

Alta Usuario (con memoria)



Administrador

: Sistema

loop / [quiere agregar usuario]

ingresarDatosPerfil(datosUsuario: DTUsuario)

alt / [es Estudiante]

ingresarEstudiante(doc: string)

[es Docente]

ingresarDocente(inst: string)

alt / [desea confirmar]

altaUsuario()

[si no]

cancelar()

El sistema recuerda
'datosUsuario'

El sistema recuerda
'doc'

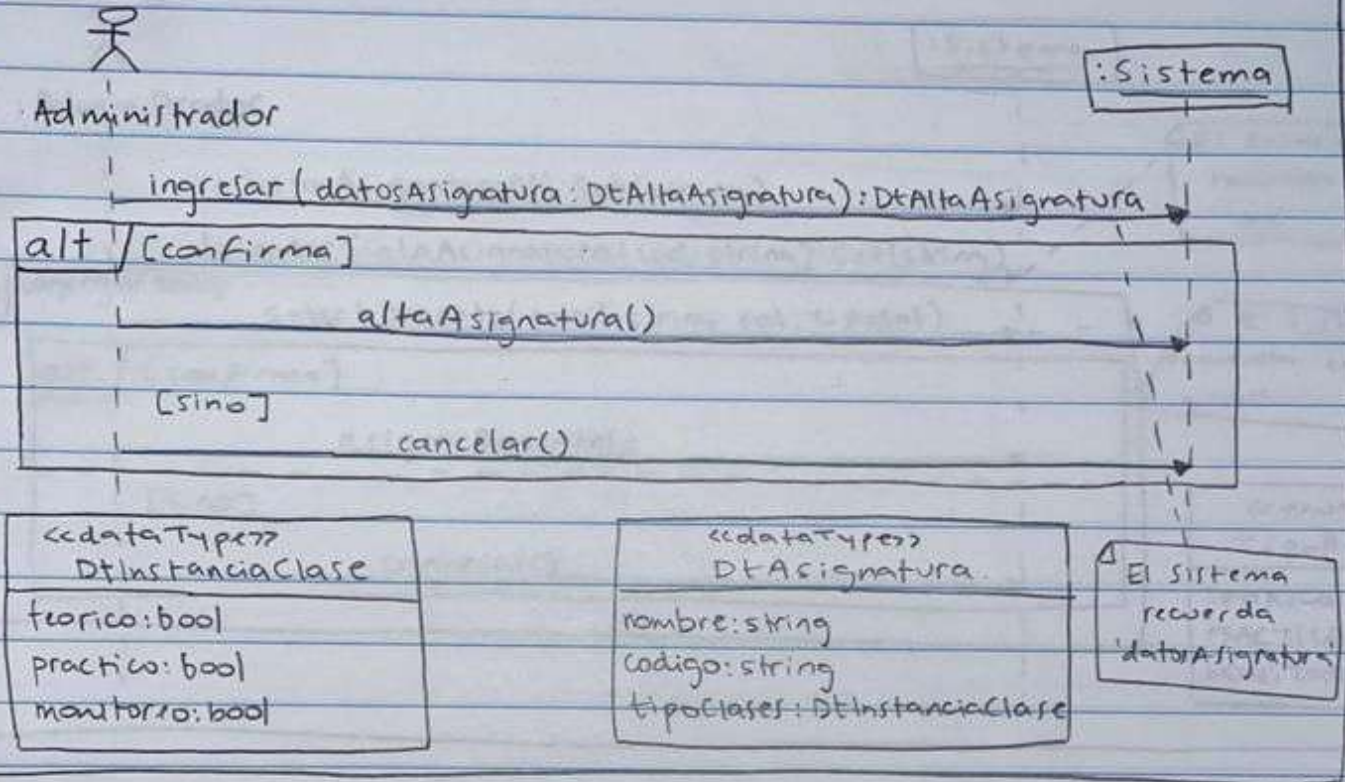
El sistema
recuerda 'inst'

<<dataTypes>>

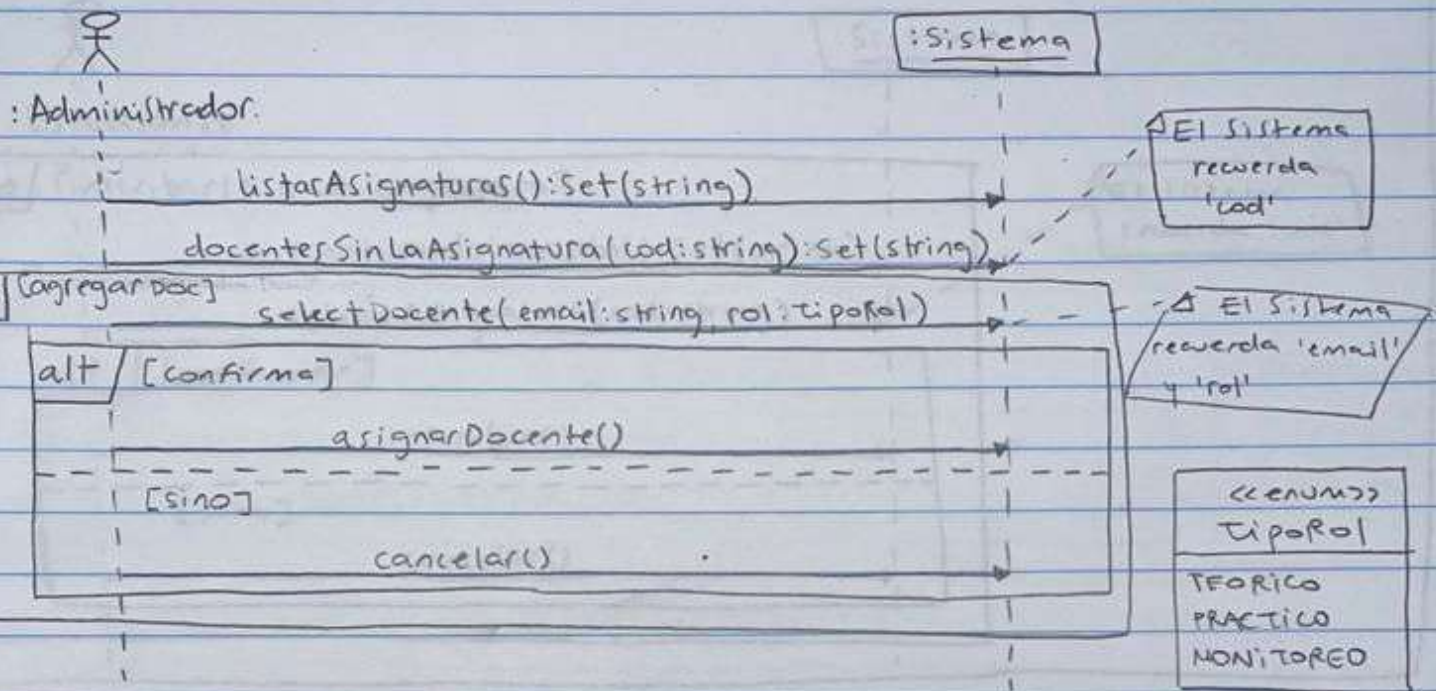
DTPerfil

nombre: string
imagenUrl: string
email: string
password: string

Alta Asignatura (con memoria)



Asignacion de docentes a una asignatura (con memoria)



Inscripcion a las asignaturas / (con memoria)



:Estudiante

:Sistema

loop / [inscribirse a una asignatura]

asignaturaNoInscripto():Set(string)

selectAsignatura(cod:string)

alt [confirmar]

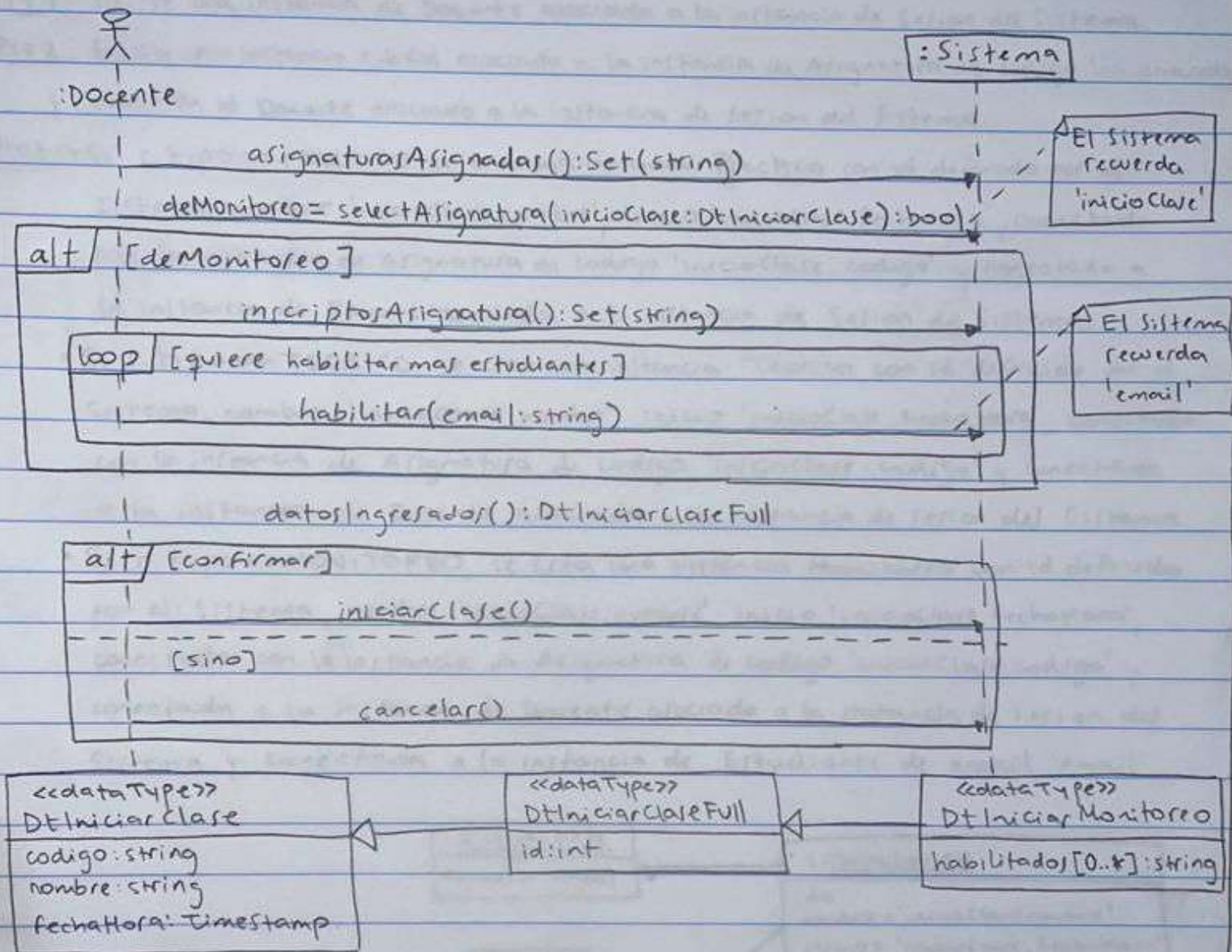
inscribir()

[sino]

cancelar()

El sistema recuerda 'cod'

Inicio de Clase (con memoria)



iniciarClase() [CONTRATO]

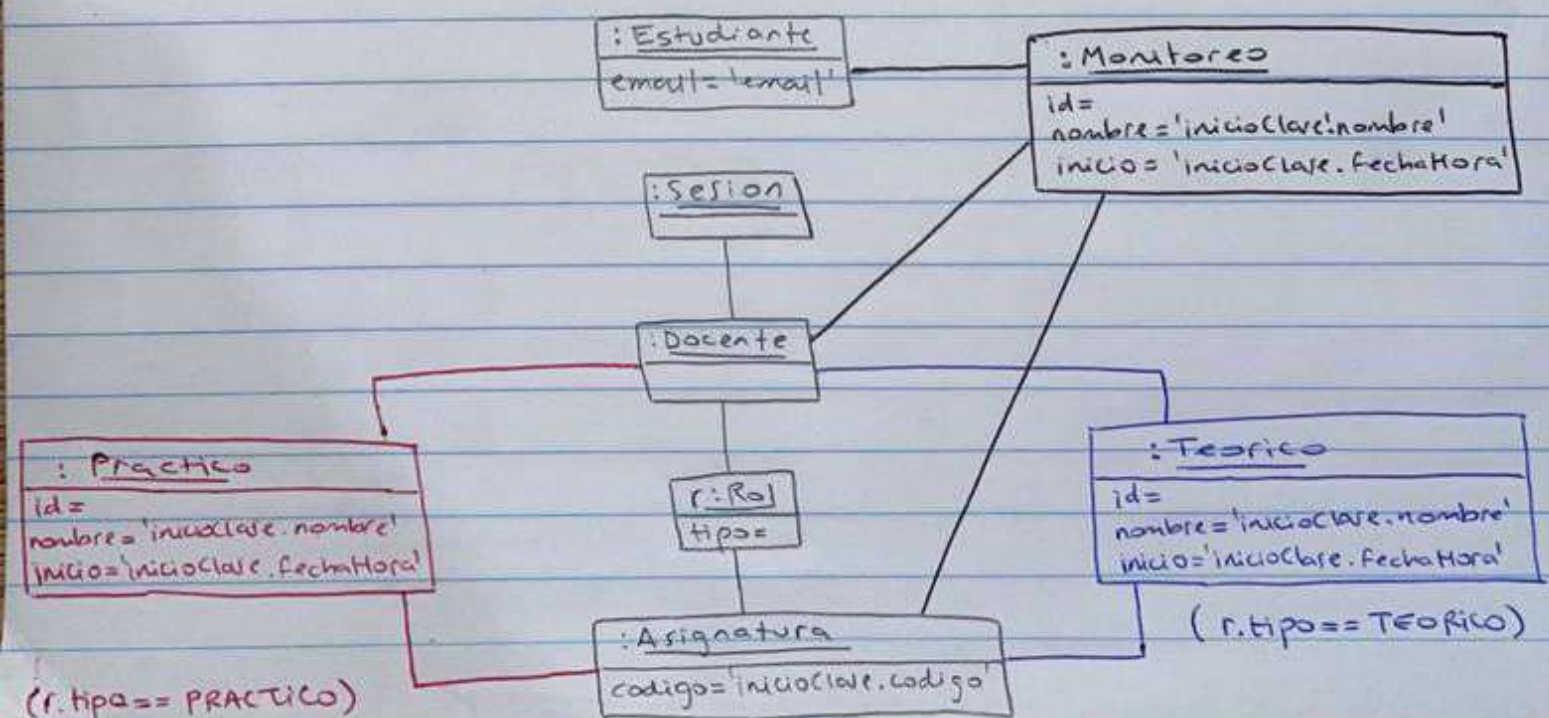
Pre1: Existe una instancia de Docente asociada a la instancia de Sesión del Sistema.

Pre2: 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.

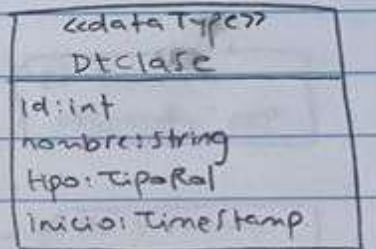
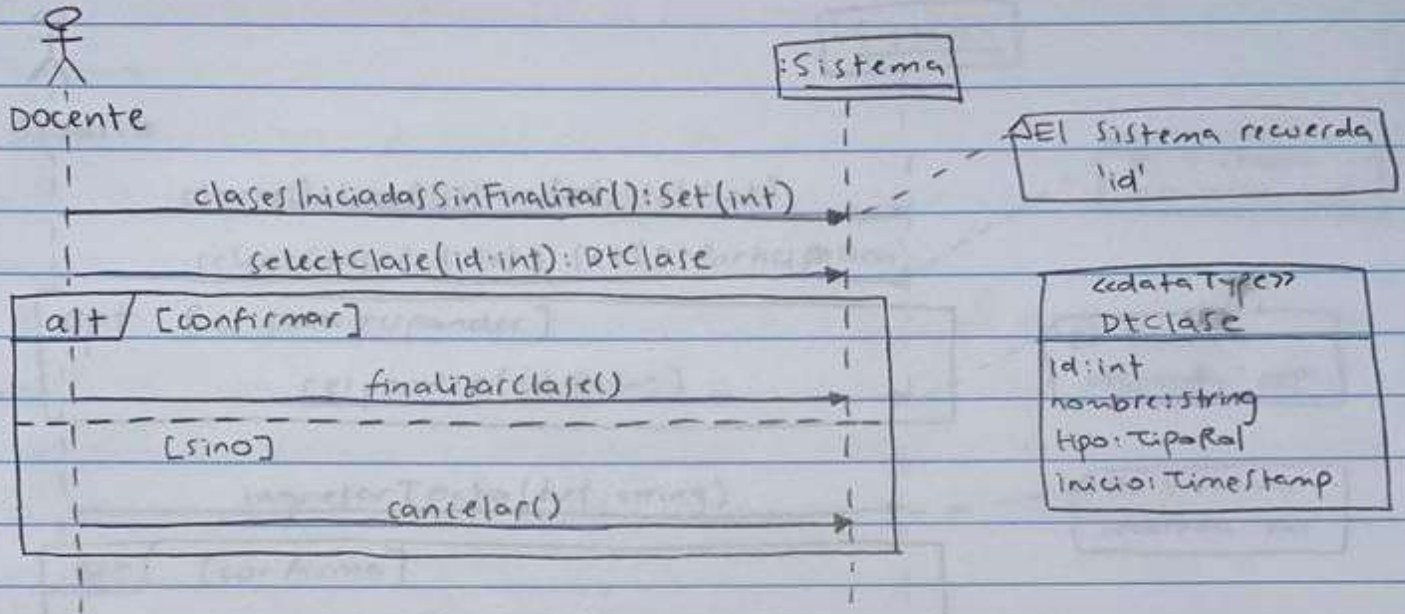
Post1: Si $r.tipo == PRACTICO$ se crea una instancia Practico con id definido por el Sistema, nombre 'inicioClase.nombre', inicio 'inicioClase.fechaHora', conectada con la instancia de Asignatura de código 'inicioClase.codigo' y conectada a la instancia de Docente asociada a la instancia de Sesión del 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.

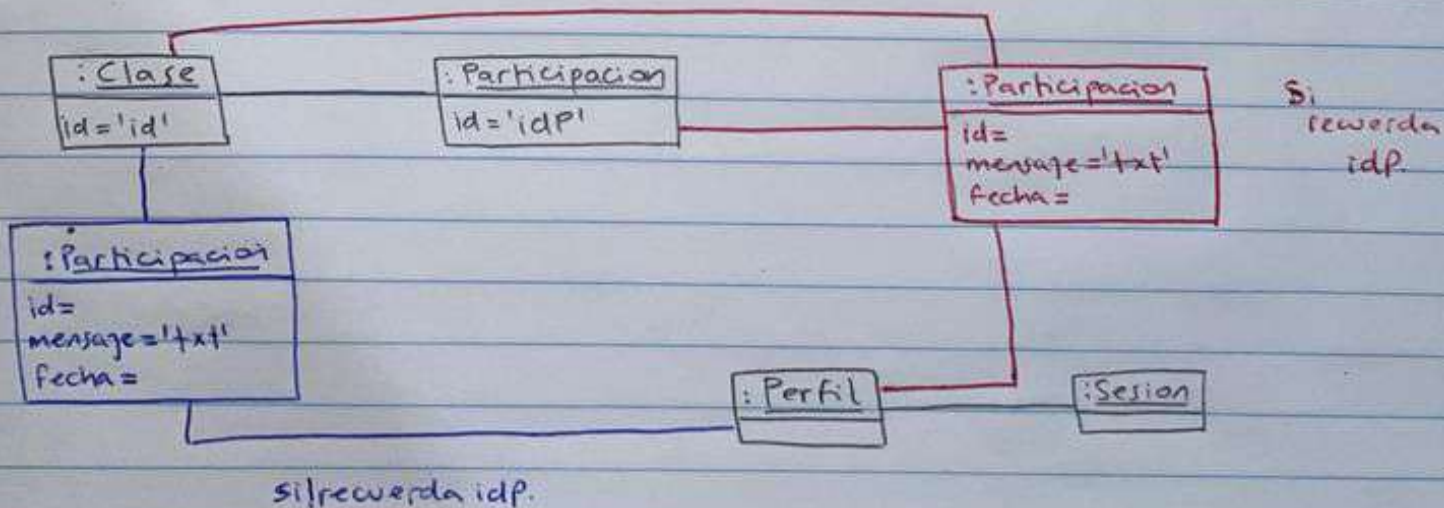
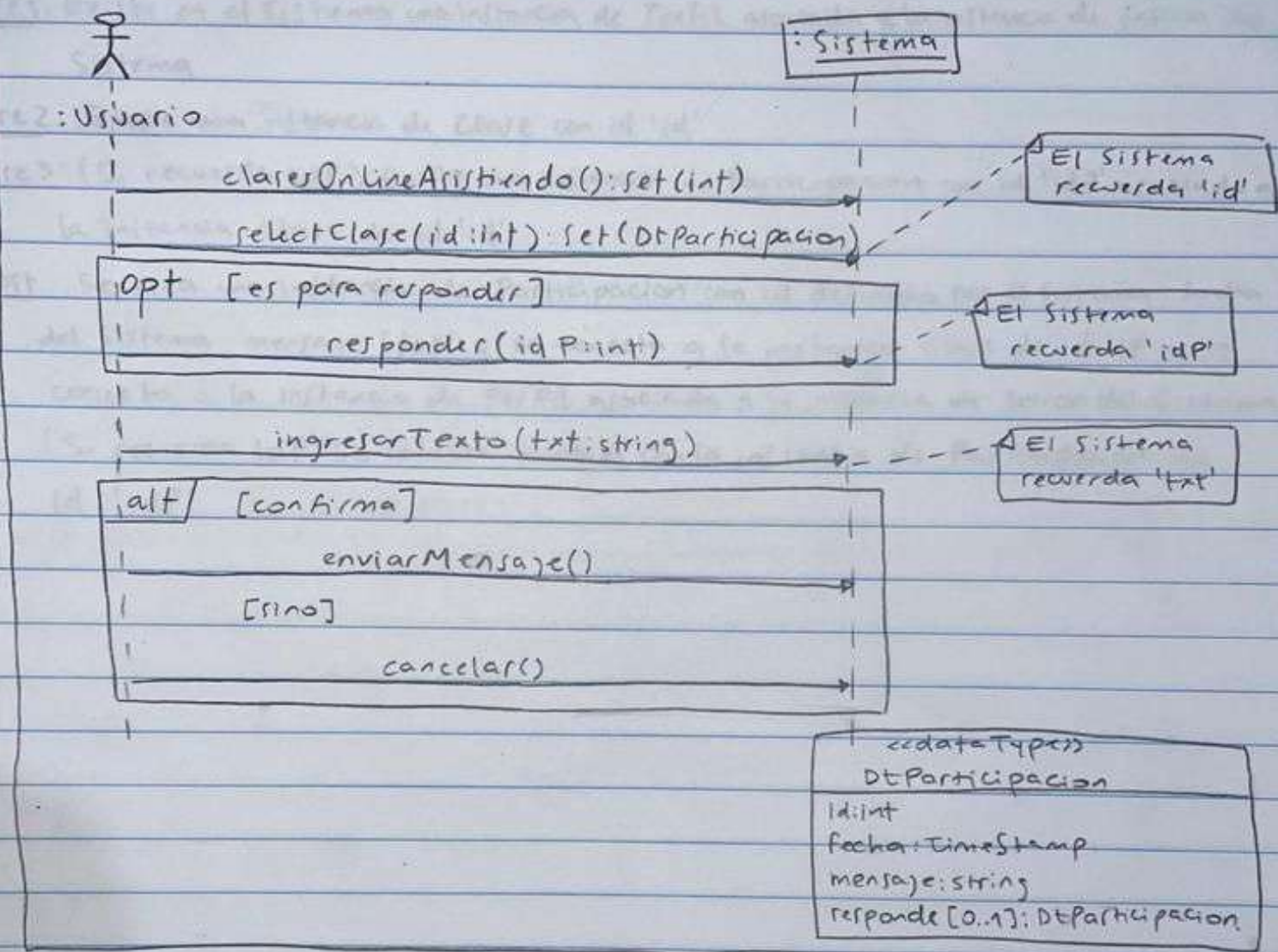
• Si $r.tipo == MONITOREO$, se crea una instancia Monitoreo con id definido por el Sistema, nombre 'inicioClase.nombre', inicio 'inicioClase.fechaHora', conectada con la instancia de Asignatura de código 'inicioClase.codigo' y conectada a la instancia de Docente asociada a la instancia de Sesión del Sistema y conectada a la instancia de Estudiante de email 'email'.



Finalización de clase



Envio de message / (con memoria)



enviarMensaje() [CONTRATO]

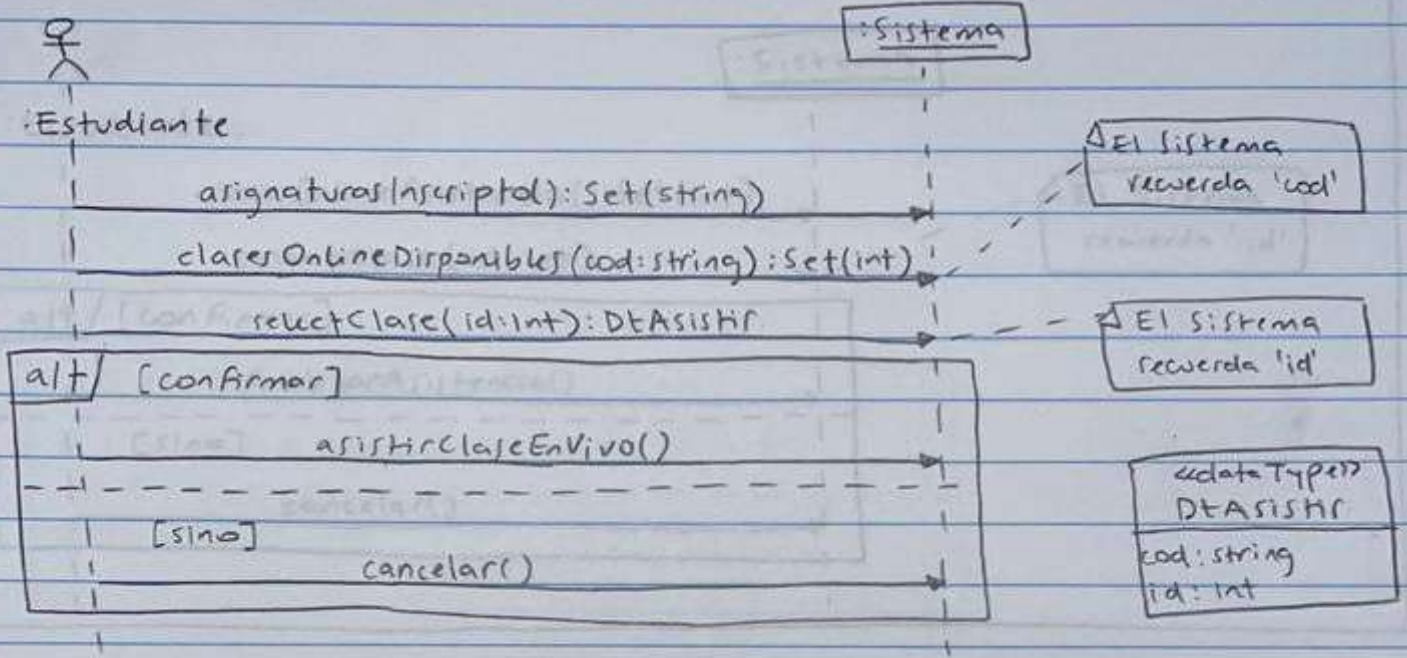
Pre1: Existe en el Sistema una instancia de Perfil asociada a la instancia de Sesión del Sistema.

Pre2: Existe una instancia de Clave con id 'id'.

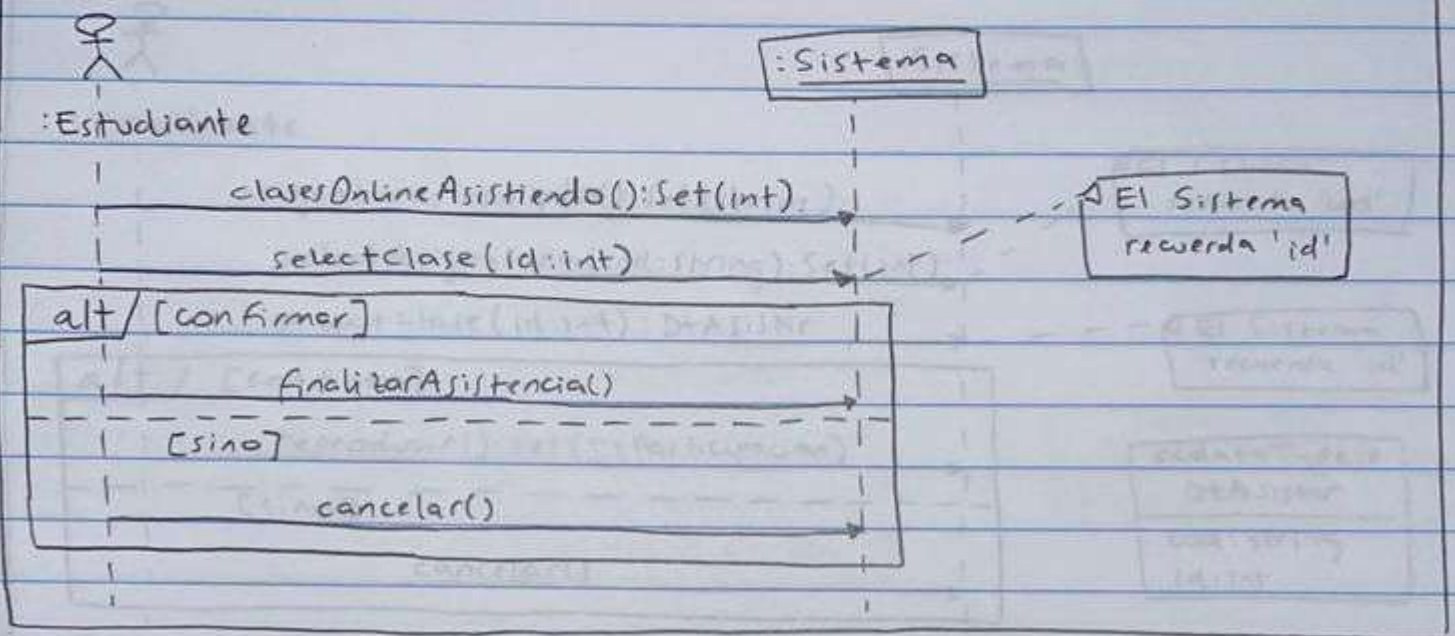
Pre3: (Si recuerda idP) Existe una instancia de Participación con id 'idP' asociada a la instancia Clave de id 'id'.

Post: Se crea una instancia de Participación con id definido por el Sistema, fecha del Sistema, mensaje 'txt' y se conecta a la instancia Clave 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'.

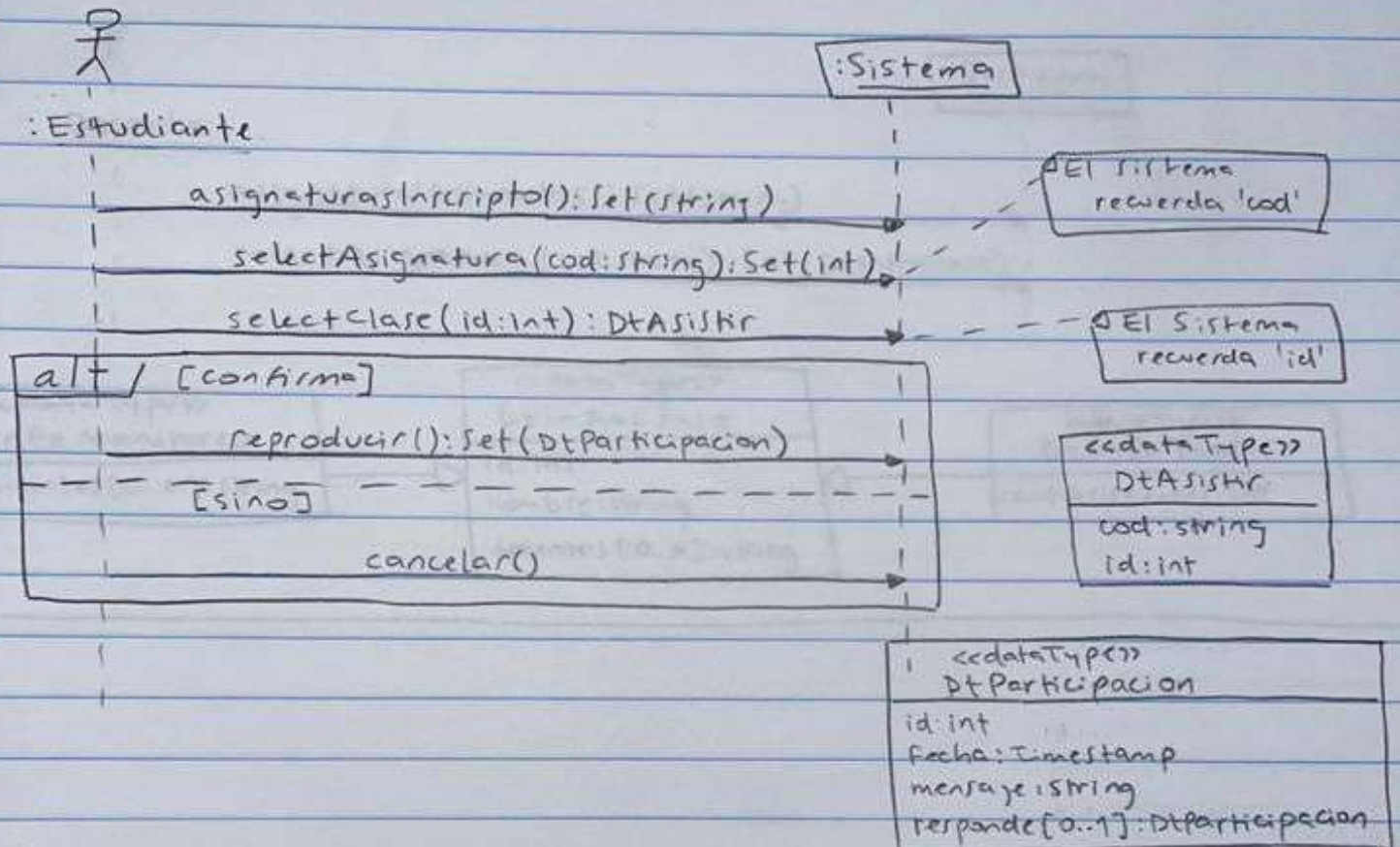
Asistencia a clase en vivo (con memoria)



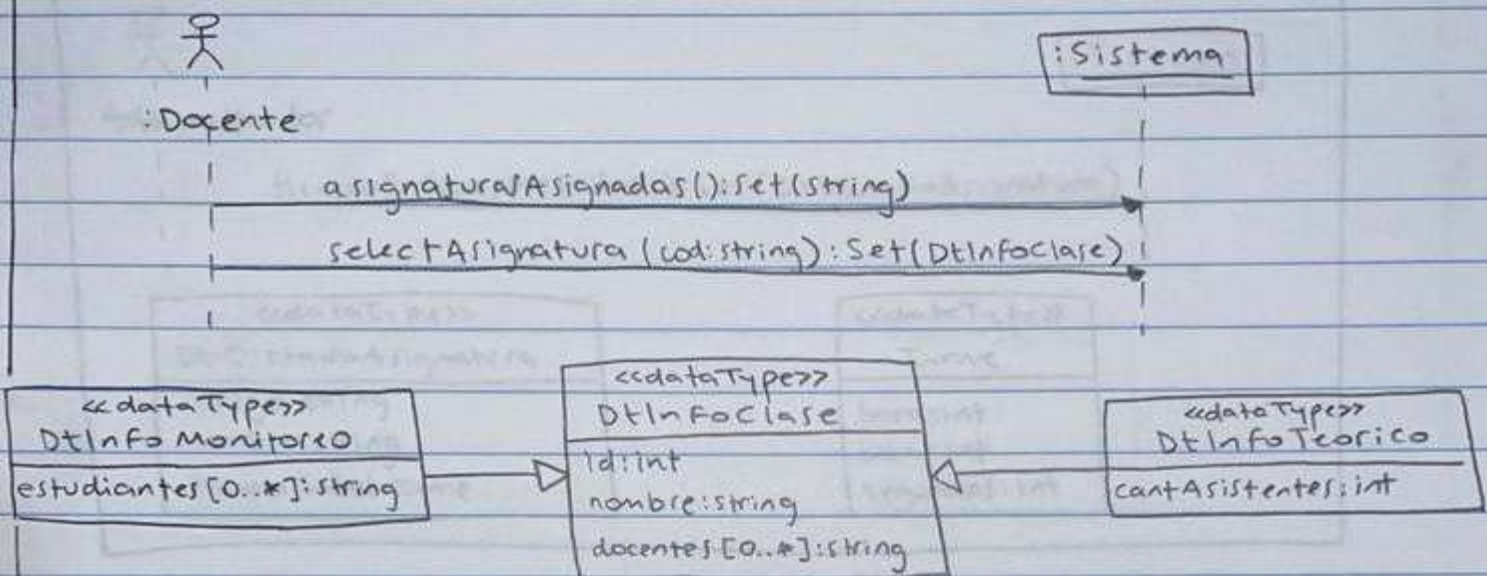
Finalización de asistencia a clase en vivo / (con memoria)



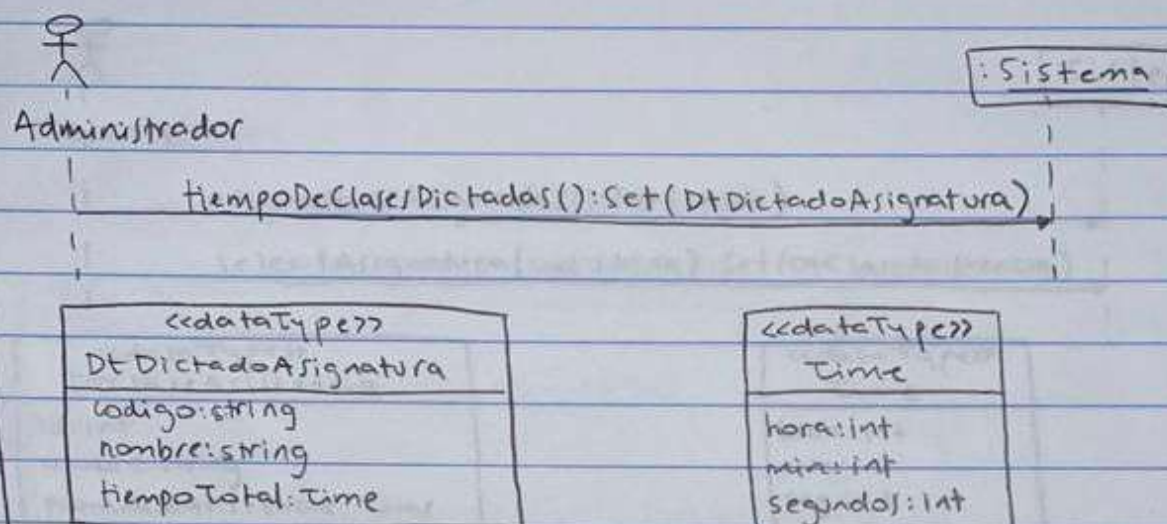
Reproducción de clase en diferido (con memoria)



Listado de clases (con memoria)



Tiempo de dictado de clases (con memoria)



asignatura: Asignatura(): Set(string)

POST Se retorna y contiene un conjunto de string, cada string es el código de cada asignatura del Sistema

class Asignatura (cod: string): Set(DtClaseAsignatura)

POST Se crea una instancia de Asignatura con código 'cod'

POST Se crea y retorna un conjunto de DtClaseAsignatura con información

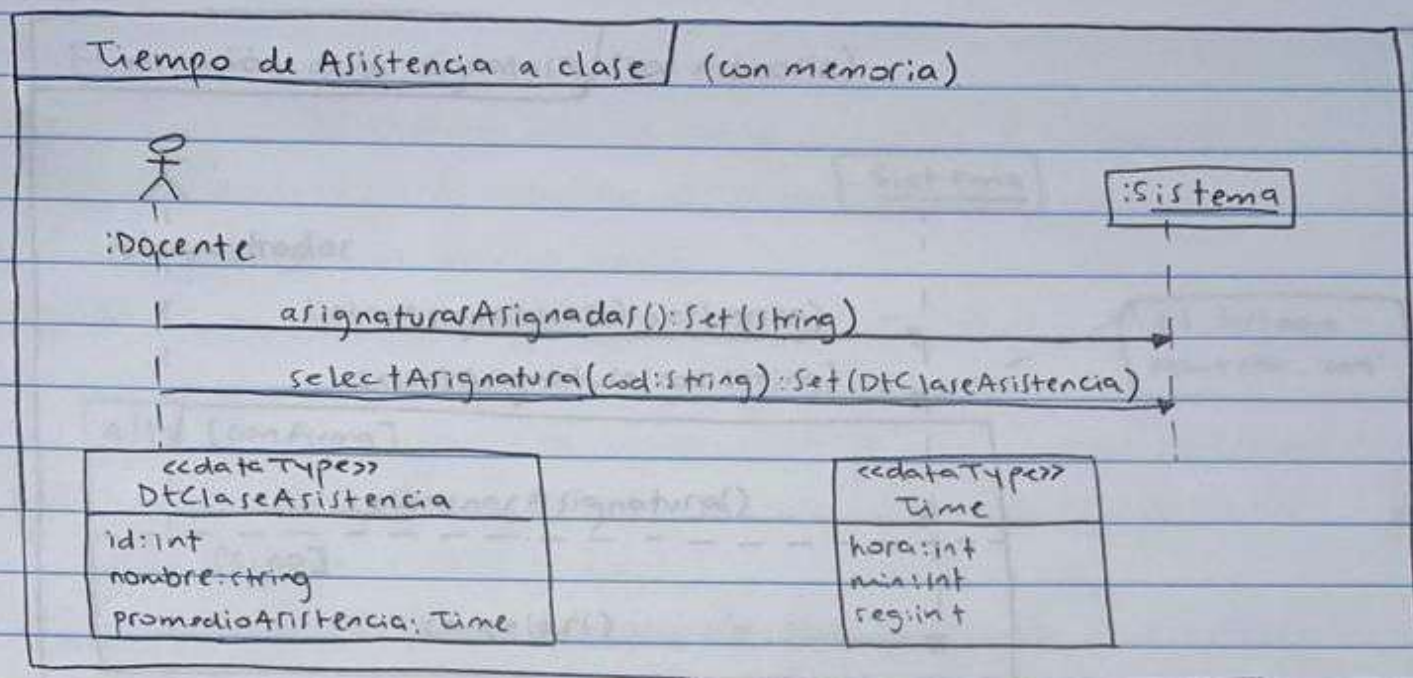
de las asignaturas. Cada 'DtClaseAsignatura' a la vez tiene la información de código 'cod'

La información que contiene cada 'DtClaseAsignatura' es:

- id

- nombre

- tiempo de dictado que es el promedio de dictado a lo largo de la clase por los estudiantes



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)

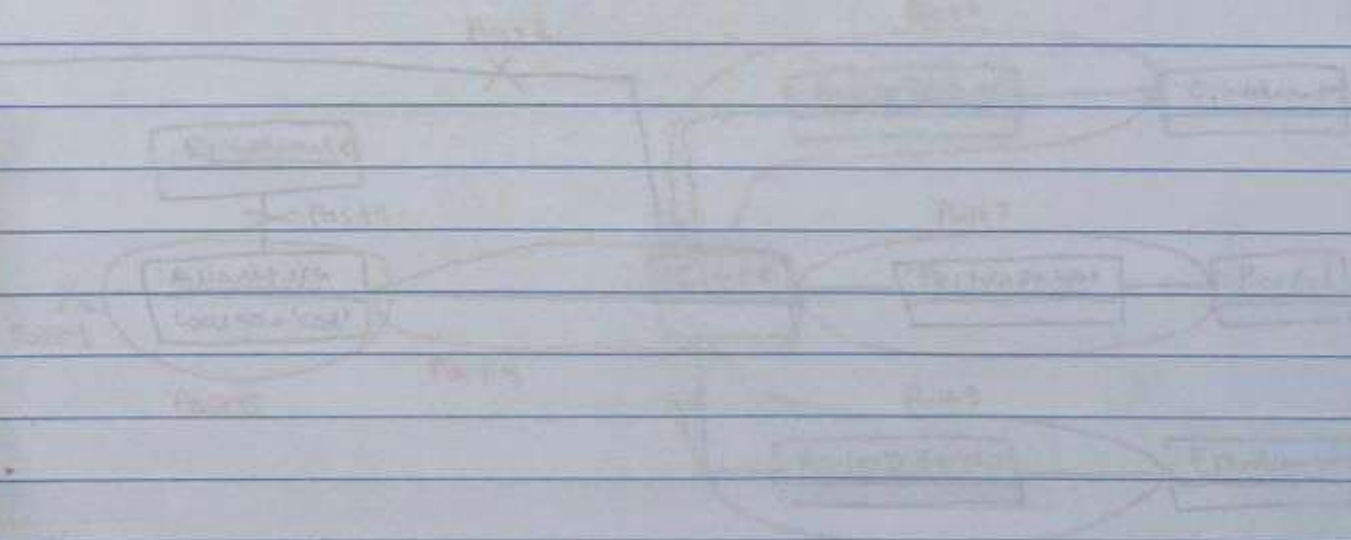
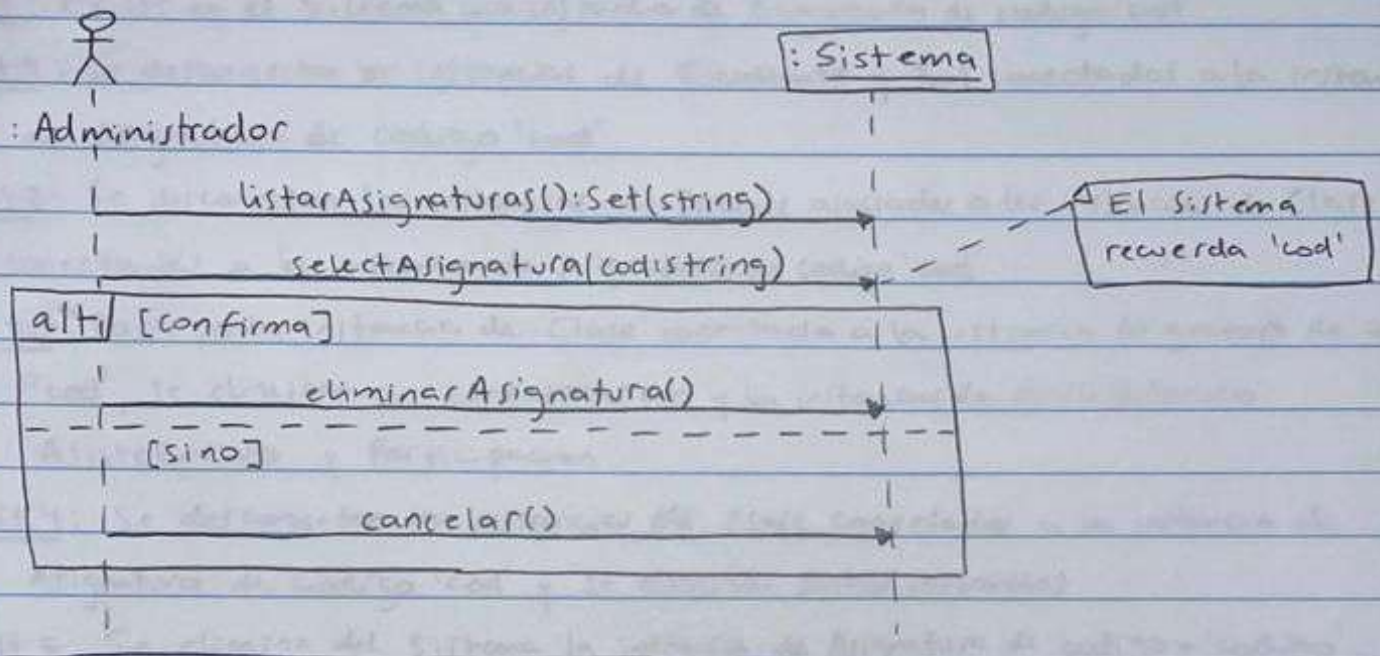
PRE: Existe una instancia de Asignatura con código 'cod'

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.

Eliminación de la asignatura (con memoria)



eliminarAsignatura() [CONTRATO]

Pre 1: Existe en el Sistema una instancia de Asignatura de código 'cod'

Post 1: Se desconectan las instancias de Estudiante y Rol conectadas a la instancia de Asignatura de código 'cod'

Post 2: Se desconectan las instancias de Docente asociadas a las instancias de Clase conectadas a la instancia de Asignatura de código 'cod'

Post 3: Para cada instancia de Clase conectada a la instancia Asignatura de código 'cod', se eliminan sus conexiones con y la instancias de AsisteDiferido, AsisteEnVivo y Participacion.

Post 4: Se desconectan las instancias de Clase conectadas a la instancia de Asignatura de código 'cod' y se eliminan dichas instancias.

Post 5: Se elimina del Sistema la instancia de Asignatura de código = 'codigo'

