

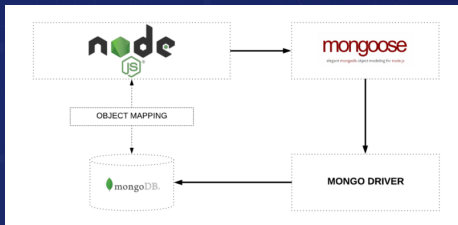
ORM y ODM en MySQL y MongoDB

Julian F. Latorre

Julio 2024

Conceptos Clave

- ORM: Object-Relational Mapping
- ODM: Object-Document Mapping
- Mapeo
- Esquema
- Modelo



ORM en MySQL

- Mapea tablas a clases
- Filas a objetos
- Columnas a atributos

Ejemplo de ORM (Sequelize)

```
const User = sequelize.define('User', {  
  firstName: {  
    type: DataTypes.STRING,  
    allowNull: false  
  },  
  lastName: {  
    type: DataTypes.STRING  
  }  
});
```

```
const newUser = await User.create({  
  firstName: 'Pepito',  
  lastName: 'Perez'  
});
```

ODM en MongoDB

- Mapea colecciones a clases
- Documentos a objetos
- Campos a atributos

Ejemplo de ODM (Mongoose)

```
const userSchema = new mongoose.Schema({
  firstName: { type: String, required: true },
  lastName: String,
  email: { type: String, required: true }
});

const User = mongoose.model('User', userSchema);

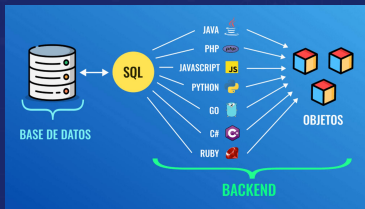
const newUser = await User.create({
  firstName: 'Pepito',
  lastName: 'Perez',
  email: 'Pepito@example.com'
});
```

ORM vs ODM

ORM	ODM
Bases de datos relacionales	Bases de datos de documentos
Tablas	Colecciones
Filas	Documentos
Columnas	Campos

Implementación en React

- React se comunica con el backend a través de API
- El backend utiliza ORM/ODM para interactuar con la base de datos



Ventajas de ORM/ODM

- Abstracción de la base de datos
- Código más limpio y mantenible
- Mejora la productividad del desarrollador
- Proporciona características como validación, migraciones y seeding