

Università degli Studi di Padova

---

Dipartimento di Matematica "Tullio Levi-Civita"

Corso di Laurea in Informatica

# Un Portale per la Gestione delle Buste Paga

*Relatore*  
Baldan Paolo

*Laureando*  
Carnuccio Mauro  
*Matricola* 1216741

---

ANNO ACCADEMICO 2022/2023

Carnuccio Mauro: *Un Portale per la Gestione delle Buste Paga*, Laurea triennale,  
© Settembre 2023.

# Sommario

Il presente documento ha lo scopo di descrivere l'attività di stage svolta presso l'azienda ConsoftInformatica s.r.l. con sede a Padova nel periodo che va dal 12 Giugno al 1 Settembre 2023.

L'obiettivo dello stage era la creazione in .NET di un'applicazione Web usufruibile tramite intranet aziendale che permettesse a degli operatori selezionati di gestire i documenti delle buste paga dei dipendenti e di inviarli tramite email.

In più, l'applicazione avrebbe dovuto permettere la gestione dei dati dei dipendenti e dei template per il corpo delle email con le buste paga da inviare, oltre che la visualizzazione dello storico degli invii.

# Ringraziamenti

*In primo luogo vorrei ringraziare il Professor Paolo Baldan, relatore della mia tesi, che mi ha aiutato sia durante il periodo di stage che durante la stesura della tesi.*

*Ringrazio con affetto la mia famiglia per il supporto, il sostegno e i consigli che mi hanno offerto.*

*Desidero ringraziare tutti i miei amici, in particolare Fabio, e i colleghi di università con cui ho stretto un legame speciale e con cui ho passato momenti meravigliosi.*

*Un ringraziamento speciale va alla mia fidanzata Elena, che è stata sempre al mio fianco e non mi ha mai fatto mancare l'affetto di cui avevo bisogno anche nei periodi difficili.*

*Padova, Settembre 2023*

Mauro Carnuccio



# Indice

<b>Sommario</b>	<b>ii</b>
<b>Ringraziamenti</b>	<b>iii</b>
<b>1 Introduzione</b>	<b>1</b>
1.1 Azienda . . . . .	1
1.2 L'idea . . . . .	1
1.3 Principali problematiche . . . . .	2
1.4 La soluzione . . . . .	2
1.5 Strumenti utilizzati . . . . .	2
1.5.1 .NET Core . . . . .	3
1.5.2 Blazor WebAssembly . . . . .	3
1.5.3 GitLab . . . . .	4
1.5.4 iText 7 . . . . .	4
1.5.5 PostgreSQL . . . . .	4
1.6 Risultato . . . . .	5
1.7 Struttura del documento . . . . .	6
<b>2 Progetto</b>	<b>7</b>
2.1 Descrizione del progetto . . . . .	7
2.2 Pianificazione . . . . .	7
2.3 Obiettivi . . . . .	8
2.3.1 Obbligatorî . . . . .	8
2.3.2 Desiderabili . . . . .	8
2.3.3 Opzionali . . . . .	9
2.4 Tecnologie utilizzate . . . . .	9
2.4.1 Linguaggi . . . . .	9
2.4.2 Framework . . . . .	9
2.4.3 Librerie . . . . .	10
2.4.4 Strumenti . . . . .	10
<b>3 Analisi dei requisiti</b>	<b>11</b>
3.1 Dominio . . . . .	11
3.1.1 Utenti . . . . .	11
3.1.2 Funzionalità . . . . .	11
3.2 Casi d'uso . . . . .	12
3.2.1 Classificazione dei casi d'uso . . . . .	12
3.2.2 Descrizione della struttura dei casi d'uso . . . . .	12

3.2.3	UC-1 - Autenticazione . . . . .	12
3.2.4	UC-2 - Caricamento file PDF . . . . .	12
3.2.5	UC-3 - Visualizzazione della lista di file PDF . . . . .	13
3.2.6	UC-4 - Filtraggio della lista di file . . . . .	13
3.2.7	UC-5 - Invio di un'email . . . . .	13
3.2.8	UC-6 - Creazione di un nuovo dipendente . . . . .	13
3.2.9	UC-7 - Modifica di un dipendente esistente . . . . .	14
3.2.10	UC-8 - Eliminazione di un dipendente esistente . . . . .	14
3.2.11	UC-9 - Dimissione di un dipendente . . . . .	14
3.2.12	UC-10 - Visualizzazione della lista di dipendenti . . . . .	14
3.2.13	UC-11 - Creazione di un nuovo template email . . . . .	15
3.2.14	UC-12 - Modifica di un template email esistente . . . . .	15
3.2.15	UC-13 - Cancellazione di un template email esistente . . . . .	15
3.2.16	UC-14 - Visualizzazione della lista dei template email . . . . .	15
3.2.17	UC-15 - Attivazione di un template . . . . .	16
3.2.18	UC-16 - Visualizzazione dello storico degli invii . . . . .	16
3.3	Requisiti . . . . .	16
3.3.1	Classificazione dei requisiti . . . . .	16
3.3.2	Requisiti funzionali . . . . .	17
3.3.3	Requisiti di qualità . . . . .	18
3.3.4	Requisiti di vincolo . . . . .	19
<b>4</b>	<b>Progettazione</b>	<b>20</b>
4.1	Database . . . . .	20
4.1.1	Legenda . . . . .	21
4.1.2	Documents . . . . .	21
4.1.3	Employees . . . . .	22
4.1.4	MailTemplates . . . . .	22
4.1.5	MailLog . . . . .	22
4.2	Back-end . . . . .	23
4.2.1	Lista API . . . . .	23
4.2.2	Diagramma UML delle classi . . . . .	26
4.3	Front-end . . . . .	29
<b>5</b>	<b>Realizzazione</b>	<b>34</b>
5.1	Homepage . . . . .	34
5.2	Barra di navigazione con accesso non eseguito . . . . .	35
5.3	Barra di navigazione con accesso eseguito . . . . .	35
5.4	Login . . . . .	36
5.5	Logout . . . . .	36
5.6	Gestione documenti . . . . .	37
5.6.1	Caricamento del file PDF . . . . .	37
5.6.2	Ricerca dei file PDF caricati . . . . .	38
5.6.3	Visualizzazione dei file PDF caricati . . . . .	38
5.7	Gestione dipendenti . . . . .	39
5.7.1	Nuovo dipendente . . . . .	40
5.7.2	Lista dipendenti . . . . .	40
5.7.3	Modifica dipendente . . . . .	41
5.7.4	Cancellazione dipendente . . . . .	42
5.8	Gestione template email . . . . .	42

<i>INDICE</i>	vii
5.8.1 Nuovo template . . . . .	43
5.8.2 Lista template . . . . .	43
5.8.3 Modifica template . . . . .	44
5.8.4 Placeholder . . . . .	45
5.8.5 Cancellazione template . . . . .	46
5.9 Visualizzazione storico invii . . . . .	46
<b>6 Conclusioni</b>	<b>48</b>
6.1 Valutazione degli strumenti utilizzati . . . . .	48
6.2 Possibili estensioni . . . . .	48
6.3 Valutazione personale . . . . .	49
<b>Riferimenti</b>	<b>51</b>
<b>Acronimi</b>	<b>53</b>
<b>Glossario</b>	<b>54</b>



# Elenco delle figure

1.1	Logo Consoft	1
1.2	Logo .NET Core	3
1.3	Logo Blazor	3
1.4	Logo GitLab	4
1.5	Logo iText 7	4
1.6	Logo PostgreSQL	4
4.1	Progettazione del database	20
4.2	Progettazione delle classi del back-end	26
4.3	Esempio di controller e service	27
4.4	Classi DTO	28
4.5	Classi Response	29
4.6	Progettazione della pagina di gestione documenti	30
4.7	Progettazione della pagina di gestione dipendenti	31
4.8	Progettazione della pagina di gestione dipendenti	32
4.9	Progettazione pagina di visualizzazione dello storico degli invii	32
4.10	Progettazione della barra di navigazione utente non autenticato	33
4.11	Progettazione della barra di navigazione dell'Amministratore	33
4.12	Progettazione della barra di navigazione dell'Operatore	33
4.13	Progettazione logout	33
5.1	Homepage	34
5.2	Barra di navigazione con accesso non eseguito	35
5.3	Barra di navigazione dell'Amministratore	35
5.4	Barra di navigazione dell'Operatore	35
5.5	Tasto di login	36
5.6	Tasto di logout	36
5.7	Pagina di gestione dei documenti	37
5.8	Sezione per il caricamento del file PDF	37
5.9	Sezione per il filtraggio dei documenti	38
5.10	Lista dei documenti	38
5.11	Selettore di pagina	39
5.12	Pagina di gestione dei dipendenti	39
5.13	Popup per la creazione di un nuovo dipendente	40
5.14	Lista dei dipendenti	40
5.15	Popup modifica del dipendente	41
5.16	Richiesta di conferma della cancellazione del dipendente	42
5.17	Pagina di gestione dei template	42

5.18	Popup di creazione di un nuovo template . . . . .	43
5.19	Lista dei template . . . . .	43
5.20	Popup di modifica del template . . . . .	44
5.21	Tooltip per i placeholder disponibili . . . . .	45
5.22	Richiesta di conferma della cancellazione del template . . . . .	46
5.23	Pagina di visualizzazione dello storico degli invii . . . . .	46

# Elenco delle tabelle

2.1	Ripartizione delle ore . . . . .	7
2.3	Obiettivi obbligatori . . . . .	8
2.4	Obiettivi desiderabili . . . . .	9
2.5	Obiettivi opzionali . . . . .	9
2.6	Lista dei linguaggi utilizzati . . . . .	9
2.7	Lista dei framework utilizzati . . . . .	9
2.8	Lista delle librerie utilizzate . . . . .	10
2.9	Lista degli strumenti utilizzati . . . . .	10
3.1	Tabella dei requisiti funzionali obbligatori . . . . .	17
3.2	Tabella dei requisiti funzionali facoltativi . . . . .	18
3.3	Tabella dei requisiti di qualità obbligatori . . . . .	18
3.4	Tabella dei requisiti di vincolo obbligatori . . . . .	19



# Capitolo 1

## Introduzione

### 1.1 Azienda

ConsoftInformatica, attiva dal 2006, opera su tutto il territorio nazionale nel settore della consulenza e dello sviluppo **software** con sedi a Roma, Milano, Napoli e Padova.



Figura 1.1: Logo Consoft

L'azienda sostiene il percorso di Digital Transformation dei propri clienti, realizzando soluzioni applicative personalizzate ed erogando servizi di consulenza professionale di qualità attraverso competenze specialistiche e partnership con aziende leader di mercato.

Presso le software factory dislocate su Roma e Padova, i vari gruppi di sviluppo hanno la possibilità di lavorare e sperimentare, come veri e propri laboratori software, le più innovative tecnologie per la realizzazione di prodotti ad hoc per i propri clienti o da immettere sul mercato.

<https://consoftinformatica.it/>

### 1.2 L'idea

L'idea di progetto proposta da ConsoftInformatica era quella di creare un'applicazione Web che permettesse agli utenti autorizzati di gestire le **buste paga** dei dipendenti e di caricarle e inviarle loro per **email** tramite pochi click, in maniera facile e automatica. L'**accesso** all'applicazione deve avvenire tramite lo stesso account **aziendale** che gli utenti addetti all'utilizzo dell'applicazione usano abitualmente, così che non ci sia la necessità di effettuare nuove registrazioni, ma sia tutto gestito da Azure Active Directory[1].

Dato che un'altra funzionalità deve essere quella di gestione dei **template** email, non risulterà necessario che l'utente scriva ogni volta il contenuto della mail da inviare. Questi template dovranno essere formati dall'oggetto e dal corpo della mail che può contenere eventualmente anche dei **placeholder** con le informazioni del dipendente che la riceve.

L'esito di ciascun invio, positivo o negativo che sia, deve essere sempre salvato all'interno di uno **storico degli invii** con relativa descrizione nel caso in cui l'esito sia negativo. Salvando questo storico sarà sempre possibile andare a verificare se tutte le mail sono state inviate o se ci sono stati problemi di vario genere.

### 1.3 Principali problematiche

La principale problematica da risolvere in questo progetto è stata la ricerca e **suddivisione** delle buste paga dal documento caricato. Questo perché i documenti PDF caricati non contengono ciascuno una busta paga, ma sono un "plico unico" contenente le buste paga di tutti i dipendenti.

Data questa struttura del documento, risulta necessario dover costruire un algoritmo che permetta di individuare, suddividere e salvare le varie buste paga trovate all'interno del plico unico in documenti separati che possano essere allegati alle mail.

Un'altra problematica importante è stata quella della progettazione e realizzazione della **base di dati** per salvare tutte le informazioni sui dipendenti, sui documenti, sui template e sullo storico.

### 1.4 La soluzione

Per la ricerca all'interno del documento e la sua suddivisione, ho deciso di utilizzare la libreria `iText7` [2] per leggere e dividere il file PDF. Ho utilizzato un'espressione **regolare** per andare a ricercare il codice fiscale del dipendente all'interno di ogni pagina. In questo modo l'algoritmo può individuare quali pagine si riferiscono ad un certo dipendente e, di conseguenza, metterle insieme creando un nuovo documento PDF contenente solo le pagine della busta paga di quel dipendente specifico.

Per quanto riguarda invece la progettazione della base di dati ho utilizzato `StarUML`[3], così da avere un diagramma dettagliato della struttura per poterne scrivere in seguito il codice **SQL** per la sua effettiva creazione (spiegato nel dettaglio al Capitolo 4.1 - Progettazione Database).

In seguito ho realizzato una progettazione dell'interfaccia grafica che andasse a soddisfare tutte le funzionalità richieste (spiegato nel dettaglio al Capitolo 4.3 - Progettazione Front-end)

### 1.5 Strumenti utilizzati

Le tecnologie che ho utilizzato per lo sviluppo di questo progetto sono state principalmente proposte da `ConsoftInformatica`, le più importanti delle quali sono riportate qui di seguito.

### 1.5.1 .NET Core



Figura 1.2: Logo .NET Core

.NET Core [4] è un **Framework** libero e open source di Microsoft per la creazione di applicazioni desktop. L'ho utilizzato per la creazione delle Web API del back-end dell'applicazione.

### 1.5.2 Blazor WebAssembly



Figura 1.3: Logo Blazor

Blazor WebAssembly [5] è un **Framework** di applicazioni a pagina singola per la creazione di app Web interattive lato client con .NET. Usa standard Web aperti senza la necessità di ricompilare il codice in altri linguaggi e funziona su tutti i browser moderni, sia per desktop che per dispositivi mobili.

Il WebAssembly è un formato compatto ottimizzato per un download e un'esecuzione veloci. [5]

### 1.5.3 GitLab



Figura 1.4: Logo GitLab

GitLab [6] è una piattaforma Web che permette di gestire repository Git, fornendo strumenti per il controllo di versione e per la gestione dei progetti.

### 1.5.4 iText 7



Figura 1.5: Logo iText 7

La libreria iText 7 [2] è la versione di iText che sostituisce iTextSharp. Questa libreria è disponibile sia per Java che per C# e permette di creare e manipolare file PDF.

### 1.5.5 PostgreSQL

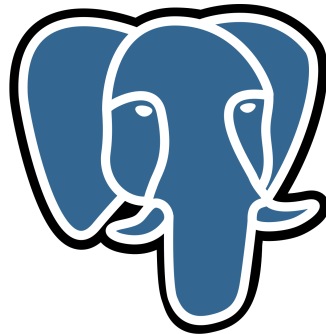


Figura 1.6: Logo PostgreSQL

PostgreSQL [7] è uno dei database relazionali più popolari al mondo. Le sue caratteristiche principali sono:

- **Scalabilità:** è in grado di gestire grandi quantità di dati e di distribuire i dati su più server;



- **Affidabilità:** è progettato per essere affidabile e resiliente, con meccanismi di recupero in caso di errore;
- **Performance:** è in grado di gestire un elevato volume di richieste, mantenendo le prestazioni elevate;
- **Estensibilità:** è altamente estensibile, con un'ampia gamma di moduli e funzioni disponibili.

## 1.6 Risultato

A seguito dello sviluppo il **front-end** risulta composto di 4 pagine Web:

- **Homepage:** semplice pagina di benvenuto dell'applicazione;
- **Gestione dei documenti:** pagina che permette di caricare i documenti PDF con le buste paga, visualizzarle e mandare le relative mail;
- **Gestione dei dipendenti:** pagina che permette di gestire i dipendenti salvati all'interno dell'applicazione aggiungendoli, modificandoli o cancellandoli;
- **Gestione dei template:** pagina che permette di gestire i template delle mail aggiungendone di nuovi, modificandoli, cancellandoli o selezionando quale rendere attivo;
- **Email Logs:** pagina che permette di visualizzare lo storico degli invii.

Il **database** prodotto è formato da 4 tabelle:

- **Employees:** tabella contenente le informazioni dell'utente necessarie all'applicazione per svolgere i suoi compiti;
- **Documents:** tabella contenente le informazioni di tutti i documenti salvati;
- **MailTemplates:** tabella contenente oggetto e corpo del template della mail, oltre che il suo attuale stato (*Attivo/Non Attivo*);
- **MailLog:** tabella contenente lo storico degli invii con tutte le descrizioni degli invii falliti.

Il **back-end** è composto principalmente da Controller che gestiscono le chiamate API, mentre la logica principale dell'applicazione risiede nella classe `PdfManipulator` che permette la suddivisione del file PDF.

## 1.7 Struttura del documento

- **Capitolo 2 - Progetto:** in questo capitolo si descrive il progetto con tutte le funzionalità richieste e i relativi obiettivi da raggiungere, oltre ad una lista completa delle tecnologie utilizzate.
- **Capitolo 3 - Analisi dei Requisiti:** in questo capitolo tecnico si formalizzano gli obiettivi e si stila una lista di requisiti con il loro relativo stato di completamento.
- **Capitolo 4 - Progettazione:** in questo capitolo verrà presentata la progettazione che è stata eseguita prima della codifica del progetto. Sono descritte tutte le sue parti: base di dati, back-end, front-end.
- **Capitolo 5 - Realizzazione:** in questo capitolo si illustreranno tutte le pagine nel loro stato finale di completamento con una spiegazione sul funzionamento di ogni sezione.
- **Capitolo 6 - Conclusioni:** in questo capitolo verranno descritte le conclusioni tratte a fine tirocinio con le possibili estensioni dell'applicazione, la valutazione delle tecnologie e un'opinione conclusiva.

## Capitolo 2

# Progetto

### 2.1 Descrizione del progetto

Il prodotto da sviluppare consisteva in una applicazione Web usufruibile tramite intranet aziendale che permettesse di:

- caricare un documento in formato **PDF** contenente tutte le buste paga dei dipendenti dell'azienda così da poterlo dividere automaticamente in tanti file quante sono le buste paga al suo interno;
- inviare mail ai dipendenti con la relativa busta paga allegata;
- gestire localmente i dati dei dipendenti;
- gestire i **template** del corpo delle mail;
- visualizzare lo storico degli invii.

### 2.2 Pianificazione

Per il conseguimento degli obiettivi del progetto di stage, sono state pianificate 320 ore di lavoro, divise in 8 ore al giorno per 5 giorni a settimana, per un totale di 40 ore settimanali.

Lo stage è iniziato il 12/06/2023 e si è concluso il 29/08/2023.

La ripartizione delle ore nel Piano di Lavoro redatto da ConsoftInformatica è stata pianificata come segue:

Tabella 2.1: Ripartizione delle ore

Durata in ore	Descrizione attività
60	Formazione: <ul style="list-style-type: none"><li>• <b>Framework</b> .NET Core 6/8 [4]</li><li>• Blazor [8] Web Assembly</li><li>• Docker[9]</li><li>• Approfondimento sui message broker</li></ul>

Durata in ore	Descrizione attività
40	Analisi: <ul style="list-style-type: none"> <li>• Analisi su elenco utenze su Active Directory [1] e arricchimento dati</li> <li>• Analisi suddivisione PDF unico</li> <li>• Analisi servizi MS Graph [10]</li> </ul>
200	Progettazione e Sviluppo: <ul style="list-style-type: none"> <li>• Progettazione della base dati e dell'architettura applicativa</li> <li>• Realizzazione della base dati</li> <li>• Realizzazione dell'applicativo front-end</li> <li>• Realizzazione dei servizi API REST back-end</li> </ul>
20	Test e Documentazione: <ul style="list-style-type: none"> <li>• Test dell'applicativo</li> <li>• Manualistica utente</li> </ul>

**320 ore totali**

## 2.3 Obiettivi

Al momento della pianificazione dello stage, ConsoftInformatica ha redatto un Piano di Lavoro che descriveva gli obiettivi da raggiungere durante il periodo di stage. In base alla loro importanza, questi obiettivi sono stati suddivisi in 3 categorie: **Obbligatori**, **Desiderabili** e **Opzionali**.

### 2.3.1 Obbligatori

Tabella 2.3: Obiettivi obbligatori

Nome	Descrizione
<b>OB1</b>	Progettazione del software applicativo e della base dati applicativa
<b>OB2</b>	Realizzazione della base dati
<b>OB3</b>	Realizzazione dell'applicativo front-end
<b>OB4</b>	Realizzazione dei servizi API REST back-end

### 2.3.2 Desiderabili

Tabella 2.4: Obiettivi desiderabili

Nome	Descrizione
<b>DE1</b>	Autonomia nella ricerca e proposizione di soluzioni
<b>DE2</b>	Realizzazione coda invii mail

### 2.3.3 Opzionali

Tabella 2.5: Obiettivi opzionali

Nome	Descrizione
<b>OP1</b>	Realizzazione della componente di sincronizzazione automatica e aggiornamento della lista dipendenti

## 2.4 Tecnologie utilizzate

### 2.4.1 Linguaggi

Tabella 2.6: Lista dei linguaggi utilizzati

Nome	Versione	Descrizione
C# [11]	11	È un linguaggio di programmazione orientato agli oggetti, creato e mantenuto da <i>Microsoft</i>
CSS [12]	3	È il linguaggio di definizione di stili standard del Web mantenuto dal consorzio W3C [13]
HTML [14]	5	È il linguaggio di markup standard per la creazione di pagine web mantenuto dal consorzio W3C [13]
JavaScript [15]	ES6	È un linguaggio di scripting che permette di creare contenuto dinamico all'interno delle pagine Web

### 2.4.2 Framework

Tabella 2.7: Lista dei framework utilizzati

Nome	Versione	Descrizione
.NET Core [4]	6.0 LTS	<b>Framework</b> di <i>Microsoft</i> per la creazione di applicazioni desktop

Nome	Versione	Descrizione
Blazor [8]	Web Assembly	<b>Framework</b> per la creazione di web app in C#, creato e mantenuto da <i>Microsoft</i>

### 2.4.3 Librerie

Tabella 2.8: Lista delle librerie utilizzate

Nome	Versione	Descrizione
Bootstrap [16]	5	Libreria di componenti grafiche di pagine Web
Flurl [17]	2.4.0	Libreria client <b>HTTP</b> e costruttore di <b>URL</b>
iText [2]	7	Libreria open source per creare e manipolare file PDF
Npgsql [18]	7.0.4	Libreria per l'accesso a PostgreSQL [7]
SqlKata [19]	3.0.7	Libreria per la costruzione di <b>query</b>

### 2.4.4 Strumenti

Tabella 2.9: Lista degli strumenti utilizzati

Nome	Versione	Descrizione
PostgreSQL [7]	15.3	Database relazionale open source basato su SQL
Visual Studio [20]	Community 2022	<b>IDE</b> ufficiale per progetti di C# e .NET distribuito da <i>Microsoft</i>
Swagger [21]	2.2.15	Suite di strumenti per sviluppatori <b>API</b>
Docker [9]	4.22.1	Piattaforma che permette di costruire, condividere ed eseguire applicazioni utilizzando container
Git [22]	2.37.3.windows.1	Software di controllo di versione distribuito
GitLab [6]	16.3	Piattaforma web che permette la gestione di repository git
Microsoft Teams [23]	1.6.00.22378	Piattaforma di comunicazione e collaborazione
StarUML [3]	5.1.0	Strumento per la modellazione di sistemi in UML

## Capitolo 3

# Analisi dei requisiti

Questo capitolo espone i casi d'uso individuati per la creazione di questo progetto. La prima parte sarà dedicata ad una spiegazione atta ad illustrare le funzionalità dell'applicazione, gli utenti che vi possono accedere e le modalità di classificazione dei casi d'uso, mentre la seconda parte è dedicata all'effettiva lista di casi d'uso con le relative informazioni ed eventuali grafici.

### 3.1 Dominio

#### 3.1.1 Utenti

Le tipologie di utenti che potranno usufruire del portale sono due, ciascuna con i propri permessi:

- **Amministratore:** ha la possibilità di accedere a tutte le pagine e tutte le funzionalità dell'applicazione senza restrizioni.
- **Operatore:** può accedere solo alla pagina di gestione dei documenti, con i permessi di caricare documenti e inviarne le rispettive email, e quella di visualizzazione dello storico degli invii mail.

#### 3.1.2 Funzionalità

Le funzionalità principali che l'applicazione fornisce sono:

- Effettuare il login;
- Creare, rimuovere, modificare e leggere i dipendenti e le relative informazioni;
- Creare, rimuovere, modificare e leggere i **template** delle mail;
- Leggere lo storico degli invii con i relativi esiti e le loro descrizioni;
- Caricare i documenti contenenti le buste paga;
- Visualizzare i documenti caricati con possibilità di applicare filtri alla lista per una visualizzazione più agevole;
- Inviare email ai dipendenti con la busta paga in allegato.

## 3.2 Casi d'uso

### 3.2.1 Classificazione dei casi d'uso

Ciascun caso d'uso è classificato come segue:

UC-{codice}

- UC: *Use Case*.

Il *codice* indicato sopra è univoco per ogni caso d'uso, così che ciascuno di essi sia univocamente riconoscibile dagli altri.

Nel caso in cui un caso d'uso numero “*x*” abbia bisogno di un livello di dettaglio maggiore, il suo codice avrà il formato “*x.y*” in cui *y* è un numero crescente per ogni caso d'uso che aggiunge dettaglio a quello originale.

### 3.2.2 Descrizione della struttura dei casi d'uso

Ciascun caso d'uso viene rappresentato dalla seguente struttura:

- **Attori:** gli attori principali e secondari del caso d'uso;
- **Descrizione:** descrizione di ciò che deve accadere secondo il caso d'uso;
- **Pre-condizione:** le condizioni che devono essere soddisfatte affinché il caso d'uso possa verificarsi;
- **Post-condizione:** le condizioni che saranno soddisfatte dopo che il caso d'uso si è verificato;
- **Scenario principale:** la situazione più comune in cui può verificarsi il caso d'uso.

### 3.2.3 UC-1 - Autenticazione

- **Attori:** Utente non riconosciuto;
- **Descrizione:** il sistema permette all'utente di effettuare l'accesso tramite:
  - inserimento dell'indirizzo email aziendale (UC-1.1);
  - inserimento della password (UC-1.2);
- **Pre-condizione:** il sistema non riconosce l'utente dato che non ha ancora effettuato l'accesso;
- **Post-condizione:** il sistema ottiene le informazioni dell'utente e il relativo ruolo;
- **Scenario principale:** l'utente vuole di effettuare l'accesso per poter usufruire delle funzionalità del sistema.

### 3.2.4 UC-2 - Caricamento file PDF

- **Attori:** Operatore, Amministratore;
- **Descrizione:** il sistema permette all'utente di caricare un file in formato PDF contenente le buste paga dei dipendenti. Per farlo devono essere inseriti:
  - mese (UC-2.1);
  - anno (UC-2.2);
  - file PDF (UC-2.3);
- **Pre-condizione:** l'utente autenticato richiede di caricare un file in formato PDF;
- **Post-condizione:** il sistema ha caricato il file PDF e l'ha diviso in tanti file, uno per ogni busta paga contenuta al suo interno;



- **Scenario principale:** l'utente vuole caricare un file PDF contenente le buste paga dei dipendenti in modo che sia automaticamente diviso in tanti file quante sono le buste paga contenute al suo interno.

### 3.2.5 UC-3 - Visualizzazione della lista di file PDF

- **Attori:** Operatore, Amministratore;
- **Descrizione:** il sistema permette all'utente di visualizzare la lista di file che sono presenti nel *server*. Per farlo devono essere inseriti:
  - mese (UC-3.1);
  - anno (UC-3.2);
- **Pre-condizione:** l'utente autenticato richiede la visualizzazione della lista delle buste paga di un certo periodo (mese e anno);
- **Post-condizione:** il sistema fornisce all'utente la lista di buste paga del periodo dato;
- **Scenario principale:** l'utente vuole visualizzare la lista delle buste paga di un certo periodo dato.

### 3.2.6 UC-4 - Filtraggio della lista di file

- **Attori:** Operatore, Amministratore;
- **Descrizione:** il sistema permette di filtrare per nome la lista di file del periodo richiesto tramite:
  - inserimento della stringa di testo secondo cui filtrare (UC-4.1);
- **Pre-condizione:** l'utente autenticato sta visualizzando la lista di file e ne richiede il filtraggio per nome;
- **Post-condizione:** il sistema fornisce all'utente la lista filtrata di buste paga;
- **Scenario principale:** l'utente vuole filtrare la lista delle buste paga per avere una visualizzazione più specifica secondo le sue esigenze.

### 3.2.7 UC-5 - Invio di un'email

- **Attori:** Operatore, Amministratore;
- **Descrizione:** il sistema permette all'utente di inviare ad un dipendente una mail contenente come corpo il template attivo e come allegato il file PDF indicato;
- **Pre-condizione:** l'utente autenticato richiede l'invio tramite email di un documento. In più:
  - il documento deve essere associato ad un dipendente;
  - deve essere presente un template attivo;
- **Post-condizione:** il sistema invia l'email al dipendente con template e documento indicati;
- **Scenario principale:** l'utente vuole inviare una email al dipendente per recapitargli la busta paga.

### 3.2.8 UC-6 - Creazione di un nuovo dipendente

- **Attori:** Amministratore;
- **Descrizione:** il sistema permette all'utente di aggiungere un nuovo dipendente tramite:
  - inserimento del nome(UC-5.1);

- inserimento del cognome(UC-5.2);
- inserimento del codice fiscale(UC-5.3);
- inserimento dell'indirizzo email principale(UC-5.4);
- inserimento dell'indirizzo email secondario(UC-5.5).
- **Pre-condizione:** l'Amministratore richiede la creazione di un nuovo dipendente;
- **Post-condizione:** il sistema ha aggiunto il nuovo dipendente con i dati forniti;
- **Scenario principale:** l'Amministratore vuole inserire i dati di un nuovo dipendente.

### 3.2.9 UC-7 - Modifica di un dipendente esistente

- **Attori:** Amministratore;
- **Descrizione:** il sistema permette all'Amministratore di modificare i dati di un dipendente esistente tramite:
  - modifica del nome(UC-6.1);
  - modifica del cognome(UC-6.2);
  - modifica del codice fiscale(UC-6.3);
  - modifica dell'indirizzo email principale(UC-6.4);
  - modifica dell'indirizzo email secondario(UC-6.5).
- **Pre-condizione:** l'Amministratore richiede la modifica dei dati di un dipendente esistente;
- **Post-condizione:** il sistema ha aggiornato i dati del dipendente con i dati forniti;
- **Scenario principale:** l'Amministratore vuole modificare i dati di un dipendente esistente.

### 3.2.10 UC-8 - Eliminazione di un dipendente esistente

- **Attori:** Amministratore;
- **Descrizione:** il sistema permette all'Amministratore di eliminare un dipendente dal sistema;
- **Pre-condizione:** l'Amministratore richiede l'eliminazione di un dipendente, e il dipendente deve essere stato già inserito nel sistema;
- **Post-condizione:** il sistema elimina il dipendente e i suoi dati;
- **Scenario principale:** l'Amministratore vuole eliminare un dipendente dal sistema.

### 3.2.11 UC-9 - Dismissione di un dipendente

- **Attori:** Amministratore;
- **Descrizione:** il sistema permette all'Amministratore di dismettere un dipendente;
- **Pre-condizione:** l'Amministratore richiede la dismissione di un dipendente attivo;
- **Post-condizione:** il sistema imposta come *dimesso* il dipendente;
- **Scenario principale:** l'Amministratore richiede la dismissione di un dipendente attivo.

### 3.2.12 UC-10 - Visualizzazione della lista di dipendenti

- **Attori:** Amministratore;

- **Descrizione:** il sistema permette di visualizzare la lista di dipendenti presenti nel sistema;
- **Pre-condizione:** l'Amministratore richiede di visualizzare la lista di dipendenti;
- **Post-condizione:** il sistema fornisce all'Amministratore la lista di dipendenti presenti nel sistema;
- **Scenario principale:** l'Amministratore richiede la visualizzazione della lista dei dipendenti.

### 3.2.13 UC-11 - Creazione di un nuovo template email

- **Attori:** Amministratore;
- **Descrizione:** il sistema permette all'Amministratore di creare un nuovo template email tramite:
  - inserimento dell'oggetto dell'email (UC-11.1);
  - inserimento del corpo dell'email (UC-11.2);
- **Pre-condizione:** l'Amministratore si trova nella pagina di gestione dei template email e richiede la creazione di un nuovo template;
- **Post-condizione:** il sistema aggiunge il nuovo template con le informazioni fornite;
- **Scenario principale:** l'Amministratore vuole aggiungere un nuovo template da utilizzare per l'invio delle email.

### 3.2.14 UC-12 - Modifica di un template email esistente

- **Attori:** Amministratore;
- **Descrizione:** il sistema permette all'Amministratore di modificare un template email tramite:
  - modifica dell'oggetto dell'email (UC-12.1);
  - modifica del corpo dell'email (UC-12.2);
- **Pre-condizione:** l'Amministratore si trova nella pagina di gestione dei template e richiede la modifica di un template esistente;
- **Post-condizione:** il sistema modifica il template con le informazioni fornite;
- **Scenario principale:** l'Amministratore vuole modificare le informazioni di un template da utilizzare per l'invio delle email.

### 3.2.15 UC-13 - Cancellazione di un template email esistente

- **Attori:** Amministratore;
- **Descrizione:** il sistema permette all'Amministratore di cancellare un template email;
- **Pre-condizione:** l'Amministratore si trova nella pagina di gestione dei template e richiede la cancellazione di un template esistente;
- **Post-condizione:** il sistema cancella il template;
- **Scenario principale:** l'Amministratore vuole eliminare un template.

### 3.2.16 UC-14 - Visualizzazione della lista dei template email

- **Attori:** Amministratore;
- **Descrizione:** il sistema permette di visualizzare la lista di template presenti nel sistema;

- **Pre-condizione:** l'Amministratore si trova nella pagina di gestione dei template;
- **Post-condizione:** il sistema fornisce la lista di template email;
- **Scenario principale:** l'Amministratore vuole visualizzare la lista di template.

### 3.2.17 UC-15 - Attivazione di un template

- **Attori:** Amministratore;
- **Descrizione:** il sistema permette di impostare un template come *Attivo*;
- **Pre-condizione:** l'Amministratore si trova nella pagina di gestione dei template e richiede di impostare un template come *Attivo*;
- **Post-condizione:** il sistema imposta come *Attivo* il template selezionato e disattiva il template precedentemente *Attivo*, se presente;
- **Scenario principale:** l'Amministratore vuole impostare come *Attivo* un template.

### 3.2.18 UC-16 - Visualizzazione dello storico degli invii

- **Attori:** Operatore, Amministratore;
- **Descrizione:** il sistema permette di visualizzare lo storico degli invii;
- **Pre-condizione:** l'utente autenticato richiede di visualizzare lo storico degli invii;
- **Post-condizione:** il sistema fornisce all'utente autenticato lo storico degli invii;
- **Scenario principale:** l'utente autenticato richiede la visualizzazione dello storico degli invii.

## 3.3 Requisiti

Di seguito saranno elencati tutti i requisiti pianificati, divisi in *Requisiti Funzionali*, *Requisiti di Qualità* e *Requisiti di Vincolo*.

Per ciascun requisito è specificato il suo codice univoco identificativo, la sua descrizione e il suo stato di completamento alla fine del periodo di stage.

### 3.3.1 Classificazione dei requisiti

Ciascun requisito ha il proprio codice identificativo. Questo codice identificativo è strutturato come segue:

R{Tipo}{Importanza}-{Codice}

Il *Tipo* può avere i seguenti valori:

- F: Funzionale;
- Q: di Qualità;
- V: di Vincolo;

*Importanza* può avere i seguenti valori:

- O: Obbligatorio;
- F: Facoltativo;

Il *Codice* invece è un valore numerico univoco per ogni requisito.

Nel caso in cui un requisito con *Codice* numero “*x*” abbia bisogno di un livello di

dettaglio maggiore, il suo codice avrà formato “ $x.y$ ” in cui  $y$  è un numero crescente per ogni requisito che aggiunge dettaglio a quello originale.

### 3.3.2 Requisiti funzionali

#### 3.3.2.1 Obbligatorî

Tabella 3.1: Tabella dei requisiti funzionali obbligatori

Codice	Descrizione	Stato
RFO-1	L'utente non autenticato può eseguire il login	Completato
RFO-2	Gli utenti Amministratore e Operatore possono caricare un file PDF	Completato
RFO-2.1	Ricevuto il file PDF, il sistema deve analizzare il documento, creare un file PDF separato per ogni busta paga trovata e salvare tutti i documenti su disco	Completato
RFO-3	Gli utenti Amministratore e Operatore possono visualizzare la lista dei file PDF caricati e creati dal sistema	Completato
RFO-4	Gli utenti Amministratore e Operatore possono filtrare la lista di file tramite parole chiave	Completato
RFO-5	Gli utenti Amministratore e Operatore possono inviare per mail il documento indicato	Completato
RFO-5.1	Al momento di inviare la mail, il sistema deve includere come oggetto quello indicato dal template attivo al momento	Completato
RFO-5.2	Al momento di inviare la mail, il sistema deve includere come corpo della mail quello indicato dal template attivo al momento	Completato
RFO-5.3	Al momento di inviare la mail, il sistema deve utilizzare come mail del destinatario la mail principale se il dipendente è considerato “Attivo”, altrimenti deve utilizzare quella secondaria	Completato
RFO-6	L'utente Amministratore deve poter creare nuovi dipendenti	Completato
RFO-7	L'utente Amministratore deve poter modificare i dipendenti esistenti	Completato
RFO-8	L'utente Amministratore deve poter eliminare i dipendenti esistenti	Completato
RFO-9	L'utente Amministratore deve poter dismettere i dipendenti Attivi	Completato
RFO-10	L'utente Amministratore deve poter visualizzare i dipendenti esistenti	Completato

Codice	Descrizione	Stato
RFO-11	L'utente Amministratore deve poter creare nuovi template email	Completato
RFO-12	L'utente Amministratore deve poter modificare i template email esistenti	Completato
RFO-13	L'utente Amministratore deve poter eliminare i template email esistenti	Completato
RFO-14	L'utente Amministratore deve poter visualizzare i template email esistenti	Completato
RFO-15	L'utente Amministratore deve poter rendere attivi i template email non attivi	Completato
RFO-16	Solo un template alla volta può essere attivo	Completato
RFO-17	Il sistema deve registrare tutte le operazioni di invio delle mail	Completato
RFO-18	Gli utenti Amministratore e Operatore possono inviare per mail tutti i documenti di un certo periodo mettendo in coda automaticamente tutte le mail	Completato

### 3.3.2.2 Facoltativi

Tabella 3.2: Tabella dei requisiti funzionali facoltativi

Codice	Descrizione	Stato
RFF-1	Il sistema deve sincronizzare automaticamente la lista di dipendenti	Non Completato
RFF-2	Il sistema deve gestire asincronamente la coda di invii delle mail	Non Completato
RFF-3	Il sistema deve sincronizzare i documenti salvati in locale in una cartella OneDrive	Non Completato

### 3.3.3 Requisiti di qualità

#### 3.3.3.1 Obbligatorie

Tabella 3.3: Tabella dei requisiti di qualità obbligatori

Codice	Descrizione	Stato
RQO-1	Deve essere fornito un manuale utente che illustri l'utilizzo dell'applicazione	Completato

### 3.3.4 Requisiti di vincolo

#### 3.3.4.1 Obbligatorie

Tabella 3.4: Tabella dei requisiti di vincolo obbligatorie

Codice	Descrizione	Stato
RVO-1	L'applicativo front-end deve essere realizzato in Blazor	Completato
RVO-2	Devono essere realizzati dei servizi API REST back-end	Completato
RVO-3	Deve essere realizzata una base di dati	Completato

## Capitolo 4

# Progettazione

### 4.1 Database

Per progettare il database ho deciso di utilizzare un diagramma **ER** così da poter rappresentare l'intera struttura con una rappresentazione che fosse indipendente dal tipo di tecnologia che avrei deciso di utilizzare in seguito.

Come è possibile notare, non è presente alcuna entità che permetta di rappresentare gli utenti che possono effettuare l'accesso o meno. Questo perché tali utenti non sono gestiti tramite questo database, ma tramite Azure Active Directory [1], in cui sono memorizzati tutti gli account aziendali dei dipendenti di ConsoftInformatica.

Una volta registrata l'applicazione all'interno della piattaforma, è possibile andare a selezionare gli utenti che possono avere accesso all'applicazione e configurare uno o più ruoli da assegnare loro. In questo modo la responsabilità della memorizzazione e gestione degli utenti è demandata ad Azure e non più all'applicazione stessa.

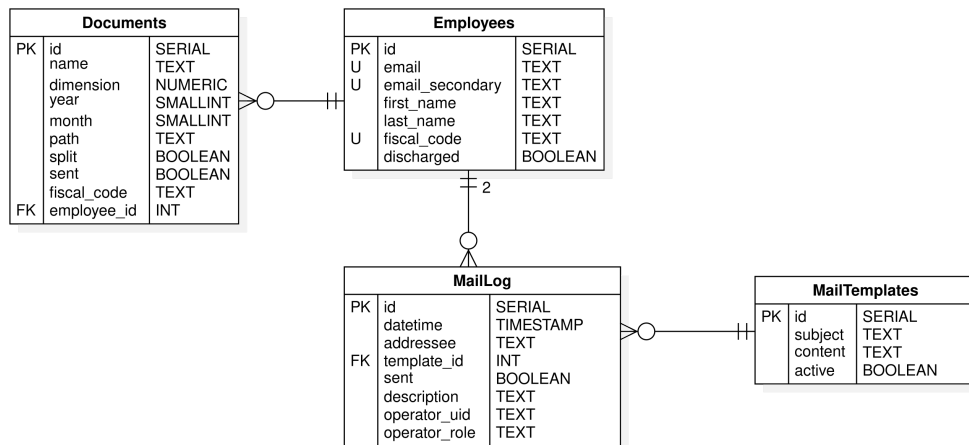


Figura 4.1: Progettazione del database



### 4.1.1 Legenda

All'interno del diagramma ER sono presenti prima di alcuni campi delle sigle che indicano alcune loro proprietà, di seguito una legenda per facilitare la comprensione di queste sigle:

- **PK: Primary Key;**
  - indica che questo campo è chiave primaria, ovvero fa parte dell'insieme di campi che contengono i valori che identificano la **tupla**;
- **FK: Foreign Key;**
  - indica che questo campo contiene solo valori presenti all'interno dei valori del campo nella tabella referenziata;
- **U: Unique;**
  - indica che il valore contenuto all'interno di un campo con questo simbolo non può ripetersi in altre tuple della tabella;

### 4.1.2 Documents

La prima entità ad essere stata progettata è stata quella che modella i documenti (Documents), ovvero i file **PDF** contenenti le buste paga dei dipendenti, dato che l'ho ritenuta una delle due entità fondamentali del progetto che sarei andato a sviluppare. Questa entità rappresenta tutte le informazioni necessarie all'applicazione per memorizzare i dati dei documenti una volta che essi verranno generati suddividendo il file originale precedentemente caricato.

I campi di questa entità sono i seguenti:

- **id: SERIAL;**
  - il codice identificativo del documento;
- **name: TEXT;**
  - il nome del file;
- **dimension: NUMERIC;**
  - il peso in KB del file;
- **year: SMALLINT;**
  - l'anno del periodo della busta paga;
- **month: SMALLINT;**
  - il mese del periodo della busta paga;
- **path: TEXT;**
  - il percorso relativo in cui è salvato il file;
- **split: BOOLEAN;**
  - valore booleano che identifica se il file è un file intero (quindi il file originale contenente tutte le buste paga) o un file già diviso (quindi un file generato dal server dividendo il file originale in tanti file per ogni busta paga presente nel file originale);
- **sent: BOOLEAN;**
  - valore booleano che identifica se il file è già stato inviato tramite email o se deve essere ancora inviato;
- **fiscal\_code: TEXT;**
  - il codice fiscale che è stato trovato all'interno del file al momento della divisione del PDF originale;
- **employee\_id: INT;**

- valore intero che rappresenta il codice identificativo del dipendente a cui è associato il documento;

### 4.1.3 Employees

La seconda entità ad essere stata progettata è stata quella che modella i dipendenti (**Employees**) dato che l'ho ritenuta una delle due entità fondamentali del progetto che sarei andato a sviluppare. Questa entità rappresenta tutte le informazioni necessarie all'applicazione per svolgere i propri compiti relativi ai dipendenti.

I campi di questa entità sono i seguenti:

- **id**: SERIAL;
  - il codice identificativo del dipendente all'interno dell'applicazione;
- **email**: TEXT;
  - l'indirizzo email principale del dipendente;
- **email\_secondary**: TEXT;
  - l'indirizzo email secondario del dipendente;
- **first\_name**: TEXT;
  - il nome del dipendente;
- **last\_name**: TEXT;
  - il cognome del dipendente;
- **fiscal\_code**: TEXT;
  - il codice fiscale del dipendente;
- **discharged**: BOOLEAN;
  - valore booleano che identifica se il dipendente è dimesso o meno;

### 4.1.4 MailTemplates

In seguito ho progettato l'entità che modella i template delle mail (**MailTemplates**). Questa è l'entità più semplice di tutte dato che deve solo contenere le informazioni minime per compilare una mail e per identificare quale sia il template attivo al momento. I campi di questa entità sono i seguenti:

- **id**: SERIAL;
  - il codice identificativo del template;
- **subject**: TEXT;
  - l'oggetto della mail;
- **content**: TEXT;
  - il corpo effettivo della mail;
- **active**: BOOLEAN;
  - valore booleano che identifica se il template è quello attivo o meno;

### 4.1.5 MailLog

Infine l'ultima entità di cui ho avuto bisogno per modellare tutte le informazioni necessarie al funzionamento dell'applicazione è quella che modella lo storico degli invii (**MailLog**). All'interno di questa entità vengono salvate le informazioni riguardo ad ogni tentato invio mail, sia che questo sia andato a buon fine, sia che questo non sia successo. Specialmente in quest'ultimo caso, infatti, è importante poter consultare uno storico per verificare quando questo fallimento ha avuto luogo e soprattutto la

motivazione.

Di seguito i campi contenuti all'interno di questa entità:

- **id**: SERIAL;
  - il codice identificativo della riga dello storico;
- **datetime**: TIMESTAMP;
  - il momento esatto in cui il tentativo di invio ha avuto luogo;
- **addressee**: TEXT;
  - il destinatario della mail;
- **template\_id**: INT;
  - il codice identificativo del template che è stato utilizzato per inviare la mail;
- **sent**: BOOLEAN;
  - valore booleano che rappresenta se l'invio della mail ha avuto successo o meno;
- **description**: TEXT;
  - testo che descrive l'esito dell'invio della mail;
- **operator\_uid**: TEXT;
  - valore identificativo dell'operatore che ha eseguito l'invio. Nello specifico è l'indirizzo email con cui l'utente ha eseguito l'accesso;
- **operator\_role**: TEXT;
  - ruolo dell'operatore che ha eseguito l'invio della mail;

## 4.2 Back-end

Per la progettazione del back-end ho deciso di utilizzare un diagramma **UML** così da avere una struttura indipendente dal linguaggio.

Ho anche creato una lista delle **Web API** che avrei dovuto sviluppare così da avere un'idea chiara delle funzionalità che il back-end avrebbe dovuto offrire.

Dato che l'applicazione necessita di un'autenticazione per eseguire qualsiasi operazione, tutte le API di seguito riportate, per essere eseguite hanno bisogno di ricevere nell'header della richiesta un **Bearer Token** così che il back-end possa verificare che chi sta richiedendo una certa operazione abbia effettivamente l'autorizzazione di eseguire tale richiesta. Riporto la lista di API qui di seguito.

### 4.2.1 Lista API

#### 4.2.1.1 Employee

- **GET** /api/v1/Employee/details/{id}
  - **Parametri**: - id: il codice identificativo del dipendente di cui si vogliono ricevere tutte le informazioni;
  - **Risposta**: un oggetto contenente tutte le informazioni relative al dipendente richiesto;
- **GET** /api/v1/Employee/getall
  - **Risposta**: un oggetto contenente la lista di tutti i dipendenti;
- **GET** /api/v1/Employee/getallpaginate
  - **Parametri**:
    - \* **page**: il numero della pagina interessata;
    - \* **perpage**: il numero di elementi per pagina;

- **Risposta:** un oggetto contenente il numero totale di pagine e la lista di dipendenti alla pagina indicata, con il numero indicato di dipendenti per pagina;
- POST /api/v1/Employee/create
  - **Corpo della richiesta:** un oggetto contenente tutte le informazioni del nuovo dipendente da creare;
  - **Risposta:** l'oggetto del dipendente appena creato;
- PUT /api/v1/Employee/edit
  - **Corpo della richiesta:** un oggetto contenente tutte le informazioni del dipendente da modificare;
  - **Risposta:** l'oggetto del dipendente appena modificato;
- DELETE /api/v1/Employee/delete/{id}
  - **Parametri:**
    - \* id: il codice identificativo del dipendente da eliminare;
  - **Risposta:** l'oggetto del dipendente appena eliminato;
- PUT /api/v1/Employee/discharge
  - **Corpo della richiesta:** un oggetto contenente tutte le informazioni del dipendente da dismettere;
  - **Risposta:** l'oggetto del dipendente appena dimesso;

#### 4.2.1.2 Log

- GET /api/v1/Log/logs/mail
  - **Parametri:**
    - \* page: il numero della pagina interessata;
    - \* perpage: il numero di elementi per pagina;
  - **Risposta:** un oggetto contenente il numero totale di pagine e la lista di storici degli invii alla pagina indicata, con il numero indicato di righe per pagina;

#### 4.2.1.3 MailTemplates

- GET /api/v1/MailTemplates/details/{id}
  - **Parametri:**
    - \* id: il codice identificativo del template di cui si vogliono ricevere le informazioni;
  - **Risposta:** un oggetto contenente le informazioni sul template;
- GET /api/v1/MailTemplates/getall
  - **Risposta:** un oggetto contenente la lista di tutti i template;
- POST /api/v1/MailTemplates/create
  - **Corpo della richiesta:** l'oggetto contenente tutte le informazioni del nuovo template da creare;
  - **Risposta:** l'oggetto del template appena creato;
- PUT /api/v1/MailTemplates/edit
  - **Corpo della richiesta:** un oggetto contenente tutte le informazioni del template da modificare;
  - **Risposta:** l'oggetto del template appena modificato;
- DELETE /api/v1/MailTemplates/delete/{id}
  - **Parametri:**
    - \* id: il codice identificativo del template da eliminare;
  - **Risposta:** l'oggetto del template appena eliminato;

- PUT /api/v1/MailTemplates/setactive/{id}
  - **Parametri:**
    - \* **id:** il codice identificativo del template da impostare come *Attivo*;
  - **Risposta:** l'oggetto del template appena impostato come attivo;

#### 4.2.1.4 Pdf

- POST /api/v1/Pdf/upload/{year}/{month}
  - **Parametri:**
    - \* **year:** l'anno del periodo della busta paga;
    - \* **month:** il mese del periodo della busta paga;
  - **Corpo della richiesta:** il file da caricare all'interno del server e dividere;
  - **Risposta:** nel caso in cui il caricamento vada a buon fine restituisce un semplice codice di successo;
- GET /api/v1/Pdf/download
  - **Parametri:** è possibile scegliere fra uno solo dei seguenti parametri:
    - \* **filename:** il nome del file che si vuole scaricare;
    - \* **id:** il codice identificativo del file che si vuole scaricare;
  - **Risposta:** uno stream di dati in formato Base64 contenente il file richiesto;
- GET /api/v1/Pdf/documents/getall/{year}/{month}
  - **Parametri:**
    - \* **year:** l'anno del periodo della busta paga;
    - \* **month:** il mese del periodo della busta paga;
  - **Risposta:** un oggetto contenente la lista di tutti i documenti presenti in un certo periodo (identificato da anno e mese);
- GET /api/v1/Pdf/documents/getallpaginate/{year}/{month}
  - **Parametri:**
    - \* **year:** l'anno del periodo della busta paga;
    - \* **month:** il mese del periodo della busta paga;
    - \* **page:** il numero della pagina interessata;
    - \* **perpage:** il numero di elementi per pagina;
  - **Risposta:** un oggetto contenente il numero totale di pagine e la lista di documenti in un certo periodo alla pagina indicata, con il numero indicato di righe per pagina;
- GET /api/v1/Pdf/documents/getallperiods
  - **Risposta:** un oggetto contenente la lista di tutti i periodi presenti nel database ordinati in ordine decrescente;
- POST /api/v1/Pdf/email/send
  - **Corpo della richiesta:** l'oggetto contenente tutte le informazioni del documento da inviare;
  - **Risposta:** nel caso in cui l'invio vada a buon fine restituisce un codice di successo, altrimenti restituisce un messaggio di errore;

4.2.2 Diagramma UML delle classi

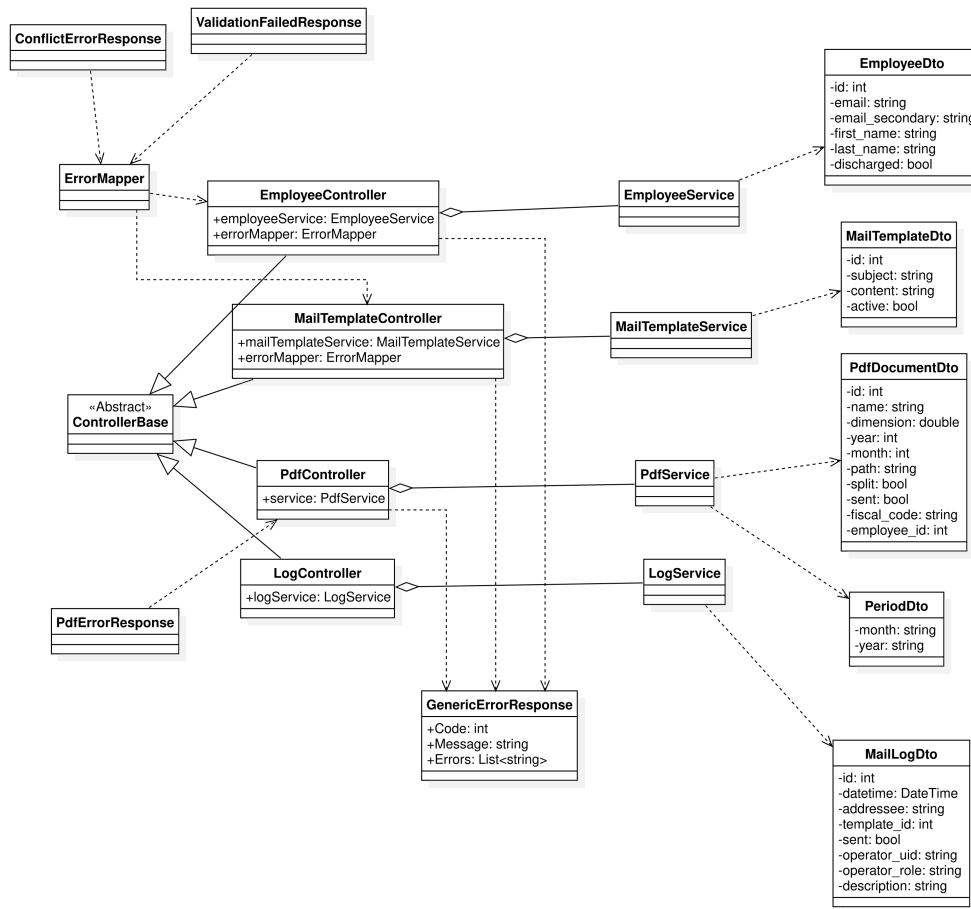


Figura 4.2: Progettazione delle classi del back-end

Questo è il diagramma UML che rappresenta le classi principali del progetto. Le classi che contengono alla fine del nome la parola “Controller” sono, appunto, i controller che restano in ascolto delle chiamate alle Web API elencate nella sezione precedente. Ogni classe Controller contiene un metodo che gestisce una singola API.

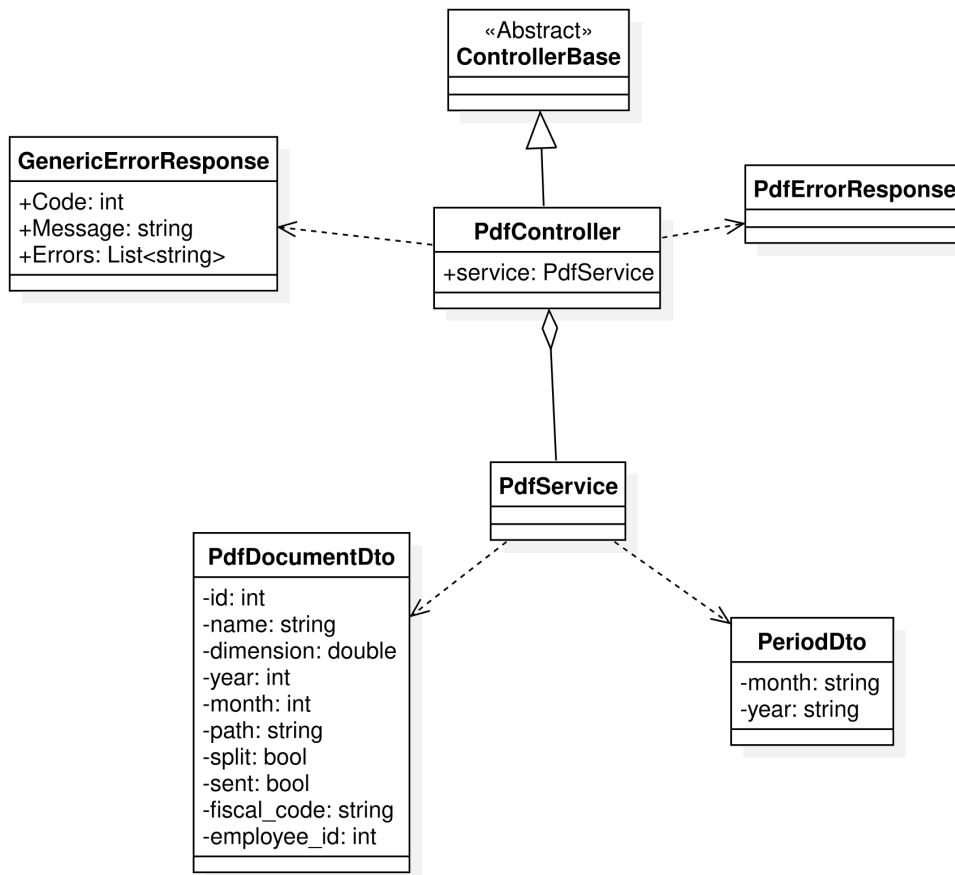


Figura 4.3: Esempio di controller e service

Quando questa API viene chiamata, il Controller chiama un metodo della classe **Service** in cui è contenuta la logica effettiva. In questo modo si può facilmente dividere l'interfaccia pubblica dalla logica interna dell'applicazione.

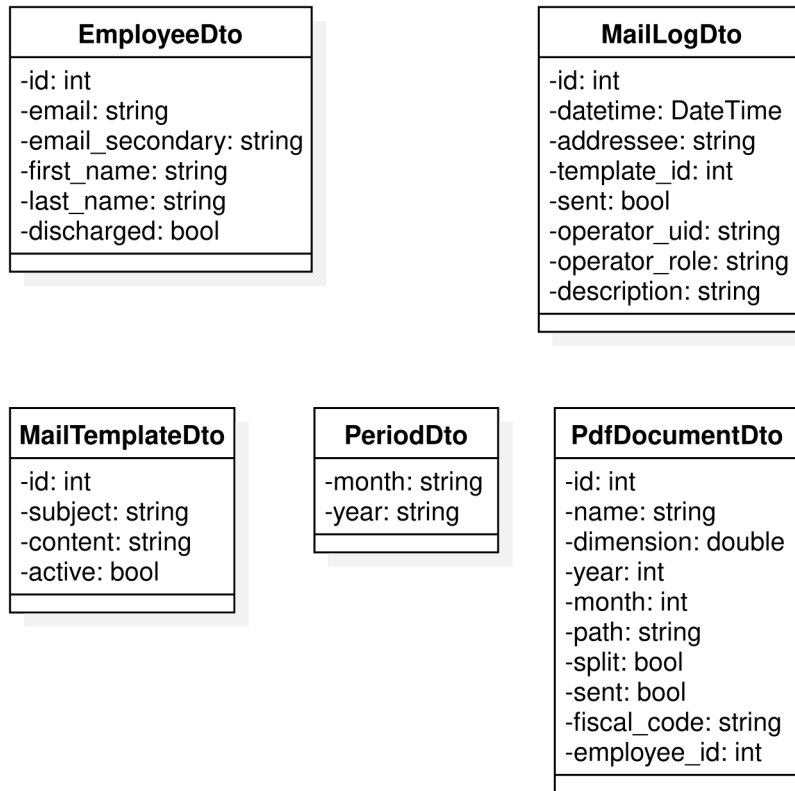


Figura 4.4: Classi DTO

Un altro gruppo di classi che è facilmente riconoscibile è quello che contiene le classi che hanno il nome che finisce con “**Dto**”. **DTO** significa “Data transfer object” e sono appunto le classi che vengono utilizzate per trasferire i dati dal back-end come risposta alle chiamate API. Infatti queste classi non contengono metodi di alcun tipo, ma soltanto gli attributi di cui le classi hanno bisogno per trasferire correttamente i dati dell’oggetto che rappresentano.



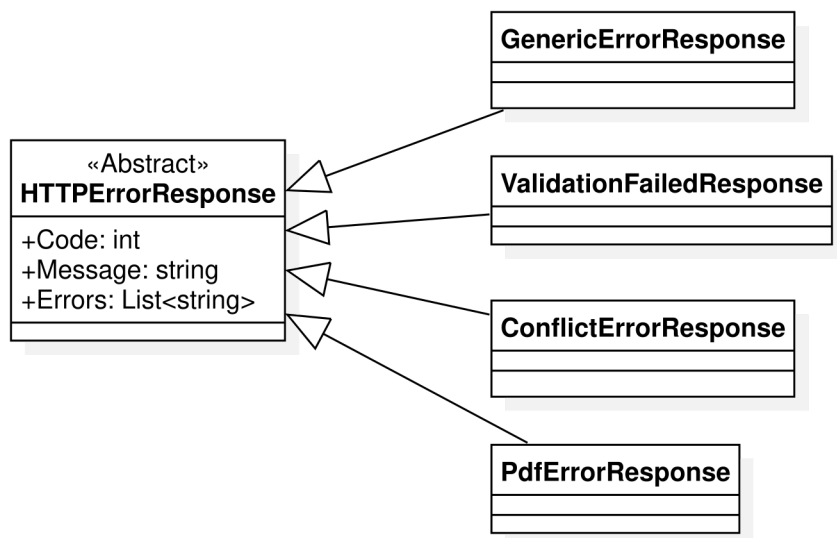


Figura 4.5: Classi Response

L'ultimo gruppo di classi di cui è importante sottolineare il ruolo è quello col nome che finisce per **"Response"**. Queste classi le ho utilizzate per restituire come risposta alle chiamate API degli oggetti che rappresentassero errori di vario tipo, per esempio conflitti all'interno del database o errori di validazione.

### 4.3 Front-end

Per quanto riguarda il front-end, la progettazione che ho fatto è stata tramite bozze dell'interfaccia grafica, della disposizione dei componenti e delle interazioni con l'utente, lasciandomi ispirare anche dai componenti grafici che offre Bootstrap, la libreria grafica che ho utilizzato in questo progetto.

The image shows a user interface for document management. It is divided into two main sections: 'Carica Pdf' (Upload PDF) and 'File Pdf Caricati' (Uploaded PDF Files).

**Carica Pdf**

This section contains two dropdown menus for selecting a month ('Luglio') and a year ('2023'), followed by a text input field labeled 'Seleziona File'.

**File Pdf Caricati**

This section contains two dropdown menus for selecting a month ('Luglio') and a year ('2023'), a green 'Cerca' (Search) button, a white search input field, and a green 'Filtra' (Filter) button.

Below the search controls is a table with the following structure:

Nome	Peso	Tipo	Stato	
File1.pdf	300KB	Intero	Non inviato	<div style="display: flex; justify-content: space-around;"><span>Invia tutte</span><span>Download</span><span>Invia</span></div>

Figura 4.6: Progettazione della pagina di gestione documenti

La prima pagina ad essere stata progettata è stata quella della gestione documenti. Ho pensato fosse una buona idea unire all'interno della stessa pagina sia la funzionalità di upload dei file, sia quelle di gestione dei documenti come la visualizzazione della lista di quelli già caricati e i tasti di invio delle mail. In questo modo è possibile accedere alle funzionalità principali dell'applicazione da una stessa pagina con il minor numero di click possibile dato che tutto è a portata di mano.

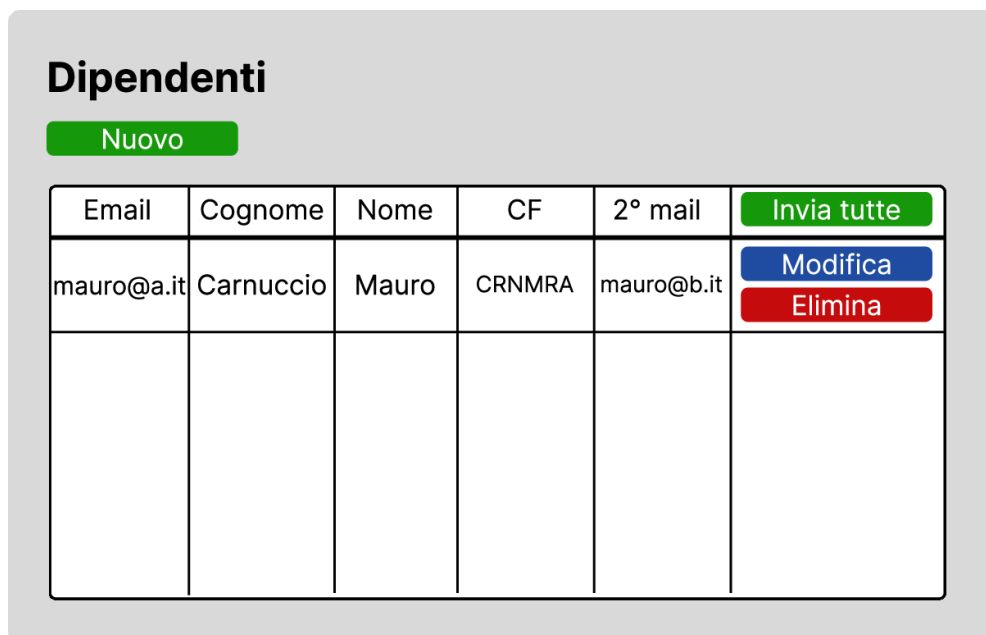


Figura 4.7: Progettazione della pagina di gestione dipendenti

In seguito ho progettato la pagina di gestione dei dipendenti in cui è possibile eseguire tutte le classiche operazioni di creazione, modifica, cancellazione e visualizzazione dei dipendenti. In più è stato deciso di aggiungere la paginazione così da alleggerire il peso della pagina e quindi diminuire il tempo di caricamento della stessa.

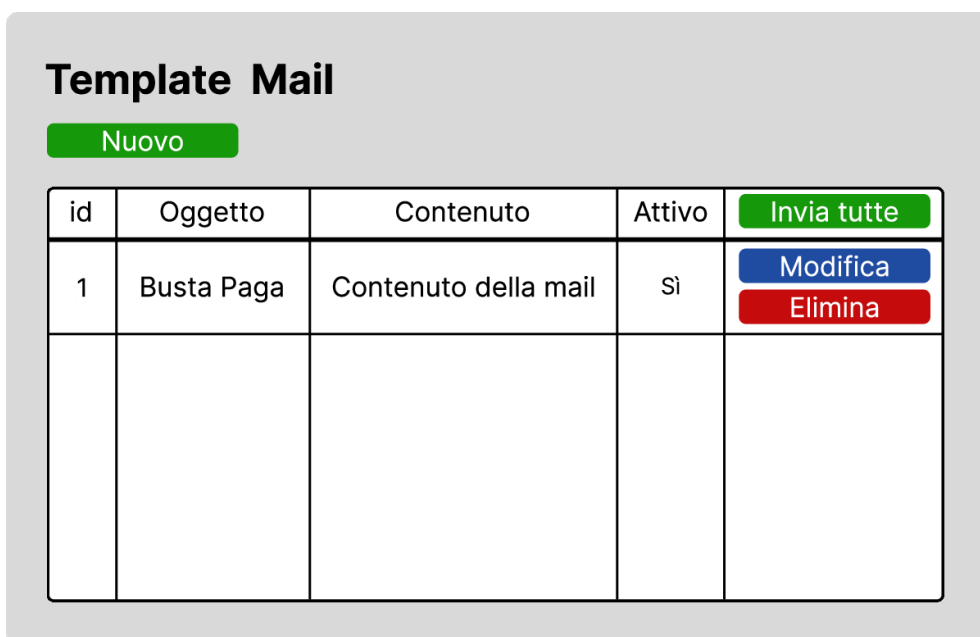


Figura 4.8: Progettazione della pagina di gestione dipendenti

Analogamente poi ho progettato la pagina di gestione dei template che, come quella di gestione dei dipendenti, implementa le operazioni di creazione, modifica, cancellazione e visualizzazione dei template.

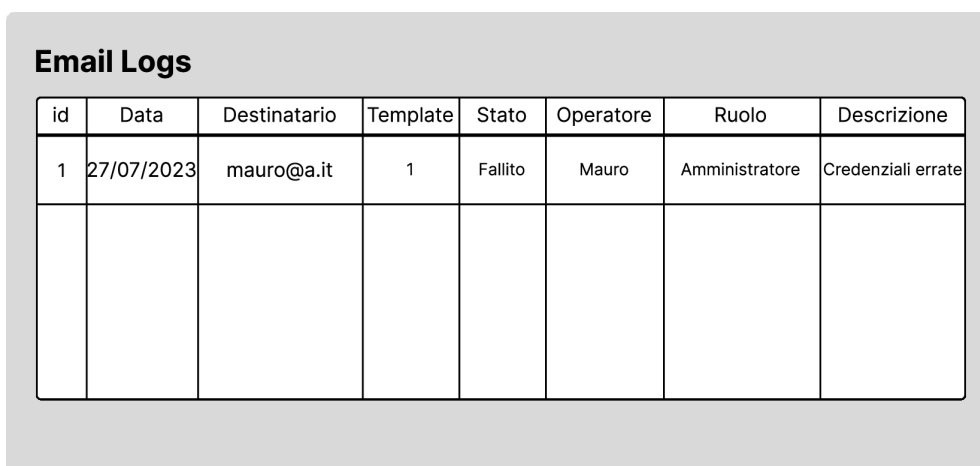


Figura 4.9: Progettazione pagina di visualizzazione dello storico degli invii

L'ultima pagina che ho progettato è stata quella di visualizzazione dello storico degli invii. Lo storico deve mostrare tutti i risultati ottenuti per ogni tentativo di invio mail, sia in caso di successo, che in quello di fallimento.

Dato che questa è l'unica funzionalità che questa pagina offre, mi sono limitato a posizionare una tabella che contenesse tutte le informazioni necessarie.



Figura 4.10: Progettazione della barra di navigazione utente non autenticato



Figura 4.11: Progettazione della barra di navigazione dell'Amministratore



Figura 4.12: Progettazione della barra di navigazione dell'Operatore

Per quanto riguarda la barra di navigazione ho voluto adottare una struttura semplice in modo da renderla pulita e di facile utilizzo. Ho tenuto allineate a sinistra le voci che portano alle pagine dell'applicazione, mentre a destra ho posizionato il tasto per eseguire l'accesso nel caso in cui l'utente non l'abbia ancora effettuato, e un bottone con le informazioni dell'utente nel caso in cui, invece, l'accesso sia stato effettuato. Come si può notare, il menù ha una quantità di voci differente in base al ruolo dell'utente che ha effettuato l'accesso, questo perché come precedentemente anticipato, ogni ruolo può usufruire di un insieme prestabilito di funzionalità.

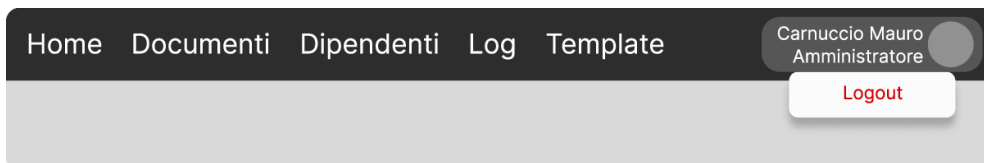


Figura 4.13: Progettazione logout

Il tasto per eseguire il logout ho voluto inserirlo come menù aggiuntivo a seguito di un click sul tasto con le informazioni dell'utente, così da non renderlo sempre visibile e rischiare che l'utente facesse il logout involontariamente.

## Capitolo 5

# Realizzazione

In questo capitolo illustrerò il risultato del progetto di stage, sia tramite immagini che tramite didascalie, descrivendone ogni sua sezione e il loro funzionamento.

### 5.1 Homepage



Figura 5.1: Homepage

Questa è una semplice pagina di benvenuto sempre visibile indipendentemente dal fatto che l'utente abbia effettuato l'accesso o meno.

## 5.2 Barra di navigazione con accesso non eseguito



Figura 5.2: Barra di navigazione con accesso non eseguito

Questa è la visualizzazione della barra di navigazione nel momento in cui l'utente non ha ancora effettuato l'accesso.

Come si può vedere, le uniche opzioni a disposizione sono la visualizzazione della Homepage e il tasto per effettuare il login.

## 5.3 Barra di navigazione con accesso eseguito

Dato che gli utenti che possono effettuare l'accesso sono di due tipi, ovvero **Amministratore** e **Operatore**, le barre di navigazione per ciascun ruolo sono diverse in base alle funzionalità a cui possono accedere. La barra di navigazione per un utente **Amministratore** rende disponibili le seguenti funzionalità:

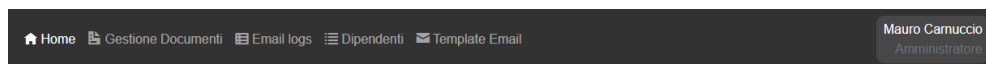


Figura 5.3: Barra di navigazione dell'Amministratore

Mentre la barra di navigazione per un utente **Operatore** rende disponibili le seguenti funzionalità:



Figura 5.4: Barra di navigazione dell'Operatore

Sulla destra della barra di navigazione si può notare il bottone con le informazioni dell'utente che ha effettuato il login con cognome, nome e ruolo. Come precedentemente anticipato, interagendo col bottone è possibile visualizzare il menù secondario con il tasto per eseguire il logout.

## 5.4 Login

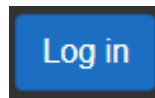


Figura 5.5: Tasto di login

Per accedere alle funzionalità offerte dall'applicazione è necessario effettuare il login. Per farlo è stato inserito il bottone che rimanda alla pagina di accesso aziendale. In questa pagina è necessario inserire l'indirizzo email e la password del proprio account aziendale. Una volta effettuato l'accesso, l'applicazione offrirà all'utente le funzionalità disponibili per il ruolo che gli è stato assegnato.

## 5.5 Logout

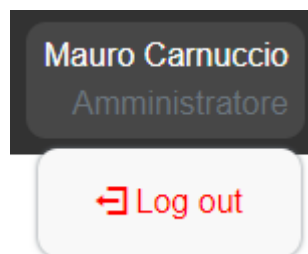
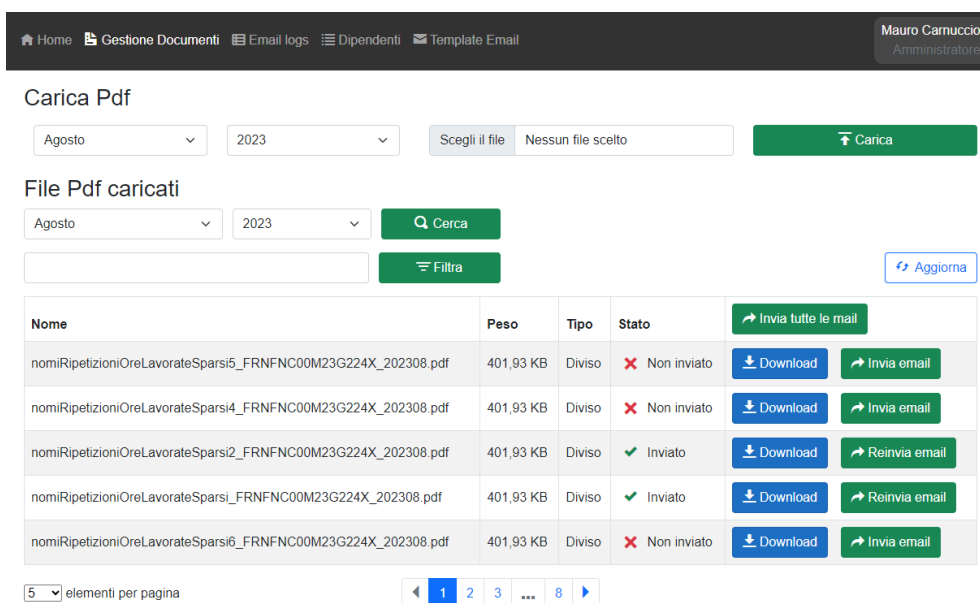


Figura 5.6: Tasto di logout

Nel caso in cui sia necessario effettuare il Logout, è presente il tasto apposito nel sotto-menù accessibile interagendo con il bottone della barra di navigazione che mostra le informazioni dell'utente che ha effettuato l'accesso.



## 5.6 Gestione documenti



Carica Pdf

Agosto 2023 Scegli il file Nessun file scelto Carica

File Pdf caricati

Agosto 2023 Cerca

Filtra Aggiorna

Nome	Peso	Tipo	Stato	Invia tutte le mail
nomiRipetizioniOreLavorateSparsi5_FRNFNC00M23G224X_202308.pdf	401,93 KB	Diviso	Non inviato	Download Invia email
nomiRipetizioniOreLavorateSparsi4_FRNFNC00M23G224X_202308.pdf	401,93 KB	Diviso	Non inviato	Download Invia email
nomiRipetizioniOreLavorateSparsi2_FRNFNC00M23G224X_202308.pdf	401,93 KB	Diviso	Inviato	Download Reinvia email
nomiRipetizioniOreLavorateSparsi_FRNFNC00M23G224X_202308.pdf	401,93 KB	Diviso	Inviato	Download Reinvia email
nomiRipetizioniOreLavorateSparsi6_FRNFNC00M23G224X_202308.pdf	401,93 KB	Diviso	Non inviato	Download Invia email

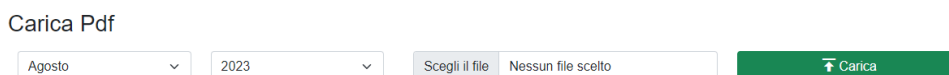
5 elementi per pagina

1 2 3 ... 8

Figura 5.7: Pagina di gestione dei documenti

Questa è la pagina più corposa in cui sono racchiuse le funzionalità principali dell'applicazione. Le illustrerò qui di seguito sezione per sezione.

### 5.6.1 Caricamento del file PDF



Carica Pdf

Agosto 2023 Scegli il file Nessun file scelto Carica

Figura 5.8: Sezione per il caricamento del file PDF

Per caricare il file **PDF** con le buste paga, è necessario andare ad inserire prima il periodo a cui il documento si riferisce. Per inserirlo basta selezionare il mese e l'anno dai menù a tendina sulla sinistra. Questi menù sono precompilati all'apertura della pagina con il mese e l'anno correnti per un'interazione più rapida con l'applicazione. In seguito basta andare a selezionare il file PDF tramite il selettore di file.

Una volta selezionato il documento basta cliccare sul tasto "**Carica**" sulla destra. Concluso l'upload del documento sarà possibile scaricare un file zip contenente le buste paga di tutto il periodo selezionato cliccando il tasto "**Download**".

### 5.6.2 Ricerca dei file PDF caricati

**File Pdf caricati**

Figura 5.9: Sezione per il filtraggio dei documenti

In questa sezione si può visualizzare una tabella contenente una lista di tutti i file presenti relativi al periodo selezionato. Come per il caricamento del file si può andare a selezionare il periodo dai menù a tendina e in seguito cliccando il tasto “Cerca” si vanno a visualizzare i documenti relativi a quel periodo.

Subito sotto è presente un campo testuale per effettuare il filtraggio dei documenti visualizzati. Questo filtraggio viene effettuato secondo il nome del documento, quindi basta inserire il nome del file di interesse o parte di esso e, premendo il tasto filtra, verranno visualizzati tutti i documenti (in quel periodo) che contengono all’interno del loro nome la stringa indicata.

### 5.6.3 Visualizzazione dei file PDF caricati

Nome	Peso	Tipo	Stato	<input type="button" value="Invia tutte le mail"/>
nomiRipetizioniOreLavorateSparsi5_FRNFNC00M23G224X_202308.pdf	401,93 KB	Diviso	✘ Non inviato	<input type="button" value="Download"/> <input type="button" value="Invia email"/>
nomiRipetizioniOreLavorateSparsi4_FRNFNC00M23G224X_202308.pdf	401,93 KB	Diviso	✘ Non inviato	<input type="button" value="Download"/> <input type="button" value="Invia email"/>
nomiRipetizioniOreLavorateSparsi2_FRNFNC00M23G224X_202308.pdf	401,93 KB	Diviso	✔ Inviato	<input type="button" value="Download"/> <input type="button" value="Reinvia email"/>
nomiRipetizioniOreLavorateSparsi_FRNFNC00M23G224X_202308.pdf	401,93 KB	Diviso	✔ Inviato	<input type="button" value="Download"/> <input type="button" value="Reinvia email"/>
nomiRipetizioniOreLavorateSparsi6_FRNFNC00M23G224X_202308.pdf	401,93 KB	Diviso	✘ Non inviato	<input type="button" value="Download"/> <input type="button" value="Invia email"/>

Figura 5.10: Lista dei documenti

Sotto a questi campi si può vedere la lista effettiva di documenti che presenta per ciascun documento: il nome, il peso in KB del documento, il tipo (*Intero/Diviso*), lo stato attuale (*Non inviato/Non inviabile/Inviato*) e una sezione di funzionalità.

Le funzionalità offerte sono le seguenti:

- **Download:** Apre una nuova scheda per la visualizzazione e l’eventuale download del documento;
- **Invia:** Se il documento è associato ad un dipendente, è presente il tasto per effettuare l’invio della mail al dipendente con il documento in allegato. Se il documento è già stato inviato il bottone presenta il testo **Reinvia**;

- **Invia tutte:** In cima alla colonna delle funzionalità è presente un tasto che permette di accodare l'invio delle mail per tutti i documenti che possono essere inviati (che hanno quindi un dipendente associato);
- **Associa:** Se il documento non è associato a nessun dipendente, sarà presente il tasto per permettere all'utente di creare un nuovo dipendente con il campo del codice fiscale già precompilato.



Figura 5.11: Selettore di pagina

In fondo alla pagina è presente il selettore delle pagine e un menù a tendina per selezionare la quantità di elementi da visualizzare per ogni pagina.

## 5.7 Gestione dipendenti

Email	Cognome	Nome	Codice fiscale	Seconda email	Stato	
franchinifanco@a.it	franchini	fanco	FRNFNC00M23G224X	franchini@b.it	● Attivo	
lambertinatina@a.it	Lamberti	natina	LMBNTN01L31H501H	natina@b.it	● Attivo	
i.natanti@a.it	natanti	ignao	NTNGNI06S10A271W	ignao@b.it	● Attivo	
pinzini@a.it	Pinzini	Rarto	PNZRR04T12D969D	pinzini.r@b.it	● Attivo	
tarzani.nadia@a.it	tarzani	nadia	TRZND05P66L424F	tarzani@b.it	● Dimesso	

Figura 5.12: Pagina di gestione dei dipendenti

Questa pagina permette di gestire tutte le informazioni riguardanti i dipendenti. La creazione, modifica, visualizzazione e cancellazione.

### 5.7.1 Nuovo dipendente

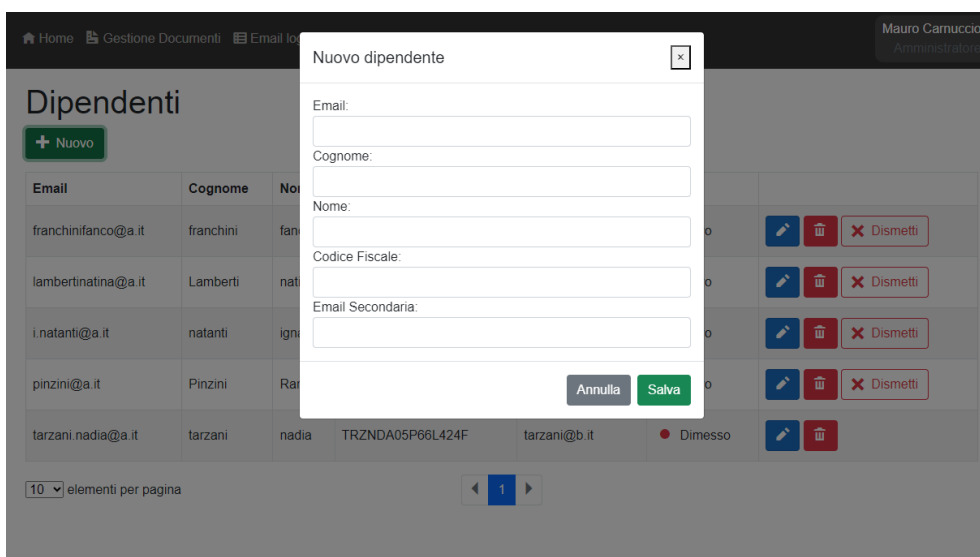


Figura 5.13: Popup per la creazione di un nuovo dipendente

Cliccando sul tasto “**Nuovo**” in cima alla tabella si apre il seguente popup per la creazione di un nuovo dipendente.

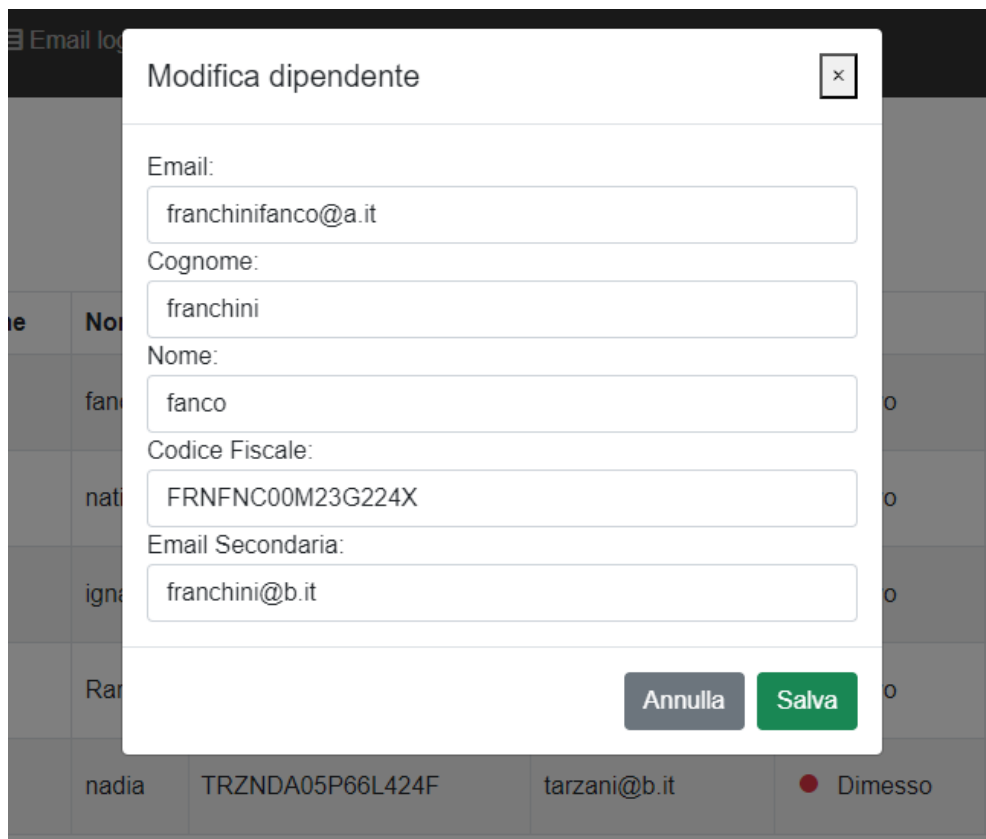
### 5.7.2 Lista dipendenti

Email	Cognome	Nome	Codice fiscale	Seconda email	Stato	
franchinifanco@a.it	franchini	fanco	FRNFNC00M23G224X	franchini@b.it	● Attivo	
lambertinatina@a.it	Lamberti	natina	LMBNTN01L31H501H	natina@b.it	● Attivo	
i.natanti@a.it	natanti	ignao	NTNGNI06S10A271W	ignao@b.it	● Attivo	
pinzini@a.it	Pinzini	Rarto	PNZRRT04T12D969D	pinzini.r@b.it	● Attivo	
tarzani.nadia@a.it	tarzani	nadia	TRZNDA05P66L424F	tarzani@b.it	● Dimesso	

Figura 5.14: Lista dei dipendenti

In seguito, per ogni dipendente nella lista si può visualizzare, da sinistra a destra, l’indirizzo email principale, il cognome, il nome, il codice fiscale, l’indirizzo email secondario, e la colonna con i tasti per la modifica e la cancellazione. Infine è presente anche un tasto “**Dismetti**” per indicare il dipendente come “Dimesso”.

### 5.7.3 Modifica dipendente



The image shows a web application interface with a modal popup titled "Modifica dipendente". The popup contains the following fields and values:

Field	Value
Email:	franchinifanco@a.it
Cognome:	franchini
Nome:	fanco
Codice Fiscale:	FRNFNC00M23G224X
Email Secondaria:	franchini@b.it

At the bottom right of the popup, there are two buttons: "Annulla" (grey) and "Salva" (green). The background shows a table with employee data, including names like "nadia", "franco", and "tarzani", and their respective tax codes and emails.

Figura 5.15: Popup modifica del dipendente

Cliccando il tasto per la modifica si apre un popup uguale a quello per la creazione, ma con i campi precompilati con le informazioni del dipendente.

### 5.7.4 Cancellazione dipendente

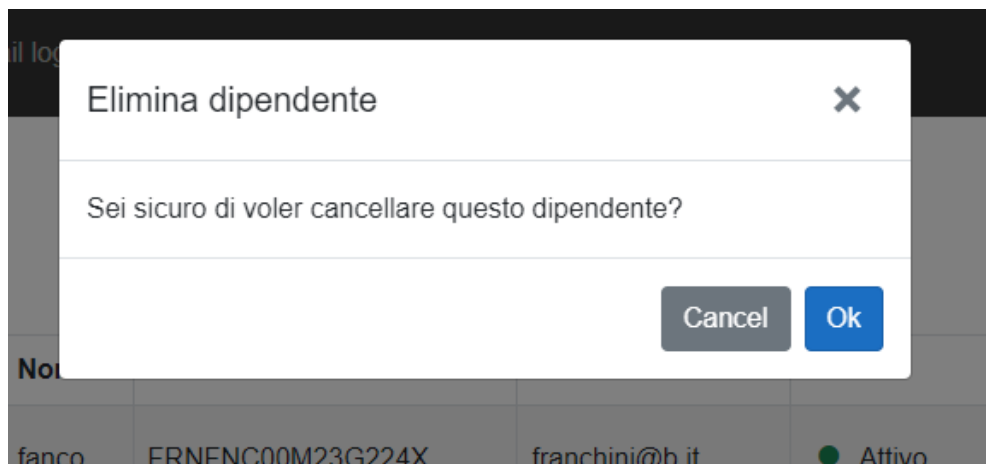


Figura 5.16: Richiesta di conferma della cancellazione del dipendente

Con il tasto di cancellazione invece si apre un popup ne che richiede la conferma. In fondo è presente il selettore della pagina come nella pagina di gestione dei documenti.

## 5.8 Gestione template email

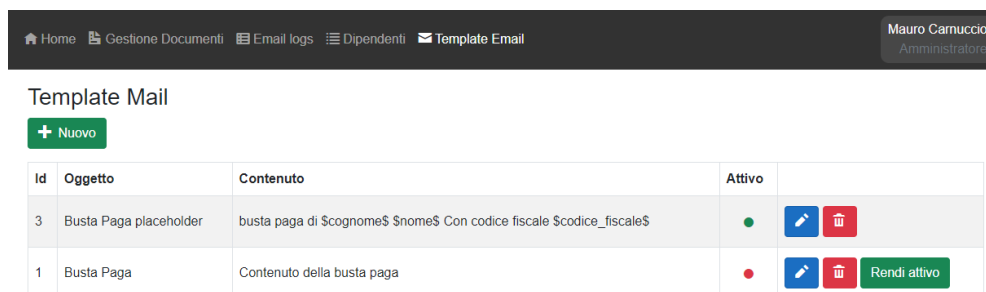


Figura 5.17: Pagina di gestione dei template

Il funzionamento di questa pagina è simile a quello della pagina di gestione dei dipendenti.

### 5.8.1 Nuovo template

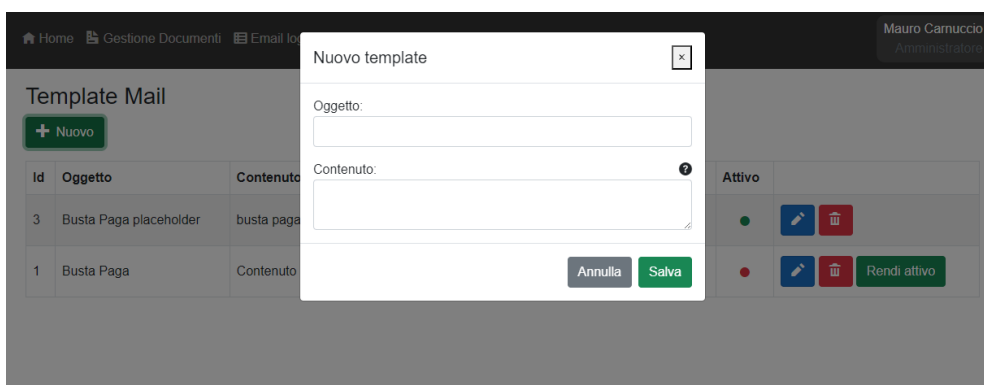


Figura 5.18: Popup di creazione di un nuovo template

Col tasto “**Nuovo**” si apre il popup di creazione di un nuovo template. In questa schermata si inseriscono l’oggetto della mail e il corpo della mail.

### 5.8.2 Lista template






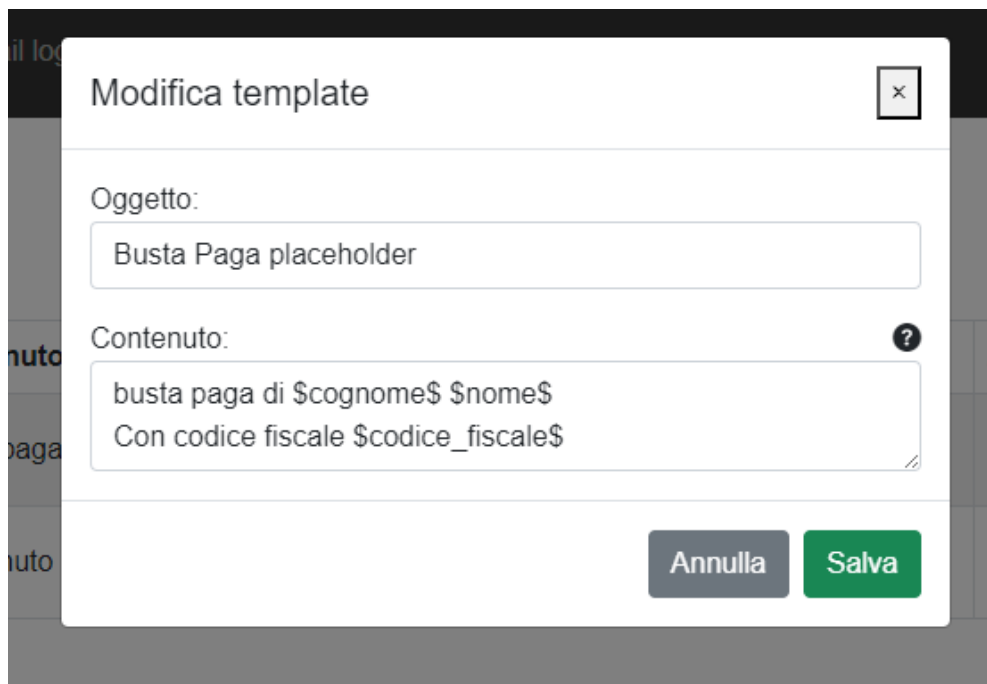
Id	Oggetto	Contenuto	Attivo	
3	Busta Paga placeholder	busta paga di \$cognome\$ \$sname\$ Con codice fiscale \$codice_fiscale\$	●	 
1	Busta Paga	Contenuto della busta paga	●	  

Figura 5.19: Lista dei template

Come nelle altre pagine, è possibile visualizzare una lista di tutti i template che sono stati inseriti. Per ciascuna riga si possono vedere l’oggetto del template, il corpo del template e la colonna con i tasti di modifica e cancellazione.

Un altro tasto disponibile è il tasto “**Rendi attivo**”. Questo tasto permette di rendere attivo il template selezionato, disattivando, di conseguenza, il template attualmente attivo.

### 5.8.3 Modifica template



Modifica template

Oggetto:

Busta Paga placeholder

Contenuto: ?

busta paga di \$cognome\$ \$nome\$  
Con codice fiscale \$codice\_fiscale\$

Annulla Salva

Figura 5.20: Popup di modifica del template

Con il tasto di modifica alla destra di ciascun template si apre il popup precompilato per modificare il template.



### 5.8.4 Placeholder

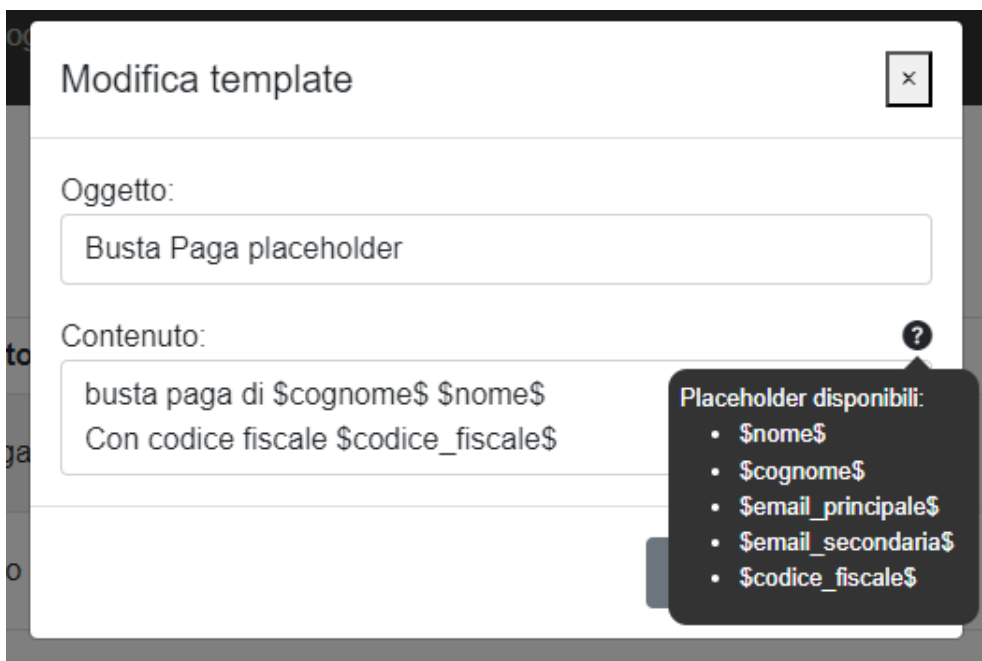


Figura 5.21: Tooltip per i placeholder disponibili

Posizionando il cursore sull'icona col punto di domanda compare un **tooltip** che informa del fatto che sono presenti dei **placeholder** utilizzabili nel corpo della mail che verranno sostituiti, al momento dell'invio, con il loro relativo valore per ciascun dipendente.

I **placeholder** disponibili sono i seguenti:

- **\$cognome\$**: viene sostituito col cognome del dipendente;
- **\$nome\$**: viene sostituito col nome del dipendente;
- **\$codice\_fiscale\$**: viene sostituito col codice fiscale del dipendente;
- **\$email\_principale\$**: viene sostituito con l'indirizzo email principale del dipendente;
- **\$email\_secondaria\$**: viene sostituito con l'indirizzo email secondario del dipendente.

### 5.8.5 Cancellazione template



Figura 5.22: Richiesta di conferma della cancellazione del template

Con il tasto di cancellazione si apre il popup di conferma.

## 5.9 Visualizzazione storico invii

 A screenshot of a web application interface showing the 'Email Logs' page. The page has a dark header with navigation links: Home, Gestione Documenti, Email logs (selected), Dipendenti, and Template Email. On the right of the header, the user 'Mauro Carnuccio' is identified as 'Amministratore'. Below the header, the title 'Email Logs' is displayed above a table. The table has 8 columns: id, Data, Destinatario, Codice template, Stato, Operatore, Ruolo dell'operatore, and Descrizione. There are 6 rows of data. The first five rows show successful sends ('Inviata') with green checkmarks. The sixth row shows a failed send ('Fallita') with a red X. Below the table, there is a pagination control showing '5 elementi per pagina' and a set of numbered buttons from 1 to 10, with button 4 highlighted.
 

id	Data	Destinatario	Codice template	Stato	Operatore	Ruolo dell'operatore	Descrizione
35	04/08/2023 14:58:33	m.carnuccio@conssoftinformatica.it	3	✓ Inviata	m.carnuccio@conssoftinformatica.it	CI_BP_ADMIN	-
34	03/08/2023 10:44:06	mauroboh@gmail.com	1	✓ Inviata	m.carnuccio@conssoftinformatica.it	CI_BP_ADMIN	-
33	02/08/2023 22:21:00	m.carnuccio@conssoftinformatica.it	1	✓ Inviata	m.carnuccio@conssoftinformatica.it	CI_BP_ADMIN	-
32	02/08/2023 14:23:25	m.carnuccio@conssoftinformatica.it	1	✓ Inviata	m.carnuccio@conssoftinformatica.it	CI_BP_ADMIN	-
31	02/08/2023 11:06:59	PRTCLR07P06L219P	1	✗ Fallita	m.carnuccio@conssoftinformatica.it	CI_BP_ADMIN	Non esiste alcun dipendente associato a questo documento

Figura 5.23: Pagina di visualizzazione dello storico degli invii

Questa è la pagina più semplice, in cui è possibile consultare il log degli invii delle mail.

La tabella contiene i seguenti campi: id (il numero di riga del log), la data in cui si è verificato l'evento, il destinatario della mail (nel caso in cui si sia tentato di inviare una mail ad un dipendente con l'indirizzo mail non indicato verrà visualizzato il codice fiscale), l'id del template utilizzato, lo stato dell'invio (*Inviata/Fallita*), l'indirizzo mail dell'utente che ha effettuato l'invio, il ruolo dell'utente che ha effettuato l'invio, e la descrizione risultato dell'invio nel caso in cui questo sia fallito.

## Capitolo 6

# Conclusioni

### 6.1 Valutazione degli strumenti utilizzati

Gli strumenti e le tecnologie che mi sono state proposte mi hanno interessato fin da subito:

1. **Blazor**, la tecnologia principale che ho utilizzato, è stato facile da imparare e da utilizzare grazie anche ai corsi offerti da ConsoftInformatica. L'unico difetto di questo **Framework** che mi sento di far notare è il fatto che, contrariamente a JavaScript, Blazor non va a modificare il **DOM** dinamicamente, ma esegue un re-render dei componenti ogni volta che un evento si verifica. Questa sua caratteristica va ad abbassare leggermente il livello di fluidità dell'applicazione limitando le possibilità di personalizzazione della pagina come ad esempio le animazioni. Per sopperire a questo problema è possibile utilizzare la *JavaScript interoperability*, però risulta più macchinosa da utilizzare e un po' meno leggibile. Ad eccezione di questo piccolo difetto, Blazor è stato piacevole da utilizzare e la sua documentazione ufficiale è stata più che sufficiente per aiutarmi a risolvere i problemi che ho riscontrato e per imparare di più sulle potenzialità di questo **Framework**.
2. Per quanto riguarda **Bootstrap**, non l'ho amata come libreria grafica, molto probabilmente perché ancora sono inesperto nel suo utilizzo, ma sicuramente è stata estremamente utile per velocizzare la creazione dell'interfaccia grafica, anche se basilare. Se dovessi dare una preferenza sceglierei ancora di scrivere CSS manualmente, in modo da poter creare un'interfaccia personalizzata appositamente per l'applicativo che sto creando. In alternativa utilizzerei in quantità minore le funzionalità offerte da Bootstrap, che alcune volte mi hanno creato più problemi di quelli che mi sarei aspettato.

### 6.2 Possibili estensioni

Durante il periodo di stage mi sono concentrato anche a ragionare su quali potessero essere delle funzionalità utili che estendessero e migliorassero il portale per fornire un'esperienza utente più piacevole:

1. Una delle funzionalità che purtroppo non si è riusciti ad implementare per via di policy aziendali, ma che sarebbe stata utile per rendere l'applicazione ulteriormente indipendente dalla macchina su cui è installato il back-end, è quella di salvare ogni file su **OneDrive**. In questo modo, nel caso in cui il back-end debba essere installato su un'altra macchina, non ci sarebbe la necessità di spostare ogni documento dalla macchina vecchia a quella nuova: l'applicazione sarebbe già istruita per connettersi automaticamente al server di OneDrive in cui i documenti sono salvati.
2. Un'altra estensione interessante che ho esplorato verso la fine del periodo di stage è stata l'utilizzo di **Electron**, o meglio, in questo caso specifico Electron.NET. Electron è un **Framework** che permette di sviluppare l'interfaccia grafica di applicazioni desktop tramite tecnologie Web. Tramite questa tecnologia quindi, si potrebbe trasformare il portale per le buste paga in una vera applicazione desktop. Per fare questo però è necessario avere un'applicazione Blazor di tipo server, e non un front-end di tipo Blazor Web Assembly come quello che ho sviluppato durante questo stage. Questo perché Electron racchiude l'intera applicazione per renderla un'applicazione desktop, quindi sia front-end che back-end. Quindi, nel caso in cui si volesse trasferire questa applicazione Web in un'applicazione desktop tramite Electron, sarà necessario spostare il progetto front-end e il progetto back-end in un terzo progetto. Questo progetto deve essere di tipo Blazor Server e deve racchiudere entrambe le parti, di conseguenza sarà necessario ri-progettare la comunicazione fra "front-end" e "back-end", dato che in questo tipo di applicazioni la comunicazione non è tramite Web API ma tramite semplice chiamata di metodi.
3. Un miglioramento per l'esperienza utente, che era previsto come obiettivo opzionale (OP1), sarebbe una **sincronizzazione** periodica automatica dei dipendenti dai server Azure dell'azienda. Con una sincronizzazione di questo genere, gli amministratori non avrebbero la necessità di inserire manualmente i dati di ogni singolo dipendente, perché questi sarebbero già inseriti automaticamente.
4. Un'alternativa a questa sincronizzazione sarebbe l'inserimento, sempre automatico, dei dipendenti durante la suddivisione del documento caricato. Si dovrebbe, quindi, modificare l'algoritmo di suddivisione del documento per fare in modo che esegua anche una ricerca delle informazioni del dipendente, per ogni busta paga che trova, e le inserisca nel database. Così si avrebbe la certezza che sia sempre presente nel database ogni dipendente di cui si trova una busta paga.

### 6.3 Valutazione personale

L'esperienza di stage appena vissuta è stata molto formativa sia relativamente alle mie conoscenze sulle tecnologie utilizzate, sia per quanto riguarda la mia mentalità e il mio approccio al mondo del lavoro. Ho potuto cimentarmi per la prima volta nello sviluppo di un'applicazione che verrà effettivamente utilizzata in futuro. In più, il fatto di aver dovuto sviluppare questa applicazione in quasi totale autonomia ha aggiunto un livello in più di responsabilità. Fortunatamente il mio tutor aziendale ha saputo consigliarmi tecnologie utili e mi ha indirizzato nella risoluzione dei problemi più grossi che ho avuto.

In più l'ambiente in cui ho lavorato è stato estremamente accogliente e mi ha permesso di fare conoscenza sia in ambito lavorativo che personale con i colleghi presenti nella sede.

In conclusione sono soddisfatto sia del lavoro svolto, nonostante sia consapevole delle possibilità di miglioramento del prodotto, che delle connessioni che ho sviluppato con i colleghi dell'ufficio.

# Riferimenti

- [1] «Azure Active Directory - Microsoft Azure». <https://azure.microsoft.com/it-it/products/active-directory>
- [2] «iText - The Leading PDF Library for Developers». <https://itextpdf.com/>
- [3] «StarUML». <https://staruml.io/>
- [4] «.NET (e .NET Core) introduzione e panoramica». <https://learn.microsoft.com/it-it/dotnet/core/introduction>
- [5] «Blazor - Build client web apps with C#». <https://learn.microsoft.com/it-it/aspnet/core/blazor/?view=aspnetcore-6.0>
- [6] «GitLab - The DevSecOps Platform». <https://about.gitlab.com/>
- [7] «PostgreSQL: The world's most advanced open source database». <https://www.postgresql.org/>
- [8] «Blazor - Build client web apps with C#». <https://dotnet.microsoft.com/en-us/apps/aspnet/web-apps/blazor>
- [9] «Docker: Accelerated Container Application Development». <https://www.docker.com/>
- [10] «Microsoft Graph Overview». <https://learn.microsoft.com/en-us/graph/overview>
- [11] «Documentazione di C#». <https://learn.microsoft.com/it-it/dotnet/csharp/>
- [12] «CSS - Cascading Style Sheets». <https://www.w3.org/Style/CSS/Overview.en.html>
- [13] «W3C». <https://www.w3.org/>
- [14] «W3C HTML». <https://www.w3.org/html/>
- [15] «JavaScript Introduction». [https://www.w3schools.com/js/js\\_intro.asp](https://www.w3schools.com/js/js_intro.asp)
- [16] «Bootstrap - The most popular HTML, CSS, and JS library in the world». <https://getbootstrap.com/>
- [17] «Flurl». <https://flurl.dev/>
- [18] «Npgsql - .NET Access to PostgreSQL». <https://www.npgsql.org/>

- [19] «SqlKata the dotnet SQL query builder». <https://sqlkata.com/>
- [20] «Visualstudio: editor di codice e IDE per sviluppatori e team software». <https://visualstudio.microsoft.com/it/>
- [21] «Swagger - API Documentation & Design Tools for Teams». <https://swagger.io/>
- [22] «Git». <https://git-scm.com/>
- [23] «Microsoft Teams - Videoconferenze, riunioni, chiamate». <https://www.microsoft.com/it-it/microsoft-teams/group-chat-software>



# Acronimi

**API** Application Programming Interface. 10, 23

**DOM** Document Object Model. 48

**DTO** Data Transfer Object. 28

**ER** Entity Relationship. 20

**HTML** HyperText MarkupLanguage. 9

**HTTP** HyperText Transfer Protocol. 10

**IDE** Integrated Development Environment. 10

**PDF** Portable Document Format. 7, 12, 21, 37

**SQL** Structured Query Language. 2

**UML** Unified Modeling Language. 23

**URL** Uniform Resource Locator. 10

# Glossario

**Access Token** Oggetto dei sistemi operativi Windows che contiene la sessione di accesso e identifica l'utente, i suoi privilegi e il gruppo di utenti al quale appartiene. [54](#)

**API REST** Un'API REST, nota anche come API RESTful, è un'interfaccia di programmazione delle applicazioni conforme ai vincoli dell'architettura REST. [8](#)

**Bearer Token** Un tipo particolare di [Access Token](#), sono usati per ottenere l'autorizzazione ad accedere ad una risorsa protetta da un Authorization Server conforme con lo standard OAuth2. [23](#)

**Espressione Regolare** sequenza di simboli che identifica un insieme di stringhe (testo). [2](#)

**Framework** Architettura logica di supporto sulla quale un software può essere progettato e realizzato. [3](#), [7](#), [9](#), [10](#), [48](#), [49](#)

**Javascript interoperability** La possibilità per un'applicazione Blazor di invocare metodi .NET e viceversa. [48](#)

**Placeholder** Letteralmente "Segnaposto", sono dei componenti, ad esempio testo o immagini, che vengono inseriti con lo scopo di riempire lo spazio destinato a del contenuto non ancora definito. [2](#), [45](#)

**Query** Interrogazione di una base dati solitamente per estrarre o aggiornare dati. [10](#)

**Server** Dispositivo che offre dei servizi agli altri dispositivi nella rete. [13](#)

**Software** Insieme di programmi che possono essere impiegati su un sistema di elaborazione. [1](#), [8](#)

**Template** Modello predefinito che consente di creare o inserire contenuti di diverso tipo. [7](#), [11](#)

**Tooltip** Messaggio a comparsa che fornisce informazioni o consigli sullo strumento. [45](#)

**Tupla** Una singola riga di una tabella della base di dati. [21](#)