# **Contratti per lo UC “Gestire Eventi”**

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

**1. creaEvento(titolo: testo,user:User data: date, n\_part?: intero, luogo?: testo)**

**Pre-condizioni:**

**Post-condizioni:**

* è stata creata un’istanza *e* di **Evento**
* *e.titolo* = titolo
* *e.data* = data
* [se è specificato n\_part ] *e.n\_part* = n\_part
* [se è specificato un luogo ] *e.luogo* = luogo
* org è **proprietario di** *e*
* *e.terminato =* no

**1a apriEvento(evento: Evento)**

**Pre-condizioni:**

* esiste un istanza e di **Evento** *e org* deve essere il **proprietario** di e

**Post-condizioni:**

**2. assegnaChef(chef: Chef)**

**Pre-condizioni:**

* è in corso la definizione di un **Evento** *e*
* *org* deve essere il **proprietario di** *e*
* *e* non deve essere ancora stato **assegnato ad** uno Chef ch

**Post-condizioni:**

* *e* **è** assegnato allo Chef ch

**2a.1 rimuoveChef(chef: Chef)**

**Pre-condizioni:**

* è in corso la definizione di un **Evento** *e*
* *org* deve essere il **proprietario** di *e*
* *e* è **assegnato** a chef ch

**Post-condizioni:**

* lo Chef ch non è più incaricato **dell’Evento** e

**3. assegnaPersonale(personale: Personale, ruolo?: testo)**

**Pre-condizioni:**

* è in corso la definizione di un **Evento** *e*
* *org* deve essere il **proprietario** di *e*

**Post-condizioni:**

* [se è specificato un ruolo] personale.ruolo = ruolo
* personale è **assegnato** a *e*

**3a rimuovePersonale(personale: Personale)**

**Pre-condizioni:**

* è in corso la definizione di un **Evento** *e*
* *org* deve essere il **proprietario** di *e*
* personale è **assegnato** ad *e*

**Post-condizioni:**

* È stata rimossa **l’assegnazione** di personale ad *e*

**3b modificaRuolo(personale: Personale, ruolo: testo)**

**Pre-condizioni:**

* è in corso la definizione di un **Evento** *e*
* *org* deve essere il **proprietario** di *e*

**Post-condizioni:**

* personale.ruolo = ruolo

## **4. modificaDatiEvento(titolo?: testo, data?: date, n\_part?: intero, luogo?: testo, service\_type?: testo )**

**Pre-condizioni:**

* è in corso la definizione di un **Evento** *e*
* *org* deve essere il **proprietario di** *e*

**Post-condizioni:**

* [se è specificato un titolo ] *e.titolo*= titolo
* [se è specificato una data ] *e.data* = data
* [se è specificato n\_part ] *e.n\_part* = n\_part
* [se è specificato un luogo ] *e.luogo* = luogo
* [se è specificato il tipo di servizio] e.service\_type= service\_type

**4a eliminaEvento(evento: Evento)**

**Pre-condizioni:**

* è in corso la gestione di un **Evento** e
* *org* deve essere il **proprietario** di **Evento**

**Post-condizioni:**

* *evento.terminato* = si

**4b terminaEvento(evento: Evento)**

**Pre-condizioni:**

* è in corso la gestione di un **Evento** e
* *org* deve essere il **proprietario** di **Evento**

**Post-condizioni:**

- e.stato = “annullato”

- è eliminato ogni Personale pers incaricato\_di evento

- è eliminato Chef ch incaricato\_di evento

**5. consultaMenù(chef: Chef, menu:Menu)**

**Pre-condizioni:**

* è in corso la gestione dell’ **Evento** e tale che *org* **organizza** e
* chef è incaricato di e ed chef è **proprietario** di menu**Post-condizioni:**

**5a. confermaMenù(chef:Chef, menu:Menu)**

**Pre-condizioni:**

* è in corso la definizione di un **Evento** *e*
* *org* deve essere il **proprietario di** *e*
* *Chef* è incaricato dell’**Evento** *e* ed è proprietario di *menù*

**Post-condizioni:**

* *e.status= “in corso”*
* *menù* in uso in *e*

**5b. rifiutaMenù(chef:Chef, menu:Menu)**

**Pre-condizioni:**

* è in corso la definizione di un **Evento** *e*
* *org* deve essere il **proprietario di** *e*
* *Chef* è incaricato dell’evento *e* ed è proprietario di *menù*

**Post-condizioni:**

* eliminato menù in uso