**Consideraciones:**

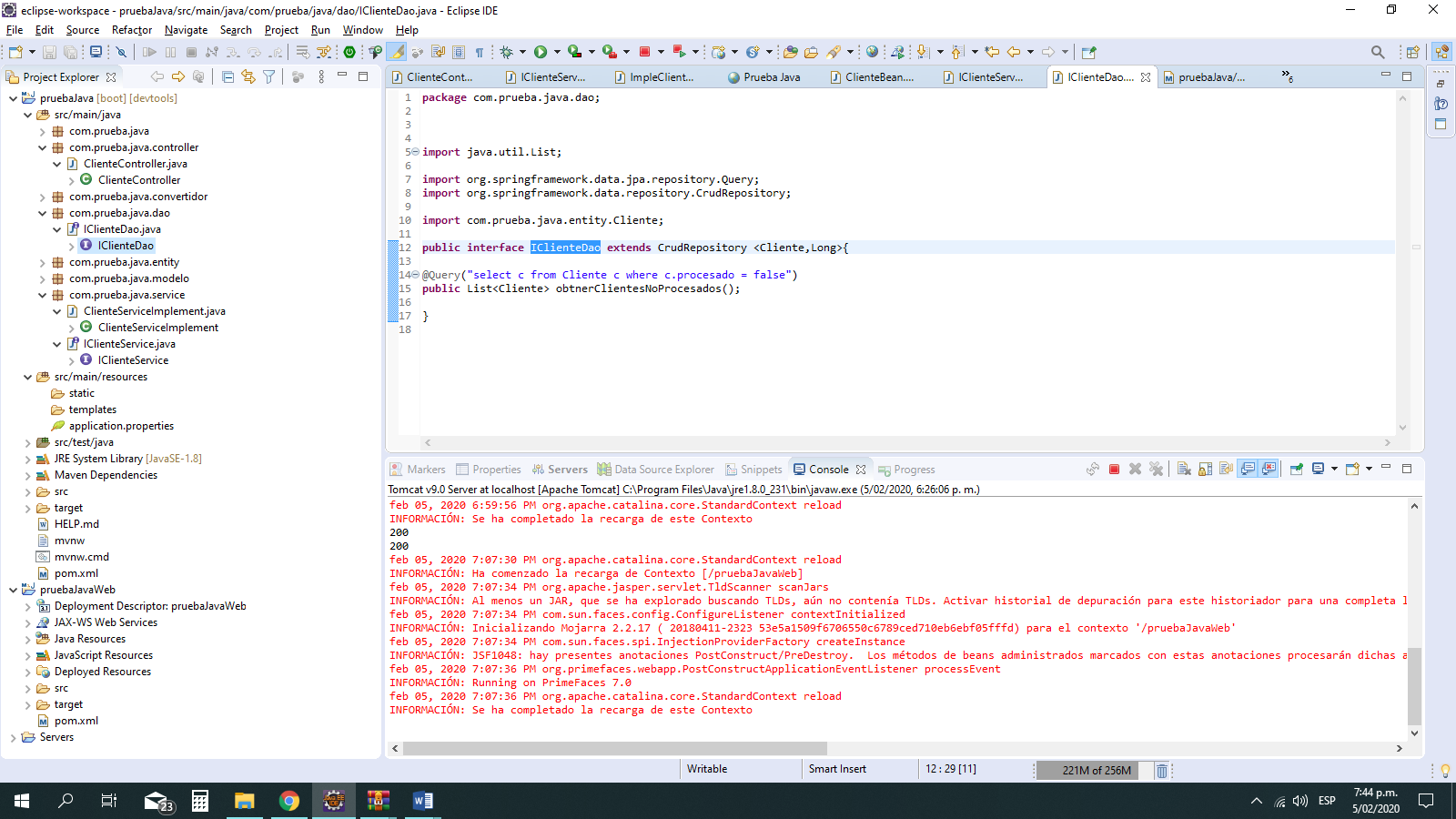
* Utilizar Eclipse o STS como IDE.
* Java 7 u 8.
* Spring 4 o superior.
* Base de datos H2 en memoria.
* Explicar brevemente la arquitectura y los patrones de diseño utilizados. Además de funcionar, se valorarán las técnicas y la calidad del código entregado.
* Entregar adjunta la solución en ZIP o un repositorio GIT privado. No se debe entregar en un repositorio GIT (o similar) que sea público.

1. Todo el software fue hecho en eclipse tanto la parte web como el back. La página web fue hecha en JSF – primeFaces y corre con el servidor apache tomcat
2. Java 8
3. Spring 5
4. Hecho
5. La arquitectura en general fue MVC donde con la vista fue hecha en JFS y, la capa de controlador y modelo soportada por el framework de spring.

En la capa de modelo y controlador podemos apreciar el uso de varios patrones:

Patron Dao:

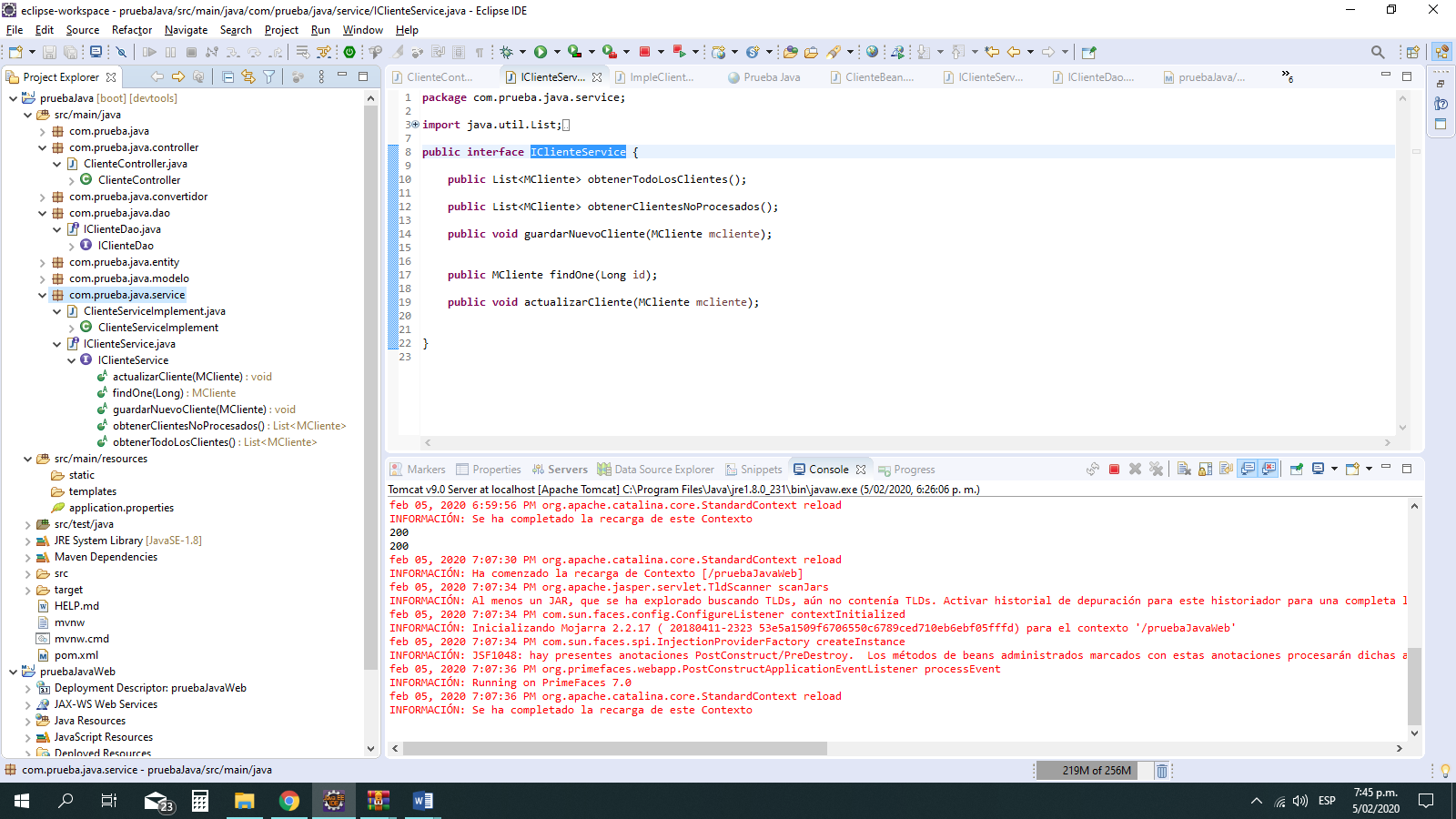
Se encarga del acceso a la base de datos y aislar los procesos del modelo. Spring JPA soporta este patrón atreves de crudRepository.



Patrón Fachade

Lo podemos apreciar en el paquete de servicios, su utilidad principal es evitar que la capa de controlador manipule directamente los Dao además de evitar su uso innecesario. Un servicio en general podría manipular diferentes dao y proporcionar los servicios que necesita los

Controladores



Patron entity and model (no recuerdo bien su nombre):

Este patrón crea un modelo por cada entidad se usa para evitar errores en la manipulación de las entidades por partes del controlador, además nos proporciona mayor versatilidad a la hora de interactuar con los modelos. En teoría el controlador trabaja a nivel modelo mientras que los Dao y repositorio a nivel de entidades y atravez de convertidor pasamos de una a la otra

