



**UNIVERSIDAD  
DE GRANADA**

**SEMINARIO 1:  
CREACIÓN DE CONTENEDORES**

Profesora: Claudia Villalonga Palliser

Asignatura: Cloud Computing: Fundamentos e Infraestructuras

Título: Máster Universitario en Ingeniería Informática

Curso: 2023/2024

# CREACIÓN DE CONTENEDORES

## SEMINARIO 1

- ✓ Este seminario presenta Docker, una de las plataformas más usadas hoy en día para el diseño y desarrollo de contenedores. Los contenedores empaquetan y aíslan las aplicaciones con todo su entorno de ejecución, lo que permite abstraer los recursos en la nube. Por lo tanto, un usuario o consumidor de la nube puede trasladar fácilmente la aplicación de una nube a otra conservando todas sus funciones.



# CONTENEDORES

## DEFINICIÓN

- ✓ Un contenedor es un entorno controlado (sandbox) aislado de todos los demás procesos que se ejecutan en la máquina anfitriona en la que se está ejecutando el contenedor.
- ✓ Un contenedor:
  - ✓ Es una instancia ejecutable de una imagen.
  - ✓ Puede ejecutarse en máquinas locales, máquinas virtuales o desplegarse en la nube.
  - ✓ Es portátil (y puede ejecutarse en cualquier sistema operativo).
  - ✓ Está aislado de otros contenedores y ejecuta su propio software, binarios, configuraciones...

<https://docs.docker.com/get-started/>



# CONTENEDORES

## DEFINICIÓN

- ✓ Los contenedores son paquetes ligeros que incluyen el código de las aplicaciones junto con sus dependencias, como versiones concretas de entornos de ejecución de ciertos lenguajes de programación y bibliotecas indispensables para ejecutar los servicios de software.

<https://cloud.google.com/learn/what-are-containers?hl=es>

- ✓ Los contenedores son unidades ejecutables de software donde se empaqueta el código de aplicación, junto con sus bibliotecas y dependencias, de forma común para que se pueda ejecutar en cualquier lugar, ya sea en el escritorio, en la TI tradicional o en la nube.

<https://www.ibm.com/es-es/topics/containers>



# CONTENEDORES

## DIFERENCIA ENTRE MÁQUINAS VIRTUALES Y CONTENEDORES

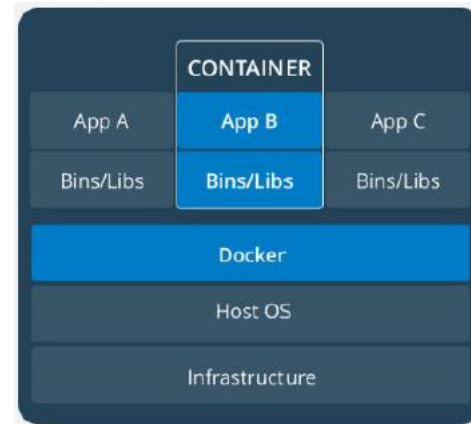
### MÁQUINAS VIRTUALES

- ✓ Sistemas operativos diferentes: cada máquina virtual tiene su propio sistema operativo.



### CONTENEDORES

- ✓ Kernel compartido: los contenedores solo contienen los binarios y las librerías necesarias para su funcionamiento.



# CONTENEDORES

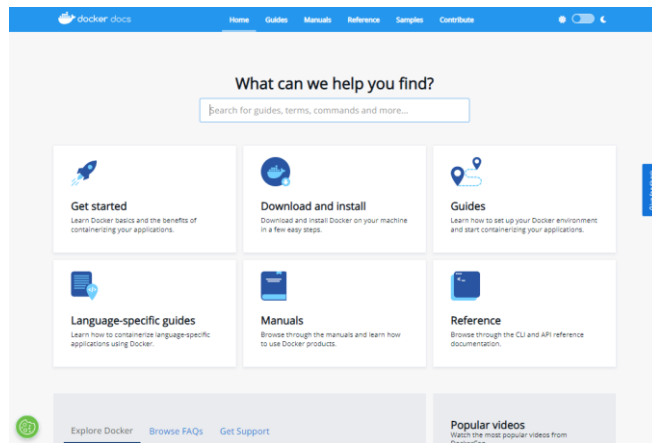
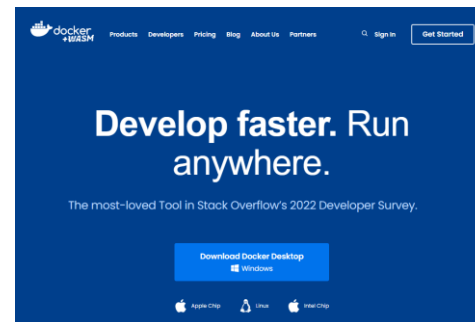
## VENTAJAS

- ✓ **Ligeros:** Son procesos que comparten el Kernel del sistema operativo.
- ✓ **Portables:** Se pueden crear localmente y ejecutar en cualquier entorno o en computación en la nube. Además, al desplegarse en proveedores en la nube, sin tener que definir una infraestructura (máquina virtual), se reducen los costes a dichos proveedores y estos a su vez ofrecen menor coste en la ejecución.
- ✓ **Intercambiables:** Se pueden desarrollar nuevas actualizaciones que son fácilmente actualizables.
- ✓ **Escalables:** Se pueden añadir y distribuir copias.
- ✓ **Flexibles:** Gran cantidad de aplicaciones pueden ser incluidas en un contenedor.

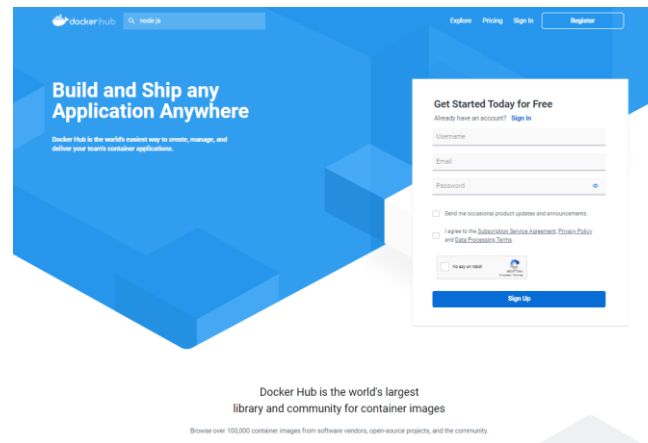
# DOCKER

PLATAFORMA PARA CONTENEDORES

<https://www.docker.com/>



<https://docs.docker.com/>



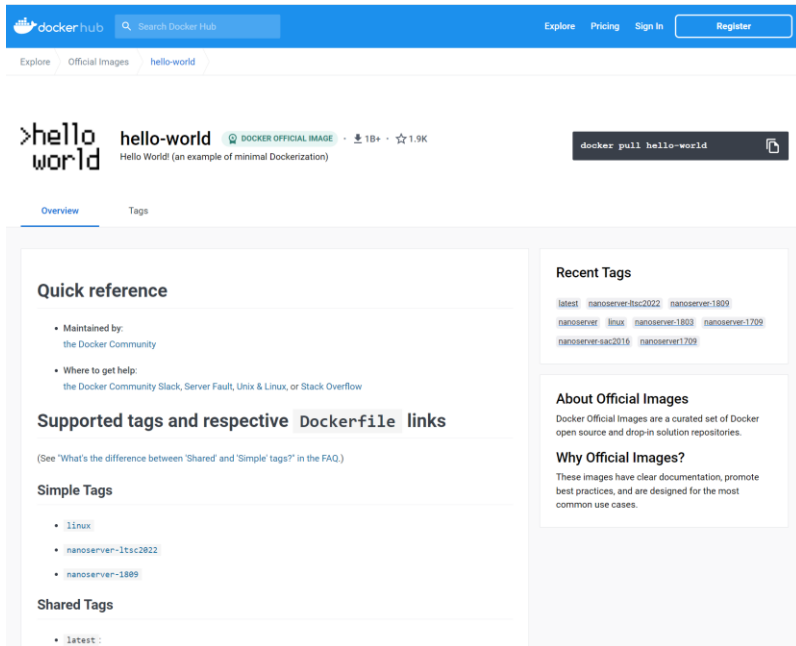
<https://hub.docker.com/>



UNIVERSIDAD  
DE GRANADA

# DOCKER

## PLATAFORMA PARA CONTENEDORES



The screenshot shows the Docker Hub interface for the 'hello-world' image. At the top, there's a blue navigation bar with the Docker Hub logo, a search bar, and links for Explore, Pricing, Sign In, and Register. Below the navigation bar, the 'hello-world' image is featured with a 'DOCKER OFFICIAL IMAGE' badge, a pull count of 1B+, and a star count of 1.9K. A 'docker pull hello-world' button is visible. The main content area is divided into two columns. The left column has an 'Overview' tab selected, showing a 'Quick reference' section with links to the Docker Community and help resources, a 'Supported tags and respective Dockerfile links' section with a link to the FAQ, and a 'Simple Tags' list containing 'linux', 'nanoserver-lts2022', and 'nanoserver-1809'. The right column has a 'Tags' tab selected, showing a 'Recent Tags' list with 'latest', 'nanoserver-lts2022', and 'nanoserver-1809', and an 'About Official Images' section explaining that these images are curated, open source, and designed for common use cases.

[https://hub.docker.com/\\_/hello-world/](https://hub.docker.com/_/hello-world/)

- Un **Contenedor** (**hello-world**) es un proceso que está ejecutándose en un entorno con restricciones de recursos y que está administrado por el sistema operativo.
- Una **Imagen** (**hello-world IMAGE ID feb5d9fea6a5**) es el conjunto de archivos necesarios para ejecutar un contenedor y los procesos que desarrolla. Es como una plantilla a partir de la cual se pueden crear una cantidad ilimitada de contenedores.
- Un **Repositorio** (**DockerHub**) es el lugar donde se almacenan las imágenes.
- Un **Registro de contenedores** es una aplicación web donde se clasifican y almacenan los repositorios.





# DOCKER

PLATAFORMA PARA CONTENEDORES

The screenshot shows the Docker Hub interface for the 'hello-world' image. At the top, there's a blue navigation bar with the Docker Hub logo, a search bar, and links for 'Explore', 'Pricing', 'Sign In', and 'Register'. Below the navigation bar, the 'hello-world' image is featured with its logo, name, and a description: 'Hello World! (an example of minimal Dockerization)'. It also shows '18+' pulls and '1.9K' stars. A 'docker pull hello-world' button is visible. The main content area is divided into two columns. The left column has a 'Quick reference' section with links to the Docker Community and Slack, and a 'Supported tags and respective Dockerfile links' section. The right column has a 'Recent Tags' section listing various tags like 'nanoserver-itsc2022' and 'nanoserver-1809', and an 'About Official Images' section explaining that these images are curated and have clear documentation.

[https://hub.docker.com/\\_/hello-world/](https://hub.docker.com/_/hello-world/)

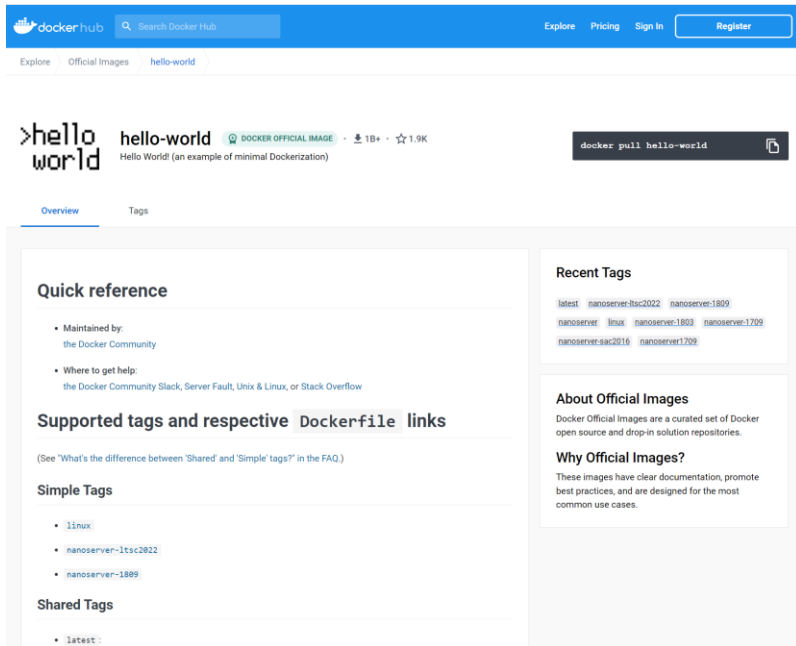
- Las **imágenes** contienen la información necesaria para ejecutar un programa dentro de un **contenedor** y se almacenan en **repositorios** que a su vez se categorizan y agrupan en **registros**.
- Las herramientas que construyen, ejecutan y administran **contenedores** necesitan acceso a las **imágenes**. Este acceso se obtiene haciendo referencia a la ruta para acceder a la imagen en el **registro**.



UNIVERSIDAD  
DE GRANADA

# DOCKER

PLATAFORMA PARA CONTENEDORES



The screenshot shows the Docker Hub interface for the 'hello-world' image. At the top, there's a blue navigation bar with the Docker Hub logo, a search bar, and links for 'Explore', 'Pricing', 'Sign In', and 'Register'. Below the navigation bar, the 'hello-world' image is featured with a 'DOCKER OFFICIAL IMAGE' badge, a pull count of '1B+', and a star count of '1.9K'. A terminal snippet shows the command 'docker pull hello-world'. The main content area is divided into two columns. The left column has tabs for 'Overview' and 'Tags', with 'Overview' selected. It contains a 'Quick reference' section with links to the Docker Community and help resources, a 'Supported tags and respective Dockerfile links' section with a link to the FAQ, and a 'Simple Tags' list including 'linux', 'nanoserver-lts2022', and 'nanoserver-1809'. The right column has a 'Recent Tags' section listing various tags like 'latest', 'nanoserver-lts2022', and 'nanoserver-1809', and an 'About Official Images' section explaining that these images are curated, open source, and designed for common use cases.

[https://hub.docker.com/\\_/hello-world/](https://hub.docker.com/_/hello-world/)

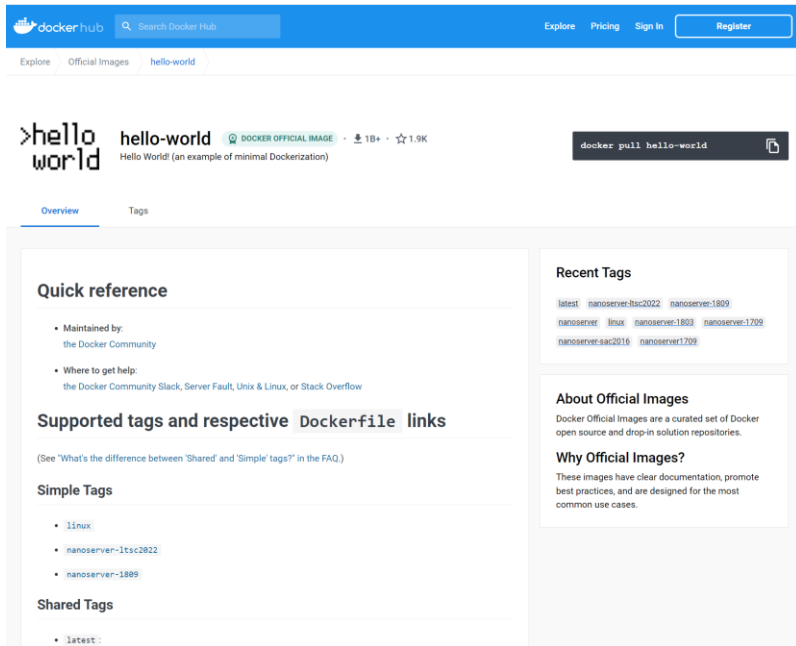
- El **entorno de ejecución de un contenedor** es el software que ejecuta aplicaciones en forma de contenedores.
- Sin el **entorno de ejecución de un contenedor**, solo se tiene la imagen del contenedor, es decir el archivo que especifica cómo es la aplicación en el contenedor.
- El **entorno de ejecución** iniciará una aplicación dentro de un contenedor y le proporcionará los recursos necesarios (CPU, almacenamiento, memoria) para que funcione.



UNIVERSIDAD  
DE GRANADA

# DOCKER

PLATAFORMA PARA CONTENEDORES



The screenshot shows the Docker Hub interface for the 'hello-world' image. At the top, there's a blue navigation bar with the Docker Hub logo, a search bar, and links for 'Explore', 'Pricing', 'Sign In', and 'Register'. Below the navigation bar, the 'hello-world' image is featured with its logo, name, and a description: 'Hello World! (an example of minimal Dockerization)'. It also shows '18+' pulls and '1.9K' stars. A 'docker pull hello-world' button is visible. The main content area is divided into sections: 'Quick reference' (listing maintainers and help links), 'Supported tags and respective Dockerfile links' (with a note about 'Shared' and 'Simple' tags), 'Simple Tags' (listing 'linux', 'nanoserver-lts2022', and 'nanoserver-1809'), and 'Shared Tags' (listing 'latest'). On the right, there's a 'Recent Tags' section listing various tags like 'nanoserver-lts2022', 'nanoserver-1809', 'nanoserver', 'linux', 'nanoserver-1803', 'nanoserver-1709', 'nanoserver-sac2016', and 'nanoserver1709'. Below that is an 'About Official Images' section explaining that these are curated, open-source images with clear documentation.

[https://hub.docker.com/\\_/hello-world/](https://hub.docker.com/_/hello-world/)

Dockerfile (sin extensión): documento de texto que contiene los comandos que un usuario podría realizar desde línea de comandos para ensamblar una imagen (es decir la secuencia de acciones que deben ejecutarse en un contenedor

<https://docs.docker.com/engine/reference/builder/>

docker-compose (.yml o .yaml): documento de texto que contiene la información necesaria para orquestar un conjunto de contenedores

<https://docs.docker.com/compose/>



UNIVERSIDAD  
DE GRANADA

# INSTALAR DOCKER

## DEMO

Podemos utilizar Docker en diferentes sistemas operativos.

En el caso de Ubuntu podemos instalar Docker con apt:

```
apt install docker.io -y
```

Docker funciona como contenedores nativos Linux, es decir, son procesos que se ejecutan en el servidor como procesos del sistema operativo, compartiendo el kernel del Linux.

También podemos instalarlo en Windows y Mac. En este caso, Docker crea una máquina virtual ligera Linux que pueden ejecutar dichos contenedores.

Una vez instalado podemos ejecutar para comprobar el estado general:

```
docker info
```



# CONSTRUIR UN CONTENEDOR

DEMO

Los contenedores se construyen a partir de una imagen. Por defecto, Docker obtiene estas imágenes del repositorio DockerHub.

Para acceder a una imagen, descargarla de Docker Hub (si no la tenemos almacenada en local) y construir un contenedor:

```
docker run <nombre_imagen>
```

Por ejemplo:

```
docker run hello-world
```



# BUSCAR IMÁGENES DE CONTENEDORES

DEMO

En el repositorio DockerHub podemos consultar una lista de miles de imágenes de contenedores preconfiguradas y listas para instalarse.

<https://hub.docker.com/>

Para buscar en los repositorios de DockerHub imágenes que coincidan con un criterio de búsqueda:

```
docker search <criterio-busqueda>
```

Por ejemplo, buscar imágenes de contenedores Ubuntu:

```
docker search ubuntu
```



# DESCARGAR IMÁGENES DE CONTENEDORES

## DEMO

Una vez que identificada la imagen que se desea usar, se descarga:

```
docker pull <nombre_imagen>
```

Docker almacena las imágenes en local, lo que permite crear nuevos contenedores en un tiempo reducido y aprovechando recursos de almacenamiento.

Para ver las imágenes almacenadas localmente:

```
docker images
```

Para borrar una imagen del almacén local:

```
docker rmi <id_imagen>
```



# EJECUTAR UN CONTENEDOR

DEMO

Los contenedores pueden ser interactivos. La combinación de los conmutadores -i y -t proporcionan un acceso interactivo del shell al contenedor.

Por ejemplo, para ejecutar un contenedor usando la imagen más reciente de Ubuntu:

```
docker run -it ubuntu
```

Una vez creado el contenedor, se puede ejecutar cualquier comando dentro del contenedor de Ubuntu.

Para salir del contenedor con exit o con Control-D.



# ADMINISTRAR CONTENEDORES

DEMO

Mostrar los contenedores activos, los que están ejecutándose:

```
docker ps
```

Ver todos los contenedores, los activos y los inactivos:

```
docker ps -a
```

Parar un contenedor:

```
docker stop <id_contenedor>
```

Iniciar un contenedor:

```
docker start <id_contenedor>
```



# ADMINISTRAR CONTENEDORES

DEMO

Borrar un contenedor:

```
docker rm <id_contenedor>
```

Información detallada de un contenedor :

```
docker inspect <id_contenedor>
```

Estadística de funcionamiento: %CPU, uso de memoria, E/S, ...

```
docker stats
```



# CREAR UNA IMAGEN DE UN CONTENEDOR

DEMO

Aunque podemos encontrar bastantes imágenes de contenedores disponibles en DockerHub es útil poder definir un contenedor propio.

Guardar la imagen de un contenedor:

```
docker commit <id_contenedor> <nombre_nueva_imagen>
```

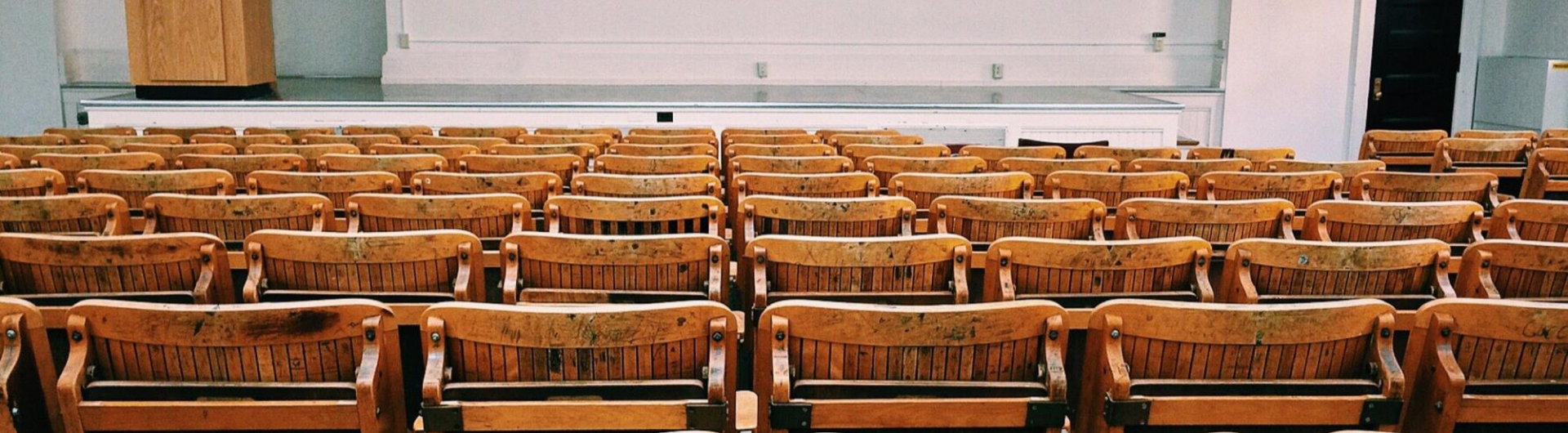
Guardar la imagen en formato .tar:

```
docker save -o <ruta_del_fichero> <nombre_imagen>
```

Recuperar la imagen de un fichero .tar:

```
docker load -i <ruta_del_fichero>
```





# GRACIAS POR SU ATENCIÓN

CONTACTO:



Claudia Villalonga Palliser

Despacho 1.3  
Edificio Auxiliar de la  
Escuela Técnica Superior de  
Ingenierías Informática y  
Telecomunicación de la  
Universidad de Granada  
18015, Granada (España)



Correo / Web

cvillalonga@ugr.es  
<https://www.ugr.es/personal/claudia-villalonga-palliser>