

# Caso de uso 1 - EIMS

Caso de Uso: *autenticación de usuarios en el sistema EIMS*

Actores: *usuarios*

Propósito: *Restringir acceso al sistema a usuarios previamente registrados en el, para posteriormente ser redireccionados a su respectivo panel dependiendo del tipo de usuario que es.*

Resumen: *Este caso de uso comienza cuando un usuario (independiente del tipo) desea hacer ingreso al sistema EIMS para llevar a cabo sus funciones, si sus datos existen, procede a realizar sus labores, una vez finalizadas estas, procede a cerrar su sesión.*

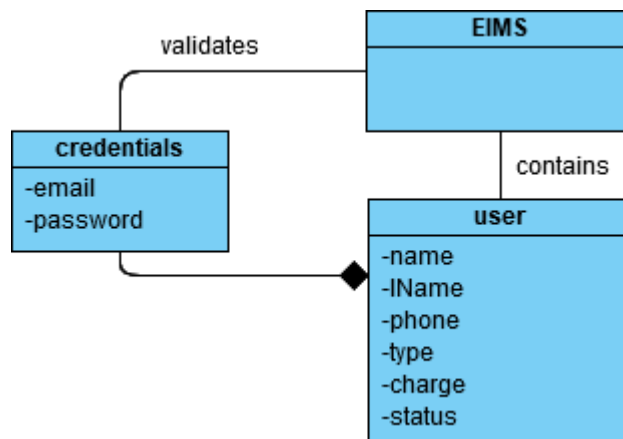
## Curso Normal de los Eventos:

Usuarios	Sistema
1.- Este caso de uso comienza el usuario se dirige en el navegador para acceder al portal de inicio de sesión del sistema EIMS.	
2.- El usuario ingresa sus datos en el formulario web de inicio de sesión.	3.- Sistema comprueba que los datos existan y estén correctamente ingresados bajo las siguientes reglas: <ul style="list-style-type: none"><li>• Nombre de usuario existe (email).</li><li>• Contraseña coincide con la del registro del usuario.</li></ul> Una vez validadas se redirecciona al panel que corresponda según tipo de usuario.
4.- Usuario termina sus labores y realiza cierra sesión en el sistema EIMS.	5.- Sistema termina la sesión del usuario. Retorna a la página principal.

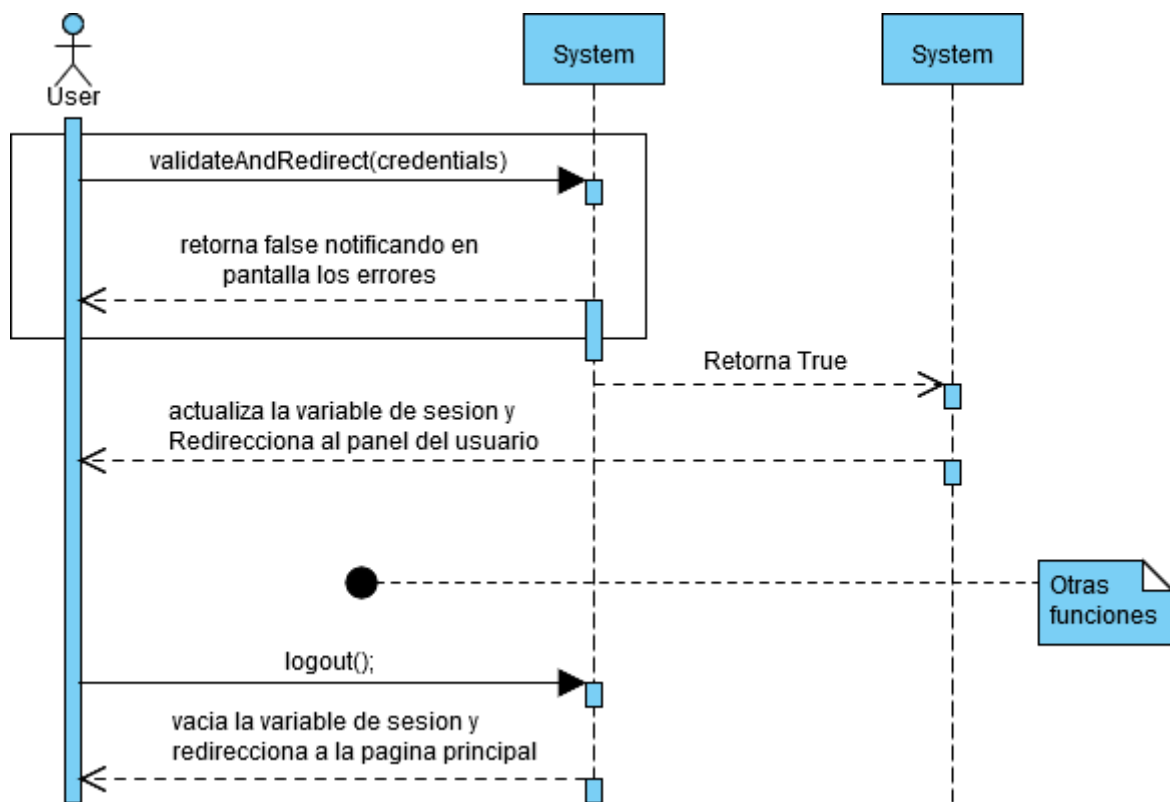
## Curso Alternativo de los Eventos (Login failed):

	3.1.- Sistema encuentra alguno de los siguientes errores en los datos ingresados: <ul style="list-style-type: none"><li>• Usuario no existe.</li><li>• Usuario existe pero la contraseña es invalida.</li><li>• Se ha excedido el número máximo de intentos.</li></ul> Sistema notifica al usuario del o los errores y le permite volver a intentar.
3.2.- Usuario cierra sesión en el sistema EIMS	

### Diagrama conceptual:



### Diagrama de Secuencias



### Contratos

#### **Contrato 1:**

#### **Responsabilidad:**

`validateAndRedirect(Credentials);`  
Valida que los datos se encuentren en los registros y redirecciona al portal correspondiente según tipo de usuario.

#### **Precondición:**

- Usuario debe existir en base de datos.
- Ingreso de datos debe ser validado.

**Postcondición:**- `B = validate(Credentials) // retorna usuario validado.`

- `This.Auth(B) //inicializa variable de sesión.`

**Contrato 2:**

**Responsabilidad:**

Logout();

Se encarga de terminar la sesión activa del usuario y de redireccionar a la página principal-

**Precondición:**

- This.Auth() != null //Tener sesión activa.

**Postcondición:**

- This.Auth() = null //Vacía variable de sesion.

**Diagrama de Colaboración**

