

# Caso de uso 2 - EIMS

Caso de Uso: Creación de nuevas cuentas de usuarios

Actores: Administrador

Precondición: CU1

Propósito: Permitir al usuario con perfil de administrador poder ver, crear, actualizar y eliminar cuentas .

Resumen: Este caso de uso comienza cuando el administrador necesita otorgar acceso al sistema EIMS a un usuario que aún no ha sido registrado para poder hacer uso de él, validando y registrando sus datos en el sistema a través de un formulario web en una vista adecuada.

## Curso Normal de los Eventos:

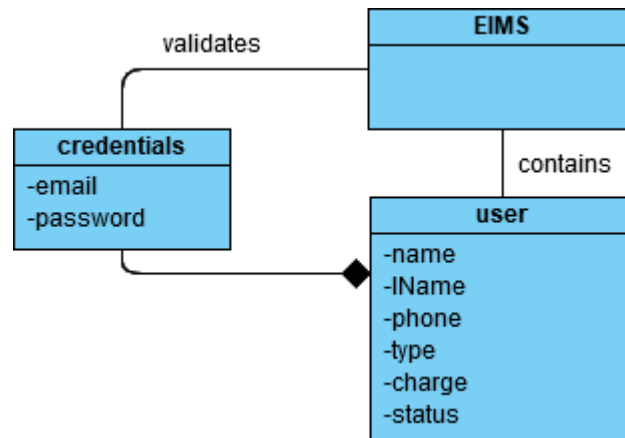
Administrador	Sistema
1.- Este caso de uso comienza cuando el administrador (una vez autenticado) se dirige al apartado “crear usuario” en el sistema EIMS.	2.- Sistema retorna la vista con el formulario de creación de usuarios.
3.- El administrador ingresa los datos del nuevo usuario en el formulario.	4.- Sistema verifica que el formulario se haya llenado correctamente, esto es: <ul style="list-style-type: none"><li>● campo nombre debe ser ingresado.</li><li>● campo apellido debe ser ingresado.</li><li>● campo cargo debe ser ingresado.</li><li>● campo teléfono debe ser ingresado y ser del formato +56 seguido de 9 dígitos.</li><li>● campo email debe ser ingresado.</li><li>● campo email debe ser único.</li></ul> notifica del éxito de la operación y consulta si desea crear otro usuario.
5.- Administrador termina sus labores y cierra sesión.	6.- Sistema termina la sesión del administrador Retorna la página principal.

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

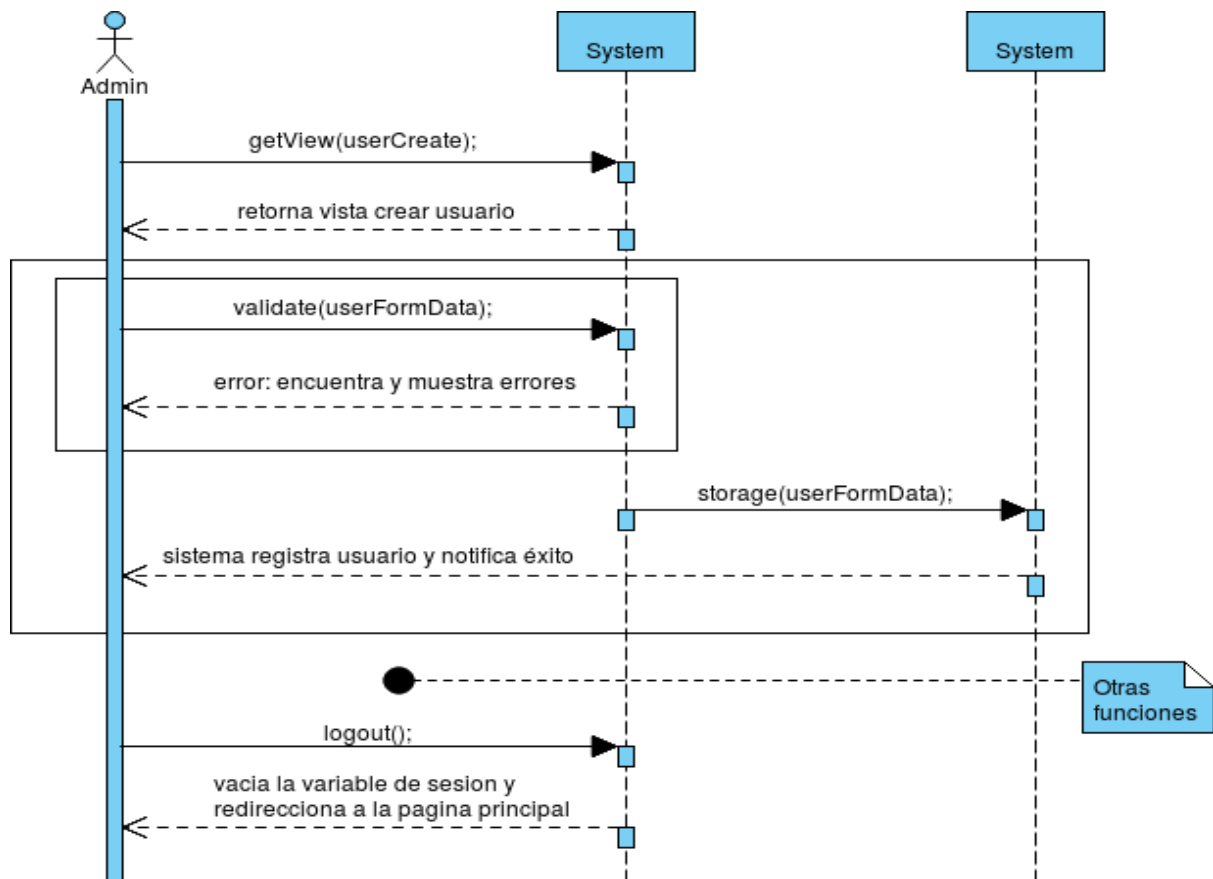
	4.1.- Sistema encuentra error(es) en los datos ingresados en el formulario que pueden ser: <ul style="list-style-type: none"><li>● campo nombre no se ha ingresado.</li><li>● campo apellido no se ha ingresado.</li><li>● campo cargo no se ha ingresado.</li><li>● campo teléfono no se ha ingresado.</li><li>● campo teléfono no cumple el formato especificado.</li><li>● campo email no se ha ingresado.</li></ul>
--	---

	<ul style="list-style-type: none"> <li>• campo email ya está en uso. notifica error(s) y le permite volver a intentar.</li> </ul>
3.2.- Usuario cierra sesión.	

### Diagrama Conceptual:



### Diagrama de Secuencias



## Contratos

### **Contratos 1:** getView(userCreate)

**Responsabilidad:** Despliega la vista "Crear usuario" que contiene el formulario web en el que se solicita los datos necesarios para poder registrar un nuevo usuario en el sistema.

**Precondición:** - This.Auth() != null //Tener sesión activa.  
-Tipo de usuario es "Admin".

### **Contratos 2:** validate(userFormData)

**Responsabilidad:** Se encarga de verificar que la información ingresada corresponda al formato requerido, indicando al usuario los errores para que pueda corregirlos.

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

**Postcondición:** - validate(userFormData) = True //Formulario llenado correctamente.  
- storage(userFormData).

### **Contrato 3:** storage(userFormData)

**Responsabilidad:** Se encarga de almacenar los datos capturados del formulario web en base de datos, entregar el mensaje de éxito y redireccionar a la vista crear usuarios nuevamente con dicho mensaje.

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

**Postcondición:** - validate(userFormData) = True //Formulario llenado correctamente.  
- User::create(userFormData->all()).

"Success message".

- M =

Equivale a crear una instancia del modelo usuario y llenarlo con todo

el contenido del 'userFormData'. Esto es:

```
create u = new::User();
u.name = userFormData.name;
u.lName = userFormData.lName;
u.email = userFormData.email;
u.phone = userFormData.phone;
u.charge = userFormData.charge;
u.status = 'active' // valor por defecto al crear
u.password = bcrypt(123456) // valor encriptado por
```

### **Contrato 4:** logout();

**Responsabilidad:** Se encarga de terminar la sesión activa del usuario y redireccionar a la página principal.

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

**Postcondición:** - This.Auth() = null.

## Diagrama Colaboración:

