

Lenguaje	Herramienta que implementa ORM	ORM	Framework (si existe)	Anotaciones de interés
Java	JPA	Hibernate	Spring Framework	<ul style="list-style-type: none"> - Soporte completo para OO: - Hibernate permite extensiones avanzadas.
Python	Django ORM	SQLAlchemy	Django, Flask	<ul style="list-style-type: none"> - Django ORM facilita el desarrollo rápido. - SQLAlchemy es más flexible y potente.
Ruby	ActiveRecord	ActiveRecord	Ruby on Rails	<ul style="list-style-type: none"> - ActiveRecord está integrado en Rails.
C#	Entity Framework	Entity Framework	.NET Core, ASP.NET	<ul style="list-style-type: none"> - EF soporta LINQ para consultas intuitivas.
C++	ODB	ODB	N/A	<ul style="list-style-type: none"> - ODB es un ORM nativo para C++.
JavaScript	Sequelize, TypeORM	Sequelize, TypeORM	Express.js, NestJS	<ul style="list-style-type: none"> - Sequelize usa Promises y soporta múltiples DB. - TypeORM es compatible con Ts y JS.
PHP	Doctrine ORM	Doctrine	Symfony, Laravel	<ul style="list-style-type: none"> - Doctrine reduce significativamente la cantidad de SQL a escribir.
Kotlin	Exposed	Hibernate	Spring Boot, Ktor	<ul style="list-style-type: none"> - Exposed es una herramienta de ORM bastante ligera. - Es compatible con muchas librerías.

Ejemplos de uso:

1. Hibernate:

```
public static void insertaEmp(Empleados emp){
    Session session = null;
    try{
        //obtiene una sesion
        SessionFactory sf = HibernateUtil.getSessionFactory();
        //abre la sesion
        session = sf.openSession();
        //comienza una transaccion
        Transaction tx = session.beginTransaction();
        System.out.println("Inserta un empleado en la tabla Empleados");
        session.save(emp);
        tx.commit();
    } catch (HibernateException hex){
        System.err.println("Error " + hex.getCause() + ", " + hex.getMessage());
    } finally {
        session.close();
    }
}
```

Figura 1: Ejemplo de Hibernate

En este caso, utilizaría Hibernate para insertar un nuevo empleado en la base de datos.

Fuente: <https://radw2020.github.io/2017/01/18/Hibernate-Teor%C3%ADa-y-Ejemplos/>

2. SQLAlchemy:

```
1 from flask import Flask
2 from flask_sqlalchemy import SQLAlchemy
3
4 app = Flask(__name__)
5 app.config["SQLALCHEMY_DATABASE_URI"] = "sqlite:///tmp/test.db"
6 db = SQLAlchemy(app)
```

Figura 2: Ejemplo de SQLAlchemy: Importación de librerías

En este ejemplo primero importaría la librería Flask para poder utilizar el ORM y luego establece la conexión a la base de datos con unos parámetros.

```
1 class Person(Base):
2     __tablename__ = "person"
3
4     # Aquí definimos el nombre de la tabla "Person"
5     # Ten en cuenta que cada columna es también un atributo normal de primera instancia de Python.
6     id = Column(Integer, primary_key = True)
7     name = Column(String(250), nullable = False)
8
9     # El método serialize convierte el objeto en un diccionario
10    def serialize(self):
11        return {
12            "id": self.id,
13            "name": self.name
14        }
```

Figura 3: Ejemplo de SQLAlchemy: Creación de base de datos

Luego, crea una tabla en la base de datos (En este caso Person) y define sus columnas.

Fuente: https://4geeks.com/es/lesson/todo-lo-necesario-para-empezar-usar-sqlalchemy#1from-flask-import-flask-2from-flask_sqlalchemy-import-sqlalchem

```
$ bin/rails generate migration CreateBooks title:string author:string
```

and results in this:

```
# Note:
# The `id` column, as the primary key, is automatically created by convention.
# Columns `created_at` and `updated_at` are added by `t.timestamps`.

# db/migrate/20240220143807_create_books.rb
class CreateBooks < ActiveRecord::Migration[8.0]
  def change
    create_table :books do |t|
      t.string :title
      t.string :author

      t.timestamps
    end
  end
end
```

Figura 4: Ejemplo de ActiveRecord

En este ejemplo se puede ver como mediante una migración (el primer código) genera una tabla con sus columnas.

Fuente: https://guides.rubyonrails.org/active_record_basics.html

4. Entity Framework:

```
1 namespace MyApp.Models
2 {
3     public class User
4     {
5         public int Id { get; set; }
6         [Required] public string UserName { get; set; }
7         [Required] public string Password { get; set; }
8     }
9 }
```

Figura 5: Ejemplo de Entity Framework

En este ejemplo, utilizan Entity Framework para crear una tabla User con los atributos Username y Password

Fuente: <https://rlbisbe.net/2011/12/13/que-es-un-orm-y-por-que-nos-interesa/>

5. Sequelize:

2. DEFINE MODELS

```
import { Sequelize, DataTypes } from 'sequelize';

const sequelize = new Sequelize('sqlite::memory:');
const User = sequelize.define('User', {
  username: DataTypes.STRING,
  birthday: DataTypes.DATE,
});
```

Defining Models →

Figura 6: Ejemplo de Sequelize

En este ejemplo, se crea una tabla User con los atributos username y birthday.

Fuente: <https://sequelize.org/>

6. ODB:

```
class person
{
  ...
private:

  string email_;

  string name_;
  unsigned short age_;
};
```

```
#pragma db object
class person
{
  ...
private:
  friend class odb::access;
  person () {}

  #pragma db id
  string email_;

  string name_;
  unsigned short age_;
};
```

Figura 7: Ejemplo de ODB

En la parte izquierda se nos muestra un ejemplo de código de una clase en C++, a la derecha sería el molde utilizando ODB.

Fuente: <https://www.codesynthesis.com/products/odb/>

7. Doctrine ORM Symfony:

```
1 $ php bin/console make:entity
2
3 Class name of the entity to create or update:
4 > Product
5
6 New property name (press <return> to stop adding fields):
7 > name
8
9 Field type (enter ? to see all types) [string]:
10 > string
11
12 Field length [255]:
13 > 255
14
15 Can this field be null in the database (nullable) (yes/no) [no]:
16 > no
17
18 New property name (press <return> to stop adding fields):
19 > price
20
21 Field type (enter ? to see all types) [string]:
22 > integer
23
24 Can this field be null in the database (nullable) (yes/no) [no]:
25 > no
26
27 New property name (press <return> to stop adding fields):
28 >
29 (press enter again to finish)
```

Figura 8: Ejemplo de Symfony: Creación de entidad

En este ejemplo podemos ver que utilizando el comando `$ php bin/console make:entity` nos irá preguntando paso a paso para ir creando la entidad.

```

1  // src/Entity/Product.php
2  namespace App\Entity;
3
4  use App\Repository\ProductRepository;
5  use Doctrine\ORM\Mapping as ORM;
6
7  #[ORM\Entity(repositoryClass: ProductRepository::class)]
8  class Product
9  {
10     #[ORM\Id]
11     #[ORM\GeneratedValue]
12     #[ORM\Column]
13     private ?int $id = null;
14
15     #[ORM\Column(length: 255)]
16     private ?string $name = null;
17
18     #[ORM\Column]
19     private ?int $price = null;
20
21     public function getId(): ?int
22     {
23         return $this->id;
24     }
25
26     // ... getter and setter methods
27 }

```

Figura 9: Ejemplo de Symfony: Entidad creada

Una vez hecho el paso anterior, se generará un archivo .php con una clase creada.

Fuente: <https://symfony.com/doc/current/doctrine.html>