

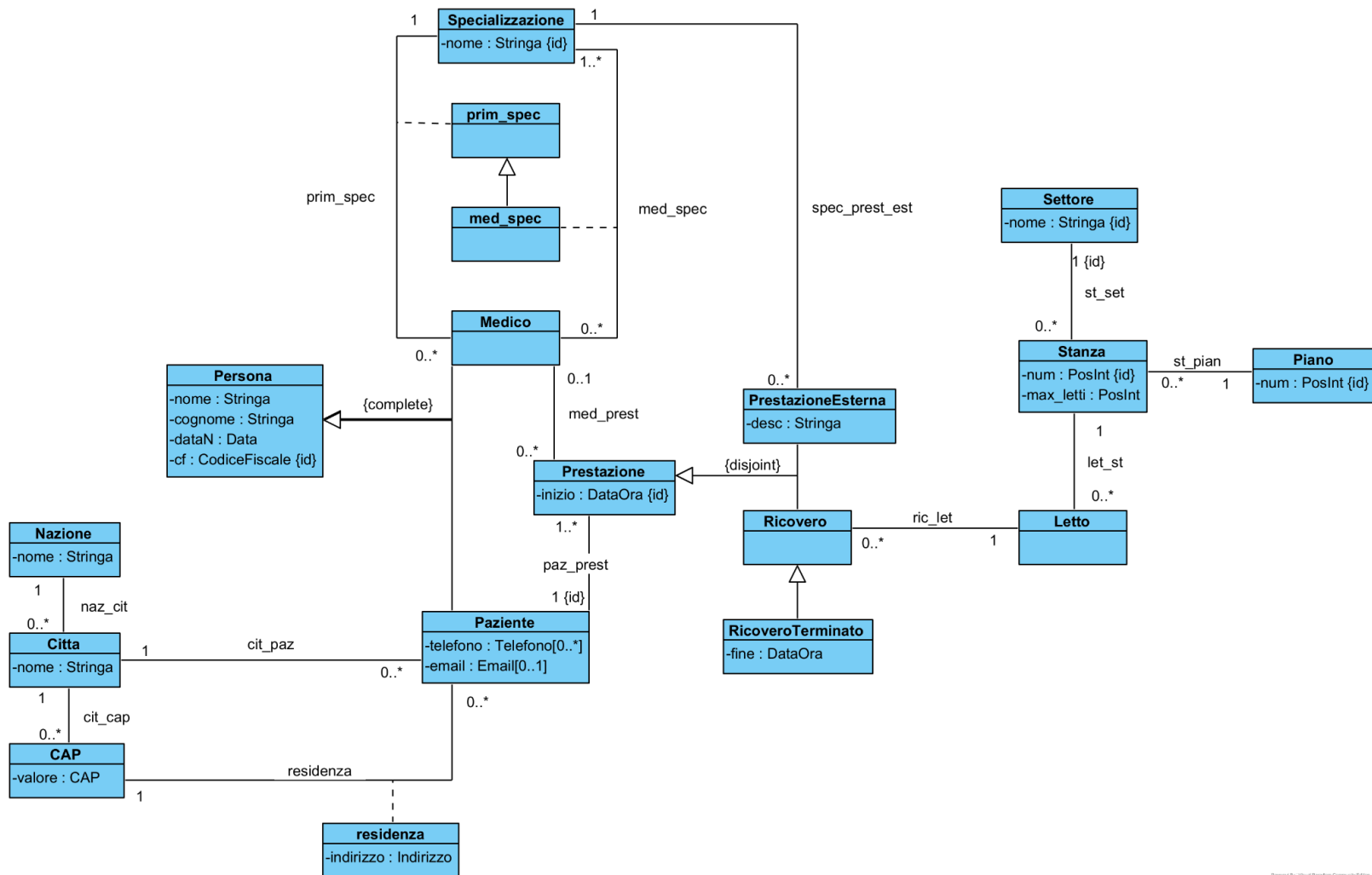
# Progetto QuickHospital

## raffinamento dati

1. Requisiti sulle persone di interesse. Di ogni persona di interessa bisogna che il sistema rappresenti:
  - 1.1. nome (stringa)
  - 1.2. cognome (stringa)
  - 1.3. data di nascita (data)
  - 1.4. codice fiscale
2. Requisiti sui pazienti. Di ogni paziente interessa:
  - 2.1. i dati anagrafici come da req. 1
  - 2.2. recapiti:
    - 2.2.1. numeri di telefono (zero o più)
    - 2.2.2. email (al più una)
    - 2.2.3. indirizzo (esattamente uno)
3. Requisiti sui medici. Di ogni medico interessa:
  - 3.1. i dati anagrafici come da req. 1
  - 3.2. specializzazione primaria (v.req. 8, esattamente una)
  - 3.3. specializzazioni secondarie (v. req. 8, zero o più)
4. Requisiti sui ricoveri. Di ogni ricovero interessa:
  - 4.1. il paziente (v. req. 2)
  - 4.2. data inizio
  - 4.3. data di fine (opzionale)
  - 4.4. il posto letto (v. req. 5.3)
  - 4.5. il medico responsabile (v. req. 3)
5. Requisiti sulle stanze. Di ogni stanza interessa:
  - 5.1. piano (intero > 0)
  - 5.2. settore (intero > 0)
  - 5.3. i posti letto (v. req. 6)
6. Requisiti sui posti letto. Di ogni posto letto interessa:
  - 6.1. la stanza (v. req. 5)
  - 6.2. il numero d'ordine nella sua stanza (intero > 0, univoco)
  - 6.3. in ogni istante di tempo, un letto può essere associato ad al massimo un ricovero (v. req. 4)

7. Requisiti sulle prestazioni esterne. Di ogni prestazione esterna interessa:
  - 7.1. data
  - 7.2. paziente (v. req. 2)
  - 7.3. specializzazione
  - 7.4. descrizione
8. Requisiti sulle specializzazioni
  - 8.1. nome (univoco)
9. Requisiti sulle funzionalità da offrire:
  - 9.1. Il persona di accettazione e quello di prenotazione delle prestazioni esterni deve poter:
    - 9.1.1. registrare un nuovo paziente, fornendo i dati di cui al req. 2
  - 9.2. Il personale di accettazione deve poter:
    - 9.2.1. registrare un ricovero, fornendo i dati di cui ai req. 4.1 (paziente già registrato a sistema), 4.2,
    - 9.2.2. registrare il termine di un dato ricovero (non terminato) fornendo la data di fine (req. 4.3)
  - 9.3. Il personale dell'ufficio prenotazioni deve poter:
    - 9.3.1. registrare una nuova prenotazione di prestazione esterna, fornendo i dati di cui al req. 7
  - 9.4. Il personale amministrativo deve poter:
    - 9.4.1. calcolare, data una prestazione esterna per la specializzazione 's', l'insieme dei medici maggiormente adatti a soddisfare: se esistono medici con specializzazione primaria pari ad 's', il risultato è l'insieme di tali medici. Altrimenti, il risultato è l'insieme dei medici che hanno 's' tra le loro specializzazioni secondarie.
    - 9.4.2. assegnare un medico ad una prestazione esterna
    - 9.4.3. assegnare un medico ad un ricovero
  - 9.5. I medici devono poter:
    - 9.5.1. calcolare il loro itinerario di visite, ovvero un insieme ordinato delle stanze associate ai ricoveri di cui è responsabile (v. req. 4.5). L'ordinamento è dato in primo luogo dal piano delle stanze ed in secondo luogo dal settore (entrambi in ordine crescente).

# Diagramma UML di classe



# Specifica dei tipi di dato

## Primitivi

- **CodiceFiscale:**
  - Stringa che rispetta la seguente espressione regolare:
    - `^[A-Z]{6}[0-9]{2}[A-Z][0-9]{2}[A-Z][0-9]{3}[A-Z]$`
- **Telefono:**
  - Stringa che rispetta la seguente espressione regolare:
    - `^(\+\\d{1,2}\\s)?\\(?\\d{3}\\)?[\\s-]\\d{3}[\\s-]\\d{4}$`
- **Email:**
  - Stringa che rispetta la seguente espressione regolare:
    - `^[a-zA-Z0-9_+.-]+@[a-zA-Z0-9-]+.[a-zA-Z0-9-]+$`
- **CAP:**
  - Stringa che rispetta la seguente espressione regolare:
    - `^\\d{5}$`
- **Civico:**
  - Stringa che rispetta la seguente espressione regolare:
    - `^\\d{1,3}[a-zA-Z]?$`
- **Indirizzo:**
  - Tipo di dato composto da:
    - via: Stringa
    - civico: Civico
- **PosInt:**
  - Numero intero maggiore di zero (Int > 0)

## Vincoli Esterni

### [V.Prestazione.med-paz]

- Per ogni istanza  $p$  dell'entità *Prestazione* deve essere vero che nel caso in cui  $m:Medico$  tale che  $(m, p):med\_prest$  sia anche istanza di *Paziente*, non esista  $(m, p):paz\_prest$

### [V.RicoveroTerminato.fine]

- Per ogni istanza  $r$  dell'entità *RicoveroTerminato*, il valore per l'attributo *fine* deve essere maggiore del valore dell'attributo *inizio*.

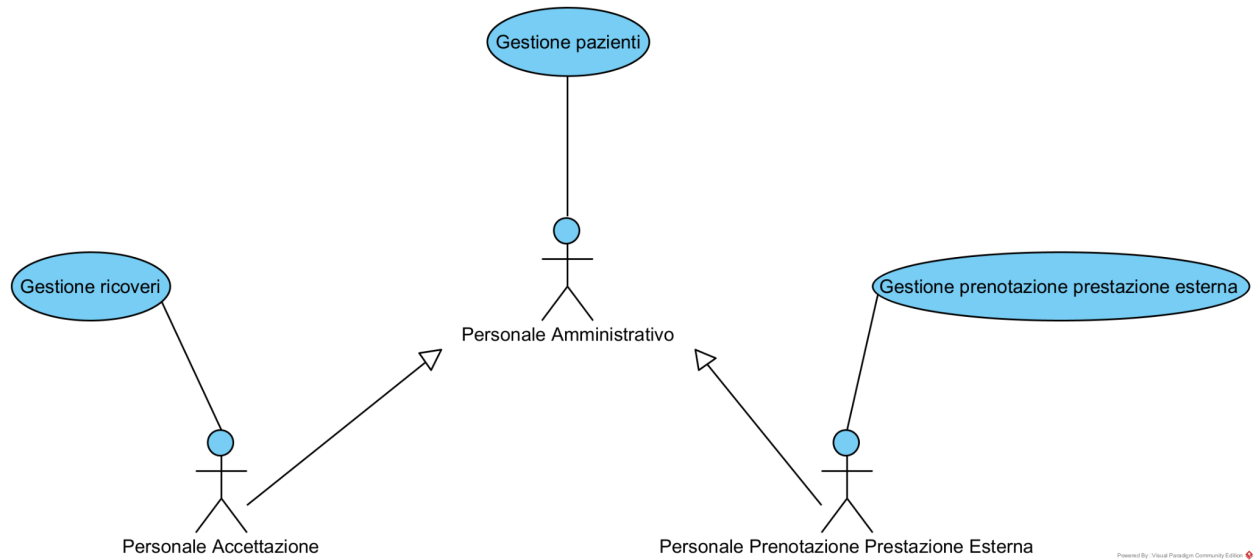
### [V.Letto.occupato]

- Non deve esistere un'istanza  $l$  dell'entità *Letto* coinvolta in due istanze  $(l, r_1): ric\_let$  e  $(l, r_2): ric\_let$  che si sovrappongono nel tempo.

### [V.Stanza.numletti]

- Ogni istanza dell'entità *Stanza* non può essere coinvolta in più link *let\_st* di quanti sono ammessi dal valore del suo attributo *max\_letti*.

# Diagramma UML degli Use-Case



## Specifica Use-Case ***Gestione pazienti***

**registra\_nuovo\_paziente**(cf:CodiceFiscale, n: Stringa, c:Stringa, nasc:Data, tel:Telefono [0..\*], em:Email [0..1], ind:Indirizzo, cap:CAP, citta:Stringa, nazione:Stringa): Paziente[0..1]

pre-condizioni:

**[Codice Fiscale univoco]:**

- Non esiste alcuna istanza  $p$  di *Persona* dove  $p.cf = cf$

post-condizioni:

- L'operazione non modifica il livello estensionale (oggetti e link).
- Il valore del risultato *result* è definito come segue:
  - Sia  $p$  l'istanza dell'entità *Paziente* dove:
    - $p.nome = n$
    - $p.cognome = c$
    - $p.dataN = nasc$
    - $p.cf = cf$
    - $p.telefono = tel$  **se definito**
    - $p.email = em$  **se definito**
  - Sia  $n$  l'istanza dell'entità *Nazione* dove:
    - $n.nome = nazione$  se non esiste istanza  $n_2:Nazione$  con valore dell'attributo *nome* = *nazione* altrimenti  $n = n_2$
  - Sia  $cit$  l'istanza dell'entità *Citta* dove:
    - $cit.nome = citta$  se non esiste istanza  $cit_2:Citta$  con valore dell'attributo *nome* = *citta* altrimenti  $cit = cit_2$
    - $(cit, n):naz\_cit$
    - $(cit, p):cit\_paz$
  - Sia  $c$  l'istanza dell'entità *CAP* dove:
    - $c.valore = cap$  se non esiste istanza  $c_2:CAP$  con valore dell'attributo *valore* = *cap* altrimenti  $c = c_2$
    - $(c, p):residenza$  dove  $r:residenza$ :
      - $r.indirizzo = ind$
    - $(c, cit):cit\_cap$
  - Sia  $result = p$

## Specifica Use-Case **Gestione ricoveri**

**registra\_ricovero(paz: Paziente): Ricovero[0..1]**

pre-condizioni: *nessuna*

post-condizioni:

- L'operazione non modifica il livello estensionale (oggetti e link).
- Il valore del risultato *result* è definito come segue:
  - Sia *l* un'istanza dell'entità *Letto* dove:
    - *l* = *letto.postoLibero()* ####Ritorna un'istanza di *Letto* che non sia occupato da un altro paziente####
  - Sia *r* un'istanza dell'entità *Ricovero* dove:
    - *r.inizio* = *adesso*
    - *(r, paz):paz\_prest*
    - *(r, l): ric\_let*
  - Sia *result* = *r*

**registra\_fine\_ricovero(paz: Paziente): RicoveroTerminato[0..1]**

pre-condizioni:

**[paziente deve essere in ricovero]:**

- *(paz, r:Ricovero): paz\_prest*

post-condizioni:

- L'operazione non modifica il livello estensionale (oggetti e link).
- Il valore del risultato *result* è definito come segue:
  - Sia *r* un'istanza dell'entità *Ricovero* tale che *(paz, r):paz\_prest*
  - Sia *r<sub>term</sub>* un'istanza dell'entità *RicoveroTerminato* dove:
    - *r<sub>term</sub>* = *r*
    - *r<sub>term</sub>.fine* = *adesso*
  - Sia *result* = *r*



## Specifica Use-Case ***Gestione pren. prestazione esterna***

**registra\_prestazione\_esterna**(paz:Paziente, d: DataOra, desc: Stringa, spec: Specializzazione): PrestazioneEsterna[0..1]

pre-condizioni: *nessuna*

post-condizioni:

- L'operazione non modifica il livello estensionale (oggetti e link).
- Il valore del risultato *result* è definito come segue:
  - Sia *p* un'istanza dell'entità *PrestazioneEsterna*:
    - *p.inizio* = *d*
    - *p.desc* = *desc*
    - (*p*, *paz*):*paz\_prest*
    - (*p*, *spec*): *spec\_prest\_est*
  - Sia *result* = *p*