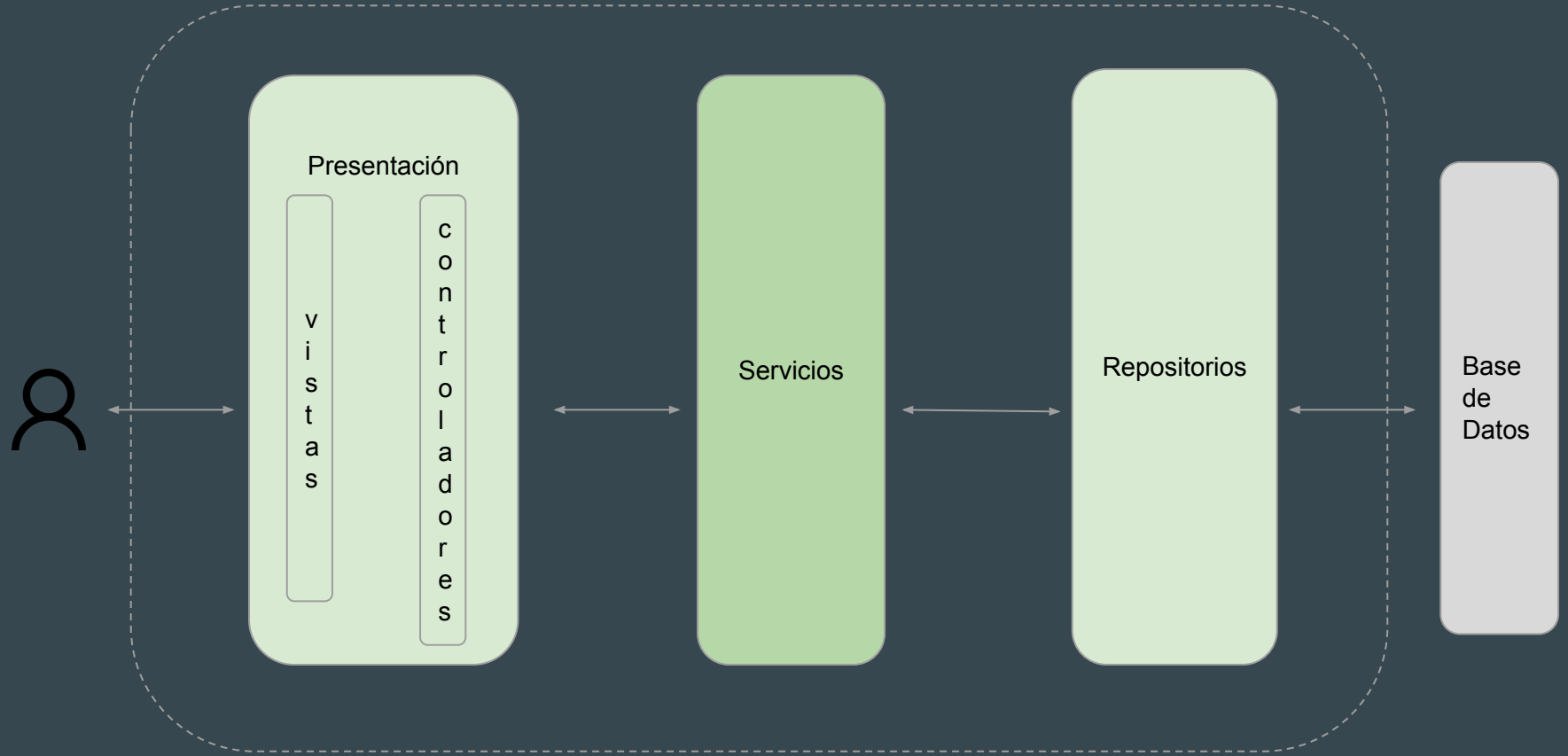
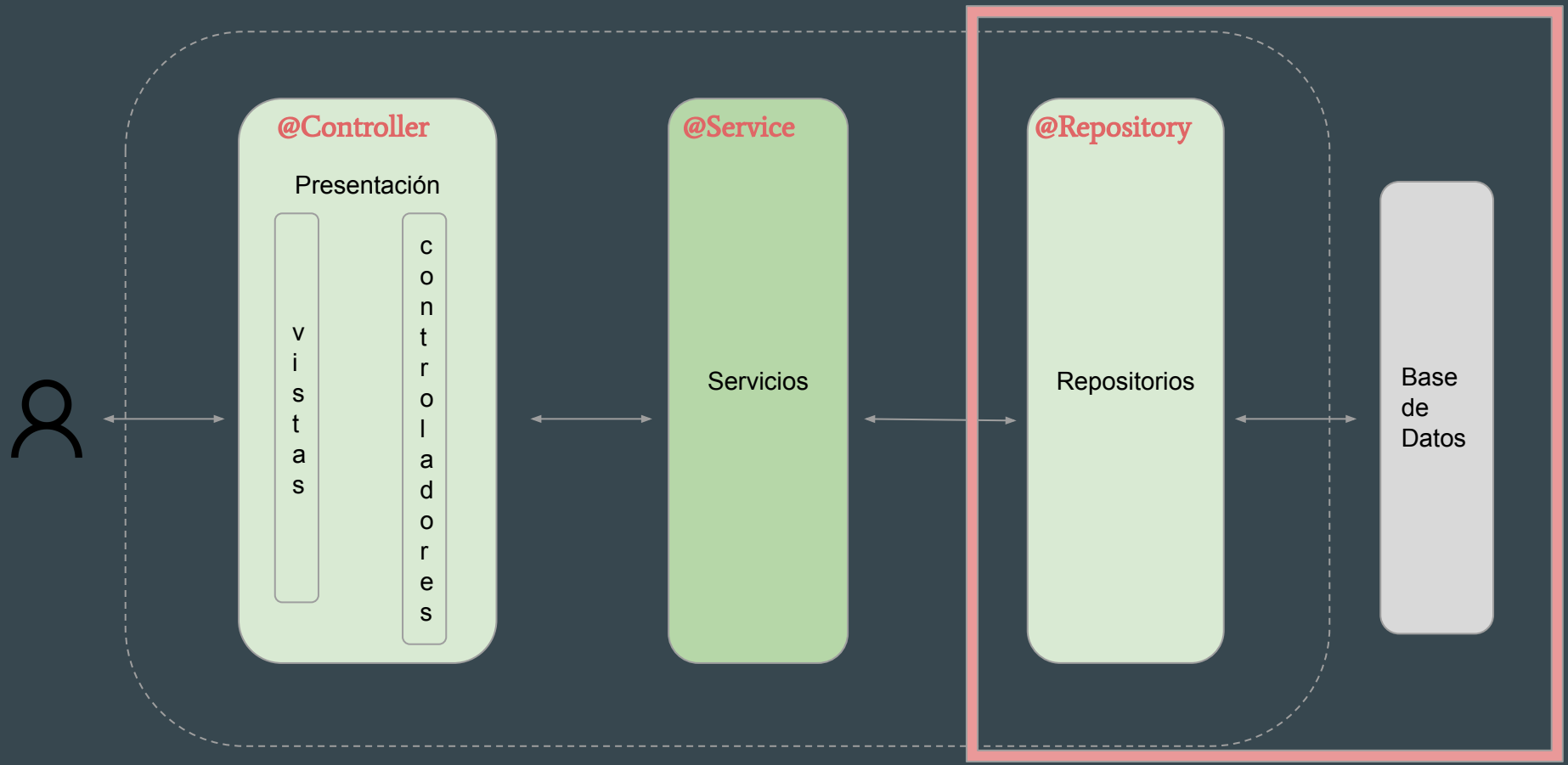
Persistencia

Taller Web I


Presentación
vistas
controladores
Servicios
Repositorios
Base de Datos

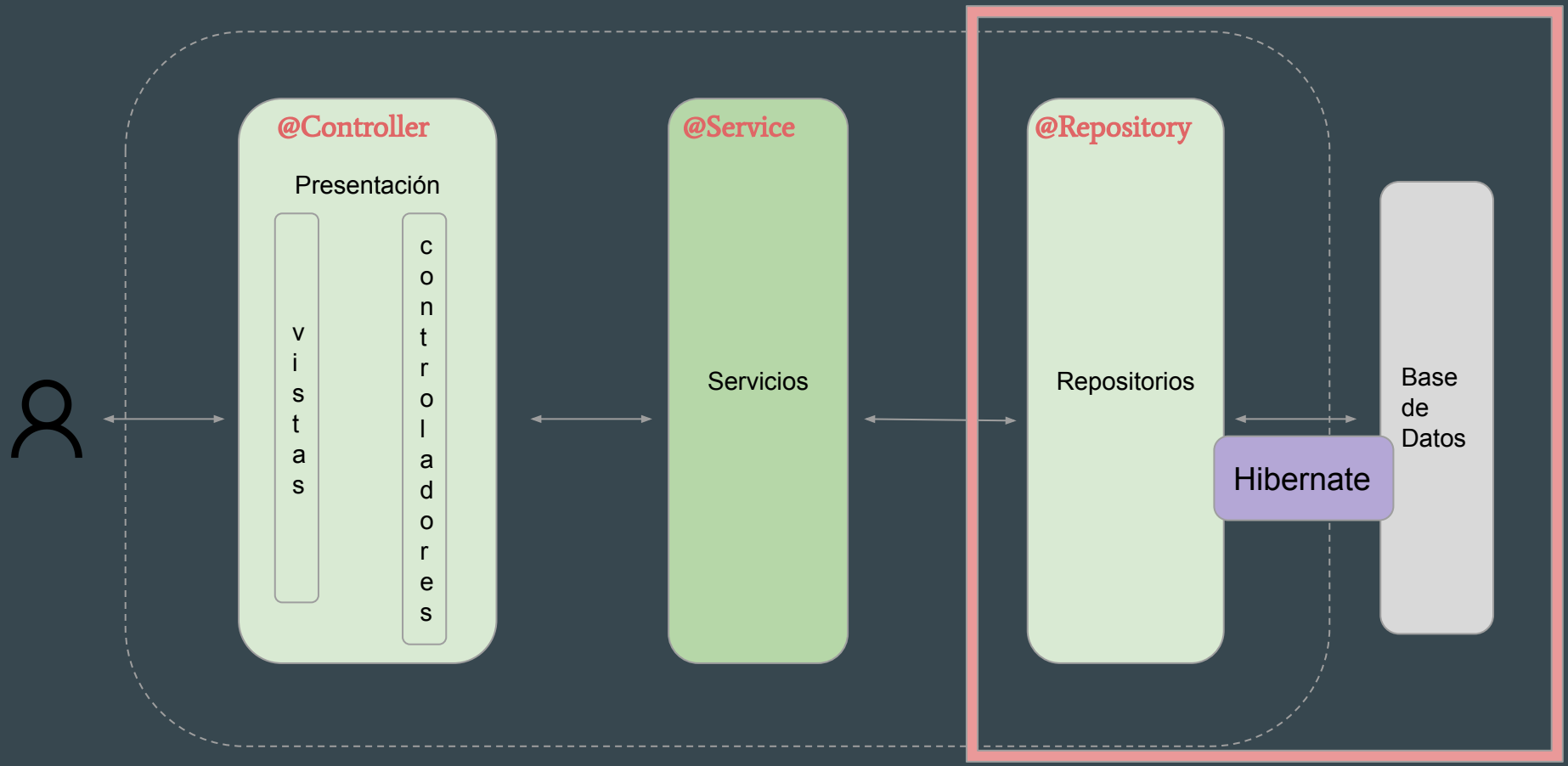
Spring MVC

¿Como le hablo a la base de datos?

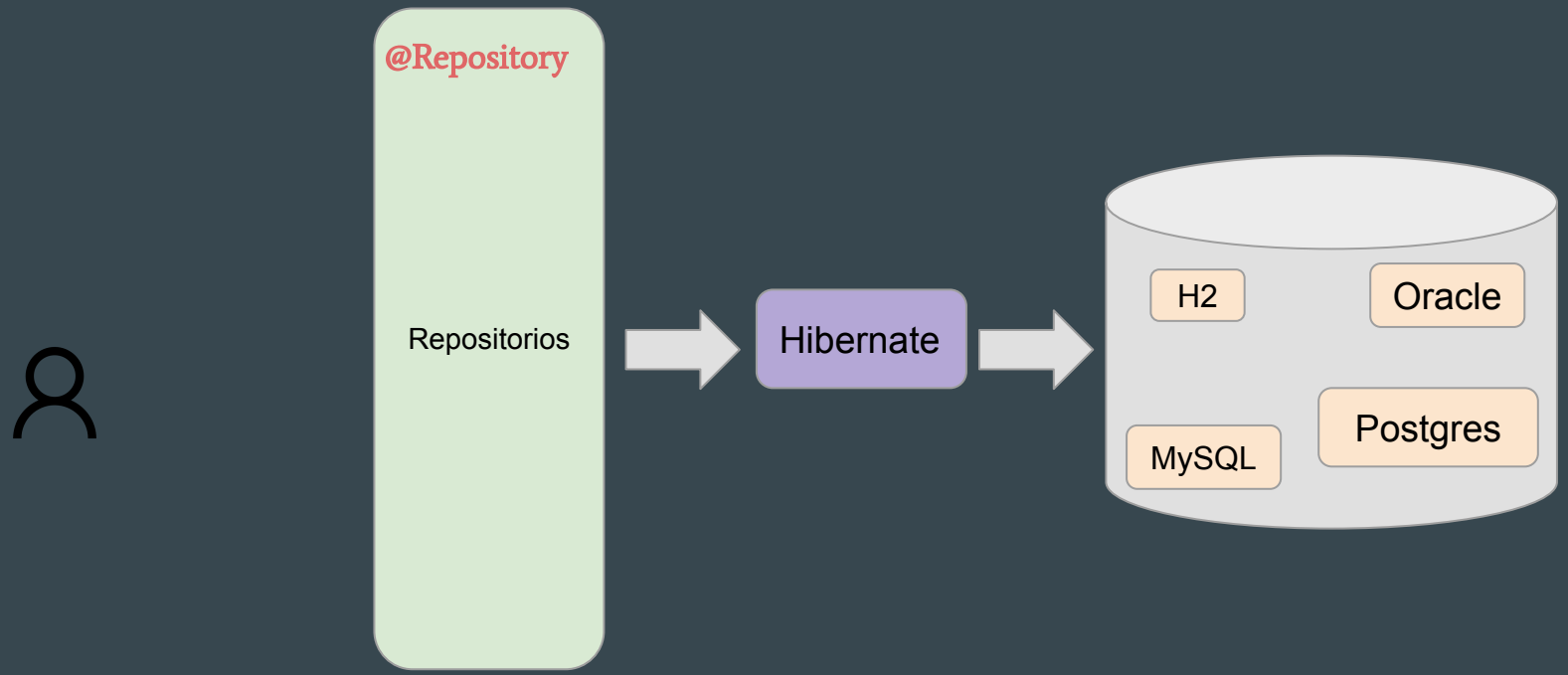


Spring MVC

¿Como le hablo a la base de datos?



Spring MVC


@Repository
Repositorios
Hibernate
H2
Oracle
MySQL
Postgres

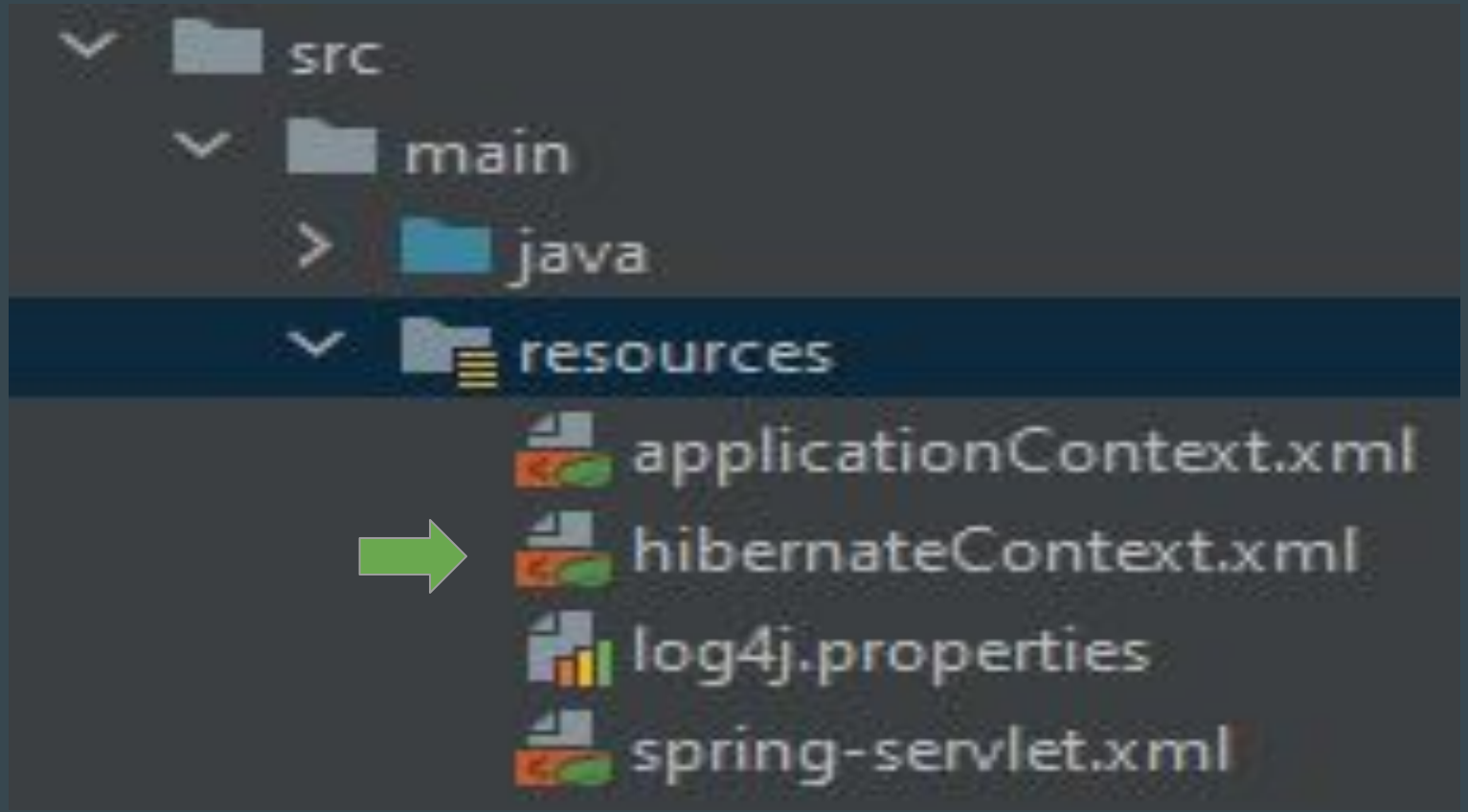
Hibernate

•••

Hibernate es un ORM para soluciones Java que permite mapear datos entre un modelo de objetos y un modelo relacional.

ORM: Object / Relational Mapping

¿Cómo lo configuro?



resources/hibernateContext.xml

```
<bean id="dataSource"
      class="org.springframework.jdbc
             .datasource
             .DriverManagerDataSource">

  <property name="driverClassName"
            value="org.hsqldb.jdbcDriver"/>
  <property name="url" value="jdbc:hsqldb:mem:db_" />
  <property name="username" value="sa" />
  <property name="password" value="" />

</bean>
```

resources/hibernateContext.xml

```
<bean id="dataSource"
      class="org.springframework.jdbc
             .datasource
             .DriverManagerDataSource">

  <property name="driverClassName"
            value="org.hsqldb.jdbcDriver"/>
  <property name="url" value="jdbc:hsqldb:mem:db_" />
  <property name="username" value="sa" />
  <property name="password" value="" />

</bean>
```

resources/hibernateContext.xml

```
<bean id="dataSource"
      class="org.springframework.jdbc
             .datasource
             .DriverManagerDataSource">

  <property name="driverClassName"
            value="org.hsqldb.jdbcDriver"/>
  <property name="url" value="jdbc:hsqldb:mem:db_" />
  <property name="username" value="sa" />
  <property name="password" value="" />

</bean>
```

¿Qué es un bean?

...

Son objetos que son inicializados, ensamblados y administrados por Spring.

https://docs.spring.io/spring-framework/docs

resources/hibernateContext.xml

```
<bean id="sessionFactory"
    class="org.springframework.orm.hibernate5
           .LocalSessionFactoryBean">

  <property name="dataSource" ref="dataSource" />
  <property name="hibernateProperties">
   <props>
    <prop key="hibernate.validator.apply_to_ddl">false</prop>
    <prop key="hibernate.validator.autoregister_listeners">false</prop>
    <prop key="hibernate.dialect">org.hibernate.dialect.HSQLDialect</prop>
    <prop key="hibernate.show_sql">true</prop>
    <prop key="hibernate.format_sql">true</prop>
    <prop key="hibernate.hbm2ddl.auto">create</prop>
   </props>
  </property>

  <property name="packagesToScan">
    <list>
     <value>ar.edu.unlam.tallerweb1.modelo</value>
    </list>
  </property>
</bean>
```

resources/hibernateContext.xml

```
<bean id="sessionFactory"
    class="org.springframework.orm.hibernate5
          .LocalSessionFactoryBean">

  <property name="dataSource" ref="dataSource" />
  <property name="hibernateProperties">
   <props>
    <prop key="hibernate.validator.apply_to_ddl">false</prop>
    <prop key="hibernate.validator.autoregister_listeners">false</prop>
    <prop key="hibernate.dialect">org.hibernate.dialect.HSQLDialect</prop>
    <prop key="hibernate.show_sql">true</prop>
    <prop key="hibernate.format_sql">true</prop>
    <prop key="hibernate.hbm2ddl.auto">create</prop>
   </props>
  </property>

  <property name="packagesToScan">
    <list>
     <value>ar.edu.unlam.tallerweb1.modelo</value>
    </list>
  </property>
</bean>
```

resources/hibernateContext.xml

```
<bean id="sessionFactory"
    class="org.springframework.orm.hibernate5
          .LocalSessionFactoryBean">

  <property name="dataSource" ref="dataSource" />
  <property name="hibernateProperties">
   <props>
    <prop key="hibernate.validator.apply_to_ddl">false</prop>
    <prop key="hibernate.validator.autoregister_listeners">false</prop>
    <prop key="hibernate.dialect">org.hibernate.dialect.HSQLDialect</prop>
    <prop key="hibernate.show_sql">true</prop>
    <prop key="hibernate.format_sql">true</prop>
    <prop key="hibernate.hbm2ddl.auto">create</prop>
   </props>
  </property>

  <property name="packagesToScan">
    <list>
     <value>ar.edu.unlam.tallerweb1.modelo</value>
    </list>
  </property>
</bean>
```

resources/hibernateContext.xml

```
<bean id="sessionFactory"
    class="org.springframework.orm.hibernate5
          .LocalSessionFactoryBean">

  <property name="dataSource" ref="dataSource" />
  <property name="hibernateProperties">
   <props>
    <prop key="hibernate.validator.apply_to_ddl">false</prop>
    <prop key="hibernate.validator.autoregister_listeners">false</prop>
    <prop key="hibernate.dialect">org.hibernate.dialect.HSQLDialect</prop>
    <prop key="hibernate.show_sql">true</prop>
    <prop key="hibernate.format_sql">true</prop>
    <prop key="hibernate.hbm2ddl.auto">create</prop>
   </props>
  </property>

  <property name="packagesToScan">
    <list>
     <value>ar.edu.unlam.tallerweb1.modelo</value>
    </list>
  </property>
</bean>
```

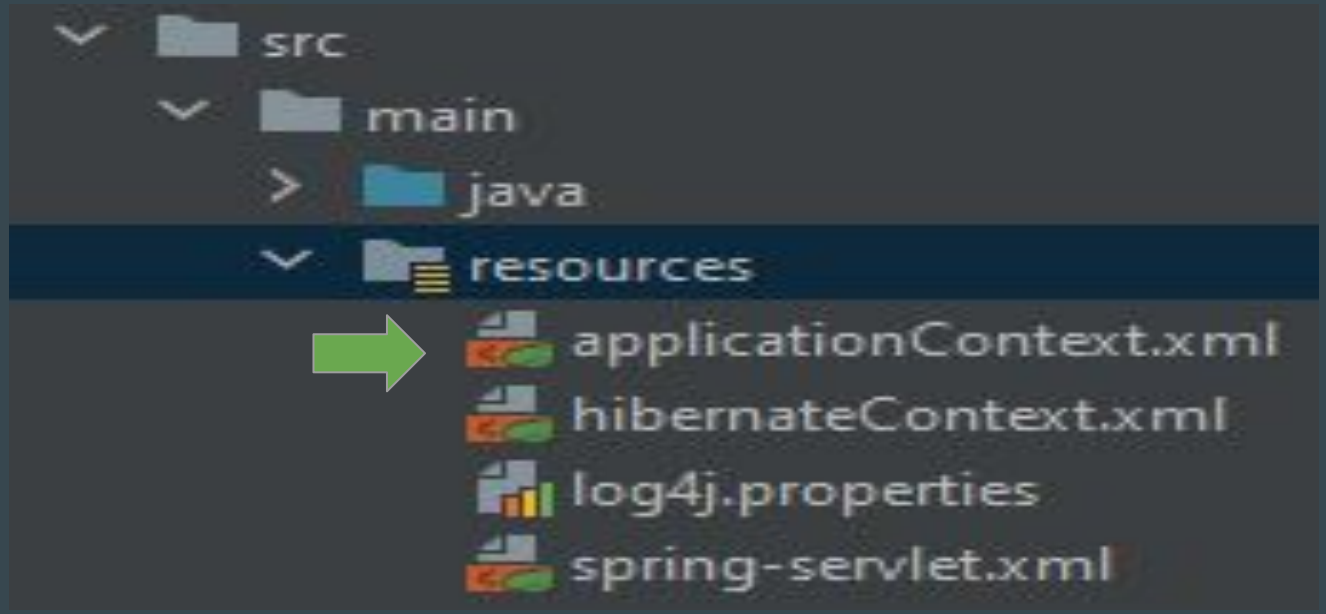
Create
Update
None

resources/hibernateContext.xml

```
<bean id="sessionFactory"
    class="org.springframework.orm.hibernate5
          .LocalSessionFactoryBean">

  <property name="dataSource" ref="dataSource" />
  <property name="hibernateProperties">
   <props>
    <prop key="hibernate.validator.apply_to_ddl">false</prop>
    <prop key="hibernate.validator.autoregister_listeners">false</prop>
    <prop key="hibernate.dialect">org.hibernate.dialect.HSQLDialect</prop>
    <prop key="hibernate.show_sql">true</prop>
    <prop key="hibernate.format_sql">true</prop>
    <prop key="hibernate.hbm2ddl.auto">create</prop>
   </props>
  </property>

  <property name="packagesToScan">
    <list>
     <value>ar.edu.unlam.tallerweb1.modelo</value>
    </list>
  </property>
</bean>
```



resources/applicationContext.xml

```
<import resource="classpath:hibernateContext.xml" />
```

¿Y en el POM?

pom.xml

```
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>5.4.2.Final</version>
</dependency>
<dependency>
  <groupId>org.hsqldb</groupId>
  <artifactId>hsqldb</artifactId>
  <version>2.3.2</version>
</dependency>
```

Ya tenemos hibernate configurado.

...

¿Cómo seguimos?

¿Cómo mapear una clase?

ar.edu.unlam.tallerweb1.modelo

```
@Entity
public class Usuario{

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(name = "CORREO", length = 30, nullable = false)
    private String email;

    // constructor

    // Getters y Setters
}
```

Hibernate User Guide

Servicio

```
@Service
@Transactional
public class ServicioUsuario implements IServicioUsuario{

    private RepositorioUsuario repositorioUsuario;

    @Autowired
    public ServicioUsuario(RepositorioUsuario repoUsuario){
        this.repositorioUsuario = repoUsuario;
    }

    @Override
    public Usuario buscarPorEmail(String email) {
        return repositorioUsuario.buscarPorEmail(email);
    }
}
```

Repositorio (1)

```
@Repository("RepositorioUsuario")
public class RepositorioUsuario implements IRepositorioUsuario{

    private SessionFactory sessionFactory;

    @Autowired
    public RepositorioUsuario(SessionFactory sessionFactory){
        this.sessionFactory = sessionFactory;
    }

    @Override
    public Usuario buscarPorEmail(String email) {
        return ¿?
    }
}
```

Repositorio (2)

```
@Repository("RepositorioUsuario")
public class RepositorioUsuario implements IRepositorioUsuario{

    ……

    @Override
    public Usuario buscarPorEmail(String email) {
        Session session = sessionFactory.getCurrentSession();

        return (Usuario)session.createCriteria(Usuario.class)
                    .add(Restrictions.eq("email", email))
                    .uniqueResult();
                    //list()
    }
}
```

Repositorio (3)

```
@Repository("RepositorioUsuario")
public class RepositorioUsuario implements IRepositorioUsuario{

    ……

    @Override
    public void modificar(Usuario usuario) {
        Session session = sessionFactory.getCurrentSession();

        session.update(usuario);
    }
}
```

Repositorio (4)

```
@Repository("RepositorioUsuario")
public class RepositorioUsuario implements IRepositorioUsuario{

    ……

    @Override
    public void guardar(Usuario usuario) {
        Session session = sessionFactory.getCurrentSession();

        session.save(usuario);
    }
}
```