

Contratti per lo UC “Gestire gli eventi”

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

1. creaEvento(nome: testo, cliente: Cliente, dataInizio: testo, periodicità?: Periodicità):

Pre-condizioni: ---

Post-condizioni: se la dataInizio non è ancora passata:

- viene creata un'istanza di *ec* di EventoCapofila
- *ec.cliente* = cliente
- *ec.nome* = nome
- *ec.periodicità* = periodicità
- è stata creata l'associazione **gestisce** tra *org* e *ec*
- vengono create tante istanze *eventoF* di Evento in base a quanto specificato in periodicità

1a.1 apriEvento(evento: Evento):

Pre-condizioni: ---

Post-condizioni: Se evento.stato = programmato/attivo e *org* **gestisce** evento

- è stata aperta la scheda evento

1a.2 annullaEvento(evento: Evento, opzione: Testo):

Pre-condizioni: evento è **periodizzato** dall' EventoCapofila *ec*

Post-condizioni: se *evento*.stato è attivo:

- *evento*.stato = annullato
- per ogni Servizio *ser* in cui è **organizzato** *evento*:
 - ogni istanza *is* Incarico di Servizio che è **svolto in** *ser* viene eliminata
 - ogni *ic* Incarico di Cucina che **svolge** *c* Compito **organizzato** dall'istanza *foglio* di Foglio Riepilogativo che **riassume** *s*, viene eliminato
- [se opzione = “questo evento e quelli successivi”] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2*.stato = programmato e che *evento*.dataInizio ≤ *ev2*.dataInizio
- [se opzione = “tutti gli eventi”] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2*.stato = programmato

1a.2a.1 terminaEvento():

Pre-condizioni: è in corso la visualizzazione di un Evento *evento*

Post-condizioni: se *evento*.stato è attivo:

- *evento*.stato = terminato

1a.2a.2 aggiungiNote(note: Testo):

Pre-condizioni: è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

Post-condizioni: se *evento.stato* è terminato:

- *ec.note* = note

1a.2b.1 eliminaEvento(opzione: Testo):

Pre-condizioni: è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

Post-condizioni: se *evento.stato* è programmato:

- per ogni Servizio *s* in cui è **organizzato** *evento*:
 - ogni istanza *is* Incarico di Servizio che è **svolto in** *s* viene eliminata
 - il *foglio* Foglio Riepilogativo che **riassume** *s* viene eliminato
 - per ogni istanza Compito *c* che *foglio* **organizza**:
 - ogni *ic* Incarico di Cucina che **svolge** *c* viene eliminato
 - *c* viene eliminato
- *s* viene eliminato
- l'istanza *evento* viene eliminata
- [se opzione = "questo evento e quelli successivi"] le operazioni effettuate sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2.stato* = programmato e che *evento.dataInizio* ≤ *ev2.dataInizio*
- [se opzione = "tutti gli eventi"] le operazioni effettuate sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2.stato* = programmato

1b.1 apreEventoTerminato(evento: Evento):

Pre-condizioni: è in corso la visualizzazione di un Evento *evento*

Post-condizioni: se *evento.stato* è terminato:

- è stata aperta la scheda evento

1b.2 aggiungiNote(note: Testo):

Pre-condizioni: è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

Post-condizioni:

- *ec.note* = note

1c.1 apreEventoAnnullato(evento: Evento):

Pre-condizioni: è in corso la visualizzazione di un Evento *evento*

Post-condizioni: se *evento.stato* è annullato:

- è stata aperta la scheda evento

2. verificaStatoCucina():

Pre-condizioni: ---

Post-condizioni: ---

3. compilaEvento(nome?: Testo, cliente?: Cliente, dataInizio?: Testo):

Pre-condizioni: è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

Post-condizioni: se dataInizio non è ancora passata e se *evento.stato* è programmato:

- [se è specificata dataInizio] *evento.dataInizio* = dataInizio
- [se è specificato il cliente] *ec .cliente* = cliente

3a.1 modificaPeriodicità(evento: Evento, periodicità: Periodicità, opzione: testo):

Pre-condizioni: è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

Post-condizioni:

- *ec.periodicità* = periodicità
- vengono create tante istanze *eventoF* di Evento in base a quanto specificato in periodicità
- [se opzione = "questo evento e quelli successivi"]: tutte le istanze di Evento *eventoObsoleto* che sono ricorrenza di *ec* , che non corrispondono più alla nuova periodicità e hanno *eventoObsoleto.stato*=programmato e *eventoObsoleto.dataInizio*<=*evento.dataInizio* vengono eliminate
- [se opzione = "tutti gli eventi"]: tutte le istanze di Evento *eventoObsoleto* che sono ricorrenza di *ec* , che non corrispondono più alla nuova periodicità e hanno *eventoObsoleto.stato*=programmato vengono eliminate

4. creaServizio(evento: Evento, dataInizio: Testo, orario: Testo, dettagli?: testo, luogo: Testo, n°Commensali: numero, opzione: testo):

Pre-condizioni: è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

Post-condizioni: se dataInizio non è ancora passata:

- Viene creata una nuova istanza *s* di Servizio:
 - *s.dataInizio* = dataInizio
 - *s.orario* = orario
 - [se è specificata dettagli] *s.dettagli* = dettagli
 - *s.luogo* = luogo
 - *s.n°Commensali* = n°Commensali
 - *s.stato* = in preparazione
- Viene creata la relazione è **organizzato** tra *evento* e *s*
- [se opzione = "questo evento e quelli successivi"] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2.stato* = programmato e che *evento.dataInizio* <= *ev2.dataInizio*
- [se opzione = "tutti gli eventi"] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2.stato* = programmato

4a.1 modificaServizio(servizio: Servizio, dataInizio?: Testo, orario?: Testo, dettagli?: testo, luogo?: Testo, n°Commensali?: numero, opzione: testo):

Pre-condizioni: è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

Post-condizioni:

- se dataInizio non è ancora passata, se servizio.stato non è attivo, annullato o terminato:
 - [se è specificata dataInizio] servizio.dataInizio = dataInizio
 - [se è specificata orario] servizio.orario = orario
 - [se è specificata dettagli] servizio.dettagli = dettagli
 - [se è specificata luogo] servizio.luogo = luogo
 - [se è specificata n°Commensali] servizio.n°Commensali = n°Commensali
- se servizio.stato è attivo:
 - se non è specificato nessun parametro opzionale eccetto n°Commensali e n°Commensali < (servizio.n°Commensali + 30%) e n°Commensali > (servizio.n°Commensali - 30%):
 - servizio.n°Commensali = n°Commensali
 - [se opzione = "questo evento e quelli successivi"] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2.stato* = programmato e che *evento.dataInizio* <= *ev2.dataInizio*
 - [se opzione = "tutti gli eventi"] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2.stato* = programmato

4b.1 annullaServizio(evento: Evento, servizio: Servizio, opzione: testo):

Pre-condizioni: è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

Post-condizioni: se servizio.stato non è annullato o terminato:

- servizio.stato = annullato
- per ogni istanza *com* Compito **organizzato** in *foglio* Foglio Riepilogativo che **riassume** servizio:
 - ogni istanza *ic* Incarico di Cucina che **svolge** *com* viene eliminata
- ogni istanza *is* Incarico di Servizio che **svolge** servizio viene eliminata
- [se opzione = "questo evento e quelli successivi"] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da ec**, che abbia *ev2*.stato = programmato e che *evento*.dataInizio <= *ev2*.dataInizio
- [se opzione = "tutti gli eventi"] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da ec**, che abbia *ev2*.stato = programmato

4c.1 eliminaServizio(evento: Evento, servizio: Servizio, opzione: testo):

Pre-condizioni: è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

Post-condizioni: se servizio.stato non è annullato o terminato o attivo:

- ogni istanza *is* Incarico di Servizio che **svolge** servizio viene eliminata
- per ogni istanza *foglio* Foglio Riepilogativo che **riassume** servizio:
 - per ogni istanza *c* Compito **organizzato** da *foglio*:
 - ogni *ic* Incarico di Cucina che **svolge** *c* viene eliminato
 - *c* viene eliminato
- *foglio* viene eliminato
- servizio viene eliminato
- [se opzione = "questo evento e quelli successivi"] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da ec**, che abbia *ev2*.stato = programmato e che *evento*.dataInizio <= *ev2*.dataInizio
- [se opzione = "tutti gli eventi"] l'operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da ec**, che abbia *ev2*.stato = programmato

5. sceltaChef(chef: Chef, opzione: testo):

Pre-condizioni: è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall' EventoCapofila *ec*

Post-condizioni:

- chef lavora a un *evento*
- [se opzione = “questo evento e quelli successivi”] l’operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2.stato* = programmato e che *evento.dataInizio* <= *ev2.dataInizio*
- [se opzione = “tutti gli eventi”] l’operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2.stato* = programmato

6. confermaMenù(evento: Evento, servizio: Servizio):

Pre-condizioni: : è in corso la visualizzazione di un Evento *evento* che è **suddiviso in** servizio

Post-condizioni: se servizio.stato non è annullato o terminato o attivo e un Menu è **scelto per** servizio:

- servizio.stato = attivo
- *evento.stato* = attivo

6a.1 modificaMenù(evento: Evento, servizio: Servizio, menu: Menu, proposta: Proposta di Modifica, opzione: testo):

Pre-condizioni: : è in corso la visualizzazione di un Evento *evento* che è **periodizzato** dall’ EventoCapofila *ec*

Post-condizioni: se servizio.stato non è annullato o terminato o attivo e un Menu è **scelto per** servizio:

- Viene creata l’associazione **propone** tra proposta e servizio
- [se opzione = “questo evento e quelli successivi”] l’operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2.stato* = programmato e che *evento.dataInizio* <= *ev2.dataInizio*
- [se opzione = “tutti gli eventi”] l’operazione effettuata sopra è applicata ad ogni Evento *ev2* che è **periodizzato da** *ec*, che abbia *ev2.stato* = programmato

7. sceltaPersonale(personale: Personale, turnoServizio: TurnoServizio, ruolo?: Ruolo):

Pre-condizioni: : è in corso la visualizzazione di un Evento *evento* che è **suddiviso in** servizio Servizio

Post-condizioni: se *servizio.stato* non è annullato o terminato:

- viene creato un *is* Incarico di Servizio e:

- [se ruolo è specificato] *is.ruolo* = ruolo
- viene creata l'associazione **si svolge in** tra *is* e turnoServizio
- viene creata l'associazione **serve in** tra personale *is*
- viene creata l'associazione **è svolto in** tra *is* e *servizio*

7a.1 rimuoviPersonale(personale: Personale, turnoServizio: TurnoServizio):

Pre-condizioni: : è in corso la visualizzazione di un Evento *evento* che è **suddiviso** in *servizio* Servizio

Post-condizioni: se *servizio.stato* non è annullato o terminato:

- per ogni *is* Incarico di Servizio **deciso per servizio**:
 - viene eliminata ogni istanza di personale che **serve in** *is*