**Sdi2-56**

Amable José Valdés Cuervo

CARLOS PON TU NOMBRE Y APELLIDOS

**Pruebas finalizadas**

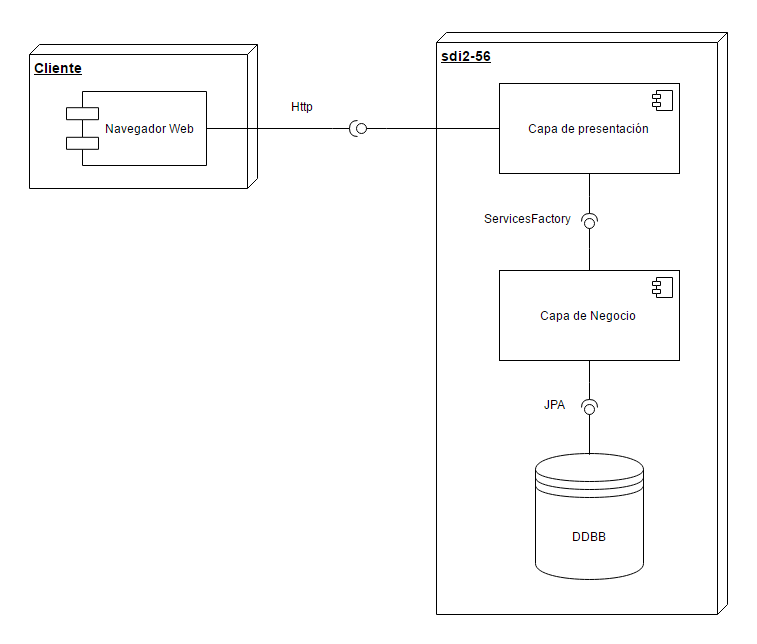
**Mapa de navegación**

**Aspectos Técnicos**

Hemos usado JPA para realizar la práctica. Esto cambia un poco la resolución del problema.

No tenemos capa de persistencia propiamente dicho; nosotros contamos con una serie de clases “Finder” con las que buscamos las clases en la base de datos.

La arquitectura del proyecto sigue siendo un N-Capas, pero en este caso solo tenemos dos capas gracias al uso del JPA.



Pero el uso de JPA tiene un precio; hemos tenido que implementar dos modelos: uno para la capa de negocio (con clases con anotaciones hibernate) y otro para la capa de presentación (DTOs). Gracias a esto el sistema es mucho más independiente entre capas ya que la capa de presentación solo trabajará con información (DTOs carentes de funcionalidad) que le llegan desde la capa de negocio y en la capa de negocio se trabajará con clases persistentes y con funcionalidad.

C:\Users\amabl\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Negocio (2).png

En el anterior diagrama se muestra la capa de negocio del sistema. En el proyecto se usan diversos patrones, los cuales los más importantes son:

* Factory: ServiceFactory nos creará todas las fachadas disponibles para administrar el sistema.
* Facade: UserService, TaskService y CategoryService son fachadas que proporcionan todos los servicios para administrar los usuarios, tareas y categorías respectivamente.
* Command: Hemos creado distintos actions en la capa de negocio. Todas estas clases actions heredan de la clase Command, la cual lo único que tiene es un método llamado “execute()”. La clase CommandExecutor, la cual está presente en todas las fachadas implementadas, ejecuta el método “execute()” de estos actions ejecutando la lógica de negocio del sistema.
* DTO: Para transferir entre capas la información se usan DTOs. La capa de negocio transforma las peticiones en objetos DTO con la información que la capa de presentación necesita y la capa de presentación transforma la información que le llega desde el navegador en DTOs para que la capa de negocio pueda trabajar.

En la capa de presentación se trabaja programáticamente con los Beans como se ha visto en clase y el diseño de las pantallas cuenta con los distintos xhtml. Para facilitar la claridad de código xhtml se han creado distintos templates y/o snippets los cuales son importados a dichos xhtml.

Los xhtml a los que solamente se puede acceder con un rol de usuario en especial se encuentran en las carpetas correspondientes a su rol dentro de la carpeta /restricted. Para esto se han creado una serie de filtros en la capa de presentación.