

Documentazione progetto "Smart calendar for business"

Filippo Lazzari [1047143],
Mauro Riva [1053644],
Marco Rodolfi [1040347]

A.A. 2021/2022

1 Iterazione 0

1.1 Introduzione e utilizzo del sistema

Il progetto si pone l'obiettivo di organizzare i turni di un'azienda, specialmente per quelle attività che devono gestire turni a rotazione e che devono essere sempre presenti per il servizio clienti, come i servizi di ristorazione, per esempio. Nel progetto non è prevista la gestione di attività molto flessibili, tipicamente coordinate direttamente dai project manager.

L'idea è di realizzare tre componenti distinte all'interno dell'applicativo:

1. Una parte lato server che gestirà i client tramite chiamate REST e farà girare l'algoritmo per ottimizzare l'allocazione dei dipendenti sui turni.
2. Una parte web che gestisca la registrazione dell'azienda al servizio e fornisca un'interfaccia per il manager.
3. Un'applicativo per dispositivi Android che permetta ai dipendenti dell'azienda di visualizzare i propri turni e inserire malattie/ferie.

1.1.1 Il server dell'applicativo

Il server deve gestire più di una singola azienda e verrà amministrato direttamente da chi gestisce il servizio. Mette inoltre a disposizione le interfacce per consentire agli utenti di modificare i dati e visualizzare lo stato dell'azienda.

1.1.2 L'applicativo Android

Per permettere una maggiore flessibilità di gestione da parte dei dipendenti che lavorano, è stato pensato di creare un'applicativo Android che permetta di gestire il proprio stato all'interno dell'azienda, richiedere malattie e verificare i propri straordinari. La comunicazione verso il server è quindi gestita tramite chiamate REST.

1.1.3 L'algoritmo

L'algoritmo deve allocare dinamicamente i turni e gestire quelli scoperti verificando se dipendenti della stessa sede possono coprire il turno con degli straordinari oppure se personale di altre sedi della stessa azienda che possono svolgere il ruolo ha la possibilità di effettuare la trasferta.

1.2 Requisiti funzionali e casi d'uso

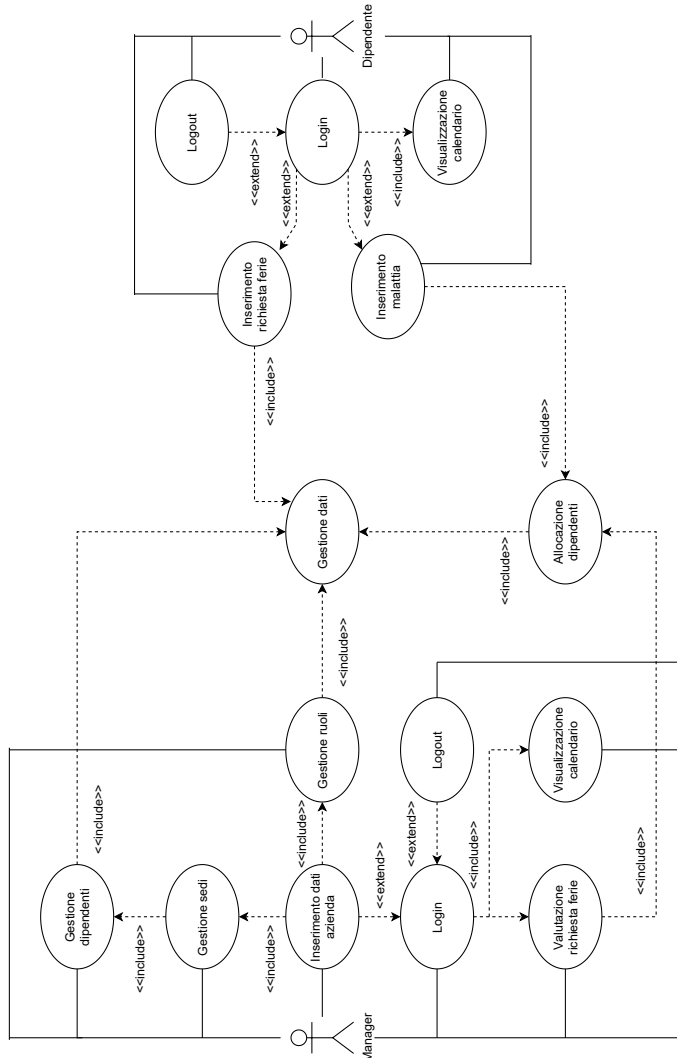


Figura 1: Casi d'uso pensati

La seguente tabella rappresenta i casi d'uso iniziali che sono stati ritenuti importanti da includere già all'inizio del progetto:

IDENTIFICATIVO	NOME FUNZIONE
UC1	Inserimento dati azienda
UC2	Login
UC3	Logout
UC4	Gestione ruoli
UC5	Gestione sedi
UC6	Gestione dipendenti
UC7	Algoritmo di allocazione dei dipendenti

Tabella 1: Alta priorità

Questa seconda tabella invece rappresenta i casi d'uso con priorità minore, cioè quelli da cui non dipendono i casi d'uso fondamentali.

IDENTIFICATIVO	NOME FUNZIONE
UC8	Visualizzazione calendario
UC9	Login e logout dipendenti
UC10	Inserimento malattia
UC11	Inserimento richiesta di ferie
UC12	Valutazione richiesta di ferie

Tabella 2: Media priorità

Infine questi sono casi d'uso che possono essere valutati in versioni future dell'applicazione:

IDENTIFICATIVO	NOME FUNZIONE
UC13	Personalizzazione calendario.
UC14	Visual. avanzata della situazione dei dip.
UC15	Limitazioni nell'inserimento delle ferie.
UC16	Modifica credenziali di accesso.

Tabella 3: Bassa priorità

1.3 Requisiti non funzionali

1.3.1 Usabilità

Per facilitare l'usabilità dell'applicazione è stato deciso di sviluppare un'app Android per i dipendenti e una web app per il manager dell'azienda. Entrambe le tipologie di client si connetteranno ad un server remoto.

1.3.2 Manutenibilità

Sono stati implementati nel codice diversi espedienti per facilitare una futura modifica dello stesso. Per esempio tramite attributi statici impiegati per rappresentare vari parametri sparsi per l'algoritmo oppure funzioni al momento molto simili tra di loro che non sono accorpate in vista di eventuali cambiamenti nelle prossime versioni.

1.3.3 Efficienza

L'efficienza dell'allocazione dei dipendenti sui turni è uno degli obiettivi del progetto. Anche l'efficienza dell'algoritmo è stata presa in considerazione ma nello stato attuale del progetto è di importanza secondaria. La progettazione dell'algoritmo ha comunque tenuto in considerazione la manutenibilità anche in ottica di miglioramenti di efficienza.

1.4 Topologia

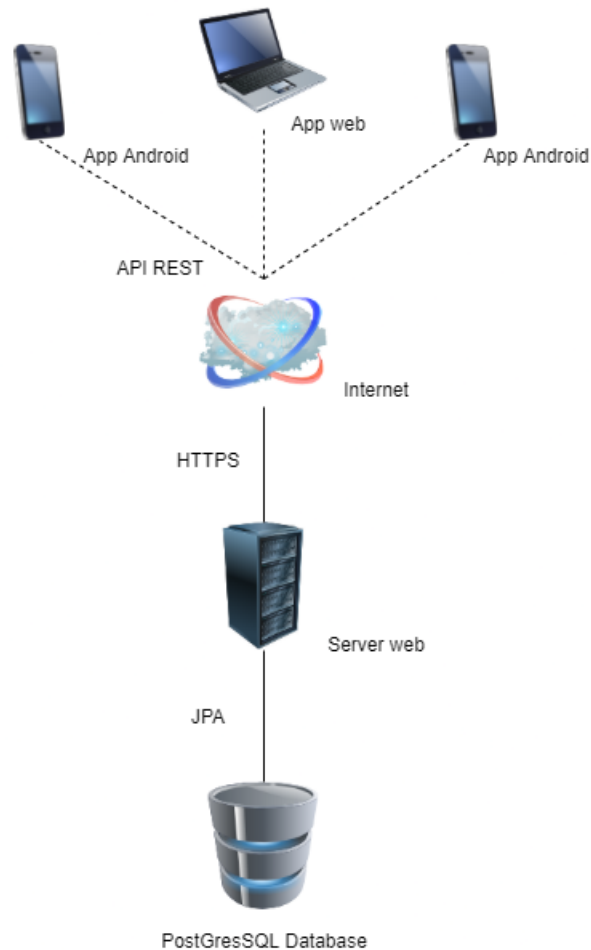


Figura 2: Topologia

La topologia dell'applicazione si divide in tre livelli:

- Il front-end che espone l'interfaccia grafica all'utente e acquisisce i dati in input. Fanno parte di questo livello l'app Android per i dipendenti, accessibile da smartphone e tablet, e l'app web per i manager, accessibile da qualsiasi browser.
- Il server web che espone le interfacce ai client consentendo l'interazione degli stessi con i dati memorizzati in remoto.
- Il database che si occupa di fornire la struttura dati persistente e garantire la consistenza e l'integrità dei dati.

1.5 Toolchain utilizzati

NOME TOOLCHAIN	UTILIZZO
IntelliJ IDEA Ultimate	Utilizzato come IDE di sviluppo per backend.
Android Studio	Utilizzato per lo sviluppo dell'applicativo Android.
Git & Gitlab	Software e piattaforma per gestire la distribuzione del codice sorgente.
CoCalc	Piattaforma utilizzata per la scrittura collaborativa della documentazione $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.
Java 17	Linguaggio di programmazione usato per il server e l'applicativo Android.
jQuery	Libreria JavaScript per applicazioni web.
Bootstrap	Framework CSS per lo sviluppo di pagine web.
Draw.io	Piattaforma utilizzata per creare diagrammi UML.
Postman	Software per il testing dinamico e la documentazione delle API.
Spring Boot	Il framework utilizzato lato server per il suo funzionamento.
Spring Data JPA	Interfaccia per l'astrazione della base di dati.
JGraphT	Libreria per i grafi utilizzata per lo sviluppo dell'algoritmo.
PostgreSQL	Il DB scelto per gestire la base di dati stessa.

Tabella 4: Toolchain Utilizzati

2 Iterazione 1

2.1 Introduzione

Nella prima interazione sono stati previsti i seguenti casi d'uso:

- UC1: Registrazione nuova azienda
- UC2: Login manager
- UC3: Logout manager
- UC4: Gestione ruoli
 - UC4.1: Inserimento nuovi ruoli
 - UC4.2: Modifica ruoli esistenti
 - UC4.3: Cancellazione ruolo
- UC5: Gestione sedi
 - UC5.1: Inserimento nuove sedi
 - UC5.2: Modifica sedi esistenti
 - UC5.3: Cancellazione sede
- UC6: Gestione dipendenti
 - UC6.1: Inserimento nuovi dipendenti
 - UC6.2: Modifica dati dipendenti
 - UC6.3: Cancellazione dipendente
- UC7: Algoritmo allocazione dipendenti su turni

2.2 UC1: Registrazione nuova azienda

Descrizione: Il manager dell'azienda inserisce le proprie credenziali registrando l'account con il quale andrà a gestire l'azienda. Per registrare l'azienda è richiesto l'inserimento della sede centrale, dei ruoli dei dipendenti, dell'orario di apertura e dei dipendenti della sede centrale.

Attori coinvolti: Manager, Sistema

Condizione di attivazione: Click del pulsante *registrati* da parte del manager

Postcondizione: Se la registrazione è avvenuta con successo, ossia se i dati sono stati caricati nel database, la pagina viene ricaricata e il manager può procedere con il login

Procedimento:

1. Il manager inserisce l'email, la password, il nome e il cognome
2. Dati azienda e sede centrale
3. Inserimento dei ruoli (almeno un ruolo)
4. Inserimento dei dipendenti della sede centrale (almeno un dipendente), un dipendente può avere al massimo 3 ruoli ed essere part-time
5. Selezione orario settimanale azienda

2.3 UC2: Login manager

Descrizione: Questa funzione è utilizzata per il login del manager all'interno dell'applicativo web. Per effettuare il login è necessario fornire email e password. Il server gestisce il login mediante la creazione di una sessione http.

Attori coinvolti: Manager, Sistema

Condizione di attivazione: Click del pulsante *login* da parte del manager dall'interfaccia web

Postcondizione: Se il manager ha inserito le credenziali corrette nel form di login accederà alla pagina principale dell'applicazione

Procedimento:

1. Click dell'utente sul tasto *login*
2. Inserimento email e password
3. Se le credenziali sono corrette il server crea la sessione http
4. Viene comunicato l'esito dell'operazione all'utente

2.4 UC3: Logout

Descrizione: Questa funzione è utilizzata per effettuare il logout del manager dalla pagina web. Il server distrugge la sessione http.

Attori coinvolti: Manager, Sistema

Condizione di attivazione: Click sul pulsante *logout*

Postcondizione: Sessione http distrutta

Procedimento:

1. Click dell'utente sul tasto *logout*
2. Distruzione della sessione
3. Aggiornamento della pagina

2.5 UC4: Gestione ruoli

Descrizione: L'applicazione deve permettere al manager di inserire, modificare ed eliminare i ruoli dal database. Un ruolo è caratterizzato dal nome e dal numero minimo di dipendenti necessari.

Attori coinvolti: Manager, Sistema

Condizione di attivazione: Selezione da parte del manager della voce nel menù *Ruoli*

Postcondizione: Un alert segnala all'utente l'avvenuta modifica del database

2.5.1 UC 4.1: Inserimento nuovi ruoli

Descrizione: Il manager inserisce il nome del ruolo e il numero richiesto di dipendenti.

Attori coinvolti: Manager, Sistema

Condizione di attivazione: Nella pagina *gestione ruoli* click su *aggiungi*

Postcondizione: Se il nuovo ruolo è stato inserito correttamente nel database comunica all'utente che l'operazione è avvenuta con successo, errore altrimenti

Procedimento:

1. L'utente ha effettuato il login
2. Accesso alla pagina *Ruoli* dal menu
3. Selezione della modalità *aggiunta*
4. Inserimento del nome del ruolo e del numero minimo di dipendenti
5. Click su *conferma*
6. Segnalazione del risultato dell'operazione di inserimento

2.5.2 UC4.2: Modifica ruoli esistenti

Descrizione: Il manager seleziona uno dei ruoli già presenti nel database e modifica il nome o il numero dei dipendenti. Il nuovo nome non deve causare conflitti con quelli degli altri ruoli dell'azienda. La variazione del numero minimo di dipendenti causa l'invalidazione di tutti i turni dell'azienda calcolati per date future.

Attori coinvolti: Manager, Sistema

Condizione di attivazione: Nella pagina *gestione ruoli* click su *modifica*

Postcondizione: Se il ruolo è stato modificato correttamente nel database comunica all'utente che l'operazione è avvenuta con successo, errore altrimenti

Procedimento:

1. L'utente ha effettuato il login
2. Accesso alla pagina *Ruoli* dal menu
3. Selezione della modalità *modifica*
4. Scelta del ruolo da modificare
5. Inserimento del nome del ruolo e del numero minimo di dipendenti
6. Click su *conferma*
7. Segnalazione del risultato dell'operazione di modifica

2.5.3 UC4.3: Cancellazione ruolo

Descrizione: Il manager seleziona uno dei ruoli già presenti nel database e lo elimina. Questa operazione deve tenere in considerazione le relazioni su database dei ruoli con i turni e con i dipendenti.

Attori coinvolti: Manager, Dipendenti, Sistema

Condizione di attivazione: Nella pagina *gestione ruoli* click su *cancella*

Postcondizione: Se il ruolo è stato eliminato correttamente dal database comunica all'utente che l'operazione è avvenuta con successo, errore altrimenti

Procedimento:

1. L'utente ha effettuato il login
2. Accesso alla pagina *Ruoli* dal menu
3. Selezione della modalità *cancella*
4. Scelta del ruolo da eliminare
5. Click su *conferma*
6. Segnalazione del risultato dell'operazione di cancellazione

2.6 UC5: Gestione sedi

Descrizione: L'applicazione deve permettere al manager di inserire, modificare ed eliminare le sedi dell'azienda dal database. Una sede è identificata dal proprio indirizzo e ha dei dipendenti assegnati (almeno uno). Una sede può inoltre avere indicata la sede a lei più vicina e la relativa distanza.

Attori coinvolti: Manager, Sistema

Condizione di attivazione: Selezione da parte del manager della voce nel menu *Sedi*

Postcondizione: Un alert segnala all'utente l'avvenuta modifica del database

2.6.1 UC5.1: Inserimento nuove sedi

Descrizione: Il manager inserisce l'indirizzo della nuova sede ed eventualmente seleziona la sede più vicina. La distanza tra le sedi è espressa in chilometri. Alla sede sono associati uno o più dipendenti

Attori coinvolti: Manager, Sistema

Condizione di attivazione: Nella pagina *gestione sedi* click su *aggiungi*

Postcondizione: Se la nuova sede è stata inserita correttamente nel database comunica all'utente che l'operazione è avvenuta con successo, errore altrimenti

Procedimento:

1. L'utente ha effettuato il login
2. Accesso alla pagina *Sedi* dal menu
3. Selezione della modalità *aggiungi*
4. Inserimento dell'indirizzo ed eventualmente della sede vicina con la distanza tra le due
5. Inserimento dei dipendenti con nome, cognome, ruoli ed eventualmente se il contratto è di tipo part-time
6. Click su *conferma*
7. Segnalazione del risultato dell'operazione di inserimento

2.6.2 UC5.2: Modifica sedi esistenti

Descrizione: Il manager seleziona una delle sedi già presenti nel database e ne modifica l'indirizzo o la sede più vicina. La variazione della sede vicina comporta l'invalidazione dei turni in trasferta calcolati per date future.

Attori coinvolti: Manager, Dipendenti, Sistema

Condizione di attivazione: Nella pagina *gestione sedi* click su *modifica*

Postcondizione: Se la sede è stata modificata correttamente nel database comunica all'utente che l'operazione è avvenuta con successo, errore altrimenti.

Procedimento:

1. L'utente ha effettuato il login
2. Accesso alla pagina *Sedi* dal menu
3. Selezione della modalità *modifica*
4. Scelta della sede da modificare
5. Selezione di una nuova sede vicina (eventualmente anche nessuna) e inserimento della distanza
6. Click su *conferma*
7. Segnalazione del risultato dell'operazione di modifica

2.6.3 UC5.3: Cancellazione sede

Descrizione: Il manager seleziona una delle sedi già presenti nel database e la elimina. Questa operazione deve tenere conto di eventuali trasferte pianificate per date future.

Attori coinvolti: Manager, Dipendenti, Sistema

Condizione di attivazione: Nella pagina *gestione sedi* click su *cancella*

Postcondizione: Se la sede è stata eliminata correttamente dal database comunica all'utente che l'operazione è avvenuta con successo, errore altrimenti

Procedimento:

1. L'utente ha effettuato il login
2. Accesso alla pagina *Sedi* dal menu
3. Selezione della modalità *cancella*
4. Scelta della sede da eliminare
5. Click su *conferma*
6. Segnalazione del risultato dell'operazione di cancellazione

2.7 UC6: Gestione dipendenti

Descrizione: L'applicazione deve permettere al manager di inserire, modificare ed eliminare i dipendenti dislocati in tutte le sedi dell'azienda dal database. Un dipendente è identificato dalla propria email e può svolgere da uno fino al massimo di tre ruoli diversi.

Attori coinvolti: Manager, Dipendenti, Sistema

Condizione di attivazione: Selezione da parte del manager della voce nel menu *Dipendenti*

Postcondizione: Un alert segnala all'utente l'avvenuta modifica del database

2.7.1 UC6.1: Inserimento nuovi dipendenti

Descrizione: Il manager inserisce il nome, il cognome, i ruoli e il tipo di contratto del dipendente (se full-time o part-time) e seleziona la sede a cui assegnarlo. L'email viene generata automaticamente dal server nel formato nome.cognome (ed eventualmente un numero, in caso di omonimie) seguito dal suffisso aziendale. La password è creata dal server e comunicata tramite l'email all'utente che dovrà poi provvedere a modificarla.

Attori coinvolti: Manager, Dipendente, Sistema

Condizione di attivazione: Nella pagina *gestione dipendenti* click su *aggiungi*

Postcondizione: Se il nuovo dipendente è stato inserito correttamente nel database comunica all'utente che l'operazione è avvenuta con successo, errore altrimenti

Procedimento:

1. L'utente ha effettuato il login
2. Accesso alla pagina *Dipendenti* dal menu
3. Selezione della modalità *aggiunta*
4. Inserimento del nome, del cognome, dei ruoli e della modalità del contratto
5. Click su *conferma*
6. Segnalazione del risultato dell'operazione di inserimento

2.7.2 UC6.2: Modifica dati dipendenti

Descrizione: Il manager seleziona un dipendente presente nel database e ne modifica il tipo di contratto o i ruoli che può svolgere. La variazione di questi parametri comporta l'invalidazione dei turni associati all'utente in date future.

Attori coinvolti: Manager, Dipendente, Sistema

Condizione di attivazione: Nella pagina *gestione dipendenti* click su *modifica*

Postcondizione: Se i dati del dipendente sono stati modificati correttamente sul database comunica all'utente che l'operazione è avvenuta con successo, errore altrimenti

Procedimento:

1. L'utente ha effettuato il login
2. Accesso alla pagina *Dipendenti* dal menu
3. Selezione della modalità *modifica*
4. Scelta della sede per visualizzarne tutti i dipendenti o ricerca diretta tramite email
5. Modifica dei ruoli e del contratto
6. Click su *conferma*
7. Segnalazione del risultato dell'operazione di modifica

2.7.3 UC6.3: Cancellazione dipendenti

Descrizione: Il manager seleziona un dipendente presente nel database e lo elimina. Questa operazione invalida i turni associati all'utente in date future

Attori coinvolti: Manager, Dipendente, Sistema

Condizione di attivazione: Nella pagina *gestione dipendenti* click su *cancella*

Postcondizione: Se il dipendente è stato eliminato correttamente dal database comunica all'utente che l'operazione è avvenuta con successo, errore altrimenti

Procedimento:

1. L'utente ha effettuato il login
2. Accesso alla pagina *Dipendenti* dal menu
3. Selezione della modalità *cancella*
4. Scelta della sede per visualizzarne tutti i dipendenti o ricerca diretta tramite email
5. Click su *conferma*
6. Segnalazione del risultato dell'operazione di cancellazione

2.8 UC7: Algoritmo allocazione dipendenti su turni

Descrizione: L'obiettivo dell'algoritmo è allocare i dipendenti ai turni in modo da non lasciare turni scoperti. Il calcolo avviene sull'intera azienda per un determinato mese. Se il mese dato in input è quello attuale verranno calcolati solo i turni dalla data odierna fino alla fine del mese, non è possibile modificare i turni nel passato.

Attori coinvolti: Manager, Dipendente, Sistema

Condizione di attivazione: Nella pagina *calendario* selezionare il mese e l'anno e click su *ricalcola*

Postcondizione: Se i turni sono stati calcolati correttamente comunica all'utente se tutti i turni sono stati assegnati o se ne è rimasto almeno uno scoperto, errore altrimenti

Procedimento:

1. L'utente ha effettuato il login
2. Accesso alla pagina *Calendario* dal menu
3. Selezione del mese e dell'anno
4. Click su *ricalcola*
5. Segnalazione del risultato dell'esecuzione dell'algoritmo

2.9 Algoritmo

L'algoritmo alloca i dipendenti ai turni di tutta l'azienda, per il mese dato in input dal manager. L'azienda è caratterizzata dall'orario di apertura, da i ruoli e dalle sedi. L'algoritmo è suddiviso in diverse sezioni:

1. Inizializzazione
2. Cancellazione vecchi turni dello stesso mese e creazione dei nuovi
3. Allocazione dei dipendenti
4. Allocazione dei turni rimasti scoperti tramite straordinari e trasferte

2.9.1 Inizializzazione

In questa prima fase sono inizializzate le variabili ed è calcolato il periodo temporale di riferimento: se il mese su cui l'algoritmo dovrà operare è nel futuro allora la data di inizio è il giorno uno e la data di fine è l'ultimo giorno di quel mese. Se invece il mese è quello attuale la data di inizio è il giorno corrente, così che i turni dei giorni precedenti non vengano modificati. Per ogni dipendente dell'azienda sono inizializzate a zero due *HashMap* che mettono in relazione con i dipendenti le ore lavorate quel mese e le ore di straordinario effettuate.

Inizializzazione

```
46 public boolean calculateBestTurni(Azienda az, int anno, int mese){
47
48     List<Sede> sedi = sedeRep.findSedeByAzienda(az);
49     List<Ruoli> ruoli = ruoliRep.getRuoliByRuoloID_Azienda(az);
50     Orario orario = orarioRep.findOrarioByAzienda(az);
51     boolean assegnamentoCompleto = true;
52
53     HashMap<Sede, List<Turno>> turniPerSede = new HashMap<>();
54     HashMap<Utente, Integer> utentiOreMensili = new HashMap<>();
55     HashMap<Utente, Integer> utentiOreStraordinario = new
    ↪ HashMap<>();
56
57     //Inizializzo le ore lavorate questo mese
58     List<Utente> uteList = utRep.findUtenteByAzienda(az);
59     for(Utente u : uteList){
60         utentiOreMensili.put(u, 0);
61         utentiOreStraordinario.put(u, 0);
62     }
63
64     //Calcolo periodo temporale
65     LocalDate startTime = LocalDate.of(anno, mese, 1);
66     //Se devo calcolare solo la parte restante del mese attuale
    ↪ (non posso modificare il passato!)
67     if(startTime.isBefore(LocalDateTime.now().toLocalDate()))
68         startTime = LocalDate.now();
69
70     LocalDate endTime = startTime.plusMonths(1);
71     endTime = endTime.withDayOfMonth(1);
```

Questo ha un costo di $O(U)$, U numero di dipendenti dell'azienda.

2.9.2 Cancellazione vecchi turni dello stesso mese e creazione dei nuovi

Nella seconda fase è necessario rimuovere dal database tutti i turni e gli straordinari che sono stati calcolati precedentemente per quel mese, a partire dalla data di inizio. Fatto ciò si può procedere a ricreare turni "vuoti", senza quindi dipendenti allocati: per ogni giorno compreso tra la data di inizio e la data di fine l'orario di apertura è suddiviso in turni da quattro ore (o meno, se le ore di apertura non sono un multiplo di quattro) per ogni ruolo presente nell'azienda. Ai turni sono assegnate un numero di ore scoperte pari alla durata del turno moltiplicata per il numero minimo di dipendenti necessari per quel ruolo. Quindi su un turno di quattro ore per tre dipendenti le ore da allocare saranno dodici. L'ultima operazione di questa fase è la ricerca nel database di eventuali turni di questo mese precedenti alla data di inizio e l'aggiornamento dei contatori delle ore lavorate e degli straordinari dei dipendenti.

Cancellazione vecchi turni dello stesso mese e creazione dei nuovi

```
46 //Per ogni sede, recupero tutti i turni disponibili.
47 for (Sede sede : sedi) {
48     List <Turno> lista =
49         ↳ turnoRep.findTurnoBySedeAndDataGreaterThanOrEqualAndDataLessThanOrderByDataDesc(sede,
50         ↳ endTime);
49 //Rimuovo eventuali turni (con annessi straordinari) già
50     ↳ presenti nel database
51 for(Turno t : lista){
52     if((t.getData().isBefore(ChronoLocalDate.from(endTime)) &&
53     ↳ t.getData().isAfter(ChronoLocalDate.from(startTime))) ||
54     ↳ t.getData().isEqual(ChronoLocalDate.from(startTime))){
55         turnoUtenteRep.deleteByTurno(t);
56         straRep.deleteByTurno(t);
57         turnoRep.delete(t);
58     }
59     else
60         ↳ if((t.getData().isBefore(ChronoLocalDate.from(endTime))))
61             break;
62 }
63 //Creo i turni a cui dovranno essere assegnati i dipendenti
64 List <Turno> listaTurni = new ArrayList<>();
65 for(Ruoli r : ruoli) {
66     if (!r.getAdmin()){
67         LocalDate time = LocalDate.from(startTime);
68         while (time.isBefore(endTime)) {
69             if (orario.isOpenForCurrentDay(time)) {
70                 Giorno giorno =
71                 ↳ orario.getCorrespondingWorkingHours(time);
72                 LocalDate data = LocalDate.from(time);
73                 int oreAperturaRimaste = giorno.getChiusura() -
74                 ↳ giorno.getApertura();
75                 int apertura = giorno.getApertura();
76                 int chiusura;
77                 //FIXME: Eventualmente da aggiungere la pausa
78                 ↳ pranzo
79                 while (oreAperturaRimaste > 0) {
80                     if (oreAperturaRimaste < durataTurno)
81                         chiusura = apertura +
82                         ↳ oreAperturaRimaste;
83                     else
84                         chiusura = apertura + durataTurno;
85                 Turno turno = new Turno(sede,
86                 ↳ r.getRuoloID(), data, apertura,
87                 ↳ chiusura);
88                 //Definisco le ore-dipendente necessarie per
89                 ↳ coprire il turno
90                 turno.setOreScoperte((chiusura - apertura) *
91                 ↳ r.getMinRuolo());
92                 listaTurni.add(turno);
93                 apertura = chiusura;
94                 oreAperturaRimaste -= durataTurno;
95             }
96         }
97     }
98 }
```


2.9.3 Allocazione dei dipendenti

Per ogni sede e per ogni ruolo sono create tre liste di dipendenti per gestire la priorità del ruolo: ogni dipendente può essere in grado di svolgere al massimo tre ruoli, si considera l'ordine di inserimento dei ruoli durante la registrazione del dipendente come indicazione della priorità in ordine decrescente. Per esempio, dato un dipendente con ruoli aiuto cuoco, cameriere e cassiere, esso sarà inserito nella lista principale per il ruolo aiuto cuoco, nella secondaria per cameriere e nella terziaria per cassiere. Una volta completate queste liste si procede all'allocazione dei dipendenti per tutti i turni della sede: per ogni turno l'algoritmo prova ad allocare i dipendenti nella lista principale del ruolo, se non ci riesce prova con quelli nella secondaria e infine nella terziaria. Se il turno è rimasto comunque scoperto viene inserito in una lista di turni scoperti che verrà gestita nella prossima fase. Un dipendente per essere considerato valido per un turno deve soddisfare diversi requisiti: Prima di tutto non deve aver lavorato nello stesso giorno più di otto ore (o quattro per i contratti part-time) e le ore rimaste fino al raggiungimento di questa soglia devono essere sufficienti a coprire interamente il turno, ciò dal suo inizio alla sua fine. Se questa prima selezione viene superata si deve controllare la presenza di sovrapposizioni con altri turni già assegnati e con periodi di malattia/ferie presenti nel database.

2.9.4 Allocazione dei turni rimasti scoperti tramite straordinari e trasferte

Questa fase viene eseguita solo se c'è almeno un turno rimasto scoperto nell'intera azienda. Per ogni turno scoperto si prova a completarlo facendo fare straordinari ai dipendenti della stessa sede che possono svolgere quel ruolo. A differenza dell'assegnazione "standard" uno straordinario può avere una durata inferiore a quella del turno. Un dipendente può effettuare al massimo due ore di straordinari al giorno e dieci al mese, per i contratti part-time questi limiti sono dimezzati. Se anche con gli straordinari non è stato possibile coprire il turno scoperto allora si procede con il valutare le trasferte di dipendenti dalle sedi vicine. La vicinanza delle sedi è gestita tramite un grafo pesato non orientato, dove il peso degli archi è la distanza in chilometri tra i nodi (ossia tra le sedi). Per trovare tutte le sedi vicine a quella del turno scoperto viene eseguita una visita in ampiezza del grafo che ritorna una lista con tutte le sedi distanti non più di cinquanta chilometri. Questa lista viene poi ordinata in modo crescente per distanza. Fatto ciò l'algoritmo prova ad assegnare il turno ai dipendenti delle sedi nella lista che possono svolgere il ruolo. Un dipendente per essere idoneo alla trasferta non deve avere turni assegnati lo stesso giorno in sedi diverse da quella di destinazione e deve poter coprire il turno completamente, dall'inizio alla fine.

3 Iterazione 2

3.1 Introduzione

3.2 UC8: Visualizzazione Calendario

Descrizione:

Attori coinvolti:

Condizione di attivazione:

Postcondizione:

Procedimento:

3.3 UC9: Login e logout dipendenti

Descrizione:

Attori coinvolti:

Condizione di attivazione:

Postcondizione:

Procedimento:

3.4 UC10: Inserimento malattia

Descrizione:

Attori coinvolti:

Condizione di attivazione:

Postcondizione:

Procedimento:

3.5 UC11: Inserimento ferie

Descrizione:

Attori coinvolti:

Condizione di attivazione:

Postcondizione:

Procedimento:

3.6 UC12: Valutazione richieste di ferie

Descrizione:

Attori coinvolti:

Condizione di attivazione:

Postcondizione:

Procedimento: