

Comparación: Acceso manual a base vs Hibernate con Spring Boot

Codigo tradicional (JDBC manual)

```
public List<Articulo> buscarArticulos() {
    List<Articulo> lista = new ArrayList<>();
    String sql = "SELECT * FROM articulo";

    try (Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/articulos_db", "root", "");
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery(sql)) {

        while (rs.next()) {
            Articulo a = new Articulo();
            a.setId(rs.getLong("id"));
            a.setNombre(rs.getString("nombre"));
            a.setPrecio(rs.getDouble("precio"));
            lista.add(a);
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }

    return lista;
}
```

Codigo con Hibernate + Spring Data JPA

```
@Autowired
private ArticuloRepository articuloRepository;

public List<Articulo> buscarArticulos() {
    return articuloRepository.findAll();
}
```

Comparacion entre ambos enfoques

1. JDBC manual:
 - Mas lineas de codigo

Comparación: Acceso manual a base vs Hibernate con Spring Boot

- Se necesita gestionar conexiones, SQL, errores
- Mas propenso a errores
- Poco reutilizable

2. Hibernate con Spring Data JPA:

- Muy pocas lineas de codigo
- No escribis SQL
- Spring gestiona errores, conexion, queries
- Ideal para CRUD y consultas comunes

Conclusion

Hibernate con Spring Boot ahorra tiempo, mejora la legibilidad y evita errores comunes. Solo deberias usar JDBC manual cuando necesitas rendimiento extremo o logica de muy bajo nivel. Para la mayoria de las aplicaciones empresariales, Spring Data JPA es la mejor opcion.