

PRACTICA 1: MICROSERVICIO CON CACHE EN REDIS Y ACCESO A BD CON MYSQL

- Tenemos dos contenedores: mysql y redis que hay que desplegar y restaurar el backup de mysql con la BD empresa3.
- El contenedor de redis actuará como cache.
- Utilizamos la clase EmpleadoRepository para las operaciones CRUD contra los empleados: id, nombre y cargo con la BD de MySQL.
- La clase RedisCache interactúa con redis con métodos: getEmpleado(id) y setEmpleado(emp) para poder recuperar y escribir un empleado en redis.
- Una clase EmpleadoService que realiza la siguiente tarea: consulta en redis si el empleado existe lo recupera de redis, si no, tiene que ir a MySQL a recuperarlo.
- Implementar un servicio en Crow que permita la recuperación de empleados, se apoyara en EmpleadoService.
- Proyecto en C++ 17
- Probar el servicio desde PostMan. Conectar al contenedor de Docker (redis) y comprobar si se han creado las claves que hemos solicitado (keys *)

- **CLASES:**
 - Empleado: puede ser una estructura: id: int, nombre: string, cargo:string
 - EmpleadoRepository: La clase que proporciona las operaciones CRUD con la BD de MySQL
 - EmpleadoCache: Las operaciones con Redis, para recuperar y grabar.
 - EmpleadoService: Capa intermedia, lógica de negocio. Utiliza el repositorio y la caché para comprobar si un empleado esta o no en la cache. Mantiene atributos de la cache y el repositorio.
 - MicroServicioCROW: Mantiene un atributo de EmpleadoService, publica una operación GET en /empleado/<int>
 - Main: Crear los objetos necesarios: persistencia, servicios, microservicio y lo pone en marcha.