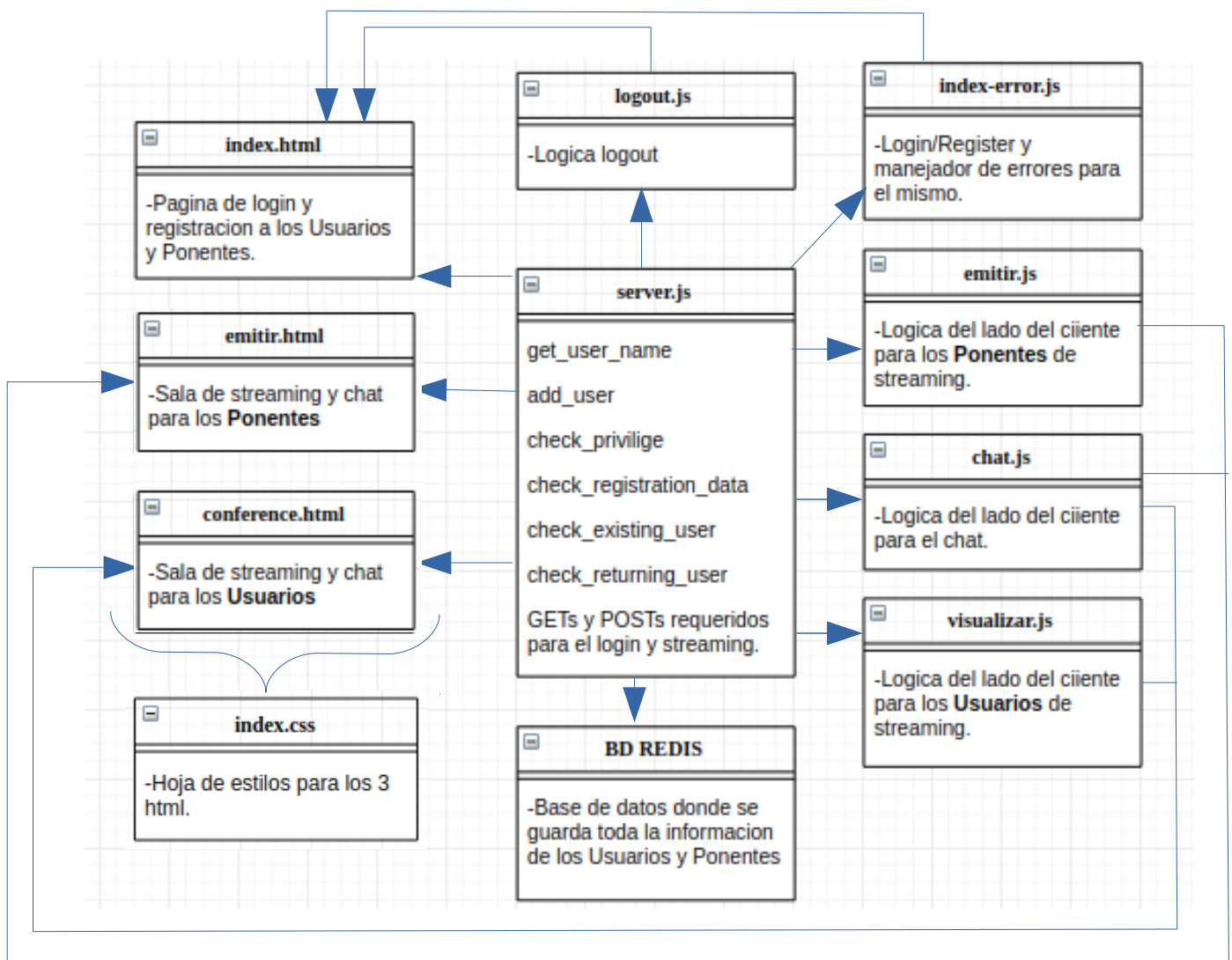


## Documentación de diseño de la Aplicación

### -Diagrama de Clases/Componentes:



### -Ejemplo de despliegue de la Aplicación:

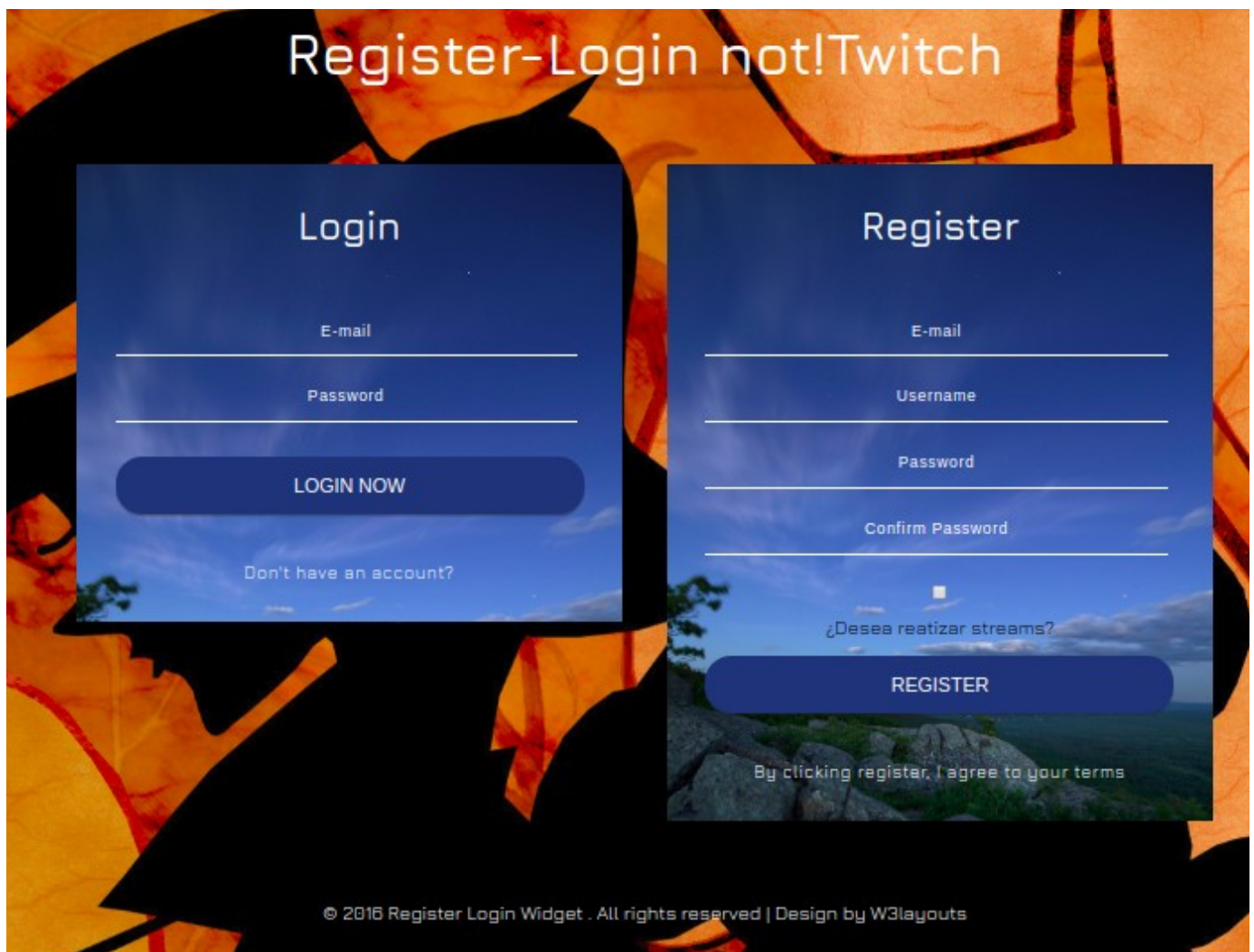
Luego de haber corrido nuestra App (véase **README.md**)...

Accedemos desde un navegador a <http://ocalhost:8080/>, donde nos encontraremos con la pagina principal de **login** y **register** de la App (nuestro [/index.html](#)).

Estos 2 formularios cuentan con **validaciones**: de contraseña (mayor a 6 caracteres), e-mail y de verificación de usuarios (que no estén registrados en la BD).

Desde aquí podremos registrarnos en “Register” (como **Ponente** tildando el checkbox “¿Desea realizar streams?”; o como **Usuario** sin tildar en el mismo).

[/index.html](#):



Este **Ponente** o **Usuario** se guardara en una base de datos en **Redis**.

Al registrarnos nos redireccionará automáticamente a **/emitir.html** (en caso de ser **Ponente**) o a **/conference.html** (en caso de ser **Usuario**) para empezar con el streaming y chat.

A **/conference.html** se podrán conectar todos los Usuarios que quieran visualizar el streaming del Ponente (sea para el dictado de cursos, conferencias y/o charlas en vivo). Este archivo .html posee

un elemento html **<canvas>**, el cual es usado para dibujar un grafico en 2 dimensiones con las imágenes que el servidor le envía (estas imágenes son las imágenes en vivo de la cámara web del Ponente que provienen de /emitir.html y que son tomadas cada 70ms).

Y en **/emitir.html** (donde accede el Ponente) no tenemos un canvas, sino que es directamente un elemento **<video>** con el video de la cámara web del ponente.

Ambos .html cuentan con un botón “LOGOUT” para desloguearse.

/emitir.html (de donde podrán acceder varios **Usuarios**):



/conference.html (de donde podrá acceder un solo **Ponente**):

