

30-5-2022

Práctica de la asignatura Tecnología Web 2021-2022



Olga González González

OGONZALEZ95@ALUMNO.UNED.ES

Daniel Bueno Monge

DBUENO41@ALUMNO.UNED.ES

Manuel Jesús Carmona Hidalgo

MCARMONA321@ALUMNO.UNED.ES

Contenido

1. Introducción.....	2
2. Composición del grupo y roles	2
3. Plan de Trabajo	3
4. Especificación de Requisitos.....	5
5. Esquema de Base de Datos.....	6
6. Arquitectura.....	9
Capa Vista	10
Capa modelo	11
Capa Controlador	12

1. Introducción

El objetivo de esta práctica para el curso 2021-2022 es la realización de un portal para la gestión de un estudio de arquitectura.

El sistema dará cabida a diferentes actores del estudio y la información se almacenará en una base de datos para consultar en cada momento el proyecto o certificado.

Para el desarrollo del proyecto se ha seleccionado las siguientes tecnologías:

- *Hibernate* para el mapeo objeto-relacional, que facilita el mapeo de atributos de una base de datos tradicional y el modelo de objetos de una aplicación mediante anotaciones en las entidades que permiten establecer relaciones. Optimiza el flujo de trabajo evitando caer en código repetitivo.
- *Apache Maven* es un software que permite gestionar proyectos. Al crear proyectos Maven, automáticamente se genera una estructura de carpetas predefinidas.
- *JSP* para contenido dinámico en aplicaciones Web con Java, específicamente para el lado de la vista. Aunque se muestra en el lado del cliente este se ejecuta en el lado del servidor.
- *Bootstrap* para diseños basados en CSS y HTML. Se utiliza en el front-end de las aplicaciones.
- *JDBC* para conectividad de base de datos y proporcionar una conexión a la base de datos.
- *Github* para alojar el proyecto y utilizar el sistema de control de versiones entre los participantes del proyecto.

Se decide por estas tecnologías para cumplir con los requerimientos técnicos descritos por el Equipo Docente y realizar un desarrollo optimizado, fácil y ajustado a las tecnologías actuales.

2. Composición del grupo y roles

Debido al uso diferentes tecnologías y al tiempo a emplear en el análisis e implementación de la práctica, se ha decidido llevar a cabo entre tres alumnos de la asignatura, simulando al máximo el trabajo en una empresa y acordando fechas de entrega para entregar el proyecto en la fecha establecida.

El desarrollo del proyecto se ha realizado teniendo en cuenta los diferentes roles (Coordinador de proyecto, Analista y Programador. Aunque todos los compañeros hemos cumplido las funciones de los diferentes roles, siempre hubo una función más específica para cada uno de nosotros.

- Manuel Jesús la función de Coordinador de proyecto.
- Olga González la función de Programadora.
- Daniel la función de Analista.

3. Plan de Trabajo

Para el desarrollo del proyecto se ha empleado la metodología desarrollo iterativo e incremental, donde se planifica diversos bloques temporales llamados iteraciones.

Estas iteraciones se entienden como mini proyectos, así en todas las iteraciones se repite un proceso de trabajo similar para proporcionar un resultado completo sobre el producto final, de manera que se puede ver los avances del proyecto de forma incremental.

Las fases del proyecto se distinguen en áreas bien definidas, aunque como se ha explicado anteriormente, en cada fase están incluidas las demás fases para un desarrollo del proyecto incremental.

- *Fase I: Investigar las diferentes tecnologías.*

La primera semana los miembros del equipo nos centramos en investigar las diferentes soluciones tecnológicas para seleccionar la más adecuada de acuerdo con los requisitos de la práctica.

La segunda semana con las tecnologías seleccionadas se realizó un mini proyecto para comprobar que las opciones eran las adecuadas y la persistencia con la base de datos era correcta.

Por último, se comprobó en la máquina proporcionada por el Equipo Docente el correcto funcionamiento de las tecnologías.

- *Fase II: Diseño de la arquitectura*

En la tercera semana, analizando los requerimientos de la práctica, se comienza en la creación del diagrama de casos de uso para obtener una primera visión en cuanto a roles y funcionalidades necesarias. En función de los roles y funciones acordadas por participantes del proyecto se procede a diseñar el diagrama de las bases de datos y un diagrama de clases.

En la cuarta semana, para finalizar el proceso de diseño, se procesa a diseñar en estático una primera visión de menús de la aplicación y los formularios.

- *Fase III: Implementación*

Se comienza con el desarrollo de la aplicación. Al participar tres personas se procede a repartir el diseño de Modelo Vista Controlador en una parte para cada uno. Se crea un repositorio en la plataforma GitHub para realizar un control de versionado y el programa de escritorio *SourceTree* para simplificar la forma de interactuar con el repositorio Git. El desarrollo de esta fase el tiempo estimado es de tres semanas.

- *Fase IV:*

Cada funcionalidad implementada se procedía a probar. Primero el desarrollo se llevaba a cabo en el entorno local y cuando ya hubo varias funcionalidades implementadas, se procedía a probar en la máquina proporcionada por el equipo docente.

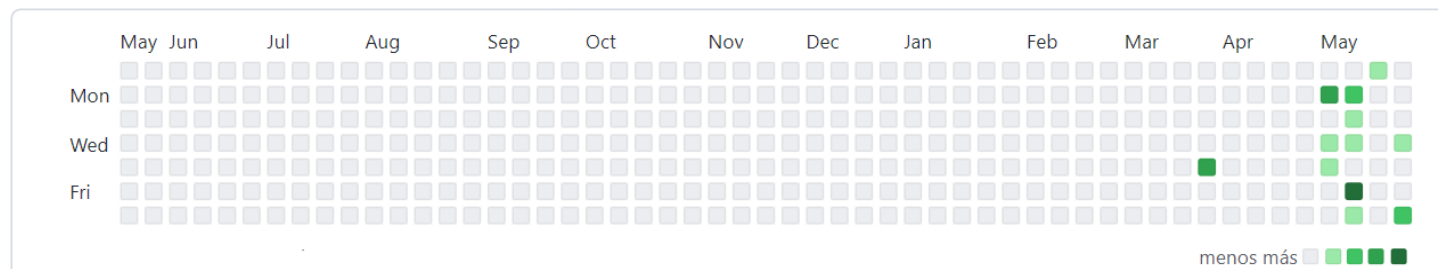
Si alguna fase no daba el resultado esperado se volvía al punto anterior para dar solución a lo propuesto con los resultados obtenidos.

Para agilizar el desarrollo del proyecto se decidió por trabajar con una base de datos en local para posteriormente probar la aplicación final con base de datos en memoria.

Para una correcta comunicación entre los miembros del proyecto se ha procedido a establecer conexión con la plataforma Zoom una vez por semana y creación de grupo a través de nube para intercambio de información y opiniones.

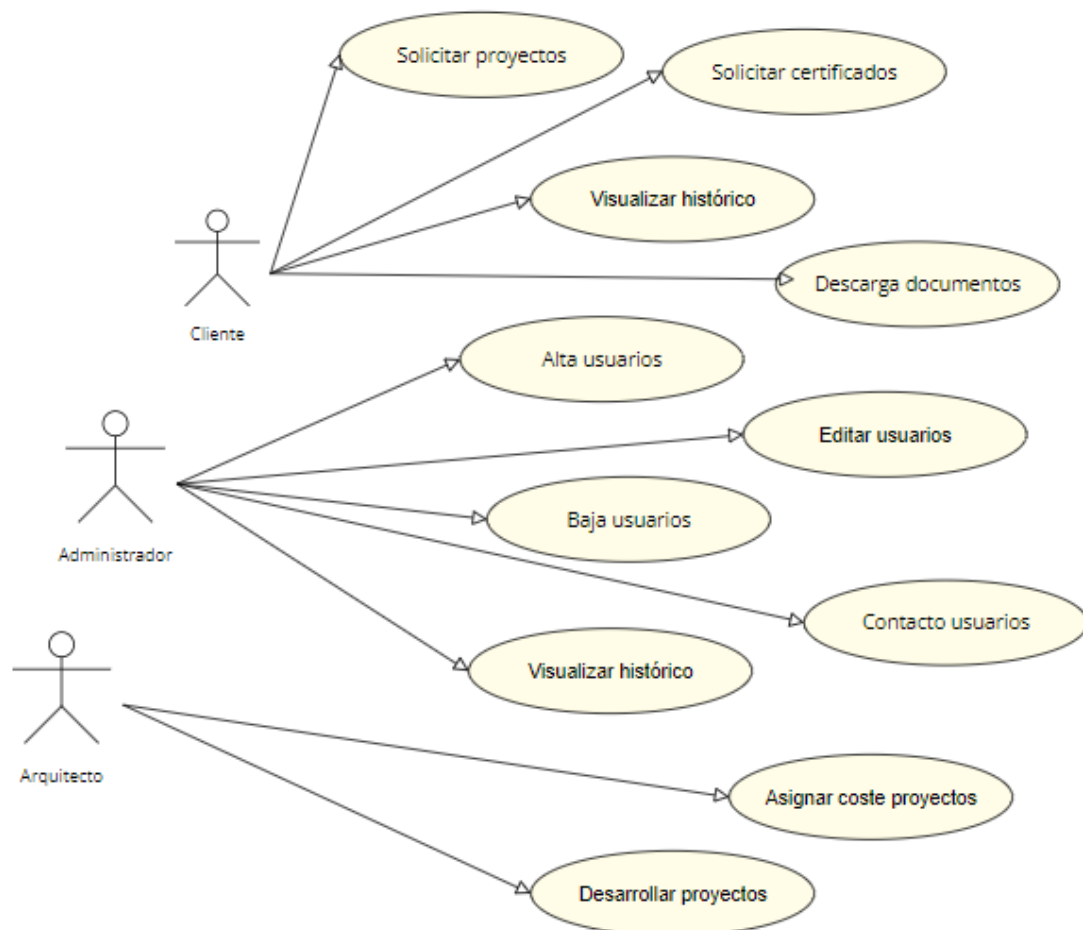
La imagen muestra un historial de los cambios realizados en el repositorio GitHub por parte de los miembros del proyecto.

21 aportes en el último año



4. Especificación de Requisitos

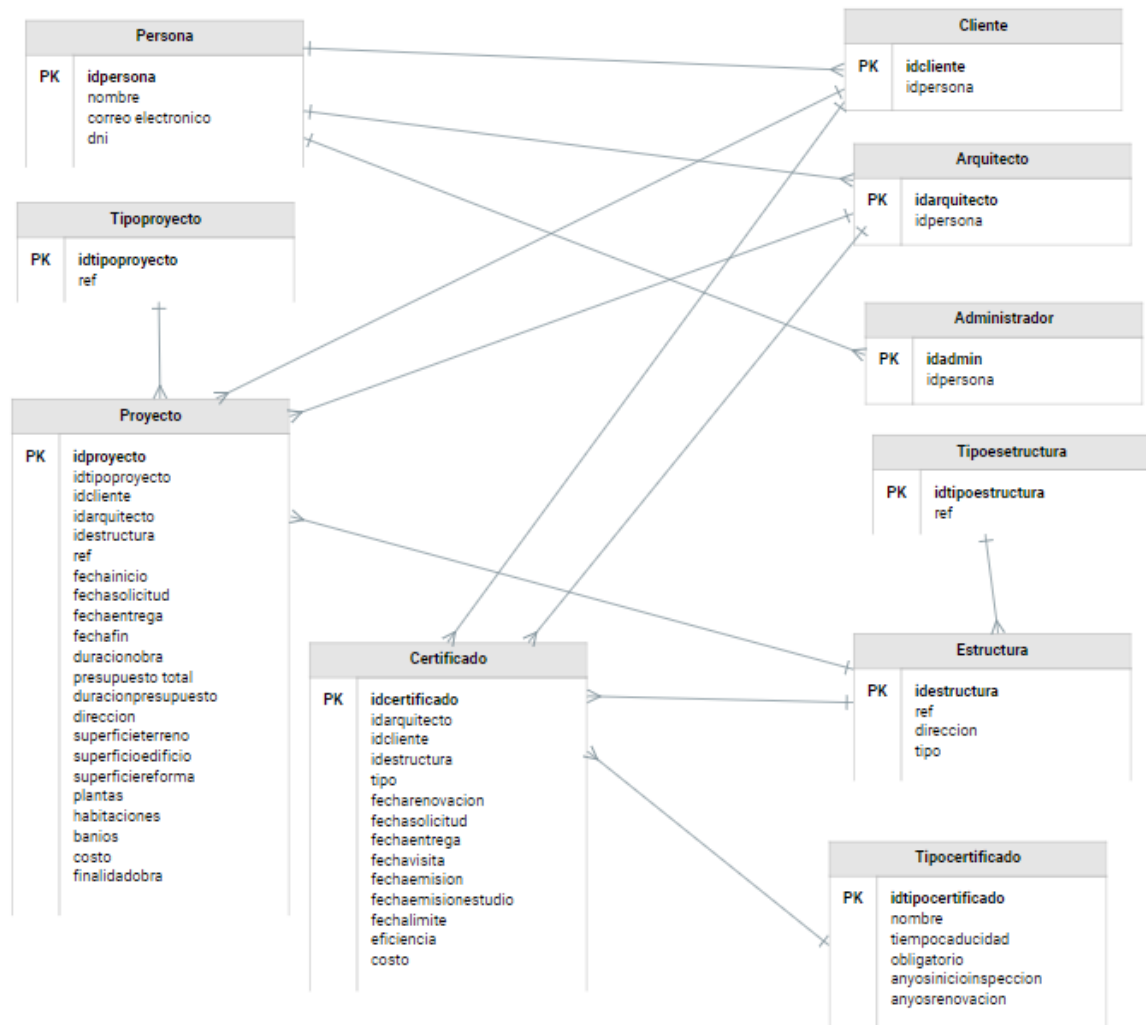
Los requisitos han sido marcados por el equipo docente, se diseña el diagrama de casos de uso, donde se visualizan los roles y funciones.



Se muestran los siguientes actores:

ACTOR	DEFINICIÓN
Cliente	Los clientes solicitan proyectos y certificados mediante el formulario. Pueden visualizar el histórico y descarga de documentos.
Administrador	Gestionar los usuarios en el sistema. Obtener listado de los clientes.
Arquitecto	Desarrollo de proyecto y certificados y asignar coste a los proyectos.

5. Esquema de Base de Datos

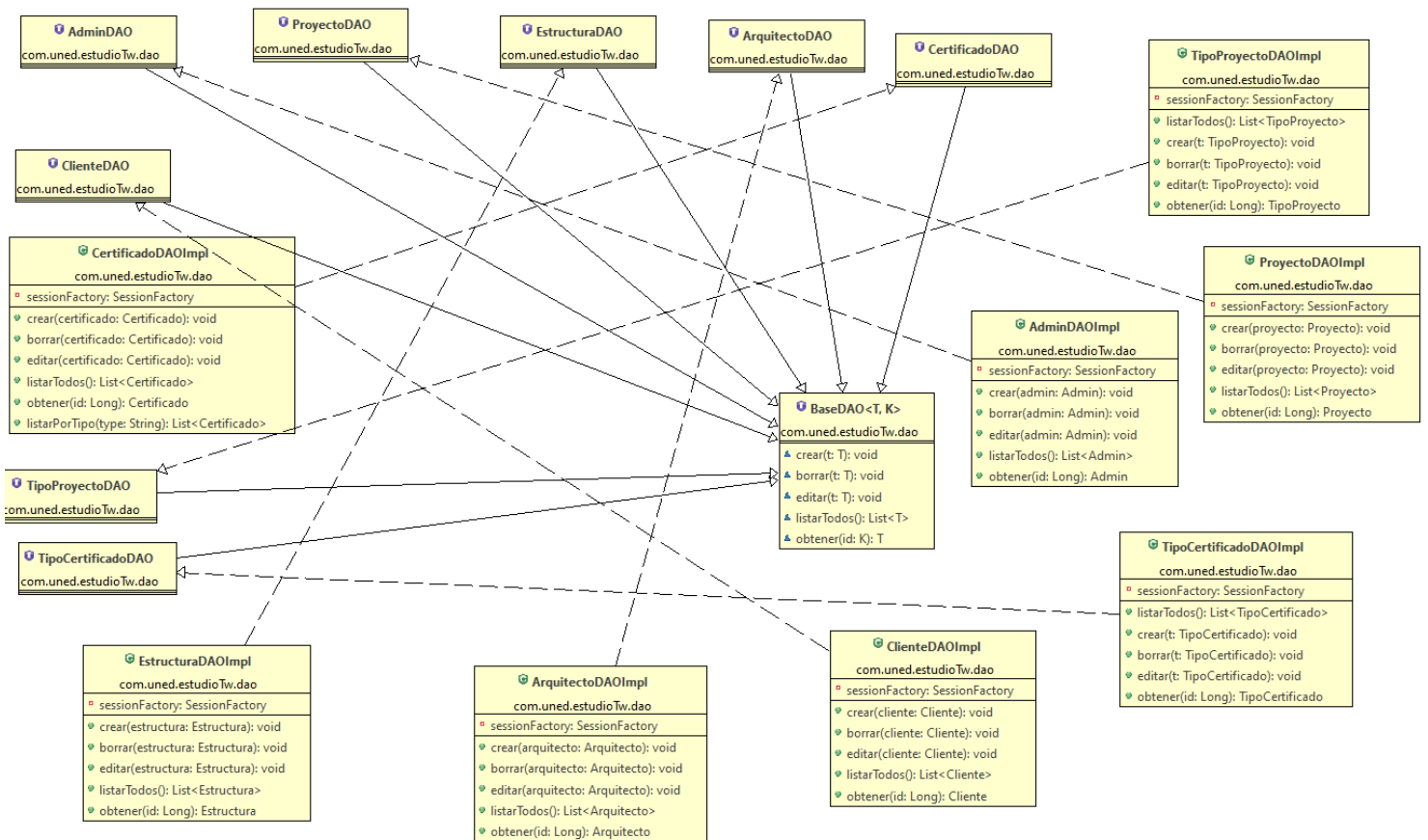


El diseño del esquema de la base de datos se ha basado en un diagrama de clases donde PK especifica la clave privada.

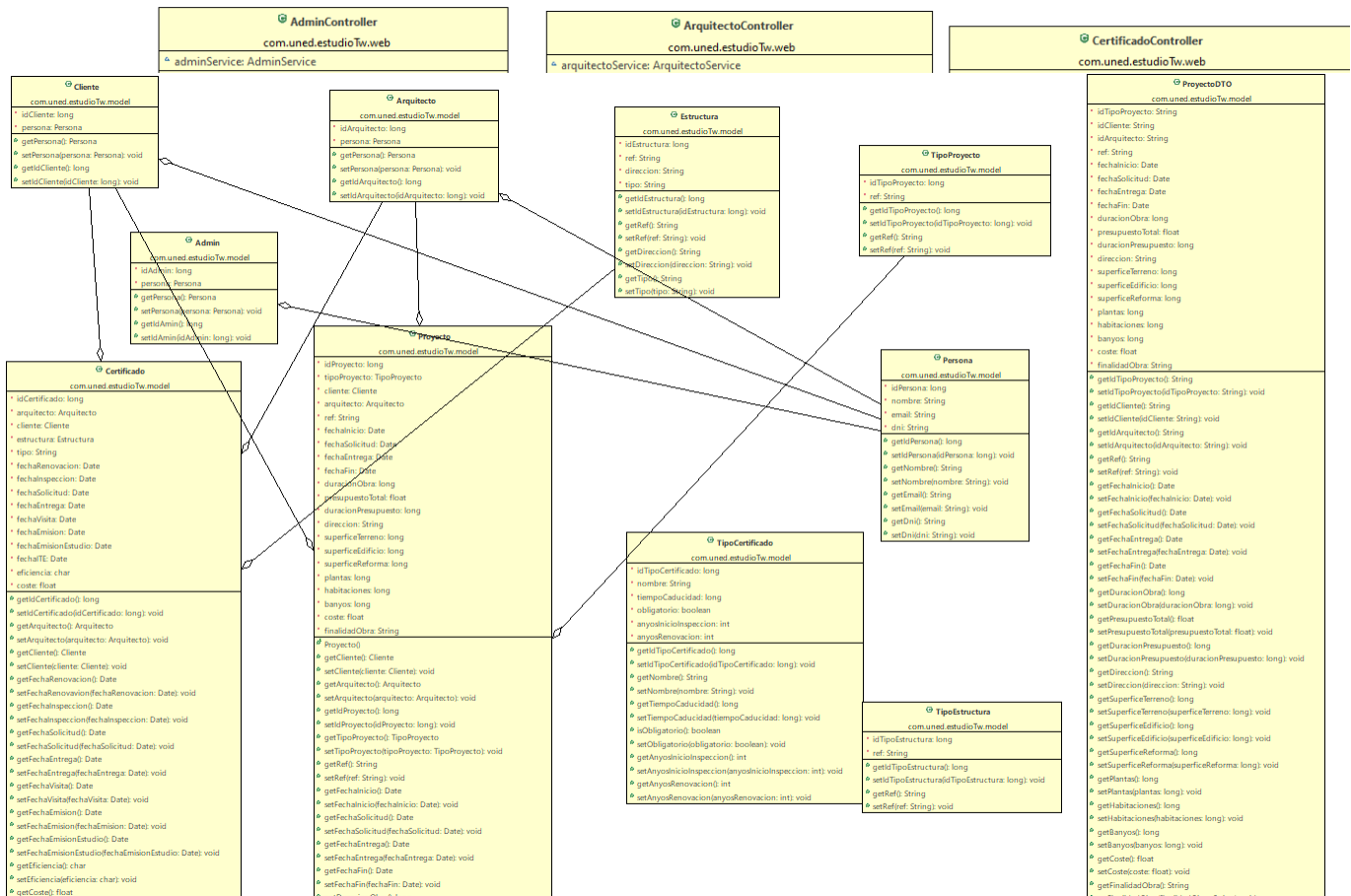
Se ha creado una tabla Persona donde hacen referencia la tabla Cliente, Administrador y Arquitecto.

La tabla Proyecto y certificado hacen referencia a Cliente y Arquitecto, donde a su vez Proyecto se referencia con Tipoproyecto. La tabla certificada se referencia con Estructura y el tipocertificado.

A continuación, se muestra el diagrama de clases del Data Acceso Object. Se muestra los métodos necesarios para insertar, actualizar, borrar y consultar información.

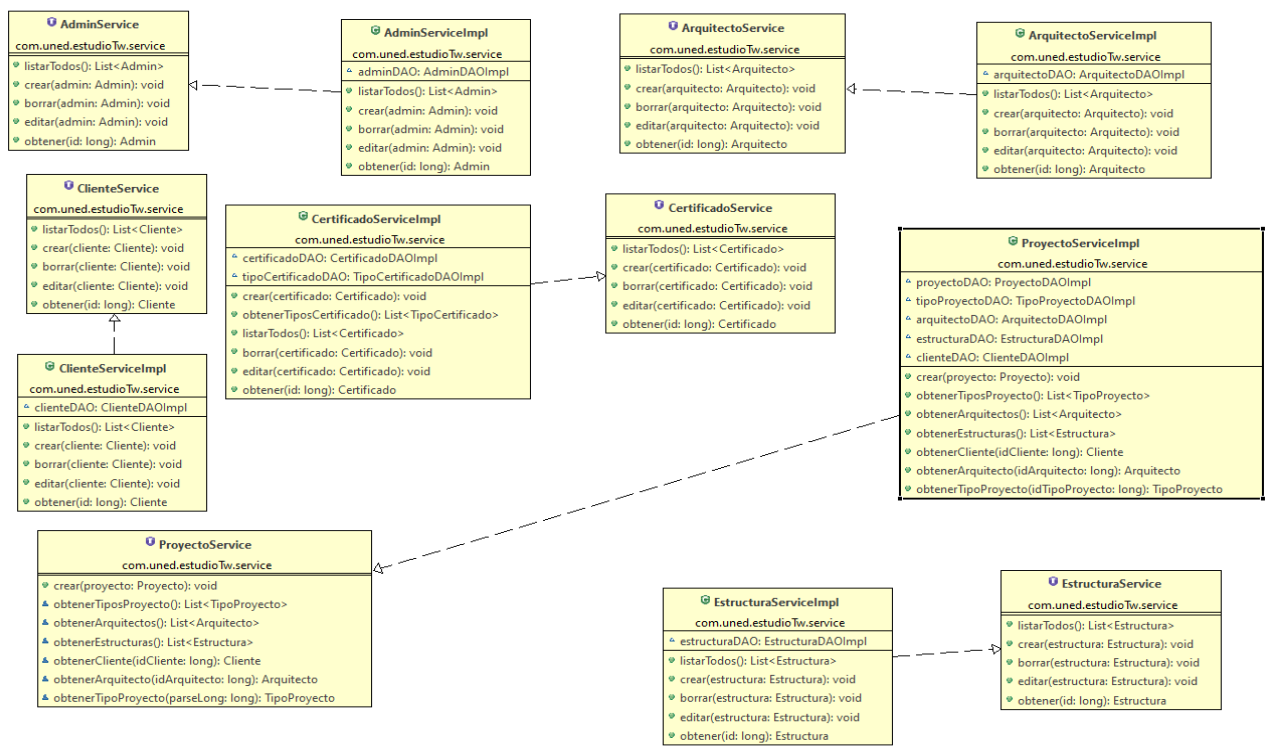


Se muestra el diagrama de clases del controlador:



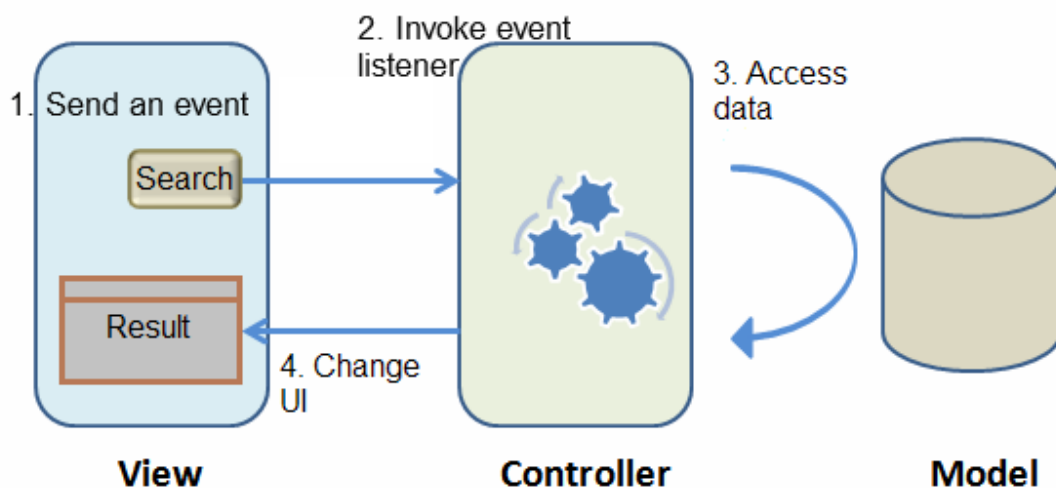
Se muestra el diagrama de clases del Modelo:

Se muestra el diagrama de clases de Service:



6. Arquitectura

El patrón utilizado es el diseño Modelo-Vista-Controlador. Se intenta desacoplar la vista del modelo, con la finalidad de mejorar la reusabilidad, mantenimiento y escalabilidad del código. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio.

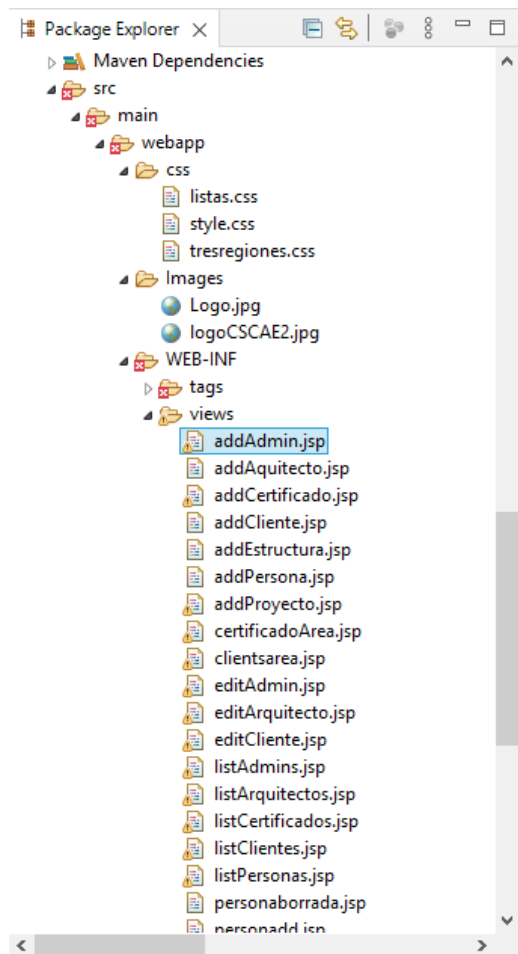


También se combina con el patrón DAO que define la relación entre la lógica de presentación y la de negocio por una parte y por otra la relación con la capa de datos. El DAO tiene una interfaz común, sea cual sea el modo y fuente de acceso a datos.

Capa Vista

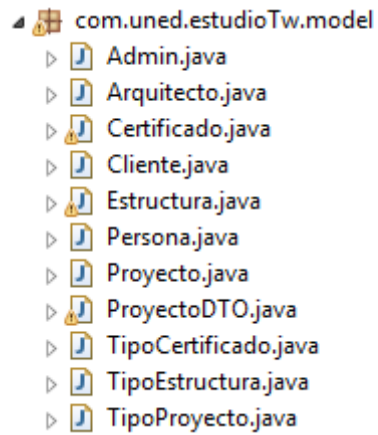
Si comenzamos con la capa de Vista, dentro de la carpeta *src/main/webapp/web-inf/views* se incluye los ficheros .jsp con las páginas web dinámicas.

También se incluyen los ficheros .css con las hojas de estilos y los logos de la aplicación.



Capa modelo

En la capa modelo se definen las clases y se anotan, para que persistan a base de datos.



Capa Controlador

Sirve de enlace entre las vistas y los modelos.

