

Acuario

Fase 2: Integración

Checkpoints

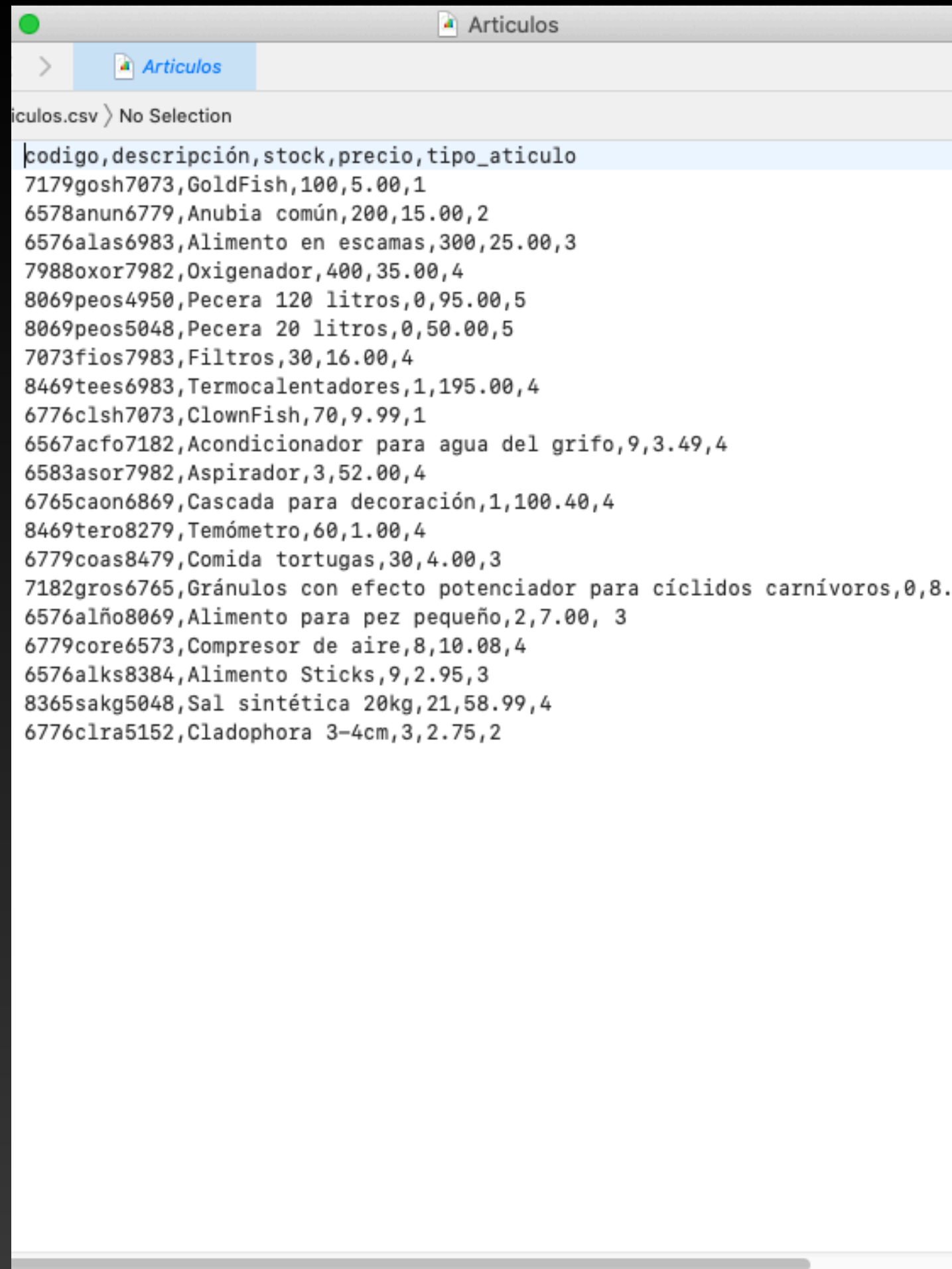
Mejoras y correcciones pendientes

- Mejorar la documentación del proyecto (readme.md, javadoc, etc.)
- Los objetos del sistema deben estar identificados por códigos únicos, no correlativos y formados por letras y números
- Presentación de las facturas completas
- Almacenar de forma persistente los datos (propuesta de la profesora: uso de ficheros)

IDs de Objetos

Creados mediante patrones y expresiones regulares

- Especies del acuario
- Estanques y salas
- Artículos de la tienda
- Clientes
- Trabajadores



codigo	descripción	stock	precio	tipo_aticulo
7179gosh7073	GoldFish	100	5.00	1
6578anun6779	Anubia común	200	15.00	2
6576alas6983	Alimento en escamas	300	25.00	3
7988oxor7982	Oxigenador	400	35.00	4
8069peos4950	Pecera 120 litros	0	95.00	5
8069peos5048	Pecera 20 litros	0	50.00	5
7073fios7983	Filtros	30	16.00	4
8469tees6983	Termocalentadores	1	195.00	4
6776clsh7073	ClownFish	70	9.99	1
6567acfo7182	Acondicionador para agua del grifo	9	3.49	4
6583asor7982	Aspirador	3	52.00	4
6765caon6869	Cascada para decoración	1	100.40	4
8469tero8279	Temómetro	60	1.00	4
6779coas8479	Comida tortugas	30	4.00	3
7182gros6765	Gránulos con efecto potenciador para cíclidos carnívoros	0	8.00	8
6576alño8069	Alimento para pez pequeño	2	7.00	3
6779core6573	Compresor de aire	8	10.08	4
6576alks8384	Alimento Sticks	9	2.95	3
8365sakg5048	Sal sintética 20kg	21	58.99	4
6776clra5152	Cladophora 3-4cm	3	2.75	2

Presentar Facturas Completas

- El método imprimir vuelca la factura en un fichero de texto
- El nombre del fichero se corresponde con el número de factura

```
Factura.java
a.java > No Selection

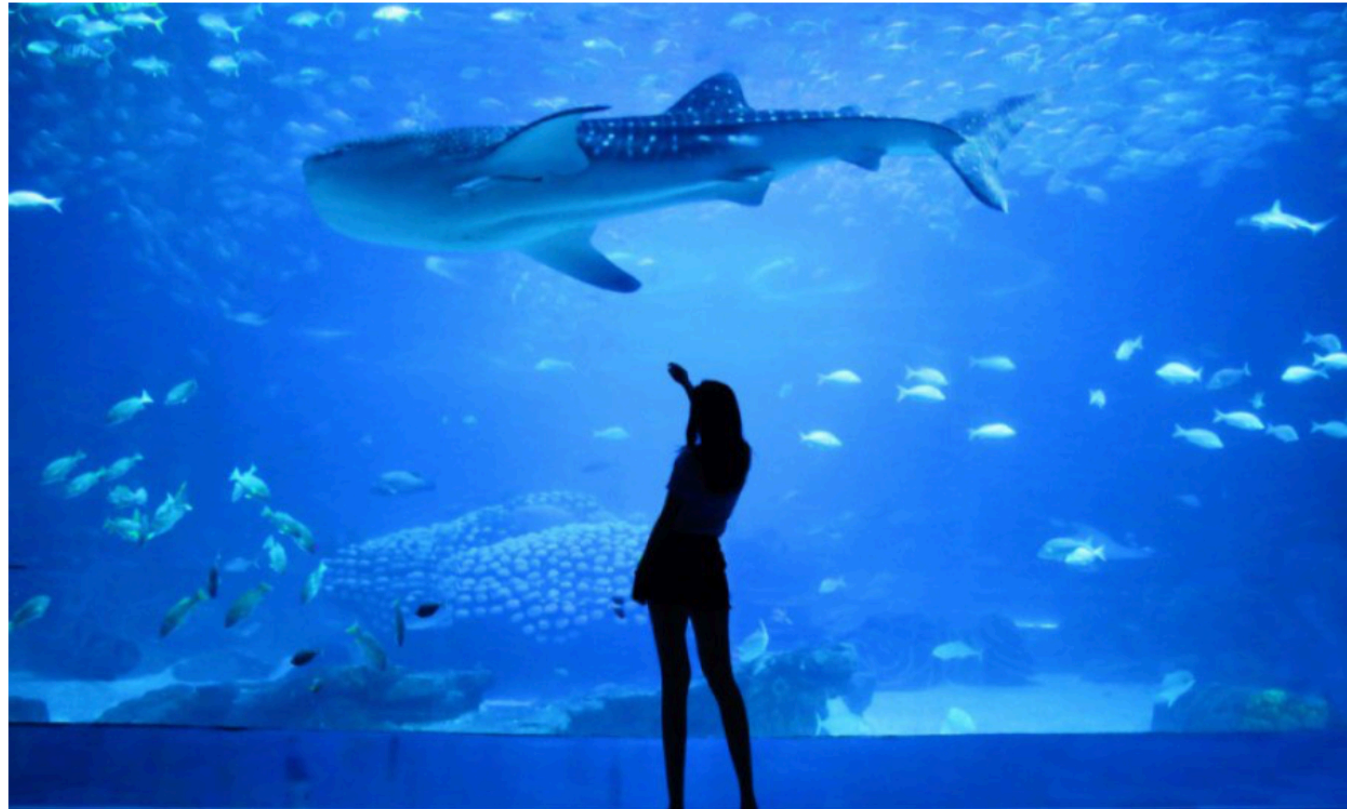
public void imprimir() {
    System.out.println("Imprimiendo factura...");
    String nombreFichero=numero+".txt";
    String fichero = new File(nombreFichero).getAbsolutePath();
    DecimalFormat df = new DecimalFormat("#.00");
    try {
        FileWriter fw=new FileWriter(fichero, false);
        // Cabecera
        fw.write("#####\n");
        fw.write("# FACTURA #\n");
        fw.write("#####\n");
        fw.write("## Número de factura: "+numero+"\n");
        fw.write("## Tienda Acuario (CIF A11199938H)\n");
        fw.write("## Fecha: "+fecha+"\n");
        fw.write("#####\n");
        // Datos del cliente
        ConnDB bbdd=ConnDB.getInstance();
        Cliente c=bbdd.getClienteByCodigo(codigoCliente);
        fw.write("## Código Cliente: "+c.getCodigo()+"\n");
        fw.write("## Nombre Cliente: "+c.getNombre()+"\n");
        fw.write("## Direccion: "+c.getDireccion()+"\n");
        fw.write("## Teléfono: "+c.getTelefono()+"\n");
        fw.write("#####\n");
        // Lineas de la factura
        fw.write("## Cantidad -- Descripcion -- Precio\n");
        fw.write("-----\n");
        // Recorremos el array para obtener los datos de cada linea
        for(LineaFactura lf : listaLineasFactura){
            String cantidad=String.format("%11d",lf.getCantidad());
            String descripcion=String.format("%-30s",lf.getDescripcion());
            String precio=String.format(df.format(lf.getPrecio()));
            fw.write(cantidad+" -- "+descripcion+" -- "+precio+"\n");
        }
        // Desglose precio, impuestos e importe
        fw.write("-----\n");
        double precio = importeTotal/1.21;
        double impuestos = importeTotal-precio;
        fw.write("## Importe (antes de impuestos) "+df.format(precio)+"\n");
        fw.write("## Impuestos (21%) "+df.format(impuestos)+"\n");
        fw.write("## IMPORTE TOTAL "+df.format(importeTotal)+"\n");
        // Observaciones
        fw.write("#####\n");
        fw.write("## Observaciones\n");
        fw.write("## \n");
        fw.write("#####\n");
        fw.close();

        System.out.println("Factura guardada como "+fichero);
    }catch(IOException ioe){
        System.out.println(ioe.getMessage());
    }
}
```

Documentación

Readme (en Github) con la explicación del problema y la solución implementada

Comentarios y Javadoc en el código



peixePOP

El repositorio se ha creado para poder desarrollar un trabajo de la asignatura Acceso a Datos. Se puede clonar y examinar el código. Si los cambios se consideran oportunos, se hará merge con la rama principal.

Integrantes:

- [Emilio Devesa](#)
- [Yudaisy Ramos](#)
- [Rocío Seoane](#)
- [Gianny Uffo](#)
- [Paulo Vazquez](#)
- [Angel Andrade](#)
- [Miguel Jaimes](#)
- [José Bayona](#)

Problema a resolver.

En la ciudad en la que reside nuestra empresa, se está construyendo un complejo multidisciplinario en la zona norte de la península, teniendo como eje central grandes acuarios. Será necesario la gestión de los datos del personal, de los clientes y de la distribución de animales(peces) y plantas en cuanto a la ubicación de sus respectivos estanques y sus salas.

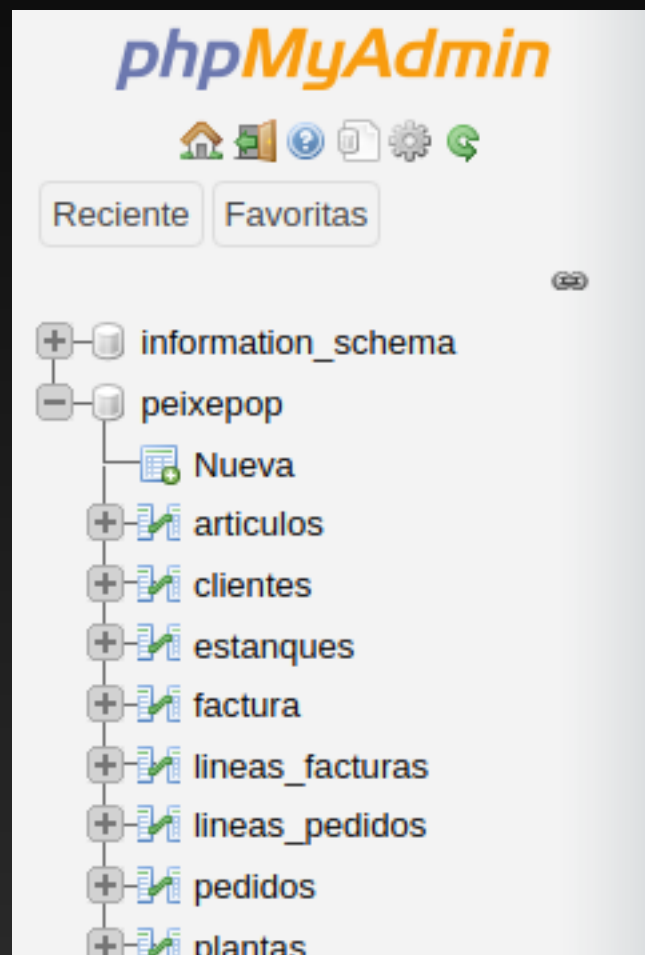
A mayores, en uno de sus locales, se establecerá una tienda de peces y alimentación de animales acuáticos, artículos de acuariofilia y mantenimiento de los mismos. Tras varias horas charlando acerca de los problemas que pueden surgir, y dado el manejo del inventario de especies y el gran volumen de ventas que se prevé, se nos ha encargado la implementación de un software de gestión de personal, inventario, ventas y finanzas en

Almacenamiento persistente de datos

Un paso más allá

Infraestructura BBDD

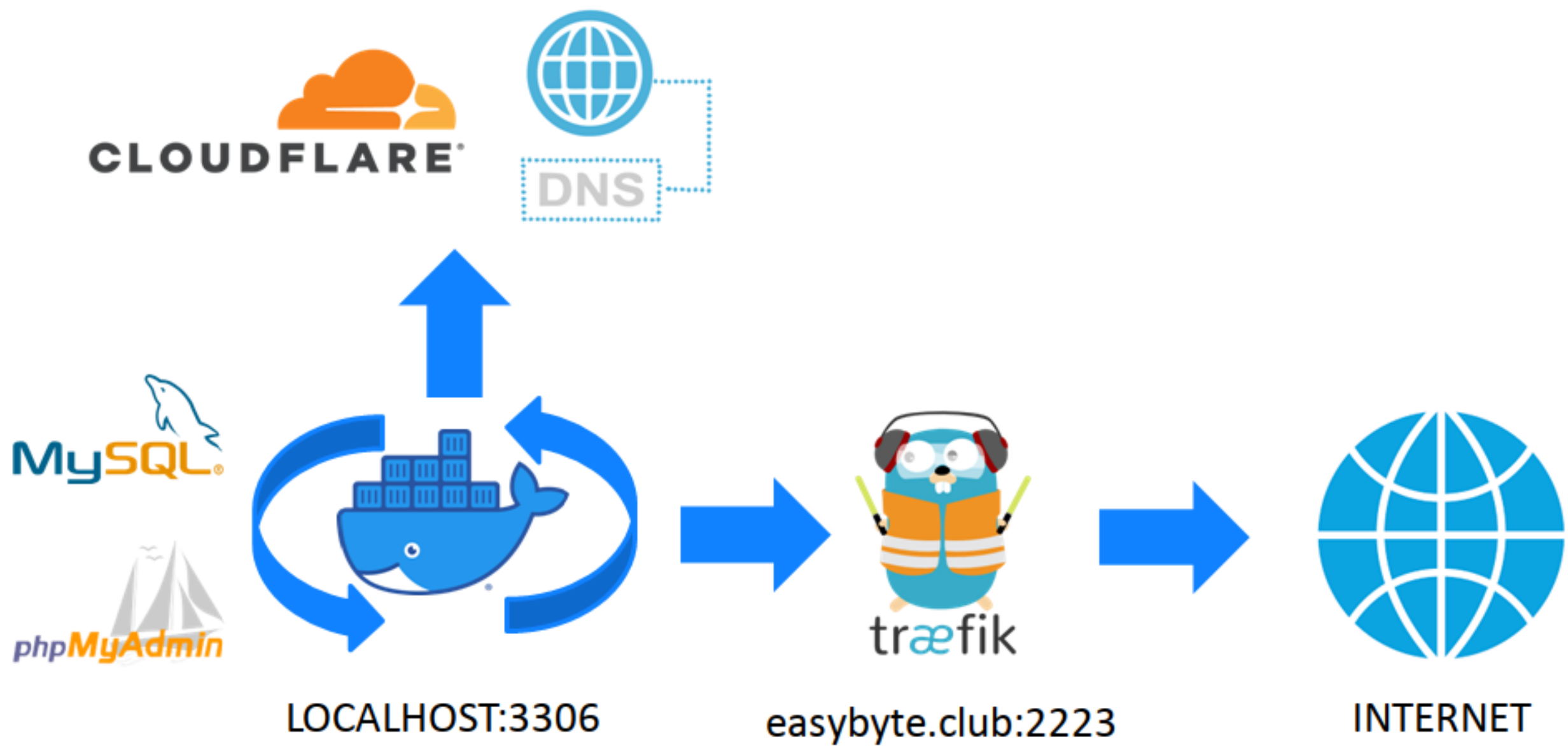
Docker + MySQL + PHPMYAdmin



```
GNU nano 5.9 docker-compose.yml
services:
  mysql:
    image: mysql:latest
    container_name: mysql-java
    restart: unless-stopped
    networks:
      traefik:
        ipv4_address: 172.19.0.20
    ports:
      - "2003:3306"
    environment:
      - MYSQL_ROOT_PASSWORD=${MYSQL_ROOT_PASS}
      - MYSQL_DATABASE=${MYSQL_DB_NAME}
      - MYSQL_USER=${MYSQL_USERNAME}
      - MYSQL_PASSWORD=${MYSQL_USERPASS}
    volumes:
      - ${DOCKERDATA_MYSQL}:/var/lib/mysql
    labels:
      - "traefik.enable=true"
      - "traefik.tcp.routers.mysql-java.rule=HostSNI(`*`)"
      - "traefik.tcp.routers.mysql-java.entrypoints=mysql-java"
      - "traefik.tcp.routers.mysql-java.service=mysql-java-svc"
      - "traefik.tcp.services.mysql-java-svc.loadbalancer.server.port=3306"
      - "com.centurylinklabs.watchtower.enable=true"

  phpmyadmin:
    image: phpmyadmin/phpmyadmin:latest
    container_name: phpmyadmin-java
    restart: unless-stopped
    networks:
      traefik:
        ipv4_address: 172.19.0.21
```

Containers							
<div>Start Stop Kill Restart Pause Resume Remove</div>							
Search...							
<input type="checkbox"/> Name	State Filter	Quick actions	Stack	Image	Created	IP Address	Published Ports
<input type="checkbox"/> phpmyadmin-java	running	📄 🔍 🏠 ➤	mysql	phpmyadmin/phpmyadmin:latest	2021-11-11 23:59:05	172.19.0.21	🔗 2002:80
<input type="checkbox"/> mysql-java	running	📄 🔍 🏠 ➤	mysql	mysql:latest	2021-11-11 23:59:04	172.19.0.20	🔗 2003:3306



Docker +
MySQL+PhpMyAdmin



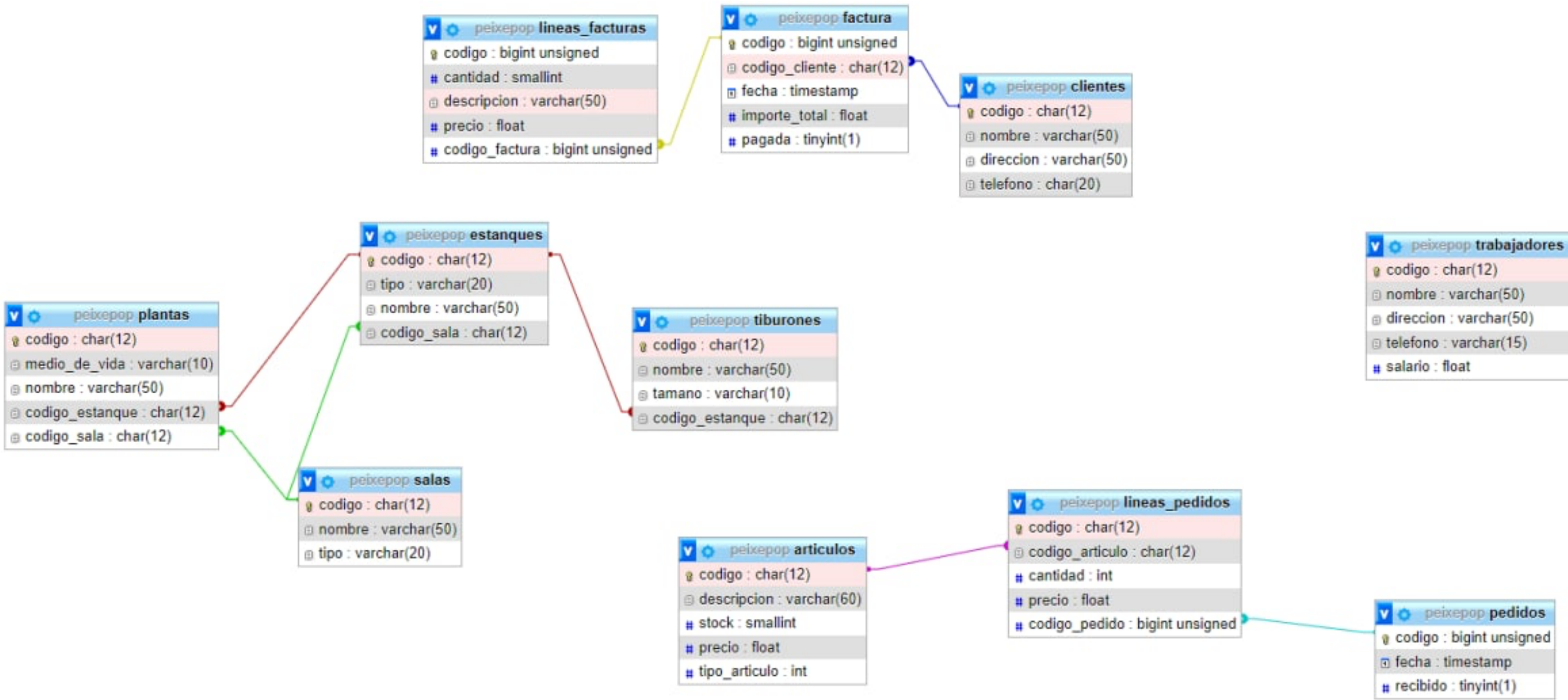
Cloudflare DNS



Traefik Proxy Inverso



<< Es muy fácil, cualquiera podría hacerlo >>
Miguel Jaimes, 2021

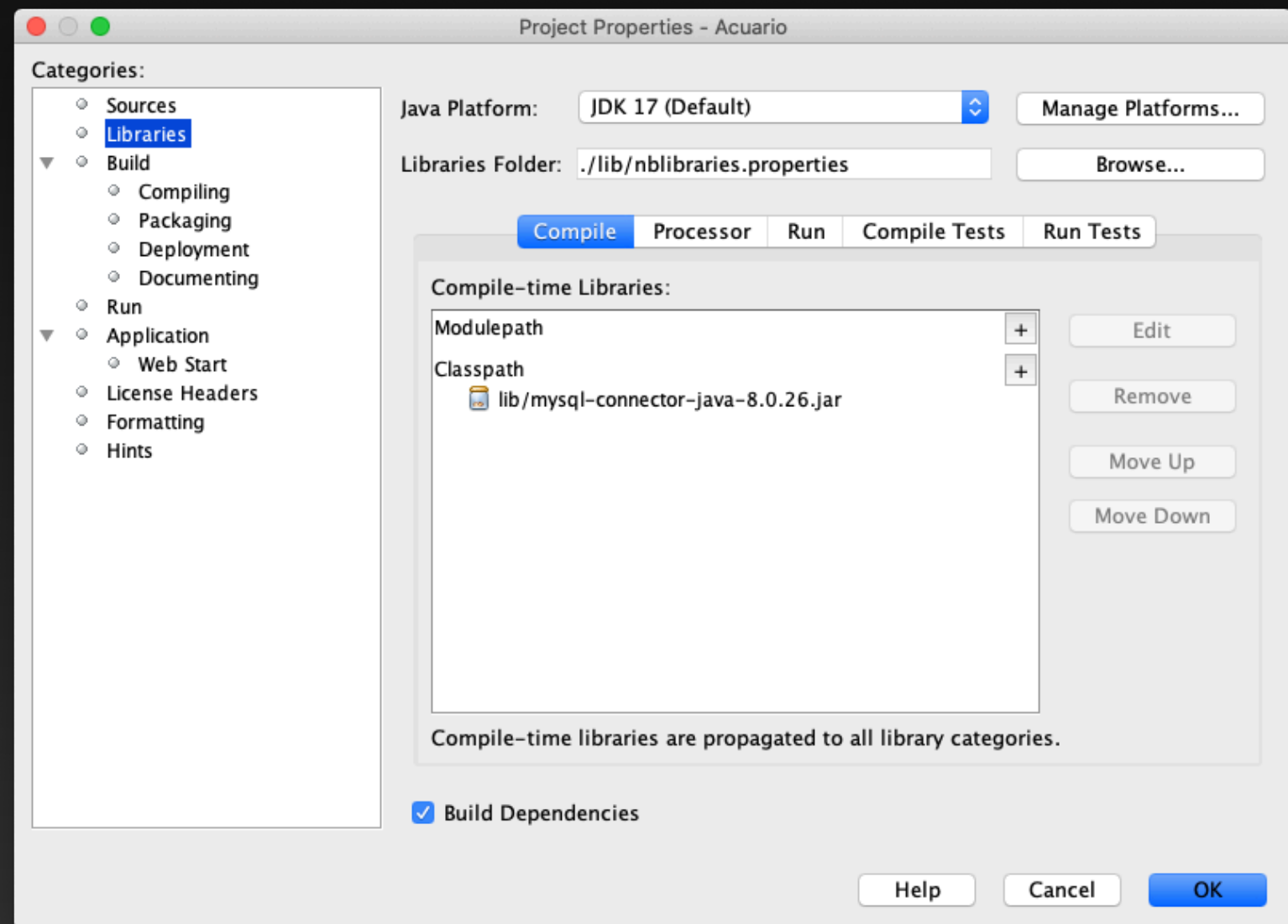
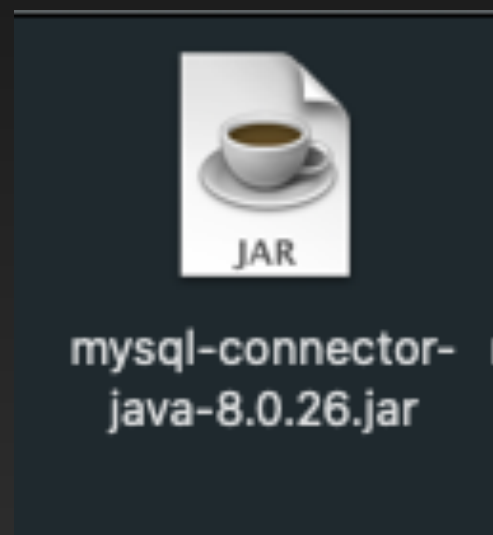


MySQL JDBC

Conectando el código con la BBDD

MySQL Connector

Es una librería que se proporciona empaquetada como JAR. Debe importarse a las librerías incluidas en el proyecto



Creación de una API para consultar la BBDD

```
public class ConnDB {  
  
    private static ConnDB instanciaUnica;  
    private static Connection conn;  
    private ResultSet rs;  
  
    /**  
     * Constructor para conectarse la base de datos por defecto  
     */  
    private ConnDB() {  
        try {  
            conn = DriverManager.getConnection("jdbc:mysql://ea  
            System.out.println(">>>>>>>>>> La conexión a la Ba  
        } catch (SQLException e) {  
            System.out.println(e.getMessage());  
            System.out.println(">>>>>>>>>> No se puede conecta  
        }  
    }  
  
    public static ConnDB getInstance(){  
        if (instanciaUnica==null){  
            instanciaUnica=new ConnDB();  
        }  
        return instanciaUnica;  
    }  
}
```

Una vez establecida esta conexión, se pueden crear métodos que ejecuten Querys y almacenen los resultados en la estructura ResultSet

