



*Universidad Nacional de Lanús*

## DESARROLLO DE SOFTWARE EN SISTEMAS DISTRIBUIDOS

### **Llamado a Procedimiento Remoto (RPC)**

## Enunciado

Utilizando **gRPC** generar un sistema que cumpla los requerimientos más abajo detallados, mediante el siguiente esquema:

1. Una interfaz web para operar con el sistema.
2. Un api gateway que recibirá las peticiones de la vista, llamando a los servicios correspondientes según el caso:
  - a. Gestión de usuarios.
  - b. Inventario de donaciones.
  - c. Eventos solidarios.

Dichos servicios son independientes, y tanto la parte cliente como la parte servidor deben ser desarrolladas en lenguajes diferentes. Por ejemplo: cliente en C# y servidor en Python, etc.



*Arquitectura propuesta*

## Requerimientos

La ONG "**Empuje Comunitario**" nos encarga el desarrollo de un sistema web para gestionar las actividades que llevan a cabo, mantener un registro de las donaciones y administrar a sus miembros. En principio, los requerimientos que la organización nos encargó implementar son los siguientes:

- 1) *Roles de usuario*
  - a. **Presidente:** será el único rol que pueda gestionar los usuarios, además de poder realizar cualquier acción correspondiente a los demás roles.
  - b. **Vocal:** accede al **Inventario de donaciones**.
  - c. **Coordinador:** se encargará de coordinar los **Eventos solidarios**.
  - d. **Voluntario:** podrá consultar los **Eventos solidarios** para participar en ellos.
- 2) *Gestión de usuarios – Roles: PRESIDENTE*
  - a. Se deberá contar con la vista correspondiente para el ABM de usuarios. Cada usuario contempla los siguientes campos:
    - Id: clave primaria (no se muestra en la vista)
    - Nombre de usuario: único (obligatorio)
    - Nombre (obligatorio)
    - Apellido (obligatorio)
    - Teléfono
    - Clave: encriptada, generada al azar en el alta.
    - Email: único (obligatorio)
    - Rol
    - Activo/Inactivo
  - b. **Alta:** al dar de alta al usuario, su clave se genera al azar, se envía por e-mail al usuario y se guarda en la base de forma encriptada
  - c. **Modificación:** se puede modificar todos los campos, excepto la clave.
  - d. **Baja:** no se borran físicamente de la base, solamente se los deja como inactivos.

### 3) *Login – Roles: TODOS*

- a. Para ingresar al sistema, se podrá hacer por nombre de usuario o email, y la contraseña.
- b. Se deberá mostrar un mensaje descriptivo según el error (usuario/email inexistente, credenciales incorrectas, etc).

### 4) *Inventario de donaciones – Roles: PRESIDENTE, VOCAL*

- a. Permitirá registrar el inventario según diferentes categorías: ROPA, ALIMENTOS, JUGUETES, UTILES\_ESCOLARES. Además, se registrará:
  - Id: clave primaria (no se muestra en la vista)
  - Categoría (obligatorio)
  - Descripción: por ejemplo "remeras"
  - Cantidad (no puede ser negativo)
  - Eliminado
  - Campos de auditoría (no se muestran en la vista):
    - Fecha y hora de alta
    - Usuario de alta
    - Fecha y hora de modificación
    - Usuario de modificación
- b. **Modificación:** se podrá modificar la descripción y cantidad.
- c. **Baja:** se aplica baja lógica, pidiendo confirmación y poniendo el flag de **Eliminado** en true. Esta acción deja registro de auditoría igual que la modificación.

### 5) *Eventos solidarios – Roles: PRESIDENTE, COORDINADOR, VOLUNTARIO (solo participación).*

- a. Los eventos solidarios son actividades que la organización realiza en una fecha determinada junto a un grupo de sus miembros. Pueden ser visitas a centros de ancianos, hogar de niños, escuelas, etc. Dependiendo el evento, es posible que se repartan las donaciones que la organización tiene en su inventario. Para registrar los eventos, se contará con estos campos:
  - Id: clave primaria (no se muestra en la vista)
  - Nombre del evento: por ejemplo "Visita a la escuela nº 99"
  - Descripción del evento: por ejemplo "Se organizarán juegos y repartirán útiles".
  - Fecha y hora del evento (debe ser a futuro, no se pueden crear eventos pasados)
  - Miembros que participarán del evento.
- b. **Modificación:** se podrá modificar el nombre del evento, fecha y hora, asignar/quitar miembros. Si se trata de un evento pasado, deberá poder registrarse las donaciones que se repartieron, dejando asentado quién hizo la modificación además de descontar las cantidades registradas del inventario.
- c. **Baja:** solo se pueden dar de baja eventos a futuro, y se eliminan físicamente de la base de datos.
- d. **Asignar/quitar miembros:** tanto PRESIDENTE como COORDINADOR podrán asignar o quitar a cualquier miembro, solamente serán elegibles aquellos miembros activos para asignar. Si se da de baja un miembro, se quitan únicamente de los eventos a futuro.  
El rol VOLUNTARIO solamente podrá agregarse o quitarse a él mismo de los eventos.

Las vistas cargarán los listados de los registros disponibles para cada una. Por el momento, **no se aplicará filtrado**, pero será una funcionalidad que posiblemente se nos pedirá implementar a futuro.

## NORMAS DE ENTREGA

El trabajo entregado deberá contener un documento incluyendo:

- La estrategia de resolución del trabajo práctico: todo aquello que consideren significativo para explicar la resolución del trabajo: diagrama del modelo de datos, arquitectura del sistema, etc.
- El código fuente, **DE PROPIA AUTORÍA**, del proyecto subido a un repositorio público de **Github**.
- Las pruebas realizadas con las respectivas capturas de pantalla.
- Integrantes del grupo y las tareas realizadas por cada uno.

El incumplimiento de cualquiera de las normas de entrega implicará la desaprobación del trabajo práctico.