

POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Informatica

Tesi di Laurea Magistrale

**Archivio Digitale: Progettazione e
Implementazione di un sistema a microservizi
per la gestione documentale bancaria**



**Politecnico
di Torino**

Relatore
prof. Antonio Vetrò

Candidato
Gianluca Maida

Marzo 2026

Sommario

La progressiva digitalizzazione dei servizi bancari ha reso necessario ripensare le modalità di distribuzione e consultazione della documentazione finanziaria, superando modelli tradizionali basati sull'invio cartaceo e su canali digitali non sempre accessibili all'intera clientela. In questo contesto si colloca il progetto Archivio Digitale, sviluppato durante un tirocinio curriculare in ambito bancario con l'obiettivo di introdurre un canale dedicato alla consultazione documentale, indipendente dai servizi di home banking ordinari. La soluzione proposta adotta un'architettura a microservizi, strutturata su più livelli applicativi e integrata con sistemi documentali e componenti legacy già presenti nell'ecosistema aziendale. L'approccio progettuale combina modellazione architetturale formale e tecnologie web moderne, al fine di garantire modularità, manutenibilità e sicurezza.

Il lavoro illustra le scelte architetturali e implementative adottate, con particolare attenzione alla progettazione dei servizi backend e ai meccanismi di integrazione con sistemi esterni. Infine, viene presentata una valutazione sperimentale delle prestazioni, finalizzata ad analizzare il comportamento del sistema sotto carico e a discuterne le caratteristiche di scalabilità.

Indice

1	Introduzione	8
1.1	Contesto e motivazioni del lavoro	8
1.2	Scenario applicativo e problema affrontato	9
1.3	Obiettivi della tesi	10
1.4	Ambito applicativo e vincoli di progetto	10
1.5	Metodologia adottata	11
2	Analisi dei requisiti	12
2.1	Perimetro del sistema	12
2.2	Perimetro di utenza e regole di eleggibilità	13
2.3	Requisiti funzionali	14
2.4	Flusso di processo dell'Archivio Digitale	16
2.4.1	Wizard introduttivo	17
2.4.2	Registrazione iniziale	18
2.4.3	Accesso e autenticazione	20
2.4.4	Contenuti e funzionalità	21
2.5	Requisiti non funzionali	23
3	Architettura del sistema	24
3.1	Visione architetturale complessiva	24
3.2	Architettura AS-IS	25
3.2.1	Descrizione dello scenario attuale	25
3.2.2	Limiti dell'architettura AS-IS	25
3.3	Architettura TO-BE	26
3.3.1	Obiettivi dell'evoluzione architetturale	26
3.3.2	Integrazione con i sistemi esistenti	26
3.4	Modellazione architetturale con C4 Model	26
3.4.1	System Context View	27
3.4.2	Container View dell'Archivio Digitale	29
3.4.3	Component View del PostaPersonaleService	32
3.5	Considerazioni architetturali	34

4	Tecnologie e framework di sviluppo	35
4.1	Contesto tecnologico	35
4.2	Stack tecnologico dell'Archivio Digitale	36
4.2.1	Frontend web	37
4.2.2	Backend for Frontend (BFFE)	37
4.2.3	Backend applicativo	38
4.3	Framework proprietario per lo sviluppo dei servizi	39
4.4	Struttura standard dei servizi applicativi	40
4.4.1	Controller e gestione delle API	41
4.4.2	Command e orchestrazione dei flussi	42
4.4.3	Service, Model e Mapper	42
4.5	Tecnologie di integrazione: REST e JPA	42
4.5.1	Integrazione REST	42
4.5.2	Integrazione JPA	43
4.6	Sicurezza applicativa	43
4.6.1	Autenticazione e gestione dei token	44
4.6.2	Autorizzazione e protezione delle operazioni	44
5	Implementazione del sistema	46
5.1	Implementazione dei servizi applicativi	46
5.2	Scelte implementative e standard di sviluppo	47
5.2.1	Approccio contract-first e definizione delle interfacce	47
5.2.2	Struttura interna e flusso esecutivo	47
5.2.3	Scelte implementative sul client REST	48
5.3	Integrazione con i sistemi esterni	48
5.3.1	Integrazione con il sistema documentale	49
5.3.2	Gestione degli errori e dei casi anomali	50
5.4	Caso di studio: PostaPersonaleService	50
5.4.1	Ruolo del servizio nel sistema Archivio Digitale	50
5.4.2	Analisi implementativa del flusso ScaricaDocumenti	51
5.4.3	Considerazioni progettuali	55
6	Valutazione sperimentale e analisi delle prestazioni	57
6.1	Obiettivi della valutazione sperimentale	57
6.2	Metodologia di analisi e metriche adottate (GQM)	58
6.3	Ambiente di test e configurazione sperimentale	59
6.3.1	Ambiente di esecuzione	60
6.3.2	Accesso ai servizi e autenticazione	60
6.3.3	Caratteristiche dei dati di test	60
6.3.4	Strumenti di testing	61
6.4	Test di carico e di stress	62

6.4.1	Scenari di test	62
6.4.2	Configurazione dei test di carico	63
6.4.3	Configurazione dei test di stress	64
6.5	Analisi dei risultati	64
6.5.1	Scenario S1 – Download di un documento	65
6.5.2	Scenario S2 – Download multiplo di 5 documenti	67
6.5.3	Scenario S3 – Download esteso di 20 documenti	70
6.5.4	Confronto tra gli scenari	72
6.6	Discussione dei risultati	74
7	Conclusioni e sviluppi futuri	76
7.1	Sintesi del lavoro svolto	76
7.2	Valutazione critica dei risultati ottenuti	77
7.3	Possibili evoluzioni e sviluppi futuri	78

Elenco delle figure

2.1	Mappa di navigazione dell'Archivio Digitale	16
2.2	Wizard introduttivo	17
2.3	Pagina di registrazione	18
2.4	Verifica del numero di telefono	19
2.5	Verifica dell'indirizzo e-mail	19
2.6	Termini e condizioni del servizio	19
2.7	Esito positivo della registrazione	20
2.8	Pagina di login	20
2.9	Verifica OTP	21
2.10	Cambio password al primo accesso	21
2.11	Pagina introduttiva del portale	22
2.12	Home page: contratti e proposte	22
3.1	System Context View dello scenario AS-IS	28
3.2	System Context View dello scenario TO-BE	29
3.3	Container View dell'Archivio Digitale	30
3.4	Component View del PostaPersonaleService	32
5.1	Class diagram del PostaPersonaleService	52
6.1	Tempo medio di risposta nello scenario S1	65
6.2	Tasso di errore nello scenario S1	66
6.3	Throughput nello scenario S1	67
6.4	Tempo medio di risposta nello scenario S2	68
6.5	Tasso di errore nello scenario S2	69
6.6	Throughput nello scenario S2	70
6.7	Tempo medio di risposta nello scenario S3	71
6.8	Tasso di errore nello scenario S3	71
6.9	Throughput nello scenario S3	72
6.10	Confronto dei tempi medi di risposta tra gli scenari S1, S2 e S3	73
6.11	Confronto del tasso di errore tra gli scenari S1, S2 e S3	73

Elenco delle tabelle

6.1	Risultati dei test per lo scenario S1	65
6.2	Risultati dei test per lo scenario S2	67
6.3	Risultati dei test per lo scenario S3	70

Capitolo 1

Introduzione

1.1 Contesto e motivazioni del lavoro

Negli ultimi anni, il settore bancario è stato interessato da un profondo processo di trasformazione digitale, guidato sia dall'evoluzione delle tecnologie informatiche sia dall'introduzione di normative sempre più stringenti in materia di gestione, conservazione e accessibilità della documentazione. In questo contesto, la digitalizzazione dei processi documentali è diventata un elemento centrale nelle strategie di innovazione delle banche, con l'obiettivo di migliorare l'esperienza del cliente, aumentare l'efficienza operativa e ridurre i costi associati ai canali tradizionali. Una spinta determinante in tale direzione è stata fornita dalla Direttiva (UE) 2021/338 (c.d. MiFID II "Quick Fix")[1], che introduce un'impostazione orientata al "digital-by-default", prevedendo la fornitura in formato elettronico delle comunicazioni rientranti nel perimetro MiFID quale modalità ordinaria. La normativa determina, di fatto, un'inversione del tradizionale modello comunicativo: il digitale diventa la forma primaria di trasmissione della documentazione, mentre la ricezione in formato cartaceo è consentita esclusivamente su richiesta espressa del cliente al dettaglio (opt-in).

In particolare, la gestione della documentazione relativa ai servizi di investimento riveste un ruolo centrale, poiché tali documenti devono essere resi disponibili ai clienti in modo puntuale, sicuro e conforme agli obblighi normativi. Storicamente, una parte significativa di questa documentazione è stata distribuita attraverso canali cartacei o tramite canali digitali ordinari, spesso caratterizzati da vincoli contrattuali e da una complessità operativa non trascurabile.

L'evoluzione del contesto normativo e l'aumento delle aspettative degli utenti hanno reso evidente la necessità di introdurre soluzioni più flessibili e

scalabili, in grado di garantire l'accesso digitale alla documentazione anche a quei clienti che non dispongono di un canale digitale ordinario attivo. In questo scenario si colloca il progetto Archivio Digitale, oggetto del presente lavoro di tesi, che si propone di fornire un canale dedicato alla consultazione documentale, integrato nell'ecosistema applicativo esistente ma allo stesso tempo autonomo rispetto ai canali tradizionali.

Le motivazioni alla base del progetto non sono esclusivamente tecnologiche, ma rispondono a esigenze concrete di business e di compliance: da un lato, la riduzione dei costi e della complessità legata alla gestione del cartaceo; dall'altro, la necessità di garantire continuità, trasparenza e tracciabilità nell'accesso alla documentazione da parte della clientela. In questo senso, l'Archivio Digitale rappresenta un esempio significativo di come le scelte architettoniche e progettuali possano incidere direttamente sull'efficacia di un sistema informativo in un contesto reale e complesso.

1.2 Scenario applicativo e problema affrontato

Il contesto applicativo in cui si inserisce l'Archivio Digitale è caratterizzato dalla presenza di una pluralità di sistemi informativi, sviluppati e stratificati nel tempo, che supportano differenti canali di interazione con il cliente. Tali sistemi includono soluzioni legacy dedicate alla produzione e conservazione della documentazione, nonché piattaforme digitali che consentono l'accesso ai servizi bancari tramite interfacce web.

All'interno di questo ecosistema, la consultazione della documentazione da parte del cliente avviene tradizionalmente attraverso due modalità principali: l'utilizzo di un canale digitale ordinario (Internet Banking), che richiede la sottoscrizione di specifici contratti, oppure la ricezione della documentazione in formato cartaceo. Entrambe le soluzioni presentano tuttavia dei limiti. Nel primo caso, non tutti i clienti sono disposti ad attivare un canale digitale completo; nel secondo, la gestione del cartaceo comporta costi elevati e una minore tempestività nella fruizione delle informazioni.

Il problema affrontato dal progetto Archivio Digitale consiste quindi nella necessità di garantire a tutti i clienti eleggibili la disponibilità di un canale valido per la consultazione della documentazione. L'Archivio Digitale si inserisce in questo quadro come alternativa al canale digitale ordinario o al canale cartaceo, assicurando la copertura digitale nei casi in cui non sia attivo un canale ordinario e non sia stata esercitata l'opzione di ricezione cartacea, mantenendo al contempo un elevato livello di sicurezza e conformità norma-

tiva. La soluzione proposta deve integrarsi con i sistemi esistenti, evitando duplicazioni funzionali e preservando le responsabilità già consolidate, ma allo stesso tempo introdurre un punto di accesso semplice e dedicato alla consultazione documentale.

Dal punto di vista ingegneristico, questo scenario pone diverse sfide, tra cui la progettazione di un'architettura modulare, la gestione delle integrazioni con sistemi esterni e la definizione di flussi di accesso e autenticazione che siano al contempo sicuri e usabili. Il progetto rappresenta quindi un caso di studio concreto per l'analisi di soluzioni architetture in un contesto aziendale reale.

1.3 Obiettivi della tesi

L'obiettivo principale della presente tesi è analizzare, progettare e valutare una soluzione software per la gestione e la consultazione digitale della documentazione in ambito bancario, prendendo come riferimento il progetto Archivio Digitale sviluppato durante un tirocinio curriculare svolto in azienda.

In particolare, la tesi si propone di analizzare i requisiti del sistema, descrivere l'architettura nello stato attuale e nella sua evoluzione, approfondire le scelte tecnologiche adottate e, infine, valutarne il comportamento attraverso un'analisi sperimentale orientata a performance e scalabilità.

Attraverso questi obiettivi, il presente lavoro mira a fornire una visione completa del ciclo di vita di un sistema software complesso, dalla fase di analisi fino alla valutazione sperimentale.

1.4 Ambito applicativo e vincoli di progetto

Il lavoro di tesi è stato svolto nell'ambito di un tirocinio curriculare presso una realtà aziendale operante nel settore tecnologico, coinvolta nello sviluppo di soluzioni software per il settore bancario. Il progetto è stato realizzato all'interno di un team strutturato, seguendo metodologie e standard di sviluppo tipici di contesti aziendali.

L'ambito applicativo del progetto è limitato alla progettazione e allo sviluppo del sistema Archivio Digitale e delle sue integrazioni principali, escludendo la gestione dei sistemi responsabili della produzione della documentazione e delle funzionalità core dei servizi di investimento. Il progetto è stato inoltre soggetto a una serie di vincoli, tra cui l'adozione di uno stack tecnologico predefinito, l'utilizzo di un framework proprietario e la necessità di

garantire l'anonimizzazione delle informazioni sensibili. Questi vincoli hanno avuto un impatto diretto sulle scelte progettuali adottate e hanno costituito un elemento centrale del lavoro svolto.

1.5 Metodologia adottata

La metodologia adottata nel corso del progetto ha seguito un approccio incrementale, articolato in più fasi. In una prima fase è stata condotta un'analisi dei requisiti, basata sull'esame dei flussi funzionali e delle esigenze operative del sistema. Successivamente, è stata definita l'architettura del sistema, con particolare attenzione alla scomposizione in componenti. La fase di implementazione ha previsto lo sviluppo dei servizi applicativi secondo gli standard definiti dal framework adottato, mentre la fase finale del lavoro si è concentrata sulla valutazione sperimentale del sistema. In questa fase sono state definite metriche di interesse e sono stati eseguiti test di carico e di stress, con l'obiettivo di analizzare il comportamento del sistema in condizioni realistiche. Questo approccio ha consentito di mantenere coerenza tra i requisiti iniziali, le scelte progettuali e le attività di verifica sperimentale.

Capitolo 2

Analisi dei requisiti

2.1 Perimetro del sistema

Il sistema di Archivio Digitale nasce con l'obiettivo di fornire ai clienti un canale digitale dedicato alla consultazione della documentazione prodotta nell'ambito dei servizi di investimento, in conformità con le normative vigenti in materia di dematerializzazione, conservazione e accessibilità dei documenti. Il progetto si colloca all'interno di un contesto aziendale articolato, caratterizzato dalla presenza di sistemi legacy e canali digitali consolidati, e si propone come un'evoluzione mirata dei processi di comunicazione documentale verso il cliente.

Uno degli obiettivi principali del sistema è quello di estendere l'accesso digitale alla documentazione anche a quei clienti che, per scelta o per condizioni contrattuali, non dispongono di un canale digitale ordinario attivo. In questo senso, l'Archivio Digitale non si configura come una semplice replica delle funzionalità offerte dai canali esistenti, ma come un canale alternativo e autonomo, progettato per garantire la continuità della consultazione documentale indipendentemente dalla presenza di altri strumenti digitali.

Dal punto di vista funzionale, l'Archivio Digitale consente al cliente finale di accedere in modo sicuro alla documentazione di propria pertinenza, visualizzarla e scaricarla, riducendo la dipendenza dai tradizionali processi di invio cartaceo. Tale approccio contribuisce non solo a migliorare l'esperienza dell'utente, ma anche a ottimizzare i costi operativi legati alla produzione, stampa e distribuzione fisica dei documenti, in linea con le strategie di digitalizzazione adottate dalle grandi organizzazioni.

Il perimetro del sistema comprende tutte le funzionalità necessarie a supportare il ciclo di vita dell'utenza e della consultazione documentale, includendo i processi di registrazione, autenticazione, gestione delle credenziali,

accesso ai contenuti, download dei documenti e gestione delle notifiche. Sono invece esplicitamente esclusi dal perimetro del sistema i processi di produzione originaria della documentazione e le funzionalità di gestione dei rapporti di investimento, che rimangono in carico a sistemi già presenti nell'ecosistema applicativo. Questa separazione consente di mantenere una chiara distinzione delle responsabilità tra i diversi sistemi e di limitare l'impatto dell'introduzione dell'Archivio Digitale sull'architettura complessiva.

2.2 Perimetro di utenza e regole di eleggibilità

Il sistema Archivio Digitale è destinato a una platea di utenti ben definita: i clienti titolari di servizi di investimento per i quali è prevista la produzione periodica di comunicazioni e documentazione soggetta a obblighi di conservazione e consultazione. L'abilitazione al servizio non è indiscriminata, ma avviene secondo un insieme di regole di eleggibilità che tengono conto dello stato del cliente rispetto ai canali digitali esistenti e alle modalità di ricezione della documentazione.

Il sistema distingue innanzitutto tra clienti già attivi e nuovi clienti. Nel caso di clienti già titolari di servizi di investimento, l'Archivio Digitale viene attivato automaticamente qualora il cliente non disponga di un canale digitale ordinario attivo e non abbia esercitato l'opzione di ricezione cartacea della documentazione. In questo scenario, il sistema svolge un ruolo fondamentale nel garantire la continuità della fruizione documentale, evitando che il cliente rimanga privo di un canale di accesso ai documenti.

Per i nuovi clienti, il processo di eleggibilità prevede una verifica preliminare della disponibilità alla sottoscrizione di un canale digitale ordinario. Qualora tale canale non venga attivato o venga esplicitamente rifiutato, l'Archivio Digitale rappresenta il canale di consultazione predefinito per la documentazione, in coerenza con l'impostazione "digital-by-default" introdotta dalla Direttiva (UE) 2021/338 (MiFID II "Quick Fix")[1], che prevede la trasmissione in formato elettronico quale modalità ordinaria di comunicazione. Resta in ogni caso facoltà del cliente esercitare l'opzione di ricezione cartacea della documentazione (opt-in). Questa logica consente di assicurare che, sin dall'inizio del rapporto, il cliente disponga di un canale valido per l'accesso alla documentazione, nel rispetto degli obblighi normativi e delle scelte individuali.

Il sistema è inoltre progettato per gestire scenari più complessi, come i casi di cointestazione dei rapporti, nei quali più soggetti sono legittimati alla con-

sultazione della documentazione. In tali situazioni, l'Archivio Digitale deve garantire che ciascun soggetto abbia accesso ai documenti di propria competenza, tenendo conto delle preferenze individuali e delle scelte effettuate in merito ai canali di comunicazione. La gestione di queste configurazioni è necessaria in un contesto reale, dove la variabilità delle situazioni contrattuali è elevata.

Le regole di eleggibilità non sono statiche, ma evolvono nel tempo in funzione delle scelte del cliente e degli eventi che interessano il rapporto, come l'attivazione o la chiusura di un canale digitale ordinario, la modifica delle preferenze di ricezione della documentazione o la cessazione dei rapporti di investimento. Il sistema deve quindi essere in grado di reagire a tali variazioni, mantenendo la coerenza tra lo stato del cliente e il canale utilizzato per la comunicazione documentale.

2.3 Requisiti funzionali

I requisiti funzionali del sistema Archivio Digitale descrivono in modo puntuale le capacità operative che il portale deve mettere a disposizione dell'utente finale per garantire l'accesso, la fruizione e la gestione della documentazione in formato digitale. Tali requisiti sono stati derivati dall'analisi dei flussi utente e delle funzionalità attese, con particolare attenzione alle esigenze operative tipiche di un contesto aziendale in cui la consultazione documentale deve avvenire in modo sicuro e tracciabile.

In primo luogo, il sistema deve consentire al cliente di attivare autonomamente la propria utenza tramite un processo di registrazione self-service, riducendo la necessità di interventi manuali e rendendo l'abilitazione al servizio scalabile anche su volumi elevati di utenti. La registrazione non si limita alla sola creazione di credenziali, ma comprende una sequenza strutturata di verifiche, finalizzate a garantire che l'utenza venga associata in maniera affidabile a un soggetto legittimato. In questo senso, il sistema deve prevedere la raccolta di informazioni identificative e la validazione di specifici recapiti. Questi ultimi risultano fondamentali sia per la fase di autenticazione sia per le notifiche legate alla disponibilità di nuovi documenti.

A registrazione completata, l'Archivio Digitale deve supportare un processo di accesso autenticato basato su credenziali e su meccanismi di sicurezza aggiuntivi. In particolare, l'autenticazione deve includere un secondo fattore, così da ridurre il rischio di accessi non autorizzati anche nel caso in cui una password venga compromessa. La gestione dell'accesso non può essere interpretata come una semplice validazione "username/password", ma deve comprendere la gestione degli scenari tipici di errore e di rischio, come ten-

tativi ripetuti di accesso, necessità di verifiche aggiuntive e flussi dedicati al primo accesso, che includono il cambio obbligatorio della password iniziale. In parallelo, il sistema deve essere in grado di gestire eventuali step informativi e conferme necessarie prima di consentire l'utilizzo completo del servizio, garantendo così sia la correttezza del processo sia la trasparenza verso l'utente.

Una volta completato l'accesso, il requisito funzionale centrale dell'Archivio Digitale è la consultazione della documentazione messa a disposizione del cliente. Questo significa che il portale deve essere in grado di presentare i documenti in maniera ordinata e comprensibile, attraverso un'organizzazione che consenta all'utente di orientarsi rapidamente tra categorie e tipologie differenti (ad esempio distinguendo tra diverse macrofamiglie documentali). La consultazione deve inoltre essere supportata da funzionalità di filtraggio, così da permettere all'utente di individuare rapidamente i contenuti rilevanti senza dover scorrere liste estese, e da funzioni di ordinamento che migliorano l'usabilità soprattutto nei casi in cui il volume documentale sia elevato.

Accanto alla consultazione, il sistema deve consentire il download dei documenti, sia in modalità singola sia in modalità multipla. Questo requisito è particolarmente importante perché risponde a un bisogno concreto dell'utente: la possibilità di archiviare localmente la documentazione, condividerla o utilizzarla per scopi amministrativi. La modalità di download multiplo consente di ottimizzare l'esperienza d'uso quando l'utente deve recuperare più documenti in un'unica sessione, riducendo la ripetitività delle operazioni e migliorando l'efficienza del processo. È inoltre necessario che il sistema gestisca correttamente gli scenari in cui alcuni documenti non siano disponibili o il recupero non vada a buon fine, fornendo messaggi chiari e in linea con lo stato dell'operazione.

Un ulteriore requisito funzionale riguarda la capacità del portale di notificare l'utente quando è disponibile nuova documentazione. In un sistema documentale, infatti, la consultazione non avviene necessariamente in modo continuo: spesso l'utente accede solo quando ha una necessità specifica. L'introduzione di notifiche consente di ridurre questo gap informativo e migliora l'efficacia complessiva del servizio, rendendo il cliente consapevole della presenza di nuovi documenti senza dover controllare manualmente il portale.

Nel complesso, le funzionalità del sistema sono progettate per essere fruibili tramite un'interfaccia web che accompagna l'utente nei diversi passaggi, limitando la probabilità di errori e migliorando la comprensione del processo. In una piattaforma destinata a un'utenza eterogenea, la gestione coerente degli stati applicativi (successo, errore, casi particolari) e la chiarezza delle comunicazioni rappresentano un requisito fondamentale, poiché contribuiscono a rendere l'interazione trasparente e affidabile.

2.4 Flusso di processo dell'Archivio Digitale

Il funzionamento complessivo del sistema Archivio Digitale può essere descritto attraverso un flusso di processo che copre l'intero ciclo di interazione dell'utente con il portale, dall'accesso iniziale fino alla consultazione dei contenuti. Tale flusso è stato progettato per essere progressivo e guidato, con l'obiettivo di accompagnare l'utente nelle diverse fasi e garantire la correttezza delle operazioni eseguite.

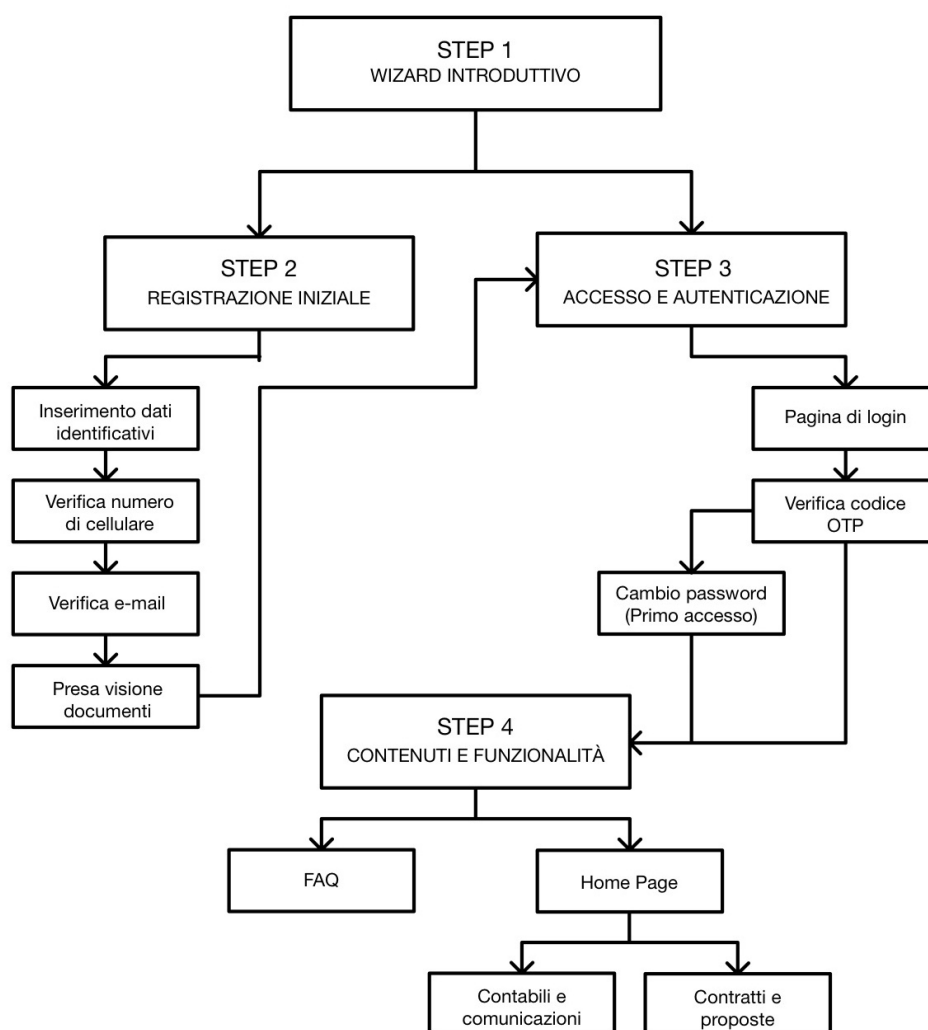


Figura 2.1: Mappa di navigazione dell'Archivio Digitale

La Figura 2.1 fornisce una visione d'insieme delle principali schermate del portale e delle transizioni tra di esse. Essa consente di comprendere come le diverse funzionalità siano integrate in un flusso coerente, evidenziando i punti di ingresso, i passaggi obbligatori e le possibili diramazioni del percorso utente.

2.4.1 Wizard introduttivo

Il wizard introduttivo rappresenta il punto di ingresso al sistema e svolge una funzione di orientamento per l'utente. La schermata iniziale presenta una breve descrizione del servizio e offre due opzioni principali: l'accesso per utenti già registrati e l'avvio della procedura di registrazione per nuovi utenti.



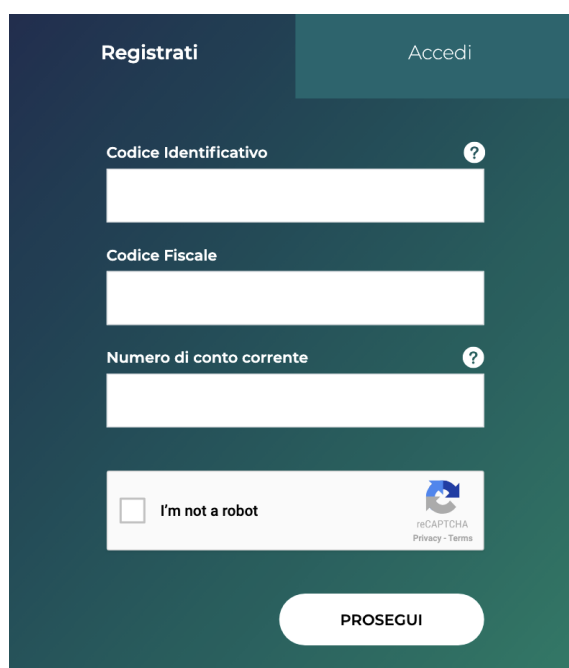
Figura 2.2: Wizard introduttivo

La Figura 2.2 mostra la schermata iniziale del portale, progettata per essere semplice e immediata. Il wizard consente inoltre di memorizzare la preferenza dell'utente relativa alla visualizzazione della schermata introduttiva nei

successivi accessi, migliorando l'esperienza d'uso e riducendo la ripetitività delle interazioni.

2.4.2 Registrazione iniziale

La registrazione iniziale consente al cliente di attivare la propria utenza di Archivio Digitale in modalità autonoma. Il processo prevede l'inserimento dei dati identificativi e l'esecuzione di controlli di sicurezza volti a prevenire accessi fraudolenti, inclusa la validazione tramite CAPTCHA.



The image shows a registration form with a dark teal background. At the top, there are two tabs: 'Registrati' (selected) and 'Accedi'. Below the tabs are three input fields: 'Codice Identificativo' with a question mark icon, 'Codice Fiscale', and 'Numero di conto corrente' with a question mark icon. At the bottom of the form is a CAPTCHA section with a checkbox labeled 'I'm not a robot' and the reCAPTCHA logo. A white 'PROSEGUI' button is centered at the bottom of the form.

Figura 2.3: Pagina di registrazione

Una volta completata la fase di inserimento dei dati, il sistema procede con la verifica dei recapiti del cliente. In particolare, viene richiesto di certificare il numero di telefono cellulare tramite l'inserimento di un codice OTP inviato via SMS e di validare l'indirizzo di posta elettronica tramite un codice OTP inviato via e-mail.

VERIFICA NUMERO DI CELLULARE

Trovi qui sotto il numero di cellulare associato al tuo nominativo.

Se il numero è corretto, clicca su **Conferma**.
Se il numero non è corretto, recati in filiale per aggiornarlo.

Numero di telefono
+39 1234567890

CONFERMA

VERIFICA NUMERO DI CELLULARE

Trovi qui sotto il numero di cellulare associato al tuo nominativo.

Se il numero è corretto, clicca su **Conferma**.
Se il numero non è corretto, recati in filiale per aggiornarlo.

Numero di telefono
+39 1234567890

Inserisci il codice a X cifre ricevuto via SMS per proseguire.

Inserisci codice

Non hai ricevuto il codice? [Richiedine uno nuovo](#)

CONFERMA

Figura 2.4: Verifica del numero di telefono

VERIFICA INDIRIZZO MAIL

Inserisci nel campo sottostante un indirizzo email valido.

Una volta cliccato su **prosegui** riceverai una mail con le istruzioni per completare la verifica.

Indirizzo mail

Inserisci indirizzo mail

PROSEGUI

VERIFICA INDIRIZZO MAIL

Ti abbiamo inviato una mail contenente un codice

Abbiamo inviato una mail all'indirizzo: mario.rossi@gmail.com ([Modifica indirizzo mail](#))

Inserisci il codice a X cifre ricevuto via mail per proseguire.

Inserisci codice

Non hai ricevuto il codice? [Richiedine uno nuovo](#)

PROSEGUI

Figura 2.5: Verifica dell'indirizzo e-mail

Questi passaggi garantiscono che i recapiti associati all'utenza siano effettivamente riconducibili al cliente e possano essere utilizzati in modo affidabile per le successive comunicazioni e notifiche.

La registrazione continua con la presa visione e l'accettazione dei termini e delle condizioni del servizio.

Termini e condizioni del servizio

Per poter procedere ti chiediamo di leggere e scaricare il documento "Termini e condizioni del servizio Archivio Digitale" e confermarne la ricezione.

[Termini e condizioni del servizio Archivio Digitale](#)

[Scarica il documento](#)

Dichiaro di aver letto e scaricato copia del documento "Termini e condizioni del servizio Archivio Digitale". ¹

CONFERMA

Figura 2.6: Termini e condizioni del servizio

La procedura di registrazione si conclude con la generazione della password temporanea di primo accesso. Quest'ultima viene inviata per metà via SMS e per metà via mail.



Figura 2.7: Esito positivo della registrazione

2.4.3 Accesso e autenticazione

L'accesso al sistema avviene tramite l'inserimento delle credenziali associate all'utenza. Il processo di autenticazione è progettato secondo criteri di sicurezza progressivi, che prevedono controlli aggiuntivi in caso di tentativi di accesso errati e l'eventuale blocco dell'utenza in situazioni di rischio.

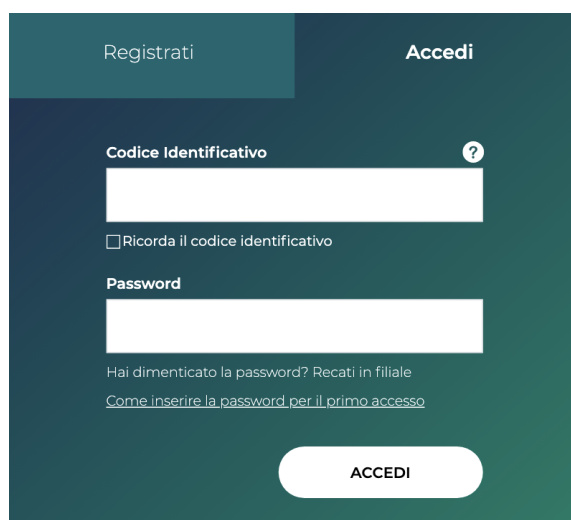


Figura 2.8: Pagina di login

Una volta validata la password, l'accesso viene completato tramite un secondo fattore di autenticazione basato su OTP inviato via SMS.

Conferma la tua identità

Inserisci il codice a X cifre ricevuto via SMS per proseguire.

Non hai ricevuto il codice? [Richiedine uno nuovo](#)

PROSEGUI

Figura 2.9: Verifica OTP

Al primo accesso, l'utente è obbligato a modificare la password temporanea ricevuta in fase di registrazione, garantendo così un livello di sicurezza adeguato fin dal primo utilizzo.

Imposta la tua nuova password

Essendo il tuo primo accesso, ti invitiamo a impostare una nuova password.

Inserisci nuova password

Conferma nuova password

REQUISITI:

- minimo 8 e massimo 30 caratteri
- non più di 3 caratteri uguali consecutivi
- almeno una lettera maiuscola, minuscola e un numero
- usare solo i seguenti caratteri speciali: _ - + £ \$ % &

PROSEGUI

Figura 2.10: Cambio password al primo accesso

2.4.4 Contenuti e funzionalità

Completato il processo di autenticazione, l'utente accede all'area di consultazione dell'Archivio Digitale. Il portale presenta una pagina introduttiva che illustra le principali funzionalità disponibili e consente di accedere alle macrocategorie documentali.

Benvenuto nel tuo archivio digitale

Accedi in modo semplice ai documenti relativi ai tuoi rapporti finanziari.
Qui puoi consultare comunicazioni, documentazione informativa, contratti e proposte in formato digitale.

CONTABILI E COMUNICAZIONI

In questa sezione trovi le comunicazioni e la documentazione contabile relativa ai tuoi rapporti finanziari, come estratti deposito titoli, rendicontazioni periodiche e informative previste dalla normativa tempo per tempo in vigore.

[VISUALIZZA →](#)

CONTRATTI E PROPOSTE

Qui sono disponibili i contratti sottoscritti e le proposte relative ai prodotti e ai servizi finanziari, con la documentazione associata e le principali condizioni.

[VISUALIZZA →](#)

Spunta la casella per non mostrare più questa pagina introduttiva e visualizzare direttamente il tuo Archivio Digitale

Figura 2.11: Pagina introduttiva del portale

La home page consente la visualizzazione ordinata dei documenti disponibili, offrendo funzionalità di ricerca, filtraggio e ordinamento. I documenti possono essere scaricati singolarmente o in modalità multipla in formato ZIP.

ARCHIVIO DIGITALEMODIFICA DIMENSIONE TESTI: A A | ESCI DAL SITO

CONTABILI E COMUNICAZIONI

CONTRATTI E PROPOSTE

Categoria

Avviata da

Stato

Periodo

Documenti trovati: 3

<input type="checkbox"/>	Pratica	Categoria	Avviata da	Stato	Data
<input checked="" type="checkbox"/>	Lorem ipsum dolor sit amet	Lorem ipsum dolor	Lorem ipsum dolor	Lorem ipsum	18/05/2024
<input type="checkbox"/>	Lorem ipsum dolor sit amet	Lorem ipsum dolor	Lorem ipsum dolor	Lorem ipsum	18/05/2024
<input type="checkbox"/>	Lorem ipsum dolor sit amet	Lorem ipsum dolor	Lorem ipsum dolor	Lorem ipsum	18/05/2024

Figura 2.12: Home page: contratti e proposte

Il sistema fornisce messaggi informativi in caso di assenza di risultati e notifica l'utente della disponibilità di nuovi documenti, contribuendo a una fruizione efficace e trasparente del servizio.

2.5 Requisiti non funzionali

Oltre ai requisiti funzionali, il sistema Archivio Digitale deve soddisfare una serie di requisiti non funzionali che risultano determinanti per garantire l'affidabilità e l'efficacia della soluzione in un contesto aziendale. Tali aspetti riguardano elementi trasversali al funzionamento del sistema e influenzano in modo significativo le scelte architettoniche e progettuali.

In primo luogo, il sistema deve garantire elevati livelli di sicurezza, sia per quanto riguarda l'accesso degli utenti sia per la protezione dei dati trattati. Questo include la gestione sicura delle credenziali, l'utilizzo di meccanismi di autenticazione a più fattori e la protezione delle comunicazioni tra i diversi componenti del sistema. La sicurezza assume un ruolo centrale anche in relazione alla tracciabilità delle operazioni, rendendo necessario il monitoraggio degli accessi e delle azioni compiute dagli utenti, al fine di supportare eventuali attività di audit.

Dal punto di vista delle performance, l'Archivio Digitale deve essere in grado di rispondere in tempi adeguati alle richieste degli utenti, anche in presenza di carichi elevati. Questo requisito è particolarmente rilevante considerando la possibilità di attivazioni massive del servizio e l'accesso contemporaneo di un numero significativo di utenti. Il sistema deve quindi essere progettato per garantire tempi di risposta stabili e prevedibili, evitando degradi percepibili dell'esperienza utente.

La scalabilità rappresenta un ulteriore requisito non funzionale di primaria importanza. Il sistema deve poter crescere nel tempo, sia in termini di numero di utenti sia di volume di documenti gestiti, senza richiedere interventi strutturali invasivi. Questo implica la necessità di adottare soluzioni architettoniche che supportino l'aumento del carico in modo progressivo e controllato.

Infine, il sistema deve garantire adeguati livelli di affidabilità e disponibilità, assicurando la continuità del servizio anche in presenza di guasti parziali o temporanei. In un contesto in cui la documentazione rappresenta un elemento essenziale del rapporto con il cliente, eventuali indisponibilità del sistema devono essere limitate e gestite in modo trasparente.

Questi aspetti costituiranno il riferimento per la valutazione sperimentale descritta nel Capitolo 6.

Capitolo 3

Architettura del sistema

3.1 Visione architetturale complessiva

Il presente capitolo descrive l'architettura del sistema Archivio Digitale, analizzandone l'evoluzione dallo stato attuale (AS-IS) alla soluzione proposta (TO-BE). L'obiettivo è fornire una rappresentazione strutturata dell'ecosistema applicativo, mettendo in evidenza le scelte progettuali adottate e il loro impatto sul funzionamento complessivo del sistema.

Considerata la complessità del contesto applicativo e la presenza di numerosi sistemi interconnessi, si è scelto di adottare il C4 Model come notazione di riferimento per la modellazione architetturale [2]. Il C4 Model, proposto da Simon Brown, si basa su una rappresentazione gerarchica dell'architettura software articolata su differenti livelli di astrazione, consentendo una comprensione progressiva del sistema e garantendo coerenza tra le diverse viste architettureali.

In particolare, l'analisi architetturale si concentra sui primi tre livelli del C4 Model:

- **System Context**, per descrivere il sistema nel suo contesto operativo e le interazioni con attori e sistemi esterni;
- **Container View**, per illustrare la scomposizione in macrocomponenti applicativi e le relative responsabilità;
- **Component View**, per analizzare la struttura interna di un servizio rappresentativo.

Il livello di Deployment non viene affrontato in questo capitolo, in quanto sarà oggetto di una trattazione separata.

3.2 Architettura AS-IS

3.2.1 Descrizione dello scenario attuale

Nello scenario attuale, la consultazione della documentazione da parte del cliente avviene attraverso due modalità principali: l'accesso tramite un canale digitale ordinario oppure la ricezione della documentazione in formato cartaceo. L'accesso digitale è reso disponibile mediante una piattaforma di Internet Banking, che consente, tra le tante funzionalità, anche la visualizzazione e il download dei documenti previa sottoscrizione di un contratto specifico, nella maggior parte dei casi associato a costi o oneri a carico del cliente. In alternativa, la documentazione viene trasmessa in formato cartaceo.

Dal punto di vista architettonico, il sistema di Home Banking agisce come intermediario tra il cliente e il sistema di gestione documentale, occupandosi sia della presentazione delle informazioni sia della gestione delle richieste di recupero dei documenti. Il sistema documentale, a sua volta, è responsabile della conservazione e dell'esposizione dei documenti attraverso interfacce dedicate.

Questa configurazione riflette un'impostazione tradizionale in cui l'accesso digitale alla documentazione è subordinato all'adesione a un servizio più ampio e non rappresenta un canale autonomo. Il cartaceo, pur non essendo più coerente con l'evoluzione normativa orientata alla digitalizzazione, continua a costituire una modalità operativa rilevante, con impatti significativi in termini di costi e tempi di distribuzione.

3.2.2 Limiti dell'architettura AS-IS

Pur garantendo la disponibilità della documentazione, l'architettura AS-IS presenta alcuni limiti strutturali rispetto agli obiettivi di digitalizzazione introdotti dall'evoluzione normativa e dalle nuove esigenze di accessibilità del servizio.

In primo luogo, l'accesso digitale alla documentazione è subordinato alla sottoscrizione di un contratto dedicato, spesso oneroso per il cliente. Questo requisito limita la platea di utenti che possono beneficiare della consultazione online, escludendo quei clienti che, pur avendo diritto alla documentazione digitale, non dispongono di un canale attivo o non intendono attivarlo.

In secondo luogo, la gestione del cartaceo comporta costi operativi elevati, tempi di recapito più lunghi e una minore flessibilità nella fruizione dei contenuti. In un contesto normativo orientato al principio del "digital-by-default", il mantenimento del cartaceo come modalità operativa estesa risulta sempre meno sostenibile sia dal punto di vista economico sia da quello organizzativo.

Dal punto di vista architetturale, la fruizione documentale risulta strettamente vincolata al canale di Home Banking, che rappresenta l'unico punto di accesso digitale disponibile.

3.3 Architettura TO-BE

3.3.1 Obiettivi dell'evoluzione architetturale

L'architettura TO-BE nasce con l'obiettivo di superare i limiti evidenziati nello scenario AS-IS, introducendo un nuovo sistema dedicato alla consultazione digitale della documentazione: l'Archivio Digitale. Questo sistema si propone come canale alternativo gratuito, accessibile direttamente dal cliente.

Dal punto di vista architetturale, l'introduzione dell'Archivio Digitale consente di affiancare al canale ordinario un punto di accesso dedicato alla documentazione, riducendo la dipendenza esclusiva dall'Internet Banking e aumentando la flessibilità complessiva dell'ecosistema applicativo. Il sistema risulta così ottimizzato specificamente per le esigenze di fruizione documentale.

3.3.2 Integrazione con i sistemi esistenti

L'Archivio Digitale non sostituisce i sistemi esistenti, ma si integra con essi, preservando le responsabilità già consolidate. In particolare, il sistema continua a fare affidamento su un sistema documentale esterno per il recupero e il download dei documenti, evitando duplicazioni di dati e garantendo la coerenza delle informazioni.

L'Archivio introduce inoltre servizi applicativi dedicati che fungono da punto di accesso specifico per il canale, orchestrando le interazioni con i sistemi sottostanti. Dal punto di vista dell'utente finale, il sistema si presenta come un canale autonomo, pur rimanendo pienamente integrato nell'ecosistema applicativo esistente.

3.4 Modellazione architetturale con C4 Model

La modellazione architetturale del sistema è stata condotta adottando il C4 Model come notazione di riferimento [2]. Il modello, proposto da Simon

Brown, consente di descrivere l'architettura software attraverso una rappresentazione gerarchica articolata su più livelli di astrazione (System Context, Container, Component e Deployment), consentendo di mantenere coerenza tra le diverse viste e di supportare una comprensione progressiva del sistema.

L'adozione del C4 Model risponde all'esigenza di rappresentare in modo strutturato un ecosistema applicativo complesso, caratterizzato dalla presenza di più sistemi interconnessi e da relazioni funzionali articolate. Tale approccio è coerente con quanto previsto dallo standard ISO/IEC/IEEE 42010:2011 [3], che sottolinea l'importanza di descrivere un'architettura attraverso viste multiple, ciascuna orientata a specifiche preoccupazioni degli stakeholder.

Nel presente capitolo l'analisi si concentra sui primi tre livelli del modello, rimandando la trattazione degli aspetti deployment a una sezione successiva, al fine di mantenere il focus sugli aspetti logici e strutturali dell'architettura.

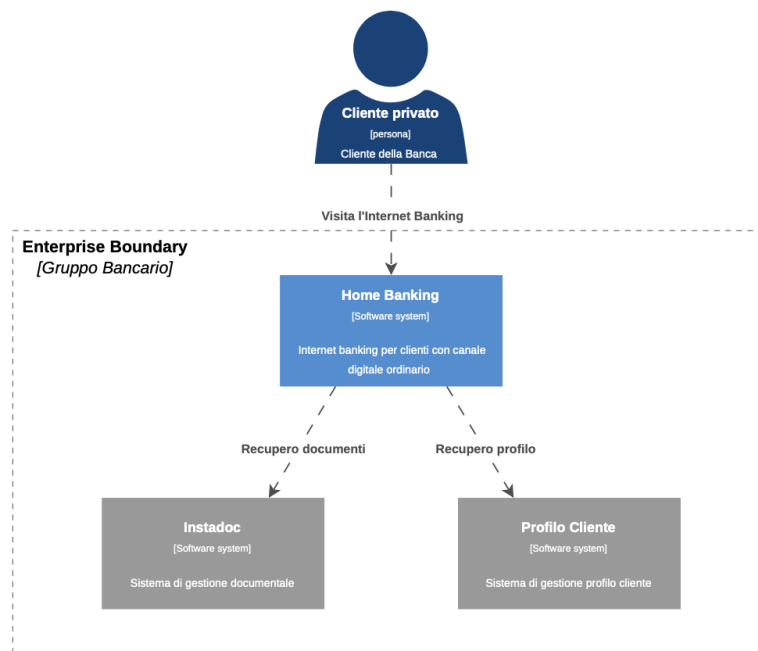
3.4.1 System Context View

Il livello di System Context fornisce una visione d'insieme del sistema, evidenziandone i confini, gli attori esterni e le principali interazioni con gli altri sistemi dell'ecosistema applicativo. Questa vista consente di collocare l'Archivio Digitale all'interno del contesto organizzativo e tecnologico in cui opera, senza entrare nel dettaglio delle tecnologie implementative o delle strutture interne.

Nel caso in esame, sono stati realizzati due diagrammi distinti: uno per lo scenario AS-IS e uno per lo scenario TO-BE. Il confronto tra i due diagrammi consente di evidenziare in modo immediato l'evoluzione architetturale introdotta dal progetto, mettendo in luce il ruolo centrale assunto dall'Archivio Digitale come nuovo punto di accesso alla documentazione.

Scenario AS-IS

Nel diagramma relativo allo scenario attuale (Figura 3.1), il cliente privato interagisce esclusivamente con il sistema di Home Banking, che rappresenta l'unico punto di accesso digitale alla documentazione. Il sistema di Home Banking media le richieste dell'utente e si integra con il sistema documentale (Instadoc) per il recupero dei documenti e con il sistema di gestione del profilo cliente (Profilo Cliente) per le informazioni anagrafiche.



System Context View: Archivio Digitale
Il System Context Diagram per il sistema Archivio Digitale

Figura 3.1: System Context View dello scenario AS-IS

In questo assetto, la consultazione documentale risulta interamente vincolata al canale di Internet Banking. Il diagramma evidenzia come il sistema documentale non sia direttamente esposto al cliente finale, ma venga sempre raggiunto attraverso il canale digitale ordinario, configurando una dipendenza esclusiva da tale piattaforma per l'accesso online ai documenti.

Scenario TO-BE

Nel diagramma relativo allo scenario evolutivo (Figura 3.2), viene introdotto un nuovo sistema: Archivio Digitale, descritto come portale documentale dedicato ai clienti senza canale digitale ordinario. In questo scenario, il cliente può accedere direttamente al portale dell'Archivio Digitale, che interagisce a sua volta con il sistema documentale (Instadoc) e con il sistema di gestione del profilo cliente.

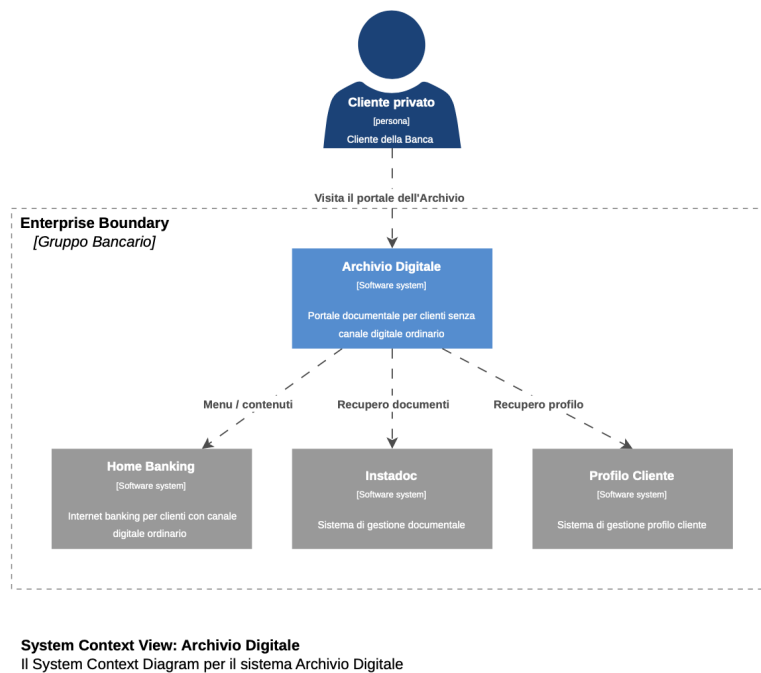


Figura 3.2: System Context View dello scenario TO-BE

Il sistema di Home Banking rimane invariato e continua a svolgere il proprio ruolo per i clienti che dispongono del canale digitale ordinario. L'introduzione dell'Archivio Digitale non modifica le responsabilità dei sistemi esistenti, ma affianca un nuovo punto di accesso dedicato alla consultazione documentale.

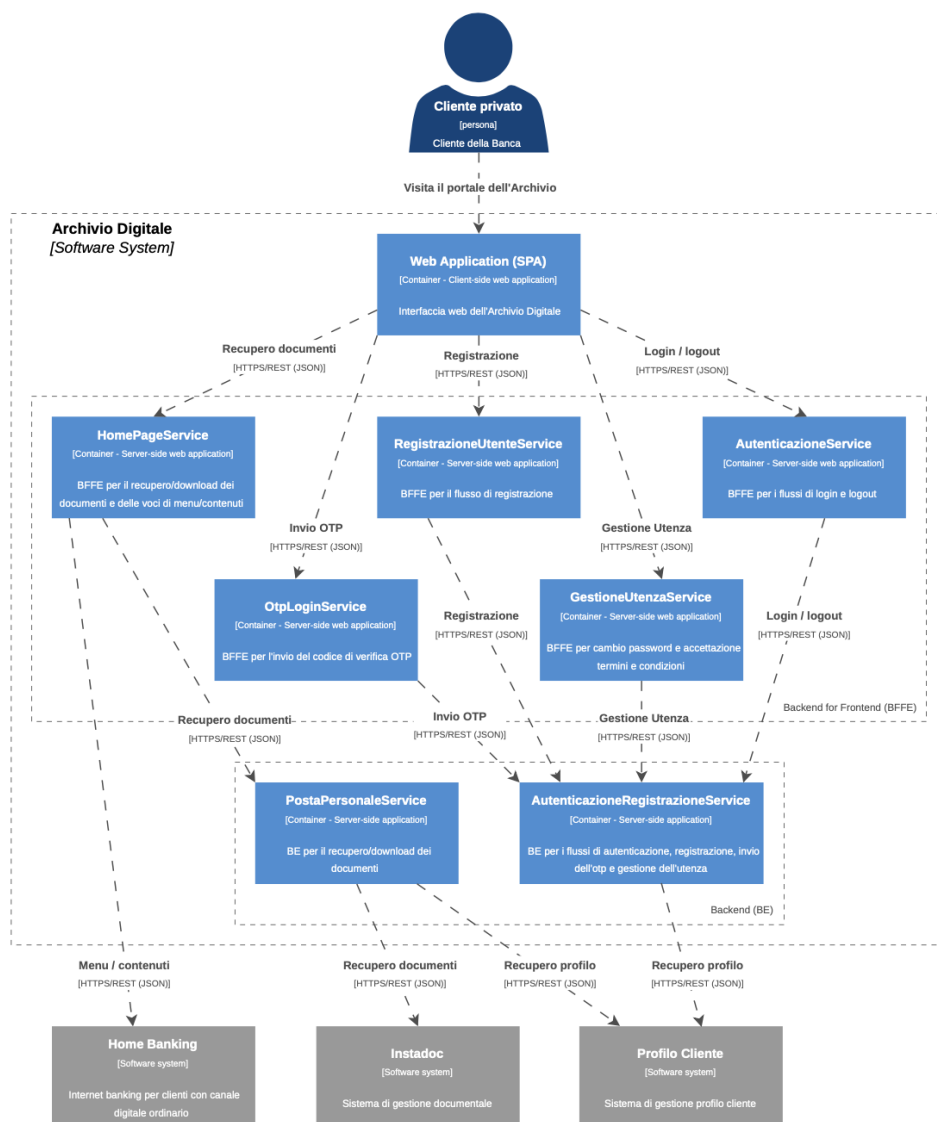
Il confronto tra i due diagrammi evidenzia quindi il cambiamento architetturale principale: la fruizione digitale della documentazione non è più vincolata a un unico canale, ma può avvenire attraverso un sistema dedicato, mantenendo inalterate le integrazioni con i sistemi documentali e anagrafici già presenti nell'ecosistema applicativo.

3.4.2 Container View dell'Archivio Digitale

Il Container View rappresenta il secondo livello di dettaglio del C4 Model e consente di descrivere la scomposizione del sistema in unità eseguibili autonome (container), evidenziandone responsabilità, interazioni e confini logici.

Nel contesto del progetto, il diagramma di Container View mostra come il sistema Archivio Digitale sia articolato in più macrocomponenti, ciascuno con un ruolo ben definito all'interno dell'architettura. In particolare, è possibile

individuare un frontend web, responsabile dell'interazione diretta con l'utente finale, uno strato di Backend for Frontend (BFFE), incaricato di gestire i flussi applicativi specifici del canale e uno strato di backend applicativo, che espone le funzionalità core e coordina le integrazioni con i sistemi esterni.



Container View: Archivio Digitale
Il Container Diagram per il sistema Archivio Digitale

Figura 3.3: Container View dell'Archivio Digitale

Tale articolazione consente di separare in modo esplicito le responsabilità di presentazione, orchestrazione dei flussi di canale e logica applicativa core, riducendo l'accoppiamento tra i livelli e favorendo manutenibilità ed evoluzione indipendente delle componenti.

Frontend Web

Il frontend dell'Archivio Digitale è realizzato come Single Page Application (SPA) ed è accessibile direttamente dal cliente tramite browser. Questo container si occupa esclusivamente della presentazione delle informazioni e della gestione dell'interazione con l'utente, delegando ogni logica applicativa ai livelli sottostanti.

La comunicazione con i livelli sottostanti avviene tramite API REST esposte su protocollo HTTPS, con scambio di payload in formato JSON. Tale scelta garantisce interoperabilità, standardizzazione delle interfacce e coerenza con le pratiche adottate nell'ecosistema applicativo aziendale.

Backend for Frontend (BFFE)

Lo strato di BFFE rappresenta un elemento chiave dell'architettura TO-BE. I servizi BFFE sono progettati per fornire al frontend un'interfaccia dedicata e ottimizzata per il canale, occupandosi della gestione dei flussi applicativi e dell'orchestrazione delle chiamate verso i servizi di backend.

Nel diagramma sono presenti diversi servizi BFFE, ciascuno dedicato a un ambito funzionale specifico, come la gestione dell'autenticazione, della registrazione dell'utente, dell'invio di codici OTP e del primo accesso. Questa specializzazione consente di mantenere i servizi focalizzati su responsabilità ben definite, migliorando la leggibilità e la manutenibilità del codice.

I BFFE comunicano sia con il frontend sia con i servizi di backend attraverso API REST su HTTPS, scambiando payload JSON. Le comunicazioni tra BFFE e backend avvengono all'interno del cluster applicativo, senza passare da gateway esterni, riducendo la latenza e semplificando la configurazione delle integrazioni.

Backend applicativo

Il backend applicativo espone le funzionalità core dell'Archivio Digitale ed è responsabile dell'integrazione con i sistemi esterni, in particolare con il sistema di gestione documentale. I servizi di backend implementano la logica applicativa principale e forniscono API REST utilizzate dai BFFE. Questa separazione consente di mantenere la logica applicativa indipendente dalle

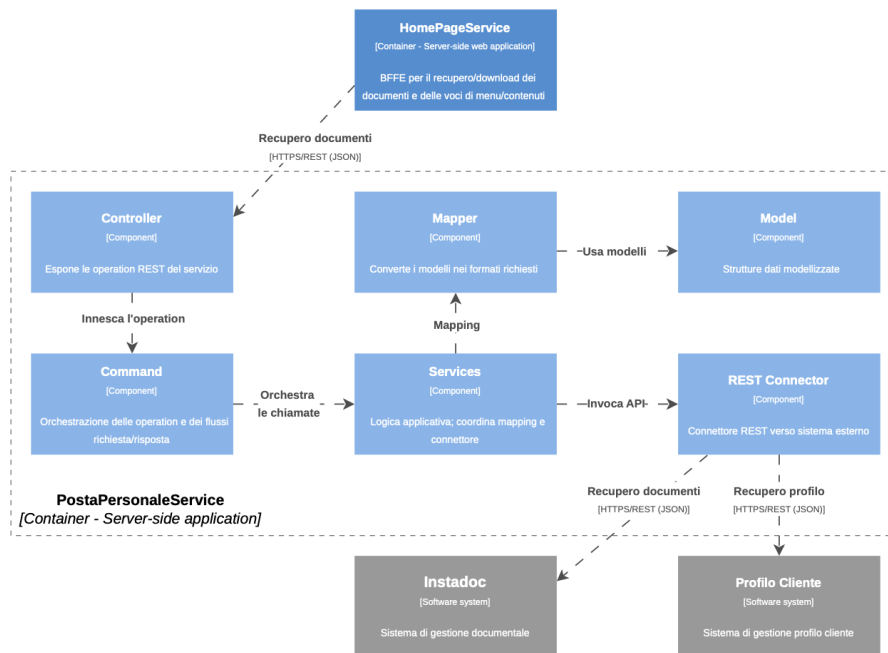
specificità del canale, favorendo potenziali riutilizzi dei servizi da parte di altri sistemi o canali futuri.

Anche in questo caso la comunicazione segue gli stessi standard HTTP-S/REST adottati negli altri livelli.

3.4.3 Component View del PostaPersonaleService

Il Component View rappresenta il terzo livello di dettaglio del C4 Model e consente di analizzare la struttura interna di un singolo container, evidenziando le componenti software che lo costituiscono e le relazioni tra esse.

Nel presente lavoro, tale livello di analisi è stato applicato al servizio PostaPersonaleService, scelto come caso rappresentativo dell'architettura dei servizi backend dell'Archivio Digitale. La selezione di questo servizio è motivata dal suo ruolo centrale nella gestione della consultazione e del download dei documenti, nonché dalla sua interazione diretta con il sistema documentale esterno.



Component View: Archivio Digitale
Il Component Diagram per il sistema Archivio Digitale

Figura 3.4: Component View del PostaPersonaleService

Struttura interna del servizio

Il diagramma di Component View evidenzia come il servizio sia organizzato secondo uno standard architetturale condiviso, adottato trasversalmente per tutti i servizi backend e BFFE del progetto. La struttura segue una chiara separazione delle responsabilità, articolandosi nei seguenti componenti principali:

- **Controller**, responsabile dell'esposizione delle API REST e della gestione delle richieste in ingresso;
- **Command**, incaricato dell'orchestrazione del flusso applicativo e del coordinamento delle operazioni necessarie al completamento della richiesta;
- **Service**, responsabile dell'implementazione della logica applicativa;
- **Mapper**, utilizzato per la trasformazione tra i modelli esposti dalle API e le strutture dati interne;
- **Model**, rappresenta le entità e le strutture dati utilizzate dal servizio;
- **REST Connector**, responsabile della comunicazione con il sistema documentale esterno.

Questa organizzazione interna consente di isolare la logica di presentazione dall'orchestrazione dei flussi e dalla logica di dominio, riducendo le dipendenze tra livelli e migliorando la testabilità e l'evoluzione del servizio. Ogni componente svolge un ruolo ben definito e interagisce con gli altri attraverso interfacce esplicite, favorendo la leggibilità del codice e la manutenibilità complessiva.

Generalizzazione del modello

Sebbene il Component View sia stato realizzato per il solo PostaPersonale-Service, la struttura rappresentata è da intendersi come generalizzabile all'intero insieme dei servizi backend e BFFE dell'Archivio Digitale. Tutti i servizi seguono infatti lo stesso schema architetturale, basato sull'adozione di un framework proprietario, costruito su Spring Boot, che impone convenzioni precise in termini di organizzazione del codice e separazione dei livelli.

L'adozione di tale modello favorisce la standardizzazione delle soluzioni implementative, riduce la variabilità tra servizi e contribuisce a rendere l'architettura complessiva più uniforme e prevedibile. In contesti aziendali

complessi, questa uniformità rappresenta un fattore determinante per contenere la complessità tecnica e facilitare l'inserimento di nuovi sviluppatori nel progetto.

3.5 Considerazioni architettureali

L'architettura TO-BE dell'Archivio Digitale introduce un cambiamento significativo rispetto allo scenario AS-IS, sia dal punto di vista funzionale sia dal punto di vista strutturale. L'introduzione di un sistema dedicato alla consultazione documentale consente di superare i limiti del modello precedente, ampliando l'accessibilità del servizio e riducendo la dipendenza dai canali digitali ordinari.

La scomposizione dell'architettura nei livelli di frontend, Backend for Frontend e backend applicativo consente di separare in modo esplicito le responsabilità di presentazione, orchestrazione e logica di dominio. Tale impostazione favorisce la modularità del sistema, ne facilita l'evoluzione nel tempo e consente di contenere la complessità delle integrazioni con i sistemi esterni.

La modellazione mediante C4 Model ha permesso di rappresentare in modo coerente l'evoluzione architettureale, evidenziando sia il posizionamento del sistema nel contesto organizzativo sia la struttura interna dei servizi. Le scelte progettuali descritte costituiscono la base tecnica per le attività di implementazione e valutazione sperimentale trattate nei capitoli successivi.

Capitolo 4

Tecnologie e framework di sviluppo

4.1 Contesto tecnologico

Il presente capitolo ha l'obiettivo di descrivere le principali tecnologie e gli strumenti adottati per la realizzazione del sistema Archivio Digitale, fornendo una visione dello stack tecnologico e delle scelte di sviluppo effettuate nel corso del progetto. In particolare, il capitolo si concentra sugli aspetti applicativi e sullo standard di sviluppo adottato, mettendo in relazione le tecnologie utilizzate con le esigenze architetturali e funzionali descritte nei capitoli precedenti.

Nel contesto di un sistema informativo complesso, come quello oggetto di questa tesi, le tecnologie incidono direttamente sulla struttura del software, sulla manutenibilità del codice e sulla capacità del sistema di evolvere nel tempo. Come evidenziato in letteratura, le decisioni tecnologiche influenzano in modo significativo le qualità architetturali del sistema, tra cui scalabilità, modificabilità e integrabilità [4]. Per questo motivo, la loro analisi non viene affrontata solo in termini descrittivi, ma come parte integrante delle scelte architetturali complessive.

È importante sottolineare che il perimetro del capitolo è volutamente limitato agli aspetti di sviluppo applicativo e integrazione. Le tematiche legate all'esecuzione del software, alla configurazione degli ambienti e al deployment verranno trattate separatamente in un capitolo successivo.

4.2 Stack tecnologico dell'Archivio Digitale

Lo stack tecnologico dell'Archivio Digitale riflette le esigenze di un sistema aziendale operante in ambito bancario, caratterizzato da requisiti stringenti in termini di sicurezza, affidabilità e integrazione con sistemi preesistenti. Le tecnologie adottate sono state selezionate per garantire un buon compromesso tra standardizzazione, flessibilità ed estendibilità, mantenendo coerenza con l'ecosistema applicativo già esistente.

Dal punto di vista architetturale, lo stack si colloca all'interno di un modello multi-tier, in cui responsabilità differenti vengono distribuite su livelli distinti secondo principi di separazione delle responsabilità e riduzione dell'accoppiamento [5]. In particolare, è possibile individuare tre macrolivelli principali:

- il frontend web, responsabile della presentazione e dell'interazione con l'utente;
- lo strato di Backend for Frontend (BFFE), dedicato all'orchestrazione dei flussi specifici del canale;
- il backend applicativo, che implementa la logica di dominio e gestisce le integrazioni con i sistemi esterni.

Nel progetto in esame, il frontend è stato realizzato come Single Page Application (SPA) mediante il framework Angular, mentre i servizi server-side (BFFE e backend applicativo) sono implementati come applicazioni Java basate su Spring Boot, all'interno delle quali si innesta un framework proprietario che standardizza struttura interna, convenzioni e modalità di integrazione.

L'adozione combinata di Angular e Spring Boot risponde a esigenze tipiche dei sistemi distribuiti moderni. Da un lato, il paradigma SPA consente di realizzare interfacce dinamiche e orientate alla composizione di componenti riutilizzabili, migliorando la fluidità dell'esperienza utente e riducendo la dipendenza da ricaricamenti completi della pagina. Dall'altro, Spring Boot fornisce un'infrastruttura applicativa consolidata per la realizzazione di servizi RESTful, integrando componenti per la gestione della configurazione, della sicurezza e della persistenza dei dati [6].

La comunicazione tra frontend e backend avviene secondo lo stile architetturale REST, su protocollo HTTPS, con scambio di rappresentazioni in formato JSON. Tale impostazione favorisce interoperabilità, disaccoppiamento tra client e server e evoluzione indipendente dei servizi, in linea con i principi definiti per le architetture web distribuite [7].

4.2.1 Frontend web

Il frontend dell'Archivio Digitale è realizzato come Single Page Application (SPA) accessibile tramite browser web e sviluppata mediante il framework Angular. Il paradigma SPA prevede che l'applicazione venga caricata inizialmente come una singola pagina HTML, mentre gli aggiornamenti dell'interfaccia avvengono dinamicamente lato client tramite manipolazione del DOM e richieste asincrone verso i servizi backend. Questo modello consente di ridurre i ricaricamenti completi della pagina e di offrire un'esperienza utente più fluida e reattiva, caratteristica tipica delle moderne applicazioni web dinamiche basate su interazioni asincrone lato client [8].

Dal punto di vista architetturale, l'adozione di una SPA favorisce una netta separazione tra livello di presentazione e logica di dominio. Angular fornisce un modello strutturato basato su componenti riutilizzabili, servizi client-side e meccanismi di dependency injection, consentendo di organizzare l'interfaccia secondo principi di modularità e separazione delle responsabilità. La gestione dello stato dell'interfaccia, della navigazione e della comunicazione HTTP viene incapsulata in elementi distinti, riducendo l'accoppiamento tra le diverse parti dell'applicazione.

Un elemento rilevante è il routing applicativo, che permette di gestire la navigazione tra viste differenti senza ricaricare l'intera pagina, facilitando l'implementazione di percorsi multi-step, come i flussi di registrazione, autenticazione e consultazione documentale descritti nel capitolo relativo ai requisiti. Tale impostazione risulta particolarmente adatta a sistemi che prevedono interazioni sequenziali e guidate.

Nel contesto dell'Archivio Digitale, il frontend agisce esclusivamente come client dei servizi esposti dal Backend for Frontend e non contiene logica di business. La validazione sostanziale dei dati, l'applicazione delle regole di dominio e l'accesso alle fonti informative rimangono responsabilità dei servizi server-side. Il frontend si occupa principalmente della presentazione delle informazioni, dell'orchestrazione delle chiamate verso i servizi backend e della gestione degli stati dell'interfaccia (caricamento, errore, conferma), garantendo consistenza nella navigazione e nell'esperienza utente.

4.2.2 Backend for Frontend (BFFE)

Lo strato di Backend for Frontend (BFFE) rappresenta un elemento centrale dell'architettura dell'Archivio Digitale. I servizi BFFE sono progettati per fungere da intermediari tra il frontend web e il backend applicativo, fornendo un insieme di API ottimizzate per le esigenze specifiche del canale.

Il pattern Backend for Frontend è stato introdotto nell'ambito delle architetture a microservizi per rispondere alla necessità di adattare le interfacce dei servizi core alle caratteristiche dei diversi client (web, mobile, ecc.), evitando che questi ultimi debbano interagire direttamente con servizi di dominio generici e potenzialmente complessi [9]. In questo modello, ciascun frontend può disporre di un backend dedicato che aggrega, orchestra e trasforma le informazioni provenienti dai servizi sottostanti.

Dal punto di vista implementativo, i servizi BFFE dell'Archivio Digitale sono sviluppati come applicazioni Java basate su Spring Boot e progettati per esporre API REST orientate al canale web. Il loro ruolo consiste nell'orchestrare le chiamate verso i servizi core di backend, aggregare le risposte provenienti da più fonti, applicare eventuali trasformazioni dei payload e centralizzare la gestione di errori e controlli di autorizzazione.

Nel progetto dell'Archivio Digitale, i servizi BFFE comunicano con il frontend e con il backend applicativo tramite API REST su HTTPS, utilizzando payload JSON. Le chiamate dal BFFE verso i servizi di backend avvengono all'interno del cluster applicativo, sfruttando la rete interna e i service endpoint, e non richiedono il passaggio attraverso gateway esterni. Questa scelta è coerente con l'obiettivo di ridurre latenza e complessità di instradamento per le comunicazioni interne tra componenti dell'architettura.

Dal punto di vista architetturale, l'introduzione di uno strato BFFE consente di:

- disaccoppiare il frontend dai servizi core di backend;
- ridurre la complessità delle interazioni lato client;
- centralizzare la gestione dei flussi applicativi legati al canale.

Ogni servizio BFFE è responsabile di un ambito funzionale ben definito (ad esempio autenticazione, registrazione utente, gestione OTP), favorendo la modularità del sistema e consentendo l'evoluzione indipendente delle componenti legate al canale di accesso.

4.2.3 Backend applicativo

Il backend applicativo costituisce il nucleo funzionale dell'Archivio Digitale ed è responsabile dell'implementazione della logica di dominio, dell'esposizione delle API core e dell'integrazione con i sistemi esterni coinvolti nei processi documentali. Dal punto di vista architetturale, esso si colloca in un modello a strati (layered architecture), nel quale le responsabilità vengono distribuite

secondo principi di separazione delle responsabilità e dipendenze orientate verso il basso [10].

I servizi di backend sono implementati come applicazioni Java basate su Spring Boot, che fornisce un'infrastruttura applicativa per lo sviluppo di servizi web RESTful, integrando meccanismi di dependency injection, gestione della configurazione, validazione degli input e integrazione con moduli di sicurezza. L'utilizzo di un framework basato su inversione del controllo (IoC) consente di ridurre l'accoppiamento tra componenti e di favorire testabilità ed estensibilità del codice.

Nel progetto considerato, Spring Boot rappresenta il fondamento tecnologico su cui si innesta il framework proprietario adottato dal team. Quest'ultimo definisce convenzioni strutturali uniformi (ad esempio la separazione tra Controller, Command, Service e Connector), mentre Spring Boot fornisce il runtime, la gestione del ciclo di vita dei componenti e i meccanismi di esposizione delle API secondo lo stile architetturale REST [7].

Per quanto riguarda l'integrazione, i servizi di backend adottano principalmente due modalità:

- connettori REST verso sistemi esterni, utilizzati per l'interazione con il sistema documentale;
- connettori basati su JPA (Java Persistence API) per l'accesso a basi dati relazionali interne.

L'utilizzo di JPA consente di astrarre il livello di persistenza attraverso un modello ORM (Object-Relational Mapping), separando la rappresentazione degli oggetti di dominio dalla struttura fisica delle basi dati [11]. L'isolamento della logica di integrazione all'interno dei connector mantiene il core applicativo indipendente dalle specifiche tecniche dei sistemi target, migliorando la manutenibilità e riducendo l'impatto di eventuali evoluzioni nelle interfacce o nelle modalità di accesso ai dati.

4.3 Framework proprietario per lo sviluppo dei servizi

Lo sviluppo dei servizi di backend e di Backend for Frontend dell'Archivio Digitale si basa su un framework architetturale proprietario aziendale sviluppato nell'ambito di un programma interno di evoluzione architetturale avviato nel 2020. Il framework è stato concepito con l'obiettivo di abilitare uno sviluppo standardizzato di servizi orientati all'omnicanalità, fornendo

linee guida, archetipi e componenti comuni riutilizzabili nei diversi progetti applicativi.

Uno degli elementi centrali del framework è l'utilizzo di archetipi Maven multi-modulo, che consentono la generazione automatica della struttura di un servizio (moduli Boot, Logic, Mapping, Model, Service, Service-Connector) secondo uno schema architetturale predefinito. Ogni modulo ha un ruolo specifico e contribuisce alla costruzione dell'ApplicationContext Spring attraverso configurazioni esplicite e component scanning strutturato.

Il framework formalizza inoltre la separazione dei layer applicativi. In particolare:

- il modulo Api-Controller gestisce esclusivamente l'esposizione delle API REST, la trasformazione dei modelli di input/output e la delega dell'esecuzione ai Command del logic layer;
- il modulo Mapping centralizza le logiche di conversione tra modelli appartenenti a layer differenti, garantendo disaccoppiamento tra rappresentazione API e modelli interni di dominio;
- i moduli Service e Service-Connector implementano rispettivamente la logica applicativa e le integrazioni verso sistemi esterni.

Dal punto di vista infrastrutturale e trasversale, il framework integra componenti comuni per la gestione centralizzata degli errori, il logging e il monitoraggio delle richieste, la gestione della sessione. Sono inoltre previsti connettori standardizzati (REST, JPA, MQ, Kafka), che consentono di uniformare le modalità di integrazione con sistemi esterni e legacy, riducendo la duplicazione di codice e favorendo la correzione centralizzata di eventuali anomalie.

L'adozione del framework garantisce quindi uniformità strutturale tra servizi, riusabilità dei componenti e gestione centralizzata degli aspetti trasversali.

4.4 Struttura standard dei servizi applicativi

Coerentemente con quanto descritto nel paragrafo precedente, i servizi di backend e di Backend for Frontend dell'Archivio Digitale adottano una struttura modulare e stratificata definita dal framework proprietario aziendale. Tale organizzazione non rappresenta una semplice convenzione stilistica, ma costituisce un vero e proprio modello architetturale applicato in modo uniforme a tutti i servizi.

Ogni servizio è articolato in componenti con responsabilità chiaramente delimitate, secondo un principio di separazione delle responsabilità (Separation of Concerns) [12]. La struttura distingue in modo esplicito il livello di esposizione delle API, il livello di orchestrazione applicativa, il livello di logica di dominio e il livello di integrazione verso sistemi esterni. Questa impostazione è riconducibile al paradigma dell'architettura a strati (Layered Architecture), in cui le dipendenze sono orientate dall'alto verso il basso e ciascun livello interagisce esclusivamente con quello immediatamente sottostante.

In particolare, la struttura dei servizi prevede la presenza dei seguenti componenti:

- Controller, responsabile dell'esposizione delle API REST e della ricezione delle richieste provenienti dai consumer;
- Command, incaricato di orchestrare il flusso applicativo associato a una specifica operazione;
- Service, che implementa la logica applicativa e coordina l'interazione tra i diversi componenti;
- Connector, utilizzato per l'accesso a sistemi esterni o a sorgenti dati persistenti;
- Model, utilizzato per la rappresentazione dei dati;
- Mapper, impiegato per la trasformazione tra modelli appartenenti a layer differenti.

Questa organizzazione favorisce la manutenibilità del codice, limita l'accoppiamento tra componenti e consente di evolvere in modo controllato i singoli livelli dell'applicazione senza impatti trasversali non desiderati.

4.4.1 Controller e gestione delle API

Il componente Controller rappresenta il punto di ingresso del servizio e implementa le operazioni REST definite nella specifica OpenAPI. In linea con le convenzioni del framework, il controller non contiene logica di business né logica di orchestrazione: il suo compito è limitato alla validazione dell'input, alla gestione delle annotazioni di mapping HTTP e alla delega dell'esecuzione al componente Command.

Questa scelta progettuale consente di mantenere il controller leggero e facilmente testabile, evitando che la complessità applicativa si concentri nel livello di esposizione delle API. Inoltre, l'allineamento tra controller e specifica OpenAPI favorisce la coerenza tra documentazione e implementazione.

4.4.2 Command e orchestrazione dei flussi

Il componente Command svolge un ruolo centrale nell'architettura dei servizi, fungendo da orchestratore del flusso applicativo associato a una specifica operazione. A partire dalla richiesta ricevuta dal controller, il command coordina l'invocazione dei servizi applicativi necessari e gestisce la costruzione della risposta finale.

Questa separazione consente di isolare la logica di orchestrazione dalla logica di dominio vera e propria, migliorando la leggibilità del codice e facilitando l'introduzione di nuove operazioni o la modifica di flussi esistenti. Inoltre, il command rappresenta un punto naturale per la gestione degli errori applicativi e delle condizioni di esito negativo.

4.4.3 Service, Model e Mapper

Il componente Service incapsula la logica applicativa del servizio ed è responsabile dell'esecuzione delle operazioni di dominio. Esso coordina l'accesso ai dati, l'invocazione dei connector e l'applicazione delle regole di business, mantenendo una visione coerente del contesto applicativo.

A supporto del service, il sistema utilizza un insieme di model per rappresentare le strutture dati interne e di mapper per gestire la conversione tra i modelli esposti dalle API e quelli utilizzati per l'integrazione con sistemi esterni. Questo approccio consente di disaccoppiare il modello interno del servizio dalle rappresentazioni esterne, riducendo l'impatto di eventuali cambiamenti nelle interfacce.

4.5 Tecnologie di integrazione: REST e JPA

L'Archivio Digitale si inserisce in un ecosistema applicativo articolato, caratterizzato dalla presenza di sistemi esterni e di basi dati legacy. Per questo motivo, le tecnologie di integrazione rappresentano un aspetto cruciale del progetto.

I servizi di backend adottano principalmente due modalità di integrazione: servizi REST per la comunicazione inter-sistema e JPA per l'accesso a basi dati relazionali.

4.5.1 Integrazione REST

L'integrazione REST è utilizzata per la comunicazione con sistemi esterni, come il sistema di gestione documentale (Instadoc). REST (Representational

State Transfer) definisce uno stile architetturale per sistemi distribuiti basato su interazioni stateless e su un modello uniforme di accesso alle risorse.

Nel progetto in esame, le chiamate REST avvengono su protocollo HTTPS, garantendo confidenzialità e integrità dei dati scambiati, e utilizzano JSON come formato di rappresentazione. L'adozione di JSON consente di mantenere i messaggi leggeri e interoperabili, favorendo l'integrazione tra tecnologie eterogenee.

Coerentemente con l'architettura descritta nei paragrafi precedenti, l'accesso ai servizi REST esterni è incapsulato all'interno di componenti connector dedicati. Tale scelta consente di isolare la logica di integrazione dal core applicativo, limitando le dipendenze e facilitando eventuali modifiche alle interfacce dei sistemi.

L'utilizzo di interfacce (definite tramite OpenAPI) contribuisce inoltre a rendere esplicite le dipendenze tra provider e consumer, favorendo un'evoluzione controllata dei servizi e riducendo il rischio di incompatibilità.

4.5.2 Integrazione JPA

L'integrazione tramite JPA (Java Persistence API) è adottata nei casi in cui il servizio debba accedere direttamente a basi dati relazionali interne. JPA rappresenta una specifica standard per la gestione della persistenza in ambiente Java, basata sul paradigma dell'Object-Relational Mapping (ORM), che consente di mappare entità di dominio su strutture relazionali.

Nel contesto dell'Archivio Digitale, JPA è utilizzata per interrogare dati persistenti necessari al completamento delle operazioni applicative. Anche in questo caso, l'accesso al database è mediato da componenti connector dedicati, che forniscono un'interfaccia astratta verso il livello di dominio.

L'adozione di un layer di persistenza standardizzato contribuisce a separare la logica applicativa dalla tecnologia di accesso ai dati, migliorando la portabilità del codice e riducendo l'impatto di eventuali evoluzioni nello schema o nel sistema di gestione del database.

4.6 Sicurezza applicativa

La sicurezza applicativa rappresenta un requisito fondamentale per il sistema Archivio Digitale, in considerazione del contesto bancario in cui esso opera e della natura sensibile delle informazioni trattate. L'architettura adotta un modello di sicurezza multilivello, finalizzato a garantire la protezione delle API, la corretta identificazione degli utenti e l'esecuzione delle operazioni esclusivamente entro i limiti dei privilegi assegnati.

Dal punto di vista architetturale, il sistema si basa su un modello di autenticazione e autorizzazione a token, coerente con gli approcci adottati nei sistemi distribuiti moderni e formalizzati nel framework OAuth 2.0 (RFC 6749)[13] e nei meccanismi di tokenizzazione come JSON Web Token (JWT, RFC 7519)[14].

4.6.1 Autenticazione e gestione dei token

L'accesso alle API dell'Archivio Digitale è regolato attraverso l'utilizzo di token di sicurezza che rappresentano il contesto di autenticazione dell'utente e i privilegi associati alla sessione logica. In linea con il paradigma dei sistemi RESTful, l'autenticazione è gestita in modalità stateless: ogni richiesta contiene le informazioni necessarie alla validazione, evitando la persistenza di stato lato server.

Prima del completamento dell'autenticazione utente, alcune operazioni preliminari sono protette tramite meccanismi di identificazione del client (ad esempio chiavi applicative), finalizzati a garantire che le richieste provengano da applicazioni autorizzate.

Il processo di autenticazione prevede più livelli di verifica, distinguendo tra operazioni a rischio standard e operazioni che richiedono un livello di sicurezza rafforzato. Per queste ultime è previsto l'utilizzo di un secondo fattore di autenticazione (Strong Customer Authentication), implementato mediante One-Time Password (OTP). Tale meccanismo riduce il rischio di accessi non autorizzati anche in caso di compromissione delle credenziali primarie.

L'utilizzo di token strutturati (ad esempio JWT) consente di trasportare in modo sicuro informazioni relative all'identità dell'utente e agli attributi di autorizzazione, garantendo integrità tramite firma crittografica.

4.6.2 Autorizzazione e protezione delle operazioni

Una volta autenticato, l'utente può accedere alle funzionalità del sistema in base ai permessi associati al proprio profilo. Le API sono protette tramite controlli di autorizzazione che verificano la presenza, la validità e gli scope del token associato alla richiesta, assicurando che ciascuna operazione sia eseguita nel rispetto delle politiche di accesso definite.

Il concetto di scope, introdotto formalmente nel modello OAuth 2.0, consente di delimitare in modo granulare le azioni consentite a un determinato token, riducendo la superficie di attacco e limitando l'impatto di eventuali compromissioni.

L'adozione di un modello a token stateless contribuisce inoltre alla scalabilità del sistema, in quanto elimina la necessità di mantenere sessioni server-side.

Capitolo 5

Implementazione del sistema

5.1 Implementazione dei servizi applicativi

L'implementazione del sistema Archivio Digitale segue un approccio coerente con l'architettura descritta nei capitoli precedenti: le funzionalità sono suddivise in servizi applicativi autonomi, ciascuno responsabile di un sotto-dominio funzionale e progettato per esporre interfacce REST ben definite. In tale impostazione, l'obiettivo principale non è soltanto separare le funzionalità, ma garantire che ogni servizio possa evolvere nel tempo mantenendo responsabilità chiare, basso accoppiamento e integrazioni controllate verso sistemi esterni.

Nel contesto di questa tesi, pur avendo contribuito anche allo sviluppo di altri servizi, il presente capitolo si concentra sul caso di studio `PostaPersonaleService`, scelto in quanto rappresentativo di un pattern implementativo ricorrente: gestione di richieste sincrone, applicazione di filtri/paginazione/ordinamenti, integrazione tramite connettori REST verso un sistema documentale esterno, e restituzione al canale di output di risposte coerenti con le esigenze del frontend.

Dal punto di vista funzionale, `PostaPersonaleService` espone operazioni riconducibili a quattro macrocategorie:

- recupero delle liste documentali del cliente, filtrabili per rapporto, categoria, tipologia, stato e intervallo temporale, con supporto a paginazione e ordinamento;
- download di uno o più documenti, con restituzione diretta del PDF nel caso singolo o generazione dinamica di un archivio ZIP nel caso multiplo, accompagnata dall'aggiornamento dello stato di lettura presso il sistema documentale;

- recupero dell'elenco dei contratti, anch'esso filtrabile e paginabile;
- recupero delle categorie e tipologie disponibili per documenti e contratti tramite operazioni informative di tipo GET.

Tali operazioni supportano l'esperienza utente prevista dal portale, consentendo una consultazione guidata della documentazione, la selezione multipla e il download dei contenuti. Considerata l'estensione temporale della documentazione consultabile (fino agli ultimi dieci anni), l'adozione sistematica di filtri, paginazione e ordinamento risulta fondamentale per garantire sia l'usabilità lato frontend sia la sostenibilità delle chiamate applicative lato backend.

5.2 Scelte implementative e standard di sviluppo

L'implementazione dei servizi applicativi segue lo standard architetturale descritto nel Capitolo 4. In questa sezione l'attenzione si concentra sulla sua applicazione concreta nel caso di studio considerato, con particolare riferimento ai servizi che integrano sistemi esterni.

5.2.1 Approccio contract-first e definizione delle interfacce

Lo sviluppo dei servizi è stato condotto secondo un approccio contract-first: le interfacce sono state definite e validate prima dell'implementazione, attraverso specifiche formali che descrivono endpoint, strutture di request/response e vincoli sui parametri [15].

Nel caso del `PostaPersonaleService`, le operazioni esposte comprendono endpoint POST per richieste con body strutturato (recupero documenti, recupero contratti, download) ed endpoint GET di natura informativa (categorie e tipologie).

La definizione anticipata delle interfacce ha consentito di stabilire in modo chiaro le responsabilità del servizio, vincolare formalmente formati, cardinalità e pattern dei dati e ridurre ambiguità tra consumer (BFFE) e provider (backend applicativo).

5.2.2 Struttura interna e flusso esecutivo

Lo standard implementativo si traduce in un flusso interno in linea con il contratto esposto e con le responsabilità assegnate al servizio. Nel caso

dell'operazione *recuperaDocumenti*, il servizio riceve i parametri di filtro, costruisce una richiesta compatibile con il sistema documentale e restituisce una risposta già coerente con la paginazione e l'ordinamento richiesti dall'interfaccia.

Il flusso *scaricaDocumenti* evidenzia in modo ancora più chiaro il ruolo attivo del microservizio: dopo aver recuperato i contenuti binari dal sistema documentale, il backend si occupa della composizione dell'output. Nel caso di richiesta multipla, i file vengono aggregati in un archivio ZIP generato lato server. Questa responsabilità non è delegata al sistema esterno, ma rappresenta una scelta progettuale precisa che mantiene il controllo della logica di composizione all'interno del dominio applicativo.

5.2.3 Scelte implementative sul client REST

L'integrazione tecnica con il sistema documentale è realizzata tramite connettori REST sincroni basati su Spring, incapsulati nell'astrazione comune fornita dal framework proprietario.

Dal punto di vista implementativo, il servizio costruisce richieste conformi al contratto del sistema esterno, attende la risposta remota e ne adatta il contenuto al modello interno previsto dall'API esposta. La comunicazione avviene tramite HTTPS e JSON per i dati strutturati, mentre nel caso del download vengono gestiti contenuti binari.

La scelta di mantenere chiamate sincrone è in linea con la natura delle operazioni esposte, che richiedono la disponibilità immediata dei dati. L'isolamento della comunicazione HTTP nei connettori contribuisce a mantenere il disaccoppiamento tra logica di dominio e dettagli infrastrutturali, riducendo l'impatto di eventuali evoluzioni tecnologiche sull'implementazione applicativa.

5.3 Integrazione con i sistemi esterni

L'integrazione con sistemi esterni rappresenta un elemento strutturale dell'implementazione dell'Archivio Digitale. Il backend applicativo non gestisce direttamente la persistenza dei contenuti documentali, ma si configura come livello di mediazione tra il portale e il sistema documentale esterno, demandando a quest'ultimo la gestione effettiva dei contenuti e dei metadati.

Il servizio *PostaPersonaleService* opera quindi come una facciata applicativa verso il sistema documentale, assumendosi la responsabilità di ricevere le richieste dal livello superiore (BFFE), trasformare i modelli interni nei

formati richiesti dal sistema esterno, orchestrare le chiamate ai connettori dedicati e ricomporre le risposte nel formato previsto dalle API esposte.

5.3.1 Integrazione con il sistema documentale

Dal punto di vista funzionale, le operazioni implementate nel servizio possono essere ricondotte a tre casi.

1. **POST /recuperaDocumenti**

Il servizio riceve una richiesta contenente l'elenco dei rapporti di riferimento, i criteri di filtro (intervallo temporale, stato di lettura, categoria, tipologia) e i parametri di paginazione e ordinamento.

Tali informazioni vengono mappate in un modello conforme alle specifiche del sistema documentale e inoltrate tramite un connettore REST dedicato. La risposta ricevuta viene trasformata in un modello interno e restituita al consumer sotto forma di lista paginata di documenti, contenente metadati quali identificativo, data, stato di lettura, descrizione e ulteriori attributi informativi.

2. **POST /scaricaDocumenti**

L'operazione di download consente il recupero di uno o più documenti selezionati dal cliente. Per ciascun identificativo ricevuto, il servizio invoca il sistema documentale al fine di ottenere il contenuto binario associato.

Dal punto di vista implementativo, il servizio mantiene un approccio atomico verso il sistema esterno: ogni documento viene recuperato tramite una chiamata dedicata. I contenuti ottenuti vengono raccolti in una struttura temporanea interna.

Il servizio effettua un'ulteriore chiamata verso il sistema documentale per aggiornare lo stato dei documenti, marcandoli come "letti". Questo aggiornamento avviene esclusivamente a seguito del recupero corretto dei contenuti, garantendo coerenza tra l'azione dell'utente e lo stato persistito.

Una volta completata questa fase, il servizio si occupa della composizione dell'output. Se la richiesta riguarda un solo documento, il contenuto viene restituito direttamente in formato PDF. Nel caso di richiesta multipla, il backend procede alla costruzione di un archivio ZIP generato lato server, contenente tutti i file precedentemente recuperati.

Tale impostazione consente di mantenere semplice l'interfaccia verso il sistema documentale, delegando al microservizio la responsabilità di orchestrare le chiamate e comporre l'output finale in modo coerente con l'esperienza utente prevista dal portale.

3. **Recupero contratti (POST /recuperaContratti)**

L'operazione segue una logica analoga al recupero documenti, con un set di filtri specifici (stato, categoria, responsabile avvio, intervallo temporale). Anche in questo caso, il servizio si occupa della trasformazione dei modelli e della restituzione di una lista paginata in linea con l'interfaccia esposta.

5.3.2 **Gestione degli errori e dei casi anomali**

La gestione degli errori distingue chiaramente tra anomalia tecnica interna e risposta contrattuale verso il consumer. Le eccezioni generate durante l'integrazione con il sistema documentale o durante la fase di composizione dei contenuti vengono classificate tramite codici applicativi interni e successivamente tradotte in risposte HTTP allineate alla specifica del servizio.

Nel caso del download multiplo, ad esempio, un errore di integrazione con il sistema documentale o un problema nella fase di aggregazione dei file viene intercettato e ricondotto a una categoria applicativa specifica. La trasformazione finale in una risposta HTTP (400, 404, 500) avviene tramite meccanismi infrastrutturali condivisi, garantendo uniformità semantica verso l'esterno.

Questa separazione tra rappresentazione tecnica dell'errore e rappresentazione contrattuale consente di mantenere stabile l'interfaccia esposta anche in presenza di variazioni interne nella gestione delle anomalie.

5.4 **Caso di studio: PostaPersonaleService**

5.4.1 **Ruolo del servizio nel sistema Archivio Digitale**

All'interno dell'Archivio Digitale, il *PostaPersonaleService* rappresenta uno dei servizi applicativi più significativi dal punto di vista funzionale e architetturale, poiché implementa le operazioni direttamente percepite dall'utente finale nell'ambito della consultazione documentale.

Come descritto nel Capitolo 2, il sistema consente al cliente di accedere alla propria documentazione, filtrarla secondo diversi criteri e procedere al download dei contenuti. Tali funzionalità trovano concreta realizzazione

proprio nel *PostaPersonaleService*, che costituisce il punto di raccordo tra il dominio applicativo dell'Archivio Digitale e il sistema documentale esterno responsabile della conservazione dei documenti.

Dal punto di vista architetturale (Capitolo 3), il servizio si colloca nel livello di backend applicativo e opera tra il Backend for Frontend e il sistema documentale. Non si limita a inoltrare richieste, ma svolge un ruolo attivo di orchestrazione, trasformazione e composizione dei dati. In questo senso, esso incarna pienamente il modello architetturale adottato nel progetto: disaccoppiamento tra canale e sistema documentale, separazione delle responsabilità e isolamento delle integrazioni esterne, assumendosi la responsabilità della coerenza tra azione utente, stato informativo e formato dell'output restituito.

La scelta di analizzare il *PostaPersonaleService* come caso di studio è motivata proprio dalla concentrazione, in un singolo servizio, di diversi aspetti tipici dei sistemi enterprise orientati all'integrazione: gestione di interazioni REST verso sistemi esterni, trattamento di contenuti binari, aggregazione di risultati multipli, aggiornamento di stato conseguente a un'azione utente e gestione strutturata delle anomalie.

Per tali ragioni, il servizio rappresenta un esempio particolarmente efficace per osservare il passaggio dalla modellazione architetturale alla concreta realizzazione implementativa, che verrà approfondita nel paragrafo successivo attraverso l'analisi dettagliata del flusso di download dei documenti.

5.4.2 Analisi implementativa del flusso ScaricaDocumenti

Tra le operazioni esposte dal *PostaPersonaleService*, il flusso di download dei documenti rappresenta quello con il maggior contenuto implementativo. Esso combina integrazione verso un sistema esterno, aggregazione di contenuti binari, gestione delle anomalie e aggiornamento dello stato informativo, sintetizzando in un'unica operazione le principali responsabilità del servizio.

Dal punto di vista architetturale, il flusso attraversa tutti i livelli interni del microservizio: *Controller*, *Command*, *Service* e *Connector*. La Figura 5.1 riporta il class diagram delle principali classi coinvolte nell'operazione, evidenziando la struttura interna del servizio, le dipendenze tra i componenti e i metodi esposti da ciascun livello.

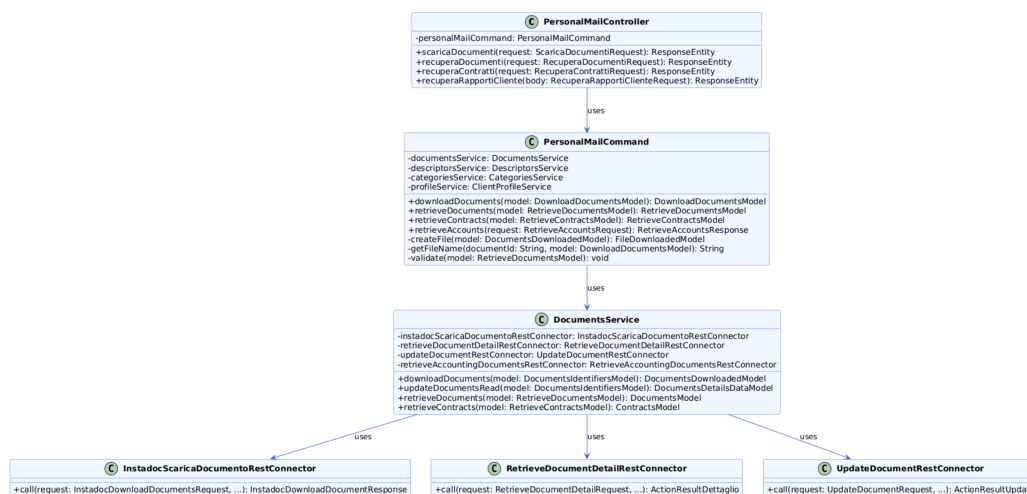


Figura 5.1: Class diagram del PostaPersonaleService

Il diagramma evidenzia come la struttura del servizio rispecchi fedelmente il modello architetturale descritto nel Capitolo 3: la classe *PersonalMailController* espone le operazioni REST e delega l'esecuzione a *PersonalMailCommand*, che orchestra le chiamate verso *DocumentsService*. Quest'ultimo si occupa dell'integrazione con il sistema documentale esterno attraverso tre connettori dedicati: *InstadocScaricaDocumentoRestConnector* per il recupero dei contenuti binari, *RetrieveDocumentDetailRestConnector* per il recupero dei metadati e *UpdateDocumentRestConnector* per l'aggiornamento dello stato di lettura.

A partire da tale struttura, si analizzano di seguito i principali passaggi implementativi del flusso.

Ricezione della richiesta nel Controller

Il punto di ingresso dell'operazione è il Controller REST, che riceve la richiesta POST /scaricaDocumenti. Il Controller non contiene logica di dominio, ma si occupa di trasformare il payload della richiesta nel modello interno, delegare l'esecuzione al livello Command e riconvertire il risultato nel formato previsto dall'API.

In forma semplificata, il comportamento può essere rappresentato come segue:

```

1
2 Controller.scaricaDocumenti(request):
3     model = map(request -> DownloadDocumentsModel)
4     resultModel = command.downloadDocuments(model)
  
```

```

5     response = map(resultModel ->
6         ScaricaDocumentiResponse)
    return response

```

Questa impostazione mantiene il Controller leggero e coerente con il principio di separazione delle responsabilità.

Orchestrazione del flusso nel Command

Il livello Command coordina l'intero flusso dell'operazione, orchestrando in sequenza le chiamate ai servizi applicativi e gestendo le anomalie. Il flusso procede secondo i seguenti passi: recupero dei contenuti binari, aggiornamento dello stato di lettura e recupero dei dettagli documentali presso il sistema documentale, e infine composizione dell'output finale.

Il comportamento può essere sintetizzato come segue:

```

1 Command.downloadDocuments(model):
2     try:
3         // 1. Recupero contenuti binari da Instadoc
4         file = service.downloadDocuments(model)
5         model.setDocuments(file)
6
7         // 2. Aggiornamento stato "letto" e recupero
8         dettagli documenti
9         details = service.updateDocumentsRead(model)
10        model.setDetails(details)
11
12        // 3. Composizione output finale (PDF singolo o
13        archivio ZIP)
14        result = createFile(model.getDocuments())
15        model.setDocument(result.getFile())
16        return model
17
18    catch InternalServiceRuntimeException e:
19        log.error("Download error", e)
20        throw e
21    catch Exception e:
22        log.error("Unexpected error during download", e)
23        throw new InternalServiceRuntimeException(
24            INSTADOC_ERROR)

```

Il Command non implementa logica di dominio, ma stabilizza il flusso esecutivo, garantisce la corretta sequenza delle operazioni e centralizza la gestione delle anomalie.

Recupero dei contenuti binari nel Service

Il Service realizza il recupero effettivo dei documenti. Per ciascun identificativo ricevuto, invoca il connettore REST ScaricaDocumentoRestConnector verso il sistema documentale, recupera il contenuto binario e lo accumula in una struttura a mappa indicizzata per identificativo documento.

```
1
2 Service.downloadDocuments(model):
3
4     documents = {}
5
6     for each documentId in model.getDocumentsId():
7         response = scaricaDocumentoRestConnector.call(
8             documentId)
9         document = mapper.map(response -> Document)
10        documents.put(documentId, document)
11
12    result.setDocuments(documents)
13    return result
```

Aggiornamento dello stato “letto”

Dopo il recupero dei contenuti, il Service aggiorna lo stato di lettura di ciascun documento presso il sistema documentale. Per ogni documento, vengono effettuate due chiamate distinte tramite connettori dedicati: la prima tramite RetrieveDocumentDetailRestConnector per recuperare i dettagli del documento, la seconda tramite UpdateDocumentRestConnector per aggiornarne lo stato. In entrambi i casi, una risposta diversa da ”OK” genera un’eccezione applicativa.

```
1
2 Service.updateDocumentsRead(model):
3     for each documentId in model.getDocumentsId():
4         // Recupero dettagli documento
5         detailResult = retrieveDetailRestConnector.call(
6             documentId)
7         // Aggiornamento stato letto
```

```

7     updateResult = updateDocumentRestConnector.call(
           documentId)
8     documentsDetailsData.add(documentId,
           detailResult)
9     return documentsDetailsData

```

Composizione dell'output nel Command

La fase finale dell'operazione è gestita dal metodo privato `createFile`, che appartiene al `Command`. Se la richiesta riguarda un singolo documento, il contenuto viene restituito direttamente in formato PDF. Nel caso di richiesta multipla, i file vengono aggregati in un archivio ZIP generato dinamicamente lato backend tramite `ZipOutputStream`. Eventuali errori nella fase di compressione vengono intercettati e tradotti in un'eccezione applicativa dedicata.

```

1
2 Command.createFile(documents):
3     if documents.size() == 1:
4         return singleDocument
5     else:
6         zipOut = new ZipOutputStream()
7         for each document in documents:
8             zipOut.addEntry(document.getName(),
9                 document.getFile())
10        result.setName("documents.zip")
11        result.setFile(zipOut.toByteArray())
12        return result

```

La responsabilità di aggregazione è quindi assunta interamente dal `Command`, che mantiene semplice l'interfaccia verso il `Service` e verso il sistema documentale esterno, isolando la logica di composizione dell'output all'interno del dominio applicativo.

5.4.3 Considerazioni progettuali

L'analisi del `PostaPersonaleService` evidenzia la coerenza tra le scelte architettoniche definite a livello logico e la loro traduzione implementativa. La struttura del servizio consente di mantenere separati i diversi livelli di responsabilità, migliorando la leggibilità del codice e facilitando l'evoluzione futura delle componenti.

Un elemento particolarmente significativo riguarda il ruolo assunto dal servizio nella gestione del flusso di download dei documenti. Il microservizio

non si limita a inoltrare richieste verso il sistema documentale, ma svolge una funzione attiva di orchestrazione: coordina le chiamate verso il sistema esterno, gestisce il recupero dei contenuti binari e, nel caso di richieste multiple, aggrega i risultati generando dinamicamente un archivio compresso. Questa scelta progettuale introduce una maggiore responsabilità lato backend, ma consente di mantenere il sistema documentale isolato da logiche di composizione e presentazione dei contenuti.

Dal punto di vista dell'integrazione, l'utilizzo di connettori dedicati consente di isolare la tecnologia di comunicazione con i sistemi esterni dal dominio applicativo. Tale separazione riduce l'accoppiamento tra componenti e rende più semplice l'eventuale evoluzione delle interfacce esterne. Analogamente, la gestione strutturata delle eccezioni permette di distinguere tra errore tecnico interno e risposta contrattuale verso il consumer, contribuendo a mantenere stabile l'interfaccia esposta dal servizio.

Nel complesso, il PostaPersonaleService rappresenta un esempio concreto di microservizio orientato all'integrazione, in cui le scelte progettuali privilegiano modularità, controllo del flusso applicativo e isolamento delle dipendenze esterne.

Capitolo 6

Valutazione sperimentale e analisi delle prestazioni

6.1 Obiettivi della valutazione sperimentale

La realizzazione dell'Archivio Digitale ha richiesto l'adozione di un'architettura basata su microservizi e su un insieme di componenti applicativi distribuiti, progettati per supportare in modo scalabile le operazioni di consultazione documentale da parte degli utenti. In un contesto di questo tipo, oltre alla correttezza funzionale delle operazioni implementate, assume particolare rilevanza la valutazione del comportamento del sistema in presenza di carico concorrente e di operazioni potenzialmente intensive dal punto di vista computazionale.

Il presente capitolo ha quindi l'obiettivo di analizzare sperimentalmente le prestazioni del sistema, con particolare riferimento al servizio applicativo `PostaPersonaleService`, già introdotto nel Capitolo 5 come caso di studio rappresentativo dell'implementazione dell'Archivio Digitale. Tale servizio costituisce infatti il punto di accesso alle funzionalità di consultazione e download dei documenti da parte dell'utente finale e rappresenta uno dei componenti più sensibili al carico, poiché coinvolge l'integrazione con il sistema documentale esterno e la gestione di contenuti binari.

L'analisi sperimentale si concentra in particolare sul flusso di download dei documenti, implementato tramite l'endpoint `POST /scaricaDocumenti`. Questa operazione rappresenta uno scenario significativo dal punto di vista prestazionale, in quanto combina più fattori potenzialmente impattanti sul tempo di risposta del sistema: l'invocazione di servizi esterni, il trasferimento di contenuti binari e, nel caso di download multipli, l'aggregazione dei file in un archivio compresso generato lato backend.

La valutazione si pone due obiettivi complementari: osservare l'evoluzione dei tempi di risposta e del throughput al crescere del carico concorrente, e verificare se l'architettura adottata sia in grado di sostenere tale carico mantenendo livelli di prestazione in linea con l'utilizzo previsto del sistema.

I risultati ottenuti consentiranno di individuare eventuali colli di bottiglia nel flusso applicativo, fornendo un riscontro concreto sull'efficacia delle scelte architettoniche e sulla capacità del sistema di operare in condizioni realistiche.

6.2 Metodologia di analisi e metriche adottate (GQM)

Per strutturare in modo sistematico la valutazione delle prestazioni del sistema è stata adottata la metodologia Goal-Question-Metric (GQM) [16]. Il modello GQM fornisce un approccio metodologico per definire obiettivi di misurazione, formulare le domande di ricerca associate e individuare le metriche necessarie per rispondere a tali domande in modo oggettivo e verificabile [17].

Secondo questo approccio, l'analisi sperimentale viene strutturata attraverso tre livelli concettuali:

- **Goal (obiettivo):** definisce lo scopo della valutazione e l'aspetto del sistema che si intende analizzare;
- **Question (domanda):** identifica le domande di ricerca che permettono di verificare il raggiungimento dell'obiettivo;
- **Metric (metrica):** individua le misure quantitative necessarie per rispondere alle domande formulate.

Nel contesto di questa tesi, il modello GQM è stato utilizzato per definire in modo esplicito gli obiettivi della valutazione prestazionale del sistema Archivio Digitale, con particolare riferimento al servizio applicativo PostaPersonaleService e al flusso di download dei documenti.

Goal

Valutare il comportamento prestazionale del servizio durante l'esecuzione delle operazioni di consultazione e download documentale, al fine di verificare la capacità dell'architettura implementata di gestire richieste concorrenti e operazioni di recupero documentale provenienti dal canale web.

Question

A partire dall'obiettivo generale, sono state formulate le seguenti domande di analisi:

- Come varia il tempo di risposta delle operazioni al crescere del numero di richieste concorrenti?
- Qual è il throughput massimo sostenibile dal servizio nelle condizioni di test considerate?
- In quali condizioni di carico emergono eventuali degradazioni delle prestazioni o errori applicativi?
- L'integrazione con il sistema documentale esterno introduce potenziali colli di bottiglia nel flusso di download?

Metric

Per rispondere alle domande individuate sono state raccolte e analizzate diverse metriche prestazionali ottenute durante l'esecuzione dei test di carico. In particolare, sono state considerate le seguenti misure:

- Tempo medio di risposta (Average Response Time): misura il tempo medio necessario al sistema per completare una richiesta;
- Throughput: numero di richieste elaborate dal sistema per unità di tempo;
- Tasso di errore (Error Rate): percentuale di richieste che terminano con errore applicativo o tecnico durante l'esecuzione dei test.

L'adozione del modello GQM consente quindi di collegare in modo esplicito gli obiettivi della valutazione alle metriche osservate, garantendo coerenza tra le domande di ricerca formulate e i risultati analizzati nelle sezioni successive dedicate alla presentazione e all'analisi dei risultati sperimentali.

6.3 Ambiente di test e configurazione sperimentale

La valutazione sperimentale del sistema Archivio Digitale è stata condotta in ambiente di sviluppo aziendale, configurato per replicare il comportamento del sistema nelle condizioni operative previste dal portale. L'obiettivo di questa fase è analizzare il comportamento del servizio applicativo in presenza di richieste concorrenti, valutando tempi di risposta, capacità di gestione del carico e possibili criticità architetturali.

6.3.1 Ambiente di esecuzione

L'accesso al sistema avviene tramite un'infrastruttura aziendale basata su Virtual Desktop Infrastructure (VDI), che consente agli sviluppatori di utilizzare un ambiente remoto controllato dalla banca. Attraverso la VDI è possibile accedere al portale dell'Archivio Digitale mediante URL intranet aziendali, ad esempio:

```
https://svil.intranet.servizi/...
```

L'ambiente di sviluppo ospita le componenti applicative descritte nei capitoli precedenti, tra cui:

- frontend web basato su Single Page Application;
- servizi di Backend for Frontend;
- microservizi di backend applicativo;
- integrazione con il sistema documentale esterno.

Il deployment dei servizi avviene tramite pipeline di integrazione continua gestite mediante Jenkins, che automatizza il processo di build, test e distribuzione delle nuove versioni del codice nei diversi ambienti applicativi.

6.3.2 Accesso ai servizi e autenticazione

Le operazioni del sistema sono esposte come API REST accessibili tramite rete intranet aziendale. Per invocare tali API è necessario effettuare preliminarmente l'autenticazione al portale, ottenendo un contesto di sessione valido.

Le richieste verso i servizi applicativi includono quindi le informazioni di autenticazione necessarie, tipicamente trasmesse tramite header HTTP (ad esempio tramite token di autorizzazione). Questo meccanismo garantisce che le operazioni siano eseguite esclusivamente da client autenticati e autorizzati, in linea con i requisiti di sicurezza descritti nel Capitolo 4.

Durante la fase di sperimentazione, le API sono state invocate direttamente tramite strumenti di testing delle API REST, consentendo di simulare richieste verso i servizi backend e analizzare il comportamento del sistema in condizioni controllate.

6.3.3 Caratteristiche dei dati di test

I test sono stati eseguiti utilizzando dati rappresentativi del dominio applicativo del sistema. In particolare, i documenti gestiti dall'Archivio Digitale

sono file in formato PDF contenenti prevalentemente testo e con dimensione media di circa 200 KB per documento.

Sono stati considerati diversi scenari di utilizzo coerenti con il comportamento reale degli utenti del portale:

- download di un singolo documento;
- download multiplo di documenti selezionati;
- download di collezioni più estese di documenti.

Nel caso del download multiplo, il backend applicativo aggrega i contenuti recuperati dal sistema documentale e genera un archivio compresso in formato ZIP prima della restituzione al client. Questa tipologia di operazione rappresenta un caso interessante dal punto di vista prestazionale, poiché combina chiamate multiple verso un sistema esterno, gestione di contenuti binari e aggregazione dei risultati lato backend.

6.3.4 Strumenti di testing

Per l'esecuzione dei test di carico è stato utilizzato Apache JMeter, uno strumento open source ampiamente utilizzato per la simulazione di carichi concorrenti su servizi web e API REST.

JMeter consente di generare richieste HTTP verso i servizi applicativi simulando il comportamento di un numero configurabile di utenti virtuali. Attraverso la configurazione dei Thread Group è possibile definire il numero di utenti concorrenti, il tempo di ramp-up e il numero di iterazioni delle richieste, riproducendo scenari di carico progressivo simili a quelli osservabili in un contesto reale.

Nel contesto di questa sperimentazione, le richieste HTTP sono state configurate per invocare l'endpoint POST/scaricaDocumenti, includendo gli header di autenticazione necessari per accedere ai servizi applicativi del sistema Archivio Digitale. In questo modo è stato possibile simulare il comportamento del frontend durante le operazioni di download dei documenti.

Durante l'esecuzione dei test, JMeter ha raccolto diverse metriche prestazionali, tra cui tempo medio di risposta, throughput e tasso di errore. I risultati sono stati analizzati tramite i report aggregati generati dallo strumento, che sintetizzano le metriche raccolte per ciascuna configurazione di test.

6.4 Test di carico e di stress

Per valutare il comportamento del servizio PostaPersonaleService in condizioni operative realistiche, è stata progettata una campagna di test finalizzata a simulare l'interazione di più utenti concorrenti con il sistema.

Nel contesto della valutazione delle prestazioni dei sistemi software è comune distinguere tra test di carico (load testing) e test di stress (stress testing).

I test di carico hanno l'obiettivo di valutare il comportamento del sistema in presenza di un numero crescente di utenti o richieste, mantenendosi entro livelli di utilizzo considerati plausibili o attesi in esercizio. Questo tipo di test consente di osservare come variano metriche come tempo di risposta, throughput e stabilità del sistema al crescere del carico applicativo [18].

I test di stress, invece, mirano a portare il sistema oltre le condizioni operative normali, con l'obiettivo di individuare i limiti di funzionamento e osservare eventuali fenomeni di degradazione delle prestazioni o comparsa di errori quando il sistema viene sottoposto a carichi particolarmente elevati [19].

La campagna sperimentale è stata quindi articolata, secondo le due tipologie di test appena descritte, con l'obiettivo di analizzare l'evoluzione dei tempi di risposta e verificare la capacità del sistema di mantenere prestazioni accettabili al crescere del numero di richieste simultanee.

6.4.1 Scenari di test

Gli scenari sperimentali sono stati progettati per rappresentare diversi pattern di utilizzo del portale da parte degli utenti. In particolare, sono stati considerati tre scenari principali relativi al download dei documenti.

Il primo scenario riguarda il download di un singolo documento, che rappresenta il caso più semplice dal punto di vista applicativo. In questo caso il backend si limita a recuperare il contenuto dal sistema documentale e a restituirlo direttamente al client.

Il secondo scenario riguarda il download multiplo di documenti, in cui l'utente seleziona più elementi dalla lista disponibile nel portale. Il backend effettua una chiamata al sistema documentale per ciascun documento richiesto e successivamente aggrega i contenuti in un archivio ZIP prima della restituzione al client.

Il terzo scenario considera un caso di download esteso, in cui viene richiesto un numero più elevato di documenti (20 elementi). Questo scenario è particolarmente interessante dal punto di vista prestazionale, poiché richiede

al servizio di gestire numerose chiamate verso il sistema documentale e di effettuare un'operazione di aggregazione più onerosa.

La tabella seguente riassume gli scenari considerati.

Scenario	Descrizione
S1	Download di un singolo documento
S2	Download multiplo di 5 documenti
S3	Download esteso di 20 documenti

6.4.2 Configurazione dei test di carico

I test di carico sono stati progettati per simulare un numero crescente di utenti concorrenti che invocano il servizio di download documenti.

Per ciascuno degli scenari definiti sono stati eseguiti test con livelli progressivi di concorrenza, al fine di osservare l'evoluzione delle prestazioni del sistema al crescere del carico applicativo.

I livelli di carico considerati sono stati:

Utenti concorrenti	Ramp up
10	5
50	10
100	20

Queste configurazioni rappresentano condizioni di utilizzo realistiche del sistema e consentono di analizzare il comportamento del servizio in presenza di un numero crescente di richieste simultanee.

Durante l'esecuzione dei test, ciascun utente simulato invoca l'endpoint di download utilizzando parametri coerenti con lo scenario di riferimento (S1, S2, S3).

Per ogni configurazione sono state raccolte diverse metriche prestazionali, tra cui:

- tempo medio di risposta delle richieste;
- tempo minimo e massimo di risposta;
- tasso di errore restituito dal servizio.
- eventuali errori restituiti dal servizio.

Queste metriche consentono di valutare la capacità del sistema di gestire richieste concorrenti mantenendo tempi di risposta accettabili.

6.4.3 Configurazione dei test di stress

Oltre ai test di carico, è stato condotto un insieme di test di stress con l'obiettivo di osservare il comportamento del sistema in condizioni di carico più elevate.

In questo caso il numero di utenti concorrenti è stato progressivamente aumentato fino a 200 utenti simultanei, un valore tale da mettere sotto pressione il servizio applicativo e le integrazioni con il sistema documentale. Sono state inoltre testate diverse configurazioni di ramp-up, per analizzare l'effetto della velocità di introduzione del carico sulle prestazioni del servizio.

Per ogni scenario sono state utilizzate le seguenti configurazioni:

<u>Utenti concorrenti</u>	<u>Ramp up</u>
200	30
200	40

Lo scopo di questa fase non è valutare il comportamento normale del sistema, ma individuare:

- eventuali degradazioni significative delle prestazioni;
- comparsa di errori applicativi;
- limiti operativi del servizio.

6.5 Analisi dei risultati

In questa sezione vengono presentati e analizzati i risultati ottenuti durante l'esecuzione della campagna sperimentale descritta nella sezione precedente. L'obiettivo dell'analisi è osservare il comportamento del servizio PostaPersonaleService al variare del numero di utenti concorrenti e della complessità dell'operazione di download documentale.

I test sono stati eseguiti considerando tre scenari principali, caratterizzati da un diverso numero di documenti richiesti tramite l'endpoint POST /scaricaDocumenti:

- S1: download di un singolo documento;
- S2: download multiplo di 5 documenti;
- S3: download esteso di 20 documenti.

Per ciascuno scenario sono stati analizzati i tempi di risposta medi, i valori minimo e massimo osservati, il throughput del sistema e la percentuale di errori restituiti dal servizio.

6.5.1 Scenario S1 – Download di un documento

Il primo scenario analizzato riguarda il download di un singolo documento, che rappresenta il caso più semplice dal punto di vista applicativo: il servizio backend deve recuperare un unico file dal sistema documentale e restituirlo direttamente al client, senza effettuare operazioni di aggregazione o elaborazione aggiuntiva.

Test	Users	Ramp	Avg (ms)	Min	Max	Error %	Throughput
T1	5	5	195	177	214	0%	1.2/sec
T2	10	5	189	167	242	0%	2.1/sec
T3	50	10	218	165	1203	0%	4.5/sec
T4	100	20	193	151	1138	1%	5.0/sec
T5	200	30	270	150	678	1.5%	10/sec
T6	200	40	213	154	450	0%	6.7/sec

Tabella 6.1: Risultati dei test per lo scenario S1

Come si può osservare dalla tabella, il sistema mantiene tempi di risposta medi relativamente stabili fino a circa 100 utenti concorrenti, con valori compresi tra circa 189 ms e 195 ms nei test con 5, 10 e 100 utenti, e un lieve incremento a circa 218 ms con 50 utenti concorrenti. Per analizzare più chiaramente questo andamento, la Figura 6.1 riporta il tempo medio di risposta in funzione del carico applicato.

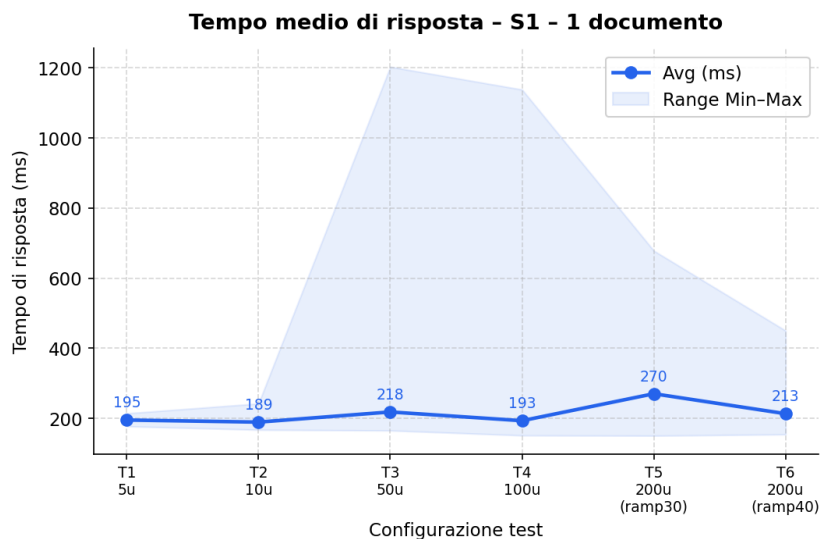


Figura 6.1: Tempo medio di risposta nello scenario S1

Con 200 utenti concorrenti si osserva un incremento più evidente della latenza, che raggiunge circa 270 ms nel test con ramp-up più rapido. Quando invece il carico viene introdotto più gradualmente, con ramp-up pari a 40 secondi, il tempo medio scende a circa 213 ms, evidenziando come la velocità di introduzione degli utenti nel sistema influenzi la stabilità delle prestazioni.

Il comportamento del sistema risulta complessivamente stabile anche in termini di affidabilità, come evidenziato dalla Figura 6.2 che riporta l'andamento del tasso di errore.

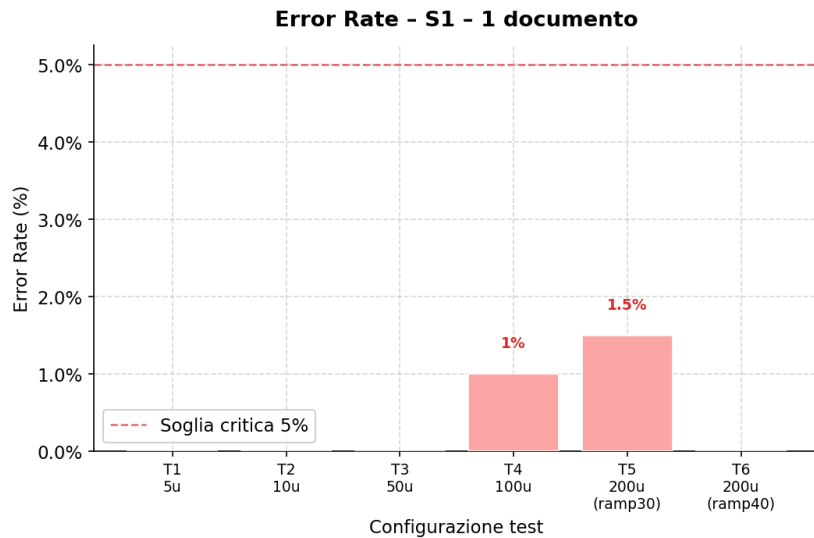


Figura 6.2: Tasso di errore nello scenario S1

Nei test con carico fino a 50 utenti non sono stati osservati errori. Con 100 utenti compare una percentuale di errore pari a circa 1%, che cresce leggermente fino a 1.5% nel test con 200 utenti e ramp-up aggressivo, mentre si azzerava nuovamente nel test con ramp-up più graduale. Questo suggerisce che, per operazioni semplici come il download di un singolo documento, il sistema è in grado di gestire carichi elevati mantenendo una buona affidabilità.

Infine, la Figura 6.3 mostra l'andamento del throughput, che cresce progressivamente al crescere del carico, raggiungendo il valore massimo di 10 req/sec nel test con 200 utenti e ramp-up più aggressivo, mentre scende a 6.7 req/sec con ramp-up più graduale.

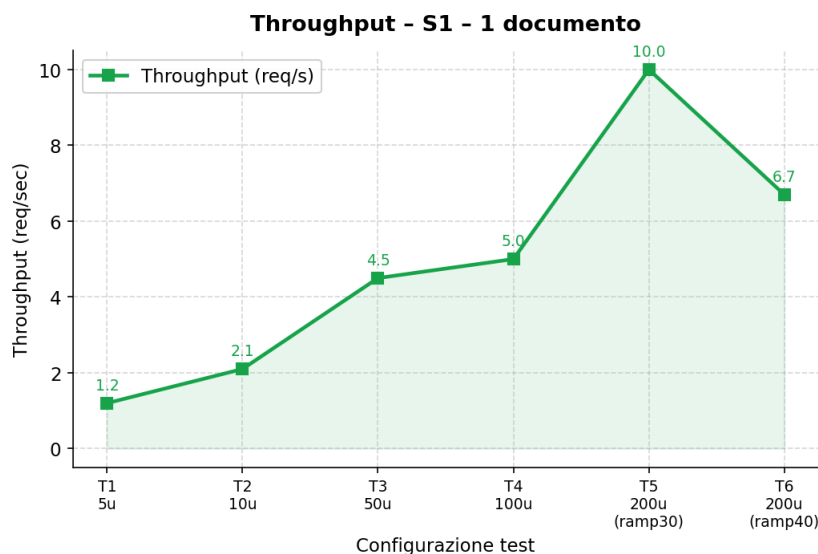


Figura 6.3: Throughput nello scenario S1

6.5.2 Scenario S2 – Download multiplo di 5 documenti

Il secondo scenario analizzato riguarda il download multiplo di 5 documenti. In questo caso il servizio backend deve effettuare più richieste verso il sistema documentale e successivamente aggregare i contenuti in un archivio ZIP prima della restituzione al client, introducendo un carico computazionale superiore rispetto allo scenario precedente.

Test	Users	Ramp	Avg (ms)	Min	Max	Error %	Throughput
T1	10	5	227	171	690	0%	1.6/sec
T2	50	10	435	353	1057	0%	4.9/sec
T3	100	20	479	188	1555	1%	4.9/sec
T4	200	30	891	195	1718	18.5%	6.5/sec
T5	200	40	593	306	1473	2.5%	5.0/sec

Tabella 6.2: Risultati dei test per lo scenario S2

I dati riportati in tabella mostrano un aumento significativo dei tempi di risposta rispetto allo scenario S1. Con 10 utenti concorrenti il tempo medio è già pari a circa 227 ms, e cresce rapidamente all'aumentare del carico: con 50 utenti raggiunge circa 435 ms, mentre con 100 utenti si attesta intorno a 479 ms. L'andamento è illustrato nella Figura 6.4.

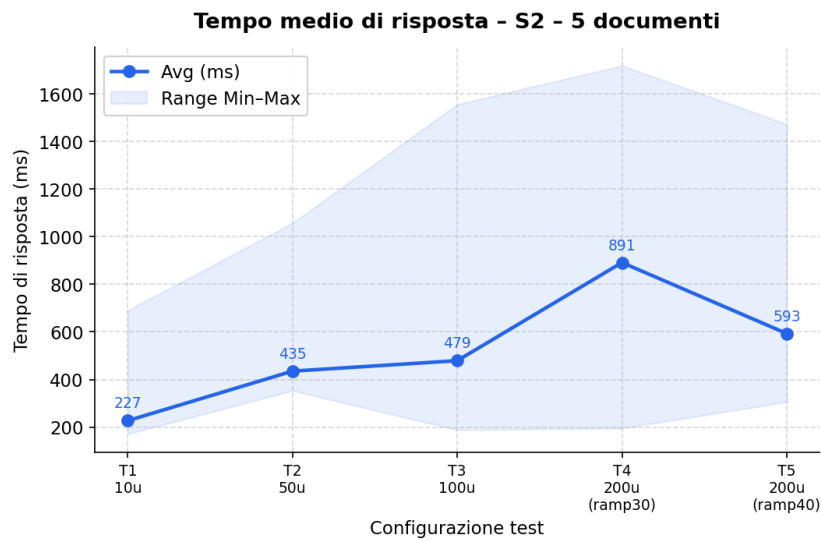


Figura 6.4: Tempo medio di risposta nello scenario S2

Nei test con 200 utenti la situazione diventa più critica: con ramp-up pari a 30 secondi il tempo medio raggiunge circa 891 ms, mentre aumentando il ramp-up a 40 secondi si riduce a 593 ms, confermando che una distribuzione più graduale del carico contribuisce a migliorare il comportamento del sistema.

Dal punto di vista dell'affidabilità, la Figura 6.5 mostra come il tasso di errore rimanga nullo fino a 50 utenti, per poi crescere progressivamente con l'aumentare del carico.

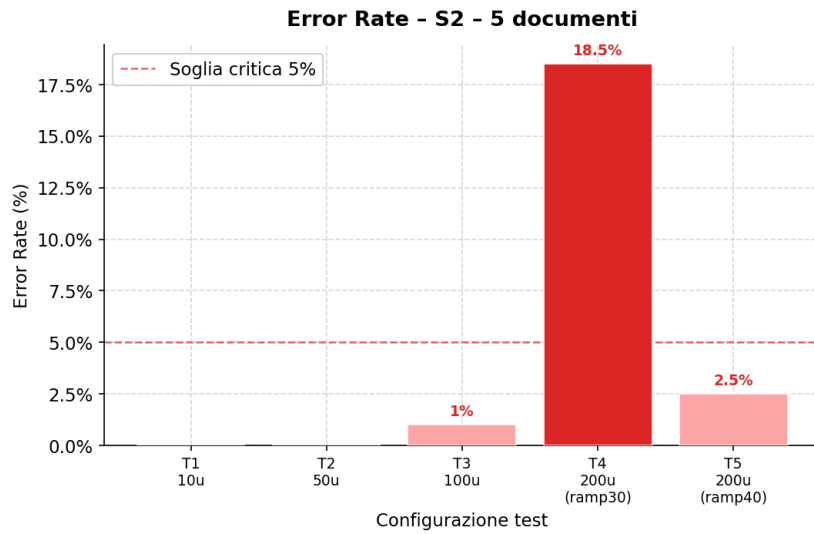


Figura 6.5: Tasso di errore nello scenario S2

Con 100 utenti compare una percentuale di errore pari a circa 1%, che nei test con 200 utenti cresce in modo significativo: raggiunge circa 18.5% con ramp-up pari a 30 secondi e si riduce a circa 2.5% con ramp-up di 40 secondi. Questo comportamento suggerisce che il sistema diventa più sensibile alla concorrenza quando l'operazione richiede la gestione di più documenti e l'aggregazione lato backend.

Per quanto riguarda il throughput, la Figura 6.6 mostra come il sistema raggiunga rapidamente la saturazione della propria capacità elaborativa, con valori che si stabilizzano tra 4.9 req/sec e 6.5 req/sec nei test con carico più elevato.

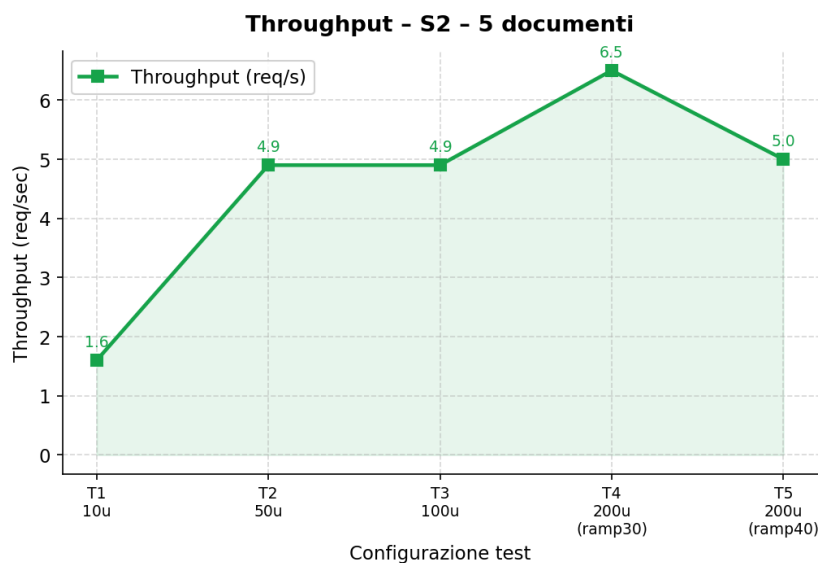


Figura 6.6: Throughput nello scenario S2

6.5.3 Scenario S3 – Download esteso di 20 documenti

Il terzo scenario rappresenta il caso più complesso analizzato nella sperimentazione, in cui l'utente richiede il download di 20 documenti da recuperare dal sistema documentale e aggregare in un archivio compresso prima della restituzione al client.

Test	Users	Ramp	Avg (ms)	Min	Max	Error %	Throughput
T1	10	5	412	310	893	0%	1.4/sec
T2	50	10	874	589	2341	2%	4.2/sec
T3	100	20	1203	612	3892	8.5%	5.1/sec
T4	200	30	2847	701	7654	32%	6.1/sec
T5	200	40	1654	698	5123	21%	4.3/sec

Tabella 6.3: Risultati dei test per lo scenario S3

Come emerge dai dati in tabella, i tempi di risposta subiscono un incremento molto significativo già con livelli di carico moderati. Con 10 utenti concorrenti il tempo medio è pari a circa 412 ms, quasi il doppio rispetto allo scenario S1, e cresce ulteriormente fino a circa 874 ms con 50 utenti. Con 100 utenti il tempo medio supera 1.2 secondi. La Figura 6.7 illustra chiaramente questo andamento.

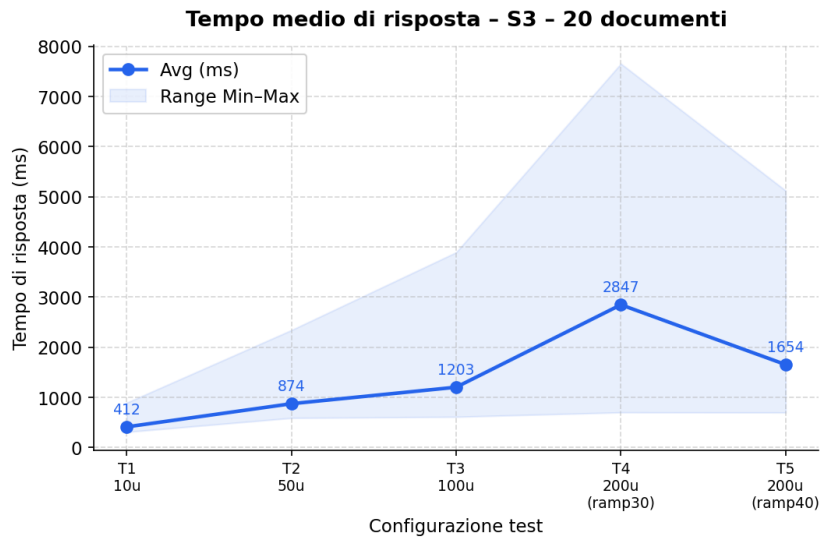


Figura 6.7: Tempo medio di risposta nello scenario S3

Nei test con 200 utenti concorrenti si osserva un forte degrado delle prestazioni: con ramp-up di 30 secondi il tempo medio raggiunge circa 2847 ms, prossimo ai 3 secondi, mentre con ramp-up di 40 secondi si riduce a circa 1654 ms, pur rimanendo sensibilmente superiore rispetto agli scenari precedenti.

L'analisi del tasso di errore, riportata nella Figura 6.8, mostra un peggioramento marcato rispetto agli scenari S1 e S2.

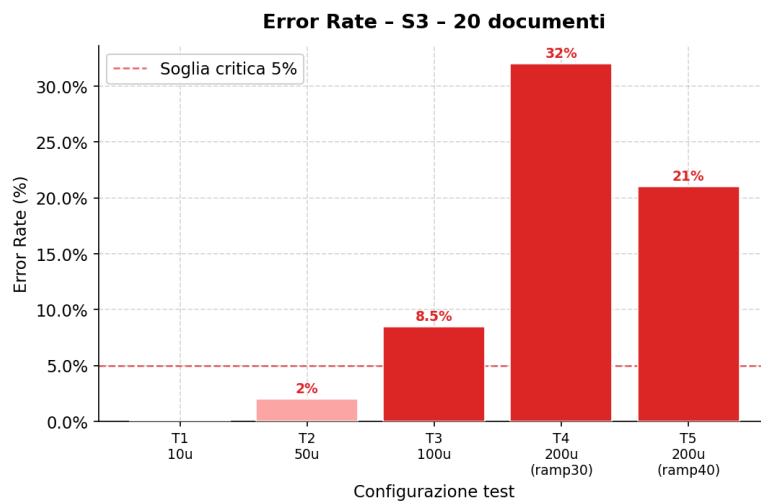


Figura 6.8: Tasso di errore nello scenario S3

Nei test con 10 utenti non sono stati osservati errori, ma già con 50 utenti compare una percentuale di errore pari al 2%, che cresce fino a 8.5% con 100 utenti. Con 200 utenti il tasso raggiunge circa 32% nel caso di ramp-up più aggressivo e circa 21% con ramp-up più graduale, evidenziando una condizione di forte pressione sul sistema.

Il throughput, illustrato nella Figura 6.9, si mantiene su valori compresi tra 4.2 req/sec e 6.1 req/sec, con un andamento non lineare che riflette la saturazione crescente del sistema al crescere del carico.

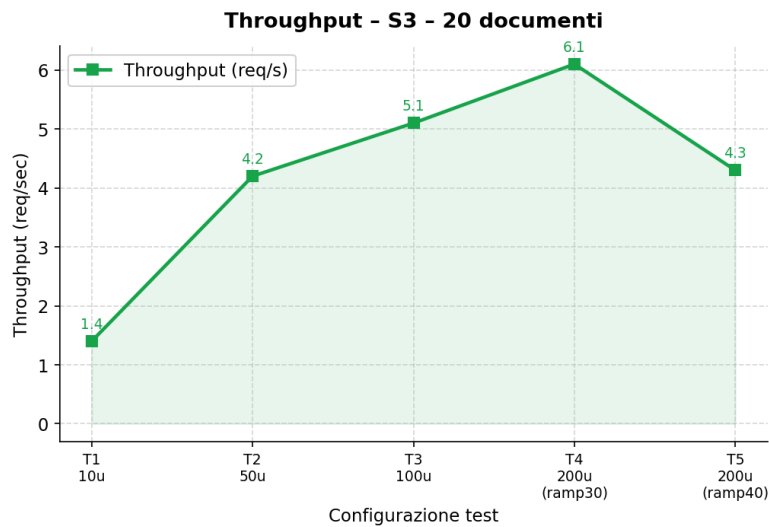


Figura 6.9: Throughput nello scenario S3

6.5.4 Confronto tra gli scenari

Il confronto tra i tre scenari analizzati consente di valutare in modo sistematico l'impatto del numero di documenti richiesti sulle prestazioni complessive del sistema. La Figura 6.10 mette a confronto i tempi medi di risposta nei tre scenari al variare del numero di utenti concorrenti.

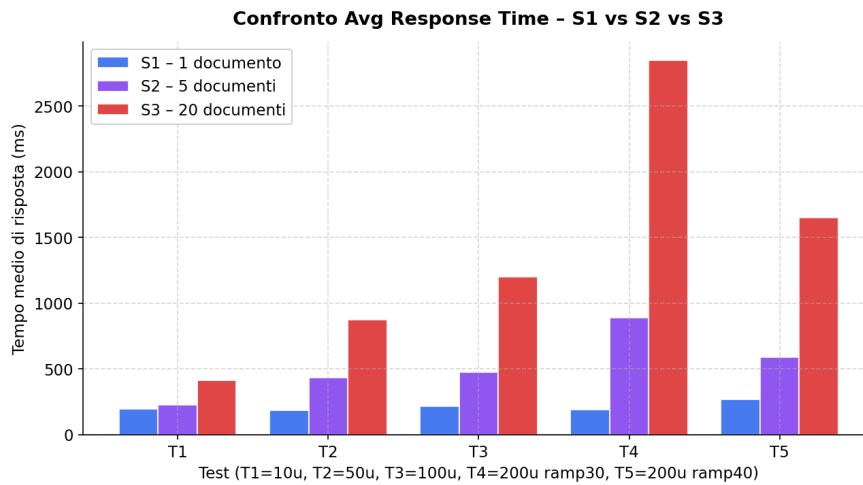


Figura 6.10: Confronto dei tempi medi di risposta tra gli scenari S1, S2 e S3

Dal grafico emerge chiaramente una crescita progressiva della latenza all'aumentare della complessità dell'operazione. Con 100 utenti concorrenti, il tempo medio di risposta rimane intorno ai 193 ms nello scenario S1, cresce fino a circa 479 ms nello scenario S2 e supera 1.2 secondi nello scenario S3. Questo andamento conferma che il numero di documenti richiesti rappresenta il principale fattore determinante del carico applicativo del servizio.

La Figura 6.11 riporta invece il confronto tra i tassi di errore nei tre scenari, evidenziando come la soglia critica del 5% venga superata in modo più rapido e con maggiore intensità negli scenari di download multiplo.

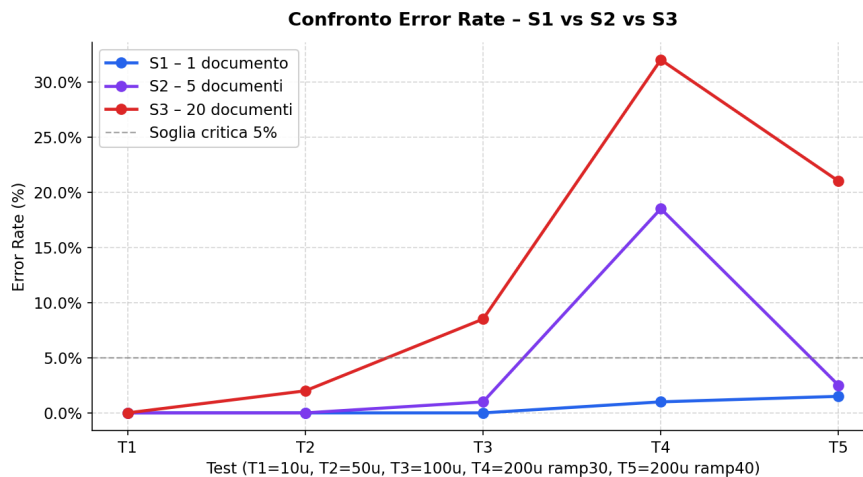


Figura 6.11: Confronto del tasso di errore tra gli scenari S1, S2 e S3

Negli stress test con 200 utenti concorrenti, lo scenario S1 mantiene un tasso di errore molto contenuto, compreso tra 0% e 1.5%. Lo scenario S2 raggiunge invece valori superiori al 18% con ramp-up aggressivo, che si riducono al 2.5% con un ramp-up più graduale. Lo scenario S3 mostra il degrado più marcato, con un tasso di errore che arriva fino al 32%, indicando una condizione di forte saturazione del sistema.

Nel complesso, i risultati mostrano che il sistema gestisce efficacemente il download singolo e i download multipli di dimensione moderata. Nei casi di download esteso con carichi elevati si osserva invece un degrado progressivo delle prestazioni, riconducibile alla moltiplicazione delle chiamate verso il sistema documentale e al costo dell'aggregazione ZIP.

6.6 Discussione dei risultati

I risultati ottenuti nella fase sperimentale permettono di formulare alcune considerazioni sul comportamento prestazionale del servizio PostaPersonaleService e, più in generale, sulla capacità dell'architettura dell'Archivio Digitale di sostenere scenari di utilizzo realistici.

Un primo elemento che emerge con chiarezza riguarda la buona stabilità del sistema nello scenario più semplice, corrispondente al download di un singolo documento. In questo caso i tempi medi di risposta rimangono contenuti anche al crescere del numero di utenti concorrenti, mantenendosi nell'ordine di poche centinaia di millisecondi fino a carichi elevati. Anche la percentuale di errore rimane nulla o molto bassa, indicando un livello di affidabilità complessivamente buono per operazioni che richiedono una sola interazione con il sistema documentale.

Il comportamento cambia invece negli scenari di download multiplo, nei quali il backend deve orchestrare più chiamate verso il sistema documentale e successivamente aggregare i risultati in un archivio ZIP. Il confronto tra gli scenari S1, S2 e S3 mostra una crescita progressiva della latenza media all'aumentare del numero di documenti richiesti, in linea con la struttura implementativa descritta nel Capitolo 5: il flusso ScaricaDocumenti richiede infatti, per ciascun documento selezionato, una chiamata dedicata verso il sistema documentale seguita da un'attività di aggregazione lato backend.

Un secondo aspetto rilevante riguarda il comportamento del sistema al crescere del numero di utenti concorrenti. Nei diversi scenari si osserva che il servizio continua a funzionare in modo stabile fino a 100 utenti concorrenti, mentre nei test con 200 utenti emergono segnali più evidenti di saturazione, come l'aumento della latenza media e la comparsa di errori. Questo comportamento indica che il sistema riesce a gestire correttamente carichi

moderati, ma può entrare in una zona di maggiore instabilità quando il numero di richieste simultanee diventa molto elevato, soprattutto negli scenari più complessi.

I test evidenziano inoltre l'influenza del ramp-up sulle prestazioni del sistema. In più casi, a parità di numero di utenti concorrenti, un aumento del tempo di ramp-up ha portato a una riduzione sia del tempo medio di risposta sia della percentuale di errore. Questo suggerisce che il sistema sia sensibile non solo al volume di richieste concorrenti, ma anche alla velocità con cui il carico viene introdotto: un ramp-up più graduale consente al sistema di stabilizzarsi progressivamente, riducendo la probabilità di errori.

Per quanto riguarda i possibili colli di bottiglia, i risultati consentono di formulare alcune ipotesi interpretative in linea con la struttura implementativa del servizio. Il principale fattore di costo prestazionale appare riconducibile alla moltiplicazione delle chiamate verso il sistema documentale e alle operazioni di aggregazione dei file lato backend: il degrado si manifesta in modo proporzionale alla complessità dell'operazione richiesta, indipendentemente dal livello di carico applicato. Un'analisi infrastrutturale tramite strumenti di monitoraggio dedicati potrebbe in futuro quantificare con maggiore precisione il contributo dei singoli componenti, rappresentando una naturale estensione del presente lavoro.

Nel complesso, la valutazione sperimentale indica che l'architettura implementata risulta adeguata a supportare scenari di utilizzo realistici, in particolare per operazioni di download singolo o multiplo di dimensione moderata, mentre le prestazioni degradano progressivamente quando il carico concorrente e la complessità dell'operazione crescono simultaneamente.

Capitolo 7

Conclusioni e sviluppi futuri

7.1 Sintesi del lavoro svolto

Il presente lavoro di tesi ha avuto come oggetto la progettazione, l'implementazione e la valutazione sperimentale di un sistema a microservizi per la gestione documentale bancaria, sviluppato nell'ambito del progetto Archivio Digitale. Il sistema nasce in risposta a un'esigenza concreta di business e di compliance normativa: garantire a tutti i clienti titolari di servizi di investimento l'accesso digitale alla propria documentazione, in conformità con le indicazioni introdotte dalla Direttiva (UE) 2021/338 (MiFID II "Quick Fix") e con il principio del digital-by-default da essa promosso.

Il percorso seguito nel lavoro ha coperto l'intero ciclo di vita del sistema, dall'analisi dei requisiti fino alla valutazione sperimentale delle prestazioni. In una prima fase è stato definito il perimetro del sistema, identificando gli attori coinvolti, le regole di eleggibilità e i requisiti funzionali e non funzionali. È stata quindi descritta l'architettura del sistema, analizzando l'evoluzione dallo scenario AS-IS, in cui la consultazione documentale era vincolata al canale di Home Banking, allo scenario TO-BE, in cui l'Archivio Digitale si affianca come canale alternativo e autonomo. La modellazione architetturale è stata condotta mediante il C4 Model, che ha consentito di rappresentare in modo strutturato e progressivo il sistema a diversi livelli di astrazione.

Nella fase di implementazione, l'attenzione si è concentrata sul servizio applicativo PostaPersonaleService, scelto come caso di studio rappresentativo dell'architettura backend dell'Archivio Digitale. L'analisi ha evidenziato come il servizio non si limiti a svolgere un ruolo di proxy verso il sistema documentale esterno, ma assuma una funzione attiva di orchestrazione: coordina le chiamate verso i sistemi esterni, gestisce il recupero dei contenuti binari e, nel caso di richieste multiple, aggrega i risultati generando dinamicamente

un archivio ZIP lato backend. Questa scelta progettuale, in linea con il modello architetturale adottato, consente di mantenere il sistema documentale isolato da logiche di composizione e presentazione dei contenuti, preservando la separazione delle responsabilità tra i diversi livelli del sistema.

Infine, la valutazione sperimentale ha consentito di analizzare il comportamento del sistema in condizioni di carico crescente, attraverso una campagna di test condotta con Apache JMeter e strutturata secondo la metodologia Goal-Question-Metric. I test sono stati articolati in tre scenari principali, differenziati per numero di documenti richiesti, e hanno permesso di osservare l'evoluzione delle prestazioni del sistema al variare del carico applicativo e della complessità delle operazioni.

7.2 Valutazione critica dei risultati ottenuti

I risultati ottenuti consentono di formulare alcune considerazioni conclusive sul sistema realizzato e sull'efficacia delle scelte progettuali adottate, sia dal punto di vista architetturale sia dal punto di vista prestazionale.

Dal punto di vista architetturale, l'introduzione dell'Archivio Digitale ha permesso di superare i limiti dello scenario AS-IS, ampliando l'accessibilità del servizio documentale e riducendo la dipendenza esclusiva dal canale di Home Banking. La scomposizione del sistema nei livelli di frontend, Backend for Frontend e backend ha consentito di separare in modo esplicito le responsabilità di presentazione, orchestrazione dei flussi e logica di dominio, favorendo la modularità del sistema e la possibilità di evolverne i componenti in modo indipendente. L'adozione del framework proprietario aziendale ha garantito coerenza strutturale tra i diversi servizi, standardizzando le convenzioni implementative e riducendo la variabilità del codice, con effetti positivi sulla manutenibilità del sistema e sulla sua comprensibilità da parte di nuovi sviluppatori.

Dal punto di vista della valutazione sperimentale, i risultati ottenuti hanno confermato che il sistema è in grado di gestire in modo efficace le operazioni di download singolo e i download multipli di dimensione moderata. Nello scenario S1, relativo al download di un singolo documento, il sistema ha mantenuto tempi di risposta medi nell'ordine delle poche centinaia di millisecondi anche con carichi elevati, con un tasso di errore complessivamente molto contenuto, a conferma della buona stabilità del servizio nelle condizioni operative più comuni. Negli scenari di download multiplo, invece, si è osservato un incremento progressivo della latenza media e del tasso di errore all'aumentare del numero di documenti richiesti e del numero di utenti concorrenti, con un degrado più marcato nello scenario S3, relativo al download

di 20 documenti, in cui il tasso di errore ha raggiunto il 32% nelle condizioni di carico più elevate.

Questi risultati sono coerenti con la struttura implementativa del servizio: il principale fattore di costo prestazionale è riconducibile al numero di chiamate sequenziali verso il sistema documentale esterno e alle operazioni di aggregazione dei contenuti lato backend, come confermato dall'andamento proporzionale del degrado al crescere della complessità dell'operazione. È inoltre emerso che una crescita più graduale del carico, regolata tramite il parametro di ramp-up, consente al sistema di distribuire meglio le richieste, riducendo sia la latenza media sia la probabilità di errori. Nel complesso, la valutazione sperimentale ha fornito un riscontro concreto sull'efficacia delle scelte architetturali adottate e ha permesso di identificare con chiarezza le condizioni operative in cui le prestazioni tendono a degradare in modo più significativo.

7.3 Possibili evoluzioni e sviluppi futuri

La valutazione sperimentale ha messo in luce alcune direzioni concrete di miglioramento del sistema.

Un primo ambito di intervento riguarda l'ottimizzazione del flusso di download multiplo. L'attuale implementazione prevede che ogni documento venga recuperato tramite una chiamata sequenziale e dedicata verso il sistema di gestione documentale. L'introduzione di un meccanismo di recupero parallelo, basato su chiamate concorrenti verso il sistema documentale, potrebbe ridurre significativamente i tempi di risposta nelle operazioni di download multiplo, limitando l'impatto dell'attesa accumulata nelle chiamate sequenziali. Questa ottimizzazione risulterebbe particolarmente efficace negli scenari con un numero elevato di documenti richiesti, come lo scenario S3 analizzato nella sperimentazione, in cui il costo delle chiamate sequenziali rappresenta il principale fattore di degrado delle prestazioni.

Un secondo sviluppo riguarda l'introduzione di un livello di caching per i contenuti documentali più frequentemente consultati. In un sistema in cui la documentazione è prevalentemente statica nel tempo, la possibilità di memorizzare temporaneamente i contenuti già recuperati consentirebbe di ridurre il numero di chiamate verso il sistema documentale esterno, alleggerendo il carico sull'infrastruttura di integrazione e migliorando i tempi di risposta per le operazioni più ricorrenti.

Nel complesso, il lavoro svolto dimostra come l'analisi prestazionale rappresenti uno strumento fondamentale per comprendere il comportamento di sistemi software complessi e per validare le scelte progettuali adottate.

Bibliografia

- [1] *Directive (EU) 2021/338 of the European Parliament and of the Council of 16 February 2021*. EUR-Lex. 2021. URL: <https://eur-lex.europa.eu/eli/dir/2021/338/>.
- [2] Simon Brown. *The C4 model for visualising software architecture*. 2018. URL: <https://c4model.com/>.
- [3] *ISO/IEC/IEEE 42010:2011 – Systems and software engineering: Architecture description*. Geneva: International Organization for Standardization, 2011. URL: <https://www.iso.org/standard/50508.html>.
- [4] Len Bass, Paul Clements e Rick Kazman. *Software Architecture in Practice*. 3^a ed. Addison-Wesley, 2012.
- [5] Martin Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley, 2002.
- [6] Craig Walls. *Spring in Action*. 6^a ed. Manning Publications, 2022.
- [7] Roy Thomas Fielding. «Architectural Styles and the Design of Network-based Software Architectures». Tesi di dott. University of California, Irvine, 2000. URL: <https://roy.gbiv.com/pubs/dissertation/top.htm>.
- [8] Google LLC. *Introduction to Angular*. 2024. URL: <https://angular.dev/overview>.
- [9] Sam Newman. *Building Microservices: Designing Fine-Grained Systems*. 2^a ed. O'Reilly Media, 2021.
- [10] Robert C. Martin. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall, 2017.
- [11] Eclipse Foundation. *Jakarta Persistence 3.1 Specification*. 2022. URL: <https://jakarta.ee/specifications/persistence/3.1/>.

- [12] Edsger W. Dijkstra. «On the role of scientific thought». In: *Selected Writings on Computing: A Personal Perspective*. Springer, 1982, pp. 60–66.
- [13] Dick Hardt. *The OAuth 2.0 Authorization Framework*. RFC 6749. Internet Engineering Task Force, 2012. URL: <https://datatracker.ietf.org/doc/html/rfc6749>.
- [14] Michael Jones, John Bradley e Nat Sakimura. *JSON Web Token (JWT)*. RFC 7519. Internet Engineering Task Force, 2015. URL: <https://datatracker.ietf.org/doc/html/rfc7519>.
- [15] OpenAPI Initiative. *OpenAPI Specification v3.0*. 2017. URL: <https://spec.openapis.org/oas/v3.0.0>.
- [16] Victor R. Basili, Gianluigi Caldiera e H. Dieter Rombach. «The Goal Question Metric Approach». In: *Encyclopedia of Software Engineering*. Wiley, 1994.
- [17] Victor R. Basili e David M. Weiss. «A Methodology for Collecting Valid Software Engineering Data». In: *IEEE Transactions on Software Engineering* 10.6 (1984), pp. 728–738.
- [18] Raj Jain. *The Art of Computer Systems Performance Analysis*. Wiley, 1991.
- [19] Elaine Weyuker e Filippos Vokolos. «Experience with Performance Testing of Software Systems: Issues, an Approach, and Case Study». In: *IEEE Transactions on Software Engineering* 26.12 (2000), pp. 1147–1156.