

Documento de especificación de la API

19/4/2023 : Versión 1.0

-El sistema deberá gestionar usuarios

-El sistema deberá gestionar grupos

Para ello mediante un análisis de dichos requerimientos se llega a la conclusión de que se necesitaran de 4 interfaces, las cuales se detallan a continuación.

<<Interface>> User quien contara con métodos como los siguientes:

getID(): string

getGruptMembership(string: id): list<string>

getIsActive(): boolean

<<Interface>> USERHandler quien contara con métodos como los siguientes:

Create(UserData: data): boolean

Remove(string: id): boolean

Update(string: id):boolean

Read(string: id): UserData

getGruptMembership(string: id): list<string>

<<Interface>> Group quien contara con métodos como los siguientes:

getID(): string

getAmount(): string

getIsActive(): boolean

<<Interface>> GroupHandler quien contara con métodos como los siguientes:

Create(UserData: data): boolean

Remove(string: id): boolean

Update(string: id):boolean

Read(string: id): UserData

addUser(string: id): boolean

removeUser(string: id): boolean

Especificación de datos(UserData):

ID: string
name: string
surname: string
DNI: string
telephone: string
gender: string
address: string
mail: string
role: string
isActive: boolean

Especificación de datos(GroupData):

ID: string
name: string
isActive: boolean
amount: string

10/5/2023: Versión 1.1

El sistema deberá gestionar permisos/accesos a recursos de información de los usuarios.

Se plantea de manera que necesitaremos de dos interfaces principales quienes nos darán el comportamiento necesario para acceder al recurso y una interface para interactuar directamente con el recurso en si.

Se plantea que:

<<Interface>> Authorizer:

Authorize(string: userID,string: resourceID): boolean

<<Interface>> AccessHandler:

Add(string: resourceID,string: groupID): boolean

Remove(string: resourceID,string: groupID): boolean

getGroupAccessByResource(string: resourceID): list<string>

getResourceAccessByGroup (string: groupID): list<string>

Luego contaremos con dos estructuras que hasta el momento no exhiben comportamiento por si mismas.

<<Interface>> ResourceHandler:

Create(resourceData: data): boolean

Remove(string: id): boolean

Update(string: id):boolean

Read(string: id): ResourceData

Luego contaremos con dos estructuras que hasta el momento no exhiben comportamiento por si mismas.

Especificación de datos(RESOURCE):

ID: string

name: string

isActive: boolean

data: list<string>

Especificación de datos(ACCESS):

ID: string

managmentLevel: string

