

# Sviluppo Applicazioni Software

---

## Gestire i turni

Luca Barra, Jasmine Bizzo

Anno accademico 2023/2024



**UNIVERSITÀ**  
**DI TORINO**



<b>CAPITOLO 1</b>	<b>GESTIRE I TURNI</b>	<b>PAGINA 2</b>
-------------------	------------------------	-----------------

<b>CAPITOLO 2</b>	<b>MODELLO DI DOMINIO</b>	<b>PAGINA 8</b>
-------------------	---------------------------	-----------------

<b>CAPITOLO 3</b>	<b>SSD - SYSTEM SEQUENCE DIAGRAM</b>	<b>PAGINA 10</b>
3.1	Scenario principale	10
3.2	Eccezioni ed estensioni - Passo 1	11
3.3	Eccezioni ed estensioni - Passo 5	12
3.4	Eccezioni ed estensioni - Passo 6	13

<b>CAPITOLO 4</b>	<b>CONTRATTI</b>	<b>PAGINA 15</b>
4.1	Passo 1	15
4.2	Passo 2	17
4.3	Passo 3	17
4.4	Passo 4	17
4.5	Passo 5	17
4.6	Passo 6	18

<b>CAPITOLO 5</b>	<b>DCD - DESIGN CLASS DIAGRAM</b>	<b>PAGINA 20</b>
-------------------	-----------------------------------	------------------

<b>CAPITOLO 6</b>	<b>DSD - DESIGN SEQUENCE DIAGRAM</b>	<b>PAGINA 22</b>
-------------------	--------------------------------------	------------------



# 1

## Gestire i turni

### Informazioni generali

**Nome Caso d'Uso:** Gestire i turni

**Portata:** Sistema

**Livello:** Obiettivo utente

**Attore primario:** Organizzatore

**Parti interessate:** Chef, Cuochi

**Pre-condizioni:** L'attore deve essere identificato come Organizzatore

**Garanzie di successo o post-condizioni:** I turni di cucina e di servizio sono stati impostati

### Scenario principale di successo

#	Attore	Sistema
1	Aggiunge un turno di cucina e, opzionalmente segna un luogo	Registra il turno della cucina.
	<i>Ripete il passo 1 finché non è soddisfatto</i>	
2	Opzionalmente, controlla gli eventi.	Visualizza gli eventi.
	<i>Se non vuole lavorare su nessun evento termina il Caso d'Uso</i>	
3	Sceglie un evento su cui lavorare.	Visualizza l'evento selezionato.
4	Imposta un orario per un turno di servizio e, opzionalmente, mette una scadenza oltre la quale non si può più togliere la disponibilità.	Registra l'orario del turno di servizio e, opzionalmente, la scadenza oltre la quale non si può togliere la disponibilità.
	<i>Se vuole lavorare su altri servizi torna al punto 4</i>	
5	Opzionalmente, modifica la scadenza.	Registra la modifica della scadenza.
6	Opzionalmente, cambia l'orario del turno di servizio.	Registra le modifiche.

	<i>Se vuole lavorare su altri eventi torna al passo 3, altrimenti termina il Caso d'Uso</i>	
--	---	--

## Estensione 1a

#	Attore	Sistema
1a.1	Modifica un turno di cucina.	Registra la modifica del turno di cucina.

## Eccezione 1a

#	Attore	Sistema
1a.1	Aggiunge un turno di cucina e , opzionalmente segna un luogo.	Non si hanno i permessi necessari.
	<i>Termina il Caso d'Uso</i>	

## Eccezione 1b

#	Attore	Sistema
1b.1	Aggiunge un turno di cucina e , opzionalmente segna un luogo.	Esiste già un turno in quel orario.
	<i>Termina il Caso d'Uso</i>	

## Eccezione 1a.1a

#	Attore	Sistema
1a.1a.1	Modifica un turno di cucina.	Non si hanno i permessi necessari.
	<i>Termina il Caso d'Uso</i>	

## Estensione 1b

#	Attore	Sistema
---	--------	---------

1b.1	Cancella un turno di cucina.	Elimina il turno di cucina specificato.
------	------------------------------	---

## Eccezione 1b.1a

#	Attore	Sistema
1b.1a.1	Cancella un turno di cucina.	Non si hanno i permessi necessari.
	<i>Termina il Caso d'Uso</i>	

## Estensione 1c

#	Attore	Sistema
1c.1	Modifica tutti i turni di cucina in un determinato giorno, settimana.	Registra le modifiche di tutti i turni di cucina aventi determinate caratteristiche.

## Eccezione 1c.1a

#	Attore	Sistema
1c.1a.1	Modifica tutti i turni di cucina in un determinato giorno, settimana.	Non si hanno i permessi necessari.
	<i>Termina il Caso d'Uso</i>	

## Estensione 1d

#	Attore	Sistema
1d.1	Elimina tutti i turni di cucina in un determinato giorno, settimana.	Registra l'eliminazione di tutti i turni di cucina aventi determinate caratteristiche.

## Eccezione 1d.1a

#	Attore	Sistema
---	--------	---------

<b>1d.1a.1</b>	Elimina tutti i turni di cucina in un determinato giorno, settimana.	Non si hanno i permessi necessari.
	<i>Termina il Caso d'Uso</i>	

## Estensione 5a

#	Attore	Sistema
<b>5a.1</b>	Rimuove la scadenza.	Toglie la scadenza oltre la quale non si può più rimuovere la disponibilità.

## Eccezione 5a

#	Attore	Sistema
<b>5a.1</b>	Opzionalmente, modifica la scadenza.	Il turno non ha una scadenza impostata.
	<i>Torna al passo 5</i>	

## Eccezione 5a.1a

#	Attore	Sistema
<b>5a.1a</b>	Rimuove la scadenza.	Il turno non ha una scadenza impostata.
	<i>Torna al passo 5</i>	

## Estensione 5b

#	Attore	Sistema
<b>5b.1</b>	Aggiunge una scadenza.	Aggiunge una scadenza oltre la quale non si può più rimuovere la disponibilità.



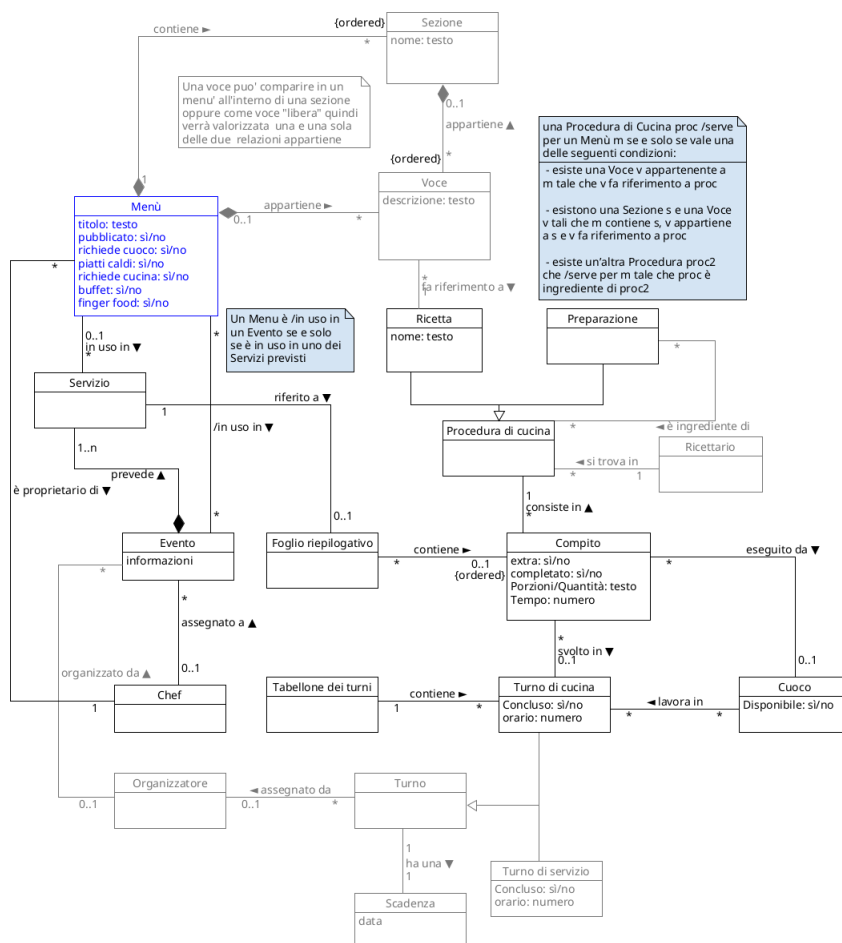
## Estensione 6a

#	Attore	Sistema
6a.1	Cancella un turno di servizio.	Registra la cancellazione del turno di servizio..



# 2

## Modello di Dominio

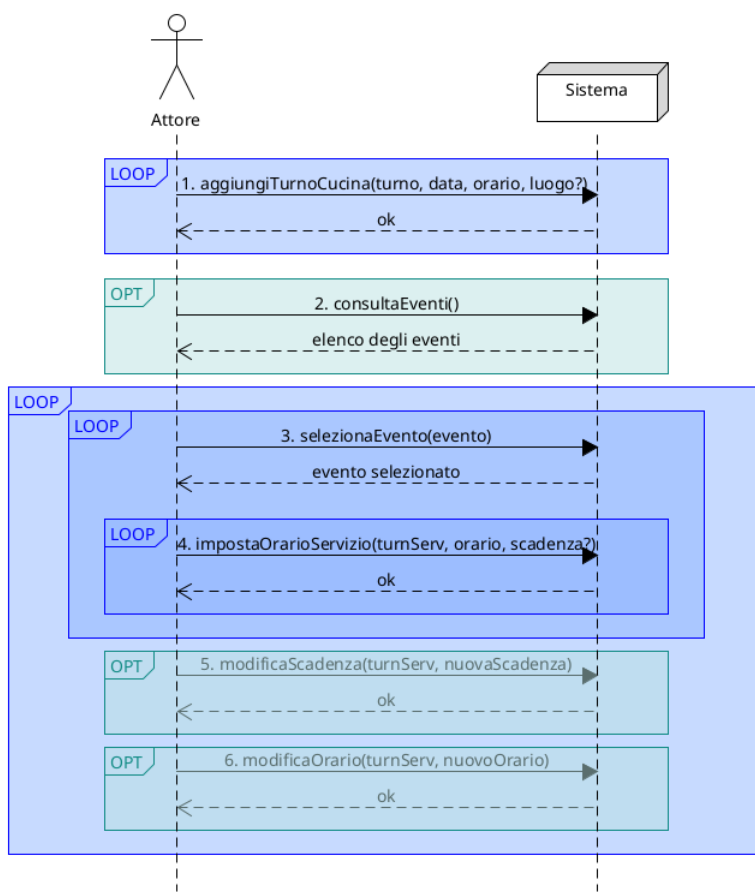




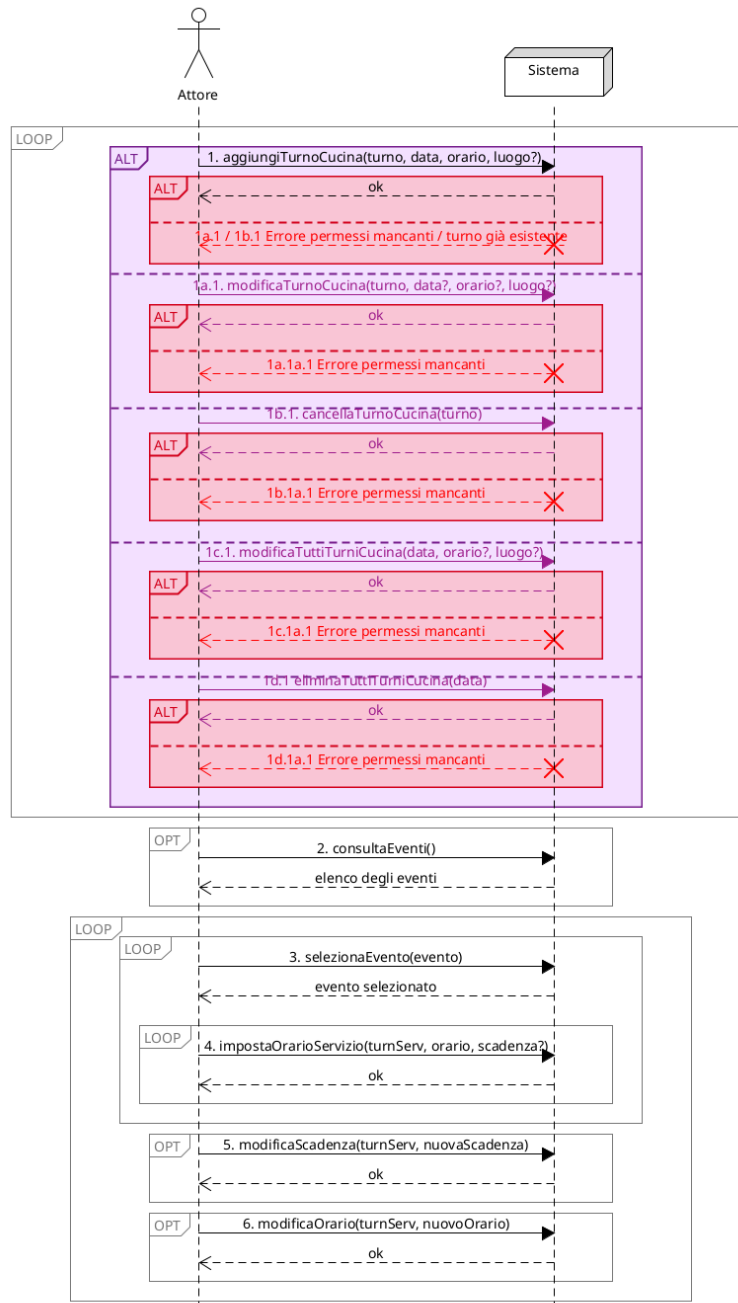
# 3

## SSD - System Sequence Diagram

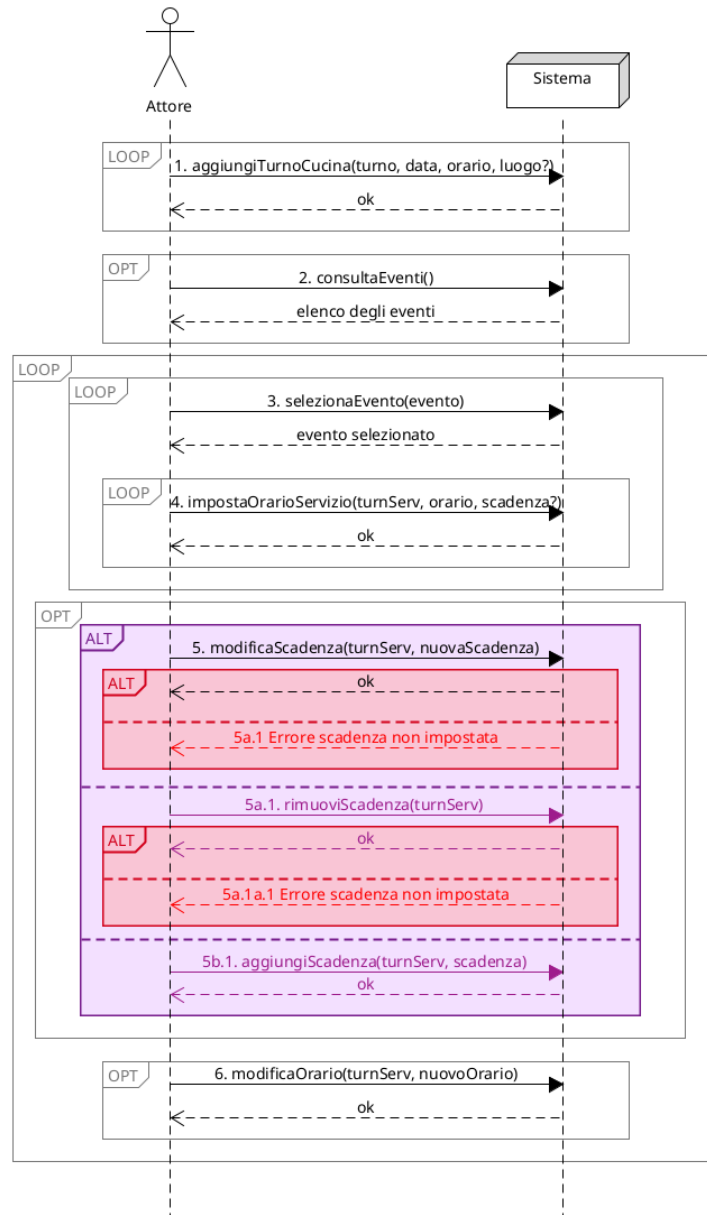
### 3.1 Scenario principale



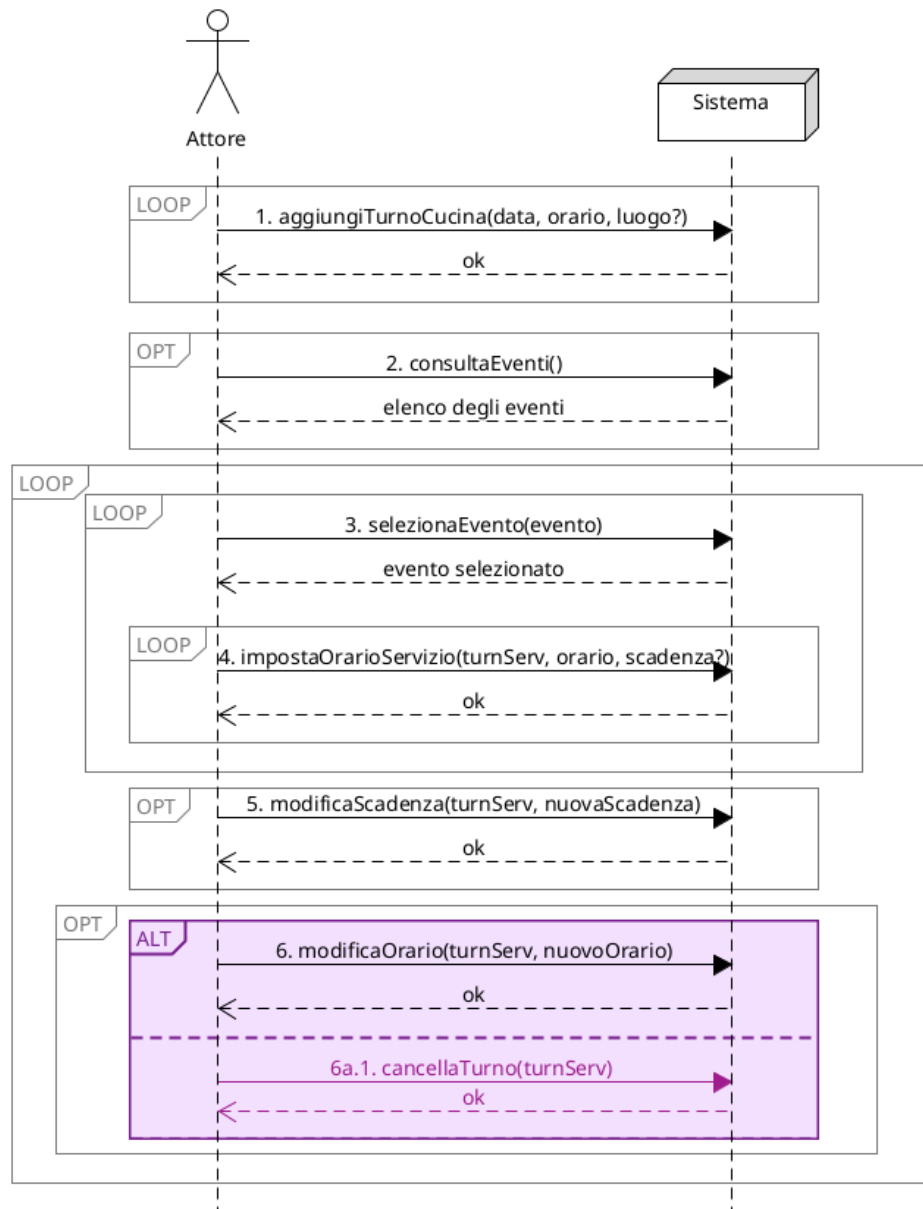
## 3.2 Eccezioni ed estensioni - Passo 1



### 3.3 Eccezioni ed estensioni - Passo 5



### 3.4 Eccezioni ed estensioni - Passo 6







# 4

## Contratti

**Pre-condizione generale:** L'attore è identificato con un'istanza *org* di Organizzatore.

### 4.1 Passo 1

1. aggiungiTurnoCucina(evento: Evento, data: Data, orario: numerico, luogo?: Testo):

**Pre-condizioni:**

- evento è organizzato da *org*;

**Post-condizioni:**

- è stata creata un'istanza *tc* di TurnoCucina;
- *tc* è contenuto in *t* TabelloneDeiTurni;
- turno.data = data;
- turno.orario = orario;
- [ se luogo è specificato ] turno.luogo = luogo;
- turno.concluso = no.

1a.1 modificaTurnoCucina(turno: Turno, data?: Data, orario?: Ora, luogo?: Luogo):

**Pre-condizioni:**

- turno è assegnato a *org*;
- *t* contiene turno;
- [ se orario è specificato ] non esiste un *tc* con lo stesso orario.

**Post-condizioni:**

- [ se data è specificato ] turno.data = data;
- [ se orario è specificato ] turno.orario = orario;
- [ se luogo è specificato ] turno.luogo = luogo.

**1b.1 cancellaTurnoCucina(turno: Turno):****Pre-condizioni:**

- turno è assegnato a *org*;
- *t* contiene turno.

**Post-condizioni:**

- $\langle$  per ogni Compito *c* **presente in** turno  $\rangle$ :
  - l'associazione tra *c* e turno è stata eliminata.
- l'associazione **contiene** tra turno e *t* è stata eliminata;
- l'istanza *tc* di TurnoCucina cessa di esistere.

**1c.1 modificaTuttiTurniCucina(data: Data, orario?: numerico, luogo?: Luogo):****Pre-condizioni:**

- turno è assegnato a *org*;
- *t* contiene turno.

**Post-condizioni:** se turno.data == data

- [ se data è specificato ] turno.data = data;
- [ se orario è specificato ] turno.orario = orario;
- [ se luogo è specificato ] turno.luogo = luogo.

**1d.1 eliminaTuttiTurniCucina(data: Data):****Pre-condizioni:**

- turno è assegnato a *org*;
- *t* contiene turno.

**Post-condizioni:** se turno.data == data

- $\langle$  per ogni Compito *c* **presente in** turno  $\rangle$ :
  - l'associazione tra *c* e turno è stata eliminata.
- l'associazione **contiene** tra turno e *t* è stata eliminata;
- l'istanza *tc* di TurnoCucina cessa di esistere.

## 4.2 Passo 2

### 2. consultaEventi():

Pre-condizioni:

Post-condizioni:

Note:-

consultaEventi() è un'interrogazione di sistema quindi non ha pre e post condizioni.

## 4.3 Passo 3

### 3. selezionaEvento(evento: Evento):

Pre-condizioni:

- evento è organizzato da *org*.

Post-condizioni:

## 4.4 Passo 4

### 4. impostaOrarioServizio(turnoServizio: TurnoServizio, orario: numerico, scadenza?: numerico):

Pre-condizioni:

- evento è assegnato a *org*;
- è in corso la modifica di un Evento e.

Post-condizioni:

- evento.orario = orario;
- [ se scadenza è specificato ] evento.scadenza = scadenza.

## 4.5 Passo 5

### 5. modificaScadenza(turnoServizio: TurnoServizio, nuovaScadenza: numerico):

Pre-condizioni:

- evento è assegnato a *org*;
- è in corso la modifica di un Evento e;

Post-condizioni: se evento.scadenza != 0

- evento.scadenza = nuovaScadenza.

### 5a.1 rimuoviScadenza(turnoServizio: TurnoServizio):

Pre-condizioni:

- turnoServizio è assegnato a *org*;
- è in corso la modifica di un Evento *e*.

Post-condizioni: se evento.scadenza != 0

- evento.scadenza = 0.

### 5b.1 aggiungiScadenza(turnoServizio: TurnoServizio, nuovaScadenza: numerico):

Pre-condizioni:

- turnoServizio è assegnato a *org*;
- è in corso la modifica di un Evento *e*.

Post-condizioni:

- evento.scadenza = nuovaScadenza.

## 4.6 Passo 6

### 6. modificaOrario(turnoServizio: TurnoServizio, nuovoOrario: numerico):

Pre-condizioni:

- turnoServizio è assegnato a *org*;
- è in corso la modifica di un Evento *e*.

Post-condizioni: se evento.scadenza non è stata superata

- evento.orario = nuovoOrario.

### 6a.1 cancellaTurnoServizio(turnoServizio: TurnoServizio):

Pre-condizioni:

- turnoServizio è assegnato a *org*;
- è in corso la modifica di un Evento *e*.

Post-condizioni:

- l'istanza *ts* di TurnoServizio cessa di esistere.



## DCD - Design Class Diagram







# 6

## DSD - Design Sequence Diagram

### 1. addKitchenShift(shift, date, time, location?)

