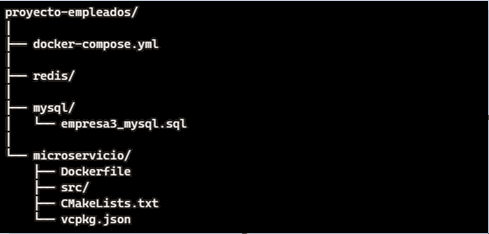
**SERVICIO CON CACHE EN DOCKER**

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:**



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

{

"name": "microservicio-empleados",

"version": "1.0.0",

"dependencies": [

{

"name": "soci",

"features": ["mysql"]

},

"hiredis",

"crow"

]

}

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 codigo fuente:

file(GLOB\_RECURSE SOURCES "src/\*.cpp")

add\_executable(microservicio ${SOURCES})

# Buscar las librerias instaladas por vcpkg

find\_package(Crow CONFIG REQUIRED)

find\_package(SOCI CONFIG REQUIRED)

find\_package(hiredis CONFIG REQUIRED)

# Enlazar las librerias al ejecutable:

target\_link\_libraries(microservicio PRIVATE

Crow::Crow

SOCI::soci\_core

SOCI::soci\_mysql

hiredis::hireredis

)