



UNIVERSITÀ DEGLI STUDI DI MILANO - BICOCCA

Scuola di Scienze

Dipartimento di Informatica, Sistemistica e Comunicazione

Corso di laurea in Informatica

Sviluppo di una Web Application per l'ottimizzazione del processo di onboarding flussi

Relatore: Prof. Guido Giuseppe Fiorino

Relazione della prova finale di:

Oleg Stoianov

Matricola 829519

Anno Accademico 2019-2020

Ringraziamenti

Un ringraziamento speciale alla mia famiglia, in particolare a mia madre e mio padre: è grazie a loro sostegno e al loro incoraggiamento se oggi sono riuscito a raggiungere questo traguardo.

Ringrazio la mia fidanzata Veronica per avermi trasmesso la sua forza e il suo coraggio. Grazie per tutto il tempo che mi hai dedicato. Grazie per la pazienza che hai avuto con me. Grazie perché ci sei sempre stata.

Ringrazio i miei colleghi, nonché amici, con cui ho condiviso l'intero percorso universitario. Grazie ai quali ho superato i momenti più difficili, e ho vissuto a pieno l'esperienza universitaria.

Infine, dedico questa tesi a me stesso, ai miei sacrifici e alla mia tenacia che mi hanno permesso di arrivare fin qui.

Indice

Elenco delle figure	IV
Elenco delle tabelle	VI
1 Introduzione	1
1.1 Scopo del documento	1
1.2 Inquadramento generale	2
1.3 Convenzioni tipografiche	2
2 Le aziende	3
2.1 Innovery S.p.A.	3
2.1.1 Profilo aziendale	3
2.1.2 Servizi offerti	3
2.2 SIA S.p.A.	5
2.2.1 Profilo aziendale	5
2.2.2 Servizi offerti	5
3 Stage	7
3.1 Introduzione al progetto	7
3.1.1 IBM® Sterling B2B Integrator	7
3.1.2 Architettura già esistente	8
3.1.3 Processo di onboarding flussi	10
3.2 Obiettivi	11
3.2.1 Obiettivi personali	11
3.2.2 Obiettivi dell'azienda	11
3.3 Piano di lavoro	11
3.3.1 Diagramma di Gantt	13
4 Strumenti e tecnologie	14
4.1 Spring	14
4.1.1 Spring Boot	15
4.1.2 Spring Data	15
4.1.3 Spring Security	16
4.2 Angular	17
4.2.1 Bootstrap	17

5	Analisi	18
5.1	Scopo del progetto	18
5.2	Attori del sistema	19
5.3	Casi d'uso	19
5.3.1	Casi d'uso formato breve	19
5.3.2	Casi d'uso formato dettagliato	20
5.3.3	Diagramma dei casi d'uso	27
5.3.4	Definizione requisiti	28
6	Progettazione e codifica	31
6.1	Strumenti di supporto	31
6.1.1	Spring Tools	31
6.1.2	GitLab	31
6.1.3	Postman	32
6.2	Progettazione	32
6.2.1	Front-end	33
6.2.2	Back-end	34
7	Verifica e validazione	38
7.1	Analisi statica	38
7.2	Analisi dinamica	38
8	Conclusioni e sviluppi futuri	39
8.1	Valutazione personale	39
8.2	Sviluppi futuri	39
A	Utilizzo Web App	40
A.1	Accesso all'applicazione	40
A.2	Creazione nuovo flusso	41
A.3	Gestione flussi esistenti	45
A.4	Gestione utenti	46
	Glossario	47
	Riferimenti bibliografici	50

Elenco delle figure

2.1	Logo Innovery S.p.A.	3
2.2	Logo SIA S.p.A.	5
3.1	Architettura software IBM® Sterling B2B Integrator	8
3.2	Architettura installata in SIA S.p.A	9
3.3	Modello del ciclo di sviluppo software Scrum	12
3.4	Diagramma di Gantt per pianificazione dello stage	13
4.1	Logo Spring Framework	14
4.2	Logo Spring Boot	15
4.3	Logo Spring Data	15
4.4	Logo Spring Security	16
4.5	Schema riassuntivo meccanismo JWT	16
4.6	Logo Angular	17
4.7	Logo Bootstrap	17
5.1	Diagramma dei casi d'uso	27
6.1	Logo Spring Tools	31
6.2	Logo GitLab	31
6.3	Logo Postman	32
6.4	Architettura software Web Application [13]	33
6.5	File di configurazione del componente <i>router</i>	34
6.6	Logo IBM	37
6.7	Architettura REST API IBM®Sterling B2B Integrator	37
7.1	Logo SonarQube	38
A.1	Pagina di login dell'applicazione	40
A.2	Pagina principale dell'applicazione	41
A.3	Pagina di retrieve	41
A.4	Parametri recupero file SFTP	42
A.5	Parametri ricezione file	42
A.6	Parametri identificazione flusso	43
A.7	Operazione selezionabili	43
A.8	Modifica o utilizzo operazione	44

A.9 Selezione destinatario	44
A.10 Sezione gestione flussi	45
A.11 Risultato ricerca flussi	45
A.12 Eliminazione flusso	46
A.13 Gestione utenze	46

Elenco delle tabelle

5.1	Tabella requisiti funzionali	29
5.2	Tabella requisiti qualitativi	30
5.3	Tabella requisiti di vincolo	30

Capitolo 1

Introduzione

1.1 Scopo del documento

Questo elaborato è la relazione finale sul lavoro svolto dal laureando Oleg Stoianov durante il periodo di stage presso l'azienda Innovery S.p.a come conclusione del percorso di laurea triennale. Lo scopo è descrivere dettagliatamente le mansioni svolte, dimostrando di aver compreso e messo in atto tutte le competenze acquisite durante questo corso di studi e complementate da una formazione specifica aziendale.

Il documento è strutturato in maniera tale da:

- Dare un'idea del contesto lavorativo in cui è stato svolto lo stage
- Elencare gli obiettivi personali e aziendali
- Definire le richieste del committente del progetto
- Riportare la pianificazione del lavoro
- Riportare l'analisi effettuata
- Spiegare la soluzione proposta
- Descrivere le tecnologie e gli strumenti utilizzati
- Riportare il risultato, conclusioni e sviluppi futuri

1.2 Inquadramento generale

Nel corso di tale attività ho avuto la possibilità di approfondire il mondo Web, in particolare ciò che riguarda le *Web Application* e lo sviluppo di quest'ultime.

Le *Web Application* sono delle applicazioni che implementano il paradigma Client/Server, ovvero un modello di interazione tra processi software, che si suddividono tra *client*, che richiedono i servizi, e *server*, che offrono servizi. L'interazione *client-server* richiede perciò una precisa definizione di un'interfaccia di servizi, che elenca quelli messi a disposizione dal *server*. Quindi il termine *Web Application* identifica un'applicazione risiedente in un Server Web alla quale si accede tramite Client Web, per esempio un *browser*, o un altro programma con funzioni di navigazione operante secondo gli standard del [WWW](#).

Per quanto riguarda lo sviluppo della *Web Application*, ho avuto modo di lavorare sia lato *client* sia lato *server*, occupando una mansione di sviluppatore [FULL-STACK](#).

Per lato *server*, detto anche *back-end* si intende tutta la parte logico-applicativa di un sito Web, con il quale l'utente interagisce indirettamente, che ne permette il funzionamento, la gestione e persistenza dei dati e il reperimento degli stessi. Le tecnologie ed i [framework](#) utilizzati in questo ambito sono state: Java Standard Edition, Java Enterprise Edition, Spring, Spring Boot, Spring Data, Spring Security.

Per lato *client*, detto anche *front-end* si indica la parte visibile all'utente che naviga un sito web e con cui egli può interagire, tipicamente un'interfaccia utente. In questa parte è stato utilizzato principalmente il [framework](#) Angular.

L'obiettivo della *Web Application* è quello ottimizzare il processo di *onboarding* dei flussi logici, ovvero permettere di velocizzare le operazioni di inserimento e di lettura dei dati all'interno di un database. I dati in questione, vengono poi letti da un'altra applicazione software, un [System Integrator](#), principalmente utilizzato dall'azienda SIA S.p.A per la gestione, il monitoraggio, l'amministrazione, l'instradamento di volumi elevati di file in entrata e in uscita. Una volta letti, il [System Integrator](#) li utilizzerà per la creazione delle regole alla base dell'instradamento dei file. Nello specifico, all'arrivo di un file trasmesso da un [business partner](#) A, il [System Integrator](#) controllerà le regole presenti all'interno del database ed in base a quest'ultime, incanalerà il file verso il [business partner](#) B.

1.3 Convenzioni tipografiche

Le convenzioni tipografiche utilizzate in questo documento sono:

- Tutti i termini usati nel documento che richiedono una breve spiegazione aggiuntiva vengono riportati nel glossario, situato a fine documento.
- Le occorrenze dei termini riportati nel glossario sono riportate con [questa](#) convenzione.
- I termini in lingua straniera o appartenenti al gergo tecnico sono evidenziati con il carattere *corsivo*.

Capitolo 2

Le aziende

2.1 Innovery S.p.A.



Figura 2.1: Logo Innovery S.p.A.

2.1.1 Profilo aziendale

Innovery è una società multinazionale nata nel 2001, che opera nell'area dei servizi [ICT](#) per le medie e grandi Aziende. Negli anni ha esteso il suo portfolio a tutte le aree della sicurezza informatica, coprendone tutti gli aspetti. Questo ha permesso di ampliare il suo mercato spingendosi sui territori internazionali. Ad oggi, infatti conta 10 sedi in tutto il mondo, coprendo oltre il territorio italiano, anche quello spagnolo e latinoamericano. Innovery offre soluzioni e servizi personalizzati, per soddisfare le esigenze specifiche dei clienti, completi di progettazione, realizzazione e supporto [\[16\]](#).

2.1.2 Servizi offerti

I servizi sono articolati in 10 unità operative: *Security, Managed Services, Big Data, Mobile Solutions, E-Business, Billing System, High Performance Computing, Physical Security, Health Care, Funded Projects* [\[17\]](#).

Tra i più rilevanti abbiamo :

- *Security Privacy Governance*

La *Security Privacy Governance* rappresenta lo sguardo più generale sulla sicurezza

e la privacy; è essenziale per una *Privacy by Design & by Default* e per ottenere la conformità alle norme sulla sicurezza e sulla privacy (GDPR).

- *Security Operation Center*

La gestione unitaria della sicurezza dei sistemi richiede l'impiego di specifiche tecnologie e di team strategici di personale qualificato, in grado di intervenire su ogni aspetto dell'in-frastruttura tecnologica e dell'organizzazione aziendale: si tratta dei *Security Operation Center* (SOC). Una specifica business unit di Innovery fornisce soluzioni e servizi SOC che consentono il monitoraggio delle infrastrutture informatiche, il rilevamento di anomalie, la gestione degli incidenti. Tutti i servizi sono basati sulle più avanzate soluzioni di mercato per le attività di *Security Information e Event Management* - SIEM (ArcSight, Splunk, RSA).

- *Identity & Access Management*

L'*Identity & Access Management* (IAM) è il punto di partenza essenziale di ogni sistema di gestione della sicurezza delle informazioni: si tratta di disporre di soluzioni integrate che consentano l'identificazione di individui e componenti dei sistemi, e di stabilire quando e quali azioni possano svolgere sulle diverse risorse aziendali.

- *Ethical Hacking/Forensic Analysis*

L'attività di *Vulnerability Assessment e Penetration Testing* (VA/PT) consente di avere una piena consapevolezza dello stato dell'infrastruttura e di tracciare un'efficace *Remediation Roadmap* delle vulnerabilità identificate

- *Network Security*

La messa in sicurezza delle reti è la prima tappa della roadmap verso un'infrastruttura tecnologica sicura. Per la sicurezza dei dati e delle infrastrutture offre servizi specialistici basati sulle migliori tecnologie: *Firewall*; *Intrusion Detection & Prevention System* (IDS/IPS); *End Point Protection System*; *Advanced Persistent Threat* (APT) *Defense System*.

- *B2B Integration & Managed File Transfer*

L'integrazione *End-to-End* e l'efficienza del flusso di transazioni fra i sistemi interni e la propria "comunità di business" è garantito da robusti strumenti come l'IBM Sterling B2B Integrator and *Managed File Transfer Solutions* (Sterling MFT). Sulla base di tale piattaforma tecnologica, Innovery offre soluzioni come l'Innovery User Gateway (IUG), una console Web di gestione centralizzata delle funzionalità della piattaforma Sterling Integrator;

2.2 SIA S.p.A.



Figura 2.2: Logo SIA S.p.A.

2.2.1 Profilo aziendale

SIA - società controllata da CDP Equity - è leader europeo nella progettazione, realizzazione e gestione di infrastrutture e servizi tecnologici dedicati alle Istituzioni Finanziarie, Banche Centrali, Imprese e Pubbliche Amministrazioni, nei segmenti *Card & Merchant Solutions*, *Digital Payment Solutions* e *Capital Market & Network Solutions*. Il Gruppo SIA eroga servizi in oltre 50 paesi [18].

2.2.2 Servizi offerti

Tra i servizi offerti: processing delle carte di credito e debito, pagamenti elettronici, servizi di rete, piattaforme per i mercati finanziari [19].

In particolar modo:

- *Card Management*
Soluzione innovativa e sicura di card management per singoli istituti di credito, grandi gruppi bancari, processor di carte e aziende. SIA supporta gli issuer nell'emissione di tutti i tipi di carte.
- Sistemi di pagamento
Lo sviluppo di strumenti di pagamento innovativi, in grado di sfruttare le potenzialità offerte dalla tecnologia informatica, consente di modernizzare le abitudini di pagamento dei cittadini, delle imprese e della Pubblica amministrazione, migliorare la fluidità delle transazioni, sostenere la crescita economica [25].

Le soluzioni si suddividono in:

- Servizi di tesoreria
- *Processing*
- Accesso al *Clearing*
- *Settlement*

- Servizi di rete
SIAnet Financial Ring, si tratta di un'infrastruttura di connettività multisede a 10 Gbps sicura, affidabile e a bassa latenza, accessibile tramite 9 *Point of Presence* (PoP) a Milano, Londra, New York, Budapest e Francoforte che offrono una soluzione di connettività diretta e dedicata ai principali CSD, broker e sedi di negoziazione, consentendo in tal modo di ridurre i costi legati alla complessità.
- *Blockchain*
Gestione digitalizzata delle fideiussioni basata su tecnologia blockchain. Le garanzie fidejussorie sono per legge obbligatorie per alcune tipologie di rapporti economici come, ad esempio, negli appalti pubblici nella P.A., nei contratti di fornitura tra aziende e nel versamento di anticipi per acquisto di case in costruzione tra privati. Alle garanzie fidejussorie sono legate alcune criticità come la falsificazione dei documenti, le frodi da parte di soggetti non abilitati, la gestione manuale delle pratiche in forma cartacea e i lunghi tempi del processo di rilascio. L'utilizzo della *Distributed Ledger Technology* (DLT) permette di dematerializzare l'iter di rilascio delle polizze fideiussorie da parte di banche, intermediari finanziari, assicurazioni e certificare in modo univoco e irrevocabile tali garanzie.
- Sicurezza fisica
SIA intelliFENCE è il brand con cui Emmeecom firma le sue soluzioni per l'antintrusione, la videosorveglianza e il *safety*: una suite modulare e integrata, basata sui protocolli di comunicazione standard del settore bancario e offerta nella logica più vicina alle esigenze del cliente.
- Gestione Documentale
 - Fatturazione Elettronica
Il processo di fatturazione elettronica, completamente automatizzato secondo una logica *Straight Thought Processing* (STP), integrato con i sistemi informativi gestionali e con i sistemi di *Corporate Banking*, consente di eliminare la discontinuità dei flussi informativi recuperare efficienza riconciliare in modo automatico le informazioni finanziarie e commerciali migliorare la gestione dei flussi di cassa.
 - Conservazione Digitale
Il servizio di conservazione digitale, conserva qualsiasi tipologia di documento e assicura gli appropriati criteri di ricerca. Il sistema supporta un set di tipologie documentali predefinite (contratti, ordinativi, fatture, log di sistema, etc) e permette di personalizzarle e crearne di nuove, tutte caratterizzate da dati di ricerca e dal *workflow* di conservazione totalmente configurabil

Capitolo 3

Stage

3.1 Introduzione al progetto

Lo stage ha avuto una durata di 3 mesi, dal 07/10/2019 al 07/01/2020 presso la sede milanese di SIA S.p.A in Via Privata Francesco Gonin 36.

Come accennato nell' introduzione, il contenuto dello stage verte principalmente sullo sviluppo di una *Web Application*, commissionata dall'azienda SIA S.p.A all'azienda Innove-ry S.p.A. Il progetto in questione ha come obiettivo principale l'ottimizzazione del processo di *onboarding* di flussi logici che devono essere condivisi tra Business Partners e che devono passare attraverso un [System Integrator](#), *IBM® Sterling B2B Integrator*.

3.1.1 IBM® Sterling B2B Integrator

IBM® Sterling B2B Integrator è un motore di transazione che esegue *Business Process*, supporta lo scambio di messaggi elettronici ad alto volume, l'instradamento complesso, la traduzione, il trasferimento gestito di file ([MFT](#)) e l'integrazione flessibile con più sistemi interni e partner commerciali esterni. Include strumenti di gestione visiva per la configurazione e la visibilità dei flussi di lavoro, le attività del sistema e dei partner commerciali, le mappe di traduzione e l'implementazione dei *Business Process*. Funziona con centinaia di protocolli aziendali e standard di comunicazione emergenti [26].

Tra i componenti principali di questo complesso sistema abbiamo:

- *Business Process*: un flusso di attività ordinato e orientato al raggiungimento degli obiettivi aziendali. Ogni *Business Process* è definito da un unico documento [BPML](#).
- *Services*: una risorsa che è possibile configurare per svolgere un'attività.
- *Adapters*: casi speciali di *Services* che interagiscono con sistemi esterni o che archiviano o gestiscono dati di stato al di fuori del contesto del flusso di lavoro.

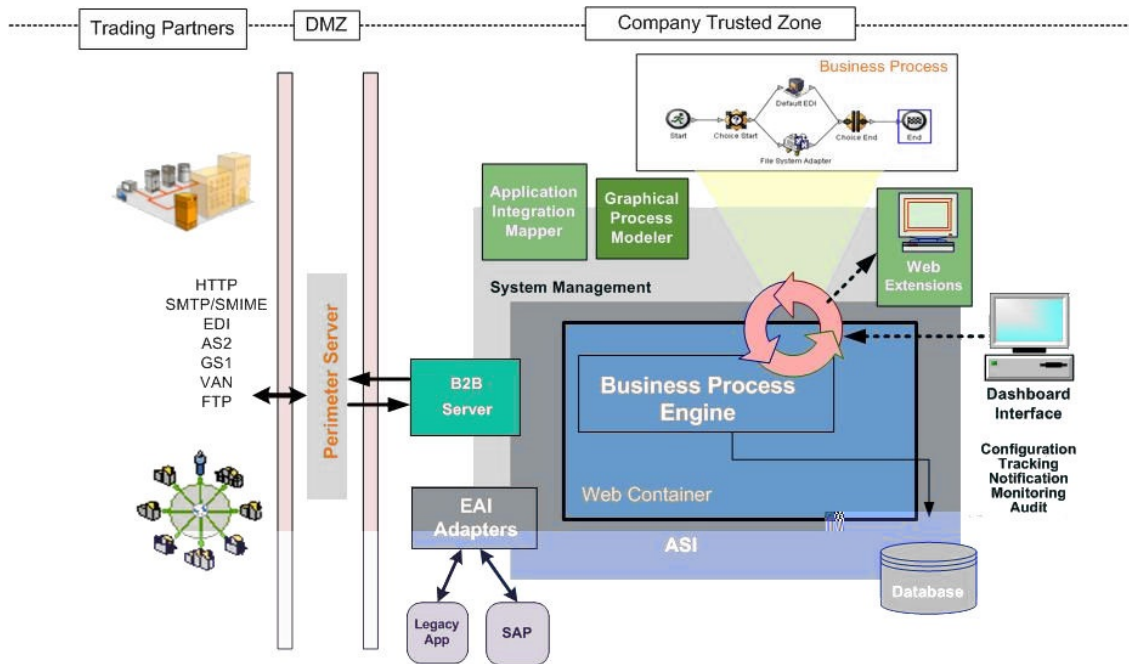


Figura 3.1: Architettura software IBM® Sterling B2B Integrator

3.1.2 Architettura già esistente

L'azienda SIA S.p.a presenta un'infrastruttura di tipo *cluster*, un insieme di computer connessi tra loro tramite una rete telematica. Scopo del *cluster* è distribuire un'elaborazione molto complessa tra i vari computer, aumentando la potenza di calcolo del sistema e garantendo una maggiore disponibilità di servizio, a prezzo di un maggior costo e complessità di gestione dell'infrastruttura [3]. Come si evince dalla figura 3.2 sono presenti due macchine sulle quali è installato IBM® Sterling B2B Integrator, ambedue i nodi condividono lo stesso database e inoltre si scambiano informazioni riguardanti il carico, lo stato dell'applicazione e del nodo. L'architettura presenta inoltre i seguenti componenti per aumentare la sicurezza dell'infrastruttura:

- *Firewall*: hardware e software per controllare il traffico in ingresso/uscita di una rete fidata rispetto a reti non sicure
- *Application Proxy*: interfaccia di comunicazione in una rete per monitorare il traffico fra applicazioni
- *DMZ*: zona demilitarizzata, indica una rete di computer, che funge tra due reti da zona cuscinetto con un proprio indirizzo IP e le delimita mediante regole di accesso rigide

Per mantenere l'integrità dell'architettura, la *Web Application* verrà installata su entrambi i nodi dove è installato il prodotto IBM® Sterling B2B Integrator, sarà però attivata solo su uno dei due. (In 3.2 IBM® Sterling B2B Integrator è abbreviato con SFG)

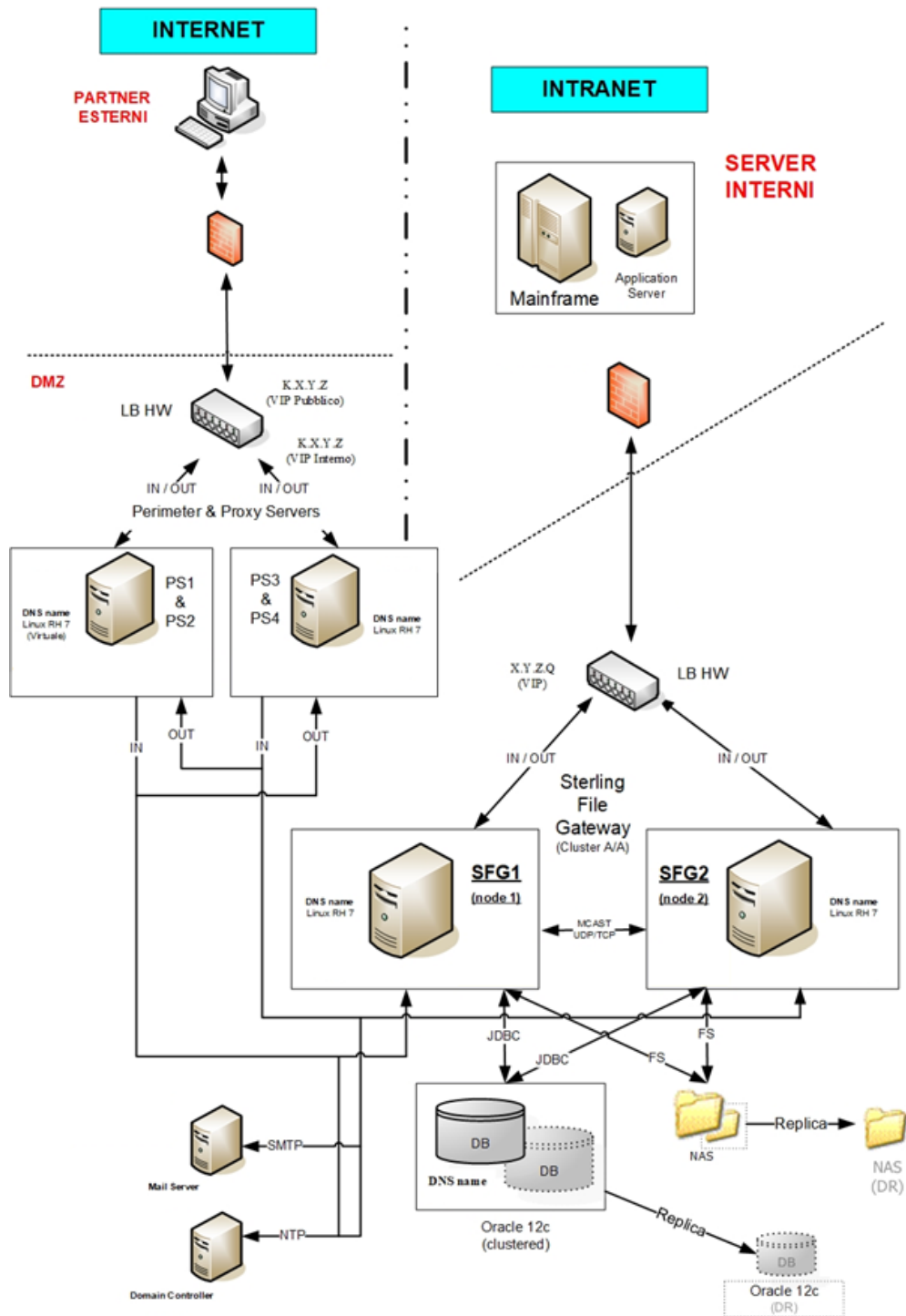


Figura 3.2: Architettura installata in SIA S.p.A

3.1.3 Processo di onboarding flussi

Per processo di *onboarding* di flussi logici, si intende la registrazione di informazioni utili per la gestione e l'implementazione di regole di instradamento per i file che passano attraverso il prodotto IBM® Sterling B2B Integrator. Le informazioni necessarie possono essere suddivise in 3 categorie: informazioni sistemistiche, operazionali e metadati.

Per informazioni sistemistiche intendiamo tutte quelle informazioni riguardanti:

- Indirizzo IP
- Porta TCP/IP
- Protocollo utilizzato

Queste informazioni servono per poter definire "*come*" un [business partner](#) ha intenzione di interfacciarsi con il [System Integrator](#), con lo scopo di inviare e/o ricevere file.

Le informazioni operazionali rappresentano informazioni riguardanti alle operazioni che possono essere effettuate sul file prima di essere consegnato al [business partner](#) destinatario. Tra le operazioni possibili abbiamo:

- Ridenominazione: rinomina il nome del file.
- Backup: salva una copia del file.
- Compressione: effettua una riduzione delle dimensioni del file attraverso uno specifico algoritmo.
- Decompressione: riporta un file precedentemente compresso alle condizioni iniziali.
- Cifratura: applica un algoritmo di cifratura specifico per rendere il contenuto del file incomprensibile.
- Decifrazione: applica un algoritmo di cifratura specifico per rendere il contenuto del file leggibile.
- Codifica [charset](#): converte i caratteri del [charset](#) iniziale in caratteri di un altro [charset](#).

Per poter instradare correttamente un file i metadati necessari sono: [business partner](#) mittente, destinatario e regola che identifica univocamente il file. La regola richiesta è nella nomenclatura del file da gestire, quest'ultima dovrà avere all'inizio del nome del file la seguente sintassi:

SERVICE.ABI.FILETYPE.[Nome file originale]

I 3 identificatori sono separati da un punto, hanno una lunghezza prefissata e vengono concordati in fase di analisi con i [business partners](#). Il primo identificatore indica il servizio di cui il file fa parte, il secondo denota il [codice ABI](#) del mittente mentre il terzo il tipo file. Nello specifico il terzo elemento denota il formato e la logica del contenuto del file.

Come accennato precedentemente, all'arrivo di un file, il prodotto IBM® Sterling B2B Integrator utilizzerà queste informazioni per poter correttamente instradare il file verso il destinatario.

3.2 Obiettivi

3.2.1 Obiettivi personali

Gli obiettivi prefissati inizialmente sono stati :

- Approfondimento delle conoscenze nell’ambito sviluppo Web, in particolar modo dello sviluppo [FULL-STACK](#), lavorando dunque sia sulla parte riguardante il *back-end* e quella riguardante il *front-end* (cfr. 1.2).
- Collaborare con un team di sviluppo con esperienza, che applichi le [best practices](#) per lo sviluppo e la metodologia *agile* (cfr. 3.3).
- Acquisire competenze nello sviluppo di software che ha lo scopo di integrarsi con altre soluzioni già esistenti.
- Approfondimento delle conoscenze nell’ambito bancario/finanziario, per quanto riguarda pagamenti, carte, banche..

3.2.2 Obiettivi dell’azienda

Gli obiettivi dell’azienda SIA S.p.A. per questo progetto sono stati:

- Ottimizzare il processo *onboarding* di flussi logici che devono essere condivisi tra [business partners](#) e che devono passare attraverso il [System Integrator IBM® Sterling B2B Integrator](#).
- Aumentare il livello di personalizzazione dei propri prodotti.
- Mantenere l’architettura già esistente intatta.

3.3 Piano di lavoro

Per il modello di sviluppo si è scelto di utilizzare il paradigma *agile*, esso definisce principalmente quattro fondamenti, ovvero: [11] [10]

- Le persone e le interazioni sono più importanti dei processi e degli strumenti;
- È più importante avere software funzionante che documentazione;
- Bisogna collaborare con i clienti oltre che rispettare il contratto;
- Bisogna essere pronti a rispondere ai cambiamenti oltre che aderire alla pianificazione;

In particolare la metodologia *agile* utilizzata in questo contesto è stata *Scrum*. Si tratta di [framework](#) per la gestione del ciclo di sviluppo del software, iterativo ed incrementale, concepito per gestire progetti e prodotti software o applicazioni di sviluppo. *Scrum* struttura il proprio sviluppo in cicli chiamati *Sprint* che durano da una a quattro settimane, al termine dei quali il team di sviluppo rilascia funzionalità immediatamente testabili.

I cicli sono *timeboxed*, il che significa che hanno durata fissa nel tempo, non possono essere estesi e terminano anche se il lavoro non è stato ultimato. All'inizio di ogni *Sprint*, durante un evento chiamato *Sprint Planning*, il team seleziona i propri task da una lista di attività priorizzate *Product Backlog* e si impegna a completare tutte le attività selezionate entro il termine dello *Sprint*. Il fine ultimo non è completare il maggior numero di attività possibile, ma produrre incrementi di software utilizzabili, raggiungendo lo *Sprint Goal* ovvero l'obiettivo che si desidera ottenere durante l'iterazione.

Il team si confronta quotidianamente mediante il *Daily Scrum Meeting*, una riunione di pochi minuti con lo scopo di fare un recap degli obiettivi del giorno. Al termine di ogni *Sprint* vi sono due incontri fondamentali: lo *Sprint Review* e il *Retrospective Meeting*. Il primo è un meeting informale in cui il team valida il lavoro svolto, il secondo che in genere è svolto al termine del primo, serve al team per riflettere sullo Sprint appena concluso [12].

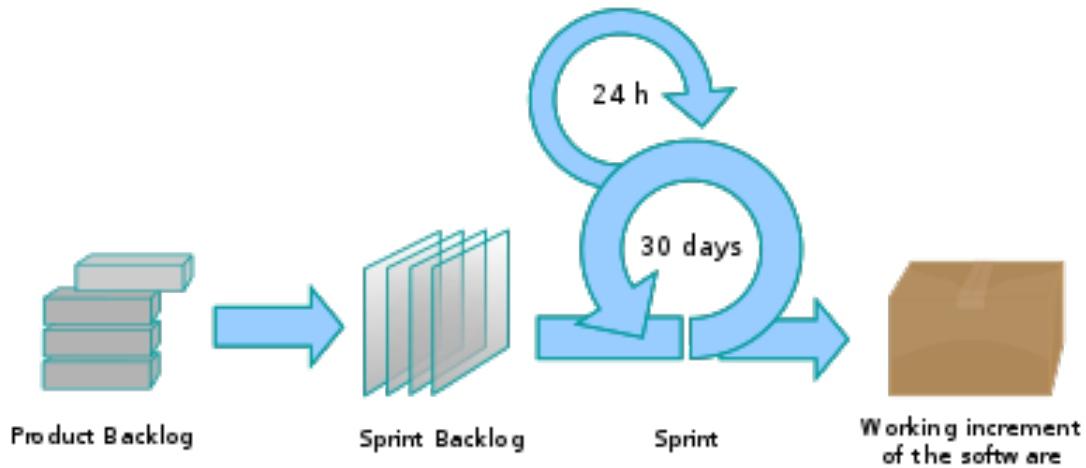


Figura 3.3: Modello del ciclo di sviluppo software Scrum

3.3.1 Diagramma di Gantt

Di seguito viene riportato il [diagramma di Gantt](#) che indica la suddivisione del lavoro durante tutto il periodo di stage. Il seguente diagramma è stato tracciato con il tutor aziendale Nicola Frongia, e condiviso con il team di sviluppo.

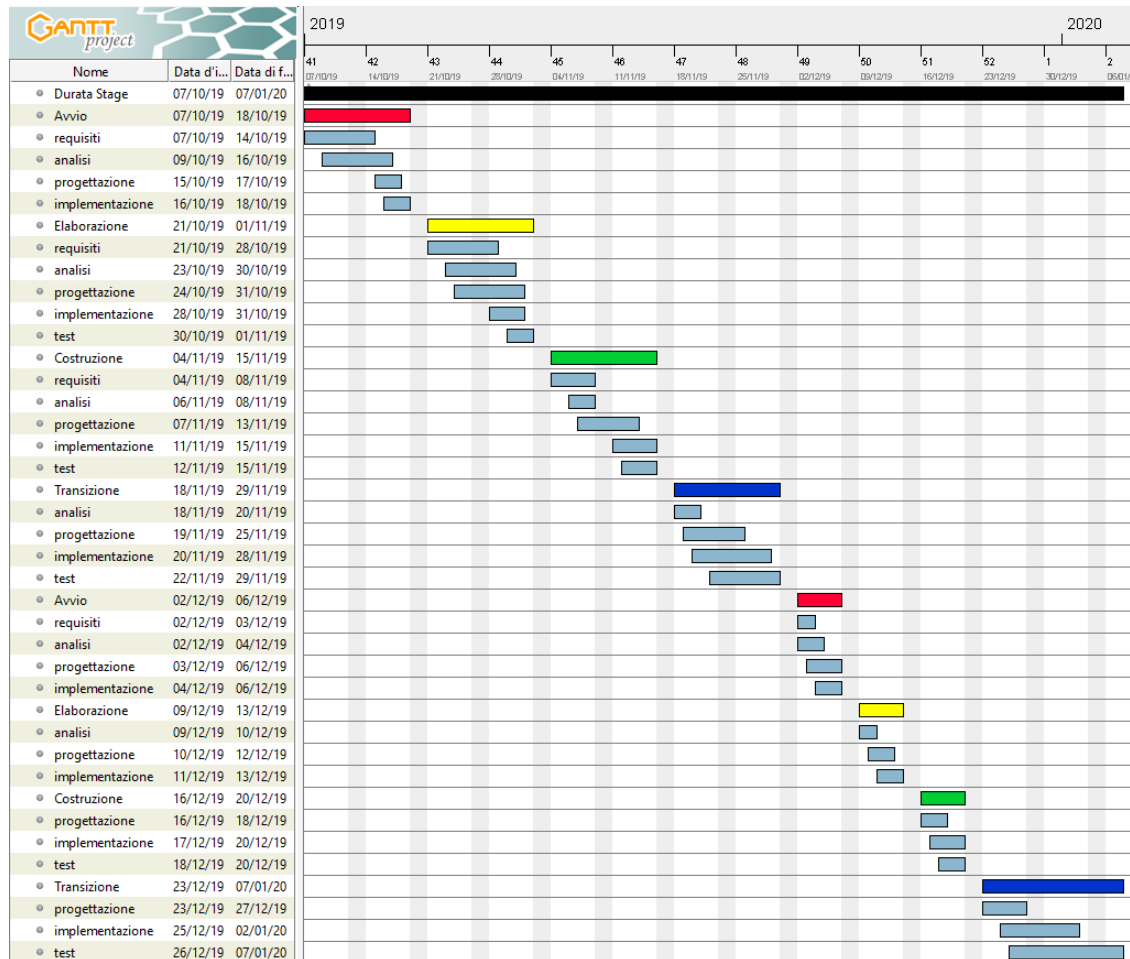


Figura 3.4: Diagramma di Gantt per pianificazione dello stage

Capitolo 4

Strumenti e tecnologie

In questa sezione vengono introdotti gli strumenti e le tecnologie utilizzate per lo sviluppo della *Web Application*. La scelta di questi si è basata principalmente sulle conoscenze e sull'esperienza del team di sviluppo.

4.1 Spring



Figura 4.1: Logo Spring Framework

Per il *back-end* è stato scelto Spring, che è un [framework](#) molto utilizzato e supportato per le moderne applicazioni basate su Java. Il lavoro principale svolto da Spring è la semplificazione dell'unione e dei collegamenti delle parti dell'applicazione, permettendo agli sviluppatori di concentrarsi unicamente sulla logica di business. È basato su concetti chiave come l'*inversion of control* e la *dependency injection* [21].

L'*inversion of control* è un pattern per cui un componente di livello applicativo riceve il controllo da un componente appartenente a una libreria. Lo scopo di questo pattern è quello di rendere le parti del sistema software più indipendenti fra loro, ottenendo così la possibilità di effettuare modifiche senza coinvolgere altre parti del sistema [8] [9].

La *dependency injection* è il secondo concetto chiave, attraverso questo [design pattern](#) è possibile implementare l'*inversion of control*. È una tecnica in cui un oggetto acquisisce le dipendenze necessarie in fase di creazione, quando gli viene *iniettato* un oggetto, il quale viene poi utilizzato come servizio dall'oggetto principale [4] [9].

4.1.1 Spring Boot



Figura 4.2: Logo Spring Boot

Spring Boot è una soluzione "*convention over configuration*" riduce la complessità di configurazione di nuovi progetti Spring. A questo scopo, Spring Boot definisce una configurazione di base che include le linee guida per l'uso del [framework](#) e tutte le librerie di terze parti rilevanti, rendendo quindi l'avvio di nuovi progetti il più semplice possibile [22].

4.1.2 Spring Data



Figura 4.3: Logo Spring Data

Spring Data ha lo scopo di fornire un modello di programmazione per l'accesso ai dati e per la gestione della persistenza di essi. Definisce un'insieme di [API](#) per semplificare l'utilizzo di tecnologie che si interfacciano con database relazionali e non. Questo progetto contiene molti sottoprogetti specifici di un determinato database. In particolare nella *Web Application* è stato utilizzato Spring Data JPA che fornisce l'accesso a database relazionali e ne permette la gestione tramite oggetti Java. Il [DBMS](#) utilizzato è stato Oracle e tramite questo modello è stato possibile mappare le tabelle del database con oggetti [23].

4.1.3 Spring Security



Figura 4.4: Logo Spring Security

Spring Security è un [framework](#) di autenticazione e controllo degli accessi altamente personalizzabile. È lo standard *de facto* per la protezione delle applicazioni basate su Spring. Per garantire l'accesso alla *Web Application* ai soli utenti autorizzati si è utilizzato inoltre il meccanismo di autenticazione *JWT*. Questo meccanismo è diventato uno standard che viene usato per “regolare” le richieste tra *client* e *server*. Dopo l'esito positivo dell'autenticazione il server genera un *token*, ovvero una stringa *JWT*, contenente informazioni specifiche per l'utente, tra cui anche la scadenza del *token*. Ad ogni successiva richiesta da parte del *client* di accedere a determinate risorse del *server*, verrà richiesto il *token* e avverrà il controllo di esso. Di seguito uno schema riassuntivo del meccanismo :

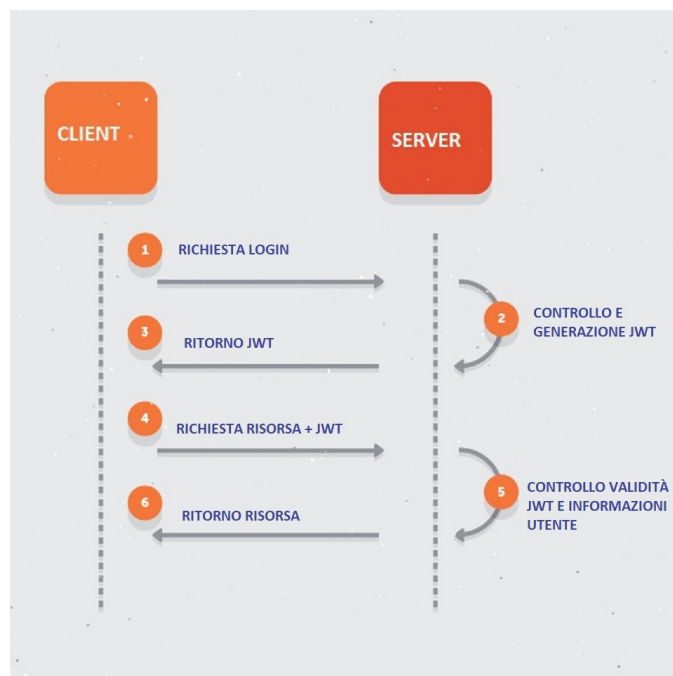


Figura 4.5: Schema riassuntivo meccanismo JWT

4.2 Angular



Figura 4.6: Logo Angular

Angular è un [framework](#) molto utilizzato per lo sviluppo di applicazioni web, in particolare modo della parte riguardante il *front-end*. Sviluppato principalmente da Google, è l'evoluzione del suo predecessore AngularJS, le due versioni non sono compatibili. Il linguaggio di programmazione usato per la prima versione è JavaScript mentre quello di Angular è TypeScript. Quest'ultimo, è un linguaggio di programmazione sviluppato da Microsoft [1]. Nasce dal crescente bisogno di un linguaggio *front-end* per lo sviluppo di applicazioni JavaScript su larga scala e dalla necessità di sicurezza e robustezza. Il linguaggio estende la sintassi di JavaScript in modo che qualunque programma scritto in JavaScript sia anche in grado di funzionare con TypeScript senza nessuna modifica. È destinato a essere compilato in JavaScript per poter essere interpretato da qualunque web browser [28]. I costrutti principali di questo framework sono: [7]

- *Component*: è l'elemento principale di un'applicazione Angular. Esso contiene la logica di interazione dati e utente che definisce l'aspetto ed il comportamento della vista. Un *component* quindi è un singolo elemento dell'interfaccia utente, dotato di caratteristiche univoche e che è possibile collegare chiaramente ad altri elementi dell'interfaccia [5].
- *Service*: è una classe che viene definita per svolgere un compito ben preciso ed effettuare delle operazioni strettamente correlate tenendo in mente il principio di singola responsabilità. Esempi di questi compiti sono: reperimento dei dati dal *server*, gestione degli errori e scambio di dati tra *component* [20].

4.2.1 Bootstrap



Figura 4.7: Logo Bootstrap

Per rendere l'interfaccia grafica più *user-friendly* si è utilizzato Bootstrap, una raccolta di strumenti basati su HTML, CSS e JavaScript che fornisce componenti già formattati e visualmente piacevoli come bottoni, barre di navigazione, *card* e molti altri.

Capitolo 5

Analisi

5.1 Scopo del progetto

La *Web Application* si pone come obiettivo principale, quello di semplificare ed ottimizzare il processo di *onboarding* di flussi logici all'interno di un database. Il raggiungimento di tale obiettivo è subordinato dall'aggiunta di uno strato logico che si colloca sopra le tabelle relazionali presenti all'interno dello [schema](#) del database, tali tabelle contengono la logica di gestione dei file che devono passare attraverso il [System Integrator](#). Lo strato logico che si vuole creare è suddiviso in 3 stati, questi rappresentano i passaggi che un file compie durante il suo ciclo di vita nel [System Integrator](#).

Il primo è lo stato di *retrieve*, in questo stato si definisce se il file in questione viene spedito verso il sistema oppure se il file dev'essere recuperato da un sistema, specificando in ambedue i casi i parametri della macchina esterna che possiede il file. Definita questa fase si presuppone che il file sia già entrato in gestione del [System Integrator](#) e dunque avviene la cosiddetta *routing*, in questa fase vengono definiti i metadati descritti nel paragrafo 3.1.3, quest'ultimi serviranno per identificare ed eseguire le opportune operazioni sul file. L'ultima fase è la *delivery*, durante questo stadio, bisogna definire come dev'essere spedito il file verso il destinatario.

La funzionalità principale richiesta è la creazione di un nuovo flusso logico, esso dunque sarà suddiviso nei 3 stati sopra descritti e ad ogni stato sarà possibile inserire le informazioni necessarie. Il progetto inoltre, deve dare la possibilità all'utente di registrarsi, scegliendo tra utente normale e utente amministratore, quest'ultimo ha permessi più elevati, ad esempio modificare informazioni degli altri utenti, eliminare utenti e visualizzare i [log](#).

Un'ulteriore funzionalità richiesta è la possibilità di utilizzare chiavi di ricerca per visualizzare i flussi presenti ed eventualmente poterli modificare se necessario. Per aumentare il livello di integrazione con il sistema esistente è richiesta la possibilità di usufruire delle informazioni già presenti internamente al [System Integrator](#), queste verranno reperite utilizzando le [API](#) che esso mette a disposizione e saranno selezionabili durante il processo di creazione del flusso.

5.2 Attori del sistema

- **Utente:** è l'utilizzatore generico dell'applicazione, facente parte del team di lavoro *sistemisti file transfer*.
- **Amministratore:** è colui che oltre ad utilizzare l'applicazione si occupa anche della manutenzione delle utenze e della visualizzazione dei [log](#) dell'applicazione.
- **IBM®Sterling B2B Integrator:** è il prodotto che utilizzerà i dati inseriti nel database dall'applicazione per la gestione dei file che passano attraverso esso.

5.3 Casi d'uso

In questa sezione verranno descritti i casi d'uso individuati durante le fasi di analisi, ogni caso d'uso ha uno scopo preciso, distinto e univoco. Inizialmente ogni caso d'uso sarà prima descritto utilizzando il formato breve e successivamente con il formato dettagliato. Infine sarà presente il diagramma [UML](#) con lo scopo di mostrare graficamente le interazioni tra gli attori e i casi d'uso.

5.3.1 Casi d'uso formato breve

CUB1-Registrazione: ogni utente per utilizzare l'applicazione dovrà prima effettuare la registrazione, compilando il form con i dati richiesti e selezionando o meno la voce che indica se l'utente si registra come amministratore.

CUB2-Login: ogni utente registrato per utilizzare l'applicazione dovrà effettuare il login utilizzando le credenziali inserite in fase di registrazione.

CUB3-Creazione nuovo flusso: ogni utente può creare un nuovo flusso logico; per completare la creazione di esso, l'utente dovrà inserire per le 3 fasi: *retrieve*, *routing*, *delivery* tutte le informazioni necessarie.

CUB4-Visualizzazione flussi esistenti: utilizzando delle chiavi di ricerca l'utente può ricercare e visualizzare i flussi presenti.

CUB5-Modifica flusso: ogni utente può effettuare la modifica di un determinato flusso, correggendo le informazioni salvate rispetto ad esso.

CUB6-Eliminazione flusso: ogni utente può eliminare tutte le informazioni riguardanti un flusso, rimuovendolo così dal database.

CUB7-Gestione utenti: l'utente amministratore può visualizzare, modificare ed eliminare i dati relativi ad altri utenti.

CUB8-Visualizzazione log: l'utente amministratore può visualizzare la tabella dei [log](#), contenente in ordine cronologico le azioni compiute dagli utenti che utilizzano l'applicazione.

CUB9-Riutilizzo operazioni esistenti: durante la creazione di un flusso, in particolare durante lo stato di *routing*, l'utente può selezionare ed utilizzare delle operazioni già definite oppure può scegliere di crearne di nuove.

5.3.2 Casi d'uso formato dettagliato

Caso d'uso CUD1: Registrazione (CUB1)

Portata: Applicazione *onboarding* flussi

Livello: Obiettivo utente

Attore primario: Utente non registrato

Attore secondario: Sistema

Attore finale: Utente non registrato

Parti interessate e interessi: Utente non registrato, vuole creare una nuova utenza per poter accedere alle funzionalità del sistema.

Pre-condizioni: L'utente che vuole registrarsi deve avere una email non ancora associata a nessun altro account.

Garanzia di successo: Il sistema registra i dati relativi all'account creato. L'utente viene indirizzato alla pagina login dove può effettuare l'accesso tramite le credenziali indicate in fase di registrazione.

Scenario principale di successo:

1. L'utente non essendo ancora registrato accede alla pagina di registrazione
2. L'utente inserisce i dati richiesti, assicurandosi di avere una mail non ancora utilizzata per altre utenze
3. L'utente seleziona l'opzione utente normale oppure utente amministratore
4. Il sistema crea l'utenza
5. L'utente viene indirizzato alla pagina di login

Estensioni(o passaggi alternativi) :

2. (a) Inserimento mail già in uso
 - i. Il sistema segnala l'errore e lo notifica all'utente
- (b) Inserimento password non corretto
 - i. Il sistema segnala l'errore e lo notifica all'utente

Caso d'uso CUD2: Login (CUB2)

Portata: Applicazione *onboarding* flussi

Livello: Obiettivo utente

Attore primario: Utente

Attore secondario: Sistema

Attore finale: Utente

Parti interessate e interessi: Utente registrato, vuole poter accedere all'applicazione tramite le sue credenziali, inserite in fase di registrazione.

Pre-condizioni: L'utente deve aver già effettuato la fase di registrazione al sistema.

Garanzia di successo: L'utente viene autenticato dal sistema e viene indirizzato verso la pagina principale dell'applicazione.

Scenario principale di successo:

1. L'utente non ha ancora effettuato il login all'applicazione
2. L'utente inserisce le credenziali del suo account, inserite in fase di registrazione
3. Il sistema verifica le credenziali
4. L'utente accede al suo account e viene indirizzato verso la pagina principale dell'applicazione

Estensioni(o passaggi alternativi) :

1. (a) L'utente ha già effettuato il login in precedenza senza aver effettuato successivamente un logout
 - i. L'utente accede automaticamente senza passare dalla schermata di login
3. (a) L'utente inserisce credenziali non valide
 - i. Il sistema segnala l'errore e lo notifica all'utente

Caso d'uso CUD3: Creazione nuovo flusso (CUB3)

Portata: Applicazione *onboarding* flussi

Livello: Obiettivo utente

Attore primario: Utente

Attore secondario: Sistema, IBM®Sterling B2B Integrator

Attore finale: Utente

Parti interessate e interessi:

Utente, vuole poter inserire le informazioni riguardanti al flusso che sta registrando.

IBM®Sterling B2B Integrator, attraverso l'utilizzo le API messe a disposizione rende possibile la scelta di alcuni parametri da inserire.

Pre-condizioni: L'utente deve aver già effettuato la registrazione all'applicazione ed aver effettuato successivamente il login.

Garanzia di successo: L'utente viene guidato dall'applicazione nell'inserimento dei dati necessari per la corretta creazione del flusso. L'inserimento avviene con un messaggio di successo e i dati vengono inviati correttamente al database.

Scenario principale di successo:

1. L'utente entra nella sezione creazione nuovo flusso
2. L'utente inserisce le informazioni riguardanti alla fase di *retrieve*
 - (a) L'utente specifica che il [System Integrator](#) si occuperà del recupero del file
 - (b) L'utente inserisce i dati riguardati alla macchina esterna al sistema che possiede i file, specificando [indirizzo IP](#), Porta [TCP/IP](#) e protocollo di comunicazione
 - (c) In base al protocollo scelto l'utente dovrà inserire le informazioni richieste da esso per la corretta configurazione
3. L'utente passa all'inserimento delle informazioni riguardanti alla fase di *routing*
 - (a) L'utente inserisce i metadati riguardanti all'identificazione del flusso, secondo la naming descritta in 3.1.3
 - (b) L'utente inserisce il numero e l'ordine delle operazioni che vuole siano effettuate durante questa fase
 - (c) Per ogni operazione selezionata, l'utente specifica i parametri necessari per l'operazione (cfr. 5.3.2)
4. L'utente inserisce i dati riguardanti alla fase di *delivery*
 - (a) L'utente specifica il destinatario del file
 - (b) L'utente visualizza le informazioni riguardanti al destinatario recuperate dal sistema
5. Il sistema notifica l'utente del salvataggio corretto del flusso creato

Estensioni(o passaggi alternativi) :

2.
 - (a) L'utente specifica che il file verrà inviato verso il [System Integrator](#)
 - (b) L'utente inserisce i dati riguardanti al protocollo attraverso il quale il file verrà inviato da parte del mittente
3.
 - (a) Il sistema si accorge che l'identificatore del flusso è già in uso
 - i. Il sistema segnala l'errore e lo notifica all'utente
5.
 - (a) Il database verso il quale il sistema è in scrittura non risponde
 - i. Il sistema segnala l'errore e lo notifica all'utente
 - (b) Il sistema si accorge di un errore generico nel flusso
 - i. Il sistema segnala l'errore e lo notifica all'utente

Caso d'uso CUD4: Visualizzazione flussi esistenti (CUB4)

Portata: Applicazione *onboarding* flussi

Livello: Obiettivo utente

Attore primario: Utente

Attore secondario: Sistema

Attore finale: Utente

Parti interessate e interessi: Utente, vuole poter utilizzare chiavi di ricerca per visualizzare tutti i flussi che rispettano tale ricerca, ordinare i flussi in ordine alfabetico oppure in ordine cronologico dalla data di creazione.

Pre-condizioni: L'utente deve aver già effettuato la registrazione all'applicazione ed aver effettuato successivamente il login.

Garanzia di successo: L'utente seleziona le chiavi di ricerca d'interesse, il sistema mostra la lista dei flussi che rispettano la ricerca. I flussi vengono ordinati in base alle richieste dell'utente.

Scenario principale di successo:

1. L'utente entra nella sezione visualizzazione flussi
2. L'utente seleziona le chiavi di ricerca d'interesse
3. Il sistema mostra il risultato della ricerca
4. L'utente seleziona un ordinamento su una delle chiavi
5. Il sistema dispone i flussi secondo il nuovo ordinamento

Estensioni(o passaggi alternativi) :

3. (a) Non è presente nessun risultato che soddisfa i criteri di ricerca
 - i. Il sistema notifica l'utente

Caso d'uso CUD5: Modifica flusso (CUB5)

Portata: Applicazione *onboarding* flussi

Livello: Obiettivo utente

Attore primario: Utente

Attore secondario: Sistema

Attore finale: Utente

Parti interessate e interessi: Utente, vuole poter modificare le informazioni riguardanti un flusso.

Pre-condizioni: L'utente deve aver effettuato la ricerca del flusso che vuole aggiornare.

Garanzia di successo: L'utente modifica dei parametri di un flusso e quest'ultimo viene aggiornato e salvato all'interno del database.

Scenario principale di successo:

1. L'utente seleziona il flusso che ha intenzione di modificare
2. L'utente modifica i parametri che vuole aggiornare
3. L'utente salva le modifiche
4. Il sistema notifica l'utente del corretto aggiornamento del flusso.

Estensioni(o passaggi alternativi) :

4. (a) Il sistema si accorge di un errore generico nel flusso modificato
 - i. Il sistema segnala l'errore e lo notifica all'utente
- (b) Il database verso il quale il sistema è in scrittura non risponde
 - i. Il sistema segnala l'errore e lo notifica all'utente

Caso d'uso CUD6: Eliminazione flusso (CUB6)

Portata: Applicazione *onboarding* flussi

Livello: Obiettivo utente

Attore primario: Utente

Attore secondario: Sistema

Attore finale: Utente

Parti interessate e interessi: Utente, vuole poter eliminare dal database un flusso.

Pre-condizioni: L'utente deve aver effettuato la ricerca del flusso che vuole eliminare.

Garanzia di successo: L'utente seleziona di voler eliminare un determinato flusso e da la conferma della scelta.

Scenario principale di successo:

1. L'utente seleziona il flusso che ha intenzione di eliminare
2. Il sistema chiedi la conferma della scelta
3. L'utente conferma
4. Il sistema notifica l'utente della corretta eliminazione del flusso.

Estensioni(o passaggi alternativi) :

3. (a) L'utente annulla la richiesta
 - i. Il sistema notifica l'utente del corretto annullamento della richiesta
4. (a) Il sistema si accorge di un errore generico nel flusso modificato
 - i. Il sistema segnala l'errore e lo notifica all'utente
- (b) Il database verso il quale il sistema è in scrittura non risponde
 - i. Il sistema segnala l'errore e lo notifica all'utente

Caso d'uso CUD7: Gestione utenti (CUB7)

Portata: Applicazione *onboarding* flussi

Livello: Obiettivo utente

Attore primario: Utente amministratore

Attore secondario: Sistema

Attore finale: Utente amministratore

Parti interessate e interessi: Utente amministratore vuole poter modificare, visualizzare ed eliminare gli utenti registrati all'applicazione se necessario.

Pre-condizioni: L'utente amministratore dev'essersi autenticato correttamente.

Garanzia di successo: L'utente amministratore seleziona l'utente da gestire , apporta le opportune modifiche e salva quest'ultime.

Scenario principale di successo:

1. L'amministratore seleziona l'utente da gestire
2. L'amministratore apporta le opportune modifiche
3. L'amministratore salva le modifiche
4. Il sistema notifica l'amministratore del corretto salvataggio delle modifiche

Estensioni(o passaggi alternativi) :

3. (a) L'amministratore annulla la richiesta di modifica
 - i. Il sistema notifica l'utente del corretto annullamento della richiesta
- (a) Il database verso il quale il sistema è in scrittura non risponde
 - i. Il sistema segnala l'errore e lo notifica all'utente

Caso d'uso CUD8: Visualizzazione log (CUB8)

Portata: Applicazione *onboarding* flussi

Livello: Obiettivo utente

Attore primario: Utente amministratore

Attore secondario: Sistema

Attore finale: Utente amministratore

Parti interessate e interessi: Utente amministratore vuole poter consultare i [log](#) dell'applicazione suddivisi per giorno.

Pre-condizioni: L'utente amministratore dev'essersi autenticato correttamente.

Garanzia di successo: L'utente amministratore seleziona un determinato giorno e visualizza correttamente i [log](#) dell'applicazione.

Scenario principale di successo:

1. L'amministratore seleziona il giorno per cui è interessato visualizzare i [log](#)
2. Il sistema mostra i dati richiesti

Estensioni(o passaggi alternativi) :

1. (a) Il giorno richiesto da parte dell'amministratore non è disponibile
 - i. Il sistema notifica l'amministratore
2. (a) Il database verso il quale il sistema è in lettura non risponde
 - i. Il sistema segnala l'errore e lo notifica all'utente

Caso d'uso CUD9: Riutilizzo operazioni esistenti (CUB9)

Portata: Applicazione *onboarding* flussi

Livello: Obiettivo utente

Attore primario: Utente

Attore secondario: Sistema

Attore finale: Utente

Parti interessate e interessi: Utente, durante la fase di *routing* vuole poter riutilizzare le operazioni definite precedentemente.

Pre-condizioni: L'utente ha compilato correttamente i dati della fase di *retrieve*. Nel sistema è presente almeno un'operazione già salvata.

Garanzia di successo: L'utente seleziona un'operazione già salvata precedentemente e la utilizza per il flusso in questione.

Scenario principale di successo:

1. L'utente seleziona una tipologia di operazione per la fase di *routing*
2. Il sistema mostra le operazioni esistenti per la tipologia selezionata
3. L'utente seleziona una delle operazioni esistenti
4. Il sistema mostra la configurazione dell'operazione selezionata
5. L'utente ripete i passi [1-4] da 0 al numero di operazioni necessarie da effettuare per il flusso

Estensioni(o passaggi alternativi) :

2. (a) Non sono presenti operazioni già esistenti per la tipologia di operazione selezionata
 - i. Il sistema notifica l'utente e permette a quest'ultimo di creare una nuova operazione
4. (a) L'utente modifica un parametro della configurazione dell'operazione esistente
 - i. Il sistema crea una nuova configurazione e lascia intatta l'operazione esistente selezionata in precedenza

5.3.3 Diagramma dei casi d'uso

Questo diagramma è una rappresentazione dei requisiti funzionali del sistema, dei servizi offerti da esso e delle interazioni che gli attori hanno con il sistema stesso.

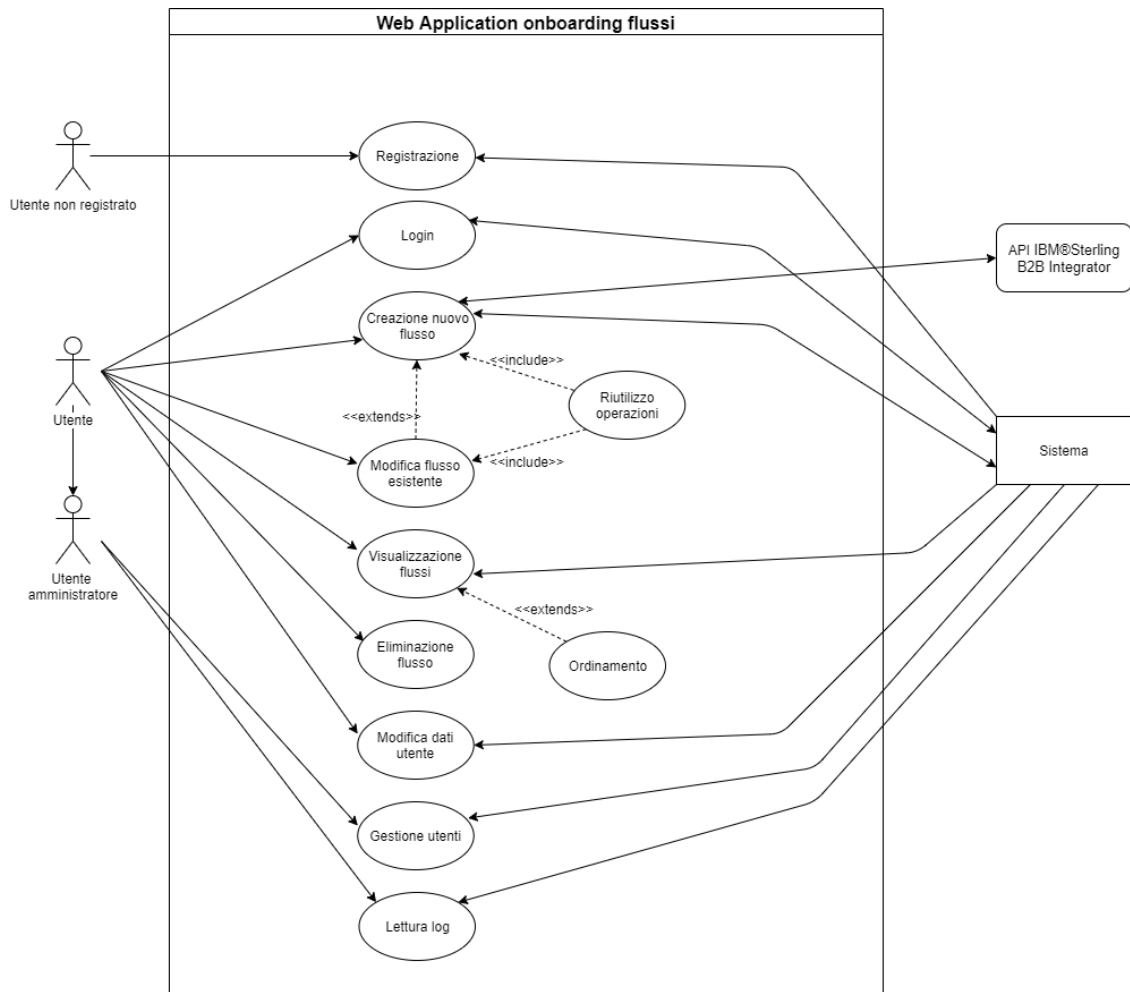


Figura 5.1: Diagramma dei casi d'uso

5.3.4 Definizione requisiti

Di seguito verrà mostrata la classificazione utilizzata per il tracciamento dei requisiti richiesti in realzione ai casi d'uso che li soddisfano:

R [Tipo][Priorità]-[Codice]

dove il significato di ogni parametro della classificazione è esposto di seguito.

- **Tipo:** Funzionale, di qualità, di vincolo
 - **F:** Requisito funzionale. Questi requisiti definiscono una funzionalità del sistema software, in termini di servizi che esso deve fornire, definendone la tipologia degli ingressi e delle uscite, nonché il comportamento;
 - **Q:** Requisito di qualità. Includono requisiti di correttezza, affidabilità, robustezza, efficienza ed usabilità del sistema software;
 - **V:** Requisito di vincolo. Rappresenta una limitazione tecnologica o strategica;
- **Priorità:** Obbligatorio, Desiderabile, Opzionale
 - **O:** obbligatorio, indica un requisito irrinunciabile per il committente;
 - **D:** requisito desiderabile, ma non strettamente necessario;
 - **Z:** opzionale, facoltativo. Verrà soddisfatto solo al completamento di tutti gli altri obiettivi.
- **Codice:** codice numerico identificativo del requisito. È costituito da uno o più numeri interi positivi separati da punti, opportunamente ordinati in maniera gerarchica, nel formato X.Y.Z.

Nella tabella qui di seguito (5.1) verranno riassunti i requisiti funzionali e i casi d'uso ad essi associati.

Requisito	Descrizione	Caso d'uso
RFO-1	L'utente non registrato deve poter registrarsi, inserendo le informazioni obbligatorie richieste.	CUD1
RFO-2	L'utente non registrato deve poter scegliere di registrarsi come utente normale oppure amministratore.	CUD1
RFO-3	L'utente deve poter effettuare il login per accedere alla applicazione.	CUD2
RFO-4	L'utente deve poter inserire le informazioni richieste per la fase di <i>retrieve</i> .	CUD3
RFO-5	L'utente deve poter inserire le informazioni richieste per la fase di <i>routing</i> .	CUD3
RFO-6	L'utente deve poter selezionare tutte le operazioni necessarie per la fase di <i>routing</i> .	CUD3
RFO-7	L'utente deve poter inserire nuove operazioni durante la fase di <i>routing</i> .	CUD3

Requisito	Descrizione	Caso d'uso
RFO-8	L'utente deve poter inserire le informazioni richieste per la fase di <i>delivery</i> .	CUD3
RFO-9	L'utente deve poter visualizzare le informazioni riguardanti il destinatario durante nella fase di <i>delivery</i> .	CUD3
RFO-10	L'utente deve poter selezionare i parametri recuperati dalle API del sistema.	CUD3
RFO-11	L'utente deve poter utilizzare delle chiavi di ricerca per poter visualizzare i flussi salvati.	CUD4
RFO-12	L'utente deve poter selezionare e modificare i parametri di un determinato flusso.	CUD5
RFO-13	L'utente deve poter selezionare ed eliminare un determinato flusso.	CUD6
RFO-14	L'amministratore deve poter visualizzare le informazioni degli altri utenti.	CUD7
RFO-15	L'amministratore deve poter modificare le informazioni degli altri utenti.	CUD7
RFO-16	L'amministratore deve poter eliminare altre utenze.	CUD7
RFO-17	L'amministratore deve poter scegliere di visualizzare i log dell'applicazione relativi ad una giornata.	CUD8
RFO-18	L'utente deve poter riutilizzare le operazioni già definite precedentemente durante la fase di <i>routing</i> .	CUD9

Tabella 5.1: Tabella requisiti funzionali

Nella tabella sottostante (5.2) verranno riassunti i requisiti qualitativi e i casi d'uso ad essi associati.

Requisito	Descrizione	Caso d'uso
RQO-1	Durante l'utilizzo dell'applicazione il sistema notifica l'utente con feedback costanti	-
RQO-2	Il salvataggio delle password degli utenti all'interno del database è criptato.	CUD1
RQO-3	L'interfaccia grafica dell'applicazione dev'essere semplice ed intuitiva.	-
RQO-4	L'interfaccia grafica dell'applicazione deve utilizzare i colori principali del logo dell'azienda committente.	-
RQO-5	Il ripperimento delle informazioni attraverso l'utilizzo delle API messe a disposizione dal sistema dev'essere ragionevolmente veloce.	CUD3
RQD-6	L'utente deve poter selezionare le operazioni che vuole utilizzare attraverso una funzione di tipo " <i>drag-and-drop</i> ".	CUD3
RQO-7	L'utente deve poter ordinare la visualizzazione dei flussi di una ricerca in ordine alfabetico.	CUD4

Requisito	Descrizione	Caso d'uso
RQD-8	L'utente deve poter ordinare la visualizzazione dei flussi di una ricerca in ordine cronologico dalla data di creazione.	CUD4
RQZ-9	L'inserimento dei log nella tabella opportuna del database dev'essere minimale.	CUD8

Tabella 5.2: Tabella requisiti qualitativi

Nella tabella sottostante (5.3) verranno riassunti i requisiti di vincolo e i casi d'uso ad essi associati.

Requisito	Descrizione	Caso d'uso
RVO-1	L'installazione dell'applicazione deve avvenire su entrambi i nodi dell'architettura esistente.	-
RVO-2	L'applicativo deve essere realizzato tramite il framework Angular in versione almeno 8.	-
RVO-3	L'applicativo deve essere realizzato tramite il framework Spring.	-
RVO-4	L'applicativo deve essere realizzato con l'ausilio della libreria Bootstrap.	-
RVO-5	Il DBMS sul quale l'applicazione deve lavorare è Oracle.	-

Tabella 5.3: Tabella requisiti di vincolo

Capitolo 6

Progettazione e codifica

6.1 Strumenti di supporto

Di seguito vengono descritti alcuni degli strumenti di supporto utilizzati per la progettazione e lo sviluppo del software. Il primo è lo strumento utilizzato per la scrittura vera propria del codice, a seguire il *tool* per gestire la continua evoluzione di esso, l'ultimo per la creazione e per il *testing* delle [API](#).

6.1.1 Spring Tools



Figura 6.1: Logo Spring Tools

Spring Tool Suite è un ambiente di sviluppo integrato, derivato dall'[IDE](#) Eclipse. Permette uno sviluppo più veloce di applicativi basati su Spring. Fornisce supporto al linguaggio Java, al [framework](#) Spring ed all'eventuale ambiente di sviluppo [\[24\]](#).

6.1.2 GitLab



Figura 6.2: Logo GitLab

GitLab è un software per il controllo di versione, esso permette la creazione di [repository](#) pubblici o privati, in cui gli sviluppatori possono caricare il proprio codice e gestire le modifiche alle varie versioni in contemporanea al lavoro di più persone [6].

6.1.3 Postman



Figura 6.3: Logo Postman

Postman è un ambiente di sviluppo che aiuta gli sviluppatori a creare, testare, documentare, monitorare e pubblicare documentazione per le loro [API](#) [14]. Le caratteristiche principali di questo strumento sono: [27]

- invio di richieste [API](#).
- [debugging](#) e salvataggio delle risposte.
- organizzazione delle [API](#) in gruppi denominati Raccolte.
- condivisione e collaborazione delle raccolte con il team.

6.2 Progettazione

Nel corso dello sviluppo dell'applicazione il team ha sviluppato l'architettura del progetto cercando di mantenerla il più semplice e lineare possibile. Il risultato della progettazione è un'architettura *multi-tier*, ovvero un'architettura software di tipo *client-server* in cui le varie funzionalità del software sono logicamente separate ovvero suddivise su più strati o livelli software differenti in comunicazione tra loro [2].

L'architettura progettata si presenta suddivisa in cinque strati:

- front-end
- [API](#)
- logica di *business*
- integrazione
- [DAO](#)

Nell'immagine sottostante (6.4) si cerca di dare una visione grafica dell'architettura progettata da parte del team e la comunicazione che avviene tra i vari livelli.

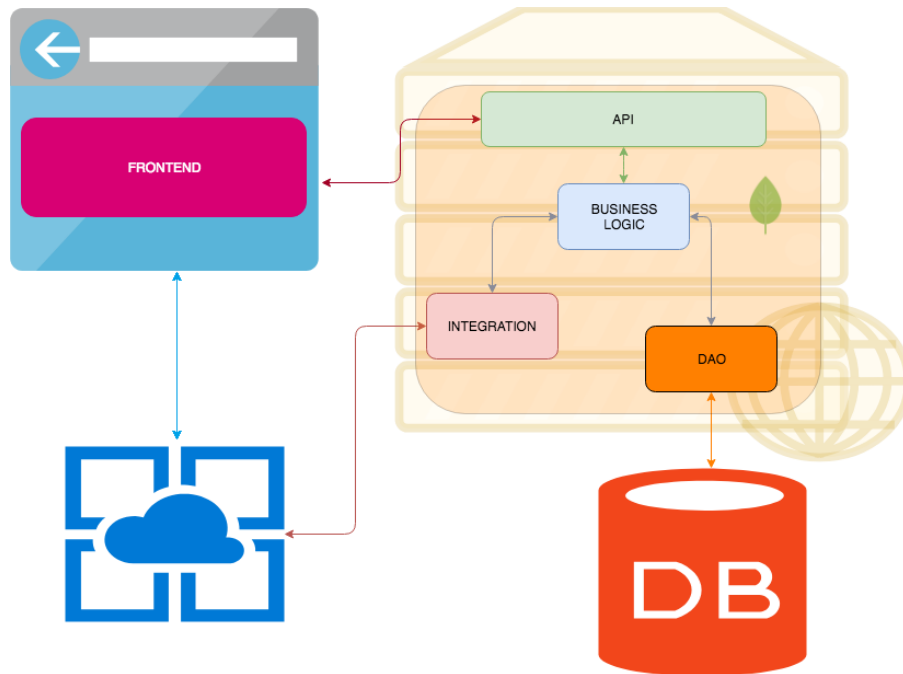


Figura 6.4: Architettura software Web Application [13]

6.2.1 Front-end

Tutti gli [URL](#) navigabili tramite l'applicazione vengono riportati qui di seguito:

- `/` : è la pagina principale, avviata al momento dell'applicazione. Sebbene sia navigabile, non esiste realmente. L'utente viene infatti immediatamente reindirizzato a `/home`;
- `/home`: è la vera pagina principale, nella quale vengono mostrate le possibili attività da svolgere nell'applicazione;
- `/login`: è la pagina nella quale l'utente registrato può effettuare il login con le sue credenziali;
- `/user_management`: è la pagina nella quale l'amministratore può gestire le altre utenze;
- `/new_flow`: è la pagina nella quale l'utente può creare un nuovo flusso logico;
- `/flow_management`: è la pagina nella quale l'utente può ricercare, visualizzare, modificare ed eliminare flussi già esistenti;

- ******: è una qualunque altra pagina che l'utente può digitare, verrà poi automaticamente reindirizzato verso la pagina **/error**;
- **/error**: è la pagina di errore che viene visualizzata in caso di errori o di richieste da parte dell'utente non gestite dall'applicazione;

```
1 const routes: Routes = [  
2   { path: '', redirectTo: '/home', pathMatch: 'full' },  
3   { path: 'home', component: HomeComponent, canActivate:  
     [AuthGuard] },  
4   { path: 'login', component: LoginComponent },  
5   { path: 'user_management', component: UserComponent,  
     canActivate: [AuthGuard], data: { roles: "0" } },  
6   { path: 'new_flow', component: FlowComponent, canActivate:  
     [AuthGuard] },  
7   { path: 'flow_management', component: FlowManagementComponent,  
     canActivate: [AuthGuard] },  
8   { path: '**', redirectTo: '/error' },  
9   { path: '/error', component: ErrorComponent, canActivate:  
     [AuthGuard] }  
10 ];  
11 }
```

Figura 6.5: File di configurazione del componente *router*

6.2.2 Back-end

End-point

Di seguito verranno riportati gli *end-points*, ovvero le [API HTTP](#) messe a disposizione dal *back-end* per essere *consumate* dal *front-end*. Le [API](#) sono raggruppate sulla base del servizio nel quale operano e per il metodo [HTTP](#) (GET, POST, PUT e DELETE) utilizzato.

User Service

- PUT `/users/{id}`: utilizzata per la modifica di un utente esistente dato l'id.
- DELETE `/users/{id}`: utilizzata per l'eliminazione di un utente dato l'id.
- POST `/login`: utilizzata per effettuare l'accesso nell'applicazione.
- POST `/users`: utilizzata per la creazione di un nuovo utente.
- POST `/usersFilter`: utilizzata per effettuare la ricerca sugli utenti.

Operation Service

- GET `/operations`: restituisce la lista di tutte le operazioni possibili.

- GET /reuseOpBkp: restituisce la lista di tutte le operazioni di *backup* esistenti.
- GET /reuseOpComp: restituisce la lista di tutte le operazioni di compressione esistenti.
- GET /reuseOpDecomp: restituisce la lista di tutte le operazioni di decompressione esistenti.
- GET /reuseOpEncr: restituisce la lista di tutte le operazioni di cifratura esistenti.
- GET /reuseOpDec: restituisce la lista di tutte le operazioni di decifrazione esistenti.
- GET /reuseOpRen: restituisce la lista di tutte le operazioni di ridenominazione esistenti.
- GET /reuseOpEnco: restituisce la lista di tutte le operazioni di codifica esistenti.

Backup Operation Service

- GET /backup/{id}: restituisce un'operazione di *backup* dato l'id.
- PUT /backup/{id}: utilizzata per modificare un'operazione di *backup* dato l'id.
- POST /newBackup: utilizzata per la creazione di una nuova operazione di *backup*.

Compress Operation Service

- GET /comp/{id}: restituisce un'operazione di compressione dato l'id.
- PUT /comp/{id}: utilizzata per modificare un'operazione di compressione dato l'id.
- POST /newComp: utilizzata per la creazione di una nuova operazione di compressione.

Decompress Operation Service

- GET /decomp/{id}: restituisce un'operazione di decompressione dato l'id.
- PUT /decomp/{id}: utilizzata per modificare un'operazione di decompressione dato l'id.
- POST /newDecomp: utilizzata per la creazione di una nuova operazione di decompressione.

Encrypt Operation Service

- GET /encrypt/{id}: restituisce un'operazione di cifratura dato l'id.
- PUT /encrypt/{id}: utilizzata per modificare un'operazione di cifratura dato l'id.

- POST /newEncrypt: utilizzata per la creazione di una nuova operazione di cifratura.

Decrypt Operation Service

- GET /decrypt/{id}: restituisce un'operazione di decifrazione dato l'id.
- PUT /decrypt/{id}: utilizzata per modificare un'operazione di decifrazione dato l'id.
- POST /newDecrypt: utilizzata per la creazione di una nuova operazione di decifrazione.

Rename Operation Service

- GET /ren/{id}: restituisce un'operazione di ridenominazione dato l'id.
- PUT /ren/{id}: utilizzata per modificare un'operazione di ridenominazione dato l'id.
- POST /newRen: utilizzata per la creazione di una nuova operazione di ridenominazione.

Encode Operation Service

- GET /encode/{id}: restituisce un'operazione di codifica dato l'id.
- PUT /encode/{id}: utilizzata per modificare un'operazione di codifica dato l'id.
- POST /newEncode: utilizzata per la creazione di una nuova operazione di codifica.

Flow Service

- PUT /flows/{id}: modifica un flusso esistente dato l'id.
- DELETE /flows/{id}: utilizzata per eliminare un flusso esistente dato l'id.
- POST /flows: utilizzata per la creazione di un nuovo flusso.
- POST /flowsFiltered: utilizzata per la ricerca dei flussi esistenti.

Search Service

- GET /findDistProducer: restituisce la lista di tutti i diversi mittenti.
- GET /findDistConsumer: restituisce la lista di tutti i diversi destinatari.
- GET /findDistService: restituisce la lista di tutti i diversi *service*.
- GET /findDistAbi: restituisce la lista di tutti i diversi *abi*.
- GET /findDistFileType: restituisce la lista di tutti i diversi *filetype*.

REST API IBM®Sterling B2B Integrator

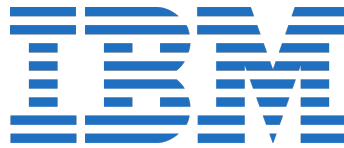


Figura 6.6: Logo IBM

Per l'integrazione tra il prodotto installato e la *Web Application* progettata è stato necessario l'utilizzo delle [REST API](#) messe a disposizione del prodotto. Attraverso esse vengono reperite le informazioni necessarie per la corretta configurazione di un nuovo flusso logico. L'architettura del servizio messo a disposizione è visibile nell'immagine 6.7 sottostante.

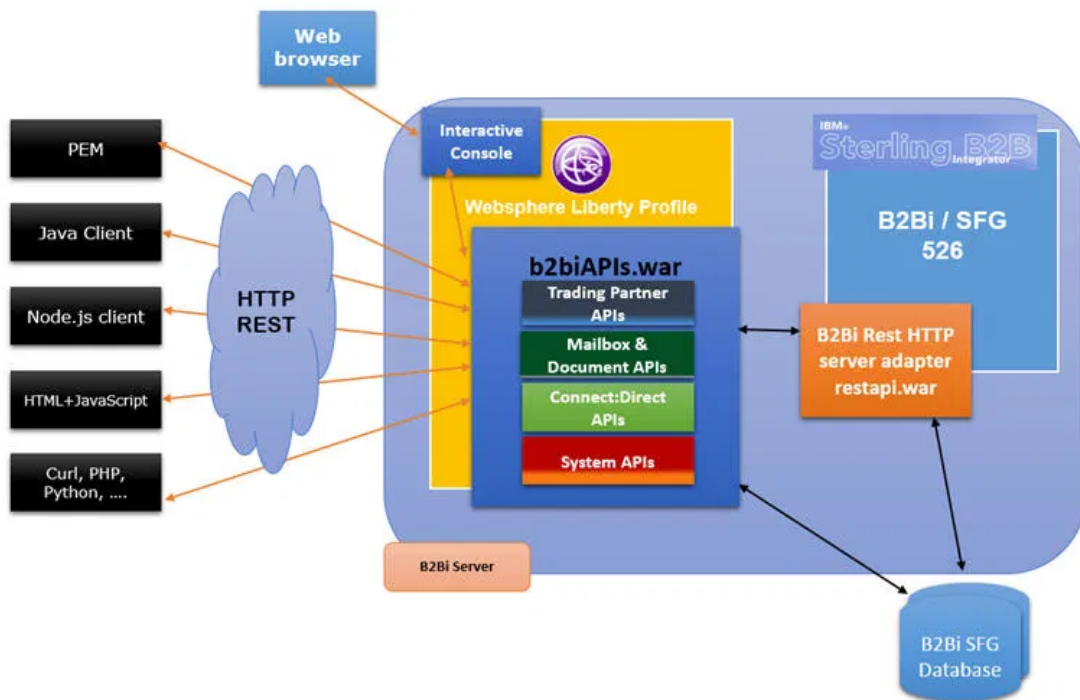


Figura 6.7: Architettura REST API IBM®Sterling B2B Integrator

Capitolo 7

Verifica e validazione

7.1 Analisi statica

Per l'analisi statica ed in particolare per alcune metriche come la complessità ciclomatica, il numero di linee di codice per unità di compilazione e per le [best practices](#) riguardanti la scrittura del codice è stato utilizzato SonarQube.



Figura 7.1: Logo SonarQube

SonarQube è una piattaforma [open source](#) per la gestione della qualità del codice. Nello specifico è un'applicazione web che produce *reports* sul codice duplicato, sugli standards di programmazione, i tests di unità, il *code coverage*, la complessità, i *bugs* potenziali, i commenti, la progettazione e l'architettura [15].

7.2 Analisi dinamica

Per l'analisi dinamica non è stato utilizzato nessuno strumento in particolare, bensì durante le fasi di collaudo con il committente sono stati effettuati i vari test per verificare la correttezza del programma e dell'analisi effettuata. La fase finale di collaudo, effettuata le ultime due settimane prima del rilascio del progetto, è stata fondamentale per la correzione degli ultimi errori rimasti.

Capitolo 8

Conclusioni e sviluppi futuri

8.1 Valutazione personale

La *Web Application* sviluppata è attualmente utilizzata in ambiente di produzione del committente. Posso ritenere dunque, l'esperienza di grande aiuto a livello professionale e personale, poichè mi ha permesso di lavorare ad un progetto importante dove ho potuto mettere in campo tutte le competenze acquisite durante il corso di studio e durante una formazione specifica aziendale. Inoltre, quest'esperienza ha arricchito notevolmente il mio bagaglio di conoscenze e l'aver lavorato in un team di sviluppo già competente ed organizzato ha alzato il mio livello di capacità di collaborazione e di suddivisione dei compiti. Non posso che valutare l'intera esperienza molto positivamente e ringraziare l'azienda presso la quale sono stato assunto dell'opportunità che mi è stata data.

8.2 Sviluppi futuri

Per la *Web application* sono stati già individuati dei possibili sviluppi futuri che il committente potrebbe richiedere:

- Installazione su entrambi i nodi dell'architettura, con la gestione della concorrenza dei flussi da parte dell'applicazione.
- Integrazione con altri sistemi presenti all'interno della struttura del committente.
- Possibilità di copiare flussi esistenti.
- Integrazione con ulteriori [API](#) del sistema.
- Monitoraggio del ciclo di vita di un flusso.

Appendice A

Utilizzo Web App

In questa appendice verrà mostrato, attraverso brevi descrizioni e immagini, il funzionamento dell'applicazione. I dati mostrati sono inventati e hanno il solo scopo di simulare l'utilizzo da parte di un utente della *Web Application*.

A.1 Accesso all'applicazione

Una volta digitato l'indirizzo dell'applicazione, l'utente verrà accolto dalla seguente schermata, dove potrà effettuare l'accesso all'applicazione.

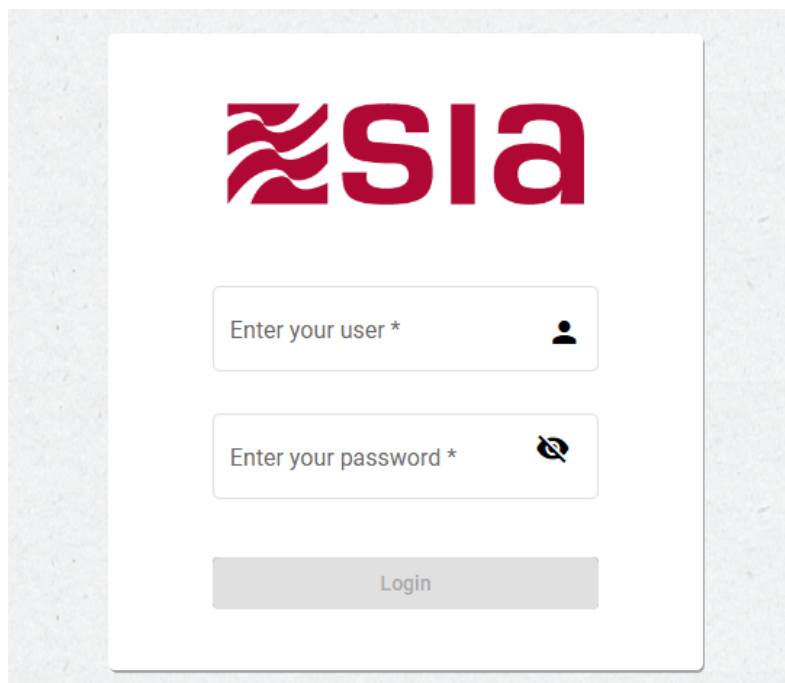


Figura A.1: Pagina di login dell'applicazione

Effettuato l'accesso l'utente sarà indirizzato verso la pagina principale, visibile in A.2.

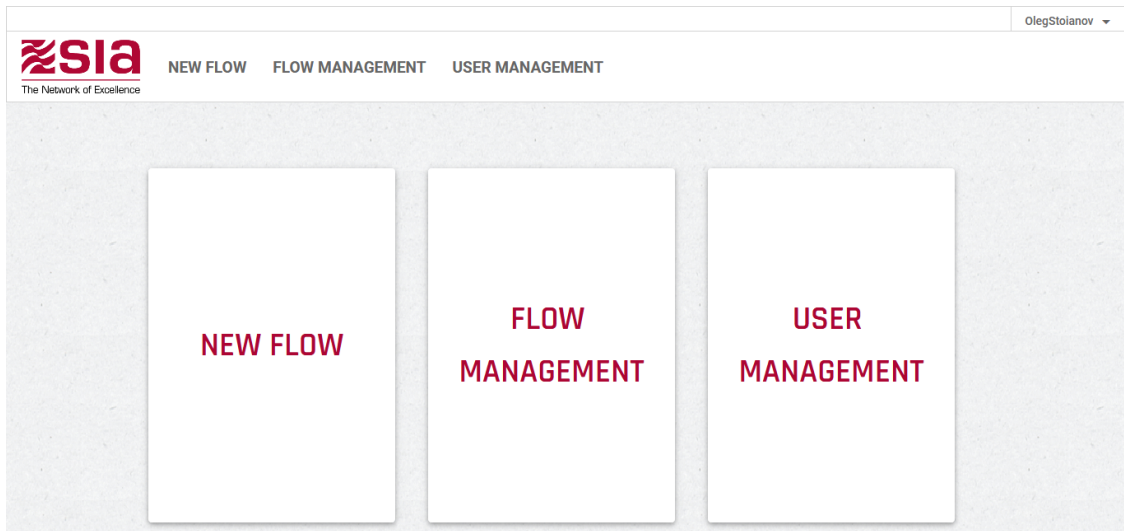


Figura A.2: Pagina principale dell'applicazione

A.2 Creazione nuovo flusso

L'utente una volta recatosi nella sezione per la creazione di un nuovo flusso, visualizza inizialmente la seguente schermata.

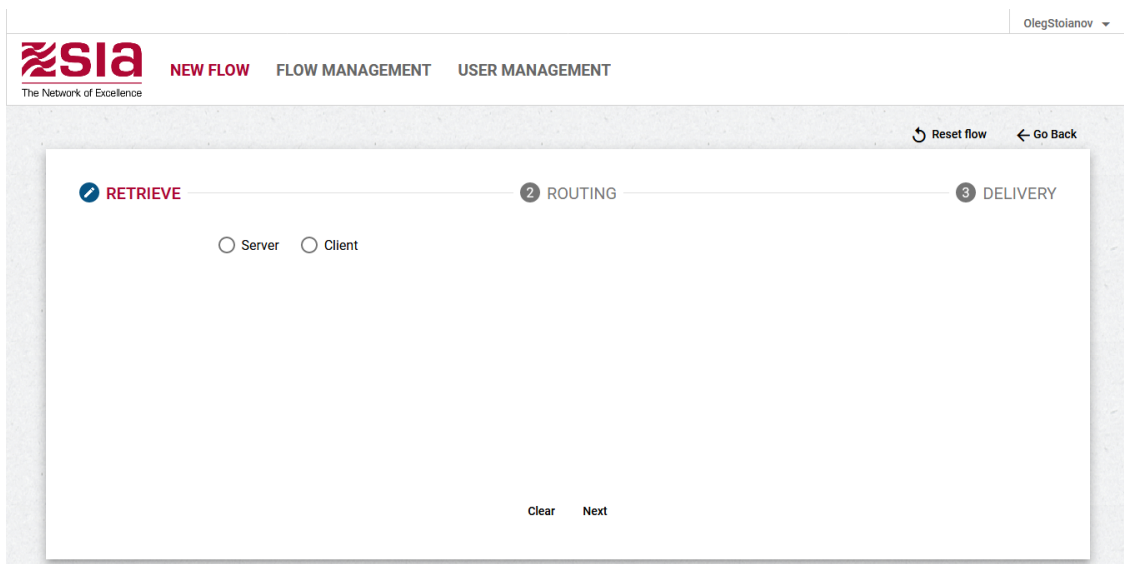


Figura A.3: Pagina di retrieve

L'utente seleziona *client* se il file dev'essere recuperato dal sistema oppure *server* se il file viene inviato al sistema. In ambedue i casi compila le informazioni richieste.

The screenshot shows the 'RETRIEVE' configuration screen in the SIA web application. The 'Client' radio button is selected. The form is divided into two columns. The left column contains fields for 'Producer *', 'Protocol *' (set to SFTP), 'Profile name *', 'Name filter *', 'Name exclusion *', 'Post get operation *', 'Post get value', 'Pre get value', 'Client adapter', and 'Path *'. The right column contains fields for 'Profile Name', 'Remote Host', 'Remote Port', 'Known Host Key', 'Remote User', 'Preferred Authentication Type', 'User Identity Key', 'Directory', 'Connection Retry Count', 'Retry Delay (secs)', 'Response Timeout(secs)', 'Preferred Cipher', and 'Preferred MAC Algorithm'. At the bottom, there are 'Clear' and 'Next' buttons. The top navigation bar includes 'NEW FLOW', 'FLOW MANAGEMENT', and 'USER MANAGEMENT'.

Figura A.4: Parametri recupero file SFTP

The screenshot shows the 'RETRIEVE' configuration screen in the SIA web application. The 'Server' radio button is selected. The form is divided into two columns. The left column contains fields for 'Producer *' and 'Protocol *'. The right column is empty. At the bottom, there are 'Clear' and 'Next' buttons. The top navigation bar includes 'NEW FLOW', 'FLOW MANAGEMENT', and 'USER MANAGEMENT'.

Figura A.5: Parametri ricezione file

Compilati i parametri per la fase di *retrieve*, l'utente deve compilare quelli successivi della *routing*.

SIA The Network of Excellence

NEW FLOW FLOW MANAGEMENT USER MANAGEMENT

OlegStoianov

Reset flow Go Back

RETRIEVE ROUTING DELIVERY

Service * ABI * Filetype * ☐ Active

OPERATIONS

Available Selected

BACKUP
COMMANDLINE
COMPRESS_TAR
DECOMPRESS_TAR
COMPRESS

Back Clear Next

Figura A.6: Parametri identificazione flusso

L'utente trascina le tipologie di operazioni che vuole siano effettuate durante questa fase per il flusso.

OPERATIONS

Available Selected

BACKUP
COMMANDLINE
COMPRESS_TAR
DECOMPRESS_TAR
COMPRESS

ENCRYPT

Figura A.7: Operazione selezionabili

Cliccando sulla chiave inglese di una determinata tipologia di operazione, visibile nella figura A.7, è possibile selezionare un'operazione esistente oppure crearne una ex novo.

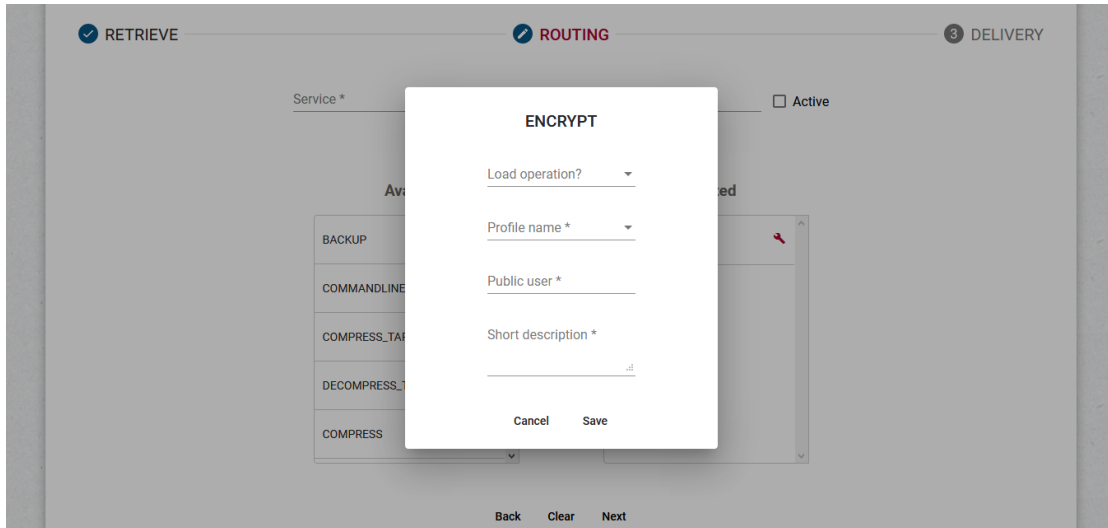


Figura A.8: Modifica o utilizzo operazione

Infine l'utente seleziona il destinatario del flusso, visualizzando le informazioni riguardanti ad esso recuperate dal sistema.

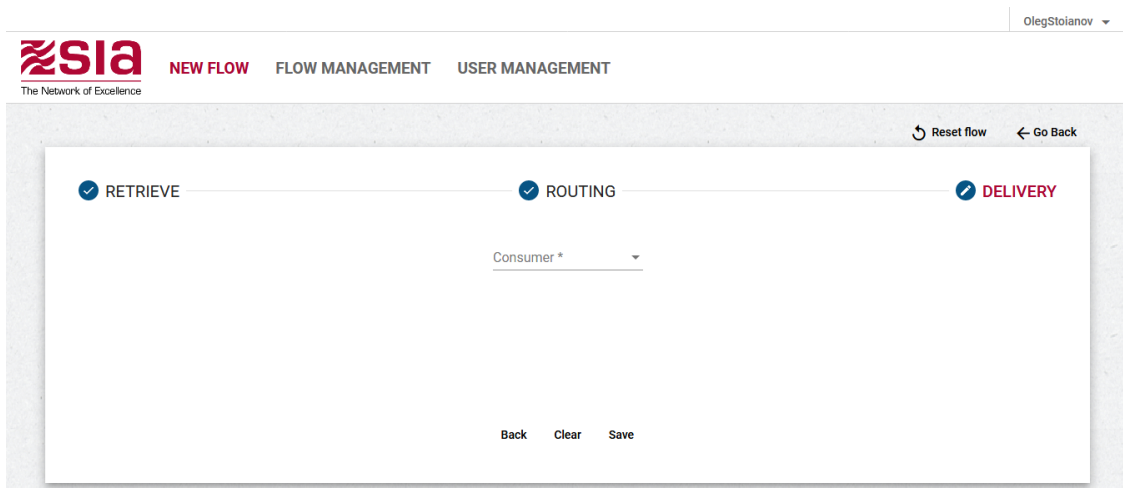


Figura A.9: Selezione destinatario

Per salvare correttamente il flusso creato l'utente deve premere con il cursore sul bottone *Save*, visibile nella parte bassa della figura A.9.

A.3 Gestione flussi esistenti

L'utente visitando la sezione riguardante la gestione dei flussi esistenti, visualizza la schermata sottostante:

The screenshot shows the SIA Flow Management interface. At the top, there is a header with the SIA logo and the text 'The Network of Excellence'. Below the header, there are three tabs: 'NEW FLOW', 'FLOW MANAGEMENT' (which is active), and 'USER MANAGEMENT'. The main area contains a set of filters: 'Producer', 'Consumer', 'Service', 'ABI', 'Filetype', 'Mode', and 'Active'. Each filter has a dropdown arrow. At the bottom right of the filter area, there are two buttons: 'Clear filters' and 'Search'.

Figura A.10: Sezione gestione flussi

Attraverso le chiavi di ricerca, visibili in figura A.10 l'utente può effettuare una ricerca e visualizzare i risultati della ricerca ordinati alfabeticamente per ognuna delle chiavi.

The screenshot shows the SIA Flow Management interface with search results. The filters are the same as in Figure A.10, but the 'Service' filter is set to 'test'. Below the filter area, there is a table with the following data:

Producer	Consumer	Service	ABI	Filetype	Mode	Active	
CD_API	ConsumerSIA	test	123	prova	Client	<input checked="" type="checkbox"/>	Edit Delete
PRODUCER_SFTP	ConsumerSIA	test	test11	12345	Server	<input checked="" type="checkbox"/>	Edit Delete

At the bottom right of the table, there is a pagination bar with the text 'Items per page: 15', '1 - 2 of 2', and navigation arrows.

Figura A.11: Risultato ricerca flussi

Cliccando sulle apposite icone posizionate sulla destra di ogni flusso, l'utente può modificare o eliminare tale flusso. Di seguito la schermata che viene visualizzata relativa all'eliminazione.

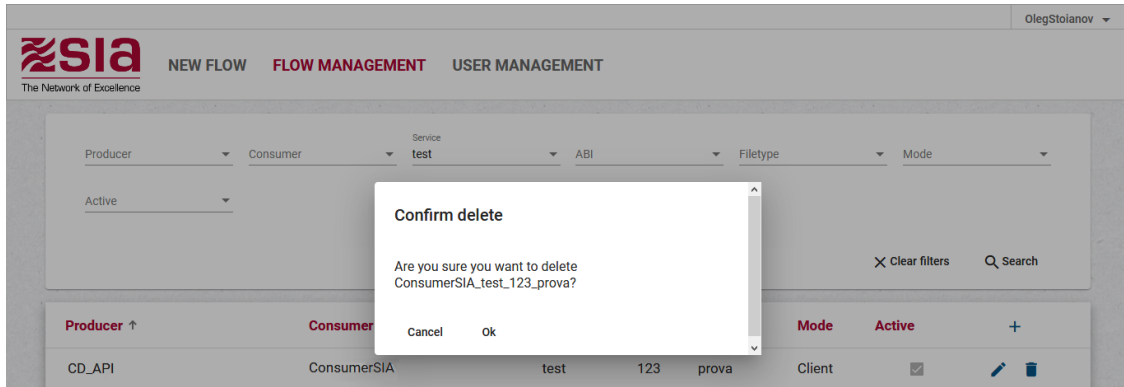


Figura A.12: Eliminazione flusso

A.4 Gestione utenti

L'amministratore può gestire le utenze attraverso la sezione dedicata. Come visibile in figura A.13 anche per quanto riguarda la gestione delle utenze è presente la possibilità di modificarle o eliminarle, cliccando sulle apposite icone posizionate sulla destra.

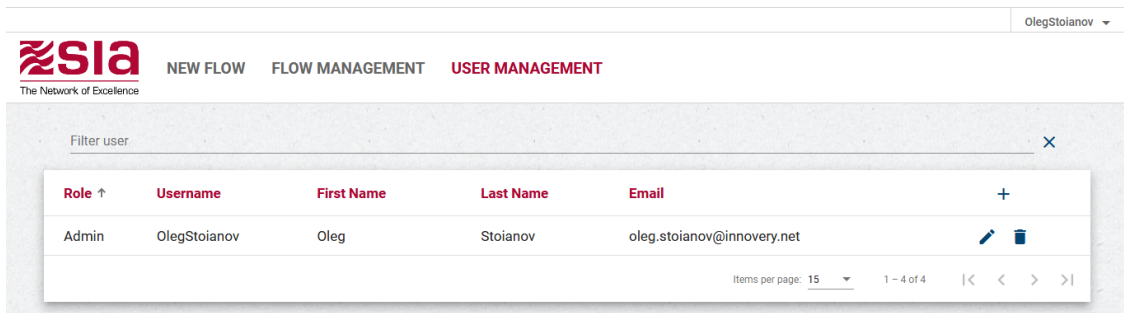


Figura A.13: Gestione utenze

Glossario

API "Application Programming Interface", sono un set di definizioni e protocolli con i quali vengono realizzati e integrati software applicativi. Consentono la comunicazione tra prodotti o servizi con altri prodotti o servizi senza sapere come vengono implementati. 15, 18, 29, 31, 32, 34, 37, 39

best practices Modo di fare che garantisce i migliori risultati in specifiche e note circostanze. 11, 38

BPML "Business Process Modeling Language", è un linguaggio basato su XML per la modellazione dei Business Process. Fornisce un modello di esecuzione astratto per processi aziendali collaborativi e transazionali basato sul concetto di una macchina a stati finiti transazionale. 7

business partner Soggetto che va ad affiancarsi ad un'impresa in un rapporto di collaborazione continuata, in merito alla distribuzione, promozione e vendita dei suoi prodotti. 2, 10, 11

charset è l'associazione fra un insieme di codici numerici e i rispettivi caratteri di un determinato linguaggio. 10

codice ABI "Associazione Bancaria Italiana", è un numero composto da cinque cifre e rappresenta l'istituto di credito. Ogni banca possiede un codice ABI che viene assegnato proprio dall'Associazione bancaria italiana. 10

DAO "Data Access Object", è un pattern architetturale per la gestione della persistenza: si tratta fondamentalmente di una classe con relativi metodi che rappresenta un'entità tabellare di un DBMS relazionale. 32

DBMS "Database Management System", è un sistema software progettato per consentire la creazione, la manipolazione e l'interrogazione efficiente di database. 15, 30

debugging indica l'attività che consiste nell'individuazione e correzione da parte del programmatore di uno o più errori (bug) rilevati nel software. 32

design pattern è un concetto che può essere definito "una soluzione progettuale generale ad un problema ricorrente". Si tratta di una descrizione o modello logico da applicare per la risoluzione di un problema. 14

diagramma di Gantt usato principalmente nelle attività di project management, è costruito partendo da un asse orizzontale a rappresentazione dell'arco temporale totale del progetto e da un asse verticale a rappresentazione delle mansioni o attività che costituiscono il progetto. 13

framework astrazione che fornisce un insieme di classi, strumenti, utilità con funzionalità generiche, adattabili al dominio applicativo specifico per l'utilizzo desiderato. Un framework fornisce una maniera standard di produrre applicazioni che seguono principi consolidati e ne facilita l'integrazione. 2, 11, 14–17, 30, 31

FULL-STACK identifica lo sviluppo di un intero sistema informatico o applicazione dal front-end al back-end e al codice software che li collega. 2, 11

HTTP "HyperText Transfer Protocol", è un protocollo a livello applicativo usato come principale sistema per la trasmissione d'informazioni sul web. 34

ICT "Information and Communication Technologies", ICT è l'insieme delle tecnologie che forniscono l'accesso alle informazioni attraverso le telecomunicazioni. A differenza dell'Information Technology, l'ICT è più focalizzata sulle tecnologie di comunicazione, come internet, reti wireless, telefoni cellulari e altri mezzi di comunicazione. 3

IDE "Integrated Development Environment", è un software che, in fase di programmazione, supporta i programmatori nello sviluppo del codice sorgente di un programma. 31

indirizzo IP è un numero del datagramma IP che identifica univocamente un dispositivo detto host collegato a una rete informatica che utilizza l'Internet Protocol come protocollo di rete per l'instradamento/indirizzamento. 10, 22

log file costituito da un elenco cronologico delle attività svolte da un sistema operativo, da un database o da altri programmi, generato per permettere una successiva verifica. 18, 19, 25, 29, 30

MFT "Managed File Transfer", Il trasferimento di file gestito si riferisce a un software o un servizio che gestisce il trasferimento sicuro di dati da un computer a un altro attraverso una rete (ad esempio Internet). Il software MFT è commercializzato per le aziende come alternativa all'utilizzo di soluzioni di trasferimento di file ad hoc, come FTP, HTTP e altri. 7

open source un software è reso tale per mezzo di una licenza attraverso cui i detentori dei diritti favoriscono la modifica, lo studio, l'utilizzo e la redistribuzione del codice sorgente. Caratteristica principale dunque delle licenze open source è la pubblicazione del codice sorgente. 38

repository è un archivio in grado di contenere dati e relativi metadati. Può offrire un sistema di versionamento in grado di tenere traccia delle modifiche effettuate al suo interno. 32

REST "Representational State Transfer", rappresenta un sistema di trasmissione di dati su HTTP. Il principio fondamentale è l'esistenza di risorse (fonti di informazioni), a cui si può accedere tramite un identificatore globale. 37

schema In un dizionario di dati, uno schema di database indica come le entità che compongono il database si relazionano tra loro, comprese le tabelle, le viste, le procedure memorizzate e altro ancora. 18

System Integrator Nel campo dell'IT gli integratori di sistemi connettono sistemi eterogenei in modo che questi possano comunicare, processare, salvare, categorizzare dati. 2, 7, 10, 11, 18, 22

TCP/IP indica una famiglia di protocolli di rete legati da dipendenze d'uso su cui si basa il funzionamento logico della rete Internet, rappresenta lo standard *de facto* nell'ambito delle reti dati. 10, 22

UML è un linguaggio che permette, tramite l'utilizzo di modelli visuali, di analizzare, descrivere, specificare e documentare un sistema software anche complesso. 19

URL "Uniform Resource Locator", è una sequenza di caratteri che identifica univocamente l'indirizzo di una risorsa su una rete di computer, come ad esempio un documento, un'immagine, un video, tipicamente presente su un host server e resa accessibile a un client. 33

WWW "World Wide Web", è uno dei principali servizi di Internet, che permette di navigare e usufruire di un insieme molto vasto di contenuti amatoriali e professionali (multimediali e non) collegati tra loro attraverso legami (link), e di ulteriori servizi accessibili a tutti o ad una parte selezionata degli utenti di Internet. 2

Bibliografia

- [1] *Angular*. URL: <https://it.wikipedia.org/wiki/Angular>.
- [2] *Architettura multi-tier*. URL: https://it.wikipedia.org/wiki/Architettura_multi-tier.
- [3] *Computer cluster*. URL: https://it.wikipedia.org/wiki/Computer_cluster.
- [4] *Dependency injection*. URL: https://it.wikipedia.org/wiki/Dependency_injection.
- [5] *Displaying data in views*. URL: <https://angular.io/guide/displaying-data>.
- [6] *GitLab*. URL: <https://it.wikipedia.org/wiki/GitLab>.
- [7] *Introduction to Angular concepts*. URL: <https://angular.io/guide/architecture>.
- [8] *Inversion of Control*. URL: https://it.wikipedia.org/wiki/Inversione_del_controllo#cite_note-1.
- [9] *IoC and Dependency Injection with Spring*. URL: <https://www.baeldung.com/inversion-control-and-dependency-injection-in-spring>.
- [10] *Manifesto agile*. URL: <https://agilemanifesto.org/iso/it/manifesto.html>.
- [11] *Metodologia agile*. URL: https://it.wikipedia.org/wiki/Metodologia_agile.
- [12] *Metodologia Scrum*. URL: <https://www.agileway.it/scrum-metodologia-agile/>.
- [13] Pasquale Paola. *Spring boot + Angular*. URL: <https://itnext.io/a-kiss-architecture-springboot-angular-2ca6adf48271>.
- [14] *Postman*. URL: <https://www.postman.com/>.
- [15] *Qualità del codice*. URL: https://www.spazioit.com/pages_it/sol_inf_it/code_quality_it/.
- [16] Innovery S.p.A. *Profilo aziendale*. URL: <https://innovery.net/it/chi-siamo/>.
- [17] Innovery S.p.A. *Servizi offerti*. URL: https://innovery.net/wp-content/uploads/2019/05/INNOVERY_Company-profile_v-1.26-2.pdf.
- [18] Sia S.p.A. *Profilo aziendale*. URL: <https://www.sia.eu/it/gruppo-sia/chi-siamo>.
- [19] Sia S.p.A. *Soluzioni*. URL: <https://www.sia.eu/it/soluzioni>.
- [20] *Servizi e Dependency Injection in Angular*. URL: https://www.mrwebmaster.it/javascript/servizi-dependency-injection-angular_12761.html.

- [21] *Spring*. URL: <https://spring.io/>.
- [22] *Spring boot*. URL: <https://www.ionos.it/digitalguide/siti-web/programmazione-del-sito-web/tutorial-su-spring-boot/>.
- [23] *Spring data*. URL: <https://spring.io/projects/spring-data>.
- [24] *Spring Tools Suite*. URL: https://it.wikipedia.org/wiki/Spring_Framework.
- [25] *Strumenti di pagamento*. URL: <https://www.bancaditalia.it/compiti/sispaga-mercati/strumenti-pagamento>.
- [26] Sterling Sync. *IBM Sterling B2B Integrator*. URL: <https://sterlingsync.com/what-is-ibm-sterling-b2b-integrator/>.
- [27] RIP Tutorial. *Postman*. URL: <https://riptutorial.com/it/postman/example/30196/cos-e-postman->.
- [28] *TypeScript*. URL: <https://it.wikipedia.org/wiki/TypeScript>.