

SERVICIO CON CACHE EN DOCKER

Nos apoyamos en la práctica de: “**practica_servicio_con_cache**” hay que escribir un **docker-compose** que lance los 3 servicios necesarios: **redis** y **mysql** como los que teníamos en la otra práctica y ahora hay que añadir un contenedor que despliegue el microservicio. Escribir el **dockerfile** con todo lo necesario, copiando el código fuente, instalando las herramientas necesarias, compilando y arrancando el servicio. Probar desde el navegador a recuperar empleados.

Estructura de carpetas:

```
proyecto-empleados/
├── docker-compose.yml
├── redis/
├── mysql/
│   └── empresa3_mysql.sql
└── microservicio/
    ├── Dockerfile
    ├── src/
    ├── CMakeLists.txt
    └── vcpkg.json
```

El fichero: **vcpkg.json** indica las librerías de vcpkg que necesita el proyecto:

```
{
    "name": "microservicio-empleados",
    "version": "1.0.0",
    "dependencies": ["crow", "soci-core", "soci-mysql", "hiredis"]
}
```

El código fuente al copiarlo a una imagen de Ubuntu puede tener problemas con LF a final de la línea.

Windows: CRLF

Linux: LF

En Notepad++

Editar → Conversión fin de línea → Fin de línea Linux

El fichero CMakeLists.txt sería algo así:

```
cmake_minimum_required(VERSION 3.15)
```

```
project(microservicio-empleados)
```

```
# Activar C++17
```

```
set(CMAKE_CXX_STANDARD 17)
```

```
set(CMAKE_CXX_STANDARD_REQUIRED ON)
```

```
# Usar vcpkg como toolchain
```

```
# Esto se define en el comando de cmake, este comando va al Dockerfile
```

```
#cmake
```

```
#DCMAKE_TOOLCHAIN_FILE=/opt/vcpkg/scripts/buildsystems/vcpkg.cmake
```

```
# Añadir los ficheros del código fuente:
```

```
file(GLOB_RECURSE SOURCES "src/*.cpp")
```

```
add_executable(microservicio ${SOURCES})
```

```
# Buscar las librerías instaladas por vcpkg
```

```
find_package(Crow CONFIG REQUIRED)
```

```
find_package(SOCI CONFIG REQUIRED)
```

```
find_package(hiredis CONFIG REQUIRED)
```

```
# Enlazar las librerías al ejecutable:
```

```
target_link_libraries(microservicio PRIVATE
```

```
    Crow::Crow
```

```
    SOCI::soci_core
```

SOCI::soci_mysql

hiredis::hireredis

)