|  |  |
| --- | --- |
| CV MANAGER  JSP - SERVLET | **CV MANAGER**  Eduardo Zaldivar - Rafael Cárdenas  **Asus**  2 DAW |

# Índice

[Gestión de usuarios interno 2](#_Toc482368986)

[Definición 2](#_Toc482368987)

[Motor del aplicativo: 2](#_Toc482368988)

[Formulario de acceso (login) 2](#_Toc482368989)

[Perfil User 3](#_Toc482368990)

[Archivo userCV.jsp 4](#_Toc482368991)

[Perfil Administrador 5](#_Toc482368992)

[AdminCV.jsp 5](#_Toc482368993)

[LoginServlet.java 6](#_Toc482368994)

[Vistas 7](#_Toc482368995)

[Login 7](#_Toc482368996)

[User 7](#_Toc482368997)

[Admin 8](#_Toc482368998)

[Base de datos 8](#_Toc482368999)

[Login 8](#_Toc482369000)

[Basic\_information 9](#_Toc482369001)

[relation\_username\_basic\_info 9](#_Toc482369002)

[Skill 9](#_Toc482369003)

[Relation\_skill\_percentage 10](#_Toc482369004)

[Language 11](#_Toc482369005)

[Relation\_lang\_percentage 11](#_Toc482369006)

# Gestión de usuarios interno

## Definición

Este aplicativo está pensado para empresas que pretenden aprovechar y valorar al máximo las habilidades de sus empleados.

Se trata de una aplicación dentro de una empresa que será capaz de realizar un login diferenciando los roles de acceso de cada usuario, estos perfiles serán rellenados por cada usuario ya que contendrá datos privados de los mismos, por ejemplo:

Se va a tener dos roles en este aplicativo el administrador y el usuario cliente

Admin: tendré una vista y permiso para crear, editar, o eliminar usuarios, también podrá aumentar contenidos, imágenes, etc.

User: este nivel solo tendrá acceso para editar ciertos datos aun no definidos, por ejemplo: datos personales, datos académicos, etc.

Sin embargo, no podrá eliminar y agregar ningún usuario

## Motor del aplicativo:

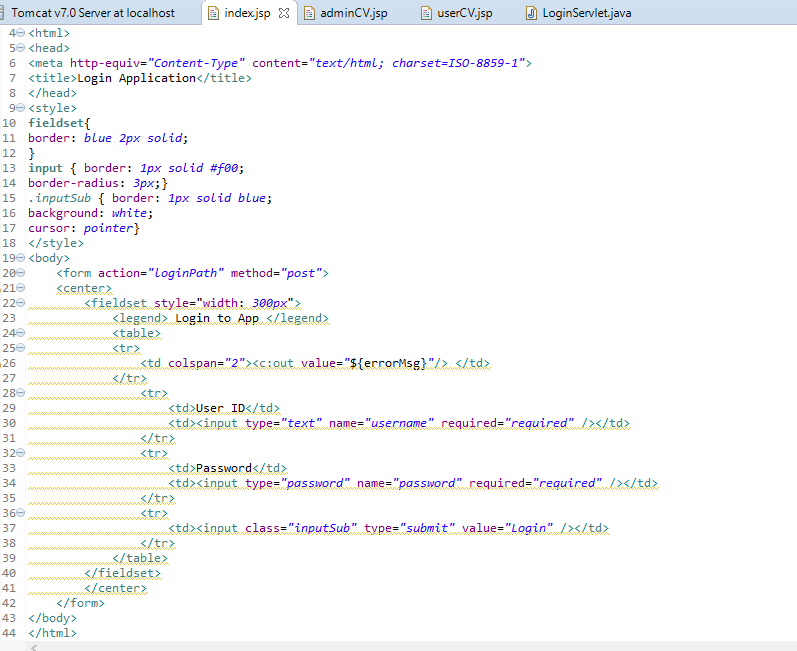
El usuario administrador será capaz de crear al usuario empleado, editar o eliminar, y limitar sus roles.

Una vez creado el acceso y perfil del user. El User podrá editar sus datos personales, académicos y laborales.

La vista de cada perfil será diferente dependiendo del rol que tenga cada usuario.

## Formulario de acceso (login)

En el formulario el aplicativo será capaz de diferenciar el rol del usuario (admin or user), cuenta con un desplegable para que el usuario pueda elegir a que perfil pertenece.



El archivo index.jsp contendrá el formulario inicial para comenzar el login y también se ha colocado una línea de error para la autentificación, por ejemplo, si el usuario o la contraseña con incorrectas.

## Perfil User

En este perfil se pretende realizar que el usuario pueda editar sus datos personales, por ejemplo:

Nombre

Apellido

Fecha de nacimiento

\*uno de los valores que no podrá editar, aunque forme parte de sus datos personales será su carnet de identificación, y su Nick ya que estos valores están anexados como campos importantes en la base de datos

También podrá editar sus datos de formación y laboral, estos datos serán editables, pero también de mucha importancia ya que la empresa podrá valorar al empleado para futuros puesto de trabajo.

### Archivo userCV.jsp



En esta imagen se ve un ejemplo para capturar la información de las consultas de la anterior imagen. Se captura con un sesión.getAtribute



Esta es la vista de los usuarios con rol user, se ha realizado consultas en la base de datos para identificar al usuario que ha realizado el login. Además las consultas nos sirve para recoger los datos ya guardados como son el nivel de idioma y de sus habilidades.

## Perfil Administrador

En este perfil el usuario administrador podrá realizar cambios, dar nuevas alta y modificar cualquier perfil, a este rol solo tendrá acceso los profesores y el informático de turno.

Aquí imagen del perfil administrador

Estos perfiles podrán editar los datos como número de identificación y Nick de los empleados.

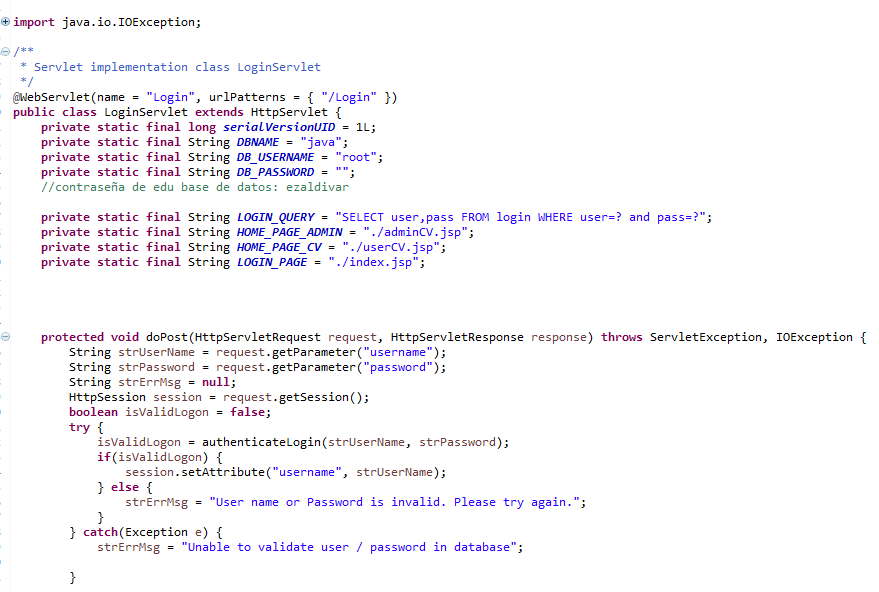
También tendrá una vista de los empleados con actividades recientes, y estadísticas de los mismo

### AdminCV.jsp

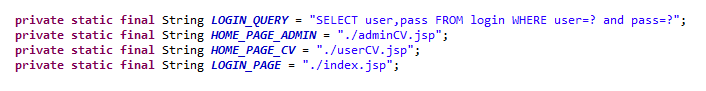


Se aprecia la vista de los usuarios con rol admin, aquí también podremos apreciar la recogida de datos a través de consultas para poder identificar al usuario, y sus datos.

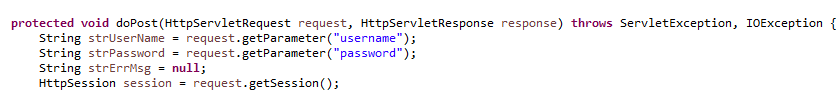
### LoginServlet.java



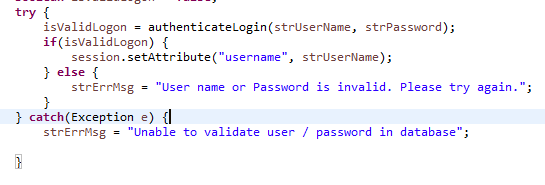
En este archivo tenemos varias operaciones importantes como por ejemplo el re direccionamiento del login, una vez se envía el formulario para ingresar en algún perfil de usuario, la consulta se realiza aquí en este archivo.



Se declara la consulta para autentificar al usuario entre el user que es el Nick y el pass que es la contraseña.

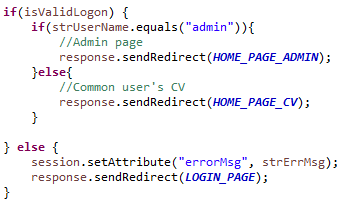


Recogemos los valores del formulario, y vemos si está en la base de datos.



Una vez identificado el usuario se compara con el Nick para ver si el admin o user.

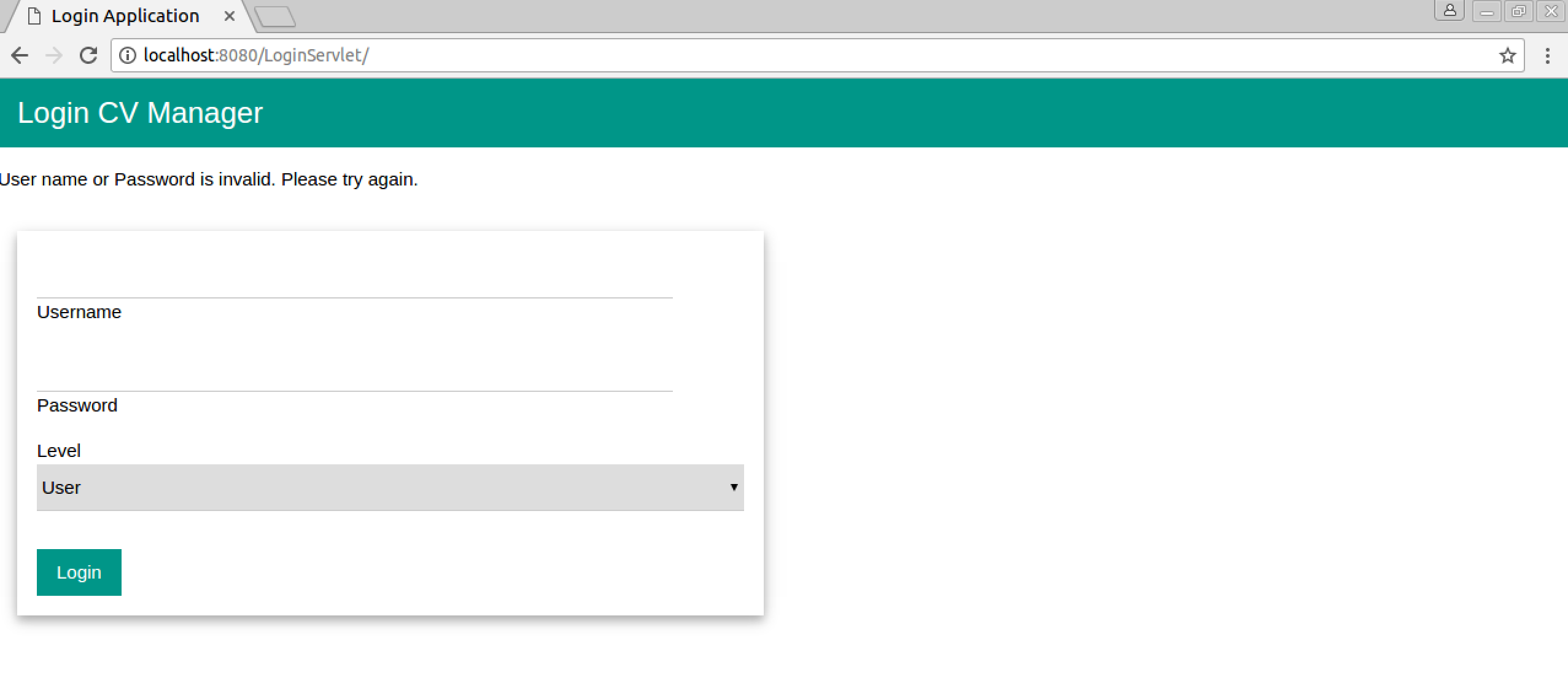
Se realiza el re direccionamiento del usuario dependiendo de su rol



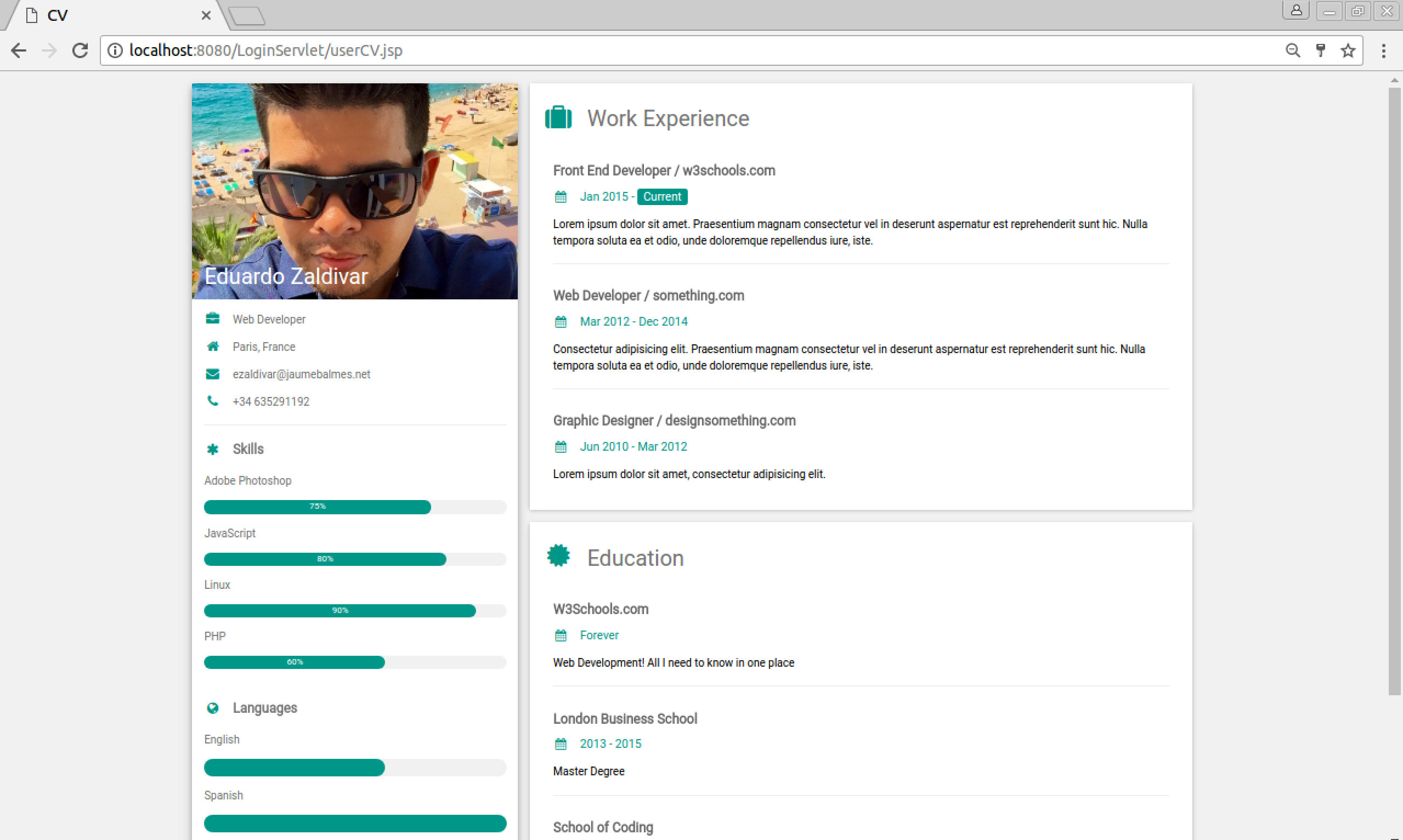
En caso de que no se logre autentificar su nivel re dirección a la página login para que lo vuelva a intentar.

## Vistas

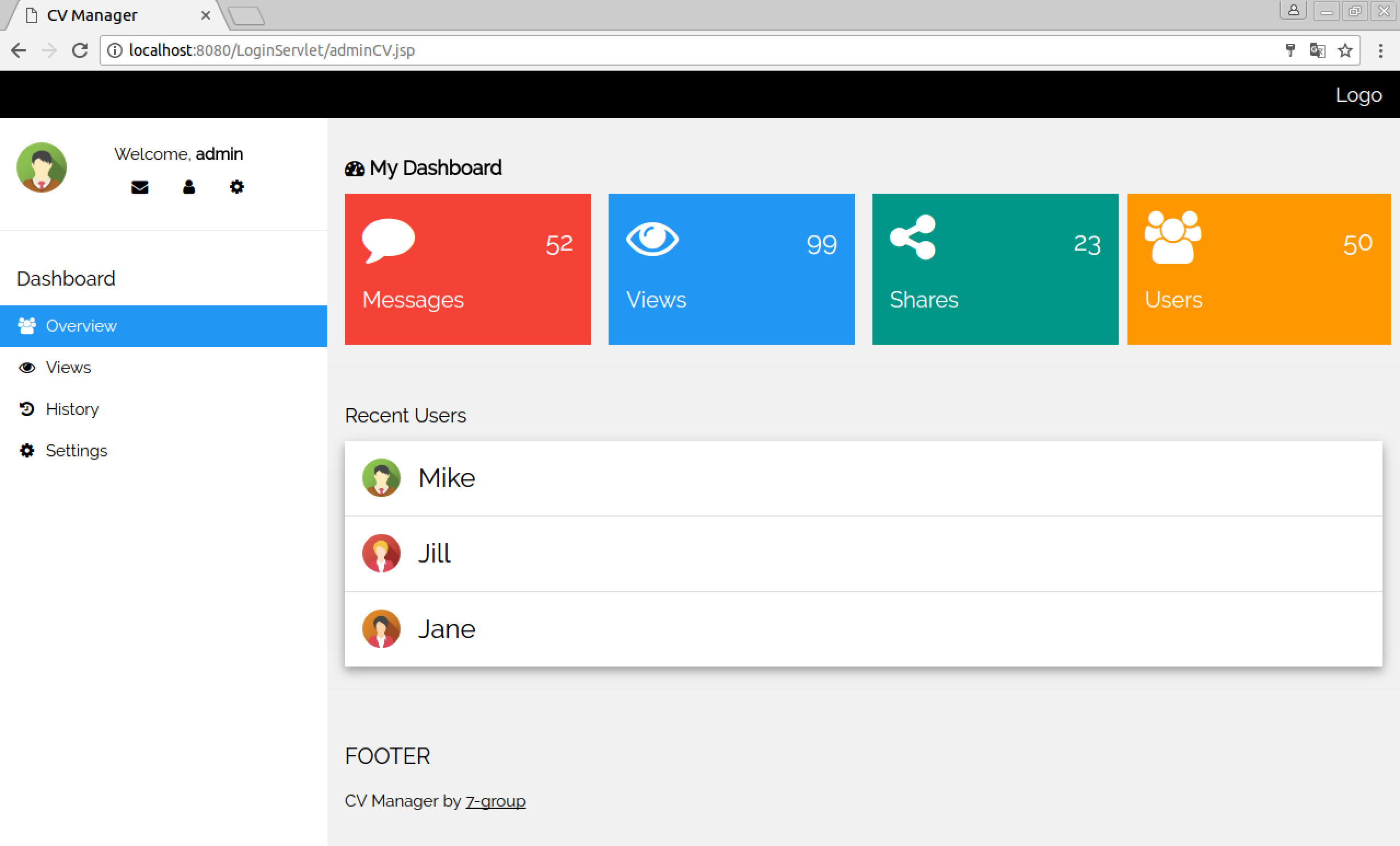
### Login



### User



### Admin

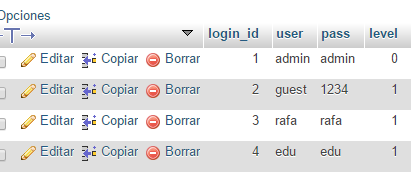


## Base de datos

La base de datos cuenta con 7 tablas.

### Login

Contiene información del usuario: su Nick, su password y su level



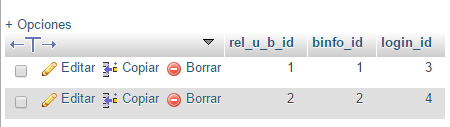
### Basic\_information

Contiene la información de los usuarios



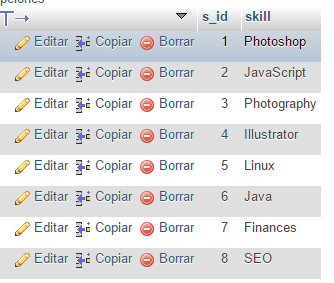
### relation\_username\_basic\_info

Contiene la relación que se ha realizado entre las tablas basic\_information y login



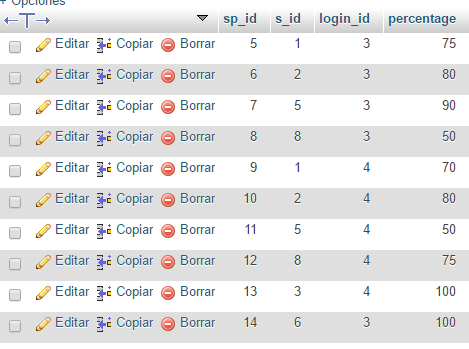
### Skill

Esta tabla contiene la información de habilidades de los usuarios



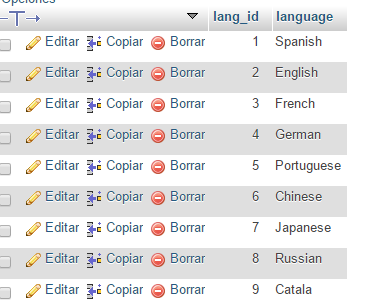
### Relation\_skill\_percentage

Esta tabla relaciona a los usuarios con sus habilidades y además con el porcentaje que tiene de conocimiento de cada habilidad.



### Language

Contiene la información de los idiomas que pueden elegir los usuarios



### Relation\_lang\_percentage

Esta tabla relaciona los usuarios con el idioma y el nivel que tienen de cada uno.

