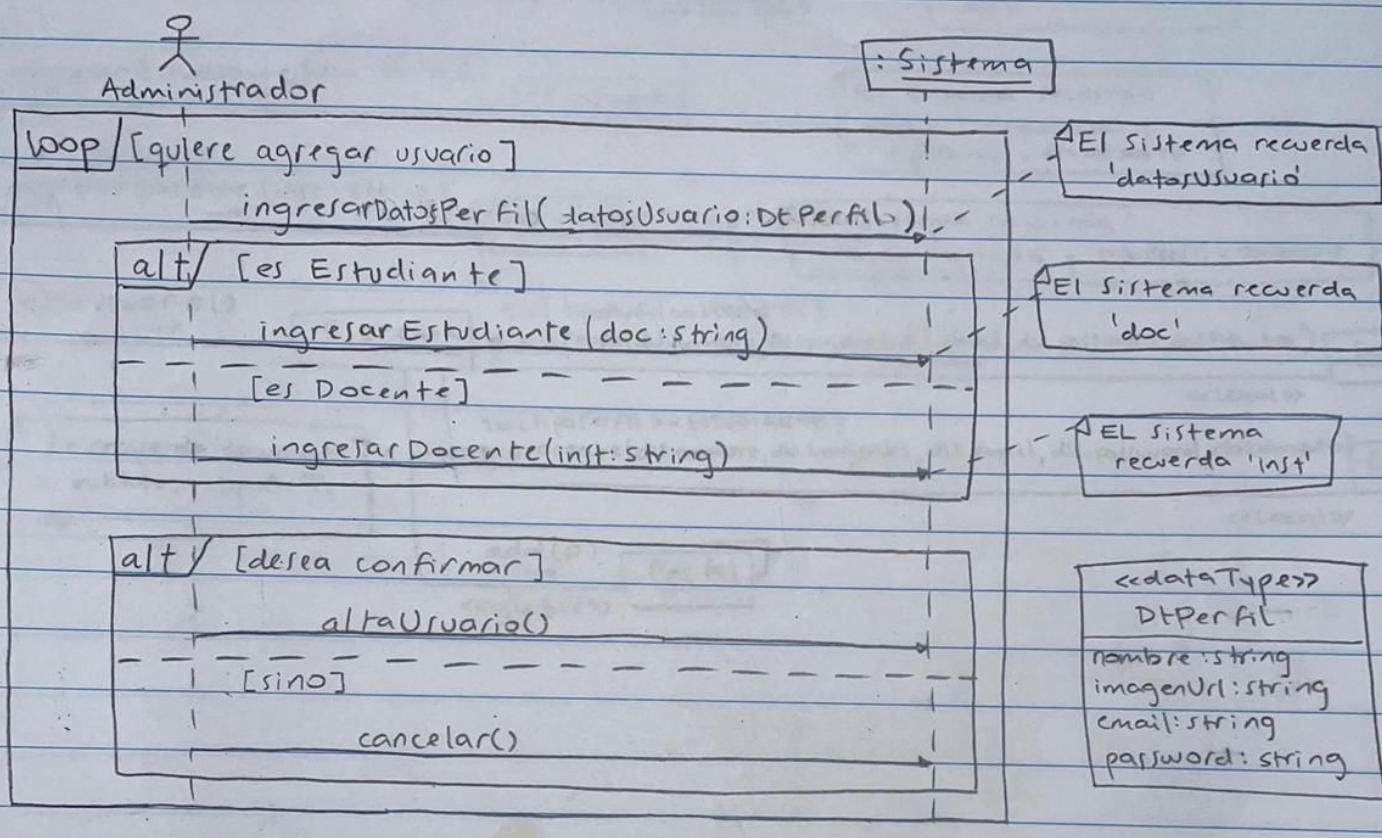
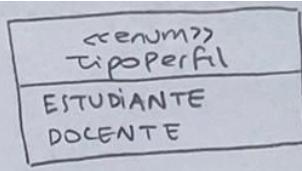


①

## Alta Usuario / (con memoria)



1



ingresarDatosPerfil (datosUsuario: DtUsuario)



:Sistema

El Sistema recuerda  
a datosUsuario en  
dU

ingresarEstudiante (doc: string)



:Sistema

El Sistema recuerda  
doc en 'documento' y  
tipoperfil = ESTUDIANTE

ingresarDocente (inst: string)



:Sistema

El Sistema recuerda  
inst en 'instituto' y  
tipoperfil = DOCENTE

altaUsuario()



:Sistema

1a [tipoperfil == DOCENTE]

p = create(dU.nombre, dU.imageUrl, dU.email, dU.password, instituto)

: Docente

<<local>>

1b [tipoperfil == ESTUDIANTE]

p = create(dU.nombre, dU.imageUrl, dU.email, dU.password, documento)

: Estudiante

<<local>>

add(p)

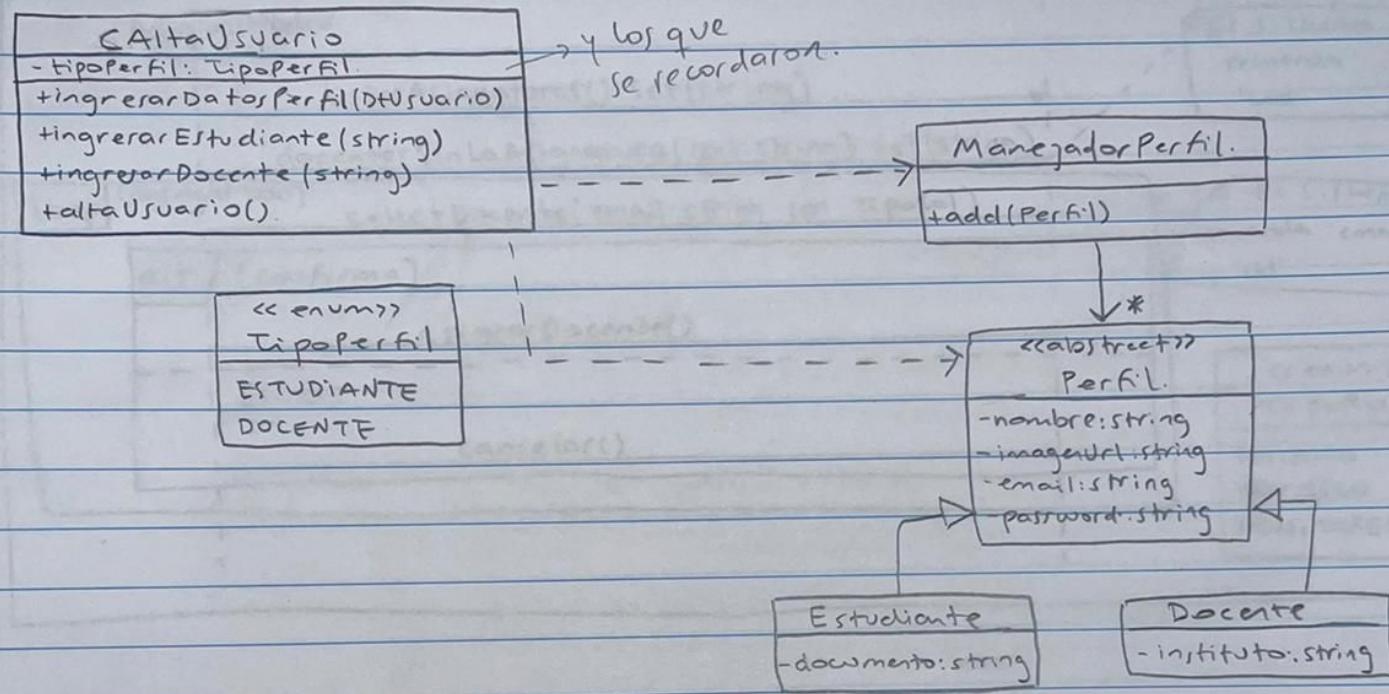
: Perfil

<<global>>

Se recuerda documento  
o instituto, y tipoperfil  
dU

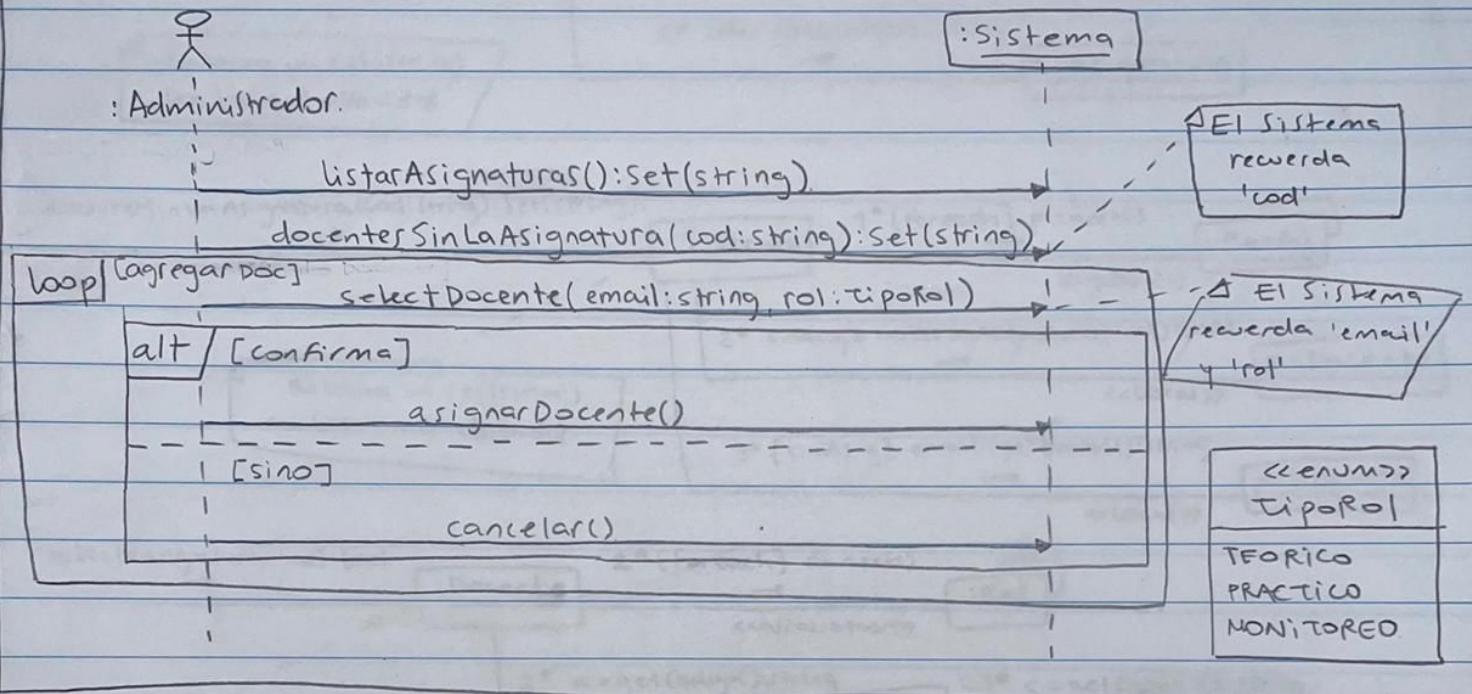
①

Definimos que `CAltaUsuario` será el que implementa las operaciones del Sistema para que el CU Alta Usuario (no el Sistema), entonces tendremos el siguiente DCD.

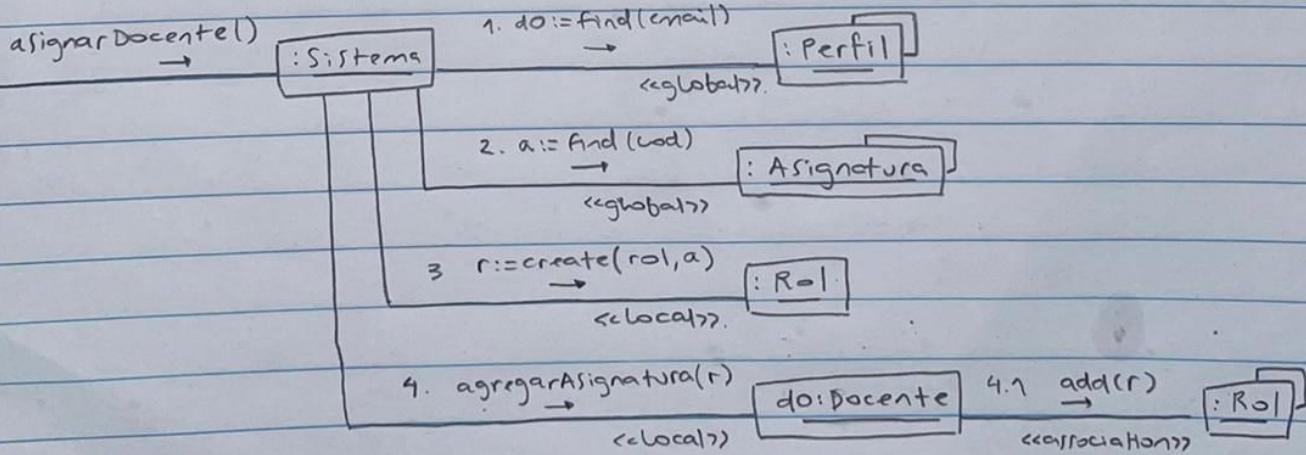
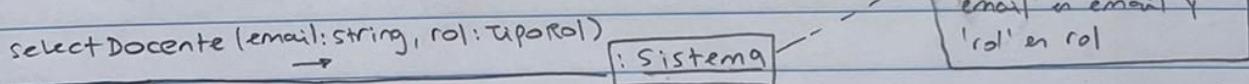
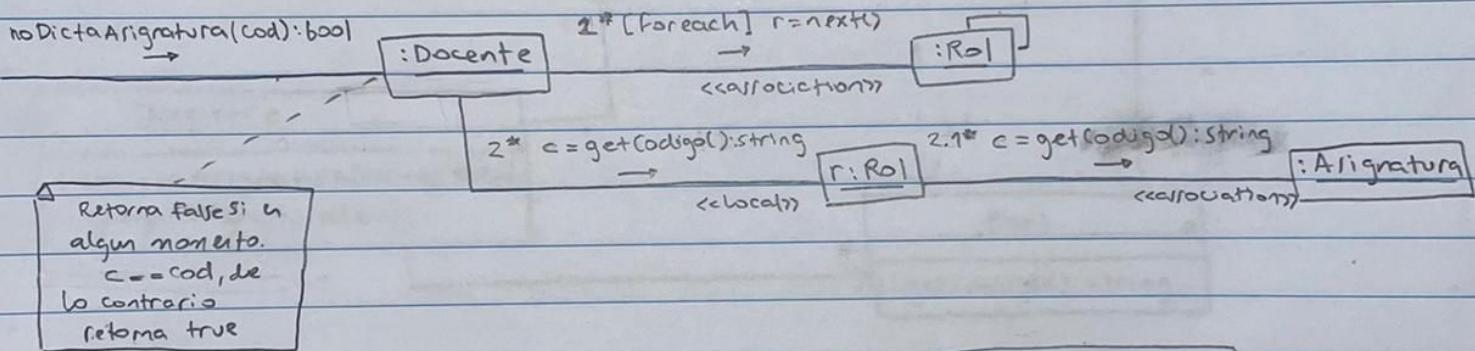
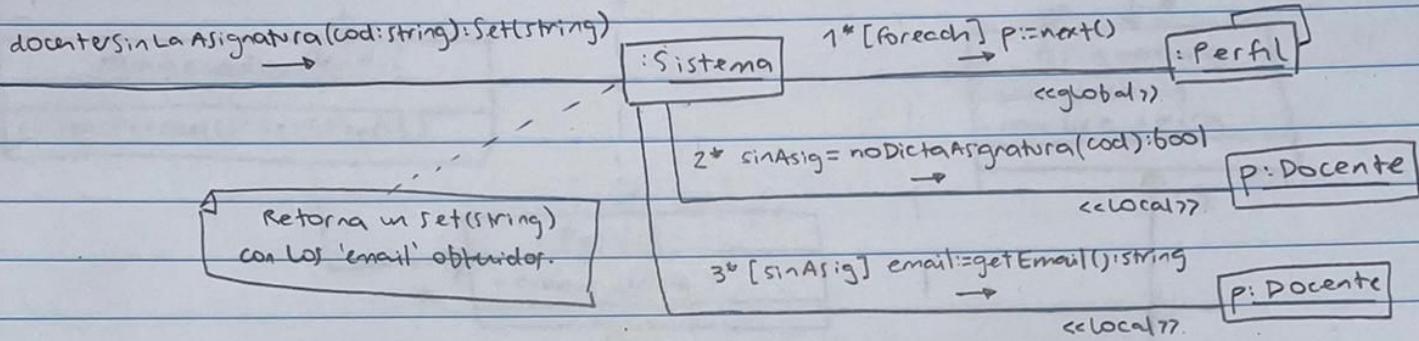
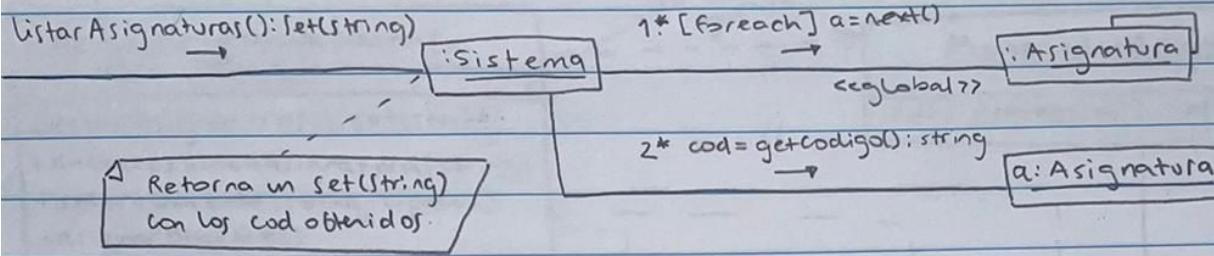


(2)

- Asignación de docentes a una asignatura (con memoria)

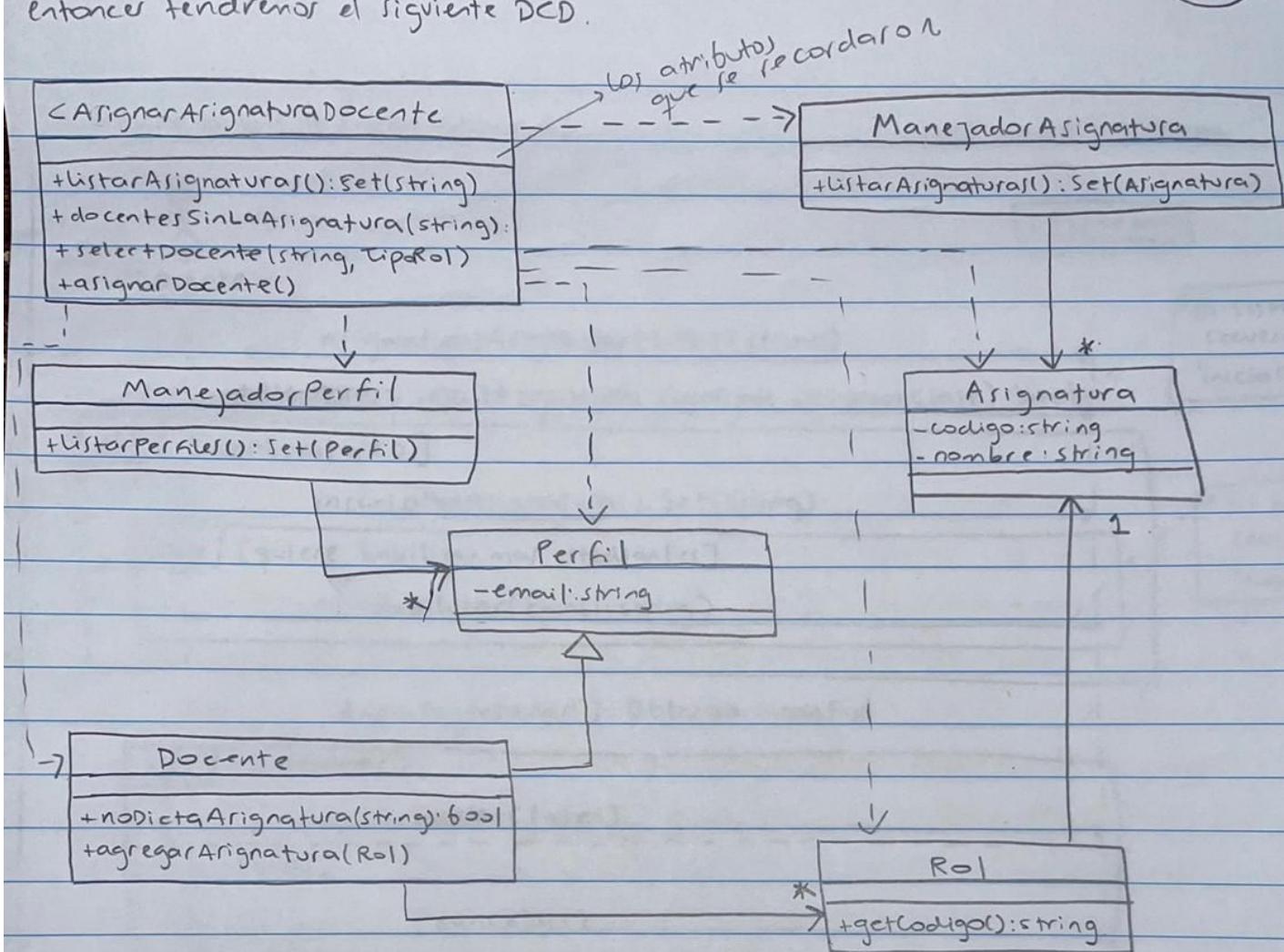


(2)



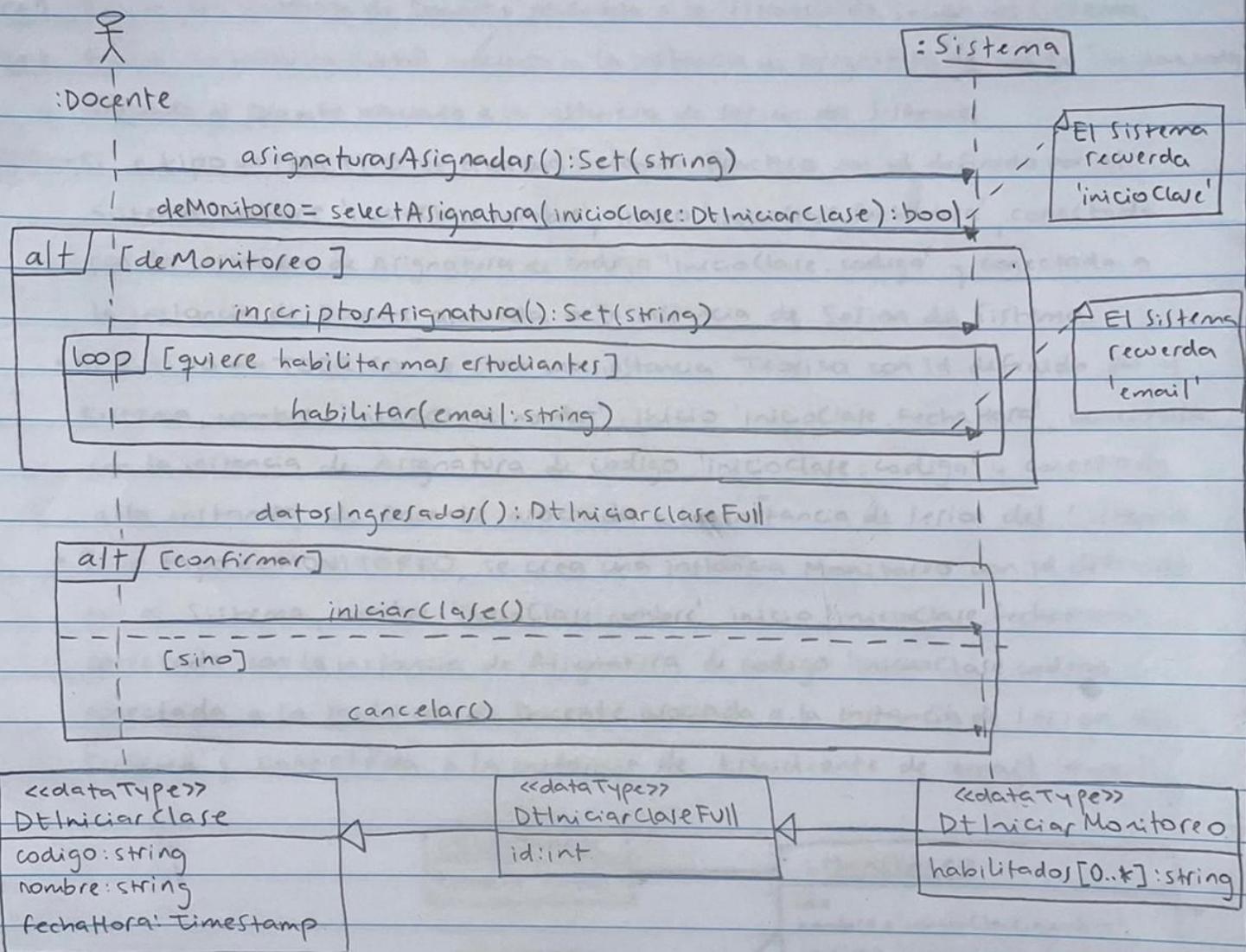
Definimos que CAssignarAsignaturaDocente será el que implementa las operaciones del Sistema para que el CU Asignación de docentes a una asignatura, entonces tendremos el siguiente DCD.

(2)



(3)

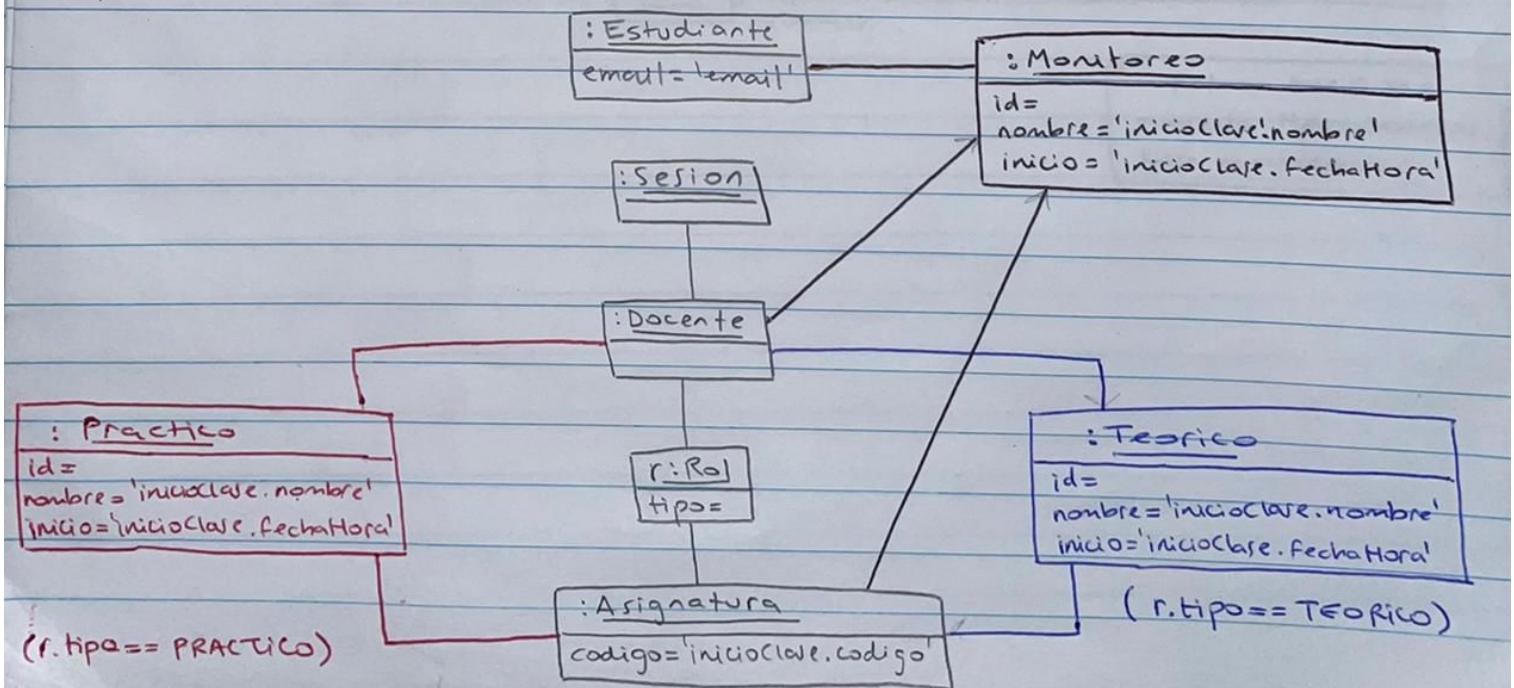
## • Inicio de Clase / (con memoria)



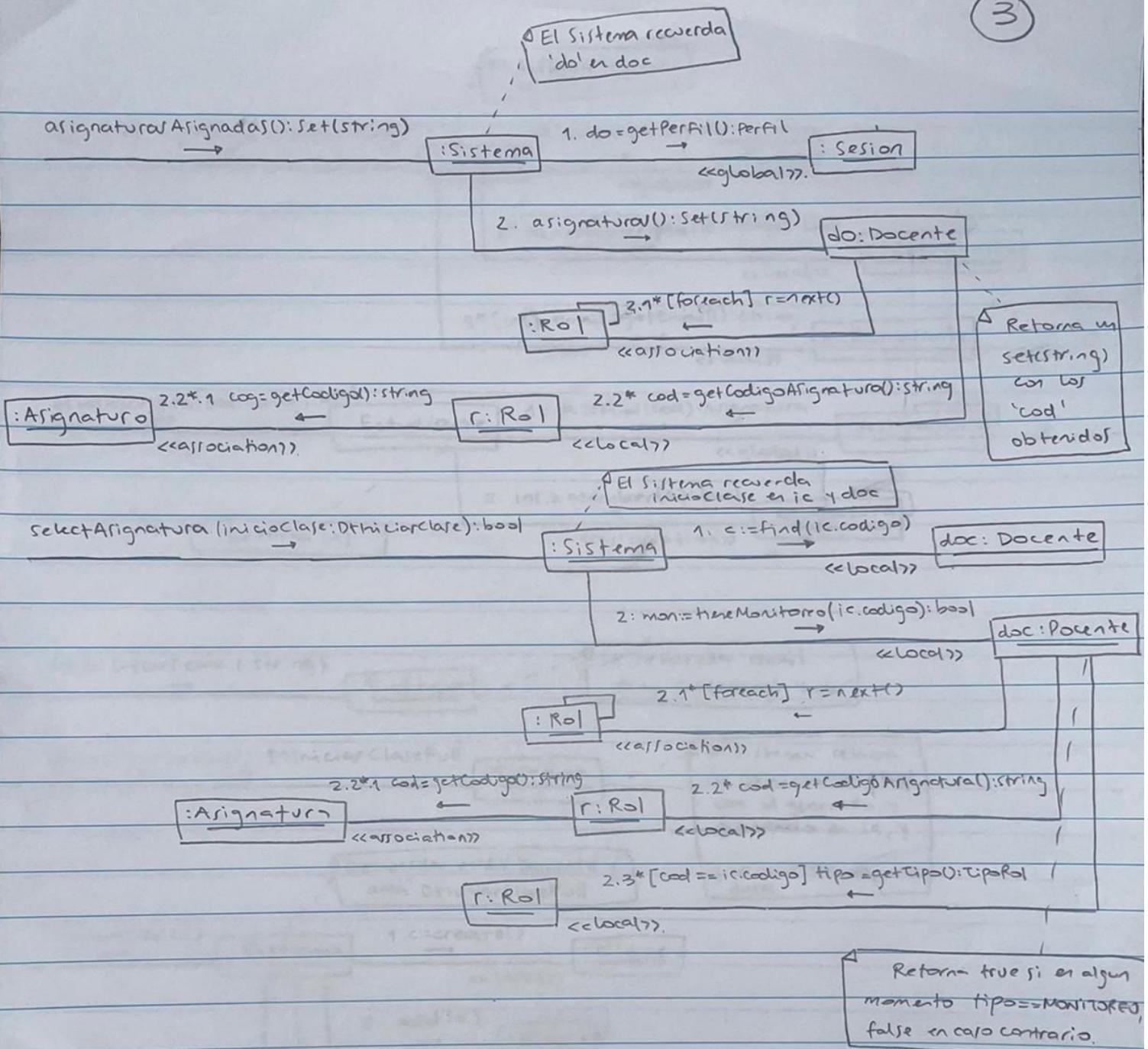
(3)

## iniciarClase() [CONTRATO]

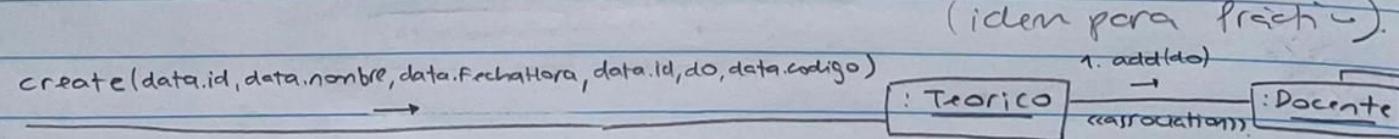
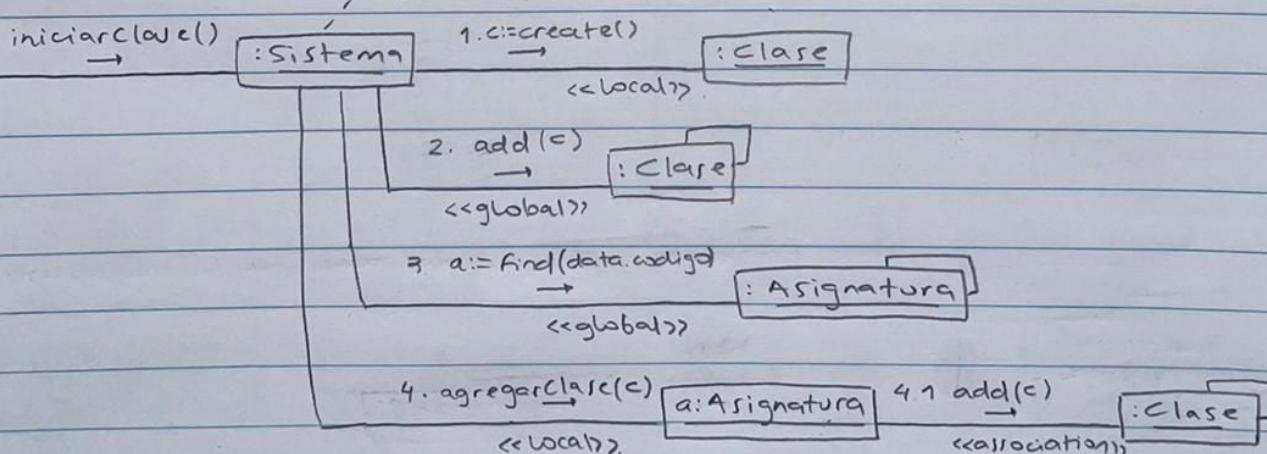
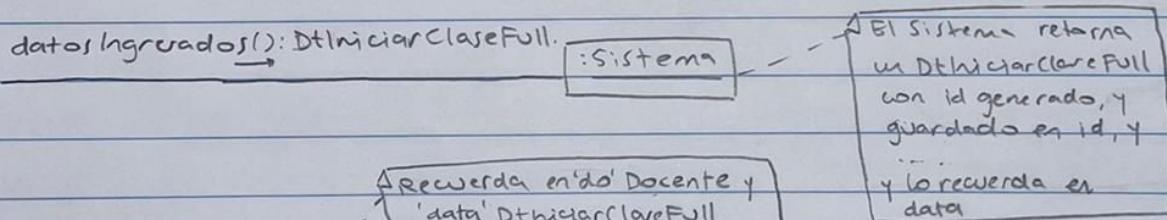
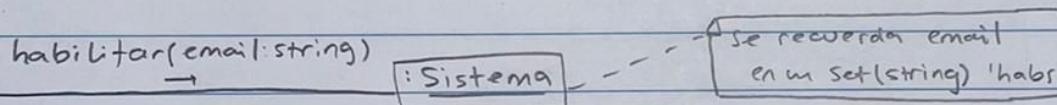
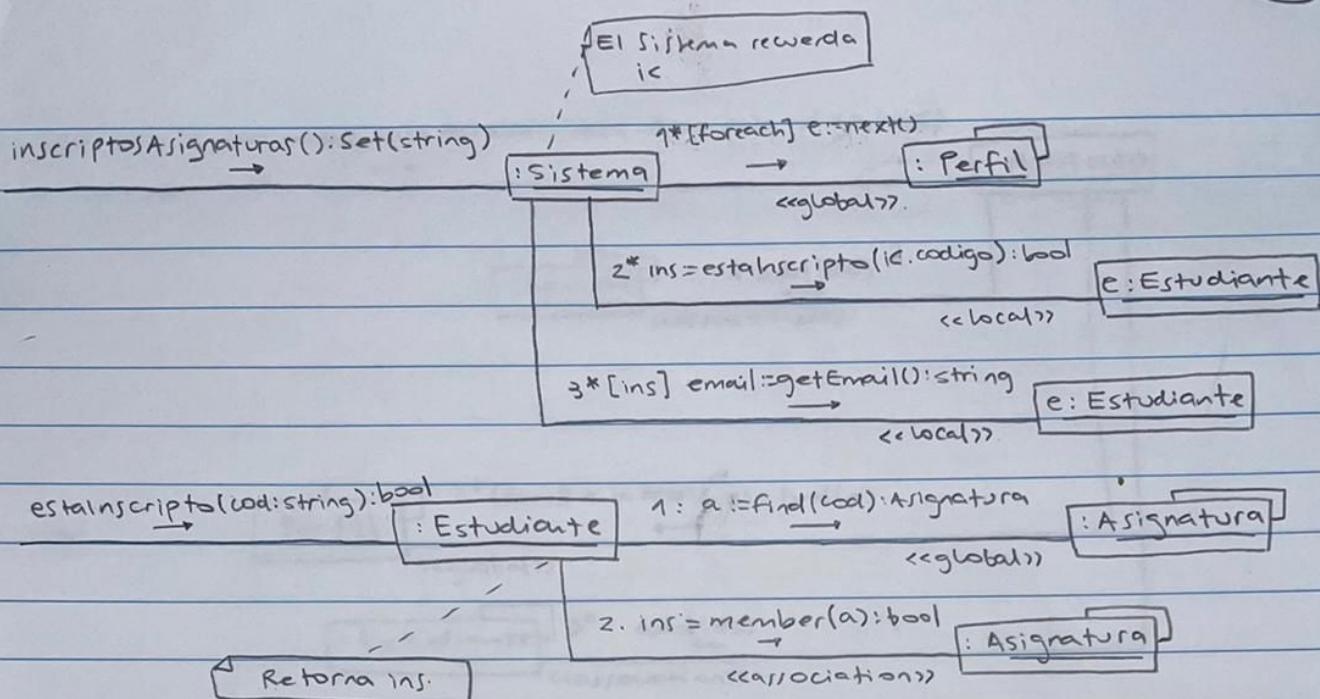
- Pre 1: Existe una instancia de Docente asociada a la instancia de Sesión del Sistema.
- Pre 2: Existe una instancia r de Rol asociada a la instancia de Asignatura de código 'inicioClase.codigo' y asociada al Docente asociado a la instancia de Sesión del Sistema.
- Post:
- Si  $r.tipo == PRACTICO$  se crea una instancia Practico con id definido por el Sistema, nombre 'inicioClase.nombre', inicio 'inicioClase.fechaHora', conectada con la instancia de Asignatura de código 'inicioClase.codigo' y conectada a la instancia de Docente asociada a la instancia de Sesión del Sistema.
  - Si  $r.tipo == TEORICO$ , se crea una instancia Teorico con id definido por el Sistema, nombre 'inicioClase.nombre', inicio 'inicioClase.fechaHora', conectada con la instancia de Asignatura de código 'inicioClase.codigo' y conectada a la instancia de Docente asociada a la instancia de Sesión del Sistema.
  - Si  $r.tipo == MONITOREO$ , se crea una instancia Monitoreo con id definido por el Sistema, nombre 'inicioClase.nombre', inicio 'inicioClase.fechaHora', conectada con la instancia de Asignatura de código 'inicioClase.codigo' y conectada a la instancia de Docente asociada a la instancia de Sesión del Sistema y conectada a la instancia de Estudiante de email 'email'.



(3)

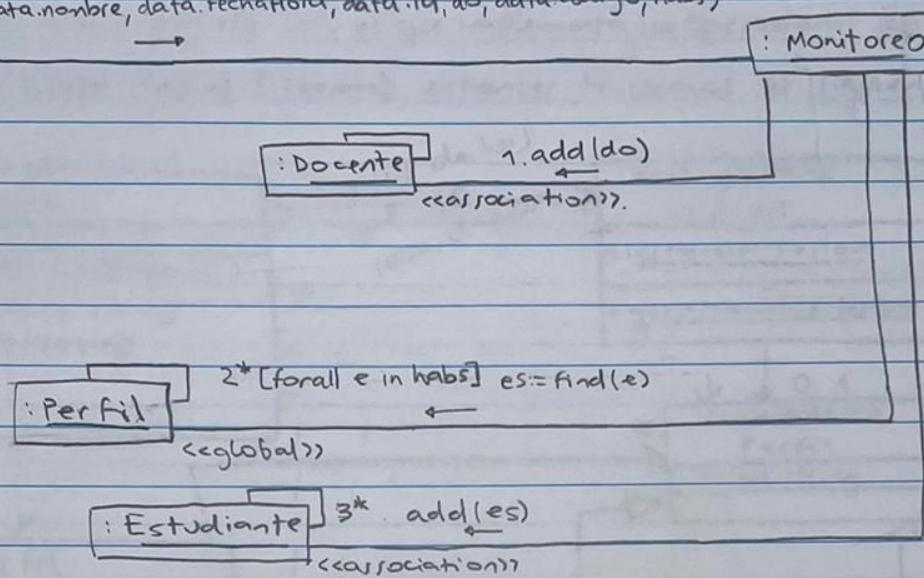


(3)



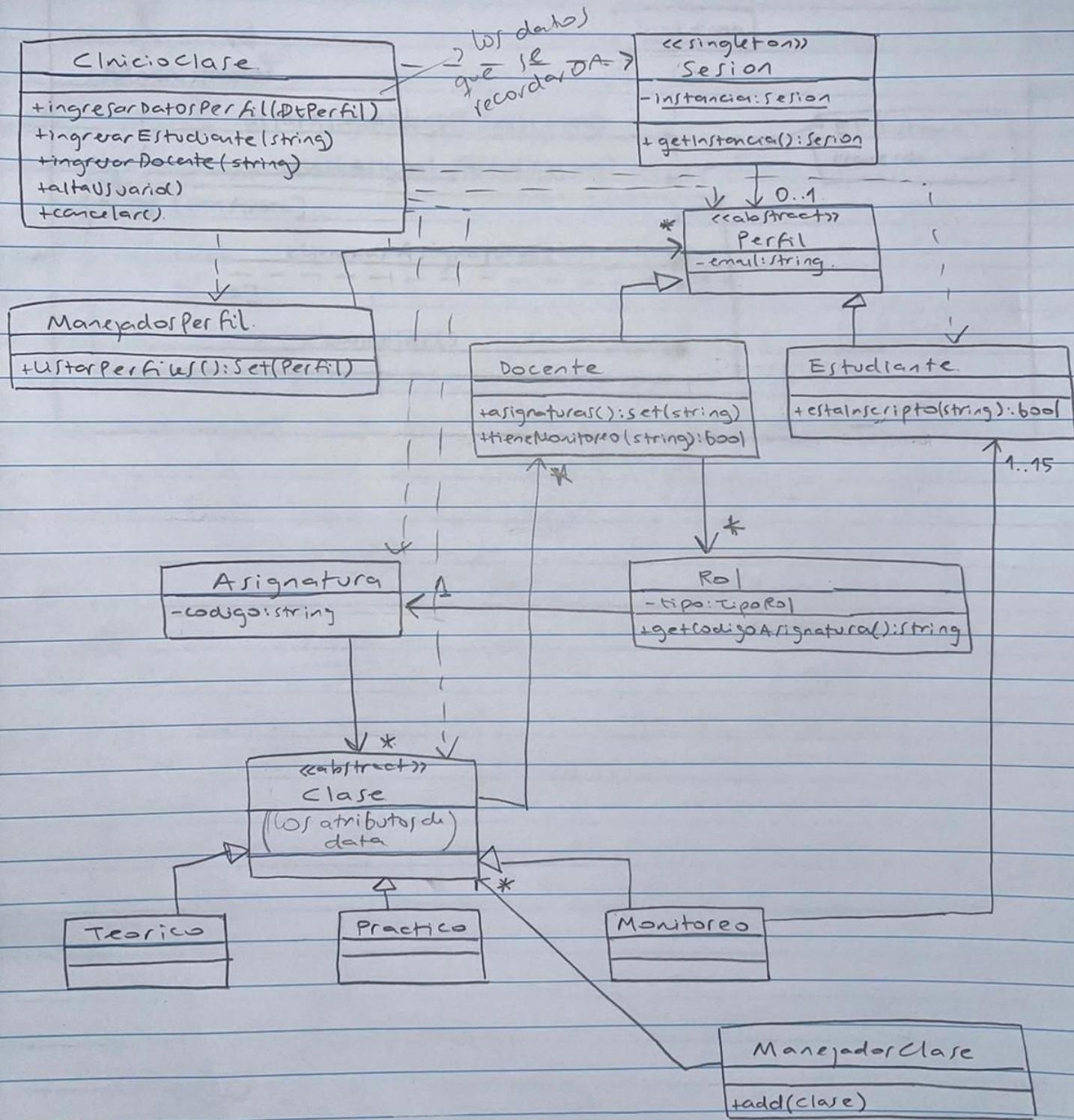
(3)

```
create(data.id, data.nombre, data.fechatoma, data.id, do, data.codigo, hab)
```



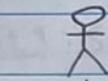
(3)

Definimos que `CInicioClase` será el que implementa las operaciones del sistema para que el CV Inicio de Clase (no el sistema), entonces tendremos el siguiente PCD.



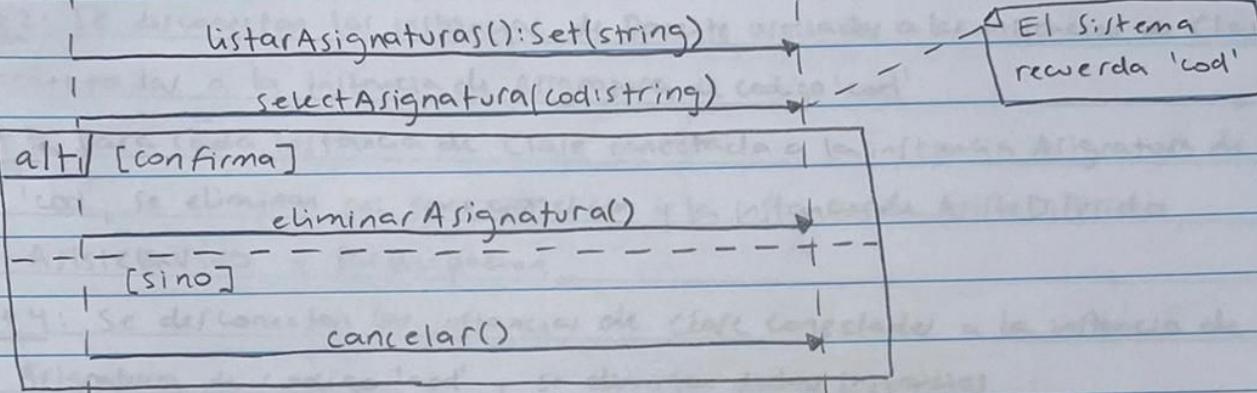
(4)

## • Eliminación de la asignatura (con memoria)



: Administrador

: Sistema



(4)

## eliminar Asignatura() [CONTRATO]

Pre1: Existe en el Sistema una instancia de Asignatura de código 'cod'

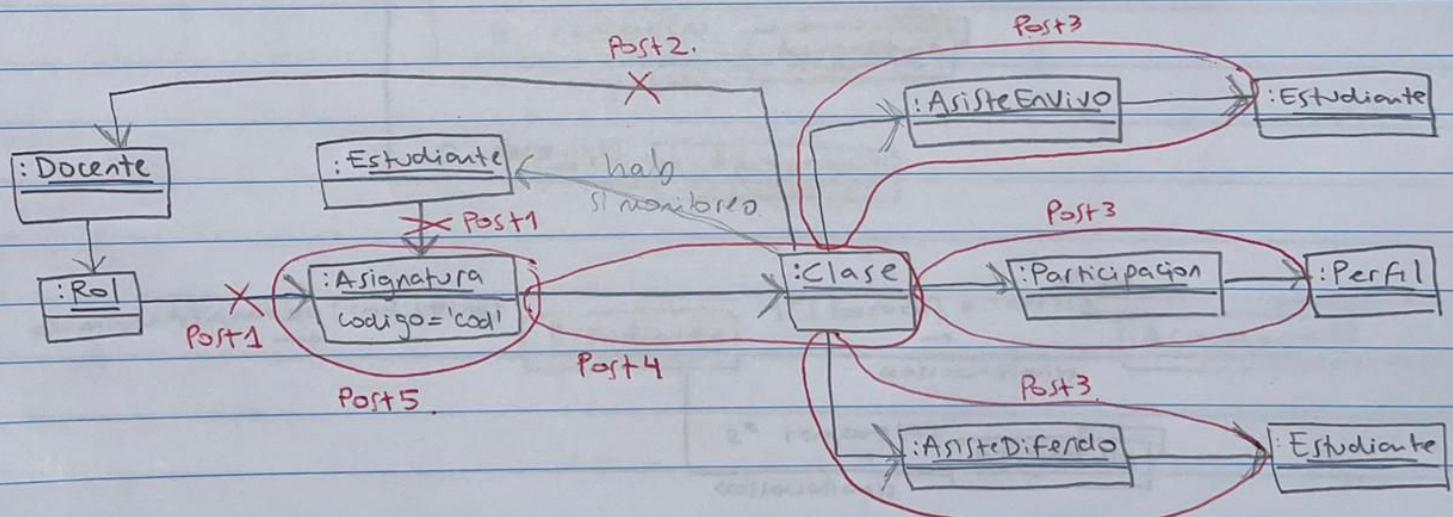
Post1: Se desconectan las instancias de Estudiante y Rol conectadas a la instancia de Asignatura de código 'cod'

Post2: Se desconectan las instancias de Docente asociadas a las instancias de Clase conectadas a la instancia de Asignatura de código 'cod'

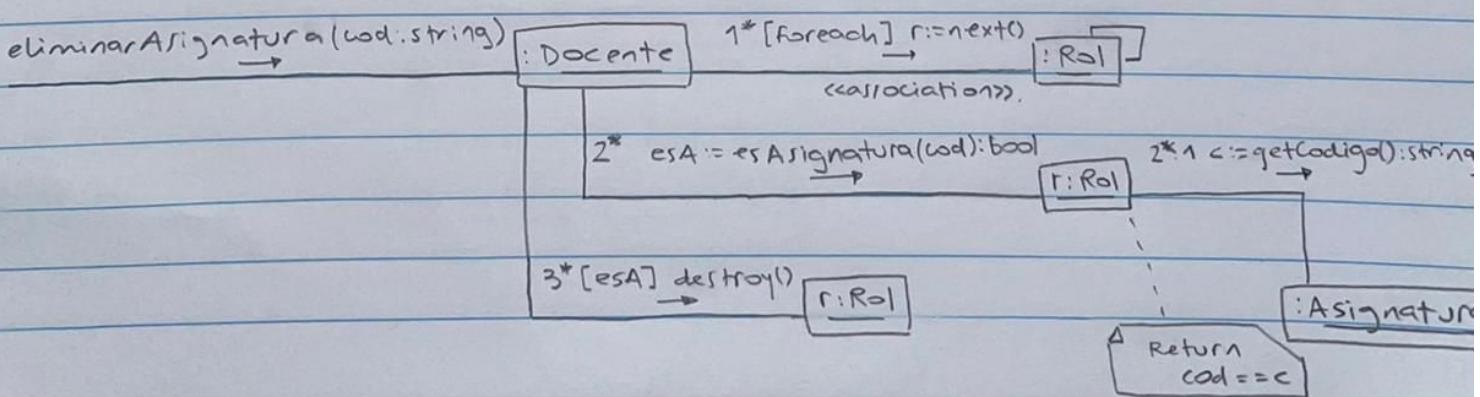
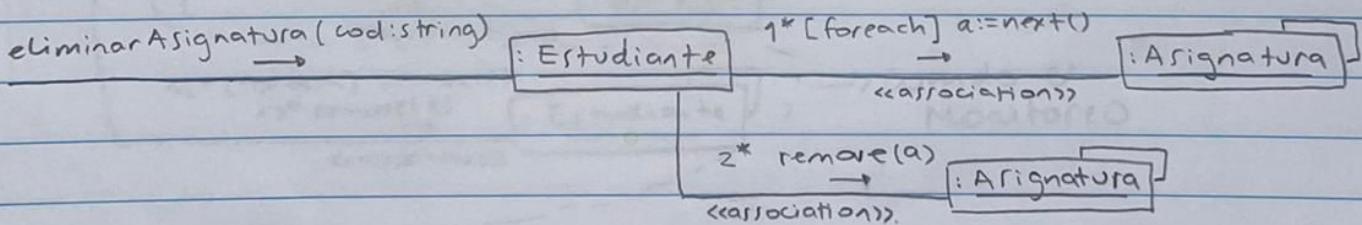
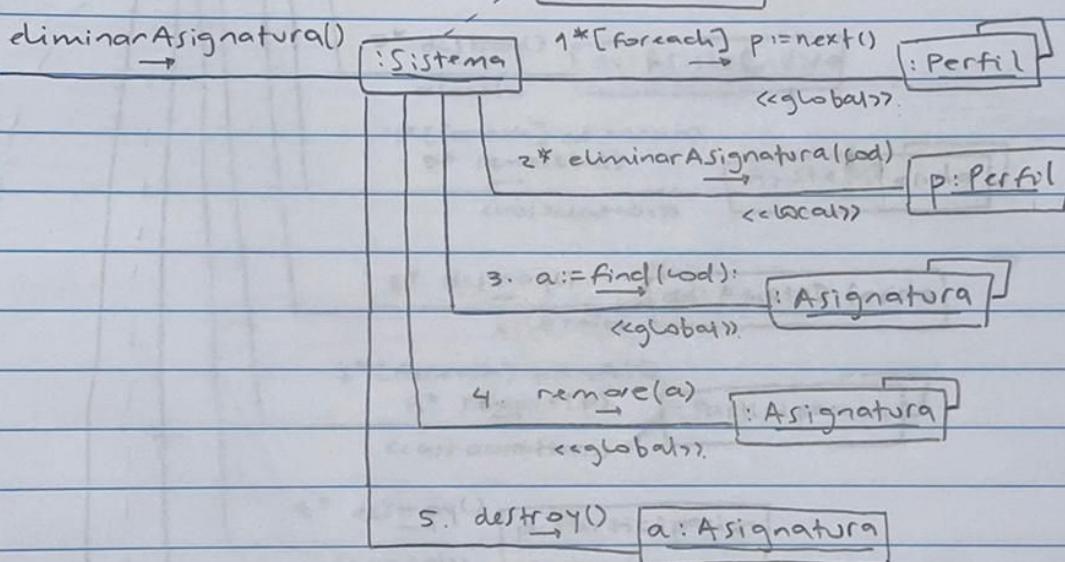
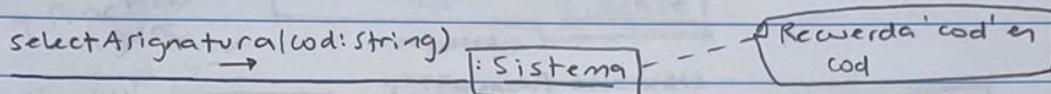
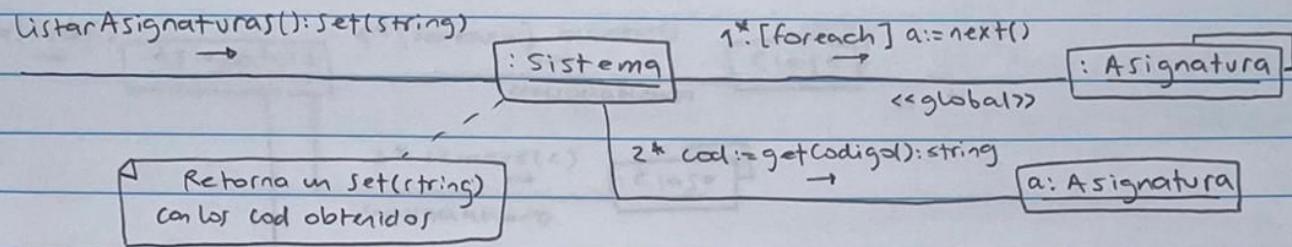
Post3: Para cada instancia de Clase conectada a la instancia Asignatura de código 'cod', se eliminan sus conexiones con y la instances de AsisteDiferido, AsisteEnVivo y Participación.

Post4: Se desconectan las instancias de Clase conectadas a la instancia de Asignatura de código 'cod' y se eliminan dichas instancias.

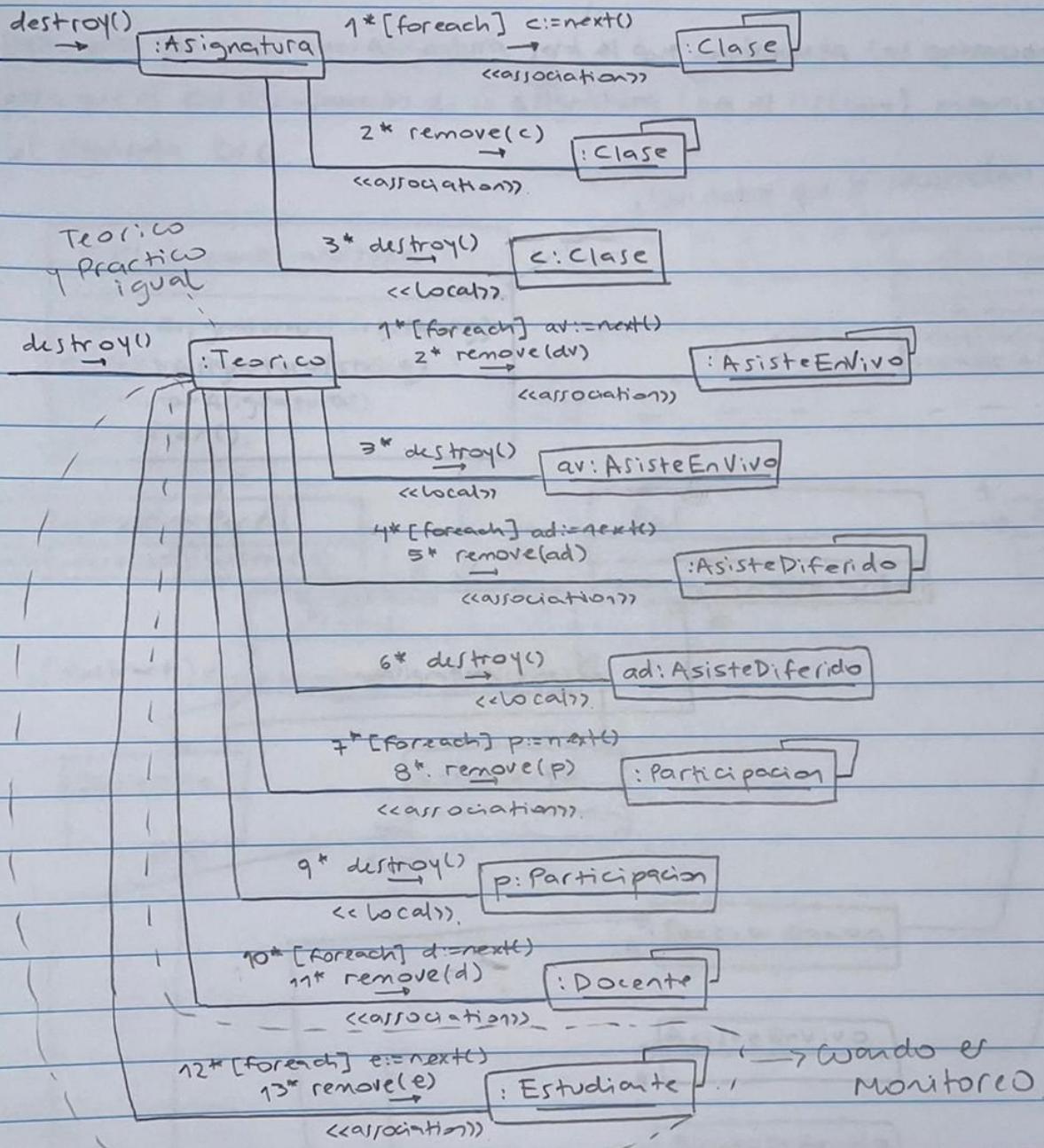
Post5: Se elimina del Sistema la instancia de Asignatura de código = 'cod'



(4)



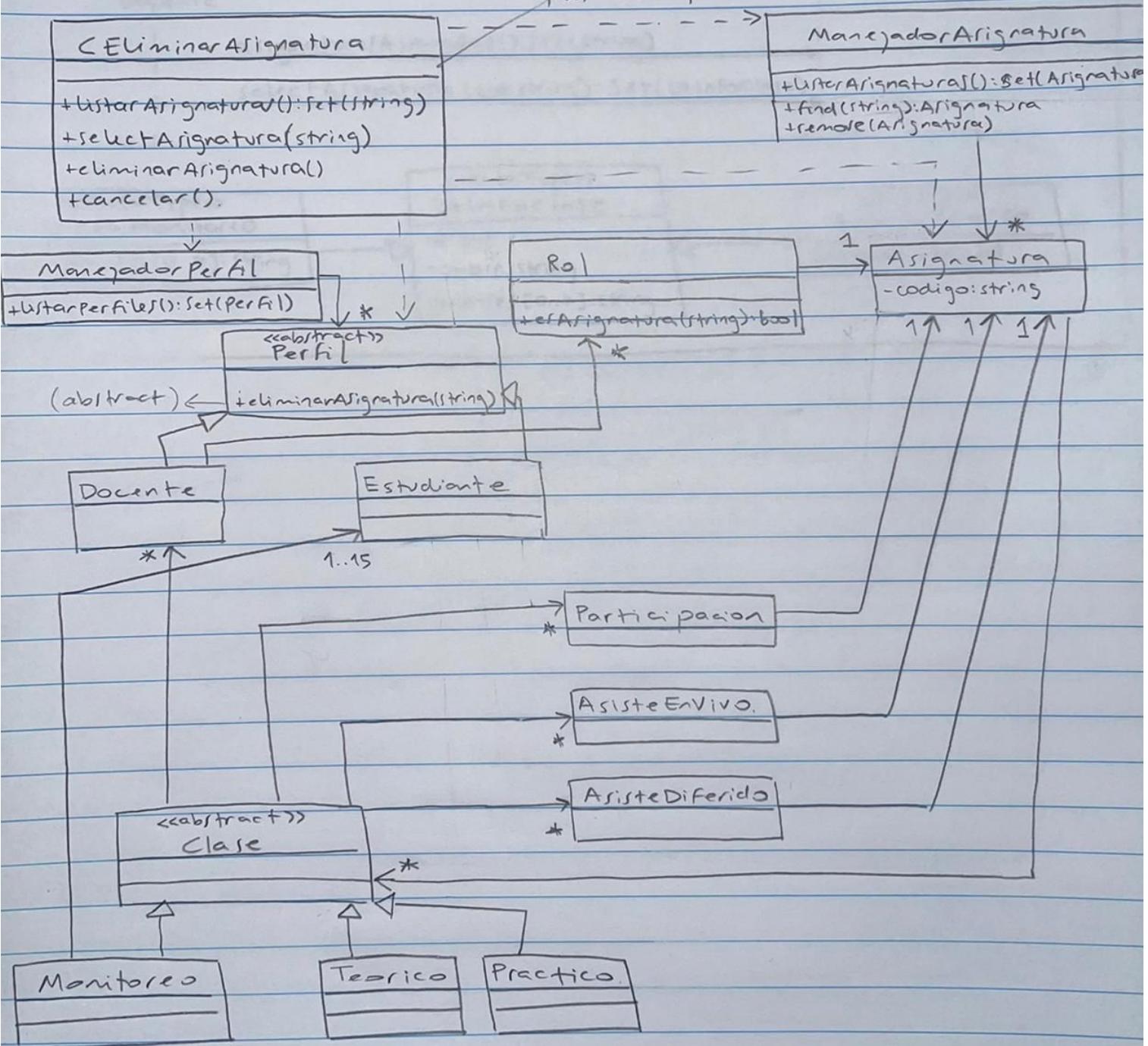
(4)



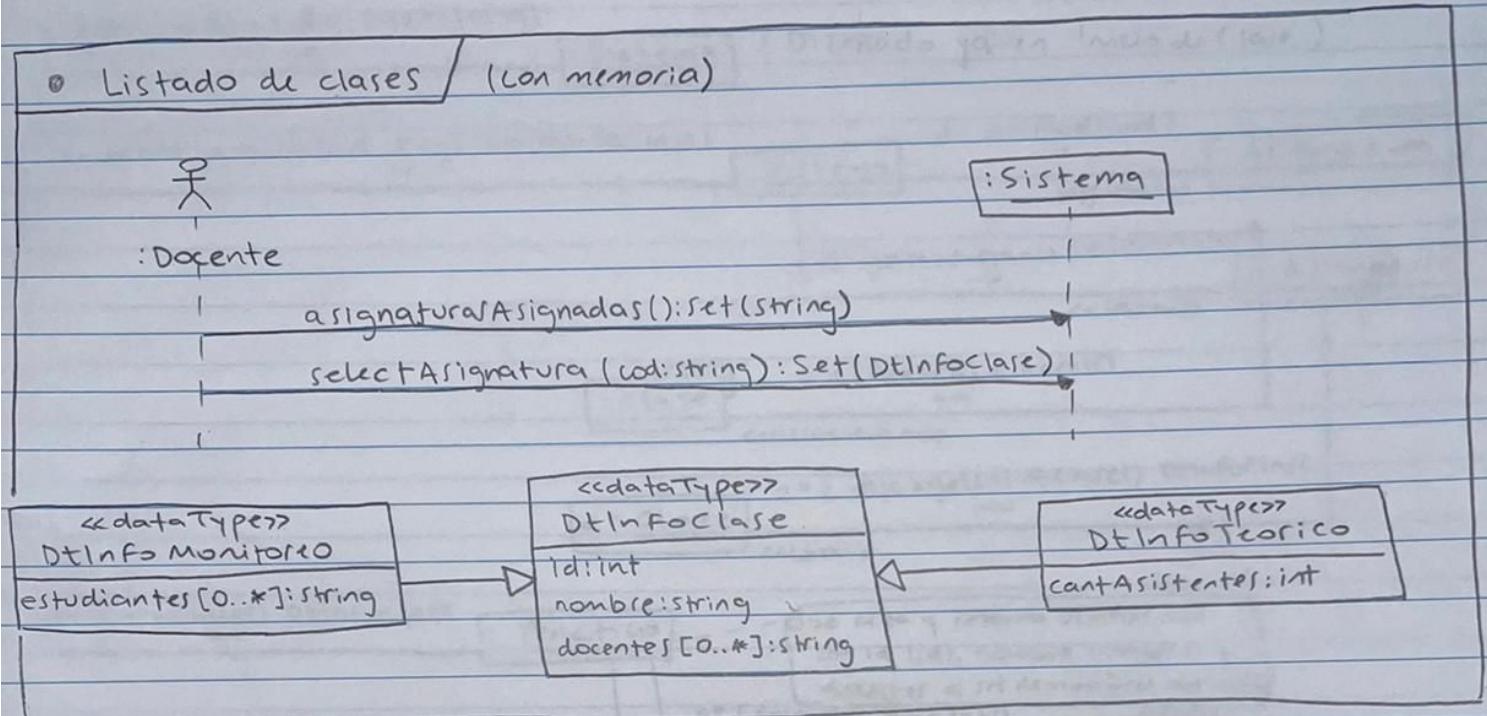
(4)

Definimos que `C EliminarAsignatura` será el que implementa las operaciones del sistema para que el CV Eliminación de la Asignatura (no el sistema), entonces tendremos el siguiente DCD.

los datos que se recuerdan.



(5)



5

asignaturasAsignadas(): Set(string)

: Sistema

(Diseñado ya en Inicio de Clase.)

selectAsignatura(cod:string): Set(DtInfoClase)

: Sistema

1. a:=find(cod)

«global»

: Asignatura

2. getDtInfoClase(): Set(DtInfoClase)

«local»

a: Asignatura

: Clase

2\*1 [foreach] c:=next()

«associations»

c: Clase

2\*2 dtlc:=getDtInfoClase().DtInfoClase

«local»

getDtInfoClase(): DtInfoClase

: Practico

Se crea y retorna DtInfoClase con id (id), nombre (nombre) y docentes el set de nombres obtenidos.

1\* [foreach] d:=next()

: Docente

«associations»

2\* nombre:=getEmail(): string

d: Docente

getDtInfoClase(): DtInfoTeorico

: Teorico

Se crea y retorna DtInfoTeorico con id (id), nombre (nombre) y docentes el set de nombres obtenidos y cantAsistentes

3. cantAsistentes =  
getcantAsistentes(); int.

1\* [foreach] d:=next()

: Docente

«associations»

2\* nombre:=getEmail(): string

d: Docente

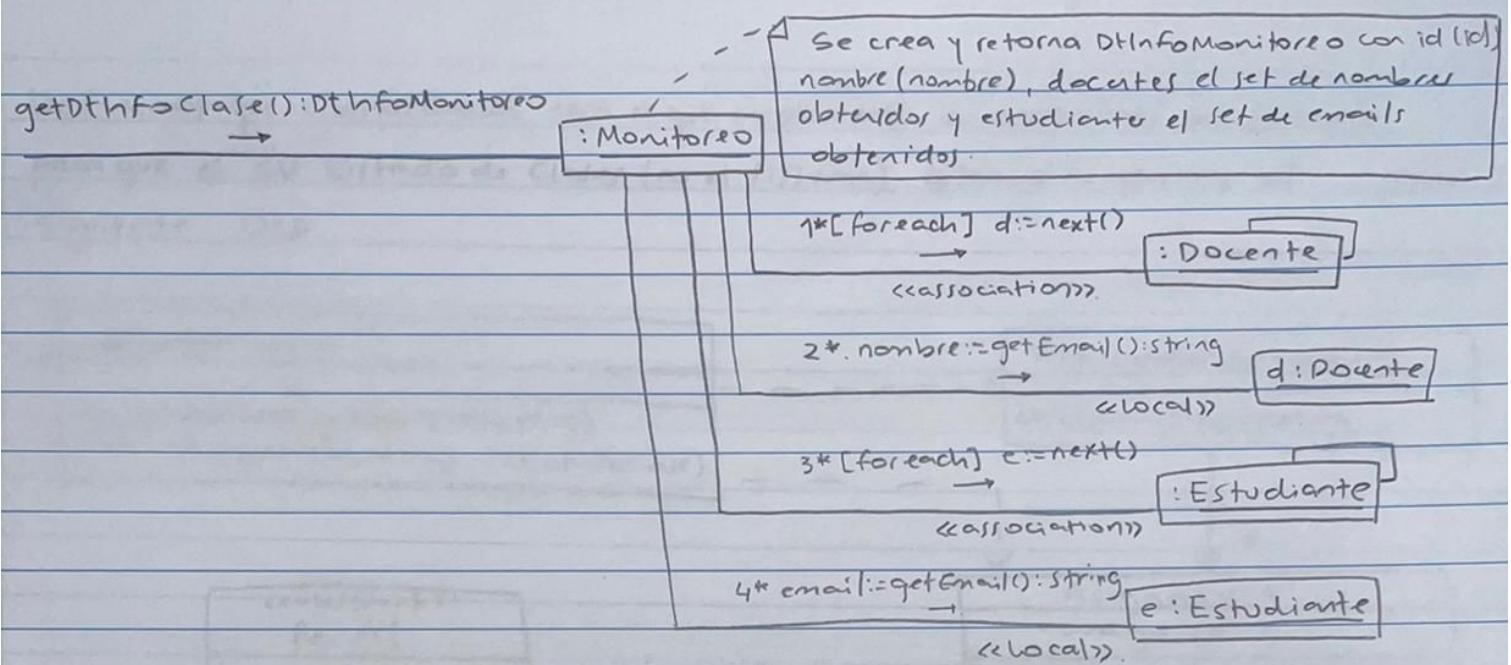
«local»

3.1\* [foreach] av:=next()

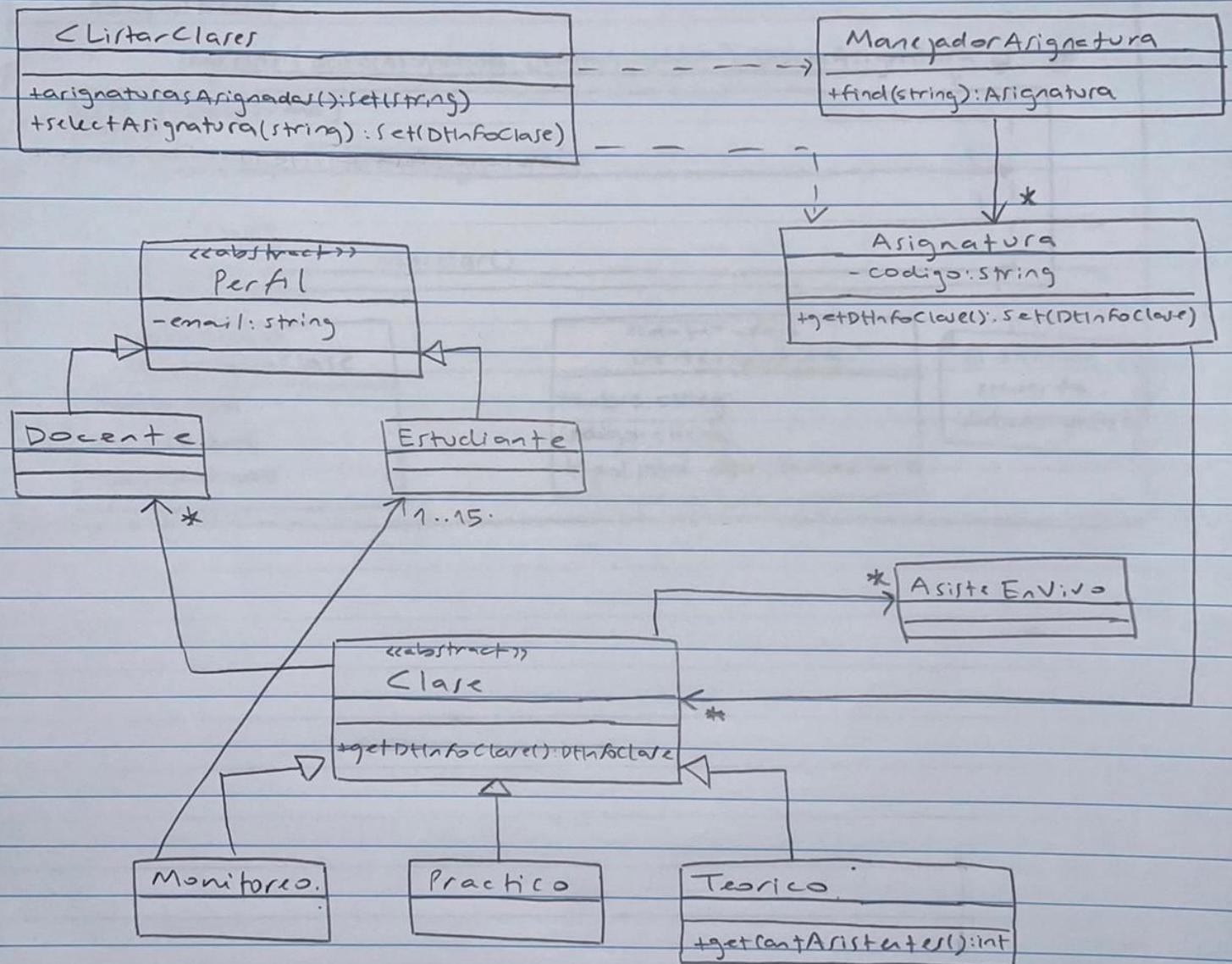
: AsisteEnVivo

Asegurante  
cantAsistentes

5



Definimos que `<ListarClases` será el que implementa las operaciones del sistema para que el CU Listado de Clases (no el sistema), entonces tendremos el siguiente DCD.



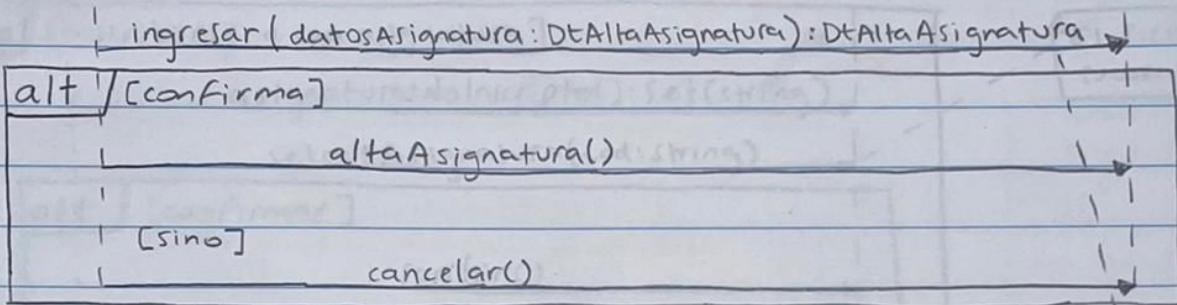
(6)

## Alta Asignatura / (con memoria)



Administrador

:Sistema



`<<datatype>>`  
DtInstanciaClase

teorico: bool  
practico: bool  
monitorio: bool

`<<datatype>>`  
DtAsignatura

nombre: string  
codigo: string  
tipoclasses: DtInstanciaClase

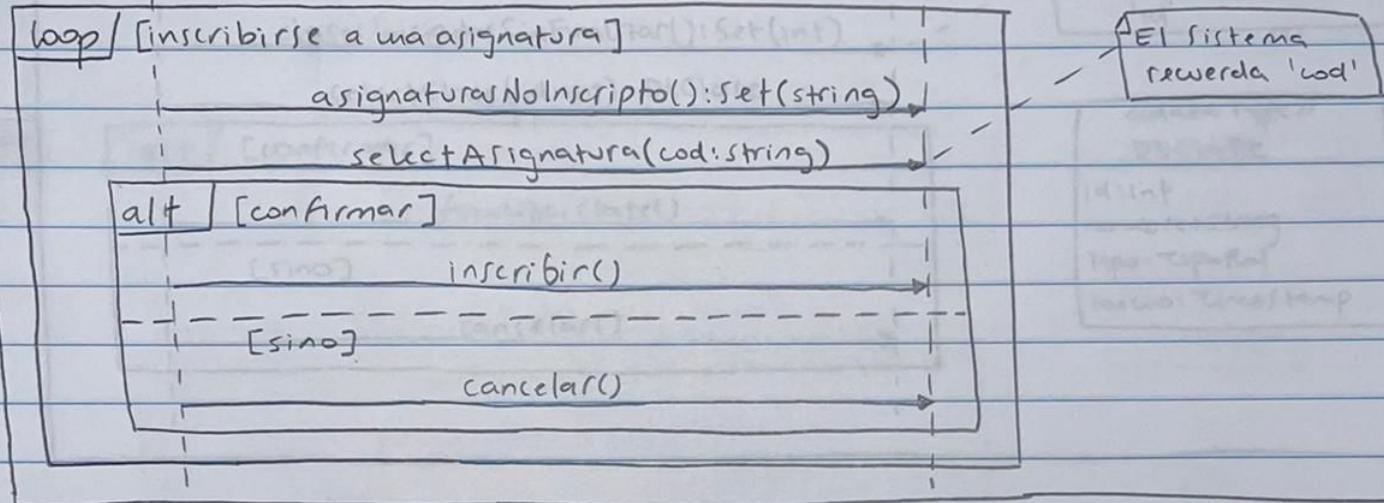
El sistema  
recuerda  
'datosAsignatura'

7

## Inscripción a las asignaturas. (con memoria)

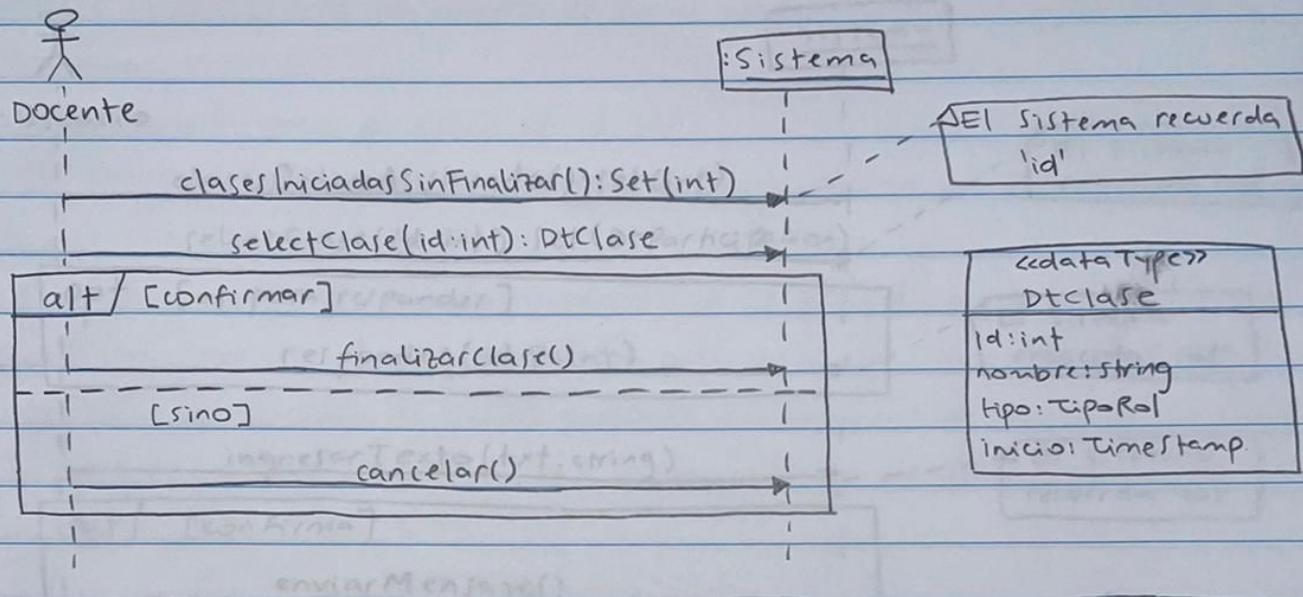


:Estudiante.



3

## Finalización de clase

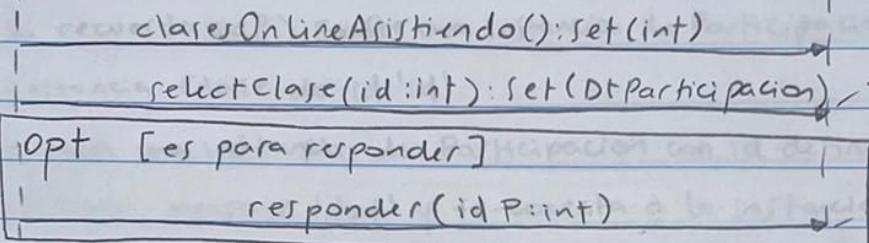


9

## Envio de mensaje / (con memoria)



: Usuario



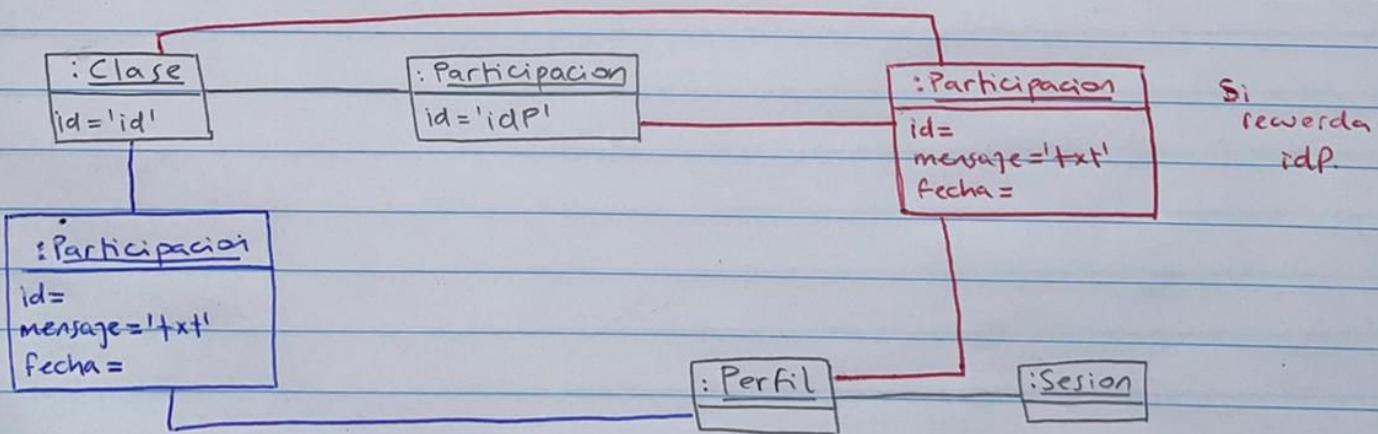
: Sistema

El sistema recuerda 'id'

El sistema recuerda 'idP'

El sistema recuerda 'txt'

`class DtParticipacion`  
`id:int`  
`fecha:Timestamp`  
`mensaje:string`  
`responde [0..1]: DtParticipacion`



(9)

## enviarMensaje() [CONTRATO]

Pre1: Existe en el sistema una instancia de Perfil asociada a la instancia de sesión del Sistema.

Pre2: Existe una instancia de Clase con id 'id'

Pre3: (Si recuerda idP) Existe una instancia de Participación con id 'idP' asociada a la instancia Clase de id 'id'

Post: Se crea una instancia de Participación con id definido por el Sistema, fecha del Sistema, mensaje 'txt' y se conecta a la instancia Clase de id 'id', y se conecta a la instancia de Perfil asociada a la instancia de sesión del Sistema. (Si recuerda idP) se conecta también con la instancia de Participación de id 'idP'.  
cancelar()

(10)

## Asistencia a clase en vivo (con memoria)



:Estudiante

asignaturasInscriptas(): Set(string)

clasesOnlineDisponibles(cod: string): Set(int)

selectClase(id: int): DtAsistir

alt [confirmar]

[Sí] asistirClaseEnVivo()

[Cancelar]

[No] cancelar()

:Sistema

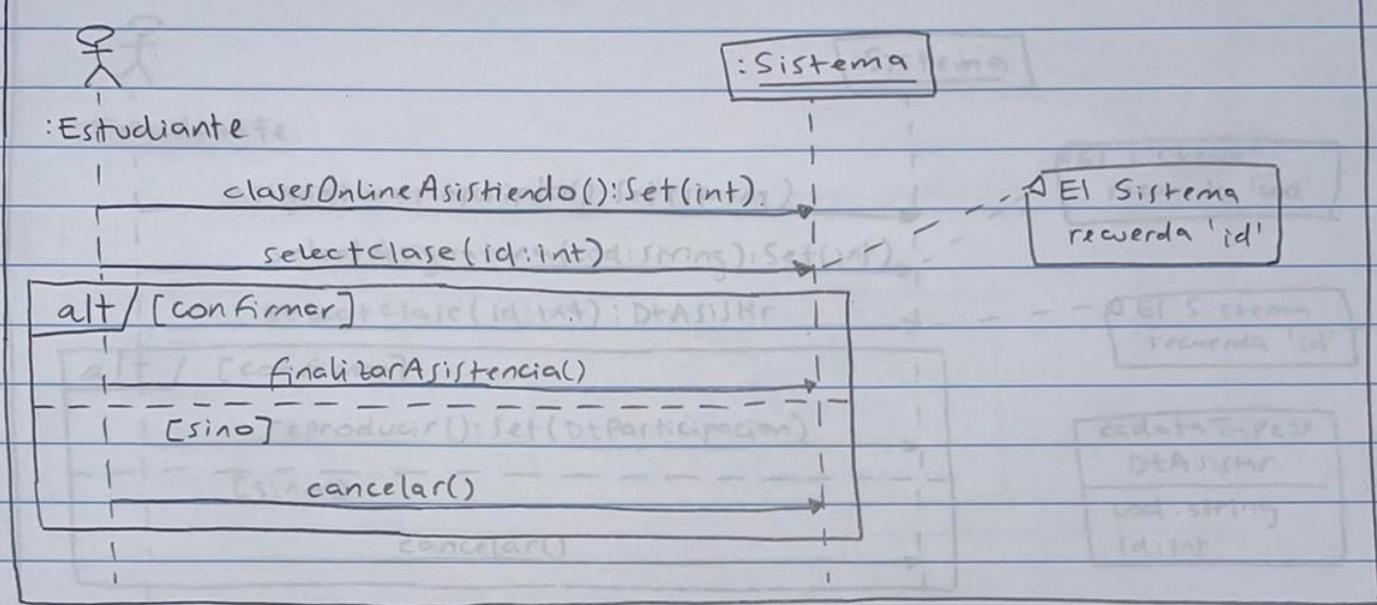
El Sistema recuerda 'cod'

El Sistema recuerda 'id'

«dataTypes»  
DtAsistir:  
cod: string  
id: int

11

Finalización de asistencia a clínica en vivo / (con memoria).



(12)

## Reproducción de clase en diferido (con memoria)



:Estudiante

| asignaturasInscripto(): Set<String>

| selectAsignatura(cod:String): Set<Int>

| selectClase(id:Int): DtAsistir

alt / [confirma]

| reproducir(): Set<DtParticipacion>

[Sí/no]

| cancelar()

:Sistema

El sistema recuerda 'cod'

El Sistema recuerda 'id'

<<dataType>>  
DtAsistir  
cod:String  
id:Int

<<dataType>>  
DtParticipacion  
id:Int  
fecha:Timestamp  
mensaje:String  
responde{0..1}: DtParticipacion

## Tiempo de dictado de clases (con memoria)



Administrador

: Sistema

tiempoDeClasesDictadas(): Set(DtDictadoAsignatura)

**<<datatype>>**  
DtDictadoAsignatura  
codigo:string  
nombre:string  
tiempoTotal:Time

**<<datatype>>**  
Time  
hora:int  
min:int  
segundos:int

asignaturaAsignadas(): Set(Asignatura)

Retorna un sistema y contiene un conjunto de strings. Cada String es el código de cada asignatura de la asignatura del Sistema.

asignaturaAsignada(cod:string): Set(ClaseAsistencia)

para: Recibe una instancia de Administrador con codigo!cod!

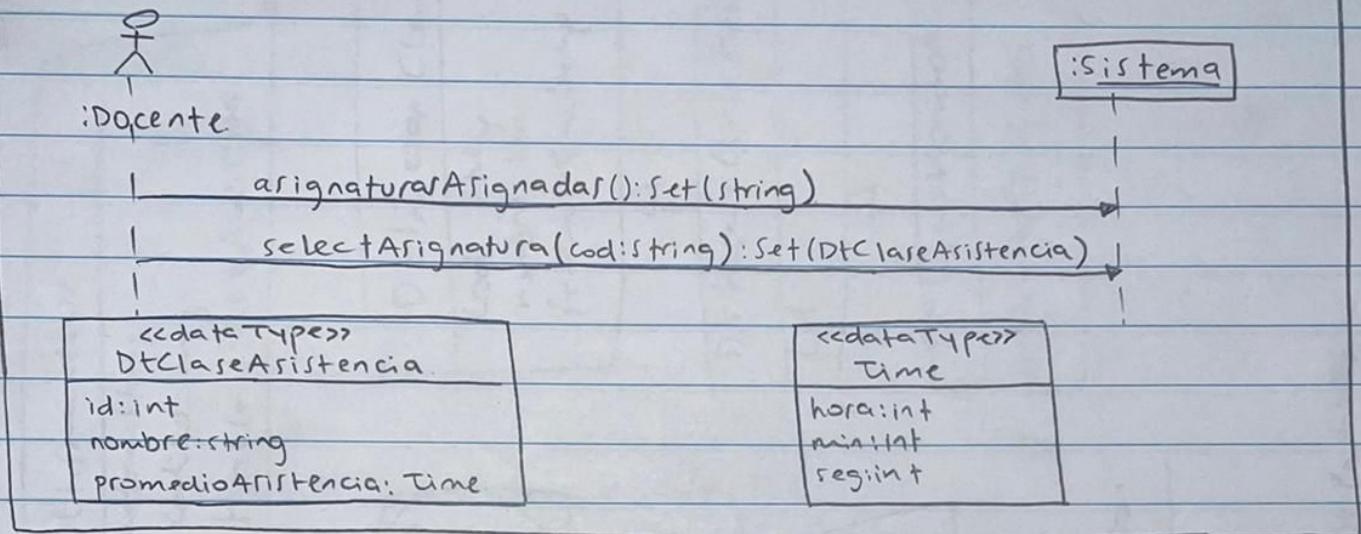
para: Se contiene y retorna un conjunto de DtDictadoAsignatura, con informacion de las asignaciones. Cada elemento sera la representacion de la asignatura del sistema 'ver' la informacion que contendra cada DtDictadoAsignatura es:

- id

- nombre

- promedioAsistica, que es el promedio de calificaciones de los estudiantes por los estudiantes

## Tiempo de Asistencia a clase / (con memoria)



`asignaturasAsignadas(): Set<String>`

POST: Se retorna y construye un conjunto de `String`, cada `String` es el código de cada instancia de `Asignatura` del Sistema.

`selectAsignatura(cod: String): Set<DtClaseAsistencia>`

PRE: Existe una instancia de `Asignatura` con código '`cod`'

POST: Se construye y retorna un conjunto de `DtClaseAsistencia`, con información de las instancias clase asociadas a la instancia de `Asignatura` de código '`cod`'.

La información que contendrá cada `DtClaseAsistencia` es:

- `id`
- `nombre`
- `promedioAsistencia`, que es el promedio de asistencia en vivo de la clase por los Estudiantes.