



ANTEPROYECTO

TFG Searcher

Patrones Software

Guillermo Gómez Olivares DNI: 03149237-P

José Ramón Casas Álvarez DNI: 03141703-H



1. Objetivos del proyecto:

La aplicación “TFG Searcher” ha sido desarrollada inicialmente en papel, en las cuales planteamos las clases junto con su contenido y posteriormente hemos desarrollado el diseño de dicha aplicación.

La aplicación dispondrá de una interfaz gráfica de inicio de sesión.



A screenshot of a login window titled "¡Bienvenido!". It features a light gray background and a standard macOS-style title bar with red, yellow, and green window control buttons. The window contains two input fields: "Usuario:" with an empty text box, and "Contraseña:" with a masked password field showing ten asterisks. Below these fields is a "Conectar" button.

Una vez iniciada la sesión, el sistema comprueba si el usuario se trata de un alumno o un administrador de TFG. En caso de ser un usuario **alumno**, se dirige a la ventana de alumno.



A screenshot of a student dashboard window. It has a light gray background and a macOS-style title bar. At the top, it shows "Usuario:" followed by a text box containing "@user". Below this, there are three buttons: "Ver Lista TFG" on the left, "Mis elecciones" on the right, and "Tus TFG Asignados" centered at the bottom.

En caso de ser un usuario **administrador**, el sistema se dirige a la siguiente ventana.



Para realizar todo lo descrito anteriormente hemos aplicado los patrones de diseño expuestos en la asignatura (patrones de creación, estructurales y de comportamiento).

2. Diseño e implementación:

En NetBeans hemos creado todas las clases y JFrame que hemos necesitado. En cuanto a clases JFrame, que son las clases de las interfaces.

En cuanto a los JFrame tenemos las 3 siguientes:

- **InicioSesion:** Es la clase JFrame encargada de validar a un usuario y determinar si dicho usuario es un alumno o un administrador.
- **InterfazAdmin:** Es la clase JFrame que permite al usuario administrador poder ver una lista con todos los TFG ofertados, ver una lista con los alumnos y sus TFG prioritarios y poder asignar los TFG a los alumnos según sus solicitudes.
- **InterfazUser:** Es la clase JFrame que permite al usuario alumno poder ver una lista detallada de todos los TFG que la universidad oferta, elegir 3 TFG que dicho alumno elija y poder ver sus elecciones y ver los TFG asignados una vez se haya realizado dicha asignación por un administrador.

En cuanto a las clases .Java:

- **GestorBD:** Es una clase que se comparte a todas las demás clases, se encarga de realizar la gestión con la base de datos (inserciones, modificaciones, eliminaciones y visualización de datos).
- **Alumno:** Objeto que tiene todos los datos de un alumno.
- **Profesor:** Objeto que tiene todos los datos de un profesor.
- **TFG:** Objeto que tiene todos los datos de un TFG.

En cuanto a los patrones que vamos a utilizar en nuestro proyecto, son los siguientes:

- Patrones de Creación:
 - **Patrón Singleton:** Singleton garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella. Esta instancia es una operación estática y extensible por herencia a la cual, los clientes pueden usar una instancia extendida sin modificar su código.
- Patrones de comportamiento:
 - **Observer:** Observer permite definir dependencias uno-a-muchos de forma que los cambios que se produzcan en un objeto se comuniquen a los demás que dependan de él.
 - **Iterator:** El iterador permite acceder de una forma coherente a los datos de una colección, independientemente del tipo de esa colección.
- Patrones de diseño:
 - **Object Pool:** Utilizaremos un patrón de pool de conexiones para las conexiones cliente con la base de datos, ya que este patrón de diseño suele mejorar significativamente el rendimiento en situaciones donde el coste de iniciar una instancia de clase es alto y cuando el número de instancias simultáneas no es muy alto.

Conforme se vaya desarrollando la práctica, iremos aplicando nuevos patrones, pero, inicialmente, estos son los que tenemos pensados aplicar.

3. Análisis de requisitos:

REQUISITOS FUNCIONALES:

- 1- Se dispondrá de una base de datos basada en archivos.
- 2- Se accede a la aplicación mediante el usuario y la contraseña establecidos por la uah a cada usuario.
- 3- Los administradores dispondrán de un usuario y una contraseña para manejar todo el sistema y poder realizar la asignación de los TFG.
- 4- La aplicación será programada en lenguaje Java.
- 5- En la ventana principal de usuarios hay 3 botones: Ver Lista TFG, Mis Elecciones y Tus TFG Asignados.
- 6- El botón “Ver Lista TFG” debe mostrar al usuario un listado con los TFG disponibles, ordenado estrictamente por código de grado.
- 7- El botón “Mis Elecciones” debe mostrar al usuario un formulario a rellenar donde seleccionar sus TFG elegidos (debe elegir 3).
- 8- El botón “Tus TFG Asignados” deberá mostrar el mensaje “Aún no se te ha asignado ningún TFG” en caso de que no se haya realizado la asignación o deberá mostrar el TFG asignado por los profesores si se ha realizado la asignación.
- 9- En la ventana principal de administradores hay 3 botones: Lista TFG, Lista Alumnos Prioridad TFG y Asignar.
- 10- El botón “Lista TFG” debe mostrar el listado de los TFG donde dicho administrador puede añadir, borrar o modificar TFGs.
- 11- El botón “Lista Alumnos Prioridad TFG” debe mostrar una lista con los alumnos que han seleccionado los TFG que desean realizar por orden de prioridad.

12- El botón “Asignar” realiza, mediante un algoritmo, la asignación de dichos TFG con los respectivos alumnos.

13- En cada ventana habrá un botón para retroceder a la ventana principal.

14- En la ventana “Ver Lista TFG” se mostrará una lista con el ID del TFG, el nombre del TFG seguido de la carrera a la que pertenece, la descripción de dicho TFG y el profesor que tutoriza el TFG.

15- En la ventana “Ver Lista TFG” se podrá seleccionar un TFG para ver más detalladamente en qué consiste.

16- En la ventana “Lista Alumnos Prioridad TFG” la información mostrada se organizará de la siguiente manera: Nombre, Apellidos, DNI, Carrera, ID TFG1, ID TFG2, ID TFG3.

17- Se dispondrá de una ventana de profesores con un listado de los TFG que oferta y si tiene alguno asignado.

18- En la ventana “Lista Alumnos Prioridad TFG” los administradores no podrán añadir, eliminar ni modificar ningún alumno debido a que los datos son recibidos de una base de datos.

19- Los profesores contarán con un máximo de 5 TFG.

20- En la ventana “Lista TFG” de los administradores, se dispondrán de 3 botones para añadir, modificar o eliminar un TFG, en caso de añadir un nuevo TFG, se dispondrá de una nueva interfaz en la cual hay un formulario que los administradores deben rellenar.

21- El botón “Asignar” de la interfaz de administrador mostrará una nueva lista con los alumnos asignados a uno de los TFG que han elegido por prioridad.

22- En la ventana nuevo candidato, tendremos lo necesario para meter los datos de los nuevos candidatos, tales como: DNI, Nombre, Teléfono, Dirección, Localidad, Cod Postal, carrera, TFG 1, TFG 2, TFG 3.

23- En la ventana nuevo candidato, dispondrá de un botón, para agregar al candidato a la base de datos, en el caso de que se hubieran introducido bien los datos.

24- En la ventana nuevo candidato, dispondrá de un botón para volver atrás, a la ventana principal.

25- En la ventana de “Asignar”, una vez buscado el TFG para el alumno o el alumno ha buscado el TFG y se le ha asignado, se borrarán de la base de datos, el alumno y el TFG.

26- En la ventana principal, existirá un botón para cerrar nuestra sesión.

27- Conocer las dependencias entre los elementos software:

- Un programa puede llamar a otros programas y también ser llamado por otros;
- Un programa puede usar ninguno o varios datos;
- Un dato puede ser usado por uno o varios programas;
- Un documento puede referirse a un programa, a un dato o a una aplicación;
- Los datos, programas y aplicaciones pueden tener varios documentos asociados.

28- Esta información se dirigirá a las personas responsables de la misma:

- Departamento de administración de la universidad.
- Los responsables de las peticiones solo los datos de sus peticiones.
- Cualquier persona del departamento técnico solamente los datos de tareas en las que está involucrado.
- Los profesores que serán tutores de TFG podrán consultar la lista de Alumnos.

REQUISITOS NO FUNCIONALES:

1. Rendimiento

- a. Requisitos mínimos: Pentium II, 512MB de RAM, 500MB de almacenamiento disponible.

2. Disponibilidad

- a. Se podrán consultar tanto alumnos como los TFG disponibles offline.
- b. Si se desean nuevos alumnos, deberá conectarse a internet.

3. Accesibilidad

- a. Aplicación con botones grandes.

4. Usabilidad

- a. Está destinado solo a alumnos de último curso que tengan que buscar TFG.
- b. Interfaz gráfica agradable e intuitiva.

5. Gestión de errores

- a. Utilización de ficheros de log para registrar cualquier incidencia en el Sistema.

6. Portabilidad

- a. La aplicación tendrá soporte físico y online para su instalación.

7. Escalabilidad

- a. La escalabilidad está íntimamente ligada al diseño ya que influye satisfactoriamente al rendimiento del programa.

8. Mantenibilidad

- a. Todo el sistema deberá estar debidamente documentado, tanto en el código fuente como en los manuales de administración y de usuario.

9. Fiabilidad

- a. La aplicación es totalmente fiable en base a pruebas de errores y su mantenimiento en el sistema.

10. Seguridad

- a. El sistema debe cumplir las disposiciones recogidas en la Ley Orgánica de Datos Personales y en el Reglamento de medidas de seguridad.
- b. Realización de copias de seguridad
- c. Comprobar usuario/contraseña

11. Restricciones Hardware y Software

- a. Ordenador con cualquier SO (Windows XP y superiores, Linux, OSX)
- b. Compatibilidad total con ordenador que posean cualquiera de los siguientes navegadores web: Mozilla Firefox, Chrome, Ópera, Safari e Internet Explorer.