



AlphaSix

# AlphaSix

## Analisi dei Requisiti

### Informazioni sul documento

<b>Nome Documento</b>	AnalisiDeiRequisiti v2.0.0.pdf
<b>Versione</b>	2.0.0
<b>Data di Creazione</b>	27 novembre 2018
<b>Data ultima modifica</b>	04 marzo 2019
<b>Stato</b>	Approvato
<b>Redazione</b>	Ciprian Voinea Nicola Carlesso
<b>Verifica</b>	Matteo Marchiori Timoty Granziero
<b>Approvazione</b>	Samuele Gardin
<b>Uso</b>	Esterno
<b>Distribuzione</b>	AlphaSix
<b>Destinato a</b>	Prof. Tullio Vardanega, Prof. Riccardo Cardin, Imola Informatica
<b>Email di riferimento</b>	alpha.six.unipd@gmail.com

### Descrizione

L'Analisi dei Requisiti descrive le funzionalità di Butterfly, determinandone i requisiti che analizziamo e valutiamo.

## Registro delle modifiche

Versione	Descrizione	Ruolo	Nominativo	Data
2.0.0	Approvazione per il rilascio	Responsabile	Samuele Gardin	2019-03-04
1.3.0	Verifica finale	Verificatore	Timoty Granziero	2019-03-01
1.2.3	Migliorato §4.1	Analista	Ciprian Voinea	2019-02-27
1.2.2	Verifica §3	Verificatore	Matteo Marchiori	2019-02-26
1.2.1	Aggiornati diagrammi UML in §3	Analista	Nicola Carlesso	2019-02-20
1.2.0	Verifica	Verificatore	Samuele Gardin	2019-02-18
1.1.3	Riviste sezioni §3	Analista	Nicola Carlesso	2019-02-16
1.1.2	Verifica §3	Verificatore	Matteo Marchiori	2019-02-09
1.1.1	Approfondimento §3	Analista	Ciprian Voinea	2019-02-06
1.1.0	Verifica	Verificatore	Matteo Marchiori	2019-02-04
1.0.3	Inserimento §3	Analista	Ciprian Voinea	2019-01-28
1.0.2	Approfondimento §2	Responsabile	Laura Cameran	2019-01-26
1.0.1	Migrazione tecnologie in appendice §A	Analista	Ciprian Voinea	2019-01-24
1.0.0	Approvazione per il rilascio	Responsabile	Nicola Carlesso	2019-01-12
0.3.0	Verifica finale	Verificatore	Matteo Marchiori	2019-01-10
0.2.6	Aggiunto §4.5	Analista	Laura Cameran	2018-12-28
0.2.5	Aggiunto §4.3	Analista	Laura Cameran	2018-12-27
0.2.4	Aggiunto §4.4	Analista	Nicola Carlesso	2018-12-23
0.2.3	Inserito §4.2	Analista	Nicola Carlesso	2018-12-20
0.2.2	Aggiunto §4.1	Analista	Laura Cameran	2018-12-19
0.2.1	Iniziati requisiti	Analista	Laura Cameran	2018-12-15
0.2.0	Verifica del documento	Verificatore	Timoty Granziero	2018-12-15
0.1.9	Inserito UC4	Analista	Timoty Granziero	2018-12-10
0.1.8	Inserito UC1	Analista	Matteo Marchiori	2018-12-10
0.1.7	Inserito UC2	Analista	Laura Cameran	2018-12-09
0.1.6	Inserito UC3	Analista	Ciprian Voinea	2018-12-08
0.1.5	Inserito UC5	Analista	Samuele Gardin	2018-12-07
0.1.4	Aggiunto UC6	Analista	Nicola Carlesso	2018-12-07
0.1.3	Aggiunto §A.7.1	Analista	Ciprian Voinea	2018-12-06
0.1.2	Aggiunto §2.3.4	Analista	Ciprian Voinea	2018-12-05
0.1.1	Aggiunto §3	Analista	Laura Cameran	2018-12-05
0.1.0	Verifica documento	Verificatore	Nicola Carlesso	2018-12-05



Versione	Descrizione	Ruolo	Nominativo	Data
0.0.6	Aggiunto §2.3.3	Analista	Ciprian Voinea	2018-12-04
0.0.5	Aggiunti §2.3.2 e §2.3.1	Analista	Ciprian Voinea	2018-12-03
0.0.4	Aggiunta §2	Analista	Matteo Marchiori	2018-12-01
0.0.3	Aggiunto §1	Analista	Nicola Carlesso	2018-11-30
0.0.2	Creazione struttura documento	Analista	Nicola Carlesso	2018-11-29
0.0.1	Creazione template	Redattore	Timoty Granziero	2018-11-27

# Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
1.1	Glossario e documenti esterni . . . . .	2
1.2	Scopo documento . . . . .	2
1.3	Scopo del prodotto . . . . .	2
1.4	Riferimenti . . . . .	2
1.4.1	Normativi . . . . .	2
1.4.2	Informativi . . . . .	2
<b>2</b>	<b>Descrizione del prodotto</b>	<b>4</b>
2.1	Funzionalità . . . . .	4
2.2	Caratteristiche degli utenti . . . . .	4
2.3	Tecnologie e scelte relative al progetto . . . . .	4
2.3.1	Producer . . . . .	5
2.3.1.1	Producer GitLab . . . . .	5
2.3.1.2	Producer Redmine . . . . .	5
2.3.2	Apache Kafka . . . . .	6
2.3.3	Consumer . . . . .	6
2.3.3.1	Consumer Telegram . . . . .	6
2.3.3.2	Consumer Email . . . . .	6
2.3.4	Gestore Personale . . . . .	6
<b>3</b>	<b>Casi d'uso</b>	<b>7</b>
3.1	Attori . . . . .	7
3.2	Sistemi e sottosistemi . . . . .	7
3.3	Elenco casi d'uso . . . . .	8
3.3.1	UC1-PR - Redmine segnala apertura issue al Producer Redmine . . . . .	8
3.3.2	UC2-PR - Redmine segnala la modifica di una issue al Producer Redmine . . . . .	8
3.3.3	UC3-PG - GitLab segnala apertura issue al Producer GitLab . . . . .	9
3.3.4	UC4-PG - Gitlab segnala la modifica di una issue al Producer Gitlab . . . . .	9
3.3.5	UC5-PG - GitLab segnala un evento di push a Producer GitLab . . . . .	10
3.3.6	UC6-GP - Producer Redmine invia messaggio al Gestore Personale . . . . .	10
3.3.7	UC6.1-GP - Producer Redmine invia messaggio di apertura issue al Gestore Personale . . . . .	11
3.3.8	UC6.2-GP - Producer Redmine invia messaggio di modifica issue al Gestore Personale . . . . .	12
3.3.9	UC6.3-GP - Producer Redmine scarta i messaggi non validi . . . . .	12
3.3.10	UC7-GP - Producer GitLab invia messaggio al Gestore Personale . . . . .	13
3.3.11	UC7.1-GP - Producer GitLab invia uno o più messaggi di commit al Gestore Personale . . . . .	14
3.3.12	UC7.2-GP - Producer GitLab invia messaggio di issue al Gestore Personale . . . . .	14
3.3.13	UC7.2.1-GP - Producer GitLab invia messaggio di una nuova issue al Gestore Personale . . . . .	15
3.3.14	UC7.2.2-GP - Producer GitLab invia messaggio di modifica di una issue al Gestore Personale . . . . .	16
3.3.15	UC7.2.3-GP - Producer GitLab scarta i messaggi non validi . . . . .	16
3.3.16	UC8-CT - Gestore Personale invia il messaggio finale al Consumer Telegram . . . . .	17
3.3.17	UC9-CE - Gestore Personale invia il messaggio finale al Consumer Email . . . . .	17
3.3.18	UC10-BT - Consumer Telegram inoltra il messaggio finale al bot Telegram . . . . .	18
3.3.19	UC11-SE - Consumer Email inoltra il messaggio finale al server Email . . . . .	18

3.3.20	UC12-GP - Accesso . . . . .	19
3.3.21	UC12.1-GP - Accesso dell'utente nel sistema . . . . .	20
3.3.22	UC12.1.1-GP - Inserimento identificativo . . . . .	21
3.3.23	UC12.2-GP - Errore identificativo inesistente . . . . .	21
3.3.24	UC13-GP - Uscita dell'utente dal sistema . . . . .	21
3.3.25	UC14-GP - Aggiunta nuovo utente . . . . .	22
3.3.26	UC14.1-GP - Utente aggiunto con successo . . . . .	23
3.3.27	UC14.1.1-GP - Inserimento nome utente . . . . .	24
3.3.28	UC14.1.2-GP - Inserimento cognome utente . . . . .	24
3.3.29	UC14.1.3-GP - Inserimento contatto Email . . . . .	24
3.3.30	UC14.1.4-GP - Inserimento contatto Telegram . . . . .	25
3.3.31	UC14.2-GP - Errore utente già presente nel sistema . . . . .	25
3.3.32	UC15-GP - Rimozione utente dal sistema . . . . .	26
3.3.33	UC15.1-GP - Rimozione avvenuta con successo . . . . .	27
3.3.34	UC15.1.1-GP - Inserimento contatto Email . . . . .	27
3.3.35	UC15.1.2-GP - Inserimento contatto Telegram . . . . .	28
3.3.36	UC15.2-GP - Errore contatto non presente nel sistema . . . . .	28
3.3.37	UC15.3-GP - L'utente rimuove se stesso dal sistema . . . . .	28
3.3.38	UC16-GP - Modifica utente . . . . .	29
3.3.39	UC16.1-GP - Selezione identificativo . . . . .	30
3.3.40	UC16.1.1-GP - Modifica utente avvenuta con successo . . . . .	31
3.3.41	UC16.1.1.1-GP - Inserimento del nuovo nome . . . . .	32
3.3.42	UC16.1.1.2-GP - Inserimento del nuovo cognome . . . . .	32
3.3.43	UC16.1.1.3-GP - Inserimento del nuovo contatto Email . . . . .	32
3.3.44	UC16.1.1.4-GP - Inserimento del nuovo contatto Telegram . . . . .	33
3.3.45	UC16.1.2-GP - Errore, nuovi dati dell'utente già esistenti . . . . .	33
3.3.46	UC16.2-GP - Errore identificativo inesistente . . . . .	33
3.3.47	UC17-GP - Aggiunta preferenze . . . . .	34
3.3.48	UC17.1-GP - Iscrizione Topic . . . . .	35
3.3.49	UC17.2-GP - Aggiunta dei giorni di indisponibilità nel calendario . . . . .	35
3.3.50	UC17.3-GP - Aggiunta della piattaforma di messaggistica preferita . . . . .	36
3.3.51	UC17.4-GP - Aggiunta persona di fiducia . . . . .	36
3.3.52	UC17.5-GP - Errore identificativo persona di fiducia inesistente . . . . .	36
3.3.53	UC17.6-GP - Aggiunta keyword per i push di GitLab . . . . .	37
3.3.54	UC17.7-GP - Errore keyword già esistente . . . . .	37
3.3.55	UC18-GP - Rimozione preferenze . . . . .	38
3.3.56	UC18.1-GP - Disiscrizione Topic . . . . .	39
3.3.57	UC18.2-GP - Rimozione di uno o più giorni di irreperibilità nel calendario . . . . .	39
3.3.58	UC18.3-GP - Rimozione preferenza piattaforma di messaggistica . . . . .	39
3.3.59	UC18.4-GP - Rimozione persona di fiducia . . . . .	40
3.3.60	UC18.5-GP - Errore identificativo persona di fiducia inesistente . . . . .	40
3.3.61	UC18.6-GP - Rimozione con successo di keyword per i push di GitLab . . . . .	41
3.3.62	UC18.7-GP - Errore keyword da rimuovere non presente . . . . .	41
<b>4</b>	<b>Requisiti</b>	<b>42</b>
4.1	Requisiti di funzionalità . . . . .	43
4.2	Requisiti di qualità . . . . .	48
4.3	Requisiti di vincolo . . . . .	50
4.4	Tracciamento . . . . .	51
4.4.1	Tracciamento fonte-requisito . . . . .	51
4.4.2	Tracciamento requisito-fonte . . . . .	58
4.5	Riepilogo . . . . .	64



<b>A</b>	<b>Approfondimenti tecnologici</b>	<b>65</b>
A.1	Webhook . . . . .	65
A.2	Redmine . . . . .	65
A.3	GitLab . . . . .	66
A.4	Apache Kafka . . . . .	66
A.5	Telegram . . . . .	66
A.5.1	BotFather . . . . .	66
A.6	Email . . . . .	66
A.7	Docker . . . . .	66
A.7.1	Container software . . . . .	66
A.7.1.1	Differenza tra container e macchina virtuale . . . . .	67
A.7.2	DockStation . . . . .	67

## Elenco delle tabelle

1	Elenco dei requisiti di funzionalità (1)	43
2	Elenco dei requisiti di funzionalità (2)	44
3	Elenco dei requisiti di funzionalità (3)	45
4	Elenco dei requisiti di funzionalità (4)	46
5	Elenco dei requisiti di funzionalità (5)	47
6	Elenco dei requisiti di qualità (1)	48
7	Elenco dei requisiti di qualità (2)	49
8	Elenco dei requisiti di vincolo	50
9	Elenco dei requisiti del capitolato (1)	51
10	Elenco dei requisiti del capitolato (2)	52
11	Elenco dei requisiti per i casi d'uso (1)	52
12	Elenco dei requisiti per i casi d'uso (2)	53
13	Elenco dei requisiti per i casi d'uso (3)	54
14	Elenco dei requisiti per gli obiettivi di qualità e verbali	54
15	Elenco dei requisiti da fonte interna (1)	55
16	Elenco dei requisiti da fonte interna (2)	56
17	Elenco dei requisiti da fonte interna (3)	57
18	Elenco dei requisiti funzionali in rapporto alle fonti (1)	58
19	Elenco dei requisiti funzionali in rapporto alle fonti (2)	59
20	Elenco dei requisiti funzionali in rapporto alle fonti (3)	60
21	Elenco dei requisiti funzionali in rapporto alle fonti (4)	61
22	Elenco dei requisiti funzionali in rapporto alle fonti (5)	62
23	Elenco dei requisiti di qualità in rapporto alle fonti (1)	63
24	Elenco dei requisiti di qualità in rapporto alle fonti (2)	64
25	Elenco dei requisiti di vincolo in rapporto alle fonti	64
26	Riepilogo dei requisiti	64

## Elenco delle figure

1	Visione generale del sistema Butterfly	4
2	UC6-GP - Producer Redmine invia messaggio al Gestore Personale	10
3	UC7-GP - Producer GitLab invia messaggio al Gestore Personale	13
4	UC7.2-GP - Producer GitLab invia messaggio di issue al Gestore Personale	14
5	UC12-GP - Accesso	19
6	UC12.1-GP - Accesso dell'utente nel sistema	20
7	UC14-GP - Aggiunta nuovo utente	22
8	UC14.1-GP - Utente aggiunto con successo	23
9	UC15-GP - Rimozione utente dal sistema	26
10	UC15.1-GP - Rimozione avvenuta con successo	27
11	UC16-GP - Modifica utente	29
12	UC16.1-GP - Selezione identificativo	30
13	UC16.1.1-GP - Modifica utente avvenuta con successo	31
14	UC17-GP - Aggiunta preferenze	34
15	UC18-GP - Rimozione preferenze	38

# 1 Introduzione

## 1.1 Glossario e documenti esterni

Al fine di rendere il documento più chiaro possibile, i termini che possono assumere un significato ambiguo o i riferimenti a documenti esterni avranno delle diciture convenzionali:

- **D**: indica che il termine si riferisce al nome di un particolare documento (ad esempio *PianoDiProgetto v2.0.0<sub>D</sub>*).
- **G**: indica che il termine si riferisce ad una voce riportata nel Glossario (ad esempio *REDMINE<sub>G</sub>*).

## 1.2 Scopo documento

Tale *DOCUMENTO<sub>G</sub>* ha l'obiettivo di esporre e analizzare i *REQUISITI<sub>G</sub>* espliciti e impliciti per la realizzazione del *PROGETTO<sub>G</sub>* Butterfly (C1) proposto dall'azienda Imola Informatica.

Il documento vuole fungere da base per la fase di progettazione del software in modo che essa sia conforme alle richieste fatte dall'azienda proponente.

## 1.3 Scopo del prodotto

Lo scopo del *PRODOTTO<sub>G</sub>* è creare un *APPLICATIVO<sub>G</sub>* per poter gestire i messaggi o le segnalazioni provenienti da diversi prodotti per la realizzazione di software, come *REDMINE<sub>G</sub>*, *GITLAB<sub>G</sub>* e opzionalmente *SONARQUBE<sub>G</sub>*, attraverso un *BROKER<sub>G</sub>* che possa incanalare questi messaggi e distribuirli a strumenti come *TELEGRAM<sub>G</sub>*, e-mail e opzionalmente *SLACK<sub>G</sub>*.

Per ciascuna di queste tecnologie viene creato un *PRODUCER<sub>G</sub>* e un *CONSUMER<sub>G</sub>* associato per l'effettiva ricezione e invio delle varie segnalazioni. Il software dovrà inoltre essere in grado di riconoscere il *TOPIC<sub>G</sub>* dei messaggi in input per poterli inviare in determinati canali a cui i destinatari dovranno iscriversi.

Dovrà essere in grado, attraverso la lettura di particolari *METADATI<sub>G</sub>*, di reindirizzare i messaggi ricevuti al destinatario più appropriato. È anche richiesto di creare un canale specifico per gestire le particolari esigenze dell'azienda. Questo viene chiamato Gestore Personale e permette agli utenti di poter personalizzare su quale piattaforma di messaggistica e da quali Topic specifici ricevere i messaggi.

## 1.4 Riferimenti

### 1.4.1 Normativi

- *NormeDiProgetto v2.0.0<sub>D</sub>*
- *CAPITOLATO<sub>G</sub>* d'appalto C1  
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C1.pdf>

### 1.4.2 Informativi

- Presentazione capitolato C1  
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C1p.pdf>
- *SLIDE<sub>G</sub>* del corso di Ingegneria del Software
  - Analisi dei requisiti  
<https://www.math.unipd.it/~tullio/IS-1/2018/Dispense/L08.pdf>





– Diagrammi dei CASI D'USO<sub>G</sub>

<https://www.math.unipd.it/~tullio/IS-1/2018/Dispense/E05b.pdf>

- WEBHOOK<sub>G</sub> di Redmine

<http://www.redmine.org/projects/redmine/wiki/Hooks>

- Webhook di GitLab

<https://docs.gitlab.com/ee/user/project/integrations/Webhooks.html>

- Bot di Telegram

<https://core.telegram.org/bots>

## 2 Descrizione del prodotto

### 2.1 Funzionalità

Le funzionalità offerte dal prodotto sono:

- Ricezione delle segnalazioni provenienti da Redmine e GitLab
- Gestione dei Topic attraverso l'utilizzo del Broker Apache Kafka
- Inoltro dei messaggi verso Telegram ed Email
- Personalizzazione da parte dell'utente finale di impostazioni riguardanti le segnalazioni interessate

### 2.2 Caratteristiche degli utenti

Gli utenti che useranno il prodotto saranno team di sviluppatori software che lavorano abitualmente usando gli strumenti per realizzare CI/CD<sub>G</sub>. Tramite Butterfly, essi potranno configurare la PIPELINE<sub>G</sub> per lo sviluppo del proprio progetto, in modo da ottenere le notifiche in tempo reale direttamente sulle piattaforme di messaggistica selezionate. Nel caso in cui la persona non sia reperibile, verrà effettuata una segnalazione in un apposito calendario sul quale il sistema farà riferimento per l'inoltro alla seconda persona più appropriata a riceverla all'interno del contesto lavorativo.

### 2.3 Tecnologie e scelte relative al progetto

L'obiettivo di questo paragrafo è descrivere in maniera specifica come Butterfly andrà a interfacciarsi con le tecnologie.

Per ciascuna di queste è previsto un MICROSERVIZIO<sub>G</sub>, il quale avrà come scopo quello di fare da tramite tra lo strumento che genera il messaggio e quello che lo riceve.

Questo pattern si chiama PUBLISHER / SUBSCRIBER<sub>G</sub> e utilizza uno strumento intermedio detto Broker per lo smistamento dei messaggi e la gestione dei flussi.

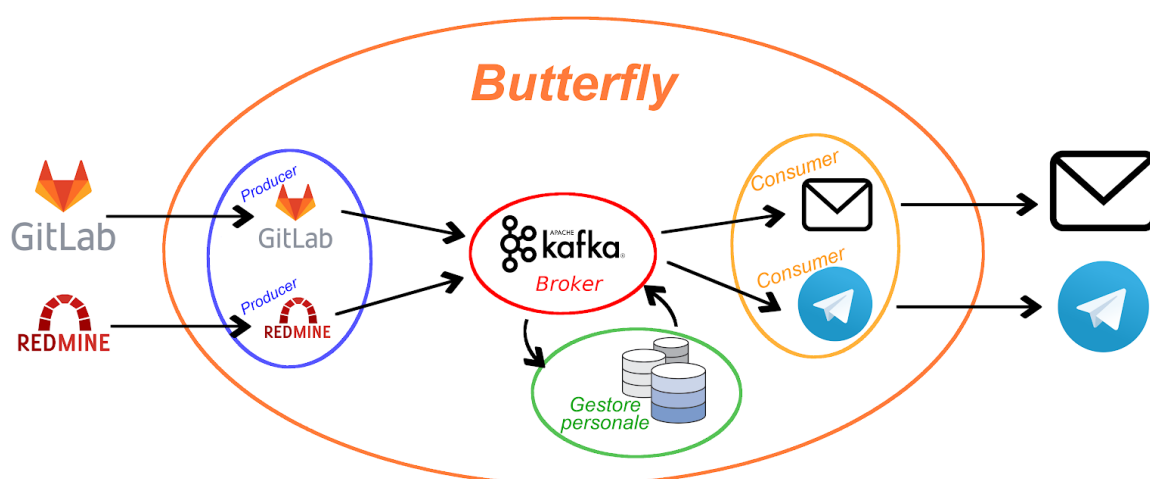


Figura 1: Visione generale del sistema Butterfly

L'immagine precedente rappresenta una suddivisione del sistema in quattro sezioni principali:

- **Arancione:** Butterfly, il sistema nella sua interezza.
- **Blu:** I Producer di GitLab e Redmine.
- **Rosso:** Il Broker. Per Butterfly useremo Apache Kafka.
- **Verde:** Il Gestore Personale.
- **Giallo:** I Consumer di Telegram ed Email

Questa scomposizione ci ha permesso di analizzare più approfonditamente i vari sottosistemi del prodotto in rapporto alle tecnologie esterne a Butterfly (GitLab, Redmine, Telegram ed Email), ai microservizi interni come i Producer / Consumer associati e il Broker.

### 2.3.1 Producer

Per ciascuno degli strumenti che invia messaggi verso il sistema è necessario creare una componente applicativa di tipo Producer che li riceva e li elabori inoltrandoli successivamente verso il Broker. Le tecnologie dalle quali vengono ricevuti i messaggi sono elencate di seguito in ordine di priorità in base a quanto richiesto dalla PROPONENTE<sub>G</sub>.

1. Redmine
2. GitLab
3. SonarQube

Abbiamo scelto di sviluppare i Producer per le prime due applicazioni in base alle priorità suggerite da Imola Informatica nel capitolato, lasciando l'applicativo relativo a SonarQube come opzionale. Ciascun messaggio ricevuto da queste tecnologie verrà analizzato dal Producer associato in modo da inserirli nel Topic appropriato.

#### 2.3.1.1 Producer GitLab

Gli eventi relativi alle REPOSITORY<sub>G</sub> di GitLab presi in considerazione sono:

- Push
- ISSUE<sub>G</sub>, che verranno suddivisi in:
  - Apertura issue
  - Chiusura issue

Le segnalazioni sono generate in base alle KEYWORD<sub>G</sub> contenute nei messaggi di commit e nelle etichette assegnate alle issue. Da quest'ultime verrà creato il Topic corrispondente in caso non sia già esistente.

#### 2.3.1.2 Producer Redmine

Gli eventi relativi ai progetti di Redmine presi in considerazione, sono:

- Issue, che verranno suddivisi in:
  - Apertura issue
  - Chiusura issue

### 2.3.2 Apache Kafka

Il ruolo di Apache Kafka è quello di restare in ascolto dei messaggi provenienti dai Producer, e smistarli in determinati Topic, in base al contenuto degli stessi. Il Gestore Personale potrà poi reperire i messaggi dai Topic specifici e, in base ai Topic ai quali gli utenti sono abbonati, immettere quei messaggi rielaborati alle code specifiche del Broker relative ai Consumer, che si occuperanno di inoltrare il messaggio all'utente finale.

### 2.3.3 Consumer

Come per gli strumenti precedentemente elencati, anche per quelli su cui il messaggio andrà ad essere inoltrato, è necessaria la creazione di un microservizio in grado di fare da tramite tra Broker e `CLIENTG` dello strumento di messaggistica scelto dall'utente. Le tecnologie verso le quali vengono inoltrati i messaggi rielaborati dal sistema sono elencate di seguito in ordine di priorità in base a quanto richiesto dalla proponente.

1. Telegram
2. Email
3. Slack

Allo stesso modo dei Producer, abbiamo scelto di sviluppare i Consumer per le prime due applicazioni in base alle priorità suggeriteci da Imola Informatica nel capitolato, lasciando l'applicativo relativo a Slack come opzionale.

#### 2.3.3.1 Consumer Telegram

Il Consumer Telegram ha il ruolo di restare in ascolto del Topic specifico per Telegram di Apache Kafka e, usando come tramite un bot Telegram dedicato, inoltrare i messaggi agli utenti che hanno impostato Telegram come piattaforma di messaggistica preferita.

Questo richiederà uno step aggiuntivo per gli utenti, che dovranno mettersi in contatto con il bot la prima volta in modo che quest'ultimo abbia il permesso di inviare loro i messaggi tramite la piattaforma.

#### 2.3.3.2 Consumer Email

Il Consumer Email ha il ruolo di restare in ascolto del Topic specifico per le Email di Apache Kafka e, interfacciandosi con un server e-mail dedicato, inoltrare i messaggi agli utenti finali che hanno impostato Email come preferenza per la ricezione.

### 2.3.4 Gestore Personale

Il Gestore Personale è quel `COMPONENTEG` che permette agli utenti di impostare le proprie preferenze e i dati personali tramite interfaccia web, installata in un server interno all'azienda. Questo è accessibile dai dipendenti interessati, i quali potranno quindi modificare, aggiungere e rimuovere, determinate impostazioni riguardanti le proprie preferenze. Quest'ultime sono:

- Scelta dell'applicativo sul quale ricevere le notifiche (Telegram / Email)
- Iscrizione / disiscrizione a determinati Topic
- Aggiunta / rimozione di determinate keyword per i commit di GitLab
- Selezione dei giorni di calendario in cui l'utente segnala la sua assenza
- Aggiunta / rimozione persona di fiducia
- Aggiunta / rimozione / modifica utenti

### 3 Casi d'uso

Questa sezione elenca le funzionalità offerte da Butterfly descritte attraverso il linguaggio UML<sub>G</sub>. Butterfly può essere visto come l'insieme di più sottosistemi che verranno in seguito elencati e che sono stati descritti in modo molto generale anche attraverso la Figura 1.

Questa mostra in maniera un po' più chiarificante la suddivisione e ne facilita l'analisi per la stesura dei casi d'uso. Abbiamo quindi come attori non soltanto le applicazioni che mandano messaggi al sistema, ma anche componenti quali i vari Producer e Consumer che interagiscono col Gestore Personale.

All'interno dei casi d'uso il termine "sistema" è inteso come l'intera applicazione Butterfly, mentre "piattaforma di messaggistica" fa sempre riferimento a Telegram o Email, il termine "persona di fiducia" indica la persona a cui inoltrare le proprie notifiche nei giorni in cui non si è disponibili e con "indisponibilità" e "irreperibilità" di una persona si intendono proprio quei giorni in cui una persona non è raggiungibile nè disponibile per proseguire con le proprie mansioni e quindi non vuole ricevere notifiche.

Il termine "identificativo" fa riferimento indifferentemente al contatto Email o all'ID Telegram.

#### 3.1 Attori

- Redmine
- GitLab
- Producer Redmine
- Producer GitLab
- Utente non acceduto
- Utente non registrato
- Utente (inteso come acceduto e registrato nel sistema)
- Consumer Telegram
- Consumer Email
- Telegram
- Email

#### 3.2 Sistemi e sottosistemi

Per indicare i vari sottosistemi nei codici dei casi d'uso sono allegati delle sigle per rendere più chiara la posizione del caso d'uso all'interno di Butterfly.

- **PR**: Producer Redmine.
- **PG**: Producer GitLab.
- **GP**: Gestore Personale.
- **CT**: Consumer telegram.
- **CE**: Consumer Email.
- **BT**: bot Telegram.
- **SE**: server Email.

### 3.3 Elenco casi d'uso

#### 3.3.1 UC1-PR - Redmine segnala apertura issue al Producer Redmine

- **Codice:** UC1-PR.
- **Titolo:** Redmine segnala apertura issue al Producer Redmine.
- **Attori primari:** Redmine.
- **Descrizione:** Redmine segnala a Butterfly l'apertura di una nuova issue tramite webhook. L'apertura di una issue in un particolare progetto su Redmine contiene i seguenti campi di interesse:
  - Status
  - Tracker
  - Subject
  - Priority e opzionalmente:
    - \* Description
    - \* Assignee

Il campo status conterrà sempre al suo interno “opened”.

- **Precondizione:** Viene aperta una issue su Redmine da segnalare a Butterfly.
- **Postcondizione:** il Producer Redmine riceve la segnalazione da Redmine.
- **Scenario principale:**
  1. Viene aperta una nuova issue su Redmine
  2. Redmine procede all'invio della segnalazione di issue al Producer Redmine

#### 3.3.2 UC2-PR - Redmine segnala la modifica di una issue al Producer Redmine

- **Codice:** UC2-PR.
- **Titolo:** Redmine segnala la modifica di una issue al Producer Redmine.
- **Attori primari:** Redmine.
- **Descrizione:** quando una issue viene modificata, l'invio di tale segnalazione avviene da parte di Redmine tramite webhook. I campi di interesse sono gli stessi dell'apertura di una issue, con la differenza che necessariamente il campo status contiene ora “updated”.
- **Precondizione:** Viene modificata una issue già aperta su un progetto di Redmine da segnalare a Butterfly.
- **Postcondizione:** il Producer Redmine riceve la segnalazione da Redmine.
- **Scenario principale:**
  1. Viene modificata una issue già esistente su Redmine
  2. Redmine procede all'invio della segnalazione di modifica issue al Producer Redmine

### 3.3.3 UC3-PG - GitLab segnala apertura issue al Producer GitLab

- **Codice:** UC3-PG.
- **Titolo:** GitLab segnala apertura issue al Producer GitLab.
- **Attori primari:** GitLab.
- **Descrizione:** GitLab segnala l'apertura di una nuova issue tramite webhook a Butterfly. L'apertura di una issue su GitLab contiene:
  - Object kind
  - Action
  - Project
  - Title e opzionalmente:
    - \* Label
    - \* Milestone
    - \* Assignees
    - \* Due Date

Il campo object kind ci permette di capire il tipo di oggetto, in questo caso infatti contiene "issue", mentre il campo action ha "open".

- **Precondizione:** viene aperta una issue su GitLab da segnalare a Butterfly.
- **Postcondizione:** il Producer GitLab riceve la segnalazione da GitLab.
- **Scenario principale:**
  1. Viene aperta una nuova issue su GitLab
  2. GitLab procede all'invio della segnalazione di issue al Producer GitLab

### 3.3.4 UC4-PG - Gitlab segnala la modifica di una issue al Producer Gitlab

- **Codice:** UC4-PG.
- **Titolo:** Gitlab segnala la modifica di una issue al Producer Gitlab.
- **Attori primari:** GitLab.
- **Descrizione:** GitLab segnala la modifica di una issue esistente tramite webhook a Butterfly. I campi di interesse sono gli stessi dell'apertura issue, con la differenza che il campo action contiene "update".
- **Precondizione:** Viene modificata una issue già aperta su un progetto di GitLab da segnalare a Butterfly.
- **Postcondizione:** il Producer GitLab riceve la segnalazione da GitLab.
- **Scenario principale:**
  1. Viene modificata una issue già esistente su GitLab
  2. GitLab procede all'invio della segnalazione di modifica issue al Producer GitLab

### 3.3.5 UC5-PG - GitLab segnala un evento di push a Producer GitLab

- **Codice:** UC5-PG.
- **Titolo:** GitLab segnala un evento di push a Producer GitLab.
- **Attori primari:** GitLab.
- **Descrizione:** GitLab segnala un evento di push tramite webhook a Butterfly. L'evento di push può essere composto da uno o più commit. I campi di interesse sono:
  - Object kind
  - Project
  - Repository
  - Commits

E in questo caso il campo object kind contiene "push".

- **Precondizione:** Viene effettuato un push su GitLab da segnalare a Butterfly.
- **Postcondizione:** il Producer GitLab riceve la segnalazione da GitLab.
- **Scenario principale:**
  1. Viene effettuato un push in GitLab
  2. GitLab procede all'invio della segnalazione di push al Producer GitLab

### 3.3.6 UC6-GP - Producer Redmine invia messaggio al Gestore Personale

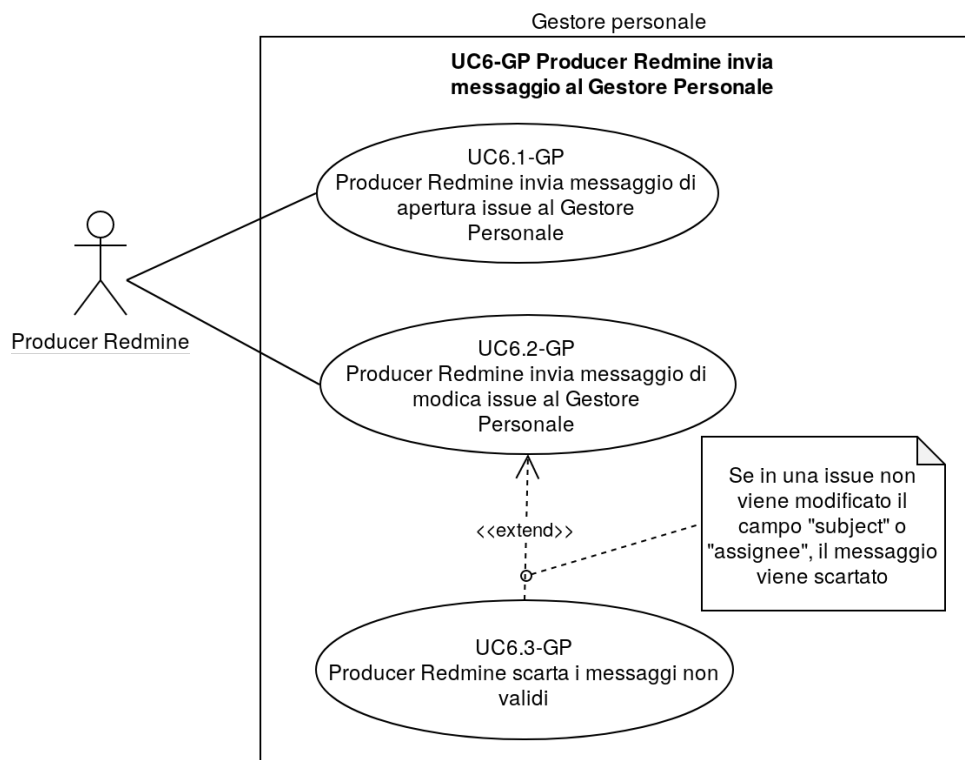


Figura 2: UC6-GP - Producer Redmine invia messaggio al Gestore Personale

- **Codice:** UC6-GP.



- **Titolo:** Producer Redmine invia messaggio al Gestore Personale.
- **Attori Pimari:** Producer Redmine.
- **Descrizione:** il Producer Redmine, dopo aver ricevuto una segnalazione da Redmine, elabora il messaggio e lo invia al Gestore Personale. Il messaggio finale, una volta terminata l'elaborazione, conterrà i campi:
  - Project
  - Topic
  - Subject e opzionalmente:
    - \* Description

Il messaggio non può contenere campi errati o mancanti per il corretto funzionamenti di Butterfly, perché i campi selezionati sono obbligatori per l'apertura di una issue su Redmine.

- **Precondizione:** il Producer Redmine ha ricevuto una segnalazione da Redmine.
- **Postcondizione:** il Producer Redmine ha elaborato e inviato al Gestore Personale il messaggio.
- **Scenario principale:**
  1. Il Producer Redmine riceve la segnalazione di issue da Redmine
  2. Il Producer Redmine prepara il messaggio di issue in modo che venga inserito correttamente nel Gestore Personale
  3. Il Producer Redmine invia il messaggio di issue al Gestore Personale

### 3.3.7 UC6.1-GP - Producer Redmine invia messaggio di apertura issue al Gestore Personale

- **Codice:** UC6.1-GP.
- **Titolo:** Producer Redmine invia messaggio di apertura issue al Gestore Personale.
- **Attori Pimari:** Producer Redmine.
- **Descrizione:** il Producer Redmine, dopo aver ricevuto una segnalazione di apertura issue da Redmine, elabora il messaggio e lo invia al Gestore Personale.
- **Precondizione:** il Producer Redmine ha ricevuto una segnalazione da Redmine.
- **Postcondizione:** il Producer Redmine ha elaborato e inviato al Gestore Personale il messaggio di apertura issue.
- **Scenario Pincipale:**
  1. Il Producer Redmine riceve la segnalazione di apertura issue da Redmine
  2. Il Producer Redmine prepara il messaggio in modo che venga inserito correttamente nel Gestore Personale
  3. Il Producer Redmine invia il messaggio di apertura issue al Gestore Personale

### 3.3.8 UC6.2-GP - Producer Redmine invia messaggio di modifica issue al Gestore Personale

- **Codice:** UC6.2-GP.
- **Titolo:** Producer Redmine invia messaggio di modifica issue al Gestore Personale
- **Attori Pimari:** Producer Redmine.
- **Descrizione:** il Producer Redmine, dopo aver ricevuto una segnalazione di modifica issue da Redmine, elabora il messaggio e lo invia al Gestore Personale.
- **Precondizione:** il Producer Redmine ha ricevuto una segnalazione da Redmine.
- **Postcondizione:** il Producer Redmine ha elaborato e inviato al Gestore Personale il messaggio di modifica issue.
- **Scenario Pincipale:**
  1. Il Producer Redmine riceve la segnalazione di modifica issue da Redmine
  2. Il Producer Redmine prepara il messaggio in modo che venga inserito correttamente nel Gestore Personale
  3. Il Producer Redmine invia il messaggio di modifica issue al Gestore Personale
- **Estensioni:**
  1. Se ci sono dei messaggi non validi, questi vengono scartati [UC6.3-GP].

### 3.3.9 UC6.3-GP - Producer Redmine scarta i messaggi non validi

- **Codice:** UC6.3-GP.
- **Titolo:** Producer Redmine scarta i messaggi non validi.
- **Attori primari:** Producer Redmine.
- **Descrizione:** il Producer Redmine, dopo aver ricevuto una segnalazione di una modifica issue da Redmine, controlla se è stato modificato il campo “subject” o “assignee”. In caso negativo, il messaggio viene scartato.
- **Precondizione:** il Producer Redmine ha ricevuto una segnalazione da GitLab.
- **Postcondizione:** il Producer Redmine ha scartato il messaggio.
- **Scenario principale:**
  1. Il Producer Redmine riceve la segnalazione di modifica issue da Redmine
  2. Il Producer Redmine vede che non è stato modificato il campo “subject” o “assignee”
  3. Il Producer Redmine scarta il messaggio non valido.

### 3.3.10 UC7-GP - Producer GitLab invia messaggio al Gestore Personale

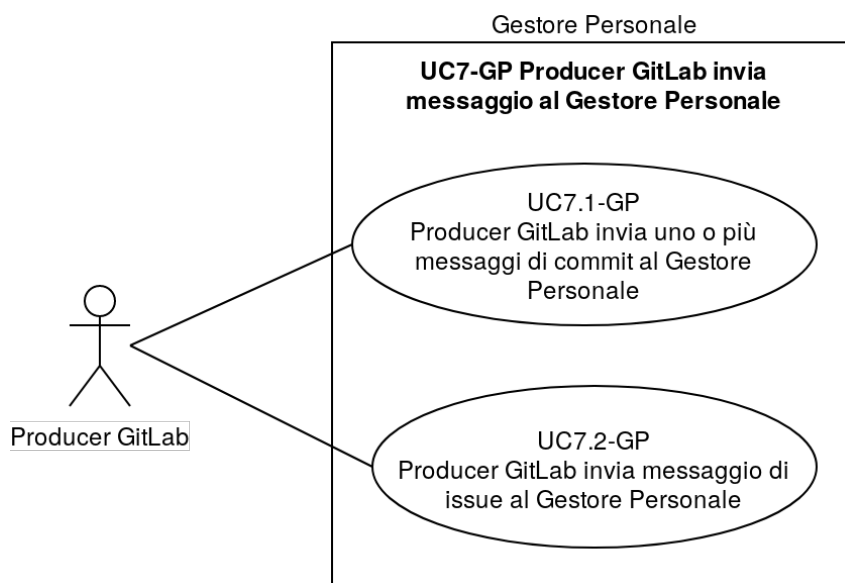


Figura 3: UC7-GP - Producer GitLab invia messaggio al Gestore Personale

- **Codice:** UC7-GP.
- **Titolo:** Producer GitLab invia messaggio al Gestore Personale.
- **Attori primari:** Producer GitLab.
- **Descrizione:** il Producer GitLab, dopo aver ricevuto una segnalazione da GitLab, elabora un messaggio da inviare al Gestore Personale.
- **Precondizione:** il Producer GitLab ha ricevuto una segnalazione da GitLab.
- **Postcondizione:** il Producer GitLab ha inviato al Gestore Personale il messaggio elaborato.
- **Scenario principale:**
  1. Il Producer GitLab riceve una segnalazione da GitLab
  2. Il Producer GitLab prepara il messaggio in modo che venga inserito correttamente nel Gestore Personale
  3. Il Poducer GitLab invia il messaggio al Gestore Personale

### 3.3.11 UC7.1-GP - Producer GitLab invia uno o più messaggi di commit al Gestore Personale

- **Codice:** UC7.1-GP.
- **Titolo:** Producer GitLab invia uno o più messaggi di commit al Gestore Personale.
- **Attori primari:** Producer GitLab.
- **Descrizione:** il Producer GitLab, dopo aver ricevuto una segnalazione di push da GitLab, elabora un messaggio particolare per i commit che verrà catalogato sotto il Topic "commits". Il messaggio elaborato conterrà i campi:
  - Project
  - Topic
  - Message
- **Precondizione:** il Producer GitLab ha ricevuto una segnalazione da GitLab.
- **Postcondizione:** il Producer GitLab ha inviato uno o più messaggi elaborati di commit.
- **Scenario principale:**
  1. Il Producer GitLab riceve la segnalazione di uno o più commit da GitLab
  2. Il Producer GitLab prepara i messaggi in modo che vengano inseriti correttamente nel Gestore Personale
  3. Il Producer GitLab invia i messaggi di commit al Gestore Personale

### 3.3.12 UC7.2-GP - Producer GitLab invia messaggio di issue al Gestore Personale

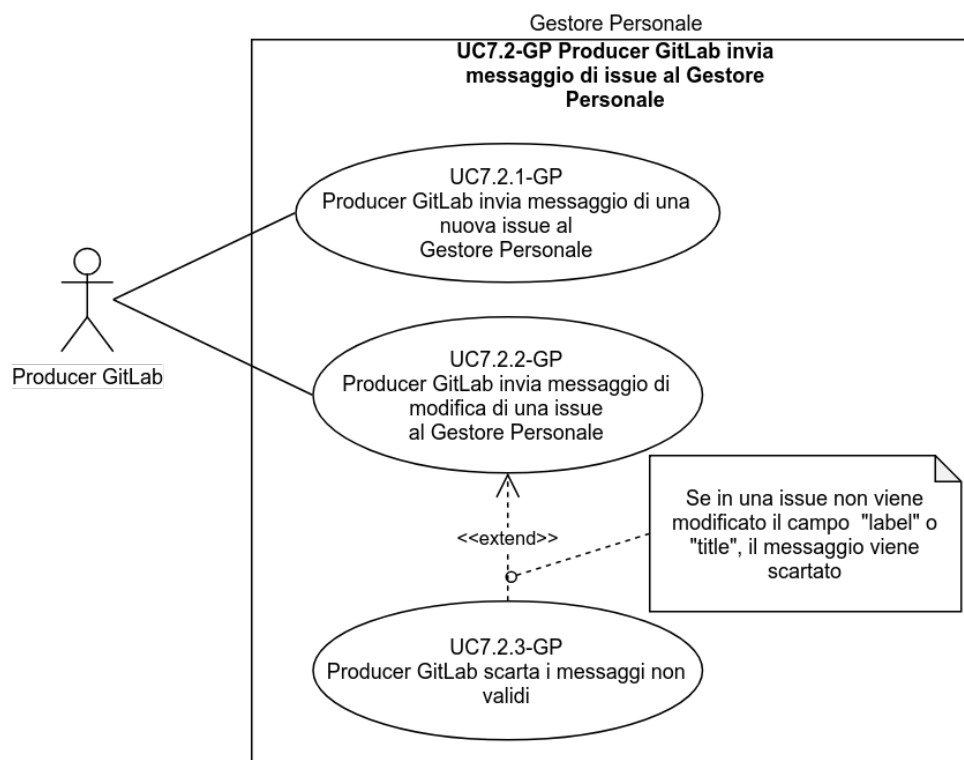


Figura 4: UC7.2-GP - Producer GitLab invia messaggio di issue al Gestore Personale

- **Codice:** UC7.2-GP.
- **Titolo:** Producer GitLab invia messaggio di issue al Gestore Personale.
- **Attori primari:** Producer GitLab.
- **Descrizione:** il Producer GitLab, dopo aver ricevuto una segnalazione di issue da GitLab, controlla se la issue è appena stata creata o si tratta della modifica di una issue preesistente. Il messaggio, una volta elaborato, conterrà i campi:
  - Project
  - Topic
  - Subject e opzionalmente:
    - \* Description
    - \* Due Date
    - \* Milestone
    - \* Assignee
- **Precondizione:** il Producer GitLab ha ricevuto una segnalazione da GitLab.
- **Postcondizione:** il Producer GitLab ha inviato al Gestore Personale il messaggio elaborato.
- **Scenario principale:**
  1. Il Producer GitLab riceve la segnalazione di issue da GitLab
  2. Il Producer GitLab prepara il messaggio di issue in modo che venga inserito correttamente nel Gestore Personale
  3. Il Producer GitLab invia il messaggio di issue al Gestore Personale

### 3.3.13 UC7.2.1-GP - Producer GitLab invia messaggio di una nuova issue al Gestore Personale

- **Codice:** UC7.2.1-GP.
- **Titolo:** Producer GitLab invia messaggio di una nuova issue al Gestore Personale.
- **Attori primari:** Producer GitLab.
- **Descrizione:** il Producer GitLab, dopo aver ricevuto una segnalazione di una nuova issue da GitLab, elabora il messaggio da inoltrare al Gestore Personale.
- **Precondizione:** il Producer GitLab ha ricevuto una segnalazione da GitLab.
- **Postcondizione:** il Producer GitLab ha inviato al Gestore Personale il messaggio elaborato di nuova issue.
- **Scenario principale:**
  1. Il Producer GitLab riceve la segnalazione di una nuova issue da GitLab
  2. Il Producer GitLab prepara il messaggio di una nuova issue in modo che venga inserito correttamente nel Gestore Personale
  3. Il Producer GitLab invia il messaggio di una nuova issue al Gestore Personale

### 3.3.14 UC7.2.2-GP - Producer GitLab invia messaggio di modifica di una issue al Gestore Personale

- **Codice:** UC7.2.2-GP.
- **Titolo:** Producer GitLab invia messaggio di modifica di una issue al Gestore Personale.
- **Attori primari:** Producer GitLab.
- **Descrizione:** il Producer GitLab, dopo aver ricevuto una segnalazione di modifica di una issue da GitLab, controlla se sono stati modificati i campi “label” o “title”. In caso positivo, viene inviato un messaggio elaborato al Gestore Personale.
- **Precondizione:** il Producer GitLab ha ricevuto una segnalazione da GitLab.
- **Postcondizione:** il Producer GitLab ha inviato al Gestore Personale il messaggio elaborato di modifica issue.
- **Scenario principale:**
  1. Il Producer GitLab riceve la segnalazione di modifica issue da GitLab
  2. Il Producer GitLab prepara il messaggio di modifica issue in modo che venga inserito correttamente nel Gestore Personale
  3. Il Poducer GitLab invia il messaggio di modifica issue al Gestore Personale
- **Estensioni:**
  1. Se ci sono dei messaggi non validi, questi vengono scartati [UC7.2.3-GP].

### 3.3.15 UC7.2.3-GP - Producer GitLab scarta i messaggi non validi

- **Codice:** UC7.2.3-GP.
- **Titolo:** Producer GitLab scarta i messaggi non validi.
- **Attori primari:** Producer GitLab.
- **Descrizione:** il Producer GitLab, dopo aver ricevuto una segnalazione di una modifica issue da GitLab, controlla se sono stati modificati i campi “label” o “title”. In caso negativo, il messaggio viene scartato.
- **Precondizione:** il Producer GitLab ha ricevuto una segnalazione da GitLab.
- **Postcondizione:** il Producer GitLab ha scartato il messaggio.
- **Scenario principale:**
  1. Il Producer GitLab riceve la segnalazione di modifica issue da GitLab
  2. Il Producer GitLab vede che non sono stati modificati i campi “label” o “title”
  3. Il Producer GitLab scarta i messaggi non validi

### 3.3.16 UC8-CT - Gestore Personale invia il messaggio finale al Consumer Telegram

- **Codice:** UC8-CT.
- **Titolo:** Gestore Personale invia il messaggio finale al Consumer Telegram.
- **Attori primari:** Gestore Personale.
- **Descrizione:** il Gestore Personale, dopo aver ricevuto il messaggio elaborato dal Producer Redmine o GitLab, controlla i Topic del messaggio, gli utenti iscritti, la loro disponibilità e se la loro preferenza è Telegram. Se tutte queste condizioni sono verificate, viene preparato il messaggio finale da inviare successivamente viene inviato al Consumer Telegram. Se il destinatario è iscritto a quel Topic ma non è disponibile, il destinatario viene cambiato con la persona di fiducia.

Il messaggio finale, una volta elaborato, conterrà i campi:

- Id della chat del destinatario
- Applicazione di provenienza
- Ora di invio
- Tipo di segnalazione(commit o issue)
- Project
- Topic
- Subject e opzionalmente
  - \* Description
  - \* Due date
  - \* Milestone
  - \* Assignee
- **Precondizione:** il Gestore Personale ha ricevuto il messaggio elaborato dal Producer Redmine o GitLab.
- **Postcondizione:** Il Gestore Personale ha inviato il messaggio finale al Consumer Telegram.
- **Scenario principale:**
  1. Il Gestore Personale riceve un messaggio dal Producer Redmine o dal Producer GitLab
  2. Il Gestore Personale valuta quali utenti sono iscritti al Topic del messaggio ricevuto e se vogliono ricevere il messaggio tramite Telegram
  3. Il Gestore Personale procede all'invio del messaggio finale al Consumer Telegram

### 3.3.17 UC9-CE - Gestore Personale invia il messaggio finale al Consumer Email

- **Codice:** UC9-CE.
- **Titolo:** Gestore Personale invia il messaggio finale al Consumer Email.
- **Attori primari:** Gestore Personale.
- **Descrizione:** il Gestore Personale, dopo aver ricevuto il messaggio elaborato dal Producer Redmine o GitLab, controlla i Topic del messaggio, gli utenti iscritti, la loro disponibilità e se la loro preferenza è Email. Se tutte queste condizioni sono verificate, viene preparato il messaggio finale da inviare successivamente viene inviato al Consumer Email. Se il destinatario è iscritto a quel Topic ma non è disponibile, il destinatario viene cambiato con la persona di fiducia.

Il messaggio finale, una volta elaborato, conterrà i campi:

- Email del destinatario
  - Applicazione di provenienza
  - Ora di invio
  - Tipo di segnalazione(commit, issue)
  - Project
  - Topic
  - Subject e opzionalmente
    - \* Description
    - \* Due date
    - \* Milestone
    - \* Assignee
- **Precondizione:** il Gestore Personale ha ricevuto il messaggio elaborato dal Producer Redmine o GitLab.
  - **Postcondizione:** Il Gestore Personale ha inviato il messaggio finale al Consumer Email.
  - **Scenario principale:**
    1. Il Gestore Personale riceve un messaggio dal Producer Redmine o dal Producer GitLab
    2. Il Gestore Personale valuta quali utenti sono iscritti al Topic del messaggio ricevuto e se vogliono ricevere il messaggio tramite Email
    3. Il Gestore Personale procede all'invio del messaggio finale al Consumer Email

### 3.3.18 UC10-BT - Consumer Telegram inoltra il messaggio finale al bot Telegram

- **Codice:** UC10-BT.
- **Titolo:** Consumer Telegram inoltra il messaggio finale al bot Telegram.
- **Attori primari:** Consumer Telegram.
- **Descrizione:** il Consumer Telegram inoltra il messaggio finale al bot Telegram, il quale notifica il destinatario finale attraverso Telegram.
- **Precondizione:** il Consumer Telegram ha ricevuto almeno un messaggio.
- **Postcondizione:** il bot Telegram ha ricevuto il messaggio finale.
- **Scenario principale:**
  1. Il Consumer Telegram riceve un messaggio dal Gestore Personale
  2. Il Consumer Telegram inoltra il messaggio finale al bot Telegram

### 3.3.19 UC11-SE - Consumer Email inoltra il messaggio finale al server Email

- **Codice:** UC11-SE.
- **Titolo:** Consumer Email inoltra il messaggio finale al server Email.
- **Attori primari:** Consumer Email.
- **Descrizione:** il Consumer Email inoltra il messaggio finale al server Email, il quale notifica il destinatario finale attraverso una mail.



- **Precondizione:** il Consumer Email ha ricevuto almeno un messaggio.
- **Postcondizione:** il server Email ha ricevuto il messaggio finale con successo.
- **Scenario principale:**
  1. Il Consumer Email riceve un messaggio dal Gestore Personale
  2. Il Consumer Email procede all'inoltro del messaggio finale al server Email

### 3.3.20 UC12-GP - Accesso

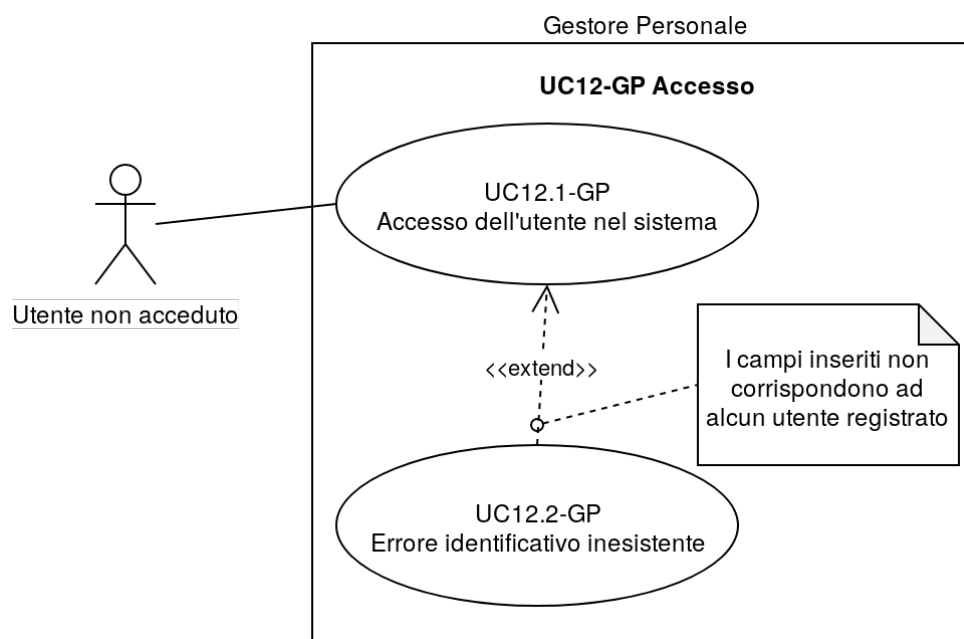


Figura 5: UC12-GP - Accesso

- **Codice:** UC12-GP.
- **Titolo:** accesso.
- **Attori primari:** utente non acceduto.
- **Descrizione:** l'utente non acceduto richiede di accedere al sistema attraverso l'inserimento del proprio identificativo.
- **Precondizione:** il sistema considera l'utilizzatore come un utente non acceduto.
- **Postcondizione:** il sistema riconosce l'utilizzatore di esso come utente acceduto.
- **Scenario principale:**
  1. L'utente non ancora riconosciuto dal sistema procede all'accesso inserendo il proprio identificativo

### 3.3.21 UC12.1-GP - Accesso dell'utente nel sistema

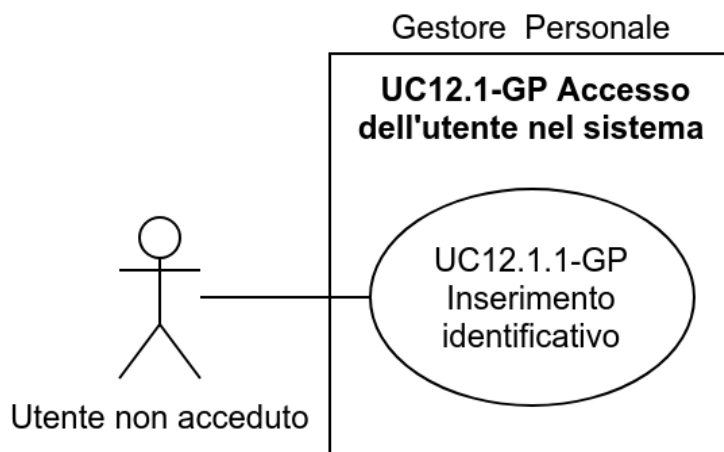


Figura 6: UC12.1-GP - Accesso dell'utente nel sistema

- **Codice:** UC12.1-GP.
- **Titolo:** accesso dell'utente nel sistema.
- **Attori primari:** utente non acceduto.
- **Descrizione:** l'utente non acceduto accede al sistema inserendo il proprio identificativo e inviandone la conferma.
- **Precondizione:** il sistema riconosce l'utilizzatore come un utente non acceduto.
- **Postcondizione:** il sistema riconosce l'utente con successo.
- **Scenario principale:**
  1. L'utente non ancora riconosciuto dal sistema richiede l'accesso inserendo il proprio identificativo.
  2. L'utente conferma l'invio dei dati
- **Estensioni:**
  1. L'accesso non va a buon fine e viene visualizzato un errore avvisando l'utente [UC12.2-GP]

### 3.3.22 UC12.1.1-GP - Inserimento identificativo

- **Codice:** UC12.1.1-GP.
- **Titolo:** inserimento identificativo.
- **Attori primari:** utente non acceduto.
- **Descrizione:** l'utente inserisce l'identificativo.
- **Precondizione:** il sistema riconosce l'utilizzatore come un utente non acceduto e permette all'utente l'inserimento dell'identificativo.
- **Postcondizione:** l'utente ha inserito l'identificativo desiderato dal sistema.
- **Scenario principale:**
  1. L'utente inserisce l'identificativo per autenticarsi

### 3.3.23 UC12.2-GP - Errore identificativo inesistente

- **Codice:** UC12.2-GP.
- **Titolo:** errore identificativo inesistente.
- **Attori primari:** utente non acceduto.
- **Descrizione:** l'utente viene avvisato che ha inserito un identificativo non riconosciuto.
- **Precondizione:** il sistema riconosce l'utilizzatore come un utente non acceduto e riceve una richiesta di accesso.
- **Postcondizione:** il sistema comunica all'utilizzatore l'errore.
- **Scenario principale:**
  1. L'utente inserisce un identificativo non esistente
  2. L'utente conferma l'invio dei dati
  3. L'utente visualizza il messaggio d'errore

### 3.3.24 UC13-GP - Uscita dell'utente dal sistema

- **Codice:** UC13-GP.
- **Titolo:** uscita dell'utente dal sistema.
- **Attori primari:** utente.
- **Descrizione:** l'utente esce dal sistema ed ha la possibilità di rientrarci come un diverso utente o come lo stesso di prima.
- **Precondizione:** l'utente è all'interno del sistema.
- **Postcondizione:** l'utente è ora un utente non acceduto e si trova a poter accedere nuovamente nel sistema.
- **Scenario principale:**
  1. L'utente riconosciuto dal sistema effettua l'uscita dallo stesso

### 3.3.25 UC14-GP - Aggiunta nuovo utente

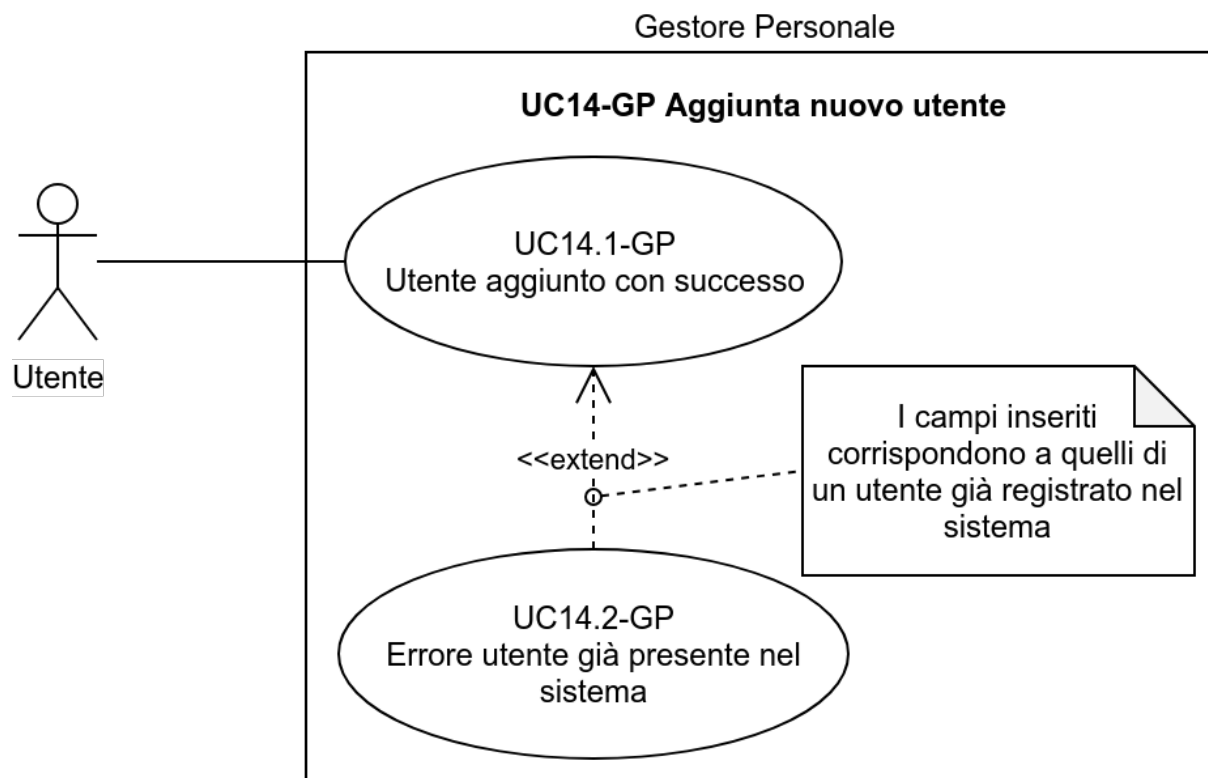


Figura 7: UC14-GP - Aggiunta nuovo utente

- **Codice:** UC14-GP.
- **Titolo:** aggiunta nuovo utente.
- **Attori primari:** utente.
- **Descrizione:** l'utente, acceduto al sistema, aggiunge un nuovo utente nel sistema. Non è possibile che un utente non acceduto si iscriva da solo per la prima volta.
- **Precondizione:** un nuovo utente deve essere aggiunto nel sistema.
- **Postcondizione:** un utente viene aggiunto al sistema.
- **Scenario principale:**
  1. L'utente aggiunge un nuovo utente

### 3.3.26 UC14.1-GP - Utente aggiunto con successo

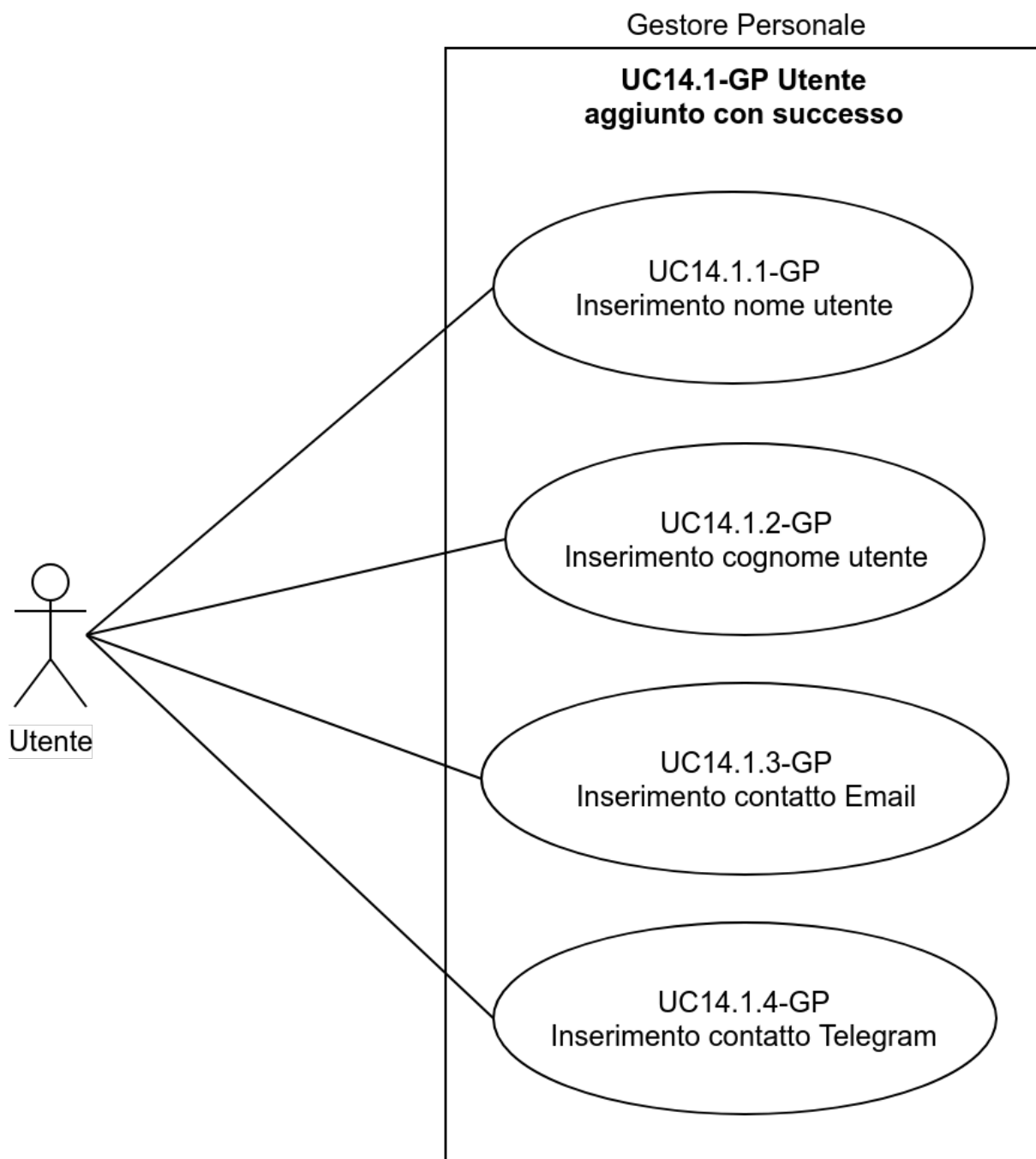


Figura 8: UC14.1-GP - Utente aggiunto con successo

- **Codice:** C14.1-GP.
- **Titolo:** utente aggiunto con successo.
- **Attori primari:** utente.
- **Descrizione:** un nuovo utente viene inserito con successo nel sistema.
- **Precondizione:** un nuovo utente deve essere aggiunto nel sistema.
- **Postcondizione:** un utente viene aggiunto con successo al sistema.

- **Scenario principale:**

1. L'utente inserisce i dati del nuovo utente da aggiungere
2. L'utente conferma l'invio dei dati
3. L'aggiunta viene effettuata con successo

- **Estensioni:**

1. Errore utente già presente nel sistema [UC14.2-GP]

### 3.3.27 UC14.1.1-GP - Inserimento nome utente

- **Codice:** UC14.1.1-GP.
- **Titolo:** inserimento nome utente.
- **Attori primari:** utente.
- **Descrizione:** l'utente inserisce il nominativo dell'utente da aggiungere.
- **Precondizione:** un nuovo utente deve essere aggiunto nel sistema.
- **Postcondizione:** il nome del nuovo utente è stato aggiunto nel sistema.
- **Scenario principale:**

1. L'utente aggiunge il nominativo del nuovo utente

### 3.3.28 UC14.1.2-GP - Inserimento cognome utente

- **Codice:** UC14.1.2-GP.
- **Titolo:** inserimento cognome utente.
- **Attori primari:** utente.
- **Descrizione:** l'utente inserisce il cognome dell'utente da aggiungere.
- **Precondizione:** un nuovo utente deve essere aggiunto nel sistema.
- **Postcondizione:** il cognome del nuovo utente è stato aggiunto nel sistema.
- **Scenario principale:**

1. L'utente aggiunge il cognome del nuovo utente

### 3.3.29 UC14.1.3-GP - Inserimento contatto Email

- **Codice:** UC14.1.3-GP.
- **Titolo:** inserimento contatto Email.
- **Attori primari:** utente.
- **Descrizione:** l'utente inserisce il contatto Email dell'utente da aggiungere.
- **Precondizione:** un nuovo utente deve essere aggiunto nel sistema.
- **Postcondizione:** il contatto Email è stato aggiunto.
- **Scenario principale:**

1. L'utente aggiunge il contatto Email del nuovo utente

### 3.3.30 UC14.1.4-GP - Inserimento contatto Telegram

- **Codice:** UC14.1.4-GP.
- **Titolo:** inserimento contatto Telegram.
- **Attori primari:** utente.
- **Descrizione:** l'utente inserisce il contatto Telegram dell'utente da aggiungere.
- **Precondizione:** un nuovo utente deve essere aggiunto nel sistema.
- **Postcondizione:** il contatto Telegram è stato aggiunto.
- **Scenario principale:**
  1. L'utente aggiunge il contatto Telegram del nuovo utente

### 3.3.31 UC14.2-GP - Errore utente già presente nel sistema

- **Codice:** UC14.2-GP.
- **Titolo:** errore utente già presente nel sistema.
- **Attori primari:** utente.
- **Descrizione:** l'utente viene avvisato che i contatti Telegram o Email immessi non sono univoci.
- **Precondizione:** un nuovo utente deve essere aggiunto nel sistema.
- **Postcondizione:** il sistema comunica all'utilizzatore l'errore e l'utente non viene inserito.
- **Scenario principale:**
  1. L'utente visualizza il messaggio d'errore

### 3.3.32 UC15-GP - Rimozione utente dal sistema

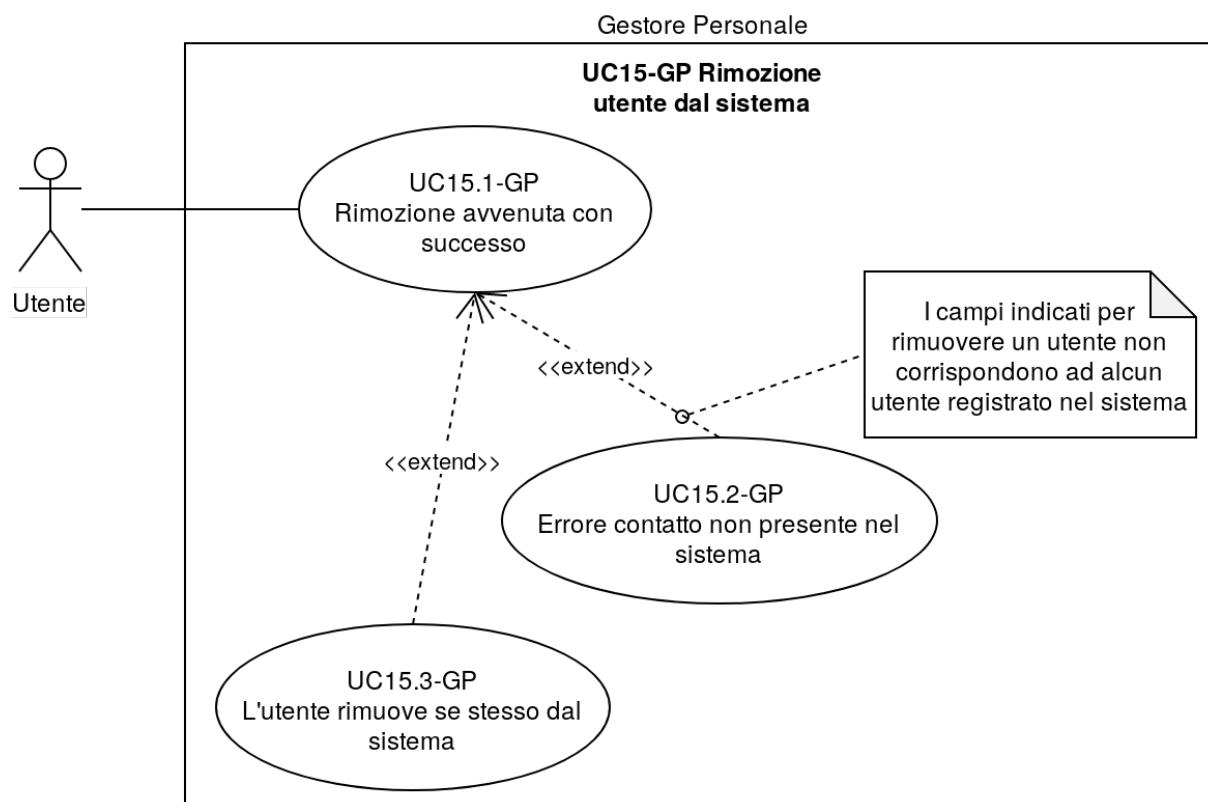


Figura 9: UC15-GP - Rimozione utente dal sistema

- **Codice:** UC15-GP.
- **Titolo:** rimozione utente dal sistema.
- **Attori primari:** utente.
- **Descrizione:** l'utente rimuove l'utente selezionato dal sistema.
- **Precondizione:** un utente già presente nel sistema deve essere rimosso.
- **Postcondizione:** un utente viene rimosso dal sistema.
- **Scenario principale:**
  1. L'utente rimuove l'utente selezionato



### 3.3.33 UC15.1-GP - Rimozione avvenuta con successo

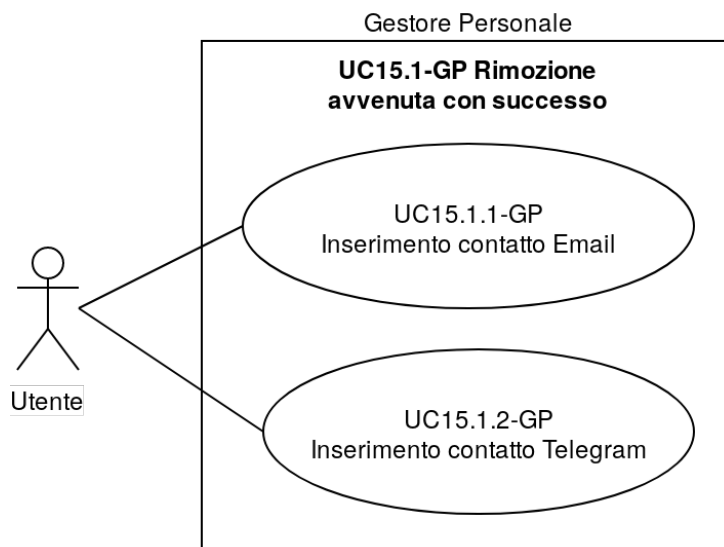


Figura 10: UC15.1-GP - Rimozione avvenuta con successo

- **Codice:** UC15.1-GP.
- **Titolo:** rimozione avvenuta con successo.
- **Attori primari:** utente.
- **Descrizione:** un utente, acceduto al sistema, rimuove un altro utente presente nel sistema tramite l'inserimento del suo contatto Email o Telegram.
- **Precondizione:** un utente già presente nel sistema deve essere rimosso.
- **Postcondizione:** un utente con il contatto Email o Telegram inserito viene rimosso dal sistema.
- **Scenario principale:**
  1. L'utente inserisce ciò che è richiesto dal sistema
  2. L'utente conferma l'invio dei dati
  3. L'utente da rimuovere è stato rimosso dal sistema
- **Estensioni:**
  - Errore contatto non presente nel sistema [UC15.2-GP]
  - L'utente vuole rimuovere se stesso [UC15.3-GP]

### 3.3.34 UC15.1.1-GP - Inserimento contatto Email

- **Codice:** UC15.1.1-GP.
- **Titolo:** inserimento contatto Email.
- **Attori primari:** utente.
- **Descrizione:** l'utente ha aggiunto il contatto Email relativo all'utente che vuole rimuovere.

- **Precondizione:** un utente già presente nel sistema deve essere rimosso.
- **Postcondizione:** il contatto dell'utente da rimuovere Email è stato inserito.
- **Scenario principale:**
  1. L'utente inserisce il contatto Email dell'utente da rimuovere

### 3.3.35 UC15.1.2-GP - Inserimento contatto Telegram

- **Codice:** UC15.1.2-GP.
- **Titolo:** inserimento contatto Telegram.
- **Attori primari:** utente.
- **Descrizione:** l'utente ha aggiunto il contatto Telegram relativo all'utente che vuole rimuovere.
- **Precondizione:** un utente già presente nel sistema deve essere rimosso.
- **Postcondizione:** il contatto Telegram dell'utente da rimuovere è stato inserito.
- **Scenario principale:**
  1. L'utente inserisce il contatto Telegram del utente da rimuovere.

### 3.3.36 UC15.2-GP - Errore contatto non presente nel sistema

- **Codice:** UC15.2-GP.
- **Titolo:** errore contatto non presente nel sistema.
- **Attori primari:** utente.
- **Descrizione:** l'utente viene avvisato che il contatto inserito non è presente nel sistema.
- **Precondizione:** un utente già presente nel sistema deve essere rimosso.
- **Postcondizione:** il sistema comunica all'utente che utilizza il sistema l'errore e nessun utente viene rimosso.
- **Scenario principale:**
  1. L'utente selezionato attraverso il contatto Telegram o Email non viene rimosso perché non presente nel sistema.

### 3.3.37 UC15.3-GP - L'utente rimuove se stesso dal sistema

- **Codice:** UC15.3-GP.
- **Titolo:** l'utente rimuove se stesso dal sistema.
- **Attori primari:** utente.
- **Descrizione:** l'utente inserisce il proprio identificativo per rimuoversi dal sistema.
- **Precondizione:** un utente già presente nel sistema deve essere rimosso.
- **Postcondizione:** l'utente non risulta più acceduto al sistema che non riconosce più l'utente ormai rimosso.

- **Scenario principale:**

1. L'utente selezionato attraverso il contatto Telegram o Email viene rimosso dal sistema.
2. L'utente viene automaticamente fatto uscire dall'applicazione.

### 3.3.38 UC16-GP - Modifica utente

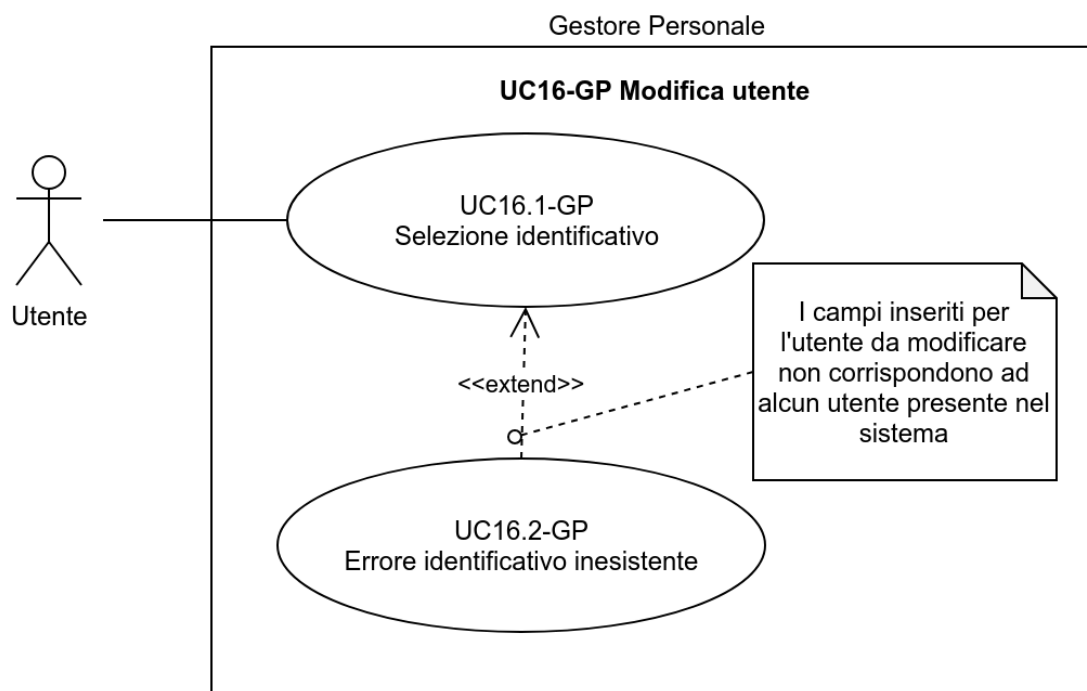


Figura 11: UC16-GP - Modifica utente

- **Codice:** UC16-GP.
- **Titolo:** modifica utente.
- **Attori primari:** utente.
- **Descrizione:** l'utente vuole modificare le informazioni relative a un altro utente, o di se stesso.
- **Precondizione:** l'utente vuole modificare i dati di un utente già presente nel sistema.
- **Postcondizione:** i campi dell'utente sono stati modificati correttamente.
- **Scenario principale:**
  1. L'utente modifica i dati relativi ad un utente

### 3.3.39 UC16.1-GP - Selezione identificativo

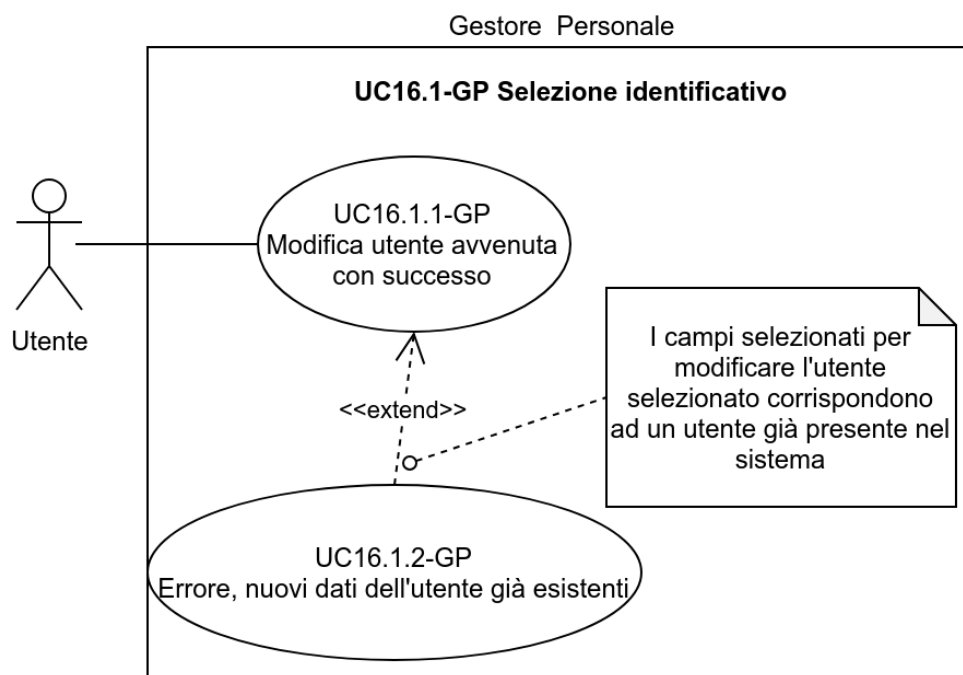


Figura 12: UC16.1-GP - Selezione identificativo

- **Codice:** UC16.1-GP.
- **Titolo:** selezione identificativo.
- **Attori primari:** utente.
- **Descrizione:** l'utente aggiunge l'identificativo dell'utente che vuole modificare.
- **Precondizione:** l'utente vuole modificare un utente già presente.
- **Postcondizione:** l'identificativo è stato inserito.
- **Scenario principale:**
  1. L'utente procede con l'inserimento dell'identificativo dell'utente da modificare
- **Estensioni:**
  - Errore identificativo inesistente [UC16.2-GP]

### 3.3.40 UC16.1.1-GP - Modifica utente avvenuta con successo

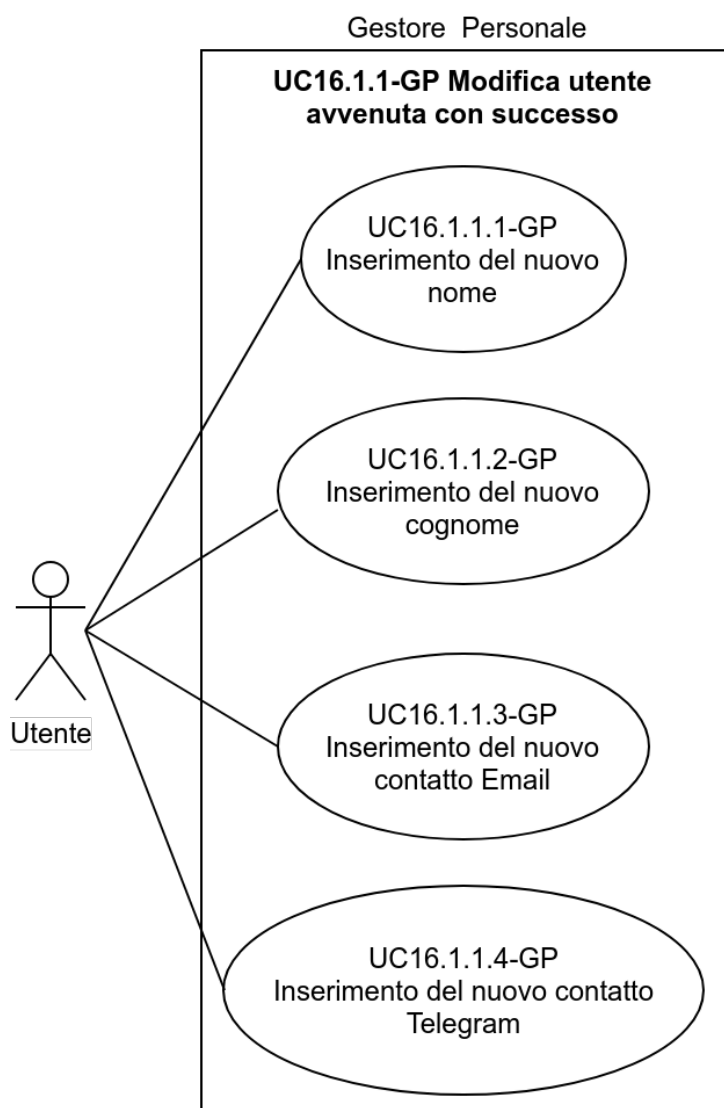


Figura 13: UC16.1.1-GP - Modifica utente avvenuta con successo

- **Codice:** UC16.1.1-GP.
- **Titolo:** modifica utente avvenuta con successo.
- **Attori primari:** utente.
- **Descrizione:** l'identificativo è presente nel sistema e ne vengono modificati i relativi campi con successo.
- **Precondizione:** l'utente vuole modificare un utente già presente nel sistema.
- **Postcondizione:** l'utente è stato modificato con successo.
- **Scenario principale:**
  1. L'utente viene modificato con successo
- **Estensioni:**
  - Errore, nuovi dati dell'utente già esistenti [UC16.1.2-GP]

### 3.3.41 UC16.1.1.1-GP - Inserimento del nuovo nome

- **Codice:** UC16.1.1.1-GP.
- **Titolo:** inserimento del nuovo nome.
- **Attori primari:** utente.
- **Descrizione:** l'utente aggiunge il nuovo nome relativo all'identificativo inserito che vuole modificare.
- **Precondizione:** l'utente vuole modificare un utente già presente nel sistema.
- **Postcondizione:** il nome è stato inserito.
- **Scenario principale:**
  1. L'utente inserisce il nuovo nome dell'utente che vuole modificare

### 3.3.42 UC16.1.1.2-GP - Inserimento del nuovo cognome

- **Codice:** UC16.1.1.2-GP.
- **Titolo:** inserimento del nuovo cognome.
- **Attori primari:** utente.
- **Descrizione:** l'utente aggiunge il nuovo cognome relativo all'identificativo inserito che vuole modificare.
- **Precondizione:** l'utente vuole modificare un utente già presente nel sistema.
- **Postcondizione:** il cognome è stato inserito.
- **Scenario principale:**
  1. L'utente inserisce il nuovo cognome dell'utente che vuole modificare.

### 3.3.43 UC16.1.1.3-GP - Inserimento del nuovo contatto Email

- **Codice:** UC16.1.1.3-GP.
- **Titolo:** inserimento del nuovo contatto Email.
- **Attori primari:** utente.
- **Descrizione:** l'utente aggiunge il nuovo contatto Email relativo all'identificativo inserito che vuole modificare.
- **Precondizione:** l'utente vuole modificare un utente già presente nel sistema.
- **Postcondizione:** il contatto Email è stato inserito.
- **Scenario principale:**
  1. L'utente inserisce il nuovo contatto Email dell'utente che vuole modificare

### 3.3.44 UC16.1.1.4-GP - Inserimento del nuovo contatto Telegram

- **Codice:** UC16.1.1.4-GP.
- **Titolo:** inserimento del nuovo contatto Telegram.
- **Attori primari:** utente.
- **Descrizione:** l'utente aggiunge il nuovo contatto Telegram relativo all'identificativo inserito che vuole modificare.
- **Precondizione:** l'utente vuole modificare un utente già presente nel sistema.
- **Postcondizione:** il contatto Telegram è stato inserito.
- **Scenario principale:**
  1. L'utente inserisce il nuovo contatto Telegram dell'utente che vuole modificare

### 3.3.45 UC16.1.2-GP - Errore, nuovi dati dell'utente già esistenti

- **Codice:** UC16.1.2-GP.
- **Titolo:** errore, nuovi dati dell'utente già esistenti.
- **Attori primari:** utente.
- **Descrizione:** i nuovi dati dell'utente da modificare che sono stati inseriti sono già presenti nel sistema, ovvero i nuovi campi corrispondono a quelli di un utente già esistente. In particolare il contatto Telegram o Email, perchè una persona può avere lo stesso nome e cognome di un altro, ma non la stessa mail e nemmeno lo stesso identificativo Telegram.
- **Precondizione:** l'utente vuole modificare un utente esistente nel sistema.
- **Postcondizione:** l'utente non è stato modificato e viene visualizzato un messaggio di errore.
- **Scenario principale:**
  1. L'utente inserisce i campi richiesti dal sistema per la modifica di un utente
  2. L'utente visualizza un messaggio di errore

### 3.3.46 UC16.2-GP - Errore identificativo inesistente

- **Codice:** UC16.2-GP.
- **Titolo:** errore identificativo inesistente.
- **Attori primari:** utente.
- **Descrizione:** l'utente inserisce l'identificativo dell'utente di cui vuole modificare le informazioni, ma viene avvisato che ha inserito un'identificativo errato perchè esso non è presente all'interno del sistema.
- **Precondizione:** l'utente vuole modificare un utente già presente.
- **Postcondizione:** il sistema comunica all'utilizzatore l'errore.
- **Scenario principale:**
  1. L'utente inserisce un identificativo errato
  2. Il sistema comunica all'utilizzatore l'errore

### 3.3.47 UC17-GP - Aggiunta preferenze

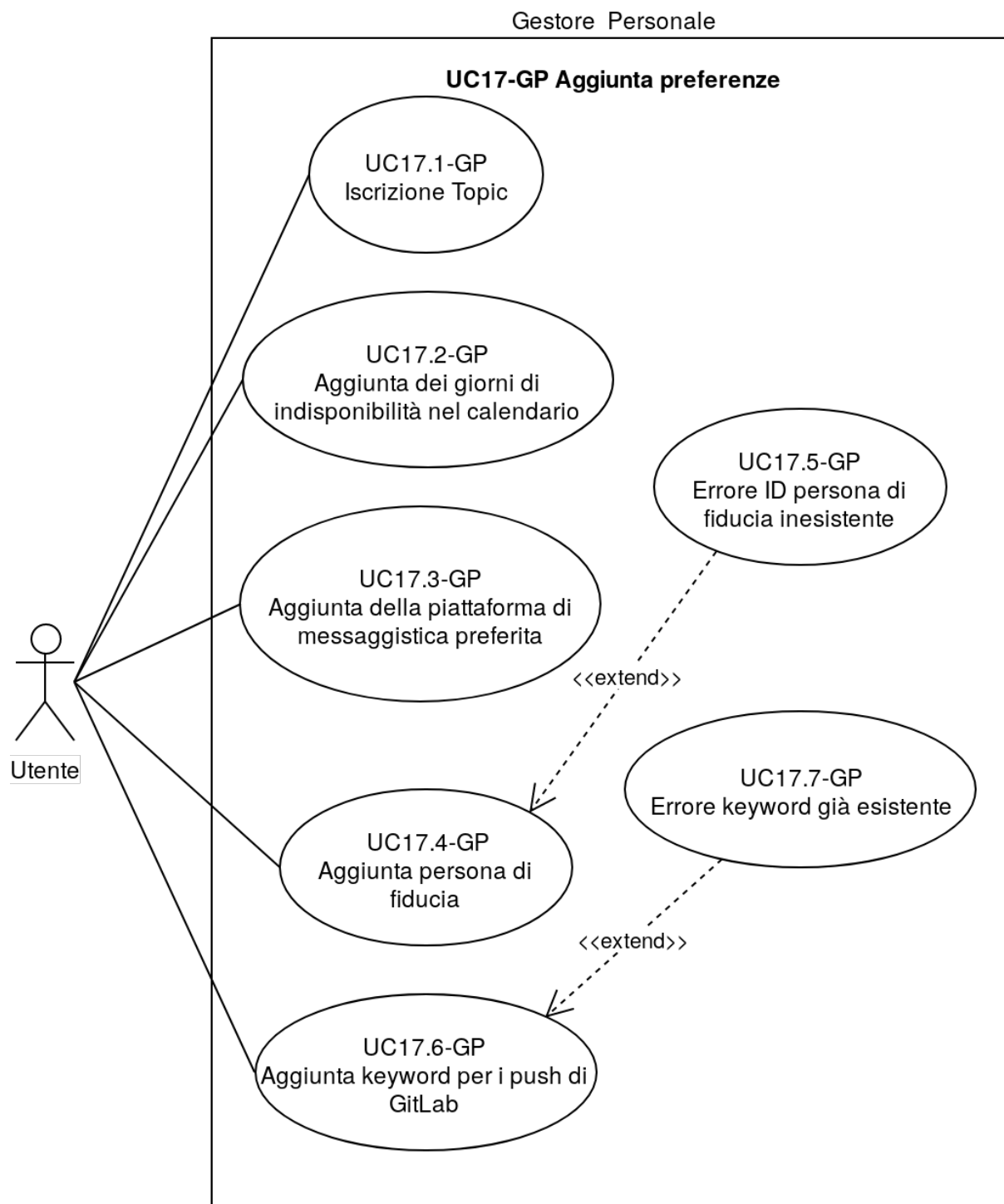


Figura 14: UC17-GP - Aggiunta preferenze

- **Codice:** UC17-GP.
- **Titolo:** aggiunta preferenze.
- **Attori primari:** utente.



- **Descrizione:** l'utente, date le varie opzioni per configurare Butterfly, aggiunge una preferenza tra Topic, giorni di calendario, piattaforma di messaggistica (Telegram o Email) preferita e la persona di fiducia che lo può sostituire.
- **Precondizione:** l'utente ha acceduto con le sue credenziali corrette nel sistema e non ha già selezionato tutte le preferenze possibili proposte da Butterfly.
- **Postcondizione:** la nuova configurazione contiene una o più preferenze in aggiunta rispetto a quella precedente.
- **Scenario principale:**
  1. L'utente aggiunge una o più preferenze

### 3.3.48 UC17.1-GP - Iscrizione Topic

- **Codice:** UC17.1-GP.
- **Titolo:** iscrizione Topic.
- **Attori primari:** utente.
- **Descrizione:** data la lista di Topic presenti, l'utente ne seleziona uno o più a cui è interessato, ricevendone una notifica. I Topic sono divisi per categoria e comprendono etichette, progetto a cui sono legate e l'applicazione di provenienza: Redmine o GitLab.
- **Precondizione:** l'utente ha acceduto correttamente nel sistema e non ha già selezionato tutti i Topic possibili proposti da Butterfly.
- **Postcondizione:** il numero di Topic a cui è interessato l'utente è aumentato.
- **Scenario principale:**
  1. L'utente procede all'iscrizione di uno o più Topic

### 3.3.49 UC17.2-GP - Aggiunta dei giorni di indisponibilità nel calendario

- **Codice:** UC17.2-GP.
- **Titolo:** aggiunta dei giorni di indisponibilità nel calendario.
- **Attori primari:** utente.
- **Descrizione:** dato il calendario lavorativo, l'utente aggiunge uno o più giorni in cui non è reperibile e non vuole ricevere notifiche.
- **Precondizione:** l'utente ha acceduto correttamente nel sistema e vuole selezionare alcuni giorni di indisponibilità.
- **Postcondizione:** il numero di giorni in cui l'utente non si rende disponibile è aumentato.
- **Scenario principale:**
  1. L'utente procede all'inserimento di uno o più giorni di indisponibilità

### 3.3.50 UC17.3-GP - Aggiunta della piattaforma di messaggistica preferita

- **Codice:** UC17.3-GP.
- **Titolo:** aggiunta della piattaforma di messaggistica preferita.
- **Attori primari:** utente.
- **Descrizione:** l'utente aggiunge la sua preferenza tra Telegram e Email dove vuole ricevere le notifiche.
- **Precondizione:** l'utente ha acceduto correttamente nel sistema e non ha già selezionato tutte le piattaforme di messaggistica possibili proposte da Butterfly.
- **Postcondizione:** il numero di piattaforme di messaggistica selezionate dall'utente è aumentato.
- **Scenario principale:**
  1. L'utente procede all'aggiunta di una o più piattaforme di messaggistica

### 3.3.51 UC17.4-GP - Aggiunta persona di fiducia

- **Codice:** UC17.4-GP.
- **Titolo:** aggiunta persona di fiducia.
- **Attori primari:** utente.
- **Descrizione:** l'utente aggiunge l'utente legato a un identificativo di sua preferenza a cui inoltrare i messaggi in caso di indisponibilità.
- **Precondizione:** l'utente ha acceduto con le sue credenziali corrette nel sistema e non ha già selezionato la persona a cui inoltrare le notifiche.
- **Postcondizione:** la preferenza viene aggiunta correttamente.
- **Scenario principale:**
  1. L'utente procede all'aggiunta della sua persona di fiducia
- **Estensioni:**
  1. Errore identificativo persona di fiducia inesistente [UC17.5-GP]

### 3.3.52 UC17.5-GP - Errore identificativo persona di fiducia inesistente

- **Codice:** UC17.5-GP.
- **Titolo:** errore identificativo persona di fiducia inesistente.
- **Attori primari:** utente.
- **Descrizione:** l'utente cerca di aggiungere una persona di fiducia ma viene avvisato che ha inserito un identificativo errato.
- **Precondizione:** l'utente ha acceduto con le sue credenziali corrette nel sistema e non ha già selezionato la persona a cui inoltrare le notifiche.
- **Postcondizione:** il sistema comunica all'utilizzatore l'errore di preferenza.

- **Scenario principale:**

1. L'utente procede all'aggiunta della sua persona di fiducia
2. Il sistema rileva che la persona di fiducia non esiste
3. L'utente visualizza l'errore

### 3.3.53 UC17.6-GP - Aggiunta keyword per i push di GitLab

- **Codice:** UC17.6-GP.

- **Titolo:** aggiunta keyword per i push di GitLab.

- **Attori primari:** utente.

- **Descrizione:** l'utente aggiunge le keyword che vuole che siano contenute nei messaggi di commit dei push di cui vuole ricevere la notifica.

- **Precondizione:** l'utente ha acceduto al sistema.

- **Postcondizione:** nelle nuove configurazioni dell'utente selezionato sono presenti una o più nuove keyword per ricevere le notifiche da push di GitLab di interesse.

- **Scenario principale:**

1. L'utente procede all'aggiunta di una o più nuove keyword

- **Estensioni:**

1. Errore keyword già esistente [UC17.7-GP]

### 3.3.54 UC17.7-GP - Errore keyword già esistente

- **Codice:** UC17.7-GP.

- **Titolo:** errore keyword già esistente.

- **Attori primari:** utente.

- **Descrizione:** la keyword che vuole aggiungere l'utente è già registrata nel sistema.

- **Precondizione:** l'utente ha acceduto al sistema.

- **Postcondizione:** il sistema comunica all'utilizzatore l'errore della keyword.

- **Scenario principale:**

1. L'utente procede all'aggiunta di keyword
2. Il sistema rileva che queste sono già presenti
3. L'utente visualizza un messaggio di errore

### 3.3.55 UC18-GP - Rimozione preferenze

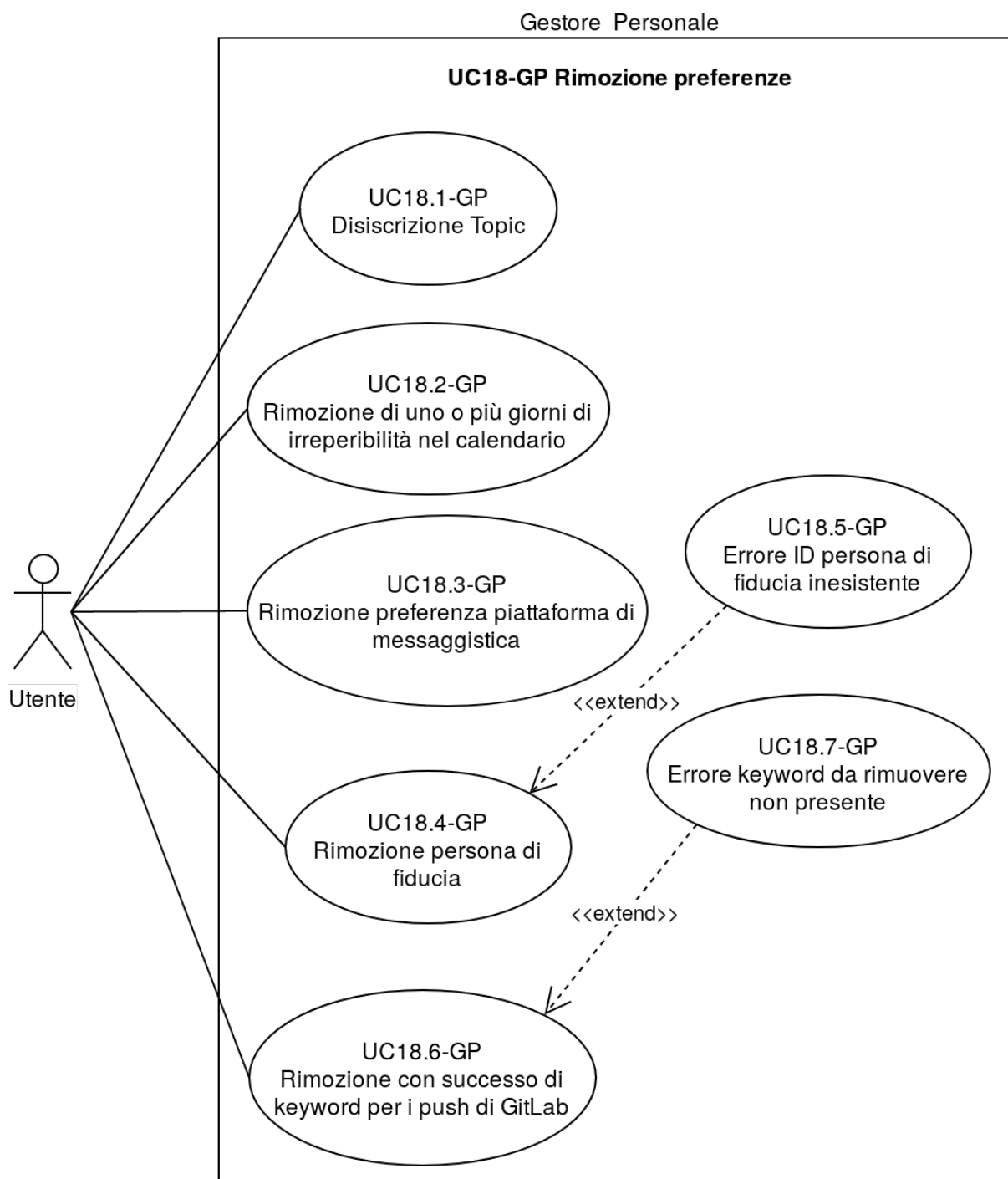


Figura 15: UC18-GP - Rimozione preferenze

- **Codice:** UC18-GP.
- **Titolo:** rimozione preferenze.
- **Attori primari:** utente.
- **Descrizione:** l'utente, dopo aver selezionato delle preferenze dalle opzioni di configurazione, ne rimuove una o più. Le preferenze consistono in Topic, date di calendario, piattaforma di messaggistica (Telegram e email) e persona di fiducia che lo può sostituire.

- **Precondizione:** l'utente ha eseguito l'accesso nel sistema ed è presente almeno una preferenza selezionata tra quelle proposte da Butterfly.
- **Postcondizione:** la nuova configurazione contiene una o più preferenze in meno rispetto a quella precedente.
- **Scenario principale:**
  1. L'utente procede alla rimozione di una o più preferenze

### 3.3.56 UC18.1-GP - Disiscrizione Topic

- **Codice:** UC18.1-GP.
- **Titolo:** disiscrizione Topic.
- **Attori primari:** utente.
- **Descrizione:** l'utente si disiscrive da uno o più Topic dai quali prima riceveva delle notifiche.
- **Precondizione:** l'utente ha acceduto correttamente nel sistema e non ha già selezionato tutti i Topic possibili proposti da Butterfly.
- **Postcondizione:** il numero di Topic a cui è iscritto l'utente è diminuito.
- **Scenario principale:**
  1. L'utente procede alla disiscrizione di uno o più Topic

### 3.3.57 UC18.2-GP - Rimozione di uno o più giorni di irreperibilità nel calendario

- **Codice:** UC18.2-GP.
- **Titolo:** rimozione di uno o più giorni di irreperibilità nel calendario.
- **Attori primari:** utente.
- **Descrizione:** l'utente rimuove i giorni di calendario in cui precedentemente non era reperibile, tornando disponibile.
- **Precondizione:** l'utente ha acceduto correttamente nel sistema ed è presente almeno un giorno di calendario selezionato.
- **Postcondizione:** il numero di giorni di calendario in cui l'utente non è reperibile è diminuito.
- **Scenario principale:**
  1. L'utente procede alla rimozione di uno o più giorni di irreperibilità

### 3.3.58 UC18.3-GP - Rimozione preferenza piattaforma di messaggistica

- **Codice:** UC18.3-GP.
- **Titolo:** rimozione preferenza piattaforma di messaggistica.
- **Attori primari:** utente.

- **Descrizione:** l'utente rimuove una o più preferenze tra Telegram e Email dalle quali non vuole più ricevere notifiche tramite Butterfly.
- **Precondizione:** l'utente ha acceduto correttamente nel sistema ed è presente almeno una piattaforma di messaggistica selezionata tra quelle proposte da Butterfly.
- **Postcondizione:** il numero di piattaforme di messaggistica da cui l'utente vuole ricevere notifiche è diminuito.
- **Scenario principale:**
  1. L'utente procede alla rimozione di una o più piattaforme di messaggistica

### 3.3.59 UC18.4-GP - Rimozione persona di fiducia

- **Codice:** UC18.4-GP.
- **Titolo:** aggiunta persona di fiducia.
- **Attori primari:** utente.
- **Descrizione:** l'utente rimuove l'utente legato a un identificativo di sua preferenza a cui inoltrare i messaggi in caso di indisponibilità.
- **Precondizione:** l'utente ha eseguito l'accesso nel sistema e vuole rimuovere la sua persona di fiducia.
- **Postcondizione:** la preferenza viene rimossa correttamente.
- **Scenario principale:**
  1. L'utente procede alla rimozione della sua persona di fiducia
- **Estensioni:**
  1. Errore identificativo persona di fiducia inesistente [UC18.5-GP].

### 3.3.60 UC18.5-GP - Errore identificativo persona di fiducia inesistente

- **Codice:** UC18.5-GP.
- **Titolo:** errore identificativo persona di fiducia inesistente.
- **Attori primari:** utente.
- **Descrizione:** l'utente viene avvisato che ha inserito un identificativo errato.
- **Precondizione:** l'utente ha acceduto con le sue credenziali corrette nel sistema e vuole rimuovere la sua persona di fiducia.
- **Postcondizione:** il sistema comunica all'utilizzatore l'errore.
- **Scenario principale:**
  1. L'utente inserisce l'identificativo della sua persona di fiducia
  2. Il sistema rileva che questo identificativo non esiste
  3. L'utente visualizza il messaggio di errore

### 3.3.61 UC18.6-GP - Rimozione con successo di keyword per i push di GitLab

- **Codice:** UC18.6-GP.
- **Titolo:** rimozione con successo di keyword per i push di GitLab.
- **Attori primari:** utente.
- **Descrizione:** l'utente seleziona e rimuove una o più keyword già presente nel sistema per non ricevere la notifica di push in cui i messaggi di commit contengono la keyword rimossa.
- **Precondizione:** l'utente ha acceduto al sistema.
- **Postcondizione:** nelle nuove configurazioni dell'utente selezionato sono state rimosse delle keyword precedentemente presenti.
- **Scenario principale:**
  1. L'utente rimuove con successo una o più keyword per cui non vuole più ricevere messaggi di push
- **Estensioni:**
  1. Errore keyword da rimuovere non presente [UC18.7-GP]

### 3.3.62 UC18.7-GP - Errore keyword da rimuovere non presente

- **Codice:** UC18.7-GP.
- **Titolo:** errore keyword da rimuovere non presente.
- **Attori primari:** utente.
- **Descrizione:** la keyword che l'utente intende rimuovere non è registrata nel sistema.
- **Precondizione:** l'utente ha acceduto al sistema.
- **Postcondizione:** viene visualizzato un messaggio d'errore con indicato che la keyword selezionata che non è presente nel sistema.
- **Scenario principale:**
  1. L'utente inserisce la keyword che vuole rimuovere
  2. Il sistema rileva che non è presente
  3. L'utente visualizza il messaggio di errore

## 4 Requisiti

Ad ogni requisito viene assegnato il codice identificativo univoco:

$$R[\text{Numero}] [\text{Tipo}] [\text{Priorità}]$$

in cui ogni parte ha un significato preciso:

- **R**: requisito.
- **Numero**: numero progressivo che segue la struttura dei documenti.
- **Tipo**: la la tipologia di requisito che può essere di:
  - **F**: funzionalità.
  - **Q**: QUALITÀ<sub>G</sub>.
  - **V**: vincolo.
- **Priorità**: indica il grado di urgenza di un requisito di essere soddisfatto, come:
  - **0**: opzionale.
  - **1**: desiderabile.
  - **2**: obbligatorio.

Esempio: R2Q1 indica il secondo requisito di qualità ed è desiderabile.



## 4.1 Requisiti di funzionalità

Codice	Requisito	Fonte
R1F2	Redmine deve poter inviare la segnalazione di apertura issue al Producer Redmine	Interno UC1-PR
R2F2	Redmine deve poter inviare la segnalazione di modifica issue al Producer Redmine	Interno UC2-PR
R3F2	GitLab deve essere in grado di segnalare l'apertura di issue al Producer GitLab	Interno UC3-PG
R4F2	GitLab deve essere in grado di segnalare la modifica issue al Producer GitLab	Interno UC4-PG
R5F2	GitLab deve poter segnalare un evento di push al Producer GitLab	Interno UC5-PG
R6F2	Il Producer Redmine deve essere in grado di inviare un messaggio al Gestore Personale	Interno UC6-GP
R6.1F2	Il Producer Redmine deve essere in grado di inviare un messaggio di apertura issue al Gestore Personale	Interno UC6.1-GP
R6.2F2	Il Producer Redmine deve essere in grado di inviare un messaggio di modifica issue al Gestore Personale	Interno UC6.2-GP
R7F1	Il Producer Redmine deve essere in grado di scartare i messaggi non validi	Interno UC6.3-GP
R8F2	Il Producer GitLab deve essere in grado di inviare un messaggio al Gestore Personale	Interno UC7-GP
R8.1F2	Il Producer GitLab deve essere in grado di inviare messaggi di commit al Gestore Personale	Interno UC7.1-GP
R8.2F2	Il Producer GitLab deve essere in grado di inviare un messaggio di issue al Gestore Personale	Interno UC7.2-GP
R8.2.1F2	Il Producer GitLab deve essere in grado di inviare un messaggio di nuova issue al Gestore Personale	Interno UC7.2.1-GP
R8.2.2F2	Il Producer GitLab deve essere in grado di inviare un messaggio di modifica issue al Gestore Personale	Interno UC7.2.2-GP
R9F1	Il Producer GitLab deve essere in grado di scartare i messaggi di issue non validi	Interno UC7.2.3-GP
R10F2	Il Gestore Personale deve poter inviare il messaggio finale al Consumer Telegram	Interno UC8-CT

Tabella 1: Elenco dei requisiti di funzionalità (1)

Codice	Requisito	Fonte
R11F2	Il Gestore Personale deve poter inviare il messaggio finale al Consumer Email	Interno UC9-CE
R12F2	Il Consumer Telegram deve poter inoltrare il messaggio finale al bot Telegram	Interno UC10-BT
R13F2	Il Consumer Email deve poter inoltrare il messaggio finale al server Email	Interno UC11-SE
R14F0	L'utente può eseguire l'accesso al Gestore Personale	Interno UC12.1-GP
R14.1F0	L'utente può inserire il proprio identificativo all'interno del sistema	Interno UC12.1.1
R15F0	Butterfly fa apparire un messaggio di errore se il tentativo di accesso non è andato a buon fine	Interno UC12.1.2
R16F0	L'utente acceduto deve poter uscire dal sistema	Interno UC13-GP
R17F0	L'utente acceduto deve poter aggiungere un nuovo utente	Interno UC14-GP
R17.1F0	L'utente acceduto deve poter inserire il nome dell'utente da aggiungere	Interno UC14.1.1-GP
R17.2F0	L'utente acceduto deve poter inserire il cognome dell'utente da aggiungere	Interno UC14.1.2-GP
R17.3F0	L'utente acceduto deve poter inserire il contatto Email dell'utente da aggiungere	Interno UC14.1.3-GP
R17.3.1F0	L'utente acceduto deve poter visualizzare un messaggio di errore se il contatto Email non è univoco	Interno UC14.2-GP
R17.4F0	L'utente acceduto deve poter inserire il contatto Telegram dell'utente da aggiungere	Interno UC14.1.4-GP
R17.4.1F0	L'utente acceduto deve poter visualizzare un messaggio di errore se il contatto Telegram non è univoco	Interno UC14.2-GP
R18F0	L'utente acceduto deve poter rimuovere un utente presente nel sistema	Interno UC15-GP
R18.1F0	L'utente acceduto deve poter inserire il contatto Email dell'utente da rimuovere	Interno UC15.1.1-GP
R18.2F0	L'utente acceduto deve poter inserire il contatto Telegram dell'utente da rimuovere	Interno UC15.1.2-GP
R18.3F0	L'utente acceduto deve poter visualizzare un messaggio di errore se l'identificativo non è presente nel sistema	Interno UC15.2-GP

Tabella 2: Elenco dei requisiti di funzionalità (2)

Codice	Requisito	Fonte
R18.4F0	L'utente acceduto deve potersi rimuovere dal sistema	Interno UC15.3-GP
R19F0	L'utente acceduto deve poter modificare le informazioni relative a un utente	Interno UC16-GP
R19.1F0	L'utente acceduto deve poter selezionare l'identificativo dell'utente da modificare	Interno UC16.1-GP
R19.1.1F0	L'utente acceduto deve poter visualizzare un messaggio di errore se l'identificativo non è riconosciuto dal sistema	Interno UC16.2-GP
R19.2F0	L'utente acceduto deve poter scegliere il nuovo nome dell'utente da modificare	Interno UC16.1.1.1-GP
R19.3F0	L'utente acceduto deve poter scegliere il nuovo cognome dell'utente da modificare	Interno UC16.1.1.2-GP
R19.4F0	L'utente acceduto deve poter scegliere il nuovo contatto Email dell'utente da modificare	Interno UC16.1.1.3-GP
R19.4.1F0	L'utente acceduto deve poter visualizzare un messaggio di errore se il contatto Email è già presente	Interno UC16.1.2-GP
R19.5F0	L'utente acceduto deve poter scegliere il nuovo contatto Telegram dell'utente da modificare	Interno UC16.1.1.4-GP
R19.5.1F0	L'utente acceduto deve poter visualizzare un messaggio di errore se il contatto Telegram è già presente	Interno UC16.1.2-GP
R20F0	L'utente acceduto deve poter aggiungere le proprie preferenze nel sistema	Interno UC17-GP
R20.1F0	L'utente acceduto deve poter aggiungere nuovi Topic di iscrizione	Interno UC17.1-GP
R20.2F0	L'utente acceduto deve poter aggiungere nuovi giorni di indisponibilità nel calendario	Interno UC17.2-GP
R20.3F0	L'utente acceduto deve poter aggiungere una nuova piattaforma di messaggistica preferita	Interno UC17.3-GP
R20.4F0	L'utente acceduto deve poter aggiungere la propria persona di fiducia	Interno UC17.4-GP
R20.4.1F0	L'utente acceduto deve poter visualizzare un messaggio di errore se il contatto Email della persona di fiducia non è presente nel sistema	Interno UC17.5-GP

Tabella 3: Elenco dei requisiti di funzionalità (3)

Codice	Requisito	Fonte
R20.4.2F0	L'utente acceduto deve poter visualizzare un messaggio di errore se il contatto Telegram della persona di fiducia non è presente nel sistema	Interno UC17.5-GP
R20.5F0	L'utente acceduto deve poter aggiungere nuove keyword di interesse per i messaggi di commit di GitLab	Interno UC17.6-GP
R20.5.1F0	L'utente acceduto deve poter visualizzare un messaggio di errore se la keyword inserita era già nella sua lista	Interno UC17.7-GP
R21F0	L'utente acceduto deve poter rimuovere le proprie preferenze dal sistema	Interno UC18-GP
R21.1F0	L'utente acceduto deve poter rimuovere i Topic a cui è iscritto	Interno UC18.1-GP
R21.2F0	L'utente acceduto deve poter rimuovere giorni di indisponibilità nel calendario	Interno UC18.2-GP
R21.3F0	L'utente acceduto deve poter rimuovere una piattaforma di messaggistica preferita	Interno UC18.3-GP
R21.4F0	L'utente acceduto deve poter rimuovere la propria persona di fiducia	Interno UC18.4-GP
R21.4.1F0	L'utente acceduto deve poter visualizzare un messaggio di errore se il contatto Email della persona di fiducia non è presente nel sistema	Interno UC18.5-GP
R21.4.2F0	L'utente acceduto deve poter visualizzare un messaggio di errore se il contatto Telegram della persona di fiducia non è presente nel sistema	Interno UC18.5-GP
R21.5F0	L'utente acceduto deve poter rimuovere keyword di interesse per i messaggi di commit di GitLab	Interno UC18.6-GP
R21.5.1F0	L'utente acceduto deve poter visualizzare un messaggio di errore se la keyword da rimuovere è assente dalla sua lista	Interno UC18.7-GP

Tabella 4: Elenco dei requisiti di funzionalità (4)

Codice	Requisito	Fonte
R22F2	Le componenti Consumer devono essere in grado di inviare i messaggi provenienti da un Topic verso il corretto destinatario	Capitolato
R23F2	Le componenti Consumer devono essere in grado di abbonarsi ai Topic scelti	Capitolato
R24F2	Le segnalazioni devono poter essere gestite in maniera automatica e personalizzabile	Capitolato
R25F2	Nel sistema deve essere presente un Broker che istanzia e gestisce le segnalazioni organizzandole per Topic	Capitolato
R26F2	Le componenti Producer devono riuscire a pubblicare le segnalazioni recuperate sotto forma di messaggi secondo i Topic corretti	Capitolato
R27F2	Le componenti devono esporre delle API <code>REST<sub>G</sub></code> per le interazioni con le altre componenti	Capitolato

Tabella 5: Elenco dei requisiti di funzionalità (5)

## 4.2 Requisiti di qualità

Codice	Requisito	Fonte
R1Q1	È stabilito un numero massimo di giorni di ritardo per la chiusura di una issue	Interno QPR001
R2Q1	L'INDICE DI GULPEASE <sub>G</sub> di ogni documento deve rientrare all'interno di un intervallo stabilito	Interno QPD001
R3Q1	Una frequenza minima di commit devono essere effettuati in una settimana	Interno QPR004
R4Q1	Un numero stabilito di requisiti desiderabili deve essere soddisfatto	Interno QPR006
R5Q1	Nessun rischio non verificato precedentemente deve accadere nel corso del progetto	Interno QPR007
R6Q1	Ogni documento deve attraversare tutte le fasi previste dal suo CICLO DI VITA <sub>G</sub>	Interno QPR008
R7Q1	Viene stabilito il numero massimo di modifiche che può ricevere un prodotto prima di essere verificato	Interno QPR009
R8Q2	Tutte le NORME <sub>G</sub> inserite nelle Norme di Progetto devono essere rispettate	Interno
R9Q2	Tutti i vincoli presenti nel Piano di Qualifica devono essere rispettati	Interno
R10Q2	Le applicazioni sviluppate devono rispettare i fattori trattati in The Twelve-Factor App segnati nel <i>PianoDiQualifica v2.0.0<sub>D</sub></i>	Capitolato <i>VE_2018-12-12<sub>D</sub></i>
R11Q2	È necessario presentare il BUG <sub>G</sub> reporting per ogni componente	Capitolato

Tabella 6: Elenco dei requisiti di qualità (1)

Codice	Requisito	Fonte
R12Q2	Deve essere redatta la documentazione sulle scelte progettuali effettuate	Capitolato
R12.1Q2	Ogni scelta descritta nella documentazione deve essere correlata dalle relative motivazioni	Capitolato
R13Q2	È necessario testare ogni prodotto software considerando ogni sistema di riferimento e interazione tra le sue parti, perciò con test d'unità, d'integrazione e di sistema	Capitolato
R13.1Q2	È necessario fornire test d'unità per ogni componente applicativo	Capitolato
R13.2Q2	È necessario fornire test d'integrazione per ogni componente applicativo	Capitolato
R13.3Q2	È necessario testare interamente il sistema con test di sistema	Capitolato
R14Q1	Per ogni problema aperto documentato, si allegano delle soluzioni da attuare in futuro	Capitolato
R14.1Q2	Deve essere redatta una documentazione su eventuali problemi riscontrati rimasti ancora aperti al termine del progetto	Capitolato
R15Q2	È necessario presentare un file README <sub>G</sub> per ogni componente	Capitolato
R15.1Q2	I file README delle componenti applicative devono contenere la documentazione delle API <sub>G</sub> esposte dal servizio	Capitolato
R15.2Q2	I file README delle componenti applicative devono contenere le istruzioni per il loro utilizzo	Capitolato
R15.3Q2	È necessario presentare un file README per il Dockerfile	Capitolato
R15.3.1Q2	Il file README per il Dockerfile deve contenere le istruzioni per l'avvio	Capitolato
R15.3.2Q2	Il file README per il Dockerfile deve contenere la documentazione delle configurazioni custom scelte	Capitolato

Tabella 7: Elenco dei requisiti di qualità (2)

### 4.3 Requisiti di vincolo

Codice	Requisito	Fonte
R1V2	Devono essere sviluppati due componenti applicativi Producer tra Redmine 4.0.1, GitLab 11.7 e SonarQube 6.7	Capitolato
R1.1V0	È possibile avere un terzo componente applicativo Producer oltre ai due obbligatori	Capitolato
R2V2	Devono essere sviluppati due componenti applicativi Consumer tra Telegram (Bot API 4.0), Email e Slack	Capitolato
R2.1V0	È possibile avere un terzo componente applicativo Consumer oltre ai due obbligatori	Capitolato
R3V2	Docker 18.09 deve essere la tecnologia di riferimento per l'istanziamento di tutte le componenti	Capitolato
R3.1V2	È necessario presentare un Dockerfile per ogni componente	Capitolato
R4V1	Per lo sviluppo dei componenti applicativi è possibile usare come linguaggio JAVA <sub>G</sub> 8 o una versione più recente, PYTHON 3.6.7 <sub>G</sub> o NODE.JS <sub>G</sub> 10.15.1	Capitolato
R5V1	È possibile utilizzare Apache Kafka 2.12 come Broker	Capitolato
R6V0	L'interfaccia è sviluppata utilizzando HTML5 e CSS3	Interno
R7V0	Il database viene sviluppato utilizzando MongoDB 4.0.6	Interno
R8V0	Il server web viene sviluppato utilizzando Python 3.6.7 con la libreria CHERRYPY <sub>G</sub>	Interno
R9V0	Gli URL dell'interfaccia web devono rispettare lo standard REST	Interno
R10V0	Ciascun componente Docker viene istanziato tramite un file Docker Compose 3.7	Interno

Tabella 8: Elenco dei requisiti di vincolo



## 4.4 Tracciamento

### 4.4.1 Tracciamento fonte-requisito

Fonte	Requisito
Capitolato	R22F2
	R23F2
	R24F2
	R25F2
	R26F2
	R27F2
	R10Q2
	R11Q2
	R12Q2
	R12.1Q2
	R13Q2
	R13.1Q2
	R13.2Q2
	R13.3Q2
	R14Q1
	R14.1Q2
	R15Q2
	R15.1Q2
	R15.2Q2
	R15.3Q2
	R15.3.1Q2
	R15.3.2Q2

Tabella 9: Elenco dei requisiti del capitolato (1)

Fonte	Requisito
Capitolato	R1V2
	R1.1V0
	R2V2
	R2.1V0
	R3V2
	R3.1V2
	R4V1
	R5V1

Tabella 10: Elenco dei requisiti del capitolato (2)

Fonte	Requisito
UC1-PR	R1F2
UC2-PR	R2F2
UC3-PG	R3F2
UC4-PG	R4F2
UC5-PG	R5F2
UC6-GP	R6F2
UC6.1-GP	R6.1F2
UC6.2-GP	R6.2F2
UC6.3-GP	R7F1
UC7-GP	R8F2
UC7.1-GP	R8.1F2
UC7.2-GP	R8.2F2
UC7.2.1-GP	R8.2.1F2
UC7.2.2-GP	R8.2.2F2
UC7.2.3-GP	R9F1
UC8-CT	R10F2
UC9-CE	R11F2

Tabella 11: Elenco dei requisiti per i casi d'uso (1)

Fonte	Requisito
UC10-BT	R12F2
UC11-SE	R13F2
UC12.1-GP	R14F0
UC12.1.1	R14.1F0
UC12.1.2	R15F0
UC13-GP	R16F0
UC14-GP	R17F0
UC14.1.1-GP	R17.1F0
UC14.1.2-GP	R17.2F0
UC14.1.3-GP	R17.3F0
UC14.2-GP	R17.3.1F0 R17.4.1F0
UC14.1.4-GP	R17.4F0
UC15-GP	R18F0
UC15.1.1-GP	R18.1F0
UC15.1.2-GP	R18.2F0
UC15.2-GP	R18.3F0
UC15.3-GP	R18.4F0
UC16-GP	R19F0
UC16.1-GP	R19.1F0
UC16.2-GP	R19.1.1F0
UC16.1.1.1-GP	R19.2F0
UC16.1.1.2-GP	R19.3F0
UC16.1.1.3-GP	R19.4F0
UC16.1.2-GP	R19.4.1F0 R19.5.1F0
UC16.1.1.4-GP	R19.5F0
UC17-GP	R20F0
UC17.1-GP	R20.1F0
UC17.2-GP	R20.2F0
UC17.3-GP	R20.3F0

Tabella 12: Elenco dei requisiti per i casi d'uso (2)

Fonte	Requisito
UC17.4-GP	R20.4F0
UC17.5-GP	R20.4.1F0
	R20.4.2F0
UC17.6-GP	R20.5F0
UC17.7-GP	R20.5.1F0
UC18-GP	R21F0
UC18.1-GP	R21.1F0
UC18.2-GP	R21.2F0
UC18.3-GP	R21.3F0
UC18.4-GP	R21.4F0
UC18.5-GP	R21.4.1F0
	R21.4.2F0
UC18.6-GP	R21.5F0
UC18.7-GP	R21.5.1F0

Tabella 13: Elenco dei requisiti per i casi d'uso (3)

Fonte	Requisito
QPR001	R1Q1
QPD001	R2Q1
QPR004	R3Q1
QPR006	R4Q1
QPR007	R5Q1
QPR008	R6Q2
QPR009	R7Q1
<i>VE_2018-12-12<sub>D</sub></i>	R10Q2

Tabella 14: Elenco dei requisiti per gli obiettivi di qualità e verbali

Fonte	Requisito
Interno	R1F2
	R2F2
	R3F2
	R4F2
	R5F2
	R6F2
	R6.1F2
	R6.2F2
	R7F1
	R8.1F2
	R8.2F2
	R8.2.1F2
	R8.2.2F2
	R9F1
	R10F2
	R11F2
	R12F2
	R13F2
	R14F0
	R14.1F0
	R15F0
	R16F0
	R17F0
	R17.1F0
	R17.2F0
	R17.3F0
	R17.3.1F0
	R17.4F0

Tabella 15: Elenco dei requisiti da fonte interna (1)

Fonte	Requisito
Interno	R17.4.1F0
	R18F0
	R18.1F0
	R18.2F0
	R18.3F0
	R18.4F0
	R19F0
	R19.1F0
	R19.1.1F0
	R19.2F0
	R19.3F0
	R19.4F0
	R19.4.1F0
	R19.5F0
	R19.5.1F0
	R20F0
	R20.1F0
	R20.2F0
	R20.3F0
	R20.4F0
	R20.4.1F0
	R20.4.2F0
	R20.5F0
	R20.5.1F0
	R21F0
	R21.1F0
	R21.2F0
	R21.3F0
	R21.4F0

Tabella 16: Elenco dei requisiti da fonte interna (2)

Fonte	Requisito
Interno	R21.4.1F0
	R21.4.2F0
	R21.5F0
	R21.5.1F0
	R1Q1
	R2Q1
	R3Q1
	R4Q1
	R5Q1
	R6Q1
	R7Q1
	R7Q1
	R9Q2
	R6V0
	R7V0
	R8V0
	R9V0
	R10V0

Tabella 17: Elenco dei requisiti da fonte interna (3)

#### 4.4.2 Tracciamento requisito-fonte

Requisito	Fonte
R1F2	Interno UC1-PR
R2F2	Interno UC2-PR
R3F2	Interno UC3-PG
R4F2	Interno UC4-PG
R5F2	Interno UC5-PG
R6F2	Interno UC6-GP
R6F2	Interno UC6.1-GP
R6.2F2	Interno UC6.2-GP
R7F1	Interno UC6.3-GP
R8F2	Interno UC7-GP
R8.1F2	Interno UC7.1-GP
R8.2F2	Interno UC7.2-GP
R8.2.1F2	Interno UC7.2.1-GP

Tabella 18: Elenco dei requisiti funzionali in rapporto alle fonti (1)



Requisito	Fonte
R8.2.2F2	Interno UC7.2.2-GP
R9F1	Interno UC7.2.3-GP
R10F2	Interno UC8-CT
R11F2	Interno UC9-CE
R12F2	Interno UC10-BT
R13F2	Interno UC11-SE
R14F0	Interno UC12.1-GP
R14.1F0	Interno UC12.1.1
R15F0	Interno UC12.1.2
R16F0	Interno UC13-GP
R17F0	Interno UC14-GP
R17.1F0	Interno UC14.1.1-GP
R17.2F0	Interno UC14.1.2-GP

Tabella 19: Elenco dei requisiti funzionali in rapporto alle fonti (2)

Requisito	Fonte
R17.3F0	Interno UC14.1.3-GP
R17.3.1F0	Interno UC14.2-GP
R17.4F0	Interno UC14.1.4-GP
R17.4.1F0	Interno UC14.2-GP
R18F0	Interno UC15-GP
R18.1F0	Interno UC15.1.1-GP
R18.2F0	Interno UC15.1.2-GP
R18.3F0	Interno UC15.2-GP
R18.4F0	Interno UC15.3-GP
R19F0	Interno UC16-GP
R19.1F0	Interno UC16.1-GP
R19.1.1F0	Interno UC16.2-GP
R19.2F0	Interno UC16.1.1.1-GP

Tabella 20: Elenco dei requisiti funzionali in rapporto alle fonti (3)

Requisito	Fonte
R19.3F0	Interno UC16.1.1.2-GP
R19.4F0	Interno UC16.1.1.3-GP
R19.4.1F0	Interno UC16.1.2-GP
R19.5F0	Interno UC16.1.1.4-GP
R19.5.1F0	Interno UC16.1.2-GP
R20F0	Interno UC17-GP
R20.1F0	Interno UC17.1-GP
R20.2F0	Interno UC17.2-GP
R20.3F0	Interno UC17.3-GP
R20.4F0	Interno UC17.4-GP
R20.4.1F0	Interno UC17.5-GP
R20.4.2F0	Interno UC17.5-GP
R20.5F0	Interno UC17.6-GP

Tabella 21: Elenco dei requisiti funzionali in rapporto alle fonti (4)

Requisito	Fonte
R20.5.1F0	Interno UC17.7-GP
R21F0	Interno UC18-GP
R21.1F0	Interno UC18.1-GP
R21.2F0	Interno UC18.2-GP
R21.3F0	Interno UC18.3-GP
R21.4F0	Interno UC18.4-GP
R21.4.1F0	Interno UC18.5-GP
R21.4.2F0	Interno UC18.5-GP
R21.5F0	Interno UC18.6-GP
R21.5.1F0	Interno UC18.7-GP
R22F2	Capitolato
R23F2	Capitolato
R24F2	Capitolato
R25F2	Capitolato
R26F2	Capitolato
R27F2	Capitolato

Tabella 22: Elenco dei requisiti funzionali in rapporto alle fonti (5)

Requisito	Fonte
R1Q1	Interno QPR001
R2Q1	Interno QPD001
R3Q1	Interno QPR004
R4Q1	Interno QPR006
R5Q1	Interno QPR007
R6Q1	Interno QPR008
R7Q1	Interno QPR009
R8Q2	Interno
R9Q2	Interno
R10Q2	Capitolato <i>VE_2018-12-12<sub>D</sub></i>
R11Q2	Capitolato
R12Q2	Capitolato
R12.1Q2	Capitolato
R13Q2	Capitolato
R13.1Q2	Capitolato
R13.2Q2	Capitolato
R13.3Q2	Capitolato
R14Q1	Capitolato
R14.1Q2	Capitolato

Tabella 23: Elenco dei requisiti di qualità in rapporto alle fonti (1)

Requisito	Fonte
R15Q2	Capitolato
R15.1Q2	Capitolato
R15.2Q2	Capitolato
R15.3Q2	Capitolato
R15.3.1F2	Capitolato
R15.3.2F2	Capitolato

Tabella 24: Elenco dei requisiti di qualità in rapporto alle fonti (2)

Requisito	Fonte
R1V2	Capitolato
R1.1V0	Capitolato
R2V2	Capitolato
R2.1V0	Capitolato
R3V2	Capitolato
R3.1V2	Capitolato
R4V1	Capitolato
R5V1	Capitolato
R6V0	Interno
R7V0	Interno
R7V0	Interno
R9V0	Interno
R10V0	Interno

Tabella 25: Elenco dei requisiti di vincolo in rapporto alle fonti

## 4.5 Riepilogo

Tipologia	Obbligatori	Desiderabili	Opzionali
Di funzionalità	23	2	43
Di qualità	17	8	0
Di vincolo	4	2	7

Tabella 26: Riepilogo dei requisiti

## A Approfondimenti tecnologici

### A.1 Webhook

I webhook sono un pattern HTTP leggero che fornisce un modello publisher/subscriber semplice per collegare tra loro Web API. Quando si verifica un evento in un servizio, viene inviata una notifica ai sottoscritti registrati sotto forma di una richiesta HTTP. La richiesta contiene informazioni sull'evento che rende possibile per il ricevitore agire di conseguenza. Le tecnologie Redmine e GitLab li integrano e utilizzando una specifica porta, dove i nostri Producer sono in ascolto, viene inviato un file `JSONG` contenente tutte le informazioni riguardanti l'evento. Questa tecnologia ha forma tale da fornirci collezioni di dati organizzati, facilitandoci così la lettura e la scrittura.

Seguono `DIZIONARI PYTHONG` chiamati genericamente `obj` contenenti i campi dei webhook di nostro interesse:

- GitLab Issue Event:

```
1  obj ["object_kind"]
2  obj ["project"] ["id"]/obj ["object_attributes"] ["project_id"]
3  obj ["project"] ["name"]
4  obj ["assignees"] [k] ["nome utente"]
5  obj ["object_attributes"] ["action"]
6  obj ["object_attributes"] ["description"]
7  obj ["labels"] [k] ["title"]
8  obj ["labels"] [k] ["project_id"]
9  obj ["changes"] ["title"]
10 obj ["changes"] ["labels"] ["previous"] [k] ["title"]
11 obj ["changes"] ["labels"] ["current"] [k] ["title"]
```

- GitLab Push Event:

```
1  obj ["object_kind"]
2  obj ["utente_id"]
3  obj ["utente_nome utente"]
4  obj ["project_id"]
5  obj ["commits"] [k] ["id"]
6  obj ["commits"] [k] ["message"]
7  obj ["commits"] [k] ["timestamp"]
8  obj ["commits"] [k] ["author"] ["name"]
9  obj ["total_commits_count"]
```

- Redmine Issue Event:

```
1  obj ["payload"] ["issue"] ["priority"] ["name"]
2  obj ["payload"] ["issue"] ["tracker"] ["name"]
3  obj ["payload"] ["issue"] ["subject"]
4  obj ["payload"] ["issue"] ["status"] ["name"]
5  obj ["payload"] ["issue"] ["description"]
6  obj ["payload"] ["issue"] ["project"] ["id"]
7  obj ["payload"] ["issue"] ["project"] ["name"]
8  obj ["payload"] ["issue"] ["action"]
```

### A.2 Redmine

Ciascuna istanza di Redmine permette l'utilizzo di webhook<sup>1</sup> che inviano segnalazioni al Producer alla modifica del progetto. Queste vengono ricevute dal server tramite una componente che resta

<sup>1</sup>Riferirsi alla voce "Webhook di Redmine" alla sezione §1.4.2

in ascolto, aggiornando in base a ciò che riceve dai dati presenti sul Gestore Personale e inoltrando i messaggi ai Consumer interessati.

### A.3 GitLab

Ciascuna istanza di GitLab, online o in un server locale, mette a disposizione la configurazione di webhook<sup>2</sup> che, alla modifica di una repository, manda un messaggio con le informazioni relative a quell'evento specifico. Questo messaggio arriva a una componente applicativa capace di aggiornare i dati presenti nel Gestore Personale e, come per Redmine, inoltrare le notifiche ai Consumer interessati.

### A.4 Apache Kafka

Apache Kafka è un software OPEN SOURCE<sub>G</sub> che permette la lettura e la scrittura di messaggi su differenti canali di comunicazioni per i dati. Questi messaggi arrivano dai Producer che ricevono le notifiche di applicazioni di terze parti immettendole nel Broker. Questo le elabora analizzandone il contenuto e contrassegnandole con Topic che verranno utilizzati per l'inoltro ai Consumer e successivamente agli utenti finali, i quali possono abbonarsi a più Topic.

### A.5 Telegram

Telegram permette l'interazione in maniera automatica con gli utenti tramite bot<sup>3</sup> che possono essere configurati per mandare messaggi ricevuti da strumenti di terze parti, in questo caso Butterfly. Il Consumer interroga il Broker per acquisire i messaggi da inoltrare e li trasmette al bot di Telegram, che si occuperà dell'invio all'utente finale.

#### A.5.1 BotFather

Per mandare messaggi tramite Telegram all'utente finale, è necessario utilizzare un bot creato con BotFather. Quest'ultimo è esso stesso un bot offerto da Telegram che permette la creazione e la configurazione di bot che possono fare da tramite tra utenti e servizi esterni dai quali ricevere, o a cui mandare, informazioni o comandi.

### A.6 Email

Per inoltrare le Email agli utenti finali utilizziamo un Consumer associato che sfrutta il SERVER SMTP<sub>G</sub> di posta in modo tale da poter inoltrare i messaggi ottenuti dal Broker all'indirizzo specificato.

### A.7 Docker

L'azienda ci consiglia di utilizzare DOCKER<sub>G</sub> per la semplicità di utilizzo e per l'adattamento all'architettura a microservizi. La configurazione avverrà tramite un DOCKERFILE<sub>G</sub> in cui verranno specificate informazioni come sistema operativo, script di avvio, numero di istanze ed altri parametri specifici.

#### A.7.1 Container software

Un CONTAINER SOFTWARE<sub>G</sub> simula un ambiente virtuale dove è possibile testare e mantenere le proprie applicazioni, permettendo di aumentare l'efficienza riducendone i costi e simulando l'esecuzione di sistema operativo su una macchina con RISORSE<sub>G</sub> condivise.

---

<sup>2</sup>Riferirsi alla voce "Webhook di GitLab" alla sezione §1.4.2

<sup>3</sup>Riferirsi alla voce "Bot di Telegram" alla sezione §1.4.2



#### A.7.1.1 Differenza tra container e macchina virtuale

A differenza delle `MACCHINE VIRTUALIG`, dove lo stato dell'ambiente viene salvato su disco, occupando memoria, i container si adattano in maniera più performante all'applicativo richiesto, in quanto il loro scopo è quello di massimizzare la quantità delle applicazioni in esecuzione riducendo al minimo il numero delle macchine per eseguirla. Sono quindi più leggeri, occupando meno memoria su disco e impiegando meno risorse.

#### A.7.2 DockStation

Per la gestione dei container in locale è stato possibile utilizzare `DOCKSTATIONG` in quanto fornisce una GUI dalla quale si possono effettuare i comandi impartiti a Docker via Terminale. È inoltre possibile avere la visione dello stato di ciascun container e vedere in tempo reale i log che questo genera, oltre che a poter controllare le risorse utilizzate da ciascun container.