

# Insulin Pump

Progetto di Ingegneria del Software

Daniele Marretta, Tommaso Maria Lopedote, Jacopo Spallotta



# Introduzione

Il sistema “Insulin Pump” è un modello per la gestione, monitoraggio e controllo per pazienti diabetici che fanno uso di una Pompa di Insulina. Il sistema è in grado di regolare i livelli di glucosio all'interno del sangue del paziente rilasciando dosi di Insulina in base a determinati valori forniti in input ed ad altri valori variabili.

## 1. Descrizione Generale

## 2. Requirements

### 2.1 Insulin Pump

#### 2.1.1 System Requirements

##### 2.1.1.1 Functional

1. Il livello di glucosio non deve scendere sotto i 50mg/dL (ipoglicemia);
2. Il livello di glucosio non deve salire sopra i x mg/dL (iperglicemia);
3. Il livello di glucosio dovrebbe rimanere il più vicino possibile a 100mg/dL;
4. Il sistema garantisce il rilascio di una dose minima di insulina anche in caso di fault di sistema;
6. Il sistema analizza il trend di glicemia in salita o in discesa nel sangue;
7. Con glicemia nei limiti di sicurezza:
  1. In caso di livelli stabili o in discesa di glucosio la pompa non rilascia insulina;
  2. In caso di aumento di glucosio ma tasso di crescita in discesa la pompa non rilascia insulina;
  3. In caso di aumento di glucosio ma tasso di crescita in ascesa la pompa calcola la dose di insulina appropriata;
  4. In caso di trend stabile la pompa non rilascia insulina;
8. Con glicemia alta:
  1. In caso di aumento di glucosio la pompa rilascia una dose appropriata;
  2. In caso di livello glucosio stabile la pompa rilascia una dose minima;
  3. In caso di diminuzione di glucosio ma tasso di decremento in ascesa la pompa non rilascia insulina;
  4. In caso di diminuzione di glucosio con tasso di decremento in discesa la pompa rilascia una dose minima;

##### 2.1.1.2 Non Functional

1. La dose di insulina giornalmente deve essere limitata ad un massimo di x/unit;

#### 2.1.2 User Requirements

##### 2.1.2.1 Functional

1. Garantire un livello di glucosio nel sangue intorno ai 100mg/dL;
2. Informare l'utente in caso di livelli troppo elevati o troppo bassi di glucosio;
3. La pompa dell'insulina avverte l'utente che la riserva sta per esaurirsi;

##### 2.1.2.2 Non Functional

1. Minimizzare il quantitativo di insulina iniettata giornalmente;
2. Massimizzare il tempo fra un controllo e il successivo, mantenendosi fra 5 e 10 minuti;
3. La pompa dell'insulina avverte l'utente che la riserva sta per esaurirsi;

## 2.2 Patient

### 2.2.1 System Requirements

#### 2.2.1.1 Functional

1. Il sistema consente l'inserimento di parametri peso, età, altezza e sesso per la definizione delle caratteristiche del paziente;
2. Il sistema esegue il logging sul database di parametri chiave per la definizione dello stato del paziente:
  1. Livello del glucosio nel sangue (G - mg/dL);
  2. Contrazione dell'insulina nel Plasma (I - pmol/L);
  3. Quantitativo di glucosio nello stomaco (Q<sub>sto</sub> - mg);
  4. Produzione endogena di glucosio (EGP - adimensionale);
  5. Utilizzo dipendente di insulina (UID - mg/kg/min);
  6. Escrezione renale (E - mg/kg/min);
  7. Secrezione di Insulina e C-peptide (ISR - pmol/min);

### 2.2.2 User Requirements

#### 2.2.2.1 Functional

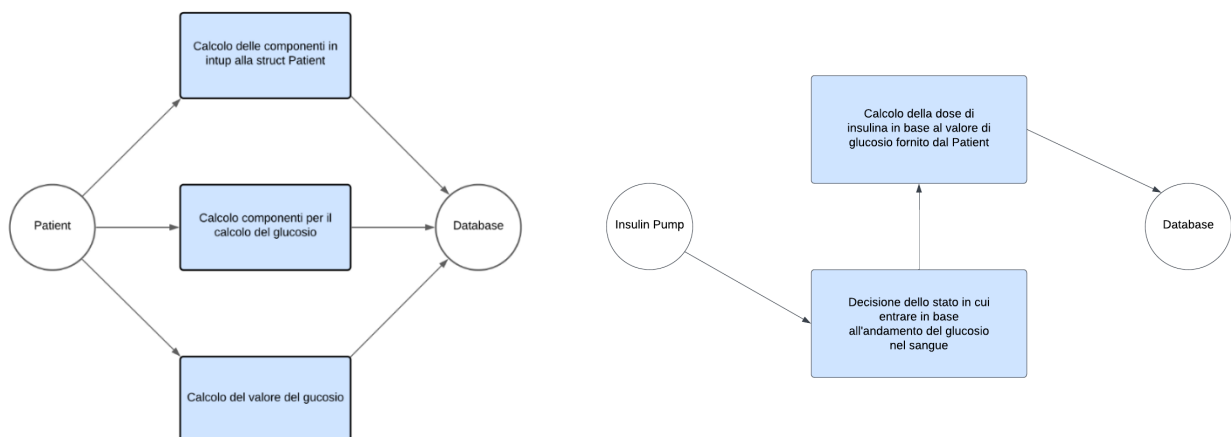
1. Il modello riproduce fedelmente un paziente con diabete di tipo 2;

## 3. Diagrams

### 3.1 Use Case UML

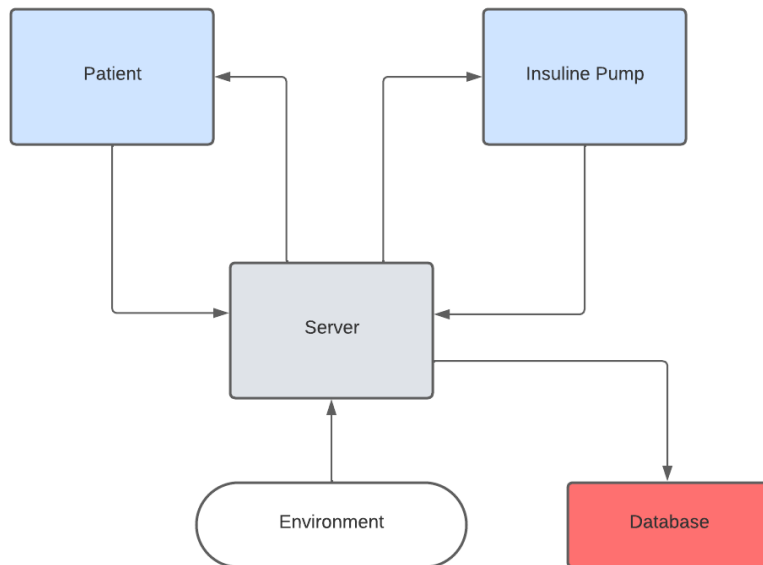
Use Case UML di

1. Patient Model per la registrazione nel database dei valori necessari al calcolo del glucosio.
2. Insulin Pump model per la registrazione nel database del valore di insulina somministrata



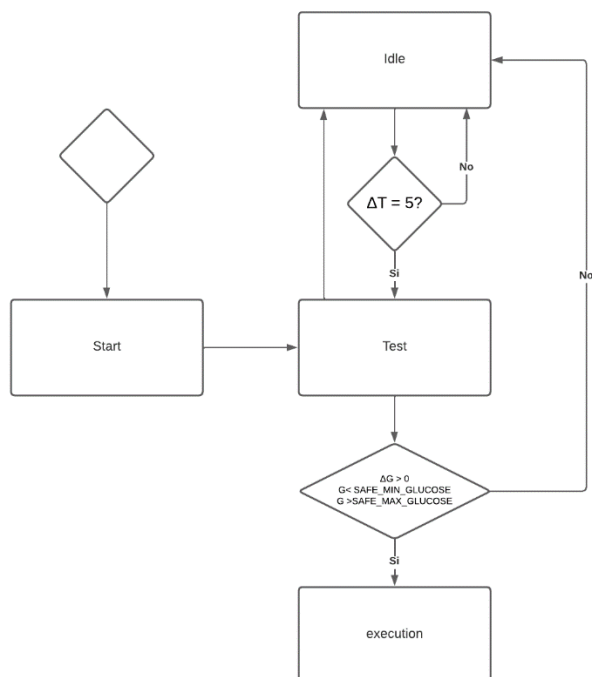
### 3.2 Architettura del sistema

Architettura del sistema che comprende due modelli (Patient ed Insuline Pump) comunicanti tra loro tramite un server Redis, sul quale agisce un Environment e salva i dati su un Database globale.



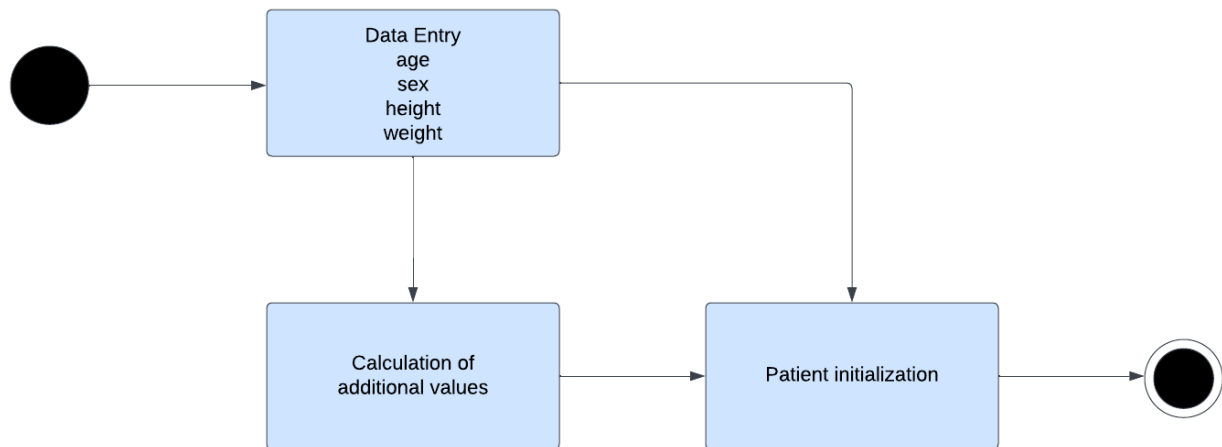
### 3.3 Activity Diagram UML

Activity Diagram UML per gli stati in cui può trovarsi la pompa di insulina.



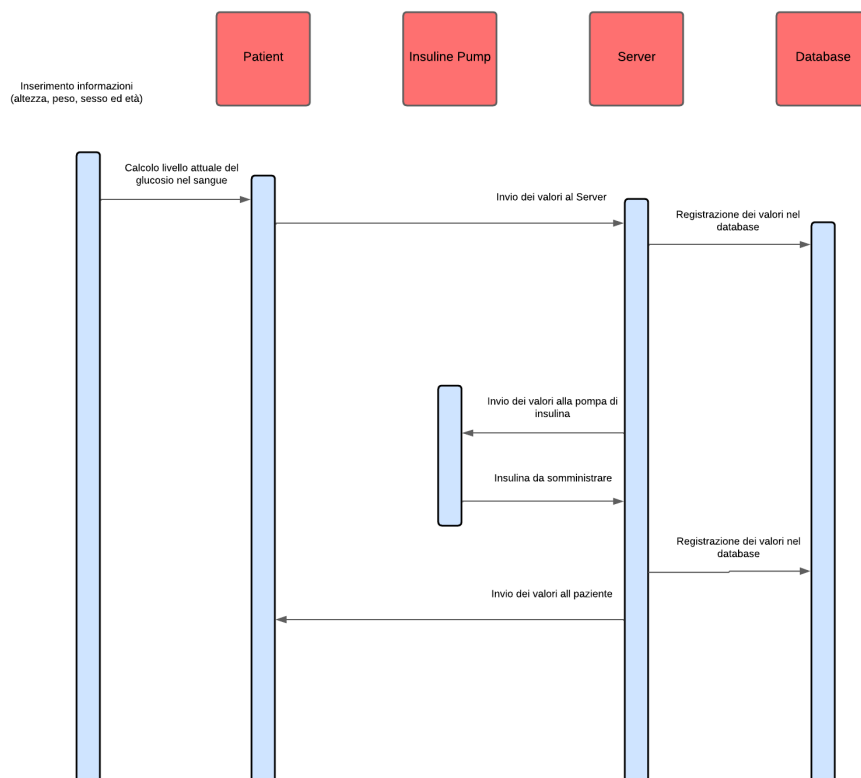
### 3.4 State Diagram UML

State Diagram UML per l'inserimento e il calcolo dei valori necessari all'inizializzazione del Patient



### 3.5 Message Sequence Chart UML

Message Sequence Chart UML per la comunicazione tra Modelli-Server-Database



### 3.6 Risultati sperimentali

Descrivere i risultati ottenuti dalla simulazione del sistema

## 4. Componenti

### 4.1 Models

#### 4.1.1 Patient

Calcola il valore di insulina necessario in base alla concentrazione di glucosio attuale nel sangue e valori correlati. Il modello “Patient” simula un paziente in grado di interagire con il modello “Insulin Pump” della Pompa di Insulina. La simulazione calcola in modo dinamico i livelli di glucosio nel sangue e l’insulina necessaria a regolare il valore del glucosio stesso.

##### 4.1.1.1 Inizializzazione

La fase di inizializzazione del Paziente richiede alla base 4 parametri necessari

- Età
- Sesso
- Peso
- Altezza

Attraverso i valori dei parametri principali ne vengono calcolati altri 6

- l'indice di massa corporea (BMI): composizione corporea in relazione all'altezza di un individuo
- l'Area Superficiale Corporea (BSA): misura della superficie esterna del corpo di un individuo
- Volume di distribuzione centrale dell'insulina ( $V_c$ ): volume apparente nel quale l'insulina è distribuita all'interno del corpo
- $b_1$ : coefficiente o un parametro che influisce sulla velocità o sulla cinetica del glucosio o dell'insulina all'interno del sistema.
- $a_1$  e  $\text{fra}$ : i quali valori vengono stabiliti in relazione al BMI

Una volta ottenuti, i valori vengono passati alla struttura del Patient che viene inizializzata

##### 4.1.1.2 Calcolo del glucosio

Il calcolo del valore del glucosio nel sangue avviene all'interno di un ciclo while e viene aggiornato ad ogni iterazione per ottenere ad ogni intervallo di tempo un valore (possibilmente anche nullo) di insulina necessaria.

I valori necessari per il calcolo del glucosio appartengono alle seguenti componenti

- Glucose Kinetics: *velocità di assorbimento, distribuzione, metabolismo ed eliminazione del Glucosio nel corpo*
- Insulin Kinetics: *velocità di assorbimento, distribuzione, metabolismo ed eliminazione dell'Insulina nel corpo*
- Rate of Glucose Appearance: *velocità con la quale il glucosio appare nel flusso sanguigno*
- Endogenous Glucose Production: *quantità di glucosio prodotta internamente dal corpo*
- Glucose Utilization: *eliminazione del glucosio attraverso i reni*
- C-Peptide Kinetics: *velocità di assorbimento, distribuzione ed eliminazione del C-peptide*
- Insulin and C-Peptide Secretion: *processo di rilascio di insulina e C-peptide da parte delle cellule beta del pancreas in risposta all'aumento di glucosio nel sangue*

Le formule per il calcolo del glucosio sono discretizzate in base al tempo, dunque nel calcolo dei valori di ogni ad ogni componente al tempo  $t+1$  sono necessari i relativi valori al tempo  $t$ .

#### 4.1.2 Insuline Pump

Riceve dal paziente il valore di insulina di cui ha bisogno e la inietta a determinate condizioni. Il modello “Insulin Pump” simula il comportamento di una Pompa di Insulina. Il sistema è caratterizzato da una serie di

controlli e basato su un ciclo temporale. Nell'esecuzione Il programma stampa periodicamente lo stato della pompa sulla console e continua l'esecuzione in modo indefinito. Configurazione del Debug:

La struttura "Insulin Pump" viene inizializzata con parametri relativi a:

1. Valore minimo del glucosio nel sangue
2. Valore safe del glucosio
3. Valore massimo del glucosio nel sangue
4. Valore safe massimo del glucosio
- 5.
- 6.
7. come valori minimi e massimi di glucosio e il puntatore a una funzione di test (test).

Il funzionamento del sistema si basa anch'esso su un ciclo while che simula il passare del tempo, attraverso il quale viene calcolato lo stato successivo in cui si trova la pompa utilizzando la funzione next.

### 4.1.3 Environment

L'Environment ha come compito principale quello di controllare la Pompa dell'Insulina ed il Paziente.

Il sistema fa uso di un database creato attraverso PostgreSQL sul quale vengono registrate le informazioni e i dati sullo stato del Paziente e sulla Pompa dell'Insulina.

La simulazione del sistema è basata su un timer assoluto grazie al quale il Paziente può alternare lo stato di alimentazione con lo stato di digiuno, indicato attraverso la variabile *delta*. La durata di entrambe le fasi, *meal\_duration* e *fasting\_duration* sono modificabili

## 4.2 Timer e Orologio

Necessari per la gestione degli intervalli di tempo tra le iniezioni ed il calcolo delle funzioni in base al tempo

## 4.3 Console

Funge da interfaccia con l'utente mostrando i valori attuali di glucosio nel sangue ed insulina somministrata. La console fornisce ad ogni intervallo di tempo

- Il valore attuale del glucosio nel sangue del paziente;
- Il valore di Insulina fornita dalla Pompa di Insulina
- Lo stato del paziente (isEating) ogni tot intervalli di tempo

## 4.4 Monitors

## 5. Database e Logs

Il Database utilizzato nel sistema, implementato attraverso l'utilizzo di PostgreSQL, è strutturato per organizzare i log generati dai modelli Patient e Insulin Pump, che vengono salvati nelle tabelle del database attraverso record con attributi associati. Di seguito analizziamo in maniera più dettagliata i record generati da Patient, Insulin Pump ed Environment

Per ogni record inserito si hanno 5 attributi che indicano

- **tempo** assoluto in nanosecondi in cui sono stati inseriti i dati
- **vid** id univoco associato alle variabili inserite
- **varvalue** valore della grandezza associata al **vid** corrispondente
- **loginfo** informazione dettagliata sulla tipologia del valore (es. Glucose)
- **t** unità di tempo interno al sistema

### 5.1 Patient

I record del **Patient** inseriti all'interno della **LogTable** sono relativi al

1. Glucose value
2. Insulin value
3. Endogenous Glucose Production
4. Glucosio nello stomaco
5. Insulin utilization
6. Renal Excretion
7. Insulin secretion

Ogni record possiede 5 attributi come indicato in 5.

### 5.2 Insuline Pump

I record dell'**Insuline Pump** inseriti all'interno della **LogTable** sono

1. Stato della pompa di insulina
2. Dose calcolata (comp\_dose)

Ogni record possiede 5 attributi come indicato in 5.

### 5.3 Environment

L'unico record dell'**Environment** inserito all'interno della **LogTable** è il

1. Cibo assunto (delta)

Ogni record possiede 5 attributi come indicato in 5.

	t integer	nanosec bigint	vid integer	sysname character varying (200)	varname character varying (30)	varvalue double precision	varinfo character varying (200)
11	1	3448597	10	Environment	delta	1	Food intake
12	1	5882486	8	Insulin pump	pump_state	2	State of insulin pump
13	1	5882486	9	Insulin pump	comp_dose	0	dose (pmol/L)
14	1	3049506	1	Patient	G	90	Plasma glucose concentration (mg/dL)
15	1	3049506	2	Patient	I	54	Plasma insulin concentration (pmol/L)
16	1	3049506	3	Patient	Qsto	0	Amount of glucose in the stomach (mg)
17	1	3049506	4	Patient	EGP	1.59	endogenous glucose production
18	1	3049506	5	Patient	Uid	0.59	Insulin-dependent utilization (mg/kg/min)
19	1	3049506	6	Patient	E	0	Renal excretion (mg/kg/min)
20	1	3049506	7	Patient	ISR	53.946645	Insulin and c-peptide secretion (pmol/min)



## 6. Server Redis

Il server Redis utilizzato nel sistema svolge anche la funzione di database di tipo chiave-valore e supporta diverse strutture dati come stringhe, liste, insiemi e altri. Nel sistema dell'Insulin Pump il server è utilizzato per scrivere e leggere valori tramite stream di dati.

L'interazione tra i modelli ed il server Redis può essere riassunta in 4 fasi differenti:

- **Connessione** I modelli si connettono al server Redis in esecuzione sulla porta predefinita (6379) sull'host "localhost" utilizzando la funzione `redisConnect()`.
- **Operazioni sui dati** Vengono eseguite varie operazioni sui dati nel server Redis, come eliminare i flussi esistenti (`DEL`), creare nuovi flussi (`XADD`) e leggere i messaggi da un flusso (`XREADGROUP`).
- **Scrittura e lettura dei dati** I dati relativi al paziente e alle varie grandezze fisiologiche vengono scritti e letti dal server Redis utilizzando i comandi di scrittura (`XADD`) e lettura (`XREADGROUP`). Ad esempio, i valori relativi al glucosio vengono scritti nel flusso di scrittura (`WRITE_STREAM`) e letti dal flusso di lettura (`READ_STREAM`).
- **Elaborazione dei dati** Dopo aver letto i dati dal server Redis, vengono eseguite operazioni di elaborazione su di essi per calcolare nuovi valori fisiologici. Questi nuovi valori vengono quindi scritti nuovamente nel server Redis per essere eventualmente letti da altri componenti del sistema.

# Da revisionare

User Requirements

Dose iniettata in base a determinati valori di glucosio

Se il glucosio è sopra o sotto i valori attesi avviene il rilascio dell'insulina

Nessuna iniezione se il valore del glucosio rientra nel valore atteso

Limiti

Dose massima per singola iniezione

Sul numero di dosi giornaliere

Sull'insulina totale giornaliera

Al rilascio di una dose avviene un confronto con la dose precedente per tenere traccia dell'andamento dei valori dell'insulina

Dose di insulina minimizzata con monitoraggio costante sul valore di glucosio con lo scopo di iniettare meno insulina possibile ogni rilascio

Controllo sul numero totale di iniezioni giornaliere

Doppio timer con tempo assoluto e tempo relativo e controllo del tempo giusto dopo ogni rilascio di insulina

**Alerts**

UNDERDOSE\_FAILURE: se va sotto il valore atteso

OVERDOSE\_FAILURE: se va sopra il valore atteso

EARLYTIME\_FAILURE: se il rilascio avviene prima dell'intervallo prestabilito

LATETIME\_FAILURE: se il rilascio avviene dopo l'intervallo prestabilito

Reset dei valori a mezzanotte

FAULT Protection e Warning

SENSOR\_FAILURE se uno dei sensori non dovesse funzionare avvisa l'utente con un warning

Manda un warning sul livello di insulina nel contenitore (100ml max)

LOW\_INSULINE\_WARNING Quando è quasi vuota (sotto un determinato valore)

EMPTY\_INSULINE\_WARNING Quando è vuota

LOW\_BATTERY\_WARNING: warning sul livello della batteria (quando è quasi scarica)

UNDERDOSE\_FAILURE & OVERDOSE\_FAILURE quando i risultati del doppio controllo non corrispondono ai valori attesi

EARLYTIME\_FAILURE & LATETIME\_FAILURE quando il rilascio di insulina avviene troppo a breve o troppo in ritardo rispetto al rilascio precedente. Il controllo viene effettuato dopo ogni rilascio confrontando un timer con tempo relativo (pari all'intervallo di tempo con cui viene rilasciata l'insulina) e un timer con tempo assoluto (orologio)