

PRÁCTICA 5 de JAVA:
Bases de datos. Derby. Git

| | | | |
|---------------------------|---|---------------------------|----------------------------|
| Fecha tope entrega | Miércoles, 10 de noviembre | Fecha tope defensa | Miércoles, 17 de noviembre |
| Tipo | 2 o 3 personas | Asunto e-mail | JAVA05 |
| Formato fichero | Apellido1_Nombre1_Apellido2_Nombre2_JAVA05.zip (enlace a git) | | |

- 1.- Crea una base de datos con el ejemplo que se proporciona (empleado) y desarrolla una aplicación para su manejo según se detalla a continuación.
- 2.- Existirá una clase de Java (Emplado) asociada a la tabla de la BDs.
 - 2.1.- Los atributos serán los mismos que los campos de la base de datos.
 - 2.3.- Los únicos métodos permitidos serán los constructores, los get y los set.
- 3.- Los objetos se cargarán desde una base de datos Apache Derby integrada en el IDE Netbeans 8.2.
 - 3.1.- Para el Jlist se recomienda una collection como ArrayList u otra. Mediante un Statement con un ResultSet que solo se puede recorrer hacia delante, ver AccesoA.java.
 - 3.2.- Para visualizar uno a uno los objetos se debe usar sobre la base de datos se utilizará un Statement con un Resultset Scroll. P.E.: ResultSet.TYPE_SCROLL_INSENSITIVE, ver ScrollingDemo.java.
- 4.- Implementa una aplicación gráfica mediante un JFrame con las siguientes opciones:
- 5.- JPanel para **visualizar en un JList** todos los elementos de la tabla Empleados.
 - 5.1.- Inicialmente cargará todos los empleados y después los mostrará.
 - 5.2.- De cada empleado se mostraran todos los datos menos la foto. Se usará toString().
 - 5.3.- Se podrá filtrar por las fechas de alta que estén entre dos fechas dadas.
 - 5.4.- (EXTRA) La fechas dadas se introducirán por **datepicker**.
- 6.- JPanel para **visualizar uno a uno** los elementos de la tabla Empleado.
 - 6.1.- Mostrara todos los datos y la foto como imagen.
 - 6.2.- Inicialmente se mostrarán todos los empleados
 - 6.3.- Tendrá los botones anterior y siguiente.
 - 6.4.- Se podrá filtrar mediante una lista desplegable de apellidos.
 - 6.5.- (EXTRA) EL desplegable se cargará con una consulta.
- 7.- Estará estructura siguiendo el MVC y los correspondientes paquetes.
 - 7.1.- Los errores **solo** se podrán mostrar desde la Vista (filosofía MVC).
 - 7.2.- Hay que tener especial cuidado con el intercambio de información con la base de datos.
- 8.- Debe desarrollarse con alguna herramienta git.