# Consideraciones técnicas.

1. Considerando que se podían producir dos tipos de errores, lógica de la aplicación o validación de campos se han hecho dos clases para diferenciar los errores.
2. Considerando que al borrar un usuario se tienen de borrar todos su expedientes y que la bbdd no estaba diseñada on delete cascade se ha puesto en el archivo Usuarios.xml en el tag ”Set” de expedientes que sea on delete cascade con el atributo “cascade=’delete’”.
3. Para poder mostrar el Usuario que ha creado el expediente se necesita o dejar la conexión abierta para poder acceder o descargar toda la información junta se ha optado para descargar toda la información esto se consigue en Expedientes.xml en el tag many-to-one que almacena el usuario que ha dado de alta el expediente se le pone el atributo “lazy=’false’” de esta forma conseguimos traernos toda la información con una única consulta.

# Consideraciones funcionales.

1. Para controlar la largada máxima de los campos de usuario o expediente en los propios setters de la clase se han puesto controles con los cuales si entra un dato con un formato incorrecto se lanza una excepción y así se puede avisar al usuario en vez de intentar guardarlo en la bbdd y que genere un error bloqueando la aplicación.
2. Expedientes
   1. Nombre
      1. Una largada máxima de 30
   2. Apellidos
      1. Una largada máxima de 30
   3. DNI
      1. Con una expresión regular se ha comprobado que sea un formato de 8 dígitos y una letra al final
   4. Cp
      1. Una largada máxima de 6
   5. Teléfono
      1. Una largada máxima de 12
3. Usuarios
   1. Nombre
      1. Una largada máxima de 25
   2. Apellidos
      1. Una largada máxima de 25
   3. DNI
      1. Con una expresión regular se ha comprobado que sea un formato de 8 dígitos y una letra al final
   4. Matricula
      1. Una largada máxima de 6
   5. Pass
      1. Una largada máxima de 8
4. Se ha considerado que la hora de editar un expediente o usuario se le mostraran todos los campos si el usuario introduce un String vació no se modificara el campo.
5. Para optimizar el menú según el usuario que esta loggeado se le devuelve un menú u otro yendo a parar todos los menús al mismo switch y para controlar el acceso de el usuario a una determinada acción se hace una validación de su tipo de usuario.
6. A la hora de borrar un usuario se ha considerado que no te puedes borrar a ti mismo ya que se podría quedar la aplicación sin un administrador.

# Consideraciones relacionadas con el diseño

1. Las opciones de menú irán numeradas para que para ejecutarla el usuario ponga el numero de la opción y las opciones de salir y cerrar sesión irán con letras para quedar más claras.
2. Cuando estes loggeado aparte de mostrarte las opciones propias de tu usuario en el menú se mostrara tu nombre de usuario y cuando no estas loggeado solo te dejara salir e iniciar sesión y aparte te saldrá un mensaje de que no estas loggeado en la aplicación.

Alumno: Pol Monleón Vives

Práctica: Práctica 2

Fecha: 11/02/2019