

# **Herramientas & Librerías**

Antonio Espín Herranz

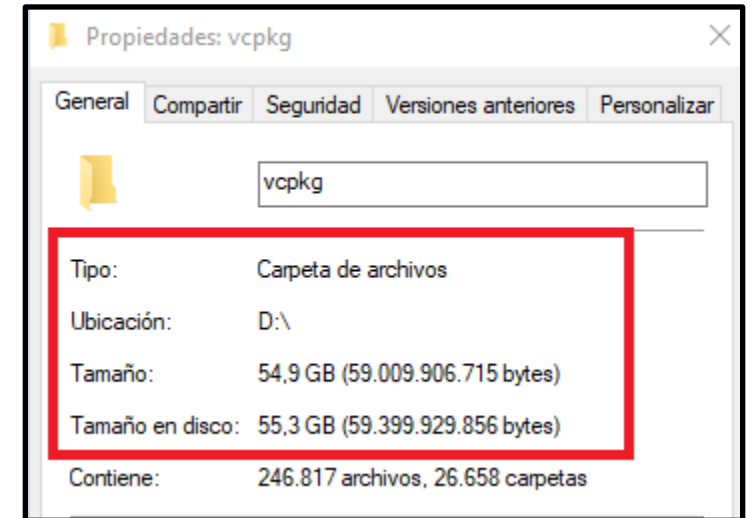
# Herramientas

- **Visual Studio Community:**  
<https://visualstudio.microsoft.com/es/downloads>
- Descarga un instalador, seleccionar desarrollo en C++
- En los detalles de la instalación tener en cuenta la versión de Windows 10 u 11
- **Cmake:** <https://cmake.org/download/>
- **Git:** <https://git-scm.com/downloads>
  - Instalar con las opciones por defecto

# Herramientas

- También se puede instalar **GitHub Desktop**
- <https://www.gitkraken.com/download>
- **vcpkg**: gestor de paquetes y librerías de C++
- Hay que clonar un repositorio de Git
- ***git clone https://github.com/microsoft/vcpkg.git***
- ***cd vcpkg***
- ***.\bootstrap-vcpkg.bat***
- Al ejecutar el bat ya veremos el .exe
- Añadir al PATH (la carpeta vcpkg)

**OJO con el espacio en disco**



***Luego se puede liberar espacio***

# vcpkg

- Una vez instalado y añadido al PATH ya se puede utilizar para instalar librerías.
  - **OJO, hay algunas que tardan poco, pero otras pueden tardar 15 o 20 min.**
- El formato (en una consola):
  - **vcpkg install librería**
- Se pueden instalar varias a la vez (***pero mejor por separado por si falla alguna***):
  - **vcpkg install librería1 librería2 ...**
- Una vez instaladas, se **integran en Visual Studio** con:
  - **vcpkg integrate install**

# Instalar librerías

- Ir a la carpeta de **vcpkg**
- Para instalar:
- **vcpkg install crow**
- Para integrar con Visual Studio (***no hay que copiar los archivos***)
- **vcpkg integrate install**
- Debería dejarnos en un proyecto:
- **#include <crow.h>**

# vcpkg

- **Problemas con la instalación de librerías**, hay veces que al instalar alguna librería podemos tener algún problema, en estos casos:
  - Ir a la carpeta donde tengamos el repositorio de vcpkg
  - Actualizar el repositorio → **git pull**
  - Actualizar el ejecutable de vcpkg (ejecutar el bat) → **bootstrap-vcpkg.bat**
  - Actualizar paquetes disponibles → **vcpkg update**
  - Reinstalar una librería que falló en la instalación →
    - **vcpkg remove librería**
    - **vcpkg install librería**

# vcpkg instalar librerías

- Desde una consola, lanzar los comandos:

**vcpkg install crow**

**vcpkg install grpc**

**vcpkg install protobuf**

**vcpkg install zeromq**

**vcpkg install simpleamqpclient**

... ..

- Con esta instrucción se integran las librerías en Visual Studio

- Se puede ejecutar después de instalar cada librería

**vcpkg integrate install**

***Lista completa de librerías  
En la siguiente página***

# vcpkg: librerías a instalar |

## **vcpkg install nombre\_libreria**

- Protocolos HTTP y webSockets → **boost-beast**
- APIs RESTful → **crow**
- Para json → **nlohmann-json**
- Comunicaciones → **grpc**
- Serializar datos (la utiliza grpc) → **protobuf**
- Comunicaciones → **zeromq**
- Para RabbitMQ → **simpleamqpclient**
- Para RabbitMQ → **cppmq**



# vcpkg: librerías a instalar II

## **vcpkg install nombre\_libreria**

- Comunicaciones → **zeromq**
- Comunicaciones → **gprc** y **protobuf**
  - Protobuf lo necesita gprc
  - Con protobuf se instala un comando llamado **protoc** y hay que añadirlo al **PATH**
  - Suele estar en esta ruta:
  - C:\vcpkg\installed\x64-windows\tools\protobuf\**protoc.exe**
  - Comprobar si se ha instalado el plugin en esta ubicación:
  - C:\vcpkg\installed\x64-windows\tools\grpc\**grpc\_cpp\_plugin.exe**
  - Si no vemos el plugin, lanzar estos comandos:
    - **vcpkg remove grpc**
    - **vcpkg install grpc --recurse**
    - **vcpkg integrate install**
- Comunicaciones → **librabbitmq**

# vcpkg: librerías a instalar III

## **vcpkg install nombre\_libreria**

- Base de datos sqlite3 → sqlite3
- Base de datos MySQL → libmysql
- Base de datos PostGreSQL → libpq
- **SOLO Si da problemas la librería de mysql: libmysql**
- Ir a la carpeta de vcpkg
  - git pull
  - vcpkg update
  - vcpkg upgrade
    - Puede dar un warning y habrá que ejecutar con:
    - vcpkg upgrade --no-dry-run
- Después de actualizar:
  - vcpkg remove libmysql
  - vcpkg remove --outdated
  - vcpkg install libmysql
  - vcpkg integrate install

# vcpkg: librerías a instalar IV

## **vcpkg install nombre\_libreria**

- Librerías **ORM**
  - (*En estas pongo el comando que hay que lanzar*)
  - **vcpkg install soci** → Con esto no se instala un backend de mysql, faltaría:
  - **vcpkg install soci[core,mysql]**
- Base de datos cliente de postgresql → **libpqxx**
- Base de datos NoSQL (mongo) → **mongo-cxx-driver**
- Base de datos clave-valor (redis) → **hiredis**

# vcpkg: librerías a instalar V

## **vcpkg install nombre\_libreria**

- JSON WEB TOKENS → **jwt-cpp**
- Cifrado, certificados, seguridad → **openssl**

# vcpkg: Liberar espacio

- Se pueden comprobar las librerías instaladas con:
- **vcpkg list**
- **Liberar espacio en vcpkg (cuando se han instalado las librerías)**
  - **Vaciar las carpetas:**
    - buildtrees (todo el contenido)
    - packages (todo el contenido)
    - downloads (solo los archivos comprimidos, **no borrar la carpeta tools**)

# Otras herramientas

- **Postman:**

- Para hacer pruebas con los servicios
- Se puede utilizar la cuenta de Google con la verificación en dos pasos.
- <https://www.postman.com/downloads/>

- **OpenSSL**

- Descargar la última versión, **añadir al PATH**
- Desde una consola podemos teclear el comando: **openssl** y comprobar que responde.
- **No instalar la versión light**
- <https://slproweb.com/products/Win32OpenSSL.html>

Overview

DEL Post data

No environment

REST API basics: CRUD, test & variable / Post data

DELETE

http://localhost:8080/items/1

Send

Params

Authorization

Headers (9)

Body

Scripts

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {
2   "value": "Estudiar Boost.Beast"
3 }
4
```

Body

Cookies

Headers (3)

Test Results (1/1)

{}

JSON

Preview

Visualize

```
1 {
2   "id": 1,
3   "message": "Elemento eliminado",
4   "status": "ok"
5 }
```

200 OK • 7 ms • 151 B • Save Response

PRUEBAS CON POSTMAN  
Durante el curso

# Bases de datos y otras herramientas

- **Utilizaremos contenedores de Docker**, esto se puede ir realizando durante el curso, suelen ser prácticas habituales con Docker.
- **Bases de datos:**
  - Mysql
  - Postgresql
  - Mongo
- **Herramientas:**
  - Prometheus, Grafana, cAdvisor, Jaeger, OpenTelemetry Collector, Fluentd, Elasticsearch
  - Utilizaremos imágenes oficiales del dockerhub, irá cada una en un contenedor de docker)