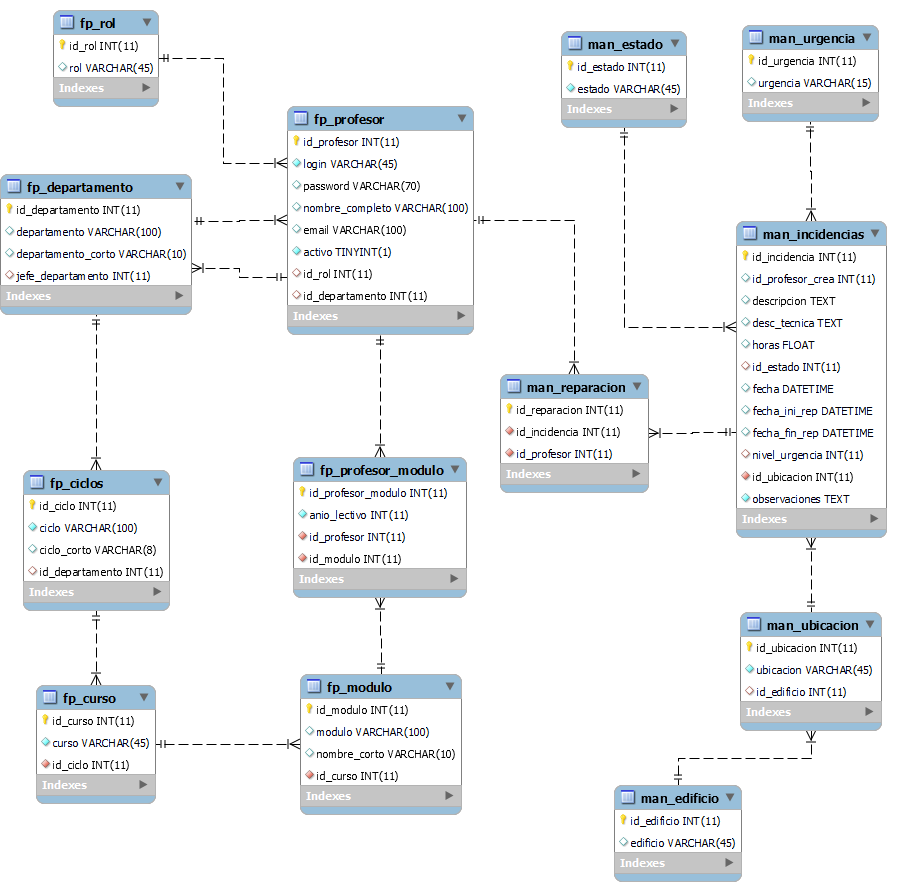
|  |
| --- |
| **Desarrollo de interfaces.**  **Proyecto final.**  **Aplicación mantenimiento.** |

Fecha de trabajo en clase: jueves 24 de febrero

Fecha de entrega: viernes 4 de marzo

Hacer una aplicación para gestionar el profesorado y el mantenimiento de un instituto. Sobre la base de datos mantenimiento\_di (usuario pas Alumno, passwd Admin1234), o desde la vuestra individual. El diagrama entidad-relación, es el siguiente:



Existen 2 tipos de tipos de tablas, las tablas de organización del centro, llamadas fp y las tablas asociadas al mantenimiento llamadas man.

Existen diferentes 3 roles y las funciones de cada uno es la siguiente:

* Miembro del profesorado.
  + Puede enviar correos a otro miembros o grupos del profesorado.
  + Puede ver al resto de profesores, para enviarles un correo.
  + Puede crear incidencias, y modificar sus propias incidencias.
* Personal técnico.
  + Las mismas tareas que el resto de miembros del profesorado.
  + Buscar y leer todas las incidencias, y actualizarlas, tanto para documentar las gestiones realizadas, como cambiarles el estado y la prioridad.
* Usuarios administradores.
  + Las mismas tareas que cualquier miembro del profesorado o personal técnico.
  + Realizar CRUD sobre las tablas (por acortar, solo sobre la tabla profesores). Resumiendo, gestionar profesores. Gestionar profesores consistirá en:
    - Alta/baja de profesores. Evitamos borrar profesores, por cuestiones de solidez de la BBDD.
    - Cambio de roles.
    - Resetear password de usuarios.
  + Obtener estadísticas, como pueden ser:
    - Incidencias cerradas en un periodo de tiempo.
    - Agrupar las incidencias por meses.
    - Resumen incidencias cerradas por cada técnico.

Lo primero que hará un usuario, será logarse, y cargar las opciones disponibles para su rol.

Una cuenta que se puede emplear para enviar correos es:

odoochomon@gmail.com

Admin\*1234

Datos del servidor:

Servidor SMTP smtp.gmail.com

Puerto SMTP 587

Authenticate with Nombre de usuario

Seguridad de la conexión TLS (STARTTLS)

**Criterios de calificación:**

**Generales**

1. (0,5) El código está organizado correctamente en carpetas y comentado. Las librerías utilizadas están en el pom.
2. (0,5) Robustez y usabilidad de la aplicación. Utilizar una única conexión a la BBDD y esta se cierra adecuadamente. Si alguna excepción cierra la aplicación, es idóneo para cerrar la conexión.
3. (0,5) Se le informa al usuario del error, con mensajes claros. Las operaciones de desactivar un profesor, crear incidencia y cerrar incidencia, necesitan mensaje de confirmación.
4. (0,7) La aplicación es usable y no daña a los ojos desde una resolución de 800x600 en adelante. Os puede servir Toolkit. getDefaultToolkit(). getScreenSize().

**Logueo**

1. (0,5) El password se guarda encriptado en la BBDD.
2. (0,5) Los menús mostrados dependen del rol del usuario logado.

**Funcionamiento**

1. (0,5) Mostrar, ordenar y filtrar las tablas de profesores y de incidencias.
   1. Profesor filtrar rol, departamento, activo
2. (0,5) Los usuarios autorizados, pueden cambiar el estado y la urgencia de una incidencia.

**Instalador de aplicaciones.**

1. (0,3) Se realiza un instalador para la aplicación.
2. (0,5) En la instalación se solicita la url, el nombre de la BBDD y el puerto a la base de datos. Esa información, se almacenará en un archivos .json, .properties o .ini en la carpeta donde este el ejecutable. (se puede utilizar nsis, inno o launch4j).
3. (0,3) El programa empaquetado (.jar) es funcional en Windows y en Ubuntu.

**Tratamiento de fechas.**

1. (0,5) Cada usuario podrá configurar el formato de las fechas, entre formato largo (2/2/2022) y corto (2/2/22).
2. (0,5) Para la introducción de fechas se utilizarán componente de la librería JCalendar, o cualquier otra librería.

**Correo.**

1. (0,5) Cada usuario puede configurar el correo saliente desde el que envía sus correos, si desea almacenar la contraseña o introducirla al enviar el correo.
2. (0,5) Se envían correos individuales.
3. (0,5) Se envían correos a grupo (por ejemplo, a los miembros de un departamento).

Posible ayuda

<https://www.youtube.com/watch?v=K-cJQlMUbgY>

**Documentación de la aplicación.**

1. (0,5) Se crea una documentación electrónica de la aplicación, con al menos 3 pantallas. Además, dependiendo desde donde se solicite la ayuda, se muestre la pantalla asociada (javahelp).

**Importar y exportar**

1. (0,7) La aplicación permite exportar e importar datos de la tabla profesor desde y hacia un Excel o fichero csv.

**Informes y gráficos.**

1. (0,5) Desde un usuario con rol de administrador, se mostrará por pantalla una estadística de las incidencias agrupadas por meses, con un gráfico de barras. Se recomienda la librería jfreechart.
2. (0,5) Desde un usuario con rol de administrador, se imprimirá un listado agrupado de las incidencias, con un gráfico de barras.

ANEXO

* Se penalizará la copia evidente de código.
* Se evaluará el trabajo continuo del alumno.
* Si se desea cambiar los tipos de gráficos, se puede previo permiso del profesor.