**Proyectos JSP SERVLET JDBC**

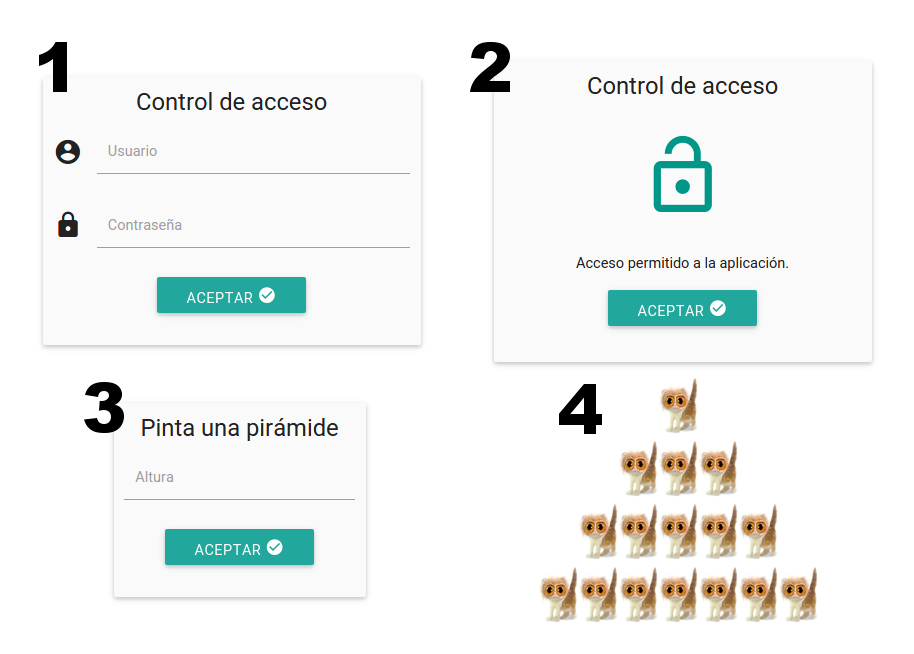
**1. Crea una aplicación Web donde lo primero que aparecerá por pantalla será un formulario pidiendo el nombre de usuario y la contraseña.**

**Si el usuario y la contraseña son correctos,se podrá acceder; en caso contrario, volverá a aparecer el formulario pidiendo los datos de acceso y no se nos dejará ejecutar la aplicación hasta que iniciemos sesión con un nombre de usuario y contraseña correctos.**

**Los nombres de usuario y contraseñas deben estar almacenados en la tabla de una base de datos.**

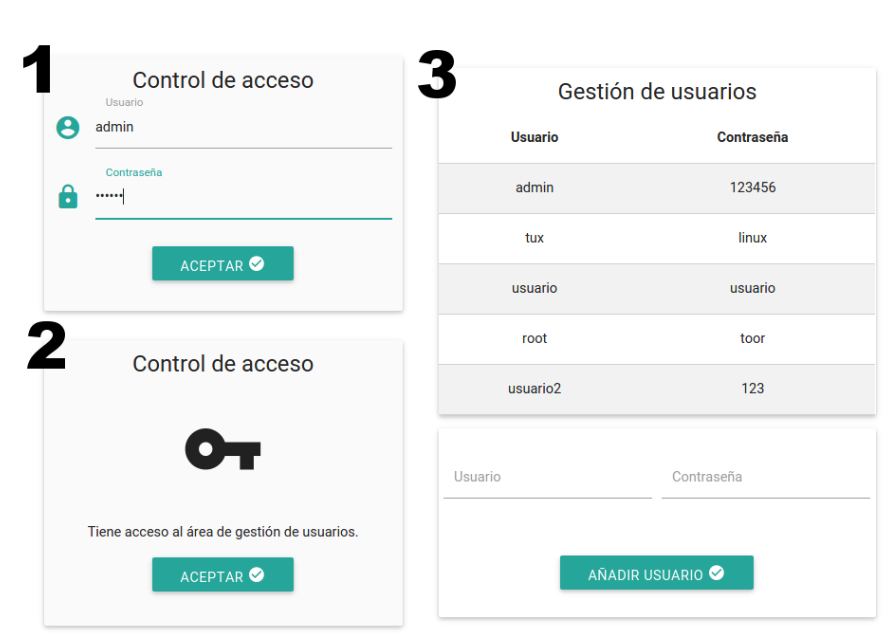
**Una vez iniciado habrá un formulario donde introduzcas una altura para una pirámide de gatos que se renderiza en la página donde se resuelve esa petición.**

**Utiliza estilos como bootstrap, y elige la imagen de gato que tú quieras.**

****

**2. Añade a la aplicación de la pirámide de gatos un CRUD o mantenimiento de usuarios. Si se introduce un nombre de usuario que no sea el administrador (admin) y una contraseña correcta, la aplicación funcionará exactamente igual que el ejercicio anterior. Si se introduce el usuario admin y la contraseña correcta, la aplicación entra en la gestión de usuarios**

**donde se podrán dar de alta nuevos usuarios indicando nombre de usuario y contraseña. No puede haber dos nombres de usuario iguales aunque sí puede haber claves repetidas.**

****

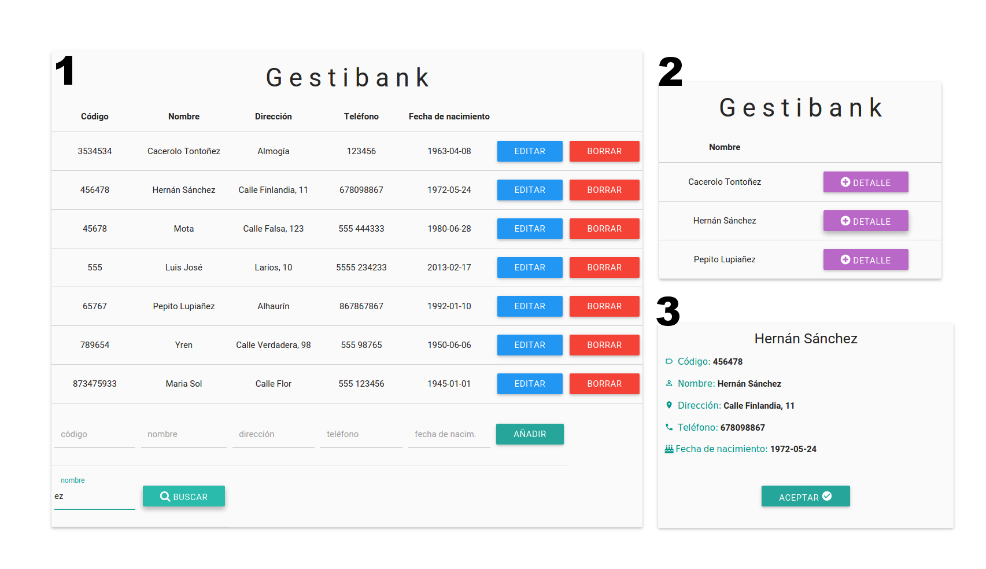
**3. Añade protección en la base de datos a las contraseñas modificando el código para que realice una persistencia en base de datos con la contraseña encriptada (hash). Para ello, tendrás que incorporar el siguiente código a la aplicación:**

***/\*\*  
 \* Método que obtiene el hash de una password, por ejemplo, dado password = admin, devuelve:***

***8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918  
 \* @param password  
 \* @return hash de encriptación de la password  
 \* @throws NoSuchAlgorithmException  
 \*/*  
 public static String hashPassword(String password ) throws NoSuchAlgorithmException {  
 MessageDigest digest;  
   
 digest = MessageDigest.getInstance("SHA-256");  
 byte[] encodedhash = digest.digest(  
 password.getBytes(StandardCharsets.UTF\_8));  
   
 return bytesToHex(encodedhash);   
   
 }  
   
 private static String bytesToHex(byte[] byteHash) {  
   
 StringBuilder hexString = new StringBuilder(2 \* byteHash.length);   
 for (int i = 0; i < byteHash.length; i++) {  
 String hex = Integer.toHexString(0xff & byteHash[i]);  
 if(hex.length() == 1) {  
 hexString.append('0');  
 }  
 hexString.append(hex);  
 }  
   
 return hexString.toString();  
   
 }**

**4. Crea el mantenimiento (CRUD + buscador) de usuarios de un banco como el siguiente:**

**La aplicación debe permitir hacer listado, alta, baja y modificación sobre la tabla cliente de la base de datos banco. Un cliente tiene su identificador, nombre completo, dirección, teléfono y fecha de nacimiento. La aplicación buscará la cadena introducida dentro del campo “nombre” y, si hay varias coincidencias, se mostrará una lista de clientes para poder seleccionar uno de ellos y ver todos los datos. Si solo hay una coincidencia, se mostrarán directamente los datos del cliente en cuestión.**

****