manera colaborativa el desarrollo de software (Dev) y las operaciones (Ops). Su objetivo es acortar el ciclo de vida de desarrollo, mejorar la colaboración entre equipos y proporcionar una entrega continua y confiable de software. DevOps promueve la automatización, la monitorización y la mejora continua para lograr un desarrollo más eficiente y una implementación más rápida.

- El diseño en cascada es un enfoque de desarrollo de software tradicional y secuencial en el que las fases del ciclo de vida del software se llevan a cabo de manera lineal y secuencial. Las etapas incluyen requisitos, diseño, implementación, verificación, y mantenimiento. Cada fase debe completarse antes de pasar a la siguiente, y los cambios significativos en etapas anteriores pueden resultar complicados y costosos.
- La relación entre DevOps y el diseño en cascada se centra en sus diferencias y en cómo DevOps aborda las limitaciones del enfoque de cascada. Mientras que el diseño en cascada tiende a ser más rígido y secuencial, DevOps promueve la agilidad, la automatización y la colaboración continua entre equipos. DevOps aborda las limitaciones del diseño en cascada al permitir cambios rápidos y frecuentes, mejorando así la capacidad de respuesta del desarrollo de software a los cambios en los requisitos y en el entorno. En lugar de seguir una secuencia lineal, DevOps adopta un enfoque más ágil y orientado a resultados.

## Ejercicio 2. (2.5 puntos)

- 2.1) ¿Para qué sirve el fichero *authorized\_keys*?
- 2.2) ¿Dónde se guarda?
- 2.3) ¿Qué contiene?
- 2.4) En el caso de uso típico ¿quién lo escribe?
- 2.5) En el caso de uso típico ¿quién lo lee?

### Respuesta:

- 2.1) Para autenticar sesiones de ssh sin necesidad de teclear contraseñas.
- 2.2) En el directorio `~/ .ssh` de la máquina remota a la que se quiere acceder.
- 2.3) Claves públicas autorizadas para establecer sesiones.
- 2.4) El usuario.
- 2.5) El demonio ssh de la máquina remota.

## Ejercicio 3. (2.5 puntos)

¿Qué pasa si ejecutamos la orden `apt update` pero no ejecutamos `apt upgrade`?

### Respuesta:

El sistema tendrá la lista de la última versión disponible de cada paquete, pero no descargará ni instalará esos paquetes.

## Ejercicio 4. (2.5 puntos)

Supongamos que pruebo un programa en mi máquina local y todo va bien. En otro momento lo vuelvo a probar, en la misma máquina, pero entrando por ssh. Ahora falla. Esto se puede deber a múltiples causas, pero hay una muy frecuente, que hemos tratado mucho en clase. ¿cuál?

### Respuesta:

Si estamos con el mismo usuario en la misma máquina, podría pensarse que el resultado siempre será el mismo porque se trata del mismo sistema, mismos ficheros, mismos permisos, etc.

Pero al comienzo de una sesión local, por ser (normalmente) no de login, se ejecuta el fichero `.bashrc`. Mientras que al comienzo de una sesión por ssh, por ser de login, se ejecuta el fichero `.bash_profile`.

Siendo dos ficheros distintos, no es raro que la configuración de las variables de entorno sea también distinta. En el segundo caso es típico que no se haya actualizado alguna variable de entorno importante como `PATH`, `PYTHONPATH`, `JAVA_HOME`, `LD_LIBRARY_PATH`, etc

Una buena solución es modificar el fichero `.bash_profile` para que llame a `.bashrc`, de forma que siempre tengamos la misma configuración, sin importar el tipo de shell.