

AUTEN

Con

NTICACION BIOMETRICA - MODELO C4

texto del sistema

Perso

Esta

Todo el perso
registro co



Personal diferente a equipo de seguridad del TDEA
[Person]

es la representación de estudiantes, profesores,
personal administrativos y terceros
que deseen ingresar al TDEA

Use

AUTENTICACION BIOMETRICA APP/WEB

nal que desee ingresar al Tecnológico de Antioquia
se usará en la versión APP del proyecto y registro

a debe
res que

Co

registrarse con
datos
proceso p
administrada p

Contexto del contenedores

Perso

Esta e

Use

AUTENTICACION BIOMETRI

no usuario en la versión APP del proyecto, y registro de datos biométricos, en este caso el rostro, para un posterior de autenticación en la versión WEB que se utilizará por el personal de seguridad de la institución en los puntos de accesos físicos.

Personal diferente a equipo de seguridad del TDEA
[Person]

Es la representación de estudiantes, profesores, personal administrativos y terceros que deseen ingresar al TDEA

car sus
erá
puntos

Use

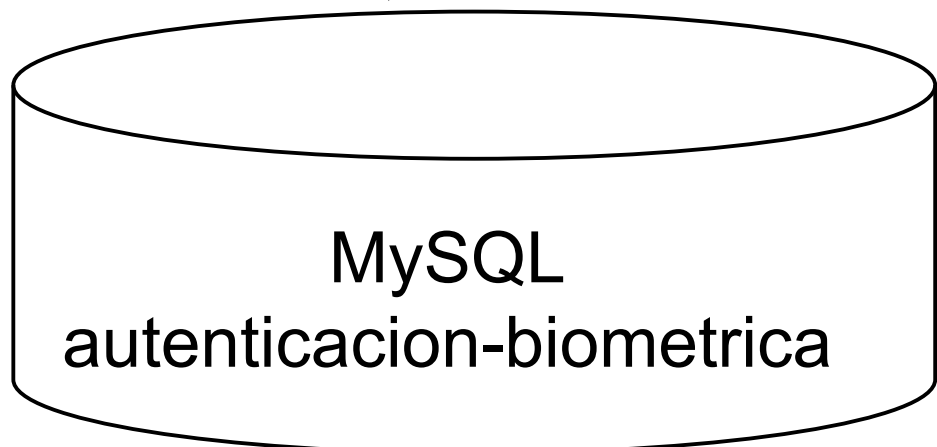
CACION BIOMETRICA

Sistema web
(Ionic. Angular)

Aplicación híbrida (WEB, ANDR

Consume HTTP

Leer/escribir



CA
OID)

AUTENTIC
(I
Aplicación h

piloto-autenticacion-biometrica (APIREST. SPRING BOOT. JAVA 11)

Esta API provee los servicios necesarios para llevar acabo el proceso de registro de datos biometricos y la autenticación biometrica. Además, diferentes proceso del modulo administrativo.

Consume HTTPS

autenticacion-biometrica-apigateway
(APIREST. API GATEWAY. AWS)

ACCION-BIOMETRICA

APP

onic. Angular)

íbrida (WEB, ANDROID)

Consume HTTP

Dia

cc

Diagrama de componentes

com.co.autenticacionbiometricaapi.rest.pilotoautent

administracion.usuario

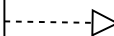
controller

UserController <<class>>

- UsuarioService usuarioService

+ UsuarioResponse crearUsuario(Us
+ UsuarioResponse editarUsuario(Us

UsuarioService <interface>



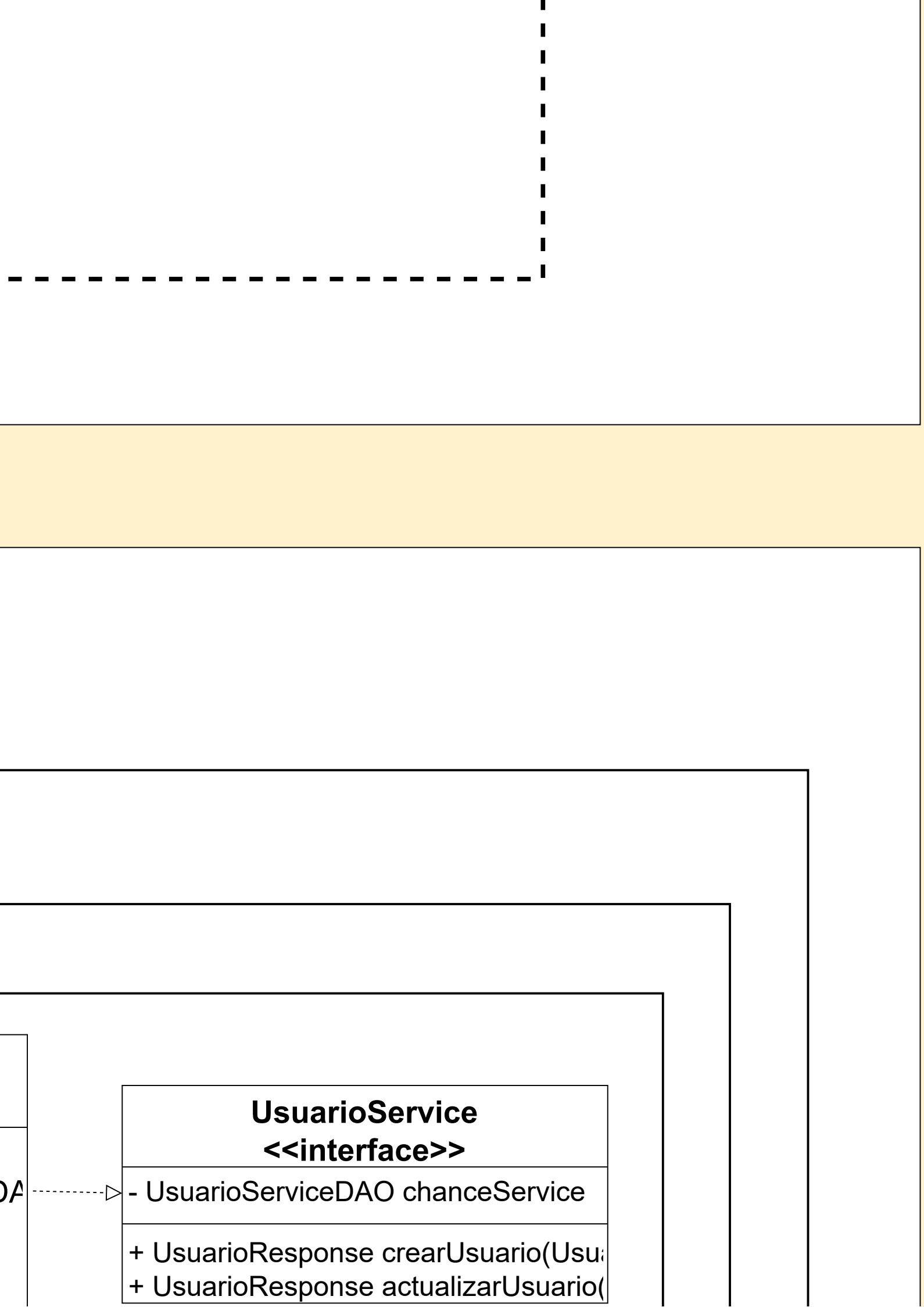
(UML RECOMMENDATIONS)

comunicacionbiometrica

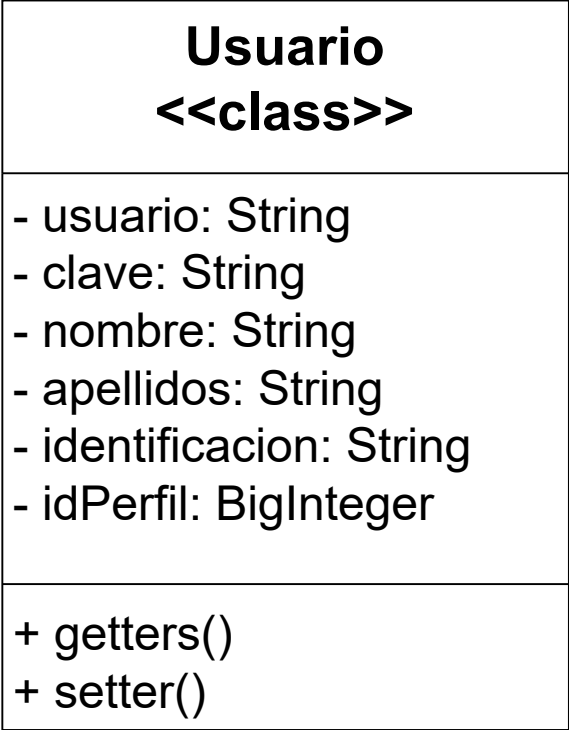
service

UsuarioServiceImpl
<<class>>

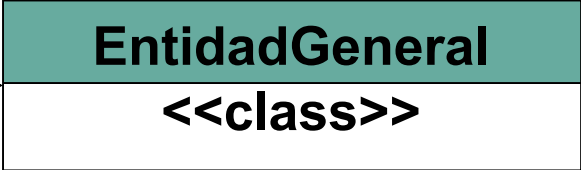
- UsuarioService usuarioService
- UsuarioServiceDAO usuarioServiceD

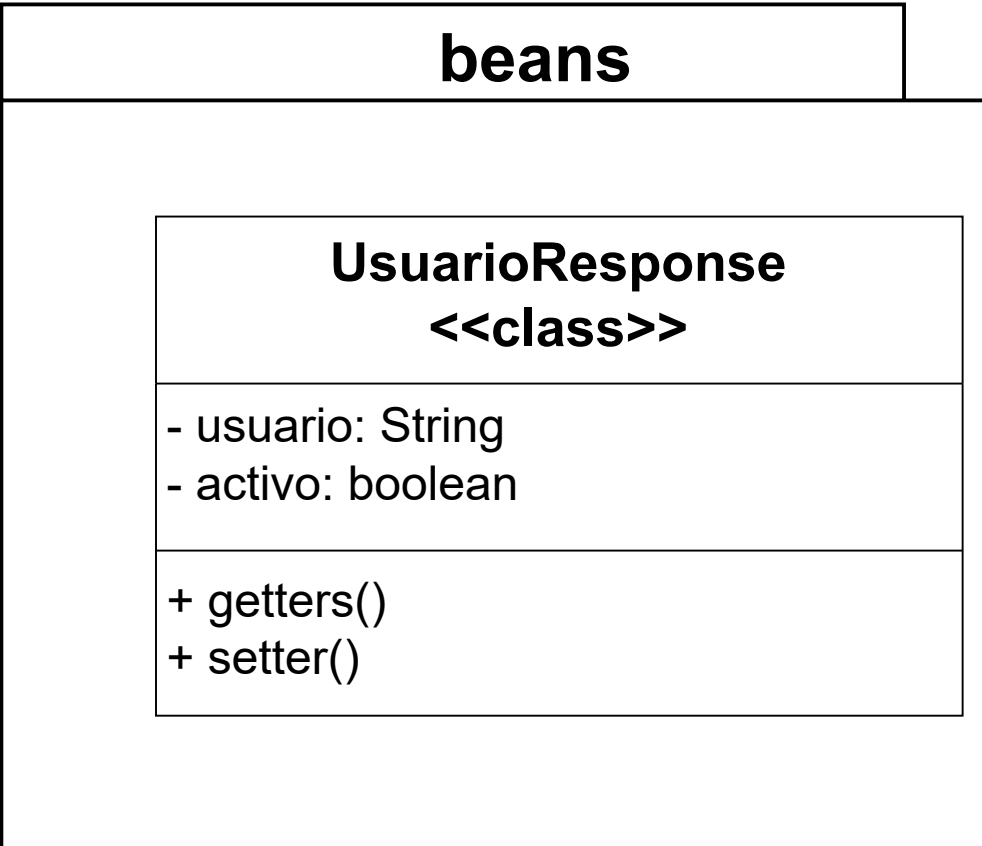
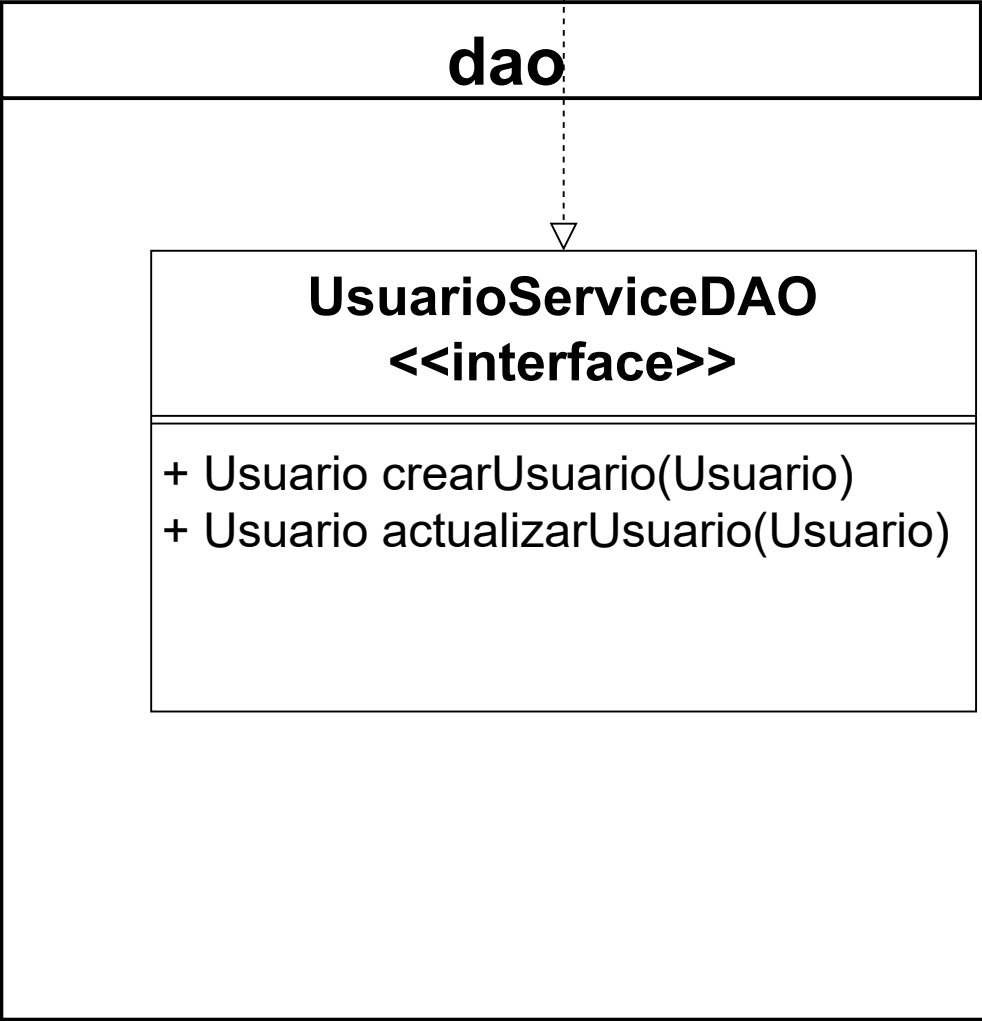


model



Extends





JpaRepository<Usuario>
<<interface>>



UsuarioRequest
<<class>>

- usuario: String
- clave: String
- nombre: String
- apellido: String
- identificación: String
- idPerfil: BigInteger

+ getters()

[illegible]

autenticacion.registro

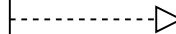
controller

RegistroController <<class>>

- RegistroService registroService

+ UserInfoBiometricResponse
 registrarDatosBiometricos (UserInfoE

RegistroService <<interface>>



model

UserInfoBiometric <<class>>

- idUsuario: String
- faceld: String
- externalId: String
- rutaBucketS3: String
- nombreArchivo: String

EntidadGenerica <<class>>



service

RegistroServiceImpl <<class>>

- RegistroService registroService
- RegistroServiceDAO registroServiceDAO

dao

RegistroServiceDAO <<interface>>

- + UserInfoBiometric almacenarDatosBior
- + UserInfoBiometric actualizarDatosBior

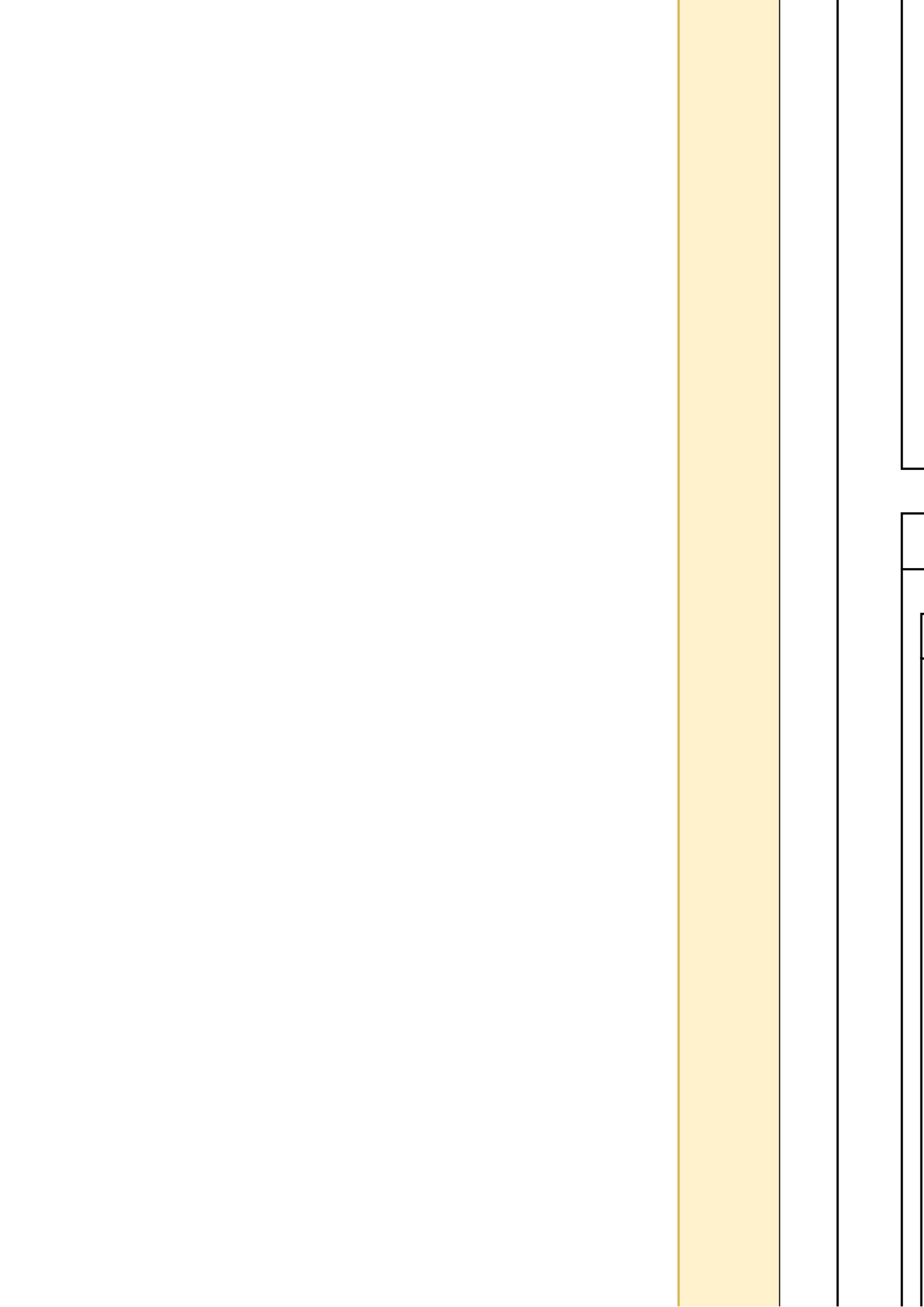
+ setter()

RegistroService
<<interface>>

- ChanceService chanceService

+ UsuarioResponse crearUsuario(UsuarioR
+ UsuarioResponse actualizarUsuario(Usua

JpaRepository<UserInfoBiometric>
<<interface>>

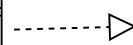


+ getters() + setter()

autenticacion.autenticacion

controller

AutenticacionController <<class>>
+AutenticacionBiometriaRequestResp autenticacionBiometrica (Autenticacion



AutenticacionServ <<interface>>

beans

AutenticacionBiometriaRequest
<<class>>

- idUsuario: String
- fotografia: String

- + getters()
- + setter()

service

AutenticacionServiceImpl
<<class>>

- RegistroService registroService

ice

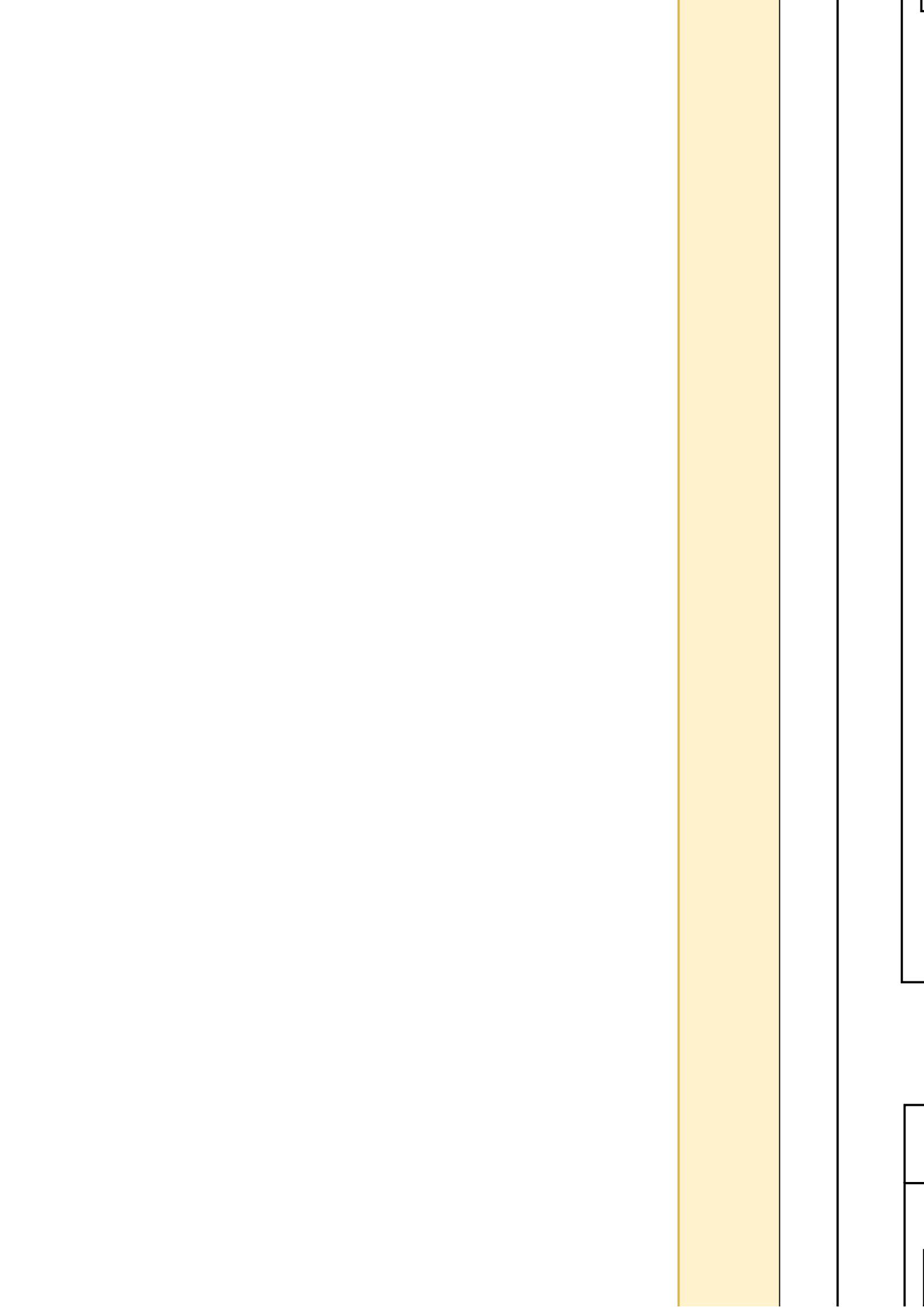
AutenticacionBiometriaRequestResponse **<<class>>**

- rostroId: String
- externalId: String
- fotografia: String

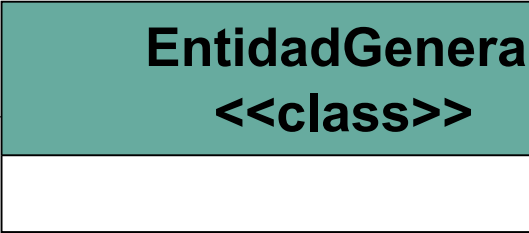
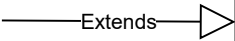
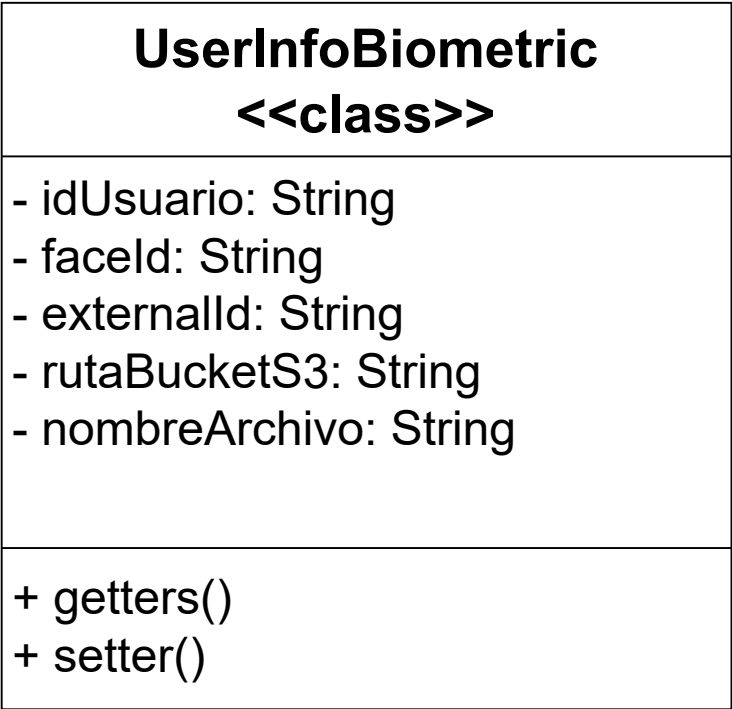
+ getters()
+ setter()

AutenticacionService **<<interface>>**

+ AutenticacionBiometriaRequestResponse
autenticacionBiometrica (AutenticacionBiometricaF



model



beans

AutenticacionBiometriaRequest <<class>>
- idUsuario: String - fotografia: String
+ getters() + setter()

AutenticacionBiometriaRequest
- rol: String - email: String - fotografia: String
+ getters() + setters()

AutenticacionBiometriaRequestResponse

<<class>>

hostId: String

externalId: String

fotoграфия: String

getters()

setter()

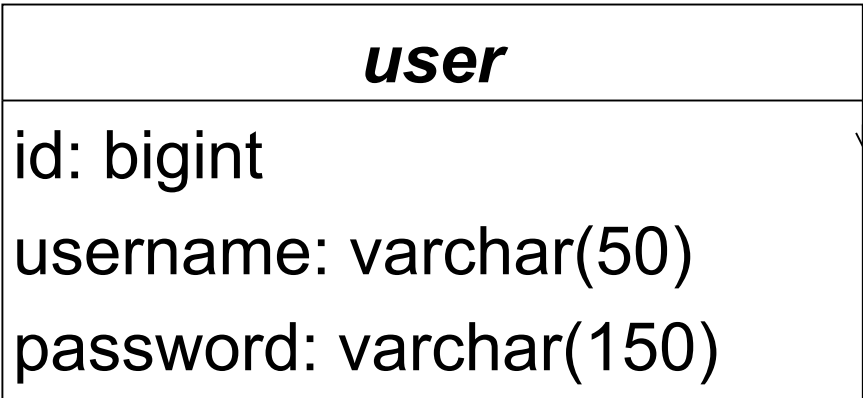
Diagn

S3Almacenamiento
<<class>

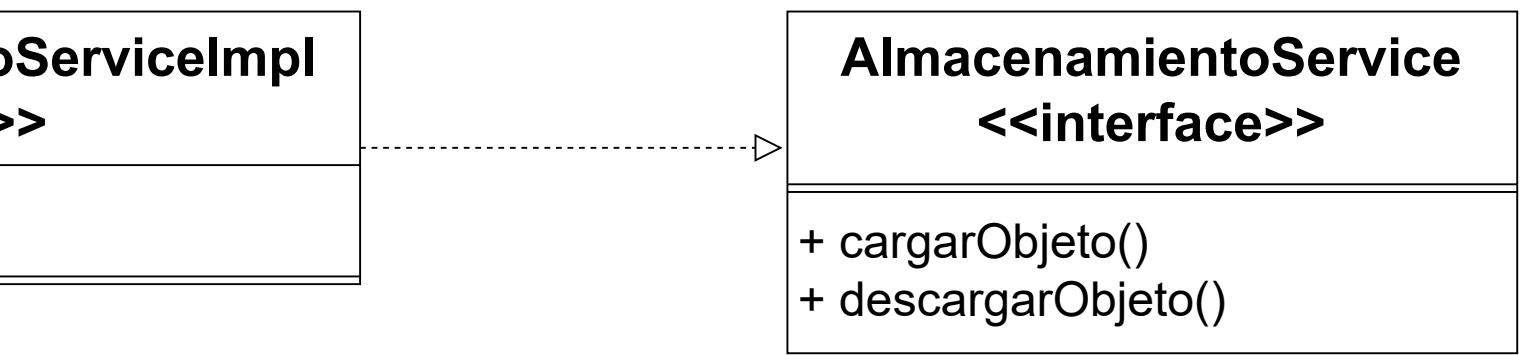
-

Diagrama de clases

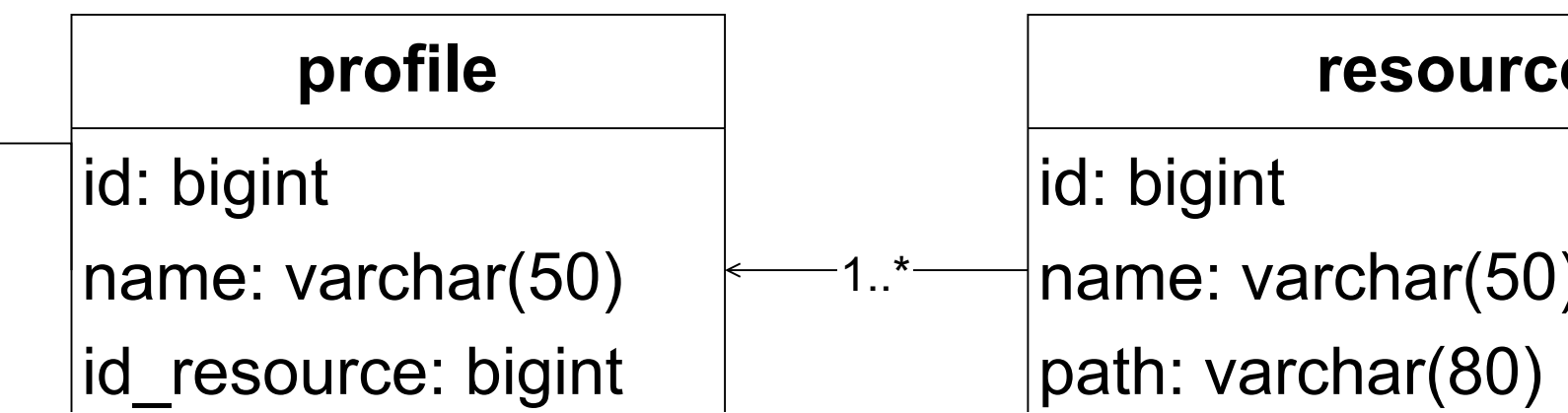
Diagrama relacional. Modulo ADMIN



1..1



ADMINISTRACION





name: varchar(50) lastname: varchar(100) identification: varchar(20) id_profile: bigint

Diagrama de relacional. Modulo RE

REGISTRO/AUTENTICACION

user

1..1

user_info_biometric

id: bigint

id_user: bigint

face_id: varchar(120)

external_id: varchar(50)

bucket_name: varchar(100)

file_name: varchar(100)

