

QuickHospital

Indice

1	Specifica	2
2	Analisi	3
2.1	Diagramma delle classi	3
2.2	Dizionario dei dati	4
2.3	Vincoli	5
2.4	Usecases	6
2.4.1	Specifica Use-Case Itinerario	6
2.4.2	Specifica dello Use-Case RegistraPazienti	7
2.4.3	Specifica dello Use-Case AccettaRicoveri	9
3	Usecase	10
3.1	Diagramma degli usecase	10
4	Ristrutturazione	11
4.1	Diagramma delle classi ristrutturato	11
4.2	Dizionario dei dati ristrutturato	12
4.3	Vincoli ristrutturati	13
4.3.1	Vincoli Esterni Aggiunti nella Fase di Ristrutturazione	13
4.3.2	Vincoli Esterni Rimasti Invariati dopo la Fase di Ristrutturazione	13
4.3.3	Vincoli Esterni Modificati durante la Fase di Ristrutturazione	13
4.4	Usecases ristrutturati	15
4.4.1	Specifica Use-Case Itinerario	15
4.4.2	Specifica dello Use-Case RegistraPazienti	16
4.4.3	Specifica dello Use-Case AccettaRicoveri	18
4.5	Progettazione della base di dati	18

1 Specifica

1. persona
 - 1.1. nome
 - 1.2. cognome
 - 1.3. data nascita
2. medico
 - 2.1. pazienti che hanno in cura
 - 2.2. specializzazione primaria
 - 2.3. specializzazioni secondarie
3. paziente
 - 3.1. recapiti telefonici
 - 3.2. email (unica)
 - 3.3. recapito postale (unico)
4. stanza
 - 4.1. i posti letto (da 1 a 8)
 - 4.2. piano $\text{int} > 0$
 - 4.3. settore $\text{int} > 0$
5. posto letto
6. ricovero
 - 6.1. paziente
 - 6.2. posto letto
 - 6.3. dataora
7. dimissione
 - 7.1. dataora dimissione
8. specializzazione
 - 8.1. nome
9. prestazione
 - 9.1. data richiesta
 - 9.2. descrizione
 - 9.3. medico che la eroga
 - 9.4. specializzazione richiesta

2.1 Diagramma delle classi

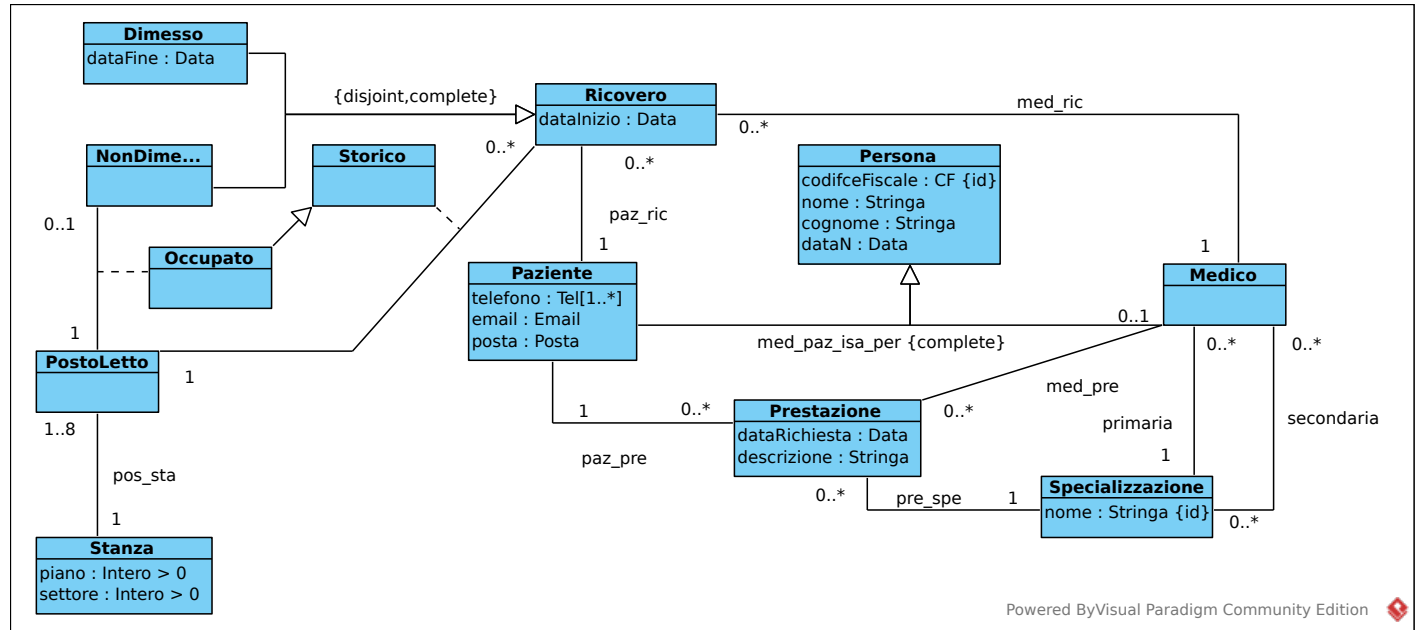


Figure 1: Analisi

2.2 Dizionario dei dati

- tel = stringa secondo standard
- cf = stringa alfanumerica secondo standard
- email = stringa secondo standard
- posta = stringa secondo standard

2.3 Vincoli

- **[V.Prestazione.medico_giusto]**

Ogni medico associato a una prestazione ha la specializzazione richiesta dalla prestazione:

$$\forall m, p, s (med_pre(m, p) \wedge pre_spe(p, s) \rightarrow (primaria(m, s) \vee secondaria(m, s)))$$

- **[V.Medico.secondaria_non_primaria]**

Se un medico è associato a una specializzazione secondaria, non può essere associato alla stessa specializzazione come primaria:

$$\forall s, m (secondaria(m, s) \rightarrow \neg primaria(m, s))$$

- **[V.Dimesso.date_consistente]**

Se un paziente è dimesso, la data di inizio del ricovero deve essere precedente o uguale alla data di fine:

$$\forall r (Dimesso(r) \rightarrow \forall d_1, d_2 (dataInizio(r, d_1) \wedge dataFine(r, d_2) \rightarrow d_1 \leq d_2))$$

- **[V.Ricovero.date_consistente]**

La data di inizio del ricovero deve essere successiva o uguale alla data di nascita del paziente:

$$\forall r, dr, p, dp (paz_ric(p, r) \wedge dataInizio(r, dr) \wedge dataN(p, dp) \rightarrow dr \geq dp)$$

- **[V.Prestazione.date_consistente]**

La data di richiesta della prestazione deve essere successiva o uguale alla data di nascita del paziente:

$$\forall paz, pre, d1, d2 (paz_pre(paz, pre) \wedge dataRichiesta(pre, d1) \wedge dataN(paz, d2) \rightarrow d1 \geq d2)$$

- **[V.Medico.date_consistente]**

La data di inizio del ricovero deve essere successiva alla data di nascita del medico:

$$\forall m, r, d1, d2 (med_ric(m, r) \wedge dataInizio(r, d1) \wedge dataN(m, d2) \rightarrow d1 > d2)$$

- **[V.Paziente.ricovero_singolo]**

Un paziente può avere un solo ricovero non dimesso alla volta:

$$\forall p (\exists r (paz_ric(p, r) \wedge NonDimesso(r)) \rightarrow \neg \exists r' (r' \neq r \wedge NonDimesso(r') \wedge paz_ric(p, r')))$$

- **[V.Paziente.non_ricoverato_da_se_stesso]**

Un paziente non può essere ricoverato da se stesso:

$$\forall p, r (paz_ric(p, r) \rightarrow \neg med_ric(p, r))$$

2.4 Usecases

2.4.1 Specifica Use-Case Itinerario

itinerario(): Stanza [0..*]

Pre-condizioni: nessuna

Post-condizioni:

- Non modifica lo spazio estensionale.
- Sia $\text{Medico}(m)$ il medico che si è autenticato nel sistema e che sta usando lo use-case.
- Sia $P = \{p \mid \exists r (paz_ric(p, r) \wedge med_ric(m, r) \wedge NonDimesso(r))\}$.
- Sia $L = \{l \mid \exists n (Occupato(n, l) \wedge n \in P)\}$.
- Sia $S = \{s \mid \exists l (pos_sta(l, s) \wedge l \in L)\}$.
- $result = S$ (poi in SQL verrà ordinato).

2.4.2 Specifica dello Use-Case RegistraPazienti

registrazionePersona(codiceFiscale: CF, nome: Stringa, cognome: Stringa, dataN: Data): Persona

Pre-condizioni:

$$\forall p1 \ (Persona(p1) \rightarrow \neg codiceFiscale(p1, codiceFiscale))$$

Post-condizioni:

- Modifica dello spazio estensionale:
- Elementi del dominio di interpretazione: α
- Nuove ennuple:
 - `Persona`(α)
 - `codiceFiscale`(α , `codiceFiscale`)
 - `nome`(α , `nome`)
 - `cognome`(α , `cognome`)
 - `dataN`(α , `dataN`)
- `result = Persona`

registrazionePaziente(p: Persona, telefoni: Tel [1..*], email: Email, posta: Posta): Paziente

Pre-condizioni:

$$\neg Paziente(p)$$

Post-condizioni:

- Modifica dello spazio estensionale:
- Nuove ennuple:
 - `Paziente`(p)
 - `telefono`(p , `telefoni`)
 - `email`(p , `email`)
 - `posta`(p , `posta`)
- `result = Paziente`

possibiliMedici(s: Specializzazione): Medico [0..*]

Pre-condizioni: nessuna

Post-condizioni:

- Non modifica lo spazio estensionale.
- Sia $P = \{m \mid primaria(m, s)\}$.
- Sia $S = \{m \mid secondaria(m, s)\}$.
- $|P| > 0 \rightarrow result = P$
 $|P| = 0 \rightarrow result = S$

accettaPrestazione(p: Paziente, s: Specializzazione, dataRichiesta: Data, descrizione: Stringa): Prestazione
Pre-condizioni:

$$\forall M (possibiliMedici_s(s, M) \rightarrow |M| > 0 \wedge (\forall d (data(adesso, d) \rightarrow dataRichiesta \geq d)))$$

Post-condizioni:

- Modifica dello spazio estensionale:
- Elementi del dominio di interpretazione: α
- Nuove ennuple:
 - `Prestazione`(α)
 - `dataRichiesta`(α , `dataRichiesta`)
 - `descrizione`(α , `descrizione`)
 - `paz_pre`(p , α)
 - `pre_spe`(α , s)
- `result` = α

2.4.3 Specifica dello Use-Case AccettaRicoveri

postiLettoLiberi(): PostoLetto[0..*]

Pre-condizioni: nessuna

Post-condizioni:

- Non viene modificato lo spazio estensionale dei dati.
- $L = \{l \mid \neg \exists n \text{ Occupato}(l, n)\}$.
- $\text{result} = L$

accettaRicoveri(p: Paziente): NonDimesso

Pre-condizioni:

$$(\neg \exists r (\text{paz_ric}(p, r) \wedge \text{NonDimesso}(r))) \wedge (\forall L (\text{postiLettoLiberi}(L) \rightarrow |L| > 0))$$

Post-condizioni:

- Modifica dello spazio estensionale:
- Sia l un elemento a caso in $\{l \mid \neg \exists n \text{ Occupato}(l, n)\}$.
- Elementi del dominio di interpretazione: α
- Nuove ennuple:
 - $\text{NonDimesso}(\alpha)$
 - $\text{Ricovero}(\alpha)$
 - $\text{Storico}(\alpha, l)$
 - $\text{pos_sta}(\alpha, l)$
- $\text{result} = \alpha$

3 Usecase

3.1 Diagramma degli usecase

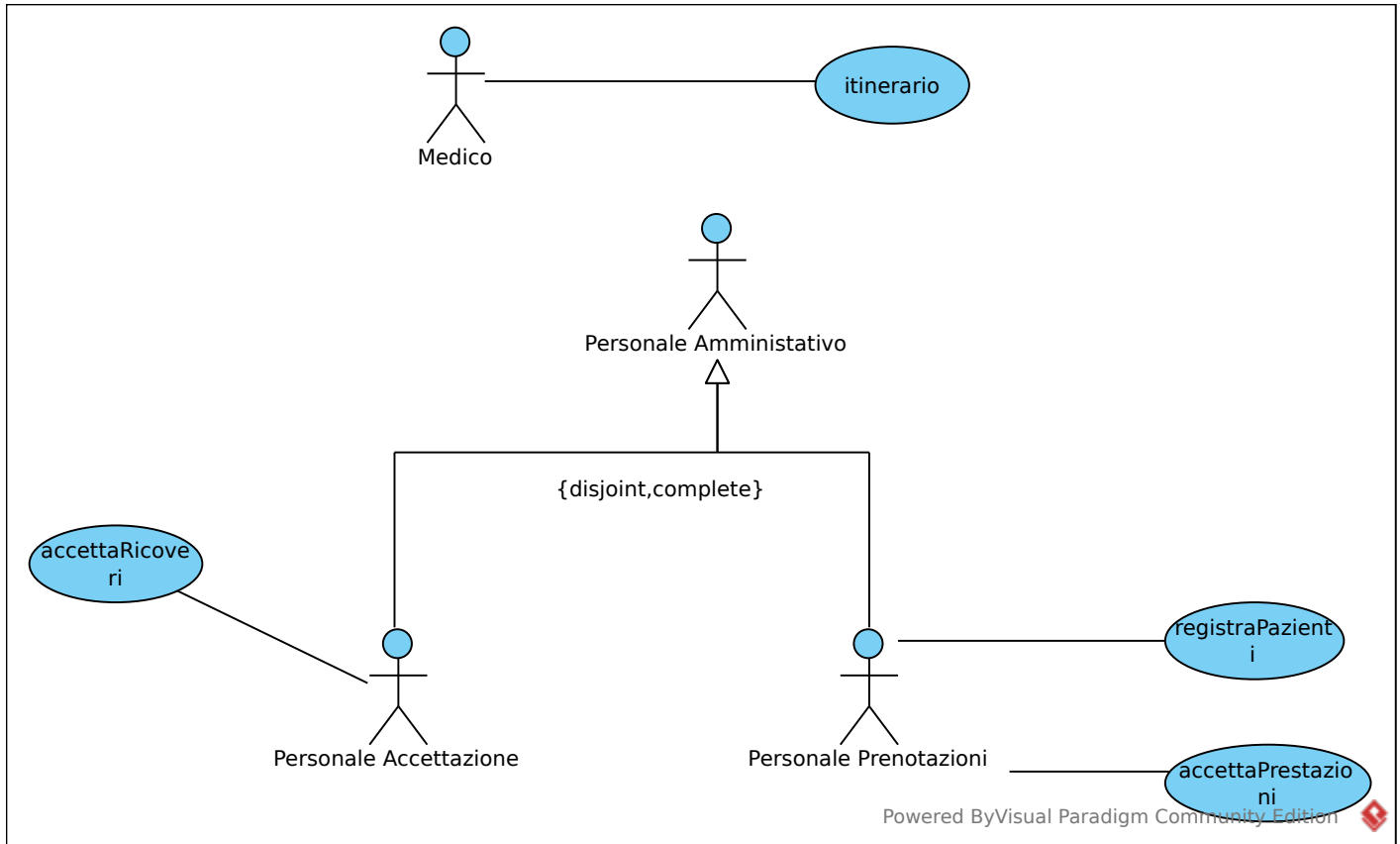


Figure 2: Usecase

4 Ristrutturazione

4.1 Diagramma delle classi ristrutturato

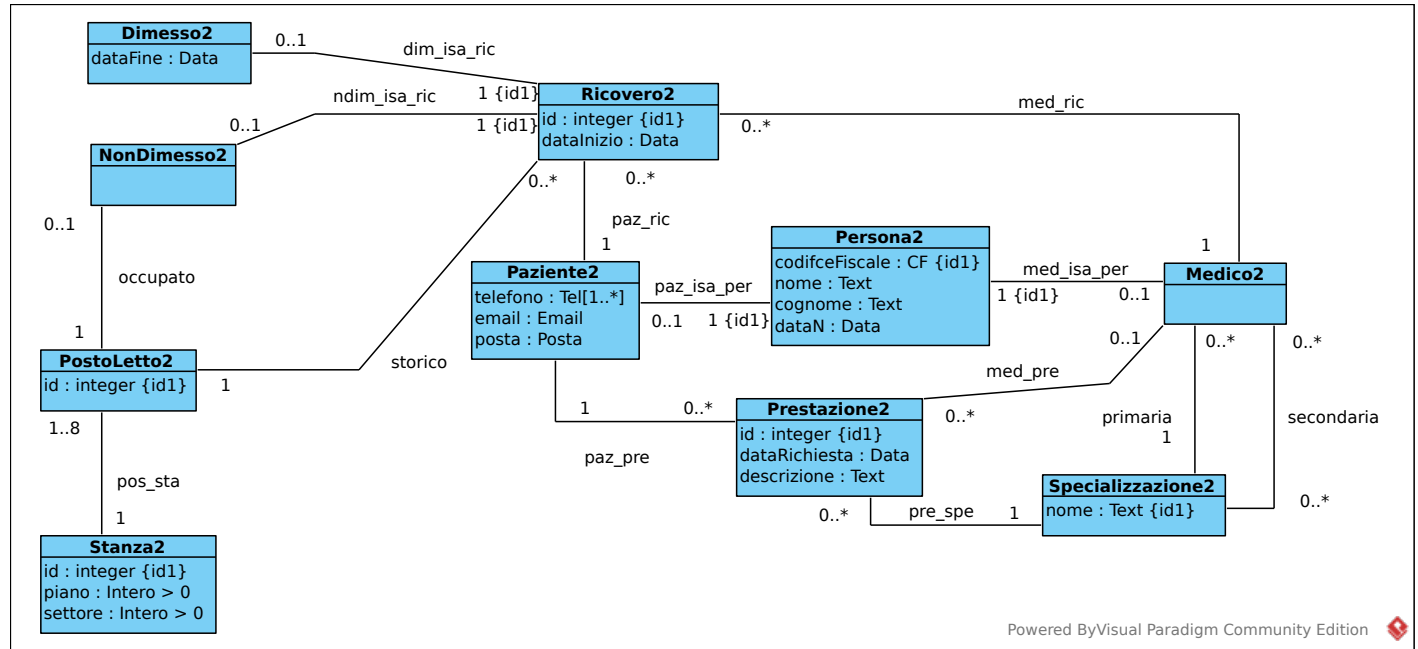


Figure 3: Ristrutturazione

4.2 Dizionario dei dati ristrutturato

- create domain tel as text check (isTel(value));
- create domain cf as text check (isCF(value));
- create domain email as text check (isEmail(value));
- create type posta as (via IndirizzoNotNull, civico CivicoNotNull);
- create domain IndirizzoNotNull as text check (value is not null);
- create domain CivicoNotNull as text check (value is not null and isCivico(value))
- create domain StringS as varchar(50);
- create domain interrogz as integer check (value \neq 0)

4.3 Vincoli ristrutturati

4.3.1 Vincoli Esterni Aggiunti nella Fase di Ristrutturazione

[V.Occupato_isa_Storico]

$$\forall r, nd, p (ndim_isa_ric(nd, r) \wedge occupato(nd, p) \rightarrow storico(r, p))$$

[V.Ricovero_Dimesso_Disjoint_NonDimesso]

$$\neg \exists r, d, nd (ndim_isa_ric(nd, r) \rightarrow \neg dim_isa_ric(d, r))$$

[V.Ricovero_Dimesso_NonDimesso_Complete]

$$\forall r (\exists nd ndim_isa_ric(nd, r)) \vee (\exists d dim_isa_ric(d, r))$$

[V.Persona_Medico_Paziente_Complete]

$$\forall p (\exists m med_isa_per(m, p)) \vee (\exists paz paz_isa_per(paz, p))$$

4.3.2 Vincoli Esterni Rimasti Invariati dopo la Fase di Ristrutturazione

[V.Prestazione.medico_giusto]

$$\forall m, p, s (med_pre(m, p) \wedge pre_spe(p, s) \rightarrow (primaria(m, s) \vee secondaria(m, s)))$$

[V.Medico.secondaria_non_primaria]

$$\forall s, m (secondaria(m, s) \rightarrow \neg primaria(m, s))$$

4.3.3 Vincoli Esterni Modificati durante la Fase di Ristrutturazione

[V.Dimesso.date_consistente]

$$\forall r, d (dim_isa_ric(d, r) \rightarrow \forall d_1, d_2 (dataInizio(r, d_1) \wedge dataFine(d, d_2) \rightarrow d_1 \leq d_2))$$

[V.Ricovero.date_consistente]

$$\forall r, dr, paz, per, dp (paz_ric(paz, r) \wedge paz_isa_per(paz, per) \wedge dataInizio(r, dr) \wedge dataN(per, dp) \rightarrow dr \geq dp)$$

[V.Prestazione.date_consistente]

$$\forall paz, per, pre, d1, d2 (paz_pre(paz, pre) \wedge paz_isa_per(paz, per) \wedge dataRichiesta(pre, d1) \wedge dataN(per, d2) \rightarrow d1 \geq d2)$$

[V.Medico.date_consistente]

$$\forall m, per, r, d1, d2 (med_ric(m, r) \wedge dataInizio(r, d1) \wedge med_isa_per(m, per) \wedge dataN(per, d2) \rightarrow d1 > d2)$$

[V.Paziente.ricovero_singolo]

$$\forall p (\exists nd, r (paz_ric(p, r) \wedge ndim_isa_ric(nd, r)) \rightarrow \neg \exists nd', r' (nd' \neq nd \wedge r' \neq r \wedge ndim_isa_ric(nd', r') \wedge paz_ric(p, r')))$$

[V.Paziente.non_ricoverato_da_se_stesso]

$$\forall paz, per, med, r \ (paz_ric(paz, r) \wedge med_isa_per(med, per) \wedge paz_isa_per(paz, per) \rightarrow \neg med_ric(med, r))$$

4.4 Usecases ristrutturati

4.4.1 Specifica Use-Case Itinerario

itinerario(): Stanza [0..*]

Pre-condizioni: nessuna

Post-condizioni:

- Non modifica lo spazio estensionale.
- Sia m tale che $\text{Medico}(m)$ il medico che si è autenticato nel sistema e che sta usando lo use-case.
- Sia $P = \{p \mid \exists r (\text{paz_ric}(p, r) \wedge \text{med_ric}(m, r) \wedge \text{NonDimesso}(r))\}$.
- Sia $L = \{l \mid \exists n (\text{Occupato}(n, l) \wedge n \in P)\}$.
- Sia $S = \{s \mid \exists l (\text{pos_sta}(l, s) \wedge l \in L)\}$.
- $\text{result} = S$ (poi in SQL verrà ordinato).

4.4.2 Specifica dello Use-Case RegistraPazienti

registrazionePersona(codiceFiscale: CF, nome: Stringa, cognome: Stringa, dataN: Data): Persona

Pre-condizioni:

$$\forall p1 (Persona(p1) \rightarrow \neg codiceFiscale(p1, codiceFiscale))$$

Post-condizioni:

- Modifica dello spazio estensionale:
- Elementi del dominio di interpretazione: α
- Nuove ennuple:
 - `Persona`(α)
 - `codiceFiscale`(α , `codiceFiscale`)
 - `nome`(α , `nome`)
 - `cognome`(α , `cognome`)
 - `dataN`(α , `dataN`)
- `result = Persona`

registrazionePaziente(p: Persona, telefoni: Tel [1..*], email: Email, posta: Posta): Paziente

Pre-condizioni:

$$\neg \exists paz \text{ paz_isa_per}(paz, p)$$

Post-condizioni:

- Modifica dello spazio estensionale:
- Elementi del dominio di interpretazione: α
- Nuove ennuple:
 - `paz_isa_per`(α , `p`)
 - `telefono`(α , `telefoni`)
 - `email`(α , `email`)
 - `posta`(α , `posta`)
- `result = Paziente`

possibiliMedici(s: Specializzazione): Medico [0..*]

Pre-condizioni: nessuna

Post-condizioni:

- Non modifica lo spazio estensionale.
- Sia $P = \{m \mid primaria(m, s)\}$.
- Sia $S = \{m \mid secondaria(m, s)\}$.

- $|P| > 0 \rightarrow result = P$
 $|P| = 0 \rightarrow result = S$

accettaPrestazione(p: Paziente, s: Specializzazione, dataRichiesta: Data): Prestazione
Pre-condizioni:

$$\forall M (possibiliMedici_s(s, M) \rightarrow |M| > 0 \wedge (\forall d (data(adesso, d) \rightarrow dataRichiesta \geq d)))$$

Post-condizioni:

- Modifica dello spazio estensionale:
- Elementi del dominio di interpretazione: α
- Nuove ennuple:
 - `Prestazione`(α)
 - `dataRichiesta`(α , `dataRichiesta`)
 - `descrizione`(α , `descrizione`)
 - `paz_pre`(`p`, α)
 - `pre_spe`(α , `s`)
- `result` = α

4.4.3 Specifica dello Use-Case AccettaRicoveri

postiLettoLiberi(): PostoLetto[0..*]

Pre-condizioni: nessuna

Post-condizioni:

- Non viene modificato lo spazio estensionale dei dati.
- $L = \{l \mid \neg \exists n \text{ Occupato}(l, n)\}$.
- $\text{result} = L$

accettaRicoveri(p: Paziente): NonDimesso

Pre-condizioni:

$$(\neg \exists r, nd \text{ (paz_ric}(p, r) \wedge \text{NonDimesso}(nd) \wedge \text{ndim_isa_ric}(nd, r))) \wedge (\forall L \text{ (postiLettoLiberi}(L) \rightarrow |L| > 0))$$

Post-condizioni:

- Modifica dello spazio estensionale:
- Sia l un elemento a caso in $\{l \mid \neg \exists n \text{ Occupato}(l, n)\}$.
- Elementi del dominio di interpretazione: α, β
- Nuove ennuple:
 - $\text{NonDimesso}(\beta)$
 - $\text{Ricovero}(\alpha)$
 - $\text{ndim_isa_ric}(\beta, \alpha)$
 - $\text{Storico}(\alpha, l)$
 - $\text{pos_sta}(\alpha, l)$
- $\text{result} = \alpha$

4.5 Progettazione della base di dati

1. Persona(codiceFiscale: cf, nome: StringS, cognome: StringS, dataN: Data)
2. Paziente(persona: cf, telefoni: Tel, email: Email, posta: Posta)
 - 2.1. foreign key: persona references persona(codicefiscale)
3. Medico(persona: cf, specializzazionePrimaria: StringS)
 - 3.1. foreign key: persona references persona(codicefiscale)
 - 3.2. foreign key: specializzazionePrimaria references specializzazione(nome)
4. SpecializzazioneSecondaria(medico: cf, specializzazione: stringaS)
 - 4.1. foreign key: medico references medico(persona)

- 4.2. foreign key: specializzazione references specializzazione(nome)
- 5. Prestazione(id: serial, dataRichiesta: Date, descrizione: text, paziente: cf, medico*: cf, specializzazioneRichiesta: StringS)
 - 5.1. foreign key: paziente references paziente(persona)
 - 5.2. foreign key: medico references medico(persona)
 - 5.3. foreign key: specializzazioneRichiesta references specializzazione(nome)
- 6. Ricovero(id: serial, dataInizio: Date, paziente: cf, medico: cf, storico: integer)
 - 6.1. paziente references paziente(persona)
 - 6.2. medico references medico(persona)
 - 6.3. storico references postoletto(id)
- 7. Dimesso(ricovero: integer, dataFine: Data)
 - 7.1. ricovero references ricovero(id)
- 8. NonDimesso(ricovero: integer)
 - 8.1. ricovero references ricovero(id)
 - 8.2. v. inclusione: nondimesso(ricovero) occorre in occupato(nondimesso)
- 9. occupato(postoletto: integer, nonDimesso: integer)
 - 9.1. altra chiave: nonDimesso
 - 9.2. foreign key: postoletto references postoletto(id)
 - 9.3. foreign key: nonDimesso references nondimesso(ricovero)
- 10. postoletto(id: serial, stanza: integer)
 - 10.1. foreign key: stanza references stanza(id)
- 11. stanza(id: serial, piano: interogz, settore: interogz)
 - 11.1. Domanda: come implemento la molteplicità da 1..8?