

Use Case GestioneRicoveri:

```
senza ordinamento (operazione ausiliaria)

ottiene_itinerario(m: Medico, d: dataOra): Stanza[0..*]
    precondizioni:
        m.nascita < d
    postcondizioni:
        l'operazione non modifica i dati
        result è uguale all'insieme S degli oggetti s:stanza
        tale che esistono
        - un link (l,s):stan_let,
        - (l,r):ric_let,
        - (this,r):med_ric
        e d sia compreso in [r.inizio, r.fine]
```

Letto:

```
Verifica_disponibilità(d: dataOra): -> Booleano
    precondizioni:
        nessuna
    postcondizioni:
        l'operazione non modifica i dati
        sia result uguale a True
        se this non è coinvolto in un altro link ric_let che
coinvolge
            un r: Ricovero tale che d è compreso nel periodo
[r.inizio, r.fine]

            oppure, più precisamente:
            un r:Ricovero tale che d >= r.inizio e, se r.fine ha un
valore, d <= r.fine

        altrimenti result = Flase

(caso ricovero: sovrapponi)
```

Use-Case GestioneLetti:

```
operazione letti_liberi(i: DataOra): Letto [0..*]
    precondizioni:
        nessuna
    postcondizioni:
        l'operazione non modifica i dati
        Il risultato result è l'insieme L degli oggetti l:Letto tale
che l.Verifica_disponibilità(i) = True
```

```
Use-Case Assisti_richiesta(d: dataOra, s: Specializzazione): n Medico
[0..*]
    precondizioni:
        nessuna
    postcondizioni:
        l'operazione non modifica i dati
```

sia P l'insieme di m:Medico tale che
m non sia coinvolto in link (r,m):med_ric o (p1,m):med_prest
durante [r.inizio,r.fine] o p1.data_richiesta
e (m,s):med_sec o (m,s):med_prim
tali che (m,s):med_prim .
Se P non è vuoto, result = P.

Se P è vuoto,
sia S l'insieme dei m:Medico tali che (m, s):med_sec
result = S