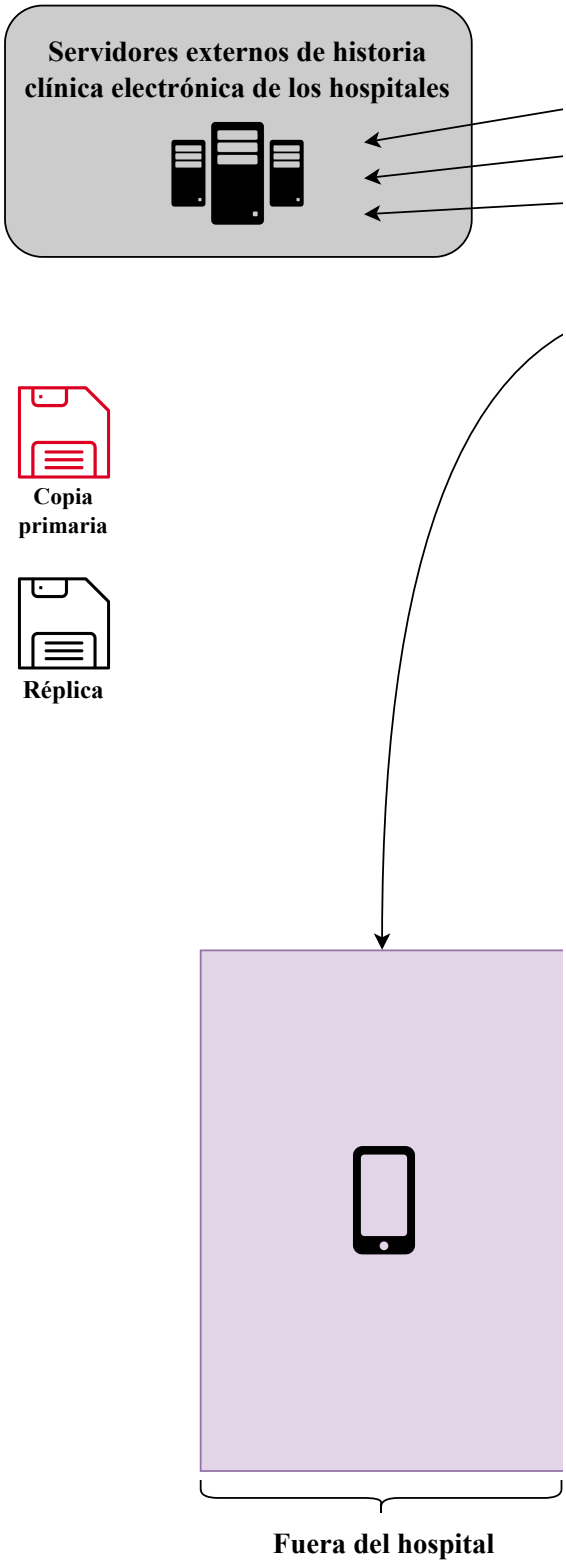
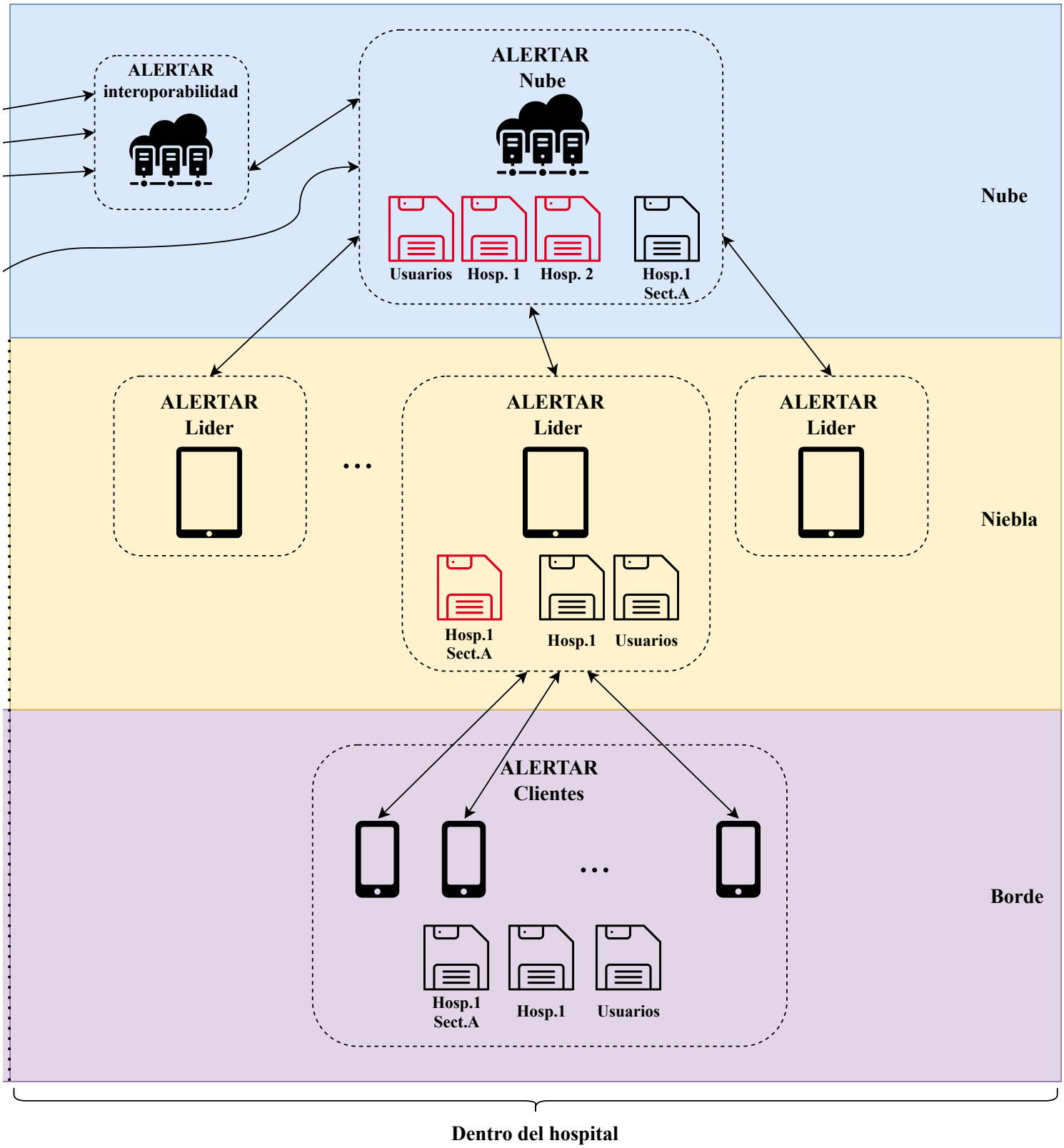
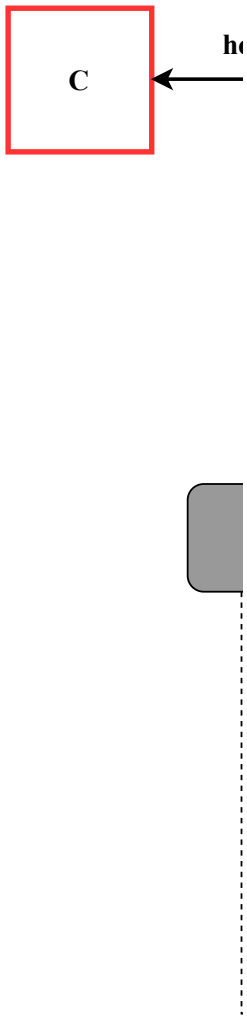
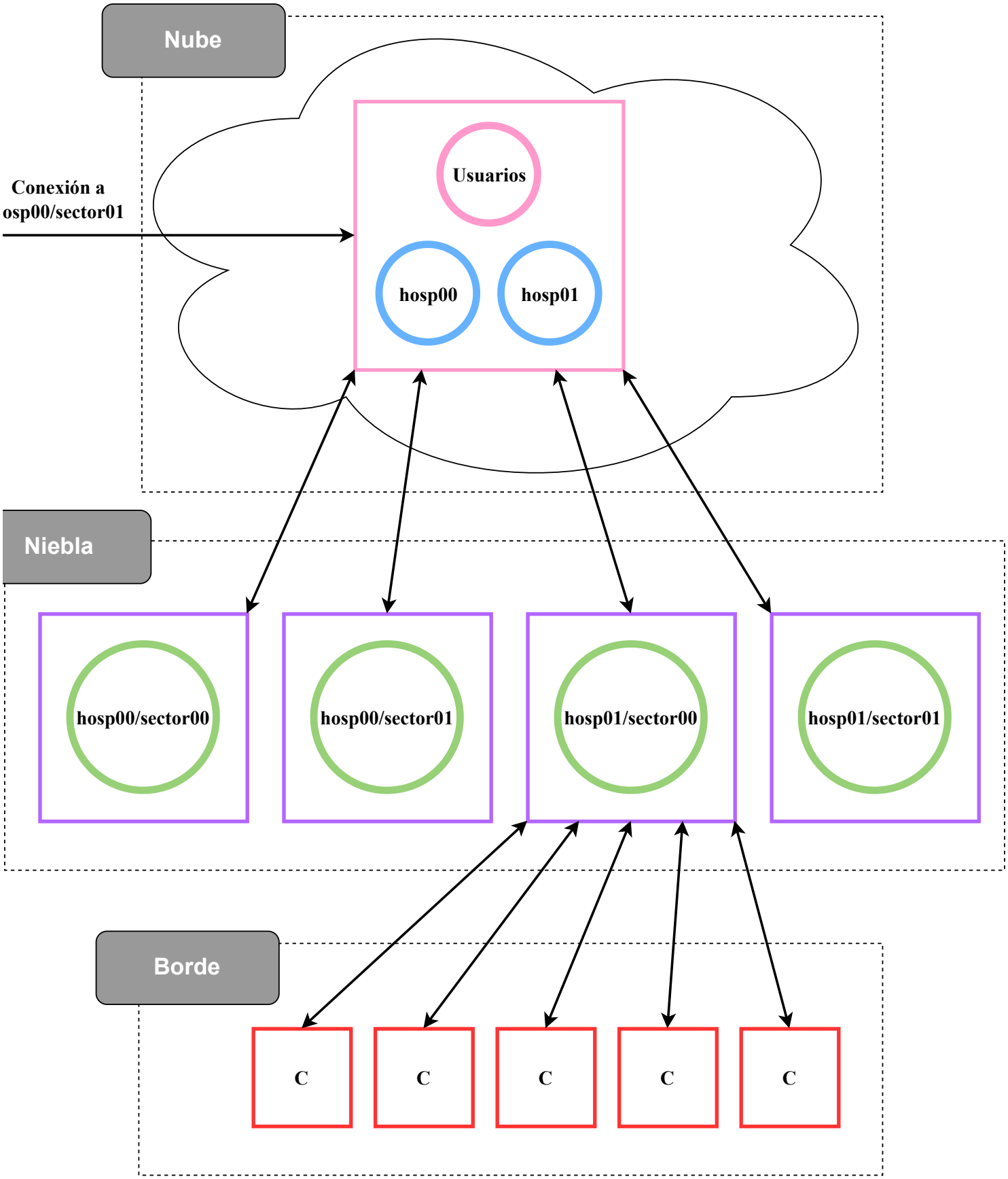

Servidores externos de historia clínica electrónica de los hospitales
Copia primaria
Réplica
Fuera del hospital


ALERTAR
interoporabilidad
ALERTAR
Nube
Usuarios
Hosp. 1
Hosp. 2
Hosp.1
Sect.A
Nube
ALERTAR
Lider
ALERTAR
Lider
ALERTAR
Lider
Hosp.1
Sect.A
Hosp.1
Usuarios
Niebla
ALERTAR
Clientes
Hosp.1
Sect.A
Hosp.1
Usuarios
Borde
Dentro del hospital


C


Nube
Usuarios
Conexión a
osp00/sector01
hosp00
hosp01
Niebla
hosp00/sector00
hosp00/sector01
hosp01/sector00
hosp01/sector01
Borde
C
C
C
C
C

Copia primaria de dominio Sectores

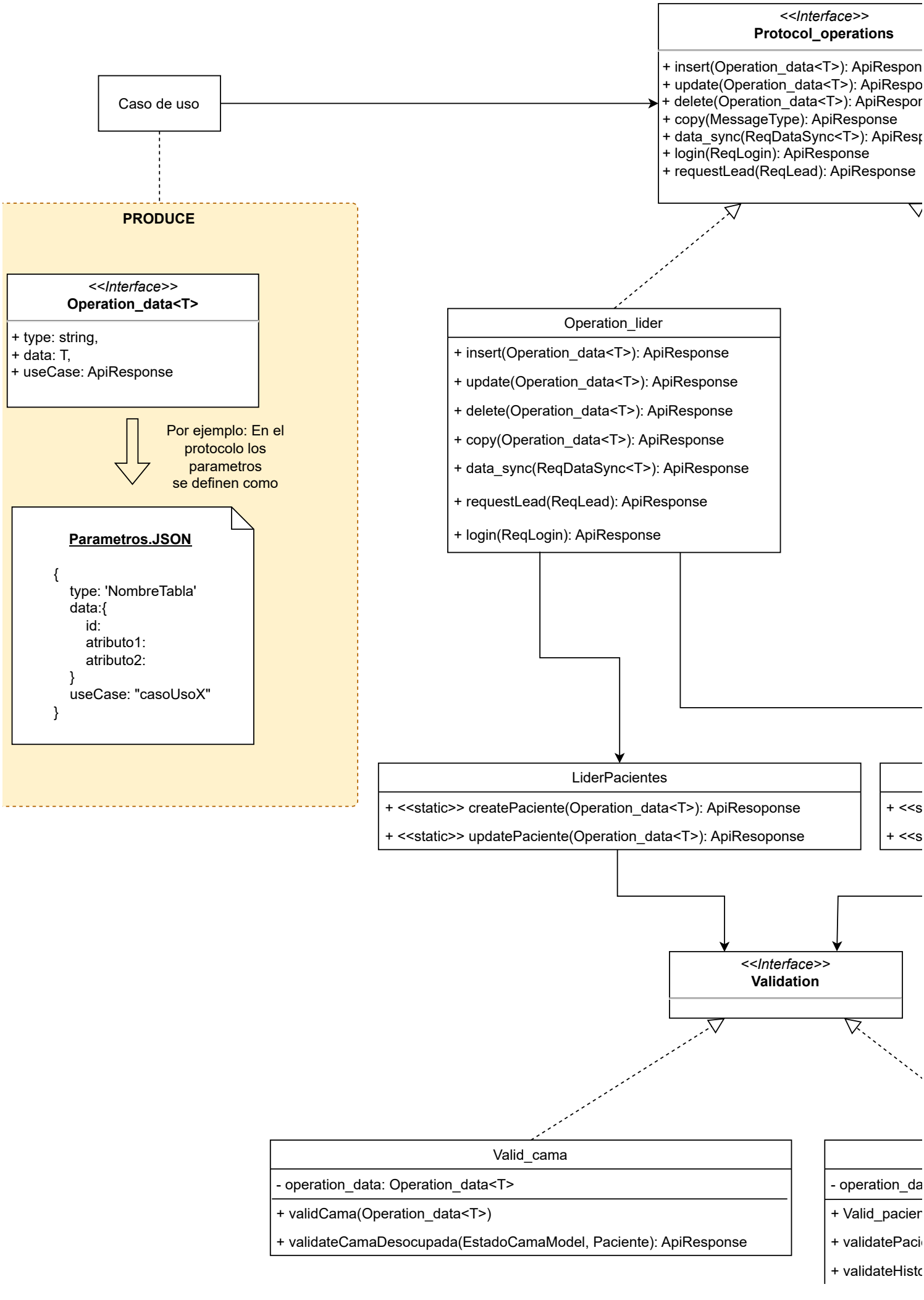
Copia primaria de dominio Hospitales

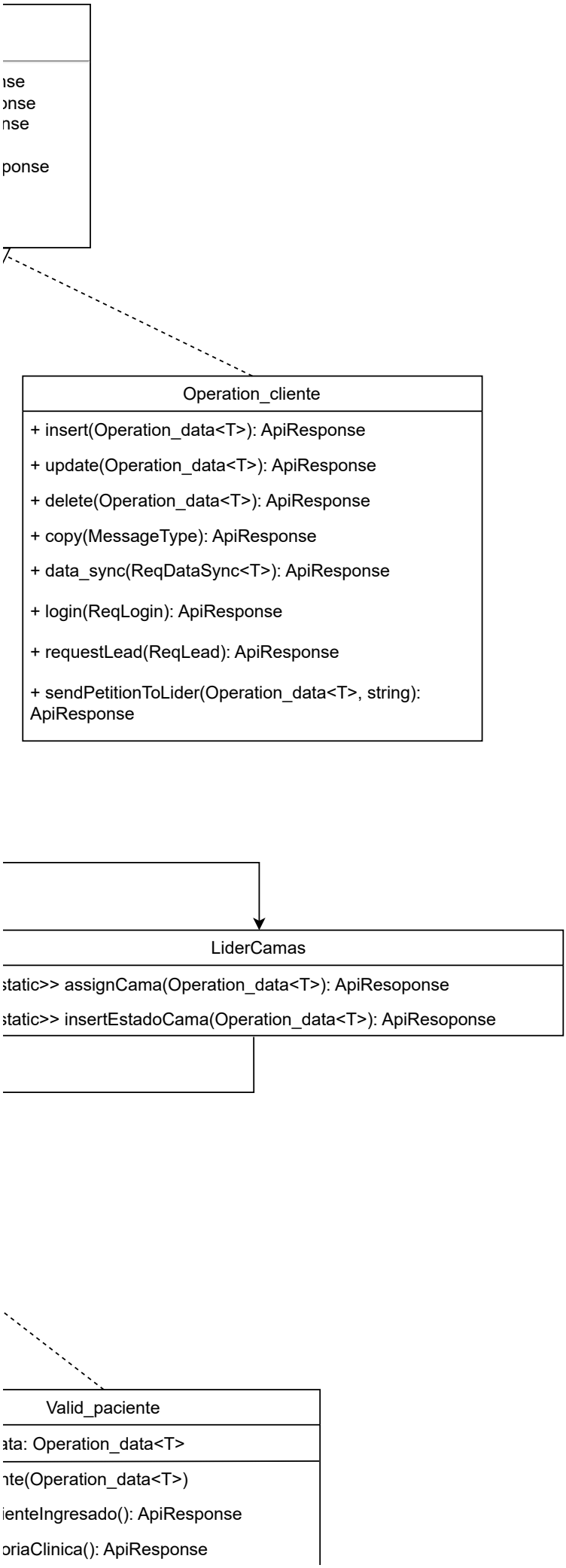
Copia primaria de dominio Usuarios

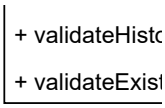
Dispositivo cliente

Dispositivo líder de sector

Servidor en la nube

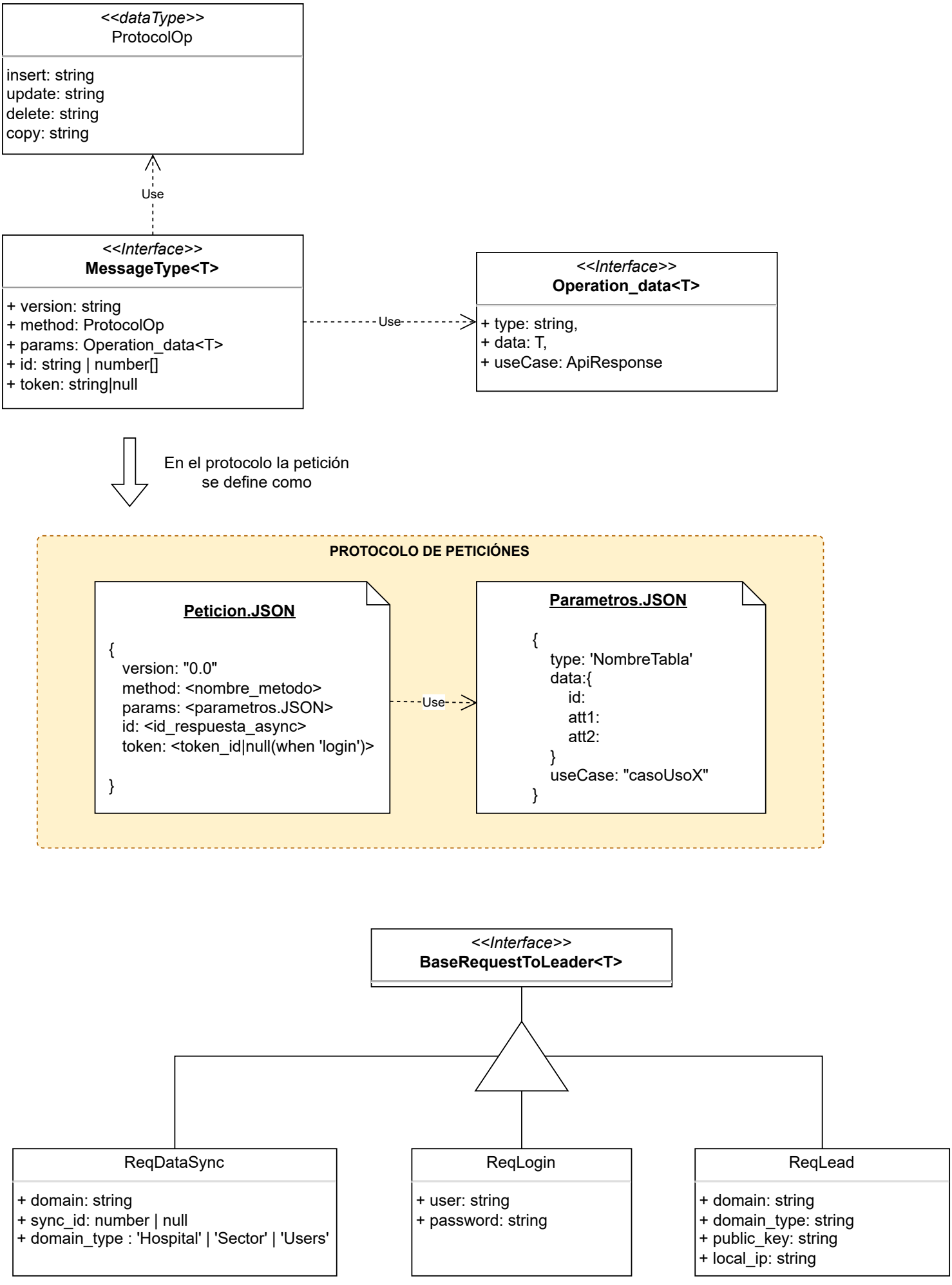

<<Interface>>
Protocol_operations
+ insert(Operation_data<T>): ApiRespon
+ update(Operation_data<T>): ApiRespo
+ delete(Operation_data<T>): ApiRespor
+ copy(MessageType): ApiResponse
+ data_sync(ReqDataSync<T>): ApiResp
+ login(ReqLogin): ApiResponse
+ requestLead(ReqLead): ApiResponse
Caso de uso
PRODUCE
<<Interface>>
Operation_data<T>
+ type: string,
+ data: T,
+ useCase: ApiResponse
Por ejemplo: En el protocolo los parametros se definen como
Parametros.JSON
{
type: 'NombreTabla'
data:{
id:
atributo1:
atributo2:
}
useCase: "casoUsoX"
}
Operation_lider
+ insert(Operation_data<T>): ApiResponse
+ update(Operation_data<T>): ApiResponse
+ delete(Operation_data<T>): ApiResponse
+ copy(Operation_data<T>): ApiResponse
+ data_sync(ReqDataSync<T>): ApiResponse
+ requestLead(ReqLead): ApiResponse
+ login(ReqLogin): ApiResponse
LiderPacientes
+ <<static>> createPaciente(Operation_data<T>): ApiResoponse
+ <<static>> updatePaciente(Operation_data<T>): ApiResoponse
<<Interface>>
Validation
Valid_cama
- operation_data: Operation_data<T>
+ validCama(Operation_data<T>)
+ validateCamaDesocupada(EstadoCamaModel, Paciente): ApiResponse

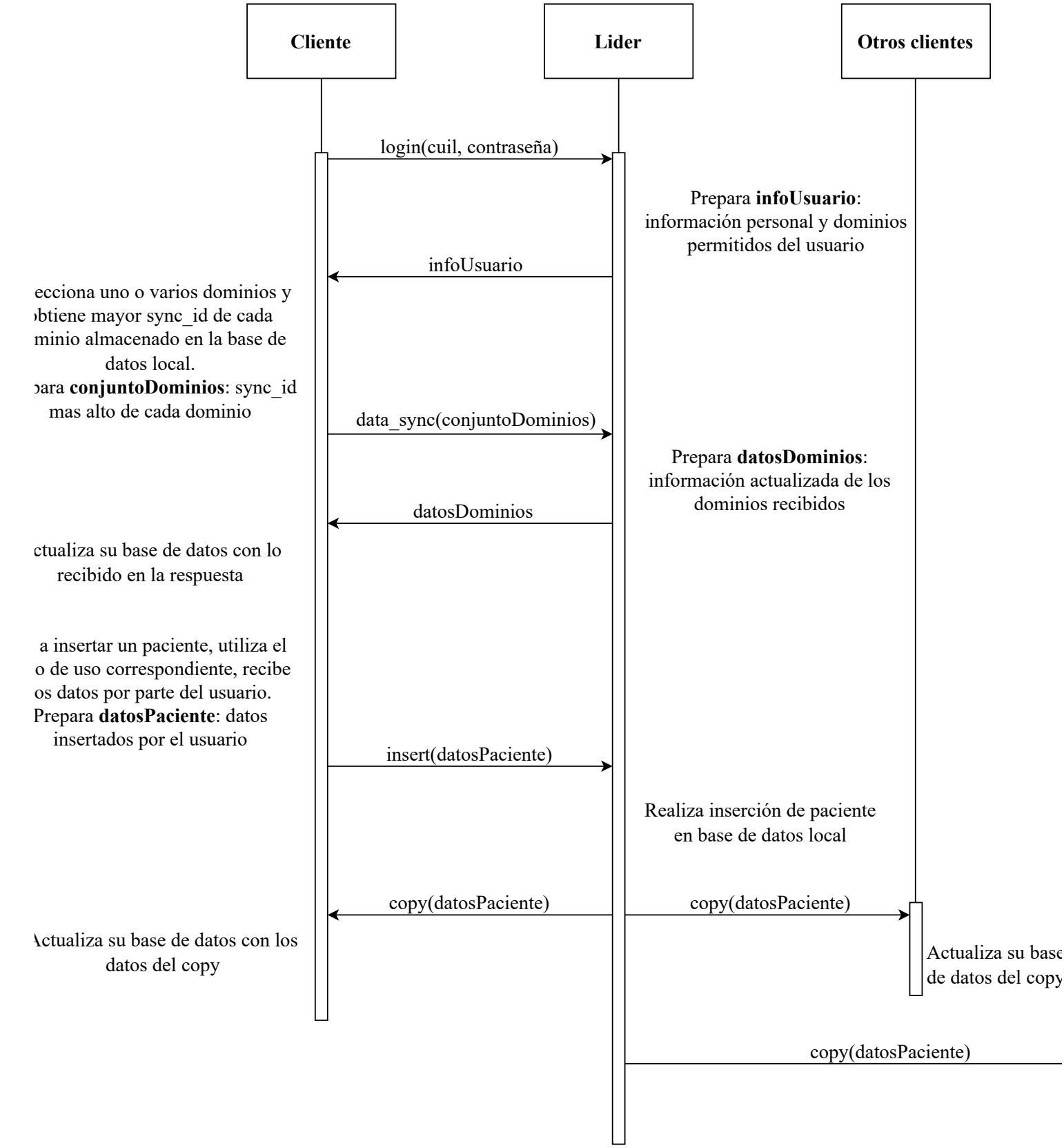

nse
onse
nse
ponse
Operation_cliente
+ insert(Operation_data<T>): ApiResponse
+ update(Operation_data<T>): ApiResponse
+ delete(Operation_data<T>): ApiResponse
+ copy(MessageType): ApiResponse
+ data_sync(ReqDataSync<T>): ApiResponse
+ login(ReqLogin): ApiResponse
+ requestLead(ReqLead): ApiResponse
+ sendPetitionToLider(Operation_data<T>, string): ApiResponse
LiderCamas
static>> assignCama(Operation_data<T>): ApiResoponse
static>> insertEstadoCama(Operation_data<T>): ApiResoponse
Valid_paciente
ata: Operation_data<T>
nte(Operation_data<T>)
ienteIngresado(): ApiResponse
oriaClinica(): ApiResponse

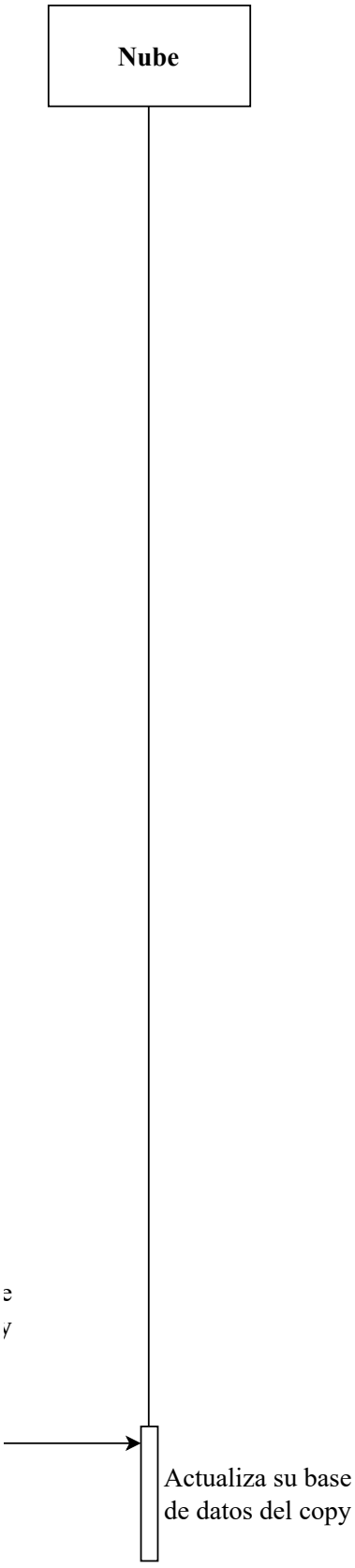

+ validateHisto
+ validateExist

oriaClinicaUpdate(): ApiResponse

tPaciente(): ApiResponse


<<dataType>>
ProtocolOp
insert: string
update: string
delete: string
copy: string
Use
<<Interface>>
MessageType<T>
+ version: string
+ method: ProtocolOp
+ params: Operation_data<T>
+ id: string | number[]
+ token: string|null
Use
<<Interface>>
Operation_data<T>
+ type: string,
+ data: T,
+ useCase: ApiResponse
En el protocolo la petición
se define como
PROTOCOLO DE PETICIÓNES
Peticion.JSON
{
version: "0.0"
method: <nombre_metodo>
params: <parametros.JSON>
id: <id_respuesta_async>
token: <token_id|null(when 'login')>
}
Use
Parametros.JSON
{
type: 'NombreTabla'
data:{
id:
att1:
att2:
}
useCase: "casoUsoX"
}
<<Interface>>
BaseRequestToLeader<T>
ReqDataSync
+ domain: string
+ sync_id: number | null
+ domain_type : 'Hospital' | 'Sector' | 'Users'
ReqLogin
+ user: string
+ password: string
ReqLead
+ domain: string
+ domain_type: string
+ public_key: string
+ local_ip: string


Cliente
Lider
Otros clientes
login(cuil, contraseña)
Prepara infoUsuario:
información personal y dominios
permitidos del usuario
infoUsuario
ecciona uno o varios dominios y
)btiene mayor sync_id de cada
minio almacenado en la base de
datos local.
)ara conjuntoDominios: sync_id
mas alto de cada dominio
data_sync(conjuntoDominios)
Prepara datosDominios:
información actualizada de los
dominios recibidos
datosDominios
ctualiza su base de datos con lo
recibido en la respuesta
a insertar un paciente, utiliza el
o de uso correspondiente, recibe
os datos por parte del usuario.
Prepara datosPaciente: datos
insertados por el usuario
insert(datosPaciente)
Realiza inserción de paciente
en base de datos local
copy(datosPaciente)
copy(datosPaciente)
Actualiza su base de datos con los
datos del copy
Actualiza su base
de datos del copy
copy(datosPaciente)


Nube
Actualiza su base
de datos del copy