



UNIVERSITÀ DEGLI STUDI DI NAPOLI  
**FEDERICO II**

**Corso di laurea in Ingegneria Informatica (Classe di Laurea L-8)**  
**Corso di Ingegneria del Software**

Elaborazione di un progetto per:  
**GESTIONE DI UN CALL CENTER**

**Prof.ssa:**

Fasolino Anna Rita

**Studenti:**

Falco Giovanni N46004975

Mario Formato N46005548

Salvatore Emanuele Di Cicco N46004863

1 SPECIFICHE INFORMALI .....	4
2 ANALISI E SPECIFICA DEI REQUISITI .....	5
2.1 Analisi nomi-verbi .....	5
2.2 Revisione dei requisiti .....	5
2.3 Glossario dei termini .....	6
2.4 Classificazione dei requisiti .....	7
2.4.1 Requisiti funzionali .....	7
2.4.2 Requisiti sui dati .....	8
2.4.3 Vincoli/Altri requisiti .....	8
2.5 Modellazione dei casi d'uso .....	9
2.5.1 Attori e casi d'uso .....	9
2.5.2 Diagrammi dei casi d'uso .....	10
2.5.3 Scenari .....	10
2.6 Diagrammi delle classi .....	14
Diagrammi delle classi di analisi: .....	14
Diagramma delle classi raffinato: .....	15
2.7 DIAGRAMMI DI SEQUENZA .....	15
2.7.1 Creazione lista numeri telefonici: .....	15
2.7.2 Modifica lista numeri telefonici (scenario normale) .....	16
2.7.3 Creazione gruppo centralinista (Scenario normale) .....	18
2.7.4 Modifica Gruppi Centralinisti (Scenario normale) .....	19
2.7.5 Assegnazione Lista .....	20
2.7.6 Chiamata Utenza .....	20
2.7.7 Creazione Appuntamento .....	21
2.7.8 Interazione con lista appuntamenti .....	22
2.7.9 Visualizza note appuntamento fissato .....	23
2.8 VERIFICA DELLA COMPLETEZZA DEI REQUISITI .....	23
3. STIMA DEI COSTI .....	24
3.1 CREAZIONE LISTA NUMERI TELEFONICI .....	24
3.2 MODIFICA LISTA NUMERI TELEFONICI .....	25
3.3 CREAZIONE GRUPPI CENTRALINISTI .....	27
3.4 MODIFICA GRUPPI CENTRALINISTI .....	28
3.5 ASSEGNA LISTA .....	29

3.6 CHIAMATA UTENZA.....	30
3.7 INTERAZIONE CON LISTA APPUNTAMENTI .....	31
3.8 CREAZIONE APPUNTAMENTO.....	32
3.9 VISUALIZZA NOTE APPUNTAMENTO FISSATO .....	33
4. TEST FUNZIONALE .....	35
4.1 Assegnazione Lista .....	35
4.2 Registrazione Esito .....	37
4.3 Rimuovi Utente .....	38
4.4 Aggiungi utente .....	39
4.5 Aggiungi centralista.....	41
4.6 Rimuovi centralista .....	43
4.7 Visualizza note appuntamenti fissati .....	44
4.8 Rimuovi Lista .....	45
4.9 Rimuovi Gruppo .....	46
4.10 Modifica note appuntamenti .....	47
4.11 Creazione Appuntamento .....	48
5. PROGETTAZIONE .....	50
5.1 DIAGRAMMA DELLE CLASSI CON PATTERN ARCHITETTURALI BCED .....	50
5.2 DIAGRAMMA DI SEQUENZA.....	50
5.2.1 Assegna Lista .....	50
5.2.2 Interazione con lista appuntamenti .....	51
5.2.3 Creazione Appuntamento .....	51
6 . IMPLEMENTAZIONE .....	51
6.1 ARTICOLAZIONE DEL SOFTWARE .....	51
Package Boundary.....	52
Package Controller .....	52
Package Entity .....	52
Package Database .....	52
6.2 STRUMENTI NECESSARI.....	53
7. TESTING.....	53
7.1 Test Strutturale .....	53
7.2 Test Funzionale .....	60
7.2.1 Assegnazione lista .....	60

## 1 SPECIFICHE INFORMALI

### **Sistema di gestione call center**

Si vuole realizzare un sistema informatico per la gestione di un call center.

Il sistema è utilizzato da amministratori e centralinisti. Gli amministratori creano liste di numeri telefonici e aggiungono o rimuovono numeri telefonici dalle liste. Ogni numero telefonico è costituito da 10 cifre.

I centralinisti, identificati da un id, nome, cognome ed e-mail sono organizzati in gruppi. Ciascun gruppo è identificato da un id e una descrizione. È compito degli amministratori creare o rimuovere i gruppi. La creazione di un gruppo consiste nell'inserimento di almeno un centralinista.

Ciascun amministratore ha inoltre il compito di assegnare ciascuna lista di numeri al più a un gruppo di centralinisti.

I centralinisti hanno il compito di chiamare i numeri della lista ad essi assegnata. A tal fine essi richiedono al sistema il prossimo numero da chiamare e, al termine della telefonata, registrano l'esito. In particolare, per ogni telefonata memorizzano l'id, la data, l'ora, note e l'esito (occupato, senza risposta, da richiamare, non interessato, appuntamento fissato). In caso di appuntamento fissato, il centralinista crea un nuovo appuntamento tra uno degli agenti vendita del call center (identificato mediante Codice Fiscale) e l'utenza chiamata. L'appuntamento contiene un id, data e ora e note relative all'incontro.

Gli agenti vendita interagiscono con il sistema per ottenere informazioni relative a tutti gli appuntamenti fissati e modificare le note degli appuntamenti.

A partire dagli appuntamenti, gli agenti vendita possono risalire alle note che il centralinista ha scritto al termine della chiamata conclusasi con un appuntamento fissato. Un appuntamento, inoltre, può essere collegato ad un altro: quando il centralinista fissa un nuovo appuntamento che segue uno fallito (es. l'utente non era disponibile al momento dell'appuntamento con l'agente), il nuovo deve referenziare il precedente.

*Lo studente dovrà consegnare un archivio zip nominato "COGNOME\_NOME\_MATRICOLA", contenente i*

*seguenti file, ognuno in una cartella differente:*

- 1) file .doc e .pdf del progetto, secondo il facsimile inviato dal docente per email,*
- 2) file .vpp (o .mdj) contenente l'intero progetto Visual Paradigm,*
- 3) cartella contenente il progetto Eclipse con l'implementazione della funzionalità richiesta.*

*Per effettuare la consegna, lo studente dovrà inviare la cartella compressa contenente tutti i file tramite il servizio FileSender di Ateneo (<https://filesender.unina.it/>) specificando come destinatario: fasolino@unina.it*

*La consegna dell'elaborato andrà effettuata entro il giorno 15/06/2023.*

*Per eventuali chiarimenti sulla traccia, è possibile contattare la docente tramite chat di Teams.*

## 2 ANALISI E SPECIFICA DEI REQUISITI

### 2.1 Analisi nomi-verbi

- Classe
- Attributo
- Funzionalità
- Attore
- Classe-Attore

Si vuole realizzare un sistema informatico per la gestione di un call center.

Il sistema è utilizzato da amministratori e centralinisti. Gli amministratori creano liste di numeri telefonici e aggiungono o rimuovono numeri telefonici dalle liste. Ogni numero telefonico è costituito da 10 cifre. I centralinisti, identificati da un id, nome, cognome ed e-mail sono organizzati in gruppi. Ciascun gruppo è identificato da un id e una descrizione. È compito degli amministratori creare o rimuovere i gruppi. La creazione di un gruppo consiste nell'inserimento di almeno un centralinista.

Ciascun amministratore ha inoltre il compito di assegnare ciascuna lista di numeri al più a un gruppo di centralinisti.

I centralinisti hanno il compito di chiamare i numeri della lista ad essi assegnata. A tal fine essi richiedono al sistema il prossimo numero da chiamare e, al termine della telefonata, registrano l'esito. In particolare, per ogni telefonata memorizzano l'id, la data, l'ora, note e l'esito (occupato, senza risposta, da richiamare, non interessato, appuntamento fissato). In caso di appuntamento fissato, il centralinista crea un nuovo appuntamento tra uno degli agenti vendita del call center (identificato mediante Codice Fiscale) e l'utenza chiamata. L'appuntamento contiene un id, data e ora e note relative all'incontro.

Gli agenti vendita interagiscono con il sistema per ottenere informazioni relative a tutti gli appuntamenti fissati e modificare le note degli appuntamenti.

A partire dagli appuntamenti, gli agenti vendita possono risalire alle note che il centralinista ha scritto al termine della chiamata conclusasi con un appuntamento fissato. Un appuntamento, inoltre, può essere collegato ad un altro: quando il centralinista fissa un nuovo appuntamento che segue uno fallito (es. l'utente non era disponibile al momento dell'appuntamento con l'agente), il nuovo deve referenziare il precedente.

### 2.2 Revisione dei requisiti

1. Gli Amministratori creano liste di numeri telefonici.
2. Gli Amministratori aggiungono numeri telefonici dalle liste.
3. Gli Amministratori rimuovono numeri telefonici dalle liste.
4. Ogni numero è costituito da 10 cifre.
5. I Centralinisti sono identificati da un id , nome , cognome ed e-mail.
6. I Centralinisti sono organizzati in gruppi.
7. Ciascun gruppo è identificato da un id e una descrizione.
8. Gli Amministratori creano gruppi di centralinisti.

9. Gli Amministratori rimuovono gruppi di centralinisti.
10. La creazione di un gruppo consiste nell'inserimento di almeno un centralinista.
11. Un gruppo deve contenere almeno un centralinista.
12. Gli Amministratori assegnano a ciascuna lista di numeri al più a un gruppo di centralinisti.
13. Ogni lista di numeri può essere assegnata al più ad un gruppo.
14. I Centralinisti chiamano i numeri della lista.
15. I Centralinisti richiedono al sistema al termine della chiamata deve essere in grado di passare il numero successivo da chiamare.
16. La telefonata è identificata da id , data , ora note ed esito (occupato, senza risposta, da richiamare, non interessato, appuntamento fissato).
17. Il Centralinista al termine della chiamata deve registrare l'esito della chiamata
18. Il Centralinista deve essere in grado di creare un appuntamento tra l'agente vendita e l'utente.
19. L'agente vendita è identificato mediante Codice Fiscale.
20. L'appuntamento è identificato da un id, data e ora e note relative all'incontro.
21. Gli Agenti vendita interagiscono con il sistema per ottenere informazioni relative a tutti gli appuntamenti fissati e modificare le note degli appuntamenti.
22. Gli agenti vendita interagiscono con il sistema per modificare gli appuntamenti fissati.
23. Gli agenti vendita possono risalire alle note che il centralinista ha scritto al termine della chiamata conclusasi con un appuntamento fissato.
24. Un appuntamento, inoltre, può essere collegato ad un altro: quando il centralinista fissa un nuovo appuntamento che segue uno fallito (es. l'utente non era disponibile al momento dell'appuntamento con l'agente), il nuovo deve referenziare il precedente.

## 2.3 Glossario dei termini

Termine	Descrizione	Sinonimi
Amministratori	Si occupano di gestire il gruppo di centralinisti, e l'elenco di numeri associato ad essi.	
Centralinisti	Si occupano di gestire gli appuntamenti tra gli utenti e gli agenti chiamata.	
Agenti di vendita	Si occupano di contrattare con l'utente.	
Utenza chiamata	Colui che viene contattato dagli agenti vendita.	Utente chiamato
Appuntamento	Incontro tra agente vendita e utente	
Gruppo	Insieme di centralinisti	
Lista	Insieme di numeri telefonici	

## 2.4 Classificazione dei requisiti

### 2.4.1 Requisiti funzionali

ID	REQUISITO	ORIGINE (n. frase dei requisiti revisionati)
RF01	Il sistema deve offrire agli amministratori una funzionalità per creare liste di numeri telefonici.	2
RF02	Il sistema deve offrire agli amministratori una funzionalità per aggiungere numeri telefonici dalle liste.	3
RF03	Il sistema deve offrire agli amministratori una funzionalità per rimuovere numeri telefonici dalle liste.	3
RF04	Il sistema deve offrire agli amministratori una funzionalità per creare gruppi di centralinisti.	5
RF05	Il sistema deve offrire agli amministratori una funzionalità per rimuovere gruppi di centralinisti.	5
RF06	Il sistema deve offrire agli amministratori una funzionalità per aggiungere il centralinista ad un gruppo	6
RF07	Il sistema deve offrire agli amministratori una funzionalità per rimuovere il centralinista ad un gruppo	6
RF08	Il sistema deve offrire agli amministratori una funzionalità per assegnare a ciascuna lista di numeri al più a un gruppo di centralinisti.	8
RF09	Il sistema deve offrire ai centralinisti una funzionalità per chiamare i numeri della lista.	9
RF10	Il sistema deve offrire ai centralinisti una funzionalità per poter richiedere un numero successivo da chiamare al termine della chiamata.	9
RF11	Il sistema deve offrire ai centralinisti una funzionalità per registrare l'esito della chiamata.	12
RF12	Il sistema deve offrire ai centralinisti una funzionalità per essere in grado di creare un appuntamento tra l'agente vendita e l'utente.	14
RF13	Il sistema deve offrire agli agenti vendita una funzionalità per interagire con il sistema per ottenere informazioni relative a tutti gli appuntamenti fissati e modificare le note degli appuntamenti.	17

<b>RF14</b>	Il sistema deve offrire agli agenti vendita una funzionalità per poter risalire alle note che il centralinista ha scritto al termine della chiamata conclusasi con un appuntamento fissato.	19
-------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----

#### 2.4.2 Requisiti sui dati

ID	REQUISITO	ORIGINE (n. frase dei requisiti revisionati)
<b>RD01</b>	Ogni numero è costituito da dieci cifre.	3
<b>RD02</b>	I centralinisti sono identificati da un id, nome, cognome ed e-mail.	5
<b>RD03</b>	I centralinisti sono organizzati in gruppi.	5
<b>RD04</b>	Ciascun gruppo è identificato da un id ed una descrizione.	7
<b>RD06</b>	La telefonata è identificata da id, data, ora note ed esito (occupato, senza risposta, da richiamare, non interessato, appuntamento fissato).	12
<b>RD07</b>	L'agente vendita è identificato mediante Codice Fiscale.	15
<b>RD08</b>	L'appuntamento è identificato da un id, data e ora e note relative all'incontro.	15
<b>RD09</b>	Un appuntamento, inoltre, può essere collegato ad un altro: quando il centralinista fissa un nuovo appuntamento che segue uno fallito (es. l'utente non era disponibile al momento dell'appuntamento con l'agente), il nuovo deve referenziare il precedente.	20

#### 2.4.3 Vincoli/Altri requisiti

ID	REQUISITO	ORIGINE (n. frase dei requisiti revisionati)
<b>V01</b>	Un gruppo deve contenere almeno un centralinista.	7



## 2.5 Modellazione dei casi d'uso

### 2.5.1 Attori e casi d'uso

#### Attori primari:

- Amministratore
- Centralinista
- Agente vendita

#### Attori secondari:

- Utenza chiamata

#### Casi d'uso:

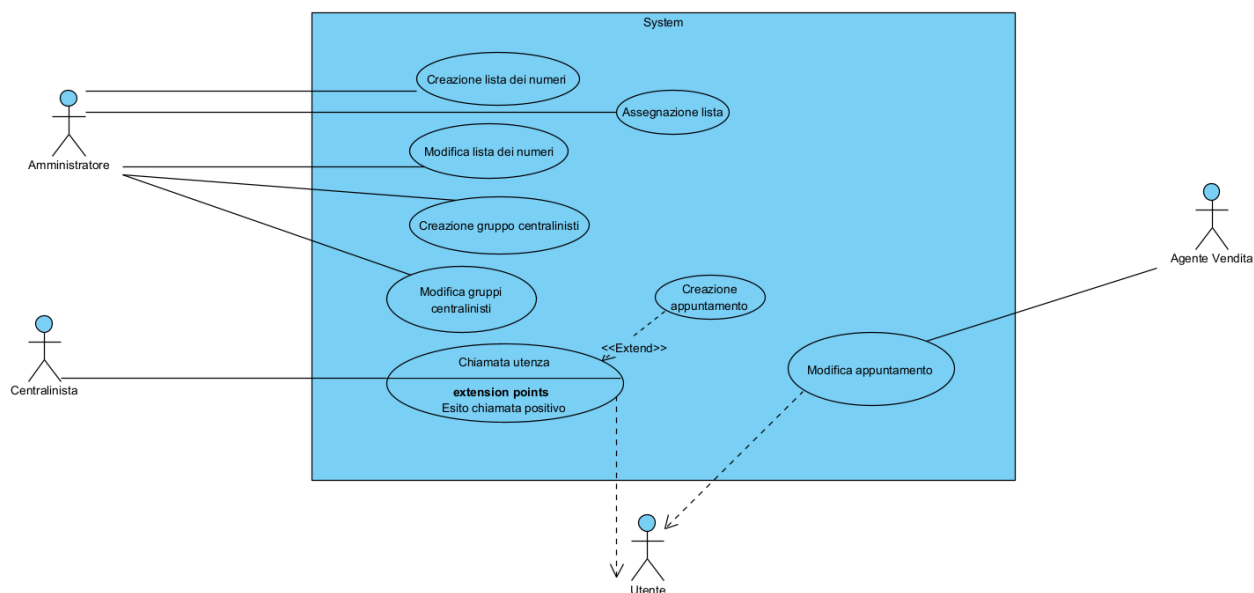
- UC1: Creazione lista di numeri telefonici -> RF01 -> FATTO
- UC2: Modifica lista di numeri telefonici -> RF02, RF03 -> FATTO
- UC3: Creazione gruppi centralinisti -> RF04, RF05 -> FATTO
- UC4: Modifica gruppi centralinisti -> RF06, RF07 -> FATTO
- UC5: Assegnazione lista a gruppi->RF08 -> FATTO
- UC6: Chiamata utenza -> RF09, RF10, RF11, RF12
- UC7: Interazione con lista appuntamenti -> RF13, RF14, RF15
- UC8: Visualizza note appuntamento fissato

#### Caso d'uso d'estensione:

- UC9: Creazione appuntamento

Caso d'uso	Attori Primari	Attori Secondari	Incl. / Ext.	Requisiti corrispondenti
UC1: Creazione lista dei numeri telefonici	Amministratore	-	-	RF01
UC2: Gestione lista dei numeri telefonici	Amministratore	-	-	RF02, RF03
UC3: Creazione gruppi centralinisti	Amministratore	-	-	RF04, RF05
UC4: Gestione gruppi centralinisti	Amministratore	-	-	RF06, RF07
UC5: Assegnazione lista gruppi	Amministratore	-	-	RF08
UC6: Chiamata utenza	Centralinista	Utenza	Creazione Appuntamento	RF09, RF10, RF11, RF12
UC7: Interazione con lista appuntamenti	Agente di vendita	Utenza	-	RF13,
UC8: Visualizza note appuntamento fissato	Agente di vendita	Utenza	-	RF14
UC9: Creazione Appuntamento	Centralinista	-	-	RF12

## 2.5.2 Diagrammi dei casi d'uso



## 2.5.3 Scenari

Caso d'uso:	Creazione lista numeri telefonici
Attore primario	Amministratore
Attore secondario	-
Descrizione	L'amministratore crea una nuova lista di numeri telefonici, inserendo almeno un numero
Pre-Condizioni	Avere un numero telefonico valido di 10 cifre
Sequenza di eventi principale	<ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando l'amministratore richiede di creare una nuova lista</li> <li>2. Il sistema genera un nuovo ID da assegnare alla lista</li> <li>3. Il sistema richiede i numeri da inserire</li> <li>4. L'amministratore inserisce i numeri <ol style="list-style-type: none"> <li>4.1. Per ogni numero inserito <ol style="list-style-type: none"> <li>4.1.1. Il sistema verifica che sia valido</li> <li>4.1.2. Se non è valido il sistema restituisce un ERRORE all'amministratore</li> </ol> </li> </ol> </li> </ol>
Post-Condizioni	Il sistema contiene una nuova lista utili di numeri da poter chiamare
Casi d'uso correlati	Nessuno
Sequenza di eventi alternativi	-

Caso d'uso:	Modifica lista numeri telefonici
Attore primario	Amministratore
Attore secondario	-
Descrizione	L'amministratore modifica una lista di numeri telefonici, aggiungendo o togliendo un numero telefonico o rimuovendo la lista
Pre-Condizioni	Avere l'ID di una lista
Sequenza di eventi principale	<ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando l'amministratore richiede di modificare una lista</li> <li>2. L'amministratore inserisce l'ID della lista che vuole modificare <ol style="list-style-type: none"> <li>2.1. Se l'ID non è valido il sistema restituisce un ERRORE all'amministratore</li> </ol> </li> </ol>

	<ol style="list-style-type: none"> <li>3. Il sistema richiede all'amministratore l'operazione da effettuare</li> <li>4. Se la richiesta è l'aggiunta di un nuovo numero <ol style="list-style-type: none"> <li>4.1. Il sistema richiede il numero da inserire</li> <li>4.2. L'amministratore inserisce il numero</li> <li>4.3. Se il numero non è valido <ol style="list-style-type: none"> <li>4.3.1. Il sistema restituisce un ERRORE all'amministratore</li> </ol> </li> </ol> </li> </ol>
<b>Post-Condizioni</b>	Una lista di numeri telefonici è stata modificata
<b>Casi d'uso correlati</b>	<i>Nessuno</i>
<b>Sequenza di eventi alternativi</b>	<p>Al punto 4 se la richiesta è la rimozione di un numero</p> <ol style="list-style-type: none"> <li>1. Il sistema richiede il numero da eliminare</li> <li>2. L'amministratore inserisce il numero</li> <li>3. Se il numero non è presente nella lista <ol style="list-style-type: none"> <li>3.1 Il sistema restituisce un ERRORE all'amministratore</li> </ol> </li> <li>4. Il sistema rimuove il numero dalla lista</li> </ol> <p>Al punto 4 se la richiesta è di rimozione della lista</p> <ol style="list-style-type: none"> <li>1 Il sistema elimina la lista</li> </ol>

<b>Caso d'uso: Creazione gruppi centralinisti</b>	
<b>Attore primario</b>	Amministratore
<b>Attore secondario</b>	-
<b>Descrizione</b>	L'amministratore crea un gruppo di centralinisti
<b>Pre-Condizioni</b>	-
<b>Sequenza di eventi principale</b>	<ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando l'amministratore richiede al sistema un nuovo gruppo</li> <li>2. Il sistema genera un nuovo ID da assegnare al gruppo</li> <li>3. Il sistema richiede i centralinisti da inserire nel gruppo</li> <li>4. L'amministratore inserisce l'ID del centralinista <ol style="list-style-type: none"> <li>4.1. Per ogni centralinista <ol style="list-style-type: none"> <li>4.1.1. il sistema controlla che l'ID sia valido</li> <li>4.1.2. Se non è valido il sistema invia un messaggio di ERRORE</li> </ol> </li> </ol> </li> </ol>
<b>Post-Condizioni</b>	Il sistema contiene un nuovo gruppo con dei centralinisti
<b>Casi d'uso correlati</b>	<i>Nessuno</i>
<b>Sequenza di eventi alternativi</b>	-

<b>Caso d'uso: Modifica gruppi di centralinisti</b>	
<b>Attore primario</b>	Amministratore
<b>Attore secondario</b>	-
<b>Descrizione</b>	L'amministratore modifica un gruppo di centralinisti aggiungendo o togliendo un centralinista o rimuovendo il gruppo
<b>Pre-Condizioni</b>	Avere l'ID del gruppo

<b>Sequenza di eventi principale</b>	<ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando l'amministratore richiede di modificare un gruppo di centralinisti</li> <li>2. L'amministratore inserisce l'ID del gruppo che vuole modificare <ol style="list-style-type: none"> <li>2.1. Se l'ID non è valido il sistema restituisce un ERRORE all'amministratore</li> </ol> </li> <li>3. Il sistema richiede all'amministratore l'operazione da effettuare</li> <li>4. Se la richiesta è l'aggiunta di un nuovo centralinista <ol style="list-style-type: none"> <li>4.4. L'amministratore inserisce l'ID desiderato</li> <li>4.5. Se l'ID non è valido <ol style="list-style-type: none"> <li>4.5.1. Il sistema restituisce un ERRORE all'amministratore</li> </ol> </li> </ol> </li> </ol>
<b>Post-Condizioni</b>	Un gruppo di centralinisti è stato modificato
<b>Casi d'uso correlati</b>	<i>Nessuno</i>
<b>Sequenza di eventi alternativi</b>	<p>Al punto 4 se la richiesta è la rimozione di un centralinista</p> <ol style="list-style-type: none"> <li>1. L'amministratore inserisce l'ID desiderato</li> <li>2 Se l'ID non è presente nella lista <ol style="list-style-type: none"> <li>2.1 Il sistema restituisce un ERRORE all'amministratore</li> </ol> </li> <li>3 Il sistema rimuove il centralinista dal gruppo</li> </ol> <p>Al punto 4 se la richiesta è di eliminazione del gruppo</p> <ol style="list-style-type: none"> <li>1. 1 Il sistema elimina il gruppo</li> </ol>

<b>Caso d'uso:</b>	<b>Assegnazione Lista</b>
<b>Attore primario</b>	Amministratore
<b>Attore secondario</b>	-
<b>Descrizione</b>	L'amministratore ha il compito di assegnare ciascuna lista di numeri al più a un gruppo di centralinisti
<b>Pre-Condizioni</b>	Sia la lista che il gruppo esistano
<b>Sequenza di eventi principale</b>	<ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando l'amministratore richiede di assegnare una lista ad un gruppo di centralinisti</li> <li>2. L'amministratore inserisce l'ID della lista che vuole assegnare al gruppo <ol style="list-style-type: none"> <li>2.1. Se l'ID non è valido il sistema restituisce un ERRORE all'amministratore</li> </ol> </li> <li>3. Il sistema richiede all'amministratore l'ID del gruppo a cui vuole assegnare la lista <ol style="list-style-type: none"> <li>3.1. Se l'id non è valido il sistema restituisce un errore all'amministratore</li> </ol> </li> <li>4. Il sistema assegna la lista al gruppo</li> </ol>
<b>Post-Condizioni</b>	La lista è stata assegnata al gruppo
<b>Casi d'uso correlati</b>	<i>Nessuno</i>
<b>Sequenza di eventi Alternativi</b>	-

<b>Caso d'uso:</b>	<b>Chiamata Utenza</b>
<b>Attore primario</b>	Centralinista
<b>Attore secondario</b>	Utente
<b>Descrizione</b>	Il centralinista appartenente ad un determinato gruppo richiede un nuovo numero da chiamare ed effettua la chiamata

<b>Pre-Condizioni</b>	Il centralinista deve appartenere ad un gruppo
<b>Sequenza di eventi principale</b>	<ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando il centralinista richiede il prossimo numero da chiamare</li> <li>2. Il sistema assegna il numero da chiamare al centralinista</li> <li>3. Il centralinista avvia la chiamata</li> <li>4. Il sistema memorizza data, ora e ID della chiamata</li> <li>5. Il centralinista inserisce l'esito della chiamata</li> </ol>
<b>Post-Condizioni</b>	Appuntamento fissato
<b>Casi d'uso correlati</b>	Creazione Appuntamento
<b>Sequenza di eventi alternativi</b>	Se l'esito della chiamata è "appuntamento fissato" <ol style="list-style-type: none"> <li>6. Il centralinista inserisce le note relative all'appuntamento</li> </ol> Caso d'estensione: Creazione Appuntamento

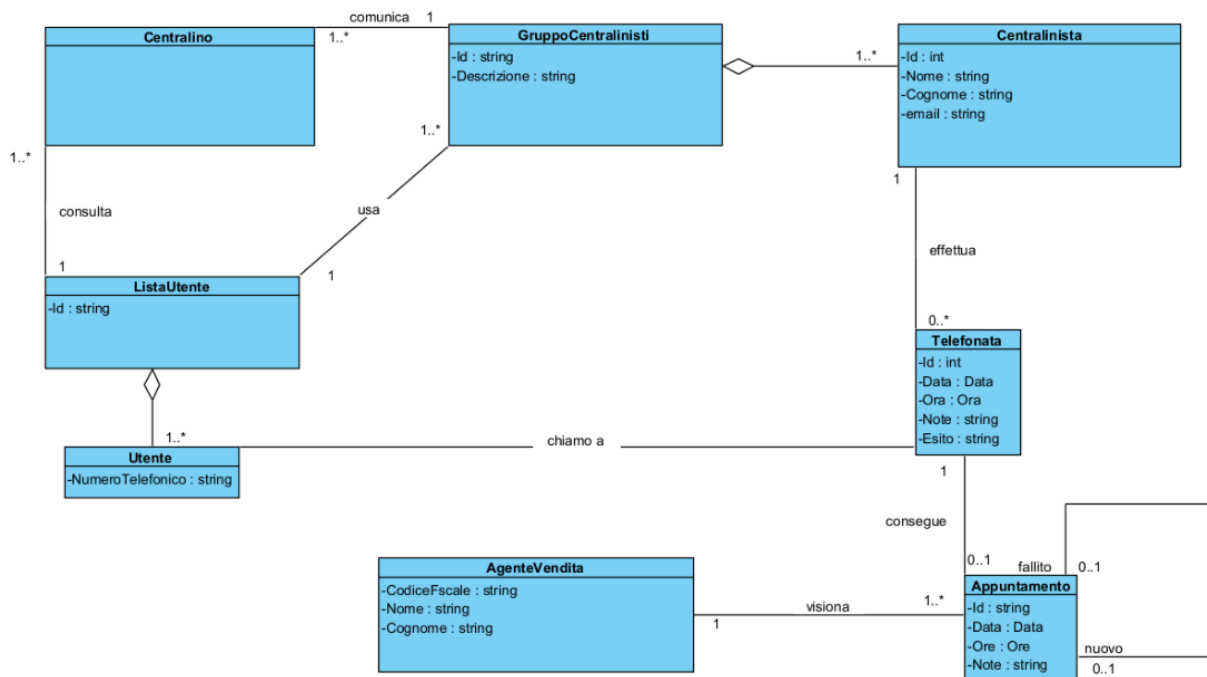
<b>Caso d'uso:</b>	<b>Interazione con lista appuntamenti</b>
<b>Attore primario</b>	Agente vendita
<b>Attore secondario</b>	Utente
<b>Descrizione</b>	L'agente vendita visualizza la lista di appuntamenti e può eventualmente modificare le note di un appuntamento
<b>Pre-Condizioni</b>	-
<b>Sequenza di eventi principale</b>	<ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando l'agente vendita richiede la lista degli appuntamenti fissati</li> <li>2. Se l'agente richiede di visualizzare un appuntamento               <ol style="list-style-type: none"> <li>2.1. Il sistema mostra le note relative all'appuntamento</li> <li>2.2. Se l'agente di vendita richiede di modificare le note                   <ol style="list-style-type: none"> <li>2.2.1. Il sistema permette all'agente di modificare le note</li> </ol> </li> </ol> </li> </ol>
<b>Post-Condizioni</b>	-
<b>Casi d'uso correlati</b>	<i>Nessuno</i>
<b>Sequenza di eventi alternativi</b>	-

<b>Caso d'uso:</b>	<b>Creazione appuntamento</b>
<b>Attore primario</b>	Centralinista
<b>Attore secondario</b>	Utente
<b>Descrizione</b>	Il centralinista chiama l'utente, il centralinista fissa un appuntamento tra l'agente vendita e l'utente.
<b>Pre-Condizioni</b>	Esito telefonata: "appuntamento fissato"
<b>Sequenza di eventi principale</b>	<ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando l'utente si concorda col centralinista per un appuntamento</li> <li>2. Il centralinista crea un appuntamento</li> <li>3. Il sistema crea un id</li> <li>4. Il centralinista inserisce: data, ora e le note</li> <li>5. Il centralinista sceglie l'agente vendita con cui l'utente interagirà</li> </ol>
<b>Post-Condizioni</b>	Il centralinista ha creato un appuntamento con l'utente e quest'ultimo interagirà con l'agente vendita
<b>Casi d'uso correlati</b>	<i>Nessuno</i>
<b>Sequenza di eventi alternativi</b>	<ol style="list-style-type: none"> <li>6. Se è presente un appuntamento fallito per lo stesso utente, l'appuntamento attuale referencia il precedente tramite l'id</li> </ol>

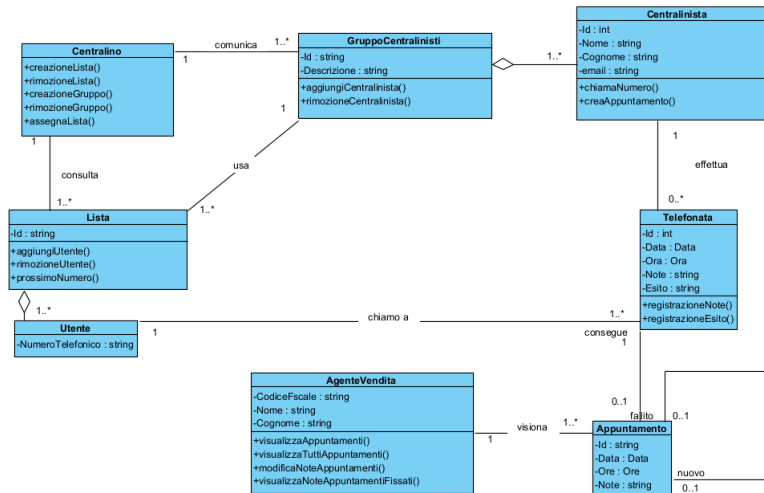
Caso d'uso:	Visualizza note appuntamento fissato
Attore primario	Agente vendita
Attore secondario	-
Descrizione	L'agente di vendita interagisce con il sistema per poter visualizzare le note relative ad un appuntamento
Pre-Condizioni	ID valido per un appuntamento
Sequenza di eventi principale	<ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando l'agente inserisce l'ID di un appuntamento</li> <li>2. Se l'ID non è valido <ol style="list-style-type: none"> <li>2.1. Il sistema restituisce un messaggio di Errore</li> </ol> </li> <li>3. Il sistema mostra le note relative all'appuntamento indicato</li> </ol>
Post-Condizioni	-
Casi d'uso correlati	Nessuno
Sequenza di eventi alternativi	-

## 2.6 Diagrammi delle classi

Diagrammi delle classi di analisi:

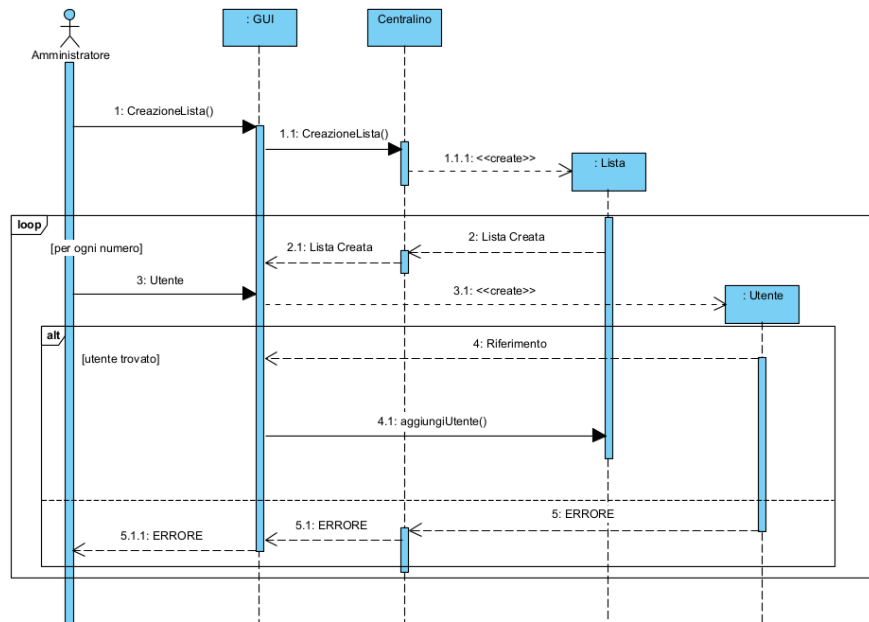


## Diagramma delle classi raffinato:

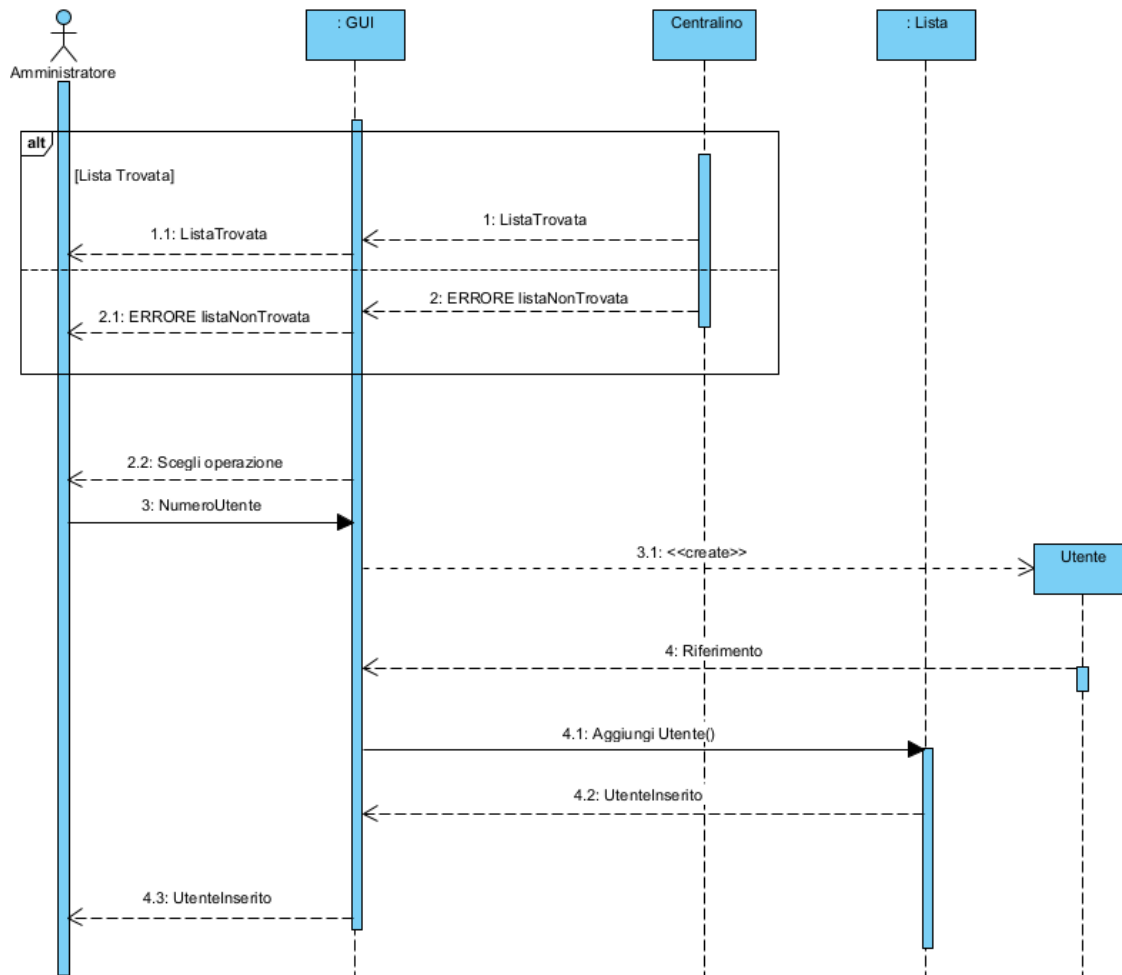


## 2.7 DIAGRAMMI DI SEQUENZA

### 2.7.1 Creazione lista numeri telefonici:



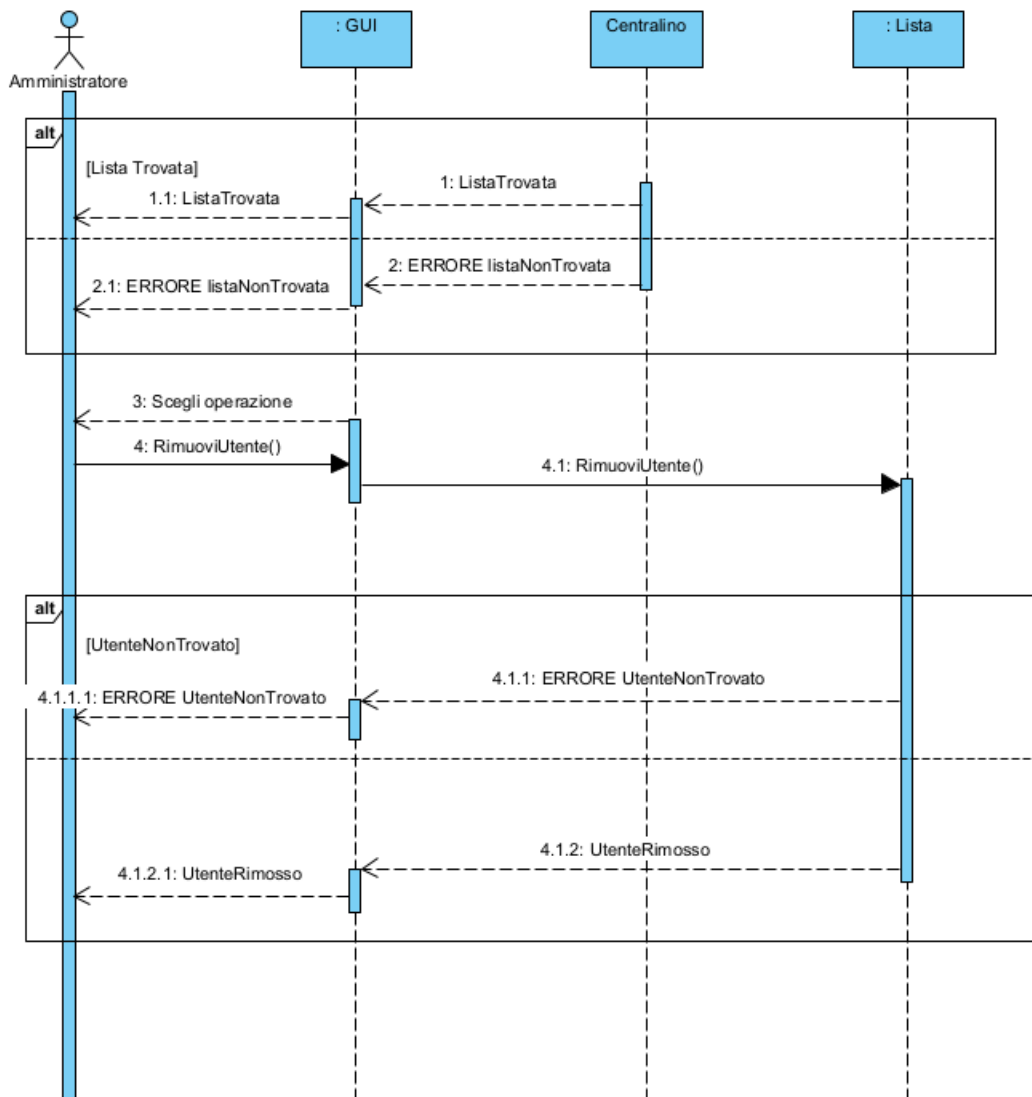
## 2.7.2 Modifica lista numeri telefonici (scenario normale)



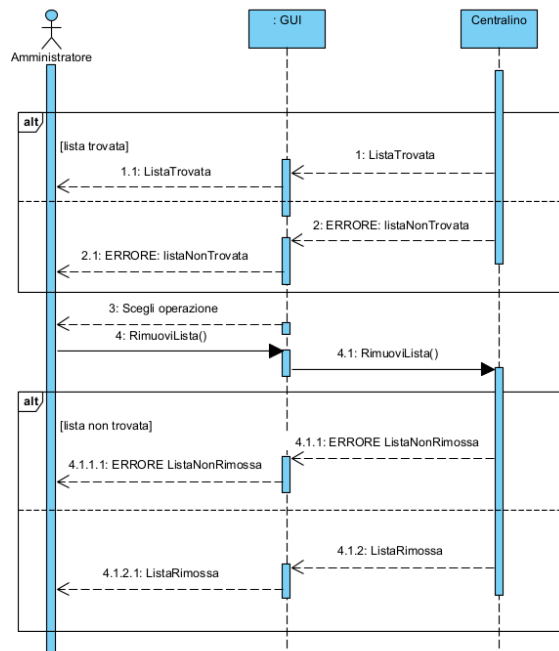


## Modifica lista numeri telefonici (scenario alternativo):

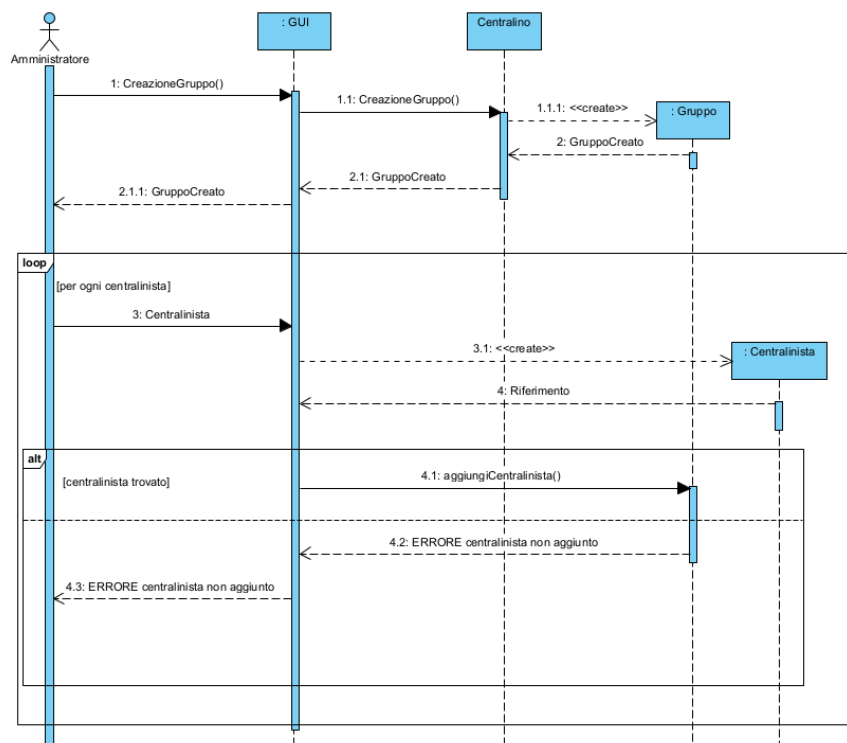
- Rimozione numero



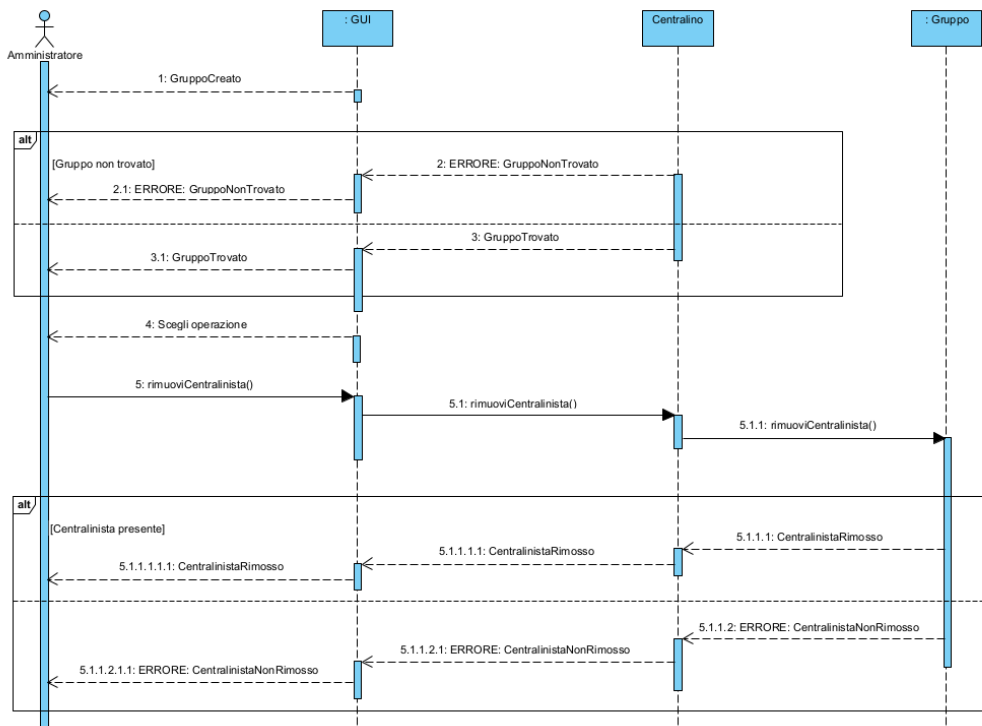
- Rimozione lista:



### 2.7.3 Creazione gruppo centralinista (Scenario normale)

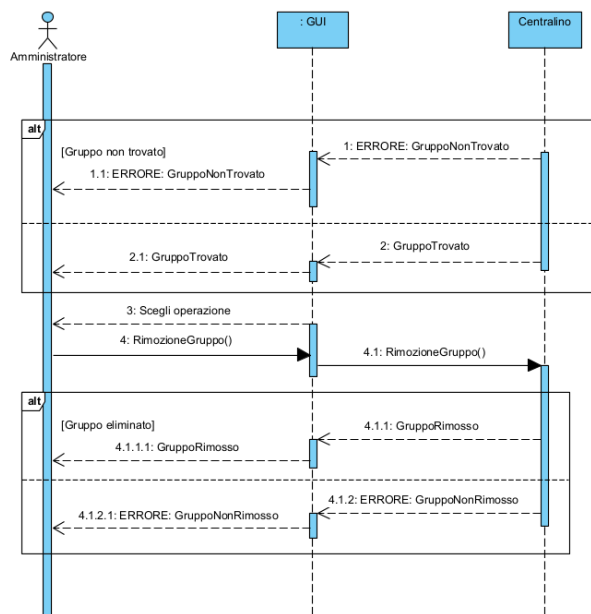


## 2.7.4 Modifica Gruppi Centralinisti (Scenario normale)

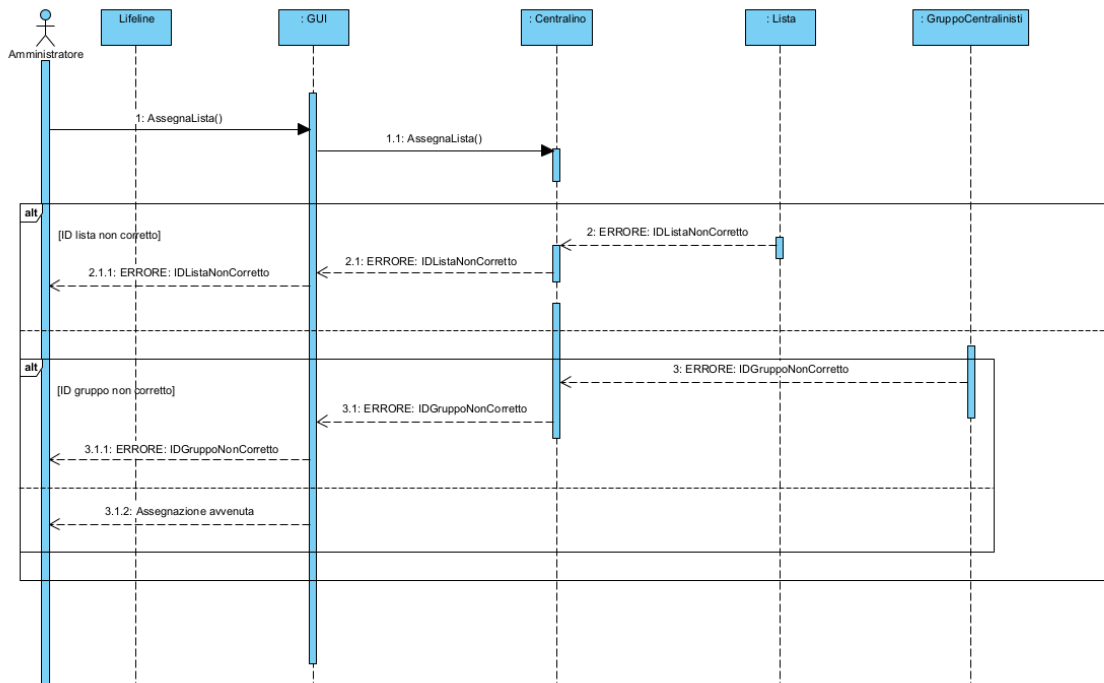


### Creazione gruppi centralinisti (Scenario alternativo):

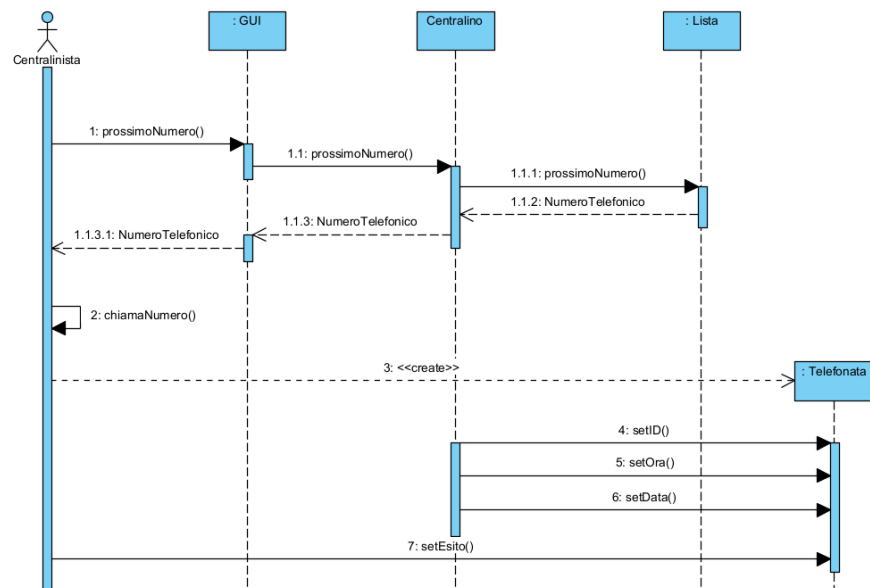
- Rimozione



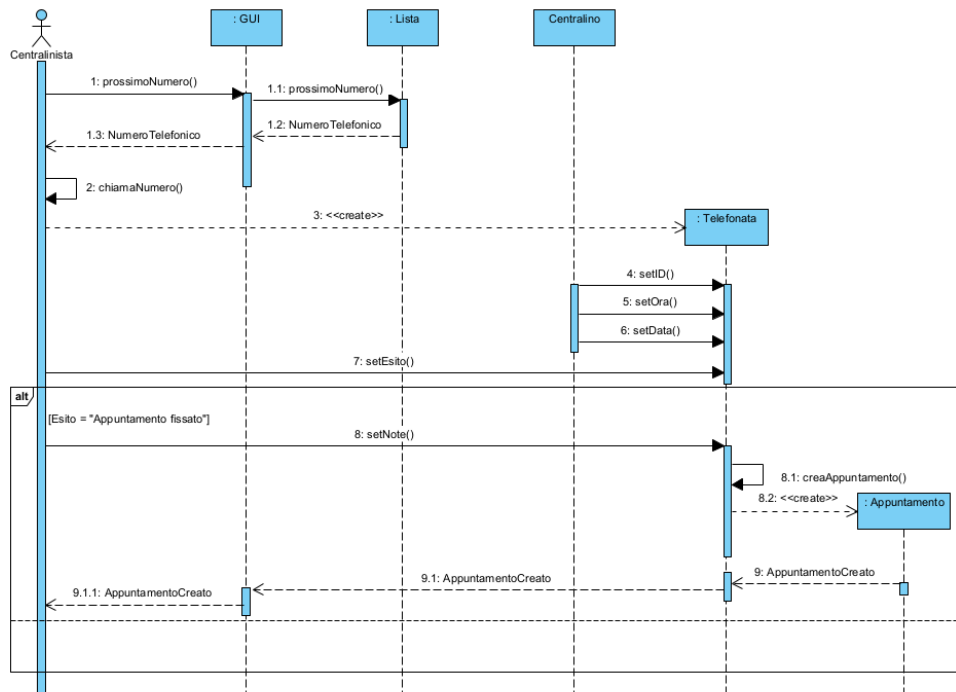
## 2.7.5 Assegnazione Lista



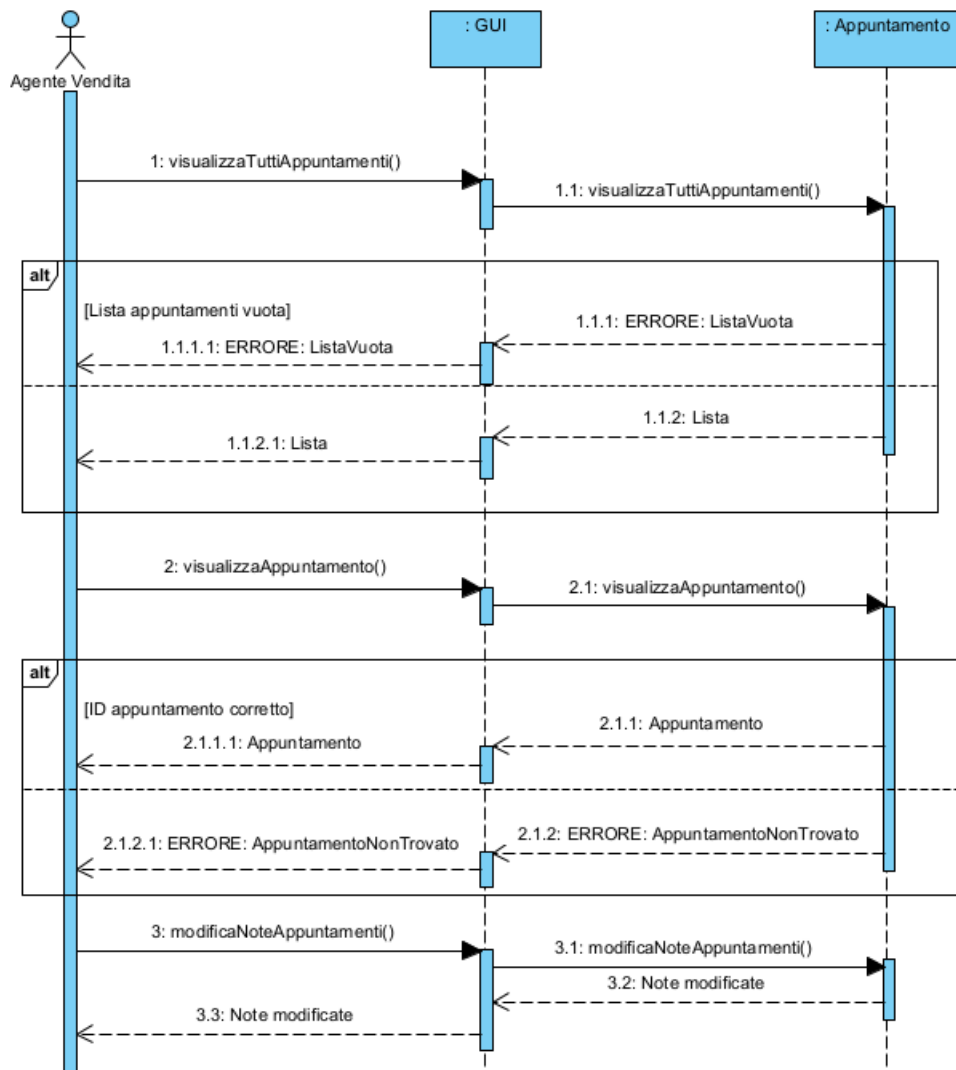
## 2.7.6 Chiamata Utente



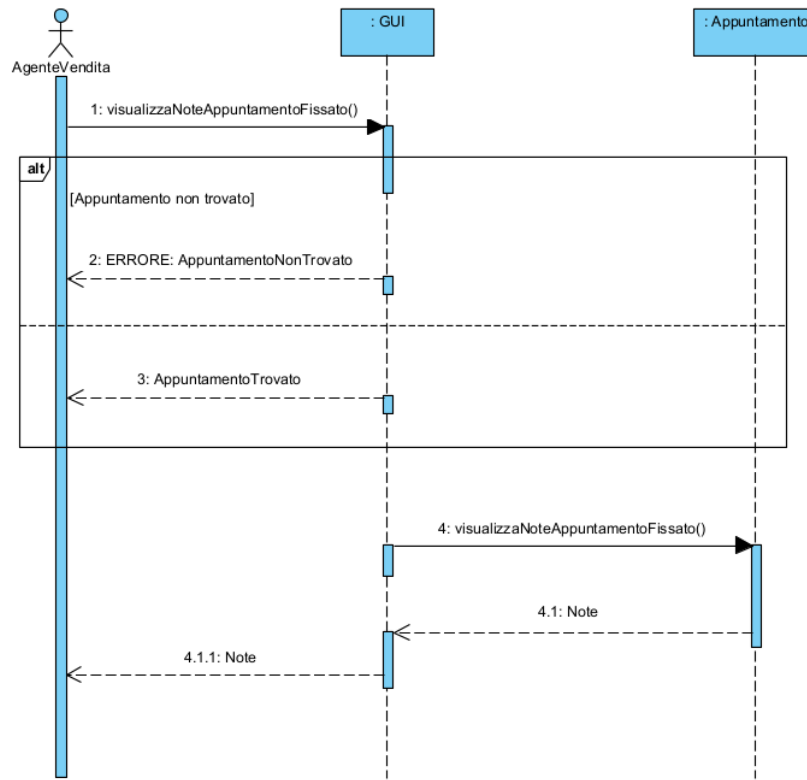
## 2.7.7 Creazione Appuntamento



## 2.7.8 Interazione con lista appuntamenti



### 2.7.9 Visualizza note appuntamento fissato



## 2.8 VERIFICA DELLA COMPLETEZZA DEI REQUISITI

Verifica che tutti i requisiti informali siano stati rappresentati nel modello UML e/o negli scenari.

Legenda: UDC= use case diagram; CD= class diagram; SD= sequence diagram.

- RF01 è modellato nell'UDC con il caso d'uso UC1.
- RF02 è modellato nell'UDC con il caso d'uso UC2.
- RF03 è modellato nell'UDC con il caso d'uso UC2.
- RF04 è modellato nell'UDC con il caso d'uso UC3.
- RF05 è modellato nell'UDC con il caso d'uso UC4.
- RF06 è modellato nell'UDC con il caso d'uso UC4.
- RF07 è modellato nell'UDC con i casi d'uso UC4.
- RF08 è modellato nell' UDC con i casi d'uso UC5.
- RF09 è modellato nell'UDC con il caso d'uso UC6.
- RF10 è modellato nell'UDC con il caso d'uso UC6.
- RF11 è modellato nell'UDC con il caso d'uso UC6.
- RF12 è modellato nell'UDC con il caso d'uso UC9.
- RF13 è modellato nell'UDC con il caso d'uso UC7.
- RF14 è modellato nell'UDC con il caso d'uso UC8.

### 3. STIMA DEI COSTI

Stima dei costi secondo il metodo dei Punti Funzione.

- Tabella di riferimento per le complessità di dati e transazioni

	SEMPLICE	MEDIO	COMPLESSO
NILF	7	10	15
NEIF	5	7	10
NEI	3	4	6
NEO	4	5	7
NEQ	4	4	6

- Tabella elenco dei fattori correttivi (il cui valore è compreso tra 0 e 5)

FATTORI CORRETTIVI
COMUNICAZIONE DATI
DISTRIBUZIONE ELABORAZIONE
PRESTAZIONI
UTILIZZO INTENSIVO CONFIGURAZIONE
FREQUENZA DELLE TRANSAZIONI
INSERIMENTO DATI INTERATTIVO
EFFICIENZA PER L'UTENTE FINALE
AGGIORNAMENTO INTERATTIVO
COMPLESSITA' ELABORATIVA
RIUSABILITA'
FACILITA' INSTALLAZIONE
FACILITA' GESTIONE OPERATIVA
MOLTEPLICITA' DI SITI
FACILITA' DI MODIFICA

#### 3.1 CREAZIONE LISTA NUMERI TELEFONICI

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	1	7	-	-	1x7=10
NEIF	-	-	-	-	-
NEI	1	3	-	-	1x3=3
NEO	-	-	-	-	-
NEQ	-	-	-	-	-

NILF = il sistema dovrà generare una lista di numeri telefonici



NEI = l'amministratore dovrà aggiungere dati per creare la lista dei numeri telefonici

UFP = 7+3= 10

LLOC/FP= 10x53= 530

FATTORI CORRETTIVI	
COMUNICAZIONE DATI	2
DISTRIBUZIONE ELABORAZIONE	1
PRESTAZIONI	1
UTILIZZO INTENSIVO CONFIGURAZIONE	0
FREQUENZA DELLE TRANSAZIONI	3
INSERIMENTO DATI INTERATTIVO	1
EFFICIENZA PER L'UTENTE FINALE	2
AGGIORNAMENTO INTERATTIVO	2
COMPLESSITA' ELABORATIVA	1
RIUSABILITA'	3
FACILITA' INSTALLAZIONE	2
FACILITA' GESTIONE OPERATIVA	2
MOLTEPLICITA' DI SITI	0
FACILITA' DI MODIFICA	1

FP= UFP \*  $(0,65 + 0,01 * \sum_{i=1}^{14} F_i) = 14 * (0,65 + 0,01 * 21) = 8,6$

LLOC/FP = 8,6\*53 = 456

Linguaggio	LLOC/FP	Linee codice stimate
Assembler	119	3218
C	97	2623
Cobol	61	1650
Java	53	1433
VB.NET	52	1407
C++	50	1352
HTML	34	920

### 3.2 MODIFICA LISTA NUMERI TELEFONICI

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	1	7	-	-	1x7=7
NEIF	-	-	-	-	-
NEI	2	3	-	-	2x3=6
NEO	-	-	-	-	-
NEQ	-	-	-	-	-

NILF = il sistema aggiungerà un nuovo elemento alla lista

NEI = l'amministratore inserisce l'id della lista che vuole modificare e inserisce il numero

$$UFP = 7+6=13$$

$$LLOC/FP = 21 \times 53 = 689$$

FATTORI CORRETTIVI	
COMUNICAZIONE DATI	2
DISTRIBUZIONE ELABORAZIONE	0
PRESTAZIONI	3
UTILIZZO INTENSIVO CONFIGURAZIONE	0
FREQUENZA DELLE TRANSAZIONI	2
INSERIMENTO DATI INTERATTIVO	3
EFFICIENZA PER L'UTENTE FINALE	3
AGGIORNAMENTO INTERATTIVO	2
COMPLESSITA' ELABORATIVA	2
RIUSABILITA'	3
FACILITA' INSTALLAZIONE	2
FACILITA' GESTIONE OPERATIVA	2
MOLTEPLICITA' DI SITI	0
FACILITA' DI MODIFICA	1

$$FP = UFP * (0,65 + 0,01 * \sum_{i=1}^{14} F_i) = 13 * (0,65 + 0,01 * 25) = 11,7$$

$$LLOC/FP = 18 \times 53 = 620$$

Linguaggio	LLOC/FP	Linee codice stimate
Assembler	119	3218
C	97	2623
Cobol	61	1650
Java	53	1433
VB.NET	52	1407
C++	50	1352
HTML	34	920

### 3.3 CREAZIONE GRUPPI CENTRALINISTI

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
<b>NILF</b>	1	7	-	-	1x7=7
<b>NEIF</b>	-	-	-	-	-
<b>NEI</b>	1	3	-	-	1x3=3
<b>NEO</b>	-	-	-	-	-
<b>NEQ</b>	-	-	-	-	-

NILF= il sistema genera il nuovo gruppo

NEI= l'amministratore inserisce l'id del centralinista da inserire nel gruppo

UFP =7+3=10

LLOC/FP= 10x53= 530

<b>FATTORI CORRETTIVI</b>	
COMUNICAZIONE DATI	2
DISTRIBUZIONE ELABORAZIONE	1
PRESTAZIONI	1
UTILIZZO INTENSIVO CONFIGURAZIONE	0
FREQUENZA DELLE TRANSAZIONI	2
INSERIMENTO DATI INTERATTIVO	3
EFFICIENZA PER L'UTENTE FINALE	3
AGGIORNAMENTO INTERATTIVO	3
COMPLESSITA' ELABORATIVA	1
RIUSABILITA'	3
FACILITA' INSTALLAZIONE	2
FACILITA' GESTIONE OPERATIVA	2
MOLTEPLICITA' DI SITI	0
FACILITA' DI MODIFICA	1

$FP = UFP * (0,65 + 0,01 * \sum_{i=1}^{14} F_i) = 10 * (0,65 + 0,01 * 24) = 8,9$

LLOC/FP= 12,88x53=472

Linguaggio	LLOC/FP	Linee codice stimate
<b>Assembler</b>	119	3218
<b>C</b>	97	2623
<b>Cobol</b>	61	1650
<b>Java</b>	53	1433

<b>VB.NET</b>	52	1407
<b>C++</b>	50	1352
<b>HTML</b>	34	920

### 3.4 MODIFICA GRUPPI CENTRALINISTI

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
<b>NILF</b>		7	-	-	1x7=7
<b>NEIF</b>	-	-	-	-	-
<b>NEI</b>	2	3	-	-	2x3=6
<b>NEO</b>	-	-	-	-	-
<b>NEQ</b>	-	-	-	-	-

NILF = il sistema modifica il gruppo

NEI = l'amministratore inserisce l'id del gruppo da modificare, l'amministratore inserisce l'id del centralinista da aggiungere

$$UFP = 7+6 = 13$$

$$LLOC/FP = 13 \times 53 = 689$$

<b>FATTORI CORRETTIVI</b>	
COMUNICAZIONE DATI	3
DISTRIBUZIONE ELABORAZIONE	0
PRESTAZIONI	3
UTILIZZO INTENSIVO CONFIGURAZIONE	0
FREQUENZA DELLE TRANSAZIONI	3
INSERIMENTO DATI INTERATTIVO	3
EFFICIENZA PER L'UTENTE FINALE	3
AGGIORNAMENTO INTERATTIVO	2
COMPLESSITA' ELABORATIVA	3
RIUSABILITA'	2
FACILITA' INSTALLAZIONE	1
FACILITA' GESTIONE OPERATIVA	2
MOLTEPLICITA' DI SITI	0
FACILITA' DI MODIFICA	1

$$FP = UFP * (0,65 + 0,01 * \sum_{i=1}^{14} F_i) = 13 * (0,65 + 0,01 * 26) = 11,83$$

$$LLOC/FP = 11,83 \times 53 = 627$$

Linguaggio	LLOC/FP	Linee codice stimate
Assembler	119	3218
C	97	2623
Cobol	61	1650
Java	53	1433
VB.NET	52	1407
C++	50	1352
HTML	34	920

### 3.5 ASSEGNA LISTA

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	1	7	-	-	1x7 = 7
NEIF	-	-	-	-	-
NEI	2	3	-	-	2x3 = 6
NEO	1	4	-	-	1x4 = 4
NEQ	-	-	-	-	-

NILF = il sistema assegna la lista al gruppo

NEI = L'amministratore inserisce l'ID della lista e l'id del gruppo

NEO = il sistema ritorna errore se l'amministratore ha sbagliato ad inserire l'id della lista o del gruppo

UFP = 7+6+4 = 17

LLOC/FP = 17x53 =901

FATTORI CORRETTIVI	
COMUNICAZIONE DATI	3
DISTRIBUZIONE ELABORAZIONE	0
PRESTAZIONI	3
UTILIZZO INTENSIVO CONFIGURAZIONE	0
FREQUENZA DELLE TRANSAZIONI	1
INSERIMENTO DATI INTERATTIVO	2
EFFICIENZA PER L'UTENTE FINALE	3
AGGIORNAMENTO INTERATTIVO	2
COMPLESSITA' ELABORATIVA	2
RIUSABILITA'	1
FACILITA' INSTALLAZIONE	1
FACILITA' GESTIONE OPERATIVA	2
MOLTEPLICITA' DI SITI	0
FACILITA' DI MODIFICA	1

$$FP = UFP * (0,65 + 0,01 * \sum_{i=1}^{14} F_i) = 22 * (0,65 + 0,01 * 21) = 18,92$$

$$LLOC/FP = 18,92 \times 53 = 1002$$

Linguaggio	LLOC/FP	Linee codice stimate
Assembler	119	3218
C	97	2623
Cobol	61	1650
Java	53	1433
VB.NET	52	1407
C++	50	1352
HTML	34	920

### 3.6 CHIAMATA UTENZA

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	1	7	-	-	1x7 = 7
NEIF	-	-	-	-	-
NEI	1	3	-	-	1x3 = 3
NEO	1	4	-	-	1x4 = 4
NEQ	2	4	-	-	2x4 = 8

NILF = il sistema assegna il numero da chiamare al centralinista

NEI = il centralinista inserisce l'ID della lista

NEO = numero telefonico dell'utente da chiamare

NEQ = il centralinista interroga il sistema per cercare la lista dei numeri, e poi interroga la lista per cercare il numero telefonico desiderato

$$UFP = 7+3+4+8 = 22$$

$$LLOC/FP = 22 \times 53 = 1166$$

FATTORI CORRETTIVI	
COMUNICAZIONE DATI	3
DISTRIBUZIONE ELABORAZIONE	0
PRESTAZIONI	3
UTILIZZO INTENSIVO CONFIGURAZIONE	1
FREQUENZA DELLE TRANSAZIONI	3
INSERIMENTO DATI INTERATTIVO	3
EFFICIENZA PER L'UTENTE FINALE	3
AGGIORNAMENTO INTERATTIVO	3

COMPLESSITA' ELABORATIVA	2
RIUSABILITA'	1
FACILITA' INSTALLAZIONE	1
FACILITA' GESTIONE OPERATIVA	2
MOLTEPLICITA' DI SITI	0
FACILITA' DI MODIFICA	1

$FP = UFP * (0,65 + 0,01 * \sum_{i=1}^{14} F_i) = 22 * (0,65 + 0,01 * 26) = 20,02$   
 $LLOC/FP = 20,02 * 53 = 1061$

Linguaggio	LLOC/FP	Linee codice stimate
Assembler	119	3218
C	97	2623
Cobol	61	1650
Java	53	1433
VB.NET	52	1407
C++	50	1352
HTML	34	920

### 3.7 INTERAZIONE CON LISTA APPUNTAMENTI

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	2	7	-	-	2*7=14
NEIF	-	-	-	-	-
NEI	2	3	-	-	2*3=6
NEO	-	-	-	-	-
NEQ	2	4	-	-	2*4=8

NILF=Il sistema mostra la lista di appuntamenti, modifica le note di un appuntamento

NEI= ID appuntamento, nuove note

NEQ=il sistema mostra trova tutti gli appuntamenti fissati, inserisce le nuove note

UFP = 14+6+8=28

LLOC/FP= 28x53=1696

FATTORI CORRETTIVI	
COMUNICAZIONE DATI	2
DISTRIBUZIONE ELABORAZIONE	0
PRESTAZIONI	2
UTILIZZO INTENSIVO CONFIGURAZIONE	0

FREQUENZA DELLE TRANSAZIONI	1
INSERIMENTO DATI INTERATTIVO	2
EFFICIENZA PER L'UTENTE FINALE	3
AGGIORNAMENTO INTERATTIVO	3
COMPLESSITA' ELABORATIVA	1
RIUSABILITA'	1
FACILITA' INSTALLAZIONE	1
FACILITA' GESTIONE OPERATIVA	2
MOLTEPLICITA' DI SITI	0
FACILITA' DI MODIFICA	1

$FP = UFP * (0,65 + 0,01 * \sum_{i=1}^{14} F_i) = 28 * (0,65 + 0,01 * 18) = 23,24$   
 $LLOC/FP = 23,24 * 53 = 1231$

Linguaggio	LLOC/FP	Linee codice stimate
Assembler	119	3218
C	97	2623
Cobol	61	1650
Java	53	1433
VB.NET	52	1407
C++	50	1352
HTML	34	920

### 3.8 CREAZIONE APPUNTAMENTO

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	1	7		-	1x7 = 7
NEIF	-	-	-	-	-
NEI	2	3	-	-	2x3 = 6
NEO	1	4	-	-	1x4 = 4
NEQ	-	-	-	-	-

NILF = il sistema dovrà permettere al centralinista di creare un nuovo appuntamento

NEI = il centralinista dovrà aggiungere i dati che permettono di fissare l'appuntamento e l'agente vendita che se ne occuperà

NEO = il sistema comunicherà con un messaggio sull'esito dell'operazione

$UFP = 7 + 6 + 4 = 17$

$LLOC/FP = 17 * 53 = 901$



FATTORI CORRETTIVI	
COMUNICAZIONE DATI	2
DISTRIBUZIONE ELABORAZIONE	0
PRESTAZIONI	3
UTILIZZO INTENSIVO CONFIGURAZIONE	0
FREQUENZA DELLE TRANSAZIONI	2
INSERIMENTO DATI INTERATTIVO	3
EFFICIENZA PER L'UTENTE FINALE	2
AGGIORNAMENTO INTERATTIVO	1
COMPLESSITA' ELABORATIVA	2
RIUSABILITA'	2
FACILITA' INSTALLAZIONE	2
FACILITA' GESTIONE OPERATIVA	2
MOLTEPLICITA' DI SITI	0
FACILITA' DI MODIFICA	1

$FP = UFP * (0,65 + 0,01 * \sum_{i=1}^{14} F_i) = 17 * (0,65 + 0,01 * 22) = 14,79$   
 $LLOC/FP = 14,79 * 53 = 783$

Linguaggio	LLOC/FP	Linee codice stimate
Assembler	119	3218
C	97	2623
Cobol	61	1650
Java	53	1433
VB.NET	52	1407
C++	50	1352
HTML	34	920

### 3.9 VISUALIZZA NOTE APPUNTAMENTO FISSATO

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	1	7	-	-	1x7 = 7
NEIF	-	-	-	-	-
NEI	-	-	-	-	-
NEO	1	4	-	-	1x4 = 4
NEQ	1	4	-	-	1x4 = 4

NILF = il sistema mostra le note di un determinato appuntamento

NEI = l'agente vendita inserirà l'ID dell'appuntamento

NEQ = il sistema cerca l'appuntamento fissato e mostra le note relative all'appuntamento

$$UFP = 7+4+4 = 15$$

$$LLOC/FP = 15 \times 53 = 954$$

FATTORI CORRETTIVI	
COMUNICAZIONE DATI	2
DISTRIBUZIONE ELABORAZIONE	0
PRESTAZIONI	2
UTILIZZO INTENSIVO CONFIGURAZIONE	0
FREQUENZA DELLE TRANSAZIONI	2
INSERIMENTO DATI INTERATTIVO	2
EFFICIENZA PER L'UTENTE FINALE	3
AGGIORNAMENTO INTERATTIVO	2
COMPLESSITA' ELABORATIVA	1
RIUSABILITA'	2
FACILITA' INSTALLAZIONE	3
FACILITA' GESTIONE OPERATIVA	2
MOLTEPLICITA' DI SITI	0
FACILITA' DI MODIFICA	1

$$FP = UFP * (0,65 + 0,01 * \sum_{i=1}^{14} F_i) = 15 * (0,65 + 0,01 * 22) = 13$$

$$LLOC/FP = 13 * 53 = 689$$

Linguaggio	LLOC/FP	Linee codice stimate
Assembler	119	3218
C	97	2623
Cobol	61	1650
Java	53	1433
VB.NET	52	1407
C++	50	1352
HTML	34	920

## 4. TEST FUNZIONALE

Il test funzionale l'analisi degli output generati dal sistema in risposta ad input, che prendono il nome di Test Cases, che vengono definiti sulla base dei soli requisiti del sistema. In questo caso, per ogni caso d'uso, applichiamo il Category Partition Testing.

Più nel dettaglio per ogni caso d'uso, andiamo a considerare gli input, e per ognuno di essi si ricava una classe di equivalenza valida (o più in caso servano), ovvero che corrisponda ad un insieme di valori validi, e un insieme di classi di equivalenza non valide, in numero pari al numero di condizioni di non validità.

### ELENCO FUNZIONI :

#### 4.1 Assegnazione Lista

ID_LISTA	ID_GRUPPO
<ul style="list-style-type: none"><li>• Lunghezza ID lista = 5</li><li>• Lunghezza ID lista &gt; 5 [ERROR]</li><li>• Lunghezza ID lista &lt; 5 [ERROR]</li><li>• ID non presente nel database [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• Lunghezza ID gruppo = 5</li><li>• Lunghezza ID gruppo &gt; 5 [ERROR]</li><li>• Lunghezza ID gruppo &lt; 5 [ERROR]</li><li>• ID non presente nel database [ERROR]</li></ul>

Il numero di test da effettuarsi senza particolari vincoli è:  $4 * 4 = 16$

Introduciamo i vincoli [ERROR] .

Il numero di test da eseguire per testare singolarmente i vincoli è 6 (3 per Lista , 3 per Gruppo).

Il numero di test risultante è:  $(1*1) + 6 = 7$ .

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti input validi	ID lista valido, ID gruppo valido	L'ID della lista e l'ID del gruppo centralinisti sono validi e presenti nel database.	{ ID lista : "1111" , ID gruppo : "2222" }	Assegnazione effettuata	Il gruppo di centralinisti ora possiede una lista
2	Lunghezza ID lista non valida, lunghezza ID gruppo valida	Lunghezza ID lista > 5 [ERROR], Lunghezza ID gruppo = 5	L'ID della lista non è valido in quanto le cifre sono maggiori di 5, l'ID del gruppo centralinisti è sia valido che presente nel database	{ID lista: "100113", ID gruppo:"11111" }	Assegnazione non effettuata perché la lunghezza dell'ID lista è maggiore di 5	L'assegnazione della lista non è avvenuta e l'amministratore riprova digitando un ID lista di 5 cifre
3	Lunghezza ID lista non valida, lunghezza ID gruppo valida	Lunghezza ID lista < 5 [ERROR], Lunghezza ID gruppo = 5	L'ID della lista non è valido in quanto le cifre sono minori di 5, l'ID del gruppo centralinisti è sia valido che presente nel database	{ID lista: "1001", ID gruppo:"11111" }	Assegnazione non effettuata perché la lunghezza dell'ID lista è minore di 5	L'assegnazione della lista non è avvenuta e l'amministratore riprova digitando un ID lista di 5 cifre
4	ID lista non presente nel database	ID lista non presente nel database [ERROR], ID gruppo valido	L'ID della lista non è presente nel database, l'id del gruppo centralinisti è sia valido che presente nel database	{ ID lista : "11122", ID gruppo:"11143" }	Assegnazione non effettuata perché l'ID della lista non è presente nel database	L'assegnazione della lista non è avvenuta e l'amministratore riprova controllando se nel database è presente l'id della lista inserito
5	ID gruppo non presente nel database	ID gruppo non presente nel database [ERROR], ID lista valido	L'ID del gruppo non è presente nel database, l'id della lista è sia valido che presente nel database	{ ID lista : "11143", ID gruppo:"11122" }	Assegnazione non effettuata perché l'ID del gruppo non è presente nel database	L'assegnazione della lista non è avvenuta e l'amministratore riprova controllando se nel database è presente l'id del gruppo inserito
6	Lunghezza ID lista valida, lunghezza ID gruppo non valida	Lunghezza ID lista = 5, Lunghezza ID gruppo > 5 [ERROR]	L'ID della lista è valido, l'ID del gruppo centralinisti non è valido in quanto le cifre sono maggiori di 5	{ID lista: "10011", ID gruppo:"11114" }	Assegnazione non effettuata perché la lunghezza dell'ID gruppo è maggiore di 5	L' assegnazione della lista non è avvenuta e l'amministratore riprova digitando un ID gruppo di 5 cifre
7	Lunghezza ID lista valida, lunghezza ID gruppo non valida	Lunghezza ID lista = 5, Lunghezza ID gruppo < 5 [ERROR]	L'ID della lista è valido, l'ID del gruppo centralinisti non è valido in quanto le cifre sono minori di 5	{ID lista: "10011", ID gruppo:"111" }	Assegnazione non effettuata perché la lunghezza dell'ID gruppo è minore di 5	L' assegnazione della lista non è avvenuta e l'amministratore riprova digitando un ID gruppo di 5 cifre

## 4.2 Registrazione Esito

ID	ESITO
<ul style="list-style-type: none"> <li>Lunghezza ID telefonata = 5</li> <li>Lunghezza ID telefonata &gt; 5 [ERROR]</li> <li>Lunghezza ID telefonata &lt; 5 [ERROR]</li> <li>ID non presente nel database [ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Esito valido ("occupato", "senza risposta", "da richiamare", "non interessato", "appuntamento fissato")</li> <li>Esito non valido [ERROR]</li> </ul>

Il numero di test da effettuarsi senza particolari vincoli è:  $4 * 2 = 8$

Introduciamo i vincoli [ERROR] .

Il numero di test da eseguire per testare singolarmente i vincoli è 4 (3 per Telefonata, 1 per Esito).

Il numero di test risultante è:  $(1*1) + 3 = 4$ .

### TEST SUITE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti input validi	ID telefonata corretto, esito telefonata valido	L'ID della telefonata è valido e la telefonata è stata registrata in modo corretto	{ID telefonata: "1111", esito: "occupato"}	Registrazione effettuata con successo	Alla telefonata è stato assegnato un esito
2	Lunghezza ID telefonata non valida, esito valido	Lunghezza ID telefonata > 5 [ERROR], esito valido	L'ID della telefonata non è valido, in quanto le cifre sono maggiori di 5, l'esito è valido	{ ID telefonata : "11118", esito: "occupato"}	Registrazione non effettuata perché la lunghezza dell'ID telefonata è maggiore di 5	La registrazione non è avvenuta e il centralinista riprova digitando un ID telefonata di 5 cifre
3	Lunghezza ID telefonata non valida, esito valido	Lunghezza ID telefonata < 5 [ERROR], esito valido	L'ID della telefonata non è valido, in quanto le cifre sono minori di 5, l'esito è valido	{ ID telefonata : "111", esito: "Occupato"}	Registrazione non effettuata perché la lunghezza dell'ID telefonata è minore di 5	La registrazione non è avvenuta e il centralinista riprova digitando un ID telefonata di 5 cifre
4	ID telefonata non presente nel database	ID telefonata non presente nel database [ERROR], esito valido	L'ID della telefonata non è presente nel database, l'esito è valido	{ ID telefonata : "11142", esito: "Occupato"}	Assegnazione non effettuata perché l'ID della telefonata non è presente nel database	La registrazione non è avvenuta e il centralinista riprova controllando se nel database è presente l'ID della telefonata inserito
5	Lunghezza ID telefonata valida, esito non valido	Lunghezza ID telefonata = 5, esito diverso da ("occupato", "senza risposta", "da	L'ID della telefonata è valido, in quanto le cifre sono 5, l'esito non è	{ ID telefonata : "11118", esito: "libero"}	Registrazione non effettuata perché l'esito non è riconosciuto	La registrazione non è avvenuta e il centralinista riprova digitando un esito tra quelli prefissati

		richiamare”, “non interessato”, “appuntamento fissato”)	valido, in quanto non rientra tra quelli prefissati			
--	--	---------------------------------------------------------	-----------------------------------------------------	--	--	--

### 4.3 Rimuovi Utente

ID_LISTA	ID_UTENTE
<ul style="list-style-type: none"> <li>Lunghezza ID lista = 5</li> <li>Lunghezza ID lista &gt; 5 [ERROR]</li> <li>Lunghezza ID lista &lt; 5 [ERROR]</li> <li>ID non presente nel database [ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Lunghezza NumeroTelefonico = 10</li> <li>Lunghezza NumeroTelefonico &lt; 10 [ERROR]</li> <li>Lunghezza NumeroTelefonico &gt; 10 [ERROR]</li> <li>NumeroTelefonico non presente [ERROR]</li> </ul>

Il numero di test da effettuarsi senza particolari vincoli è:  $4 * 4 = 16$

Introduciamo i vincoli [ERROR] .

Il numero di test da eseguire per testare singolarmente i vincoli è 6 (3 per Lista, 3 per Utente).

Il numero di test risultante è:  $(1*1) + 6 = 7$ .

### TEST SUITE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti input validi	ID lista valido, cifre numero telefonico = 10	L' ID della lista è valido e presente nel database e il numero telefonico è del formato giusto	{ID lista: "11122", NumeroTelefonico: "3421789003"}	Rimozione effettuata con successo	Un numero telefonico di un utente è stato rimosso dalla lista
2	Lunghezza ID lista non valida, numero telefonico valido	Lunghezza ID lista > 5 [ERROR], lunghezza NumeroTelefonico = 10	L'ID della lista non è valido, in quanto le cifre sono maggiori di 5, il numero telefonico è del formato giusto	{ID lista: "11109", NumeroTelefonico: "3421789003"}	Rimozione non effettuata perché la lunghezza dell'ID lista è maggiore di 5	La rimozione non è avvenuta e il centralinista riprova digitando un ID lista di 5 cifre
3	Lunghezza ID lista non valida, numero telefonico valido	Lunghezza ID lista < 5 [ERROR], lunghezza NumeroTelefonico = 10	L'ID della lista non è valido, in quanto le cifre sono minori di 5, il numero telefonico è del formato giusto	{ID lista: "109", NumeroTelefonico: "3421789003"}	Rimozione non effettuata perché la lunghezza dell'ID lista è minore di 5	La rimozione non è avvenuta e il centralinista riprova digitando un ID lista di 5 cifre

4	ID lista non presente nel database	ID lista non presente nel database [ERROR], lunghezza NumeroTelefonico = 10	L'ID della lista non è presente nel database, il numero telefonico è del formato giusto	{ ID lista : "14142", NumeroTelefonico: "3421789003" }	Rimozione non effettuata perché l'ID della lista non è presente nel database	La rimozione non è avvenuta e il centralinista riprova controllando se nel database è presente l'ID della lista inserito
5	Numero telefonico non presente nel database	NumeroTelefonico non presente nel database [ERROR], lunghezza ID lista = 5	Il numero telefonico non è presente nel database, lunghezza ID lista = 5	{ ID lista : "11142", NumeroTelefonico: "3421781103" }	Rimozione non effettuata perché il numero telefonico non è presente nel database	La rimozione non è avvenuta e il centralinista riprova controllando se nel database è presente il numero telefonico inserito
6	Lunghezza ID lista valida, lunghezza numero telefonico non valida	Lunghezza ID lista = 5, lunghezza NumeroTelefonico < 10 [ERROR]	L'ID della lista è valido, in quanto le cifre sono 5, il numero telefonico non è del formato giusto	{ID lista: "11109", NumeroTelefonico: "34217893" }	Rimozione non effettuata perché la lunghezza del numero telefonico è minore di 10	La rimozione non è avvenuta e il centralinista riprova digitando un numero telefonico di 10 cifre
7	Lunghezza ID lista valida, lunghezza numero telefonico non valida	Lunghezza ID lista = 5, lunghezza NumeroTelefonico > 10 [ERROR]	L'ID della lista è valido, in quanto le cifre sono 5, il numero telefonico non è del formato giusto	{ID lista: "11109", NumeroTelefonico: "34217893221" }	Rimozione non effettuata perché la lunghezza del numero telefonico è maggiore di 10	La rimozione non è avvenuta e il centralinista riprova digitando un numero telefonico di 10 cifre

#### 4.4 Aggiungi utente

ID_LISTA	ID_UTENTE
<ul style="list-style-type: none"> <li>Lunghezza ID lista = 5</li> <li>Lunghezza ID lista &gt; 5 [ERROR]</li> <li>Lunghezza ID lista &lt; 5 [ERROR]</li> <li>ID non presente nel database [ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Lunghezza NumeroTelefonico &lt; 10 [ERROR]</li> <li>Lunghezza NumeroTelefonico &gt; 10 [ERROR]</li> <li>Lunghezza NumeroTelefonico = 10</li> <li>NumeroTelefonico non presente [ERROR]</li> </ul>

Il numero di test da effettuarsi senza particolari vincoli è:  $4 * 4 = 12$

Introduciamo i vincoli [ERROR] .

Il numero di test da eseguire per testare singolarmente i vincoli è 6 (3 per Lista, 3 per Utente).

Il numero di test risultante è:  $(1*1) + 6 = 7$ .

## TEST SUITE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti input validi	ID lista valido, cifre numero telefonico = 10	L' ID della lista è valido e presente nel database e il numero telefonico è del formato giusto	{ID lista: "11122", NumeroTelefoni co: "3421789003"}	Aggiunta effettuata con successo	Un numero telefonico di un utente è stato aggiunto in lista
2	Lunghezza ID lista non valida, numero telefonico valido	Lunghezza ID lista > 5 [ERROR], lunghezza NumeroTelefoni co = 10	L'ID della lista non è valido, in quanto le cifre sono maggiori di 5, il numero telefonico è del formato giusto	{ID lista: "111099", NumeroTelefoni co: "3421789003"}	Aggiunta non effettuata perché la lunghezza dell'ID lista è maggiore di 5	L'aggiunta non è avvenuta e il centralinista riprova digitando un ID lista di 5 cifre
3	Lunghezza ID lista non valida, numero telefonico valido	Lunghezza ID lista < 5 [ERROR], lunghezza NumeroTelefoni co = 10	L'ID della lista non è valido, in quanto le cifre sono minori di 5, il numero telefonico è del formato giusto	{ID lista: "109", NumeroTelefoni co: "3421789003"}	Aggiunta non effettuata perché la lunghezza dell'ID lista è minore di 5	L'aggiunta non è avvenuta e il centralinista riprova digitando un ID lista di 5 cifre
4	ID lista non presente nel database	ID lista non presente nel database [ERROR], lunghezza NumeroTelefoni co = 10	L'ID della lista non è presente nel database, il numero telefonico è del formato giusto	{ ID lista : "14142", NumeroTelefoni co: "3421789003"}	Aggiunta non effettuata perché l'ID della lista non è presente nel database	L'aggiunta non è avvenuta e il centralinista riprova controllando se nel database è presente l'ID della lista inserito
5	Numero telefonico non presente nel database	NumeroTelefoni co non presente nel database [ERROR], lunghezza ID lista = 5	Il numero telefonico non è presente nel database, lunghezza ID lista = 5	{ ID lista : "11142", NumeroTelefoni co: "3421781103"}	Aggiunta non effettuata perché il numero telefonico non è presente nel database	L'aggiunta non è avvenuta e il centralinista riprova controllando se nel database è presente il numero telefonico inserito
6	Lunghezza ID lista valida, lunghezza numero telefonico non valida	Lunghezza ID lista = 5, lunghezza NumeroTelefoni co < 10 [ERROR]	L'ID della lista è valido, in quanto le cifre sono 5, il numero telefonico non è del formato giusto	{ID lista: "11109", NumeroTelefoni co: "34217893"}	Aggiunta non effettuata perché la lunghezza del numero telefonico è minore di 10	L'aggiunta non è avvenuta e il centralinista riprova digitando un numero telefonico di 10 cifre
7	Lunghezza ID lista valida, lunghezza	Lunghezza ID lista = 5, lunghezza	L'ID della lista è valido, in quanto le cifre	{ID lista: "11109", NumeroTelefoni	Rimozione non effettuata perché la	L'aggiunta non è avvenuta e il centralinista



	numero telefonico non valida	NumeroTelefoni co > 10 [ERROR]	sono 5, il numero telefonico non è del formato giusto	co: "34217893221" }	lunghezza del numero telefonico è maggiore di 10	riprova digitando un numero telefonico di 10 cifre
--	------------------------------	-----------------------------------	-------------------------------------------------------	---------------------------	--------------------------------------------------	----------------------------------------------------

#### 4.5 Aggiungi centralista

ID	NOME	COGNOME	EMAIL
<ul style="list-style-type: none"> <li>Lunghezza ID centralinista = 5</li> <li>Lunghezza ID centralinista &gt; 5 [ERROR]</li> <li>Lunghezza ID centralinista &lt; 5 [ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Stringa di caratteri di lunghezza &lt;= 40</li> <li>Stringa di caratteri di lunghezza &gt; 40 [ERROR]</li> <li>Stringa non presente nel database [ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Stringa di caratteri di lunghezza &lt;= 40</li> <li>Stringa di caratteri di lunghezza &gt; 40 [ERROR]</li> <li>Stringa non presente nel database [ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Stringa di carattere contenente il carattere "@"</li> <li>Stringa di carattere non contenente il carattere "@" [ERROR]</li> <li>Email non presente nel database [ERROR]</li> </ul>

Il numero di test da effettuarsi senza particolari vincoli è:  $3 * 3 * 3 * 3 = 81$

Introduciamo i vincoli [ERROR] .

Il numero di test da eseguire per testare singolarmente i vincoli è 8 (2 per ID , 2 per nome, 2 per cognome, 2 per email).

Il numero di test risultante è:  $(1*1*1*1) + 8 = 9$ .

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti input validi	ID valido, Nome valido, Cognome valido, Email valido	L'ID, nome, cognome ed email sono valido e presenti nel database.	{ ID: "14411" , Nome : "Giovanni", Cognome : "Falco", Email : "falco@gmail.com" }	Aggiunta centralinista avvenuta con successo	Il centralinista è stato aggiunto
2	Lunghezza ID non valida	Lunghezza ID > 5 [ERROR], Nome valido, Cognome valido, Email valida	L'ID non è valido, in quanto le cifre sono maggiori di 5	{ID: "111089", Nome : "Giovanni", Cognome : "Falco", Email : "falco@gmail.com" }	Aggiunta centralinista non effettuata perché la lunghezza dell'ID è maggiore di 5	Il centralinista non è stato aggiunto e l'amministratore riprova digitando un ID di 5 cifre
3	Lunghezza ID non valida	Lunghezza ID < 5 [ERROR], Nome valido, Cognome valido, Email valida	L'ID non è valido, in quanto le cifre sono minori di 5	{ID lista: "109", Nome : "Giovanni", Cognome : "Falco", Email : "falco@gmail.com" }	Aggiunta centralinista non effettuata perché la lunghezza dell'ID è minore di 5	Il centralinista non è stato aggiunto e l'amministratore riprova digitando un ID di 5 cifre
4	Lunghezza nome non valida	ID valido, Stringa di caratteri del nome >40 [ERROR],	La stringa di caratteri del nome è maggiore	{ ID: "11761" , Nome : "aaaaaaaaa" }	Aggiunta centralinista non effettuata in quanto	Il centralinista non è stato aggiunto e l'amministratore

		Cognome valido, Email valida	di 40 caratteri, l'id , il cognome e l'email sono sia valide che presente nel database	aaaaaaaaaaaa aaaaaaaaaaaa aaaaaaaaaaaa", Cognome : "Falco", Email : "falco@gmail. com"} }	la stringa del nome non è corretta	dovrà controllare se ha inserito il nome corretto
5	Nome non presente nel database	ID valido, Nome non presente nel database [ERROR], Cognome valido, Email valida	La stringa di caratteri del nome non è presente nel database, l'id , il cognome e l'email sono sia valide che presente nel database	{ ID: "11221" , Nome :"Gianmaria", Cognome : "Falco", Email : "falco@gmail. com"} }	Aggiunta centralinista non effettuata in quanto la stringa del nome non è contenuta nel database	Il centralinista non è stato aggiunto e l'amministratore dovrà controllare il nome del centralinista da rimuovere nel database
6	Lunghezza cognome non valida	ID valido, Nome valido Stringa di caratteri del cognome >40 [ERROR], Email valida	La stringa di caratteri del cognome è maggiore di 40 caratteri, l'id , il cognome e l'email sono sia valide che presente nel database	{ ID: "10911" , Nome :"Giovanni", Cognome : " aaaaaaaaaaaa aaaaaaaaaaaa aaaaaaaaaaaa", Email : "falco@gmail. com"} }	Aggiunta centralinista non effettuata in quanto la stringa del cognome non è corretta	Il centralinista non è stato aggiunto e l'amministratore dovrà controllare se ha inserito il cognome corretto
7	Cognome non presente nel database	ID valido, Nome valido Stringa di caratteri del cognome non presente nel database [ERROR], Email valida	La stringa di caratteri del cognome non presente , l'id , il cognome e l'email sono sia valide che presente nel database	{ ID: "18711" , Nome :"Giovanni", Cognome : " Falchetto", Email : "falco@gmail. com"} }	Aggiunta centralinista non effettuata in quanto la stringa del cognome non è contenuta nel database	Il centralinista non è stato aggiunto e l'amministratore dovrà controllare il cognome del centralinista da rimuovere nel database
8	Email non del formato giusto	ID valido, Nome valido, cognome valido, Stringa di caratteri dell'email non contiene il carattere "@"	La stringa di caratteri dell'email non presenta il carattere "@", l'id , il cognome e il nome sono sia valide che presente nel database	{ ID: "11221" , Nome :"Giovanni", Cognome : " Falchetto", Email : "falcogmail.co m"} }	Aggiunta centralinista non effettuata in quanto la stringa dell'email non contiene il carattere "@"	Il centralinista non è stato aggiunto e l'amministratore dovrà controllare l' email del centralinista in quanto ha mancato il carattere "@"
9	Email non presente nel database	ID valido, Nome valido, cognome valido, l'email non è presente nel database	L'email non è contenuta nel database mentre il cognome e il nome sono sia valide che presente nel database	{ ID: "13111" , Nome :"Giovanni", Cognome : " Falchetto", Email : "falchettonnn e@hotmail.com "} }	Aggiunta centralinista non effettuata in quanto l'email non è presente nel database	Il centralinista non è stato aggiunto e l'amministratore dovrà controllare l'email del centralinista da rimuovere nel database

## 4.6 Rimuovi centralinista

ID
<ul style="list-style-type: none"> <li>Lunghezza ID centralinista = 5</li> <li>Lunghezza ID centralinista &gt; 5 [ERROR]</li> <li>Lunghezza ID centralinista &lt; 5 [ERROR]</li> <li>ID non presente nel database [ERROR]</li> </ul>

Il numero di test da effettuarsi senza particolari vincoli è: 3

Introduciamo i vincoli [ERROR] .

Il numero di test da eseguire per testare singolarmente i vincoli è 3 (3 per ID).

Il numero di test risultante è:  $1 + 3 = 4$ .

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti input validi	ID valido	L'ID è valido e presente nel database	{ ID centralinista: "11121" }	Rimozione centralinista avvenuta con successo	Il centralinista è stato eliminato
2	Lunghezza ID non valida	Lunghezza ID > 5 [ERROR]	L'ID non è valido, in quanto le cifre sono maggiori di 5	{ID centralinista: "111089"}	Rimozione centralinista non effettuata perché la lunghezza dell'ID è maggiore di 5	Il centralinista non è stato rimosso e l'amministratore riprova digitando un ID di 5 cifre
3	Lunghezza ID non valida	Lunghezza ID < 5 [ERROR]	L'ID non è valido, in quanto le cifre sono minori di 5	{ID centralinista: "109"}	Rimozione centralinista non effettuata perché la lunghezza dell'ID è minore di 5	Il centralinista non è stato rimosso e l'amministratore riprova digitando un ID di 5 cifre
4	ID non presente nel database	ID non presente nel database [ERROR]	L'id non è presente nel database	{ ID centralinista: "13311" }	Rimozione centralinista non effettuata in quanto l'id non è presente nel database	Il centralinista non è stato eliminato e l'amministratore dovrà controllare l'ID se presente nel database

## 4.7 Visualizza note appuntamenti fissati

ID
<ul style="list-style-type: none"><li>• Lunghezza ID appuntamento = 5</li><li>• Lunghezza ID appuntamento &gt; 5 [ERROR]</li><li>• Lunghezza ID appuntamento &lt; 5 [ERROR]</li><li>• ID non presente nel database [ERROR]</li></ul>

Il numero di test da effettuarsi senza particolari vincoli è: 1

Introduciamo i vincoli [ERROR] .

Il numero di test da eseguire per testare singolarmente i vincoli è 3 (3 per Appuntamento).

Il numero di test risultante è:  $1 + 3 = 4$ .

### TEST SUITE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Input valido	ID appuntamento corretto	L'ID dell'appuntamento è valido e presente nel database	{ID appuntamento o: "11331" }	Appuntamento riconosciuto	L'agente vendita controlla le informazioni relative all'appuntamento
2	Lunghezza ID non valida	Lunghezza ID > 5 [ERROR]	L'ID non è valido, in quanto le cifre sono maggiori di 5	{ID appuntamento o: "111089" }	Appuntamento non riconosciuto perché la lunghezza dell'ID è maggiore di 5	L'appuntamento non è stato riconosciuto e l'agente vendita riprova digitando un ID di 5 cifre
3	Lunghezza ID non valida	Lunghezza ID < 5 [ERROR]	L'ID non è valido, in quanto le cifre sono minori di 5	{ID appuntamento o: "109" }	Appuntamento non riconosciuto perché la lunghezza dell'ID è minore di 5	L'appuntamento non è stato riconosciuto e l'agente vendita riprova digitando un ID di 5 cifre
4	ID non presente nel database	ID non presente nel database [ERROR]	L'id non è presente nel database	{ID appuntamento o: "13311" }	Appuntamento non riconosciuto in quanto l'ID non è presente nel database	L'appuntamento non è stato riconosciuto e l'agente vendita dovrà controllare l'ID se presente nel database

## 4.8 Rimuovi Lista

ID
<ul style="list-style-type: none"><li>• Lunghezza ID lista = 5</li><li>• Lunghezza ID lista &gt; 5 [ERROR]</li><li>• Lunghezza ID lista &lt; 5 [ERROR]</li><li>• ID non presente nel database [ERROR]</li></ul>

Il numero di test da effettuarsi senza particolari vincoli è: 4  
Introduciamo i vincoli [ERROR] .

Il numero di test da eseguire per testare singolarmente i vincoli è 3 (3 per ID).  
Il numero di test risultante è:  $1 + 3 = 4$ .

### TEST SUITE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Input valido	ID lista corretto	L'ID lista è valido, in quanto è maggiore di 0	{ID lista: "11331" }	Rimozione lista avvenuta con successo	La lista è stata eliminata
2	Lunghezza ID non valida	Lunghezza ID lista > 5 [ERROR]	L'ID lista non è valido, in quanto le cifre sono maggiori di 5	{ID lista: "111089"}	Rimozione lista non effettuata perché la lunghezza dell'ID è maggiore di 5	La lista non è stata rimossa e l'amministratore riprova digitando un ID lista di 5 cifre
3	Lunghezza ID non valida	Lunghezza ID lista < 5 [ERROR]	L'ID lista non è valido, in quanto le cifre sono minori di 5	{ID lista: "109"}	Rimozione lista non effettuata perché la lunghezza dell'ID è minore di 5	La lista non è stata rimossa e l'amministratore riprova digitando un ID lista di 5 cifre
4	ID non presente nel database	ID lista non presente nel database [ERROR]	L'ID non è presente nel database	{ ID lista: "13311" }	Rimozione lista non effettuata in quanto l'ID non è presente nel database	La lista non è stata eliminata e l'amministratore dovrà controllare l'ID se presente nel database

## 4.9 Rimuovi Gruppo

ID
<ul style="list-style-type: none"><li>• Lunghezza ID lista = 5</li><li>• Lunghezza ID lista &gt; 5 [ERROR]</li><li>• Lunghezza ID lista &lt; 5 [ERROR]</li><li>• ID non presente nel database [ERROR]</li></ul>

Il numero di test da effettuarsi senza particolari vincoli è: 4  
Introduciamo i vincoli [ERROR] .

Il numero di test da eseguire per testare singolarmente i vincoli è 3 (3 per ID).

Il numero di test risultante è:  $1 + 3 = 4$ .

### TEST SUITE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Input valido	ID lista corretto	L'ID gruppo è valido, in quanto le cifre sono 5	{ID gruppo: "11331" }	Rimozione gruppo avvenuta con successo	Il gruppo è stato eliminato
2	Lunghezza ID non valida	Lunghezza ID gruppo > 5 [ERROR]	L'ID gruppo non è valido, in quanto le cifre sono maggiori di 5	{ID gruppo: "111089"}	Rimozione gruppo non effettuata perché la lunghezza dell'ID è maggiore di 5	Il gruppo non è stato rimosso e l'amministratore riprova digitando un ID lista di 5 cifre
3	Lunghezza ID non valida	Lunghezza ID gruppo < 5 [ERROR]	L'ID gruppo non è valido, in quanto le cifre sono minori di 5	{ID gruppo: "109"}	Rimozione gruppo non effettuata perché la lunghezza dell'ID è minore di 5	Il gruppo non è stato rimosso e l'amministratore riprova digitando un ID lista di 5 cifre
4	ID non presente nel database	ID gruppo non presente nel database [ERROR]	L'ID gruppo non è presente nel database	{ ID gruppo: "13311" }	Rimozione gruppo non effettuata in quanto l'ID non è presente nel database	Il gruppo non è stato eliminato e l'amministratore dovrà controllare l'ID se presente nel database

## 4.10 Modifica note appuntamenti

ID	NOTE
<ul style="list-style-type: none"> <li>Lunghezza ID appuntamento = 5</li> <li>Lunghezza ID appuntamento &gt; 5 [ERROR]</li> <li>Lunghezza ID appuntamento &lt; 5 [ERROR]</li> <li>ID non presente nel database [ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Lunghezza note &lt; 45</li> <li>Lunghezza note &gt; 45 [ERROR]</li> </ul>

Il numero di test da effettuarsi senza particolari vincoli è: 6

Introduciamo i vincoli [ERROR] .

Il numero di test da eseguire per testare singolarmente i vincoli è 4 (3 per Appuntamento, 1 per Note).

Il numero di test risultante è:  $(1 \cdot 1) + 4 = 5$ .

### TEST SUITE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti input validi	ID appuntamento corretto, lunghezza note valida	L'ID dell'appuntamento è valido, in quanto le cifre sono 5, le note sono valide, in quanto non superano i 45 caratteri	{ID appuntamento o: "16511", Note: "Appuntamento stabilito secondo delle direttive" }	Appuntamento riconosciuto e note modificabili	L'appuntamento è identificato e l'agente vendita ha modificato le note
2	Lunghezza ID appuntamento corretta, lunghezza note non corretta	Lunghezza ID appuntamento corretta, lunghezza note > 45 [ERROR]	L'ID dell'appuntamento è valido, in quanto le cifre sono 5, le note non sono valide, in quanto superano i 45 caratteri	{ID appuntamento o: "12331", Note: "Appuntamento stabilito secondo delle direttive [...]"} }	Note appuntamento non modificate poiché la lunghezza è maggiore di 45 caratteri	L'appuntamento è stato identificato, l'agente vendita riprova controllando che la lunghezza delle note non superi i 45 caratteri
3	Lunghezza ID appuntamento non corretta, lunghezza note corretta	Lunghezza ID appuntamento > 5 [ERROR], lunghezza note < 45	L'ID dell'appuntamento non è valido, in quanto le cifre sono maggiori di 5, le note sono valide, in quanto non superano i 45 caratteri	{ ID appuntamento o: "112633", Note: "Appuntamento stabilito secondo delle direttive" }	Appuntamento non riconosciuto poiché la lunghezza dell'ID è maggiore di 5 caratteri	L'appuntamento non è stato identificato e l'agente vendita riprova digitando un ID appuntamento di 5 cifre

4	Lunghezza ID appuntamento non corretta, lunghezza note corretta	Lunghezza ID appuntamento < 5 [ERROR], lunghezza note < 45	L'ID dell'appuntamento o non è valido, in quanto le cifre sono minori di 5, le note sono valide, in quanto non superano i 45 caratteri	{ ID appuntamento o : "1633", Note: "Appuntamento stabilito secondo delle direttive" }	Appuntamento non riconosciuto poiché la lunghezza dell'ID è minore di 5 caratteri	L'appuntamento non è stato identificato e l'agente vendita riprova digitando un ID appuntamento di 5 cifre
5	ID appuntamento non corretto, lunghezza note < 45	ID telefonata non presente nel database [ERROR], lunghezza note < 45	L'ID dell'appuntamento o non è valido, in quanto non è stato trovato nel database, le note sono valide, in quanto non superano i 45 caratteri	{ ID appuntamento o : "11421", Note: "Appuntamento stabilito secondo delle direttive" }	Appuntamento non trovato	L'appuntamento non è stato identificato e l'agente vendita riprova digitando un ID tra quelli riconosciuti nel database

#### 4.11 Creazione Appuntamento

ID	DATA	ORA	NOTE
<ul style="list-style-type: none"> <li>Lunghezza ID appuntamento = 5</li> <li>Lunghezza ID appuntamento &gt; 5 [ERROR]</li> <li>Lunghezza ID appuntamento &lt; 5 [ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Formato data valido</li> <li>Formato data non valido [ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Formato ora valido</li> <li>Formato ora non valido [ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Lunghezza note &lt; 45</li> <li>Lunghezza note &gt; 45 [ERROR]</li> </ul>

Il numero di test da effettuarsi senza particolari vincoli è:  $3 * 2 * 2 * 2 = 24$

Introduciamo i vincoli [ERROR] .

Il numero di test da eseguire per testare singolarmente i vincoli è 5 (2 per appuntamento, 1 per data, 1 per ora, 1 per note).

Il numero di test risultante è:  $(1 * 1 * 1 * 1) + 5 = 6$ .

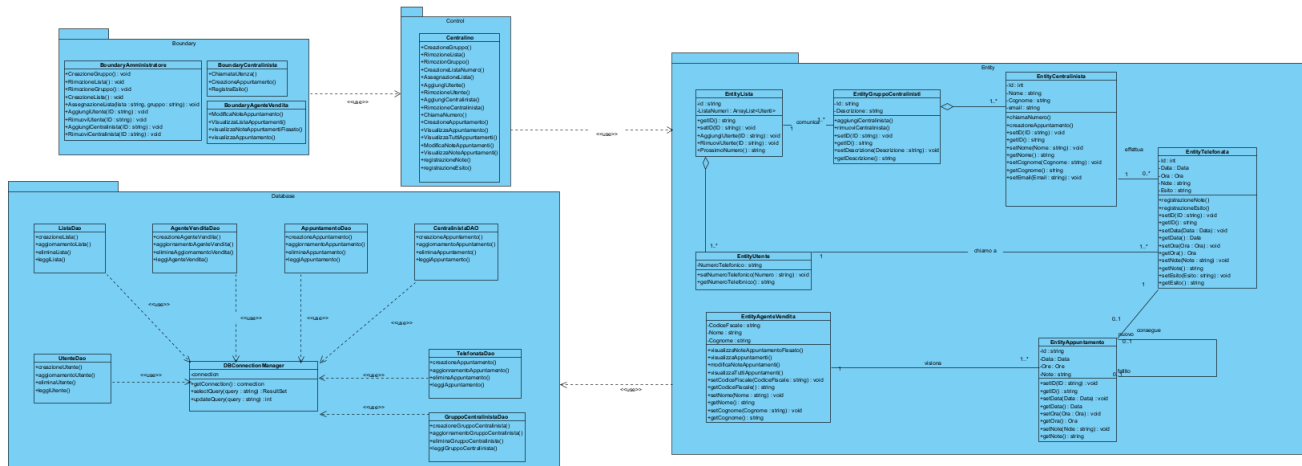
Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Input tutti validi	ID appuntamento valido, formato data valido, formato ora valido, lunghezza note < 45	Formato data valido, Formato ora valido, note valide	{ID: "14411", Data: "12-03-2022", Ora: "12:30", Note: "Appuntamento stabilito secondo delle direttive"}	Creazione appuntamento avvenuta con successo	Appuntamento creato
2	Lunghezza ID appuntamento non corretta	Lunghezza ID appuntamento > 5 [ERROR]	L'ID dell'appuntamento non è valido, in quanto le cifre	{ ID appuntamento : "112633" }	Appuntamento non creato poiché la lunghezza dell'ID è	L'appuntamento o non è stato identificato e il centralinista riprova



			sono maggiori di 5		maggiore di 5 caratteri	digitando un ID appuntamento di 5 cifre
3	Lunghezza ID appuntamento non corretta	Lunghezza ID appuntamento < 5 [ERROR]	L'ID dell'appuntamento non è valido, in quanto le cifre sono minori di 5	{ ID appuntamento : "1633"}	Appuntamento non creato poiché la lunghezza dell'ID è minore di 5 caratteri	L'appuntamento non è stato identificato e il centralinista riprova digitando un ID appuntamento di 5 cifre
4	Formato data non valido	Formato data non valido [ERROR]	La data non ha un formato corretto	{ID: "13611", Data:"12-13-2022", Ora:"12:30", note: " Appuntamento stabilito secondo delle direttive"}	Data non valida	Appuntamento non creato
5	Formato ora non valido	Formato ora non valido [ERROR]	L'ora non ha un formato corretto	{ ID: "13351", Data:"12-03-2022", Ora:"12:70", note: " Appuntamento stabilito secondo delle direttive"}	Ora non valida	Appuntamento non creato
6	Lunghezza note non valida	Lunghezza note > 45 [ERROR]	Le note hanno una dimensione superiore a quella consentita	{ ID: "13311", Data:"12-03-2022", Ora:"12:30", note: " Appuntamento stabilito secondo delle direttive [...]"}	Lunghezza note maggiore di quella consentita	Appuntamento non creato

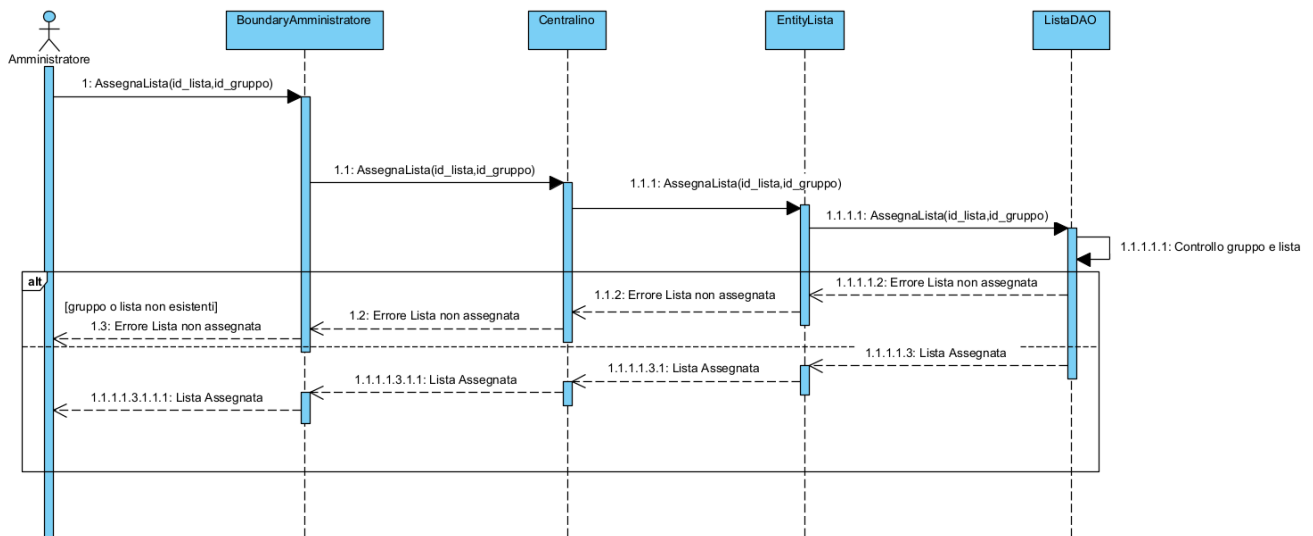
## 5. PROGETTAZIONE

### 5.1 DIAGRAMMA DELLE CLASSI CON PATTERN ARCHITETTURALI BCED

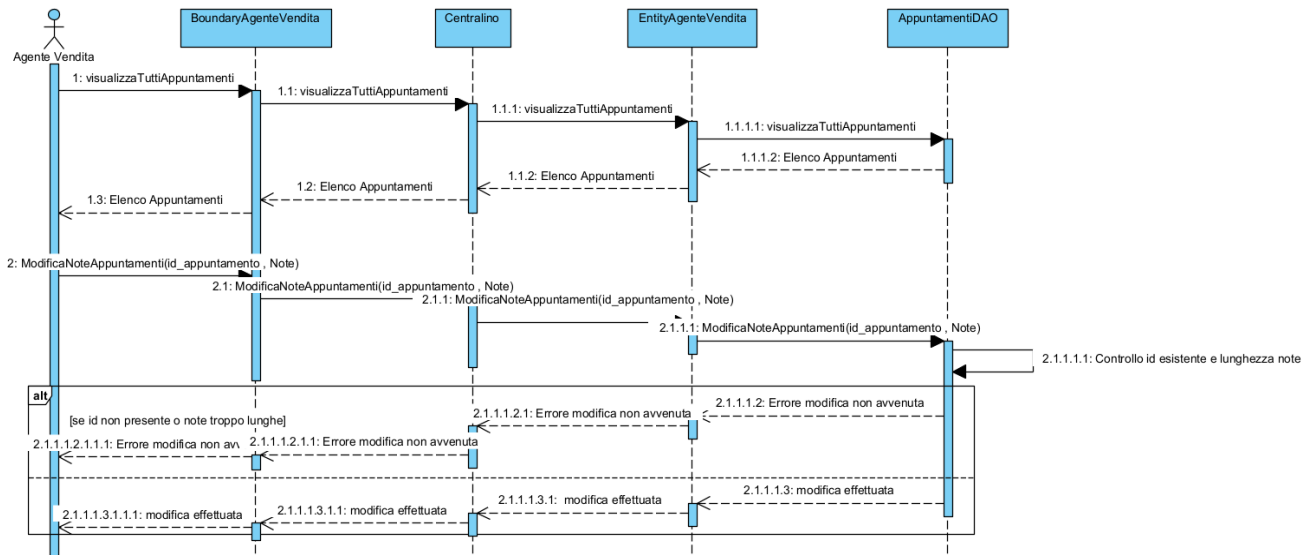


### 5.2 DIAGRAMMA DI SEQUENZA

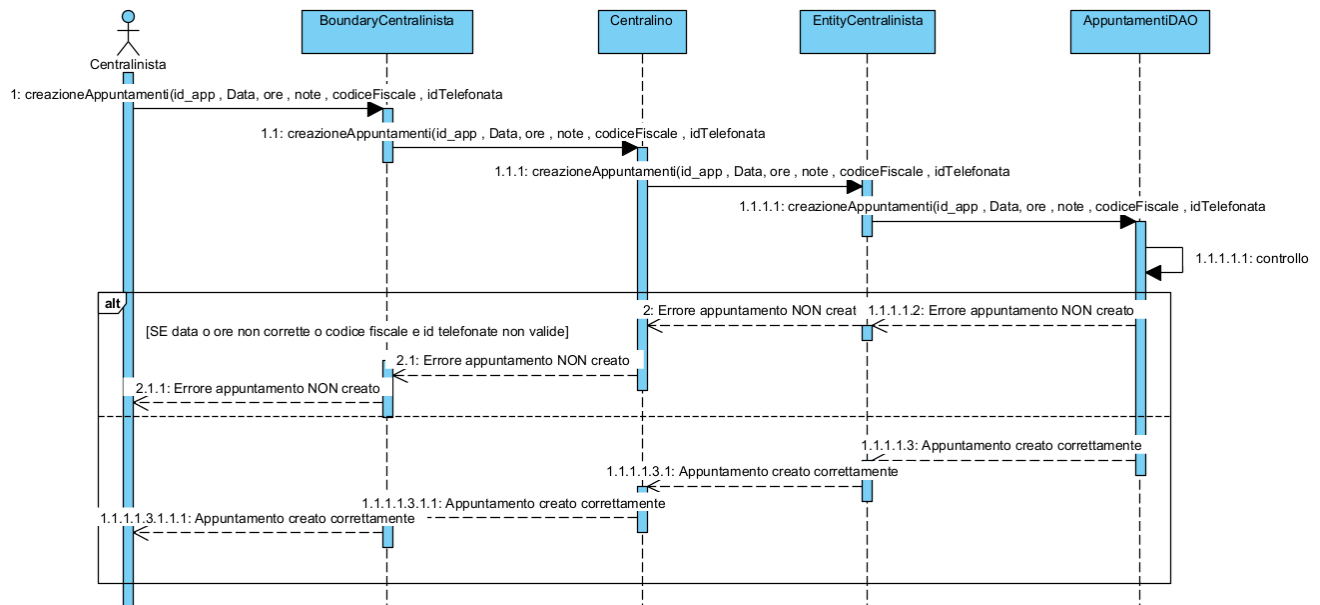
#### 5.2.1 Assegna Lista



## 5.2.2 Interazione con lista appuntamenti



## 5.2.3 Creazione Appuntamento



# 6 . IMPLEMENTAZIONE

## 6.1 ARTICOLAZIONE DEL SOFTWARE

Il Software è stato realizzato secondo il protocollo BCED e ritroviamo i seguenti packages :

## Package Boundary

**Amministratore:** Interfaccia grafica per le operazioni che devono essere effettuate dall'amministratore, ritroviamo le seguenti funzioni: crea gruppo, crea lista, rimuovi lista, assegna lista, aggiungi utente, rimuovi utente, aggiungi centralinista, rimuovi centralinista, e rimuovi gruppo.

**Agente Vendita:** Interfaccia grafica per le operazioni che devono essere effettuate dall'agente vendita, ritroviamo le seguenti funzioni: visualizza appuntamento, visualizza tutti appuntamenti, visualizza note appuntamenti fissati e modifica note appuntamento.

**Centralinista:** Interfaccia grafica per le operazioni che devono essere effettuate dal centralinista, ritroviamo le seguenti funzioni: chiama numero, registrazione esito e creazione appuntamento.

## Package Controller

**Centralino:** centralino funge da intermediario tra il Boundary (interfaccia utente) e l'entity (entità). Quindi elabora le richieste provenienti dal boundary e richiamando le opportune operazioni o metodi sull'entity per gestire la logica di business.

## Package Entity

Il Package Entity prende in considerazione gli oggetti di natura persistente che interagiscono tra loro e che compongono il sistema software in elaborazione. Tali sono:

**AgentiVendita:** Entità che è descritta dagli attributi "Codice Fiscale, Nome, Cognome" che si occupa di visionare/modificare gli appuntamenti creati.

**Appuntamenti:** Entità che è descritta dagli attributi "ID, Data, Ora, Note" che è utilizzata dagli agenti vendita e dall'entità Telefonata, poiché eventuale conseguenza di quest'ultima. È presente anche una ricorsione dell'entità stessa, in quanto un appuntamento può collegarsi ad un altro appuntamento per eventualità che il primo fallisca.

**Centralinisti:** Entità che è descritta dagli attributi "ID, Nome, Cognome, e-mail", che interagisce con l'entità Telefonata, dato che il compito del centralinista è quello di effettuare chiamate per poi registrarne l'esito, ed eventualmente creare appuntamenti.

**GruppoCentralinisti:** Entità che è descritta dagli attributi "ID, Descrizione", che rappresenta un insieme di centralinisti al quale vengono assegnate più liste di numeri telefonici che i singoli centralinisti chiameranno

**Liste:** Entità che è descritta dagli attributi "ID, ListaNumeri", che rappresenta l'insieme dei numeri telefonici da attribuire ai gruppi dei centralinisti

**Telefonate:** Entità che è descritta dagli attributi "ID, Data, Ora, Note, Esito" rappresenta la telefonata avvenuta tra centralinista e utente

**Utenti:** Entità che è descritta dall'attributo "NumeroTelefonico" rappresenta l'utente chiamato

## Package Database

Il Package Database è un'interfaccia verso il database. Ritroviamo :

**DBConnectionManager:** classe che si interfaccia con il Database e contiene funzioni per creare una connessione con il Database (get connection), informare il DB di richieste di operazioni (select query e update query).

**AgentiVenditaDAO:** Utilizzato per la descrizione dei singoli agenti vendita. Contiene funzioni set, get, costruttori, carica da dB che crea una connessione al DB per ricavare tutte le informazioni dal database.

**AppuntamentiDAO:** Utilizzato per la descrizione degli appuntamenti. Contiene funzioni set, get, costruttori , carica da db in più alcune query per interrogare il database per alcune funzionalità come visualizza appuntamenti, modifica appuntamenti.

**CentralinistaDAO:** Utilizzato per la descrizione dei centralinisti. Contiene funzioni set, get, costruttori, carica da db, per interrogare il database per alcune funzionalità come aggiungi centralinista, rimuovi centralinista e aggiorna centralinista.

**GruppoCentralinistiDAO:** Utilizzato per la descrizione dei gruppi. Contiene funzioni set, get, costruttori, carica db e alcune query utili per interrogare il database per alcune funzionalità come inserisci gruppo, rimuovi gruppo, aggiorna gruppo.

**ListaDAO:** Utilizzato per la descrizione delle liste. Contiene funzioni set, get, costruttori, carica db e alcune query utili per interrogare il database per alcune funzionalità come inserisci lista, rimuovi lista, aggiorna lista.

**TelefonateDAO:** Utilizzato per la descrizione delle telefonate. Contiene funzioni set, get, costruttori, carica db.

**UtenteDAO:** Utilizzato per la descrizione delle liste. Contiene funzioni set, get, costruttori, carica db e alcune query utili per interrogare il database per alcune funzionalità come inserisci utente, rimuovi utente, aggiorna utente.

## 6.2 STRUMENTI NECESSARI

Per consentire la comunicazione con il database abbiamo utilizzato una libreria: **mysql-connector-java-8.0.30.jar**.

Per l'interfaccia abbiamo utilizzato la classe **JFrame** della libreria Swing di Java, che ci ha consentito di creare finestre con cui l'utente può interagire.

Un **Main** che avvia la Gui.

Per la stesura della documentazione progettuale è stato utilizzato **Word**.

Per la produzione dei grafici abbiamo utilizzato **Visual Paradigm**.

LOC e LLOC : le righe di codice effettive sono : 3359.

## 7. TESTING

### 7.1 Test Strutturale

#### 7.1.1 Assegnazione lista

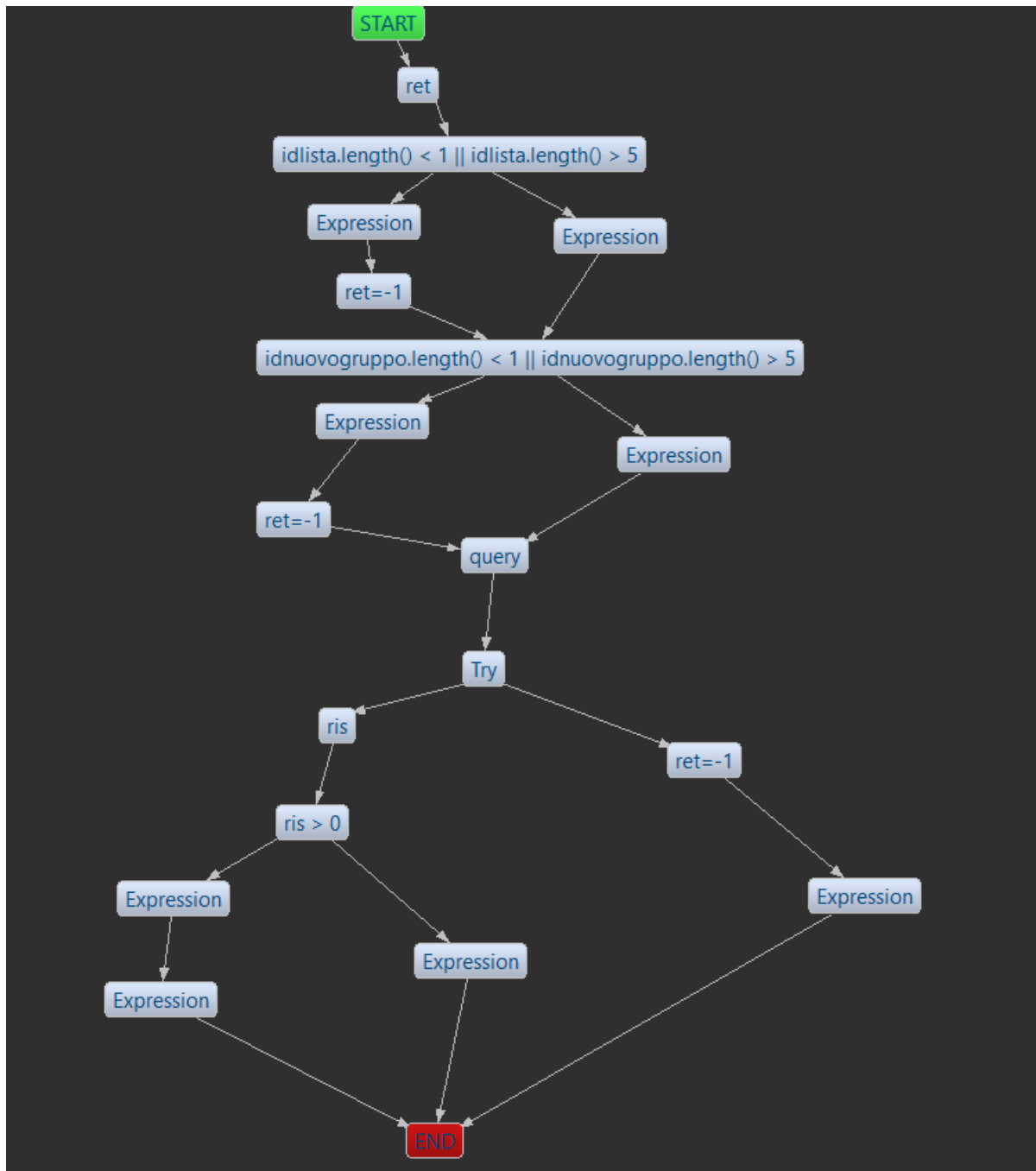
Di seguito riportiamo il codice Java del metodo update utilizzata per assegnare lista :

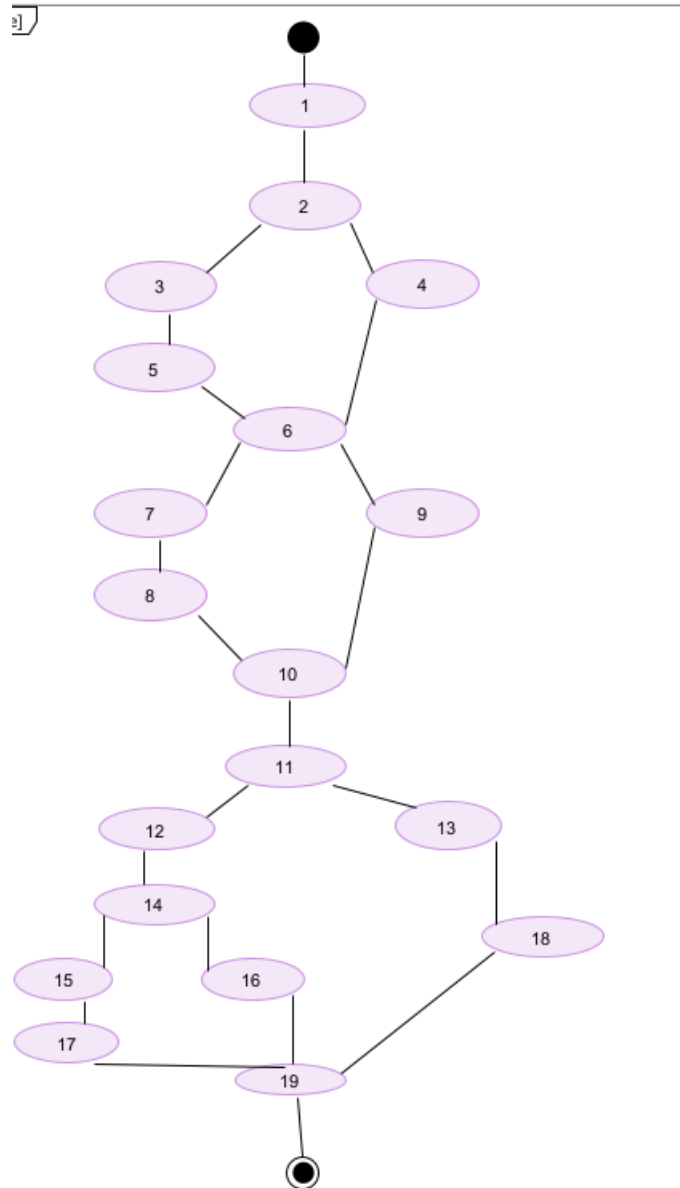
```
public int update(String idlista, String idnuovogruppo) {
    int ret = 0;
    if(idlista.length()<1 || idlista.length()>5) {
        System.out.println(" \n errore id lista non valida \n ") ;
        ret=-1 ;
    }else {
        System.out.println(" \n id lista valida \n ") ;
    }
    if(idnuovogruppo.length()<1 || idnuovogruppo.length()>5) {
        System.out.println(" \n errore id gruppo non valido \n ") ;
        ret=-1 ;
    }
}
```

```

    }else {
        System.out.println( "\n id lista valido \n " ) ;
    }
    String query = new String("UPDATE liste SET idgruppocentralinisti =
'" + idnuovogruppo + "' WHERE idLista= '" + idlista + "'");
    try {
        int ris = DBConnectionManager.updateQuery(query);
        if (ris > 0) {
            System.out.println("Lista aggiornata :\n");
            System.out.println(" \n ID nuovo : " + idnuovogruppo);
        } else {
            System.out.println("\n Lista non aggiornata \n ");
        }
    } catch (ClassNotFoundException | SQLException e) {
        ret = -1;
        e.printStackTrace();
    }
    return ret;
}

```





### NUMERO CICLOMATICO

il numero dei cammini linearmente indipendenti di un programma è pari al numero cicломatico di

**McCabe:**

- $V(G) = \text{Numero di regioni chiuse del grafo} + 1 = 4 + 1 = 5$
- $V(G) = \text{Numero di nodi predicato (2,6,11,15)} + 1 = 4 + 1 = 5$
- $V(G) = \#archi - \#nodi + 2 = 22 - 19 + 2 = 5$

**CAMMINI :**

- 1) 1-2-4-6-9-10-11-13-18-19
- 2) 1-2-3-5-6-7-8-10-11-12-14-15-17-19
- 3) 1-2-3-5-6-7-8-10-11-12-14-16-19
- 4) 1-2-3-5-6-7-8-10-11-13-18-19
- 5) 1-2-4-6-7-8-10-11-12-14-16-19



### 7.1.2 Modifica Note Appuntamenti

```
public void modificaNoteAppuntamenti ( String id , String note)
{
    if(id.length()<1 || id.length()>5) {
        System.out.println("\n errore id non valido \n ");
    }else {
        System.out.println("\n id e' scelto : "+ id+" \n ");
    }

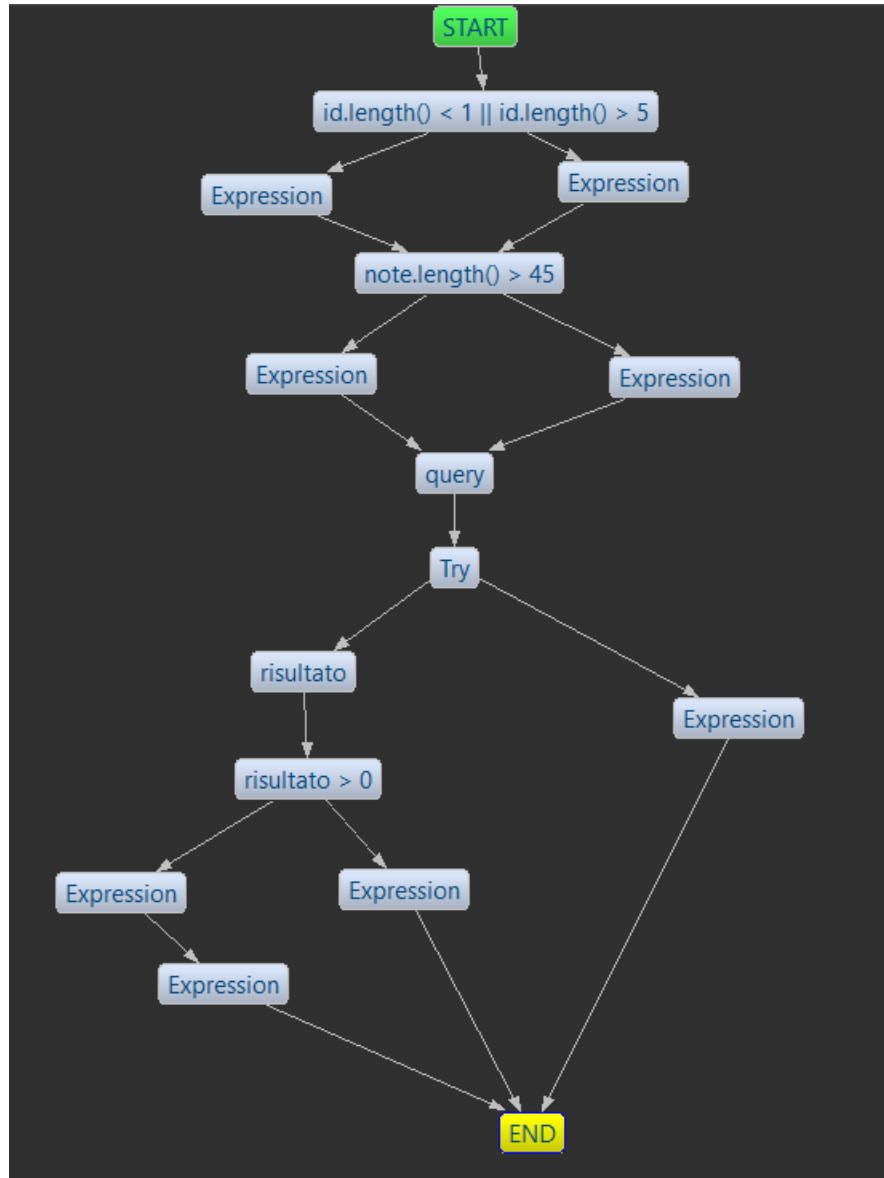
    if(note.length()>45) {
        System.out.println("\n note troppo lunghe , riprova !\n ");
    }else {
        System.out.println(" \n note scelte :"+ note+" \n") ;
    }

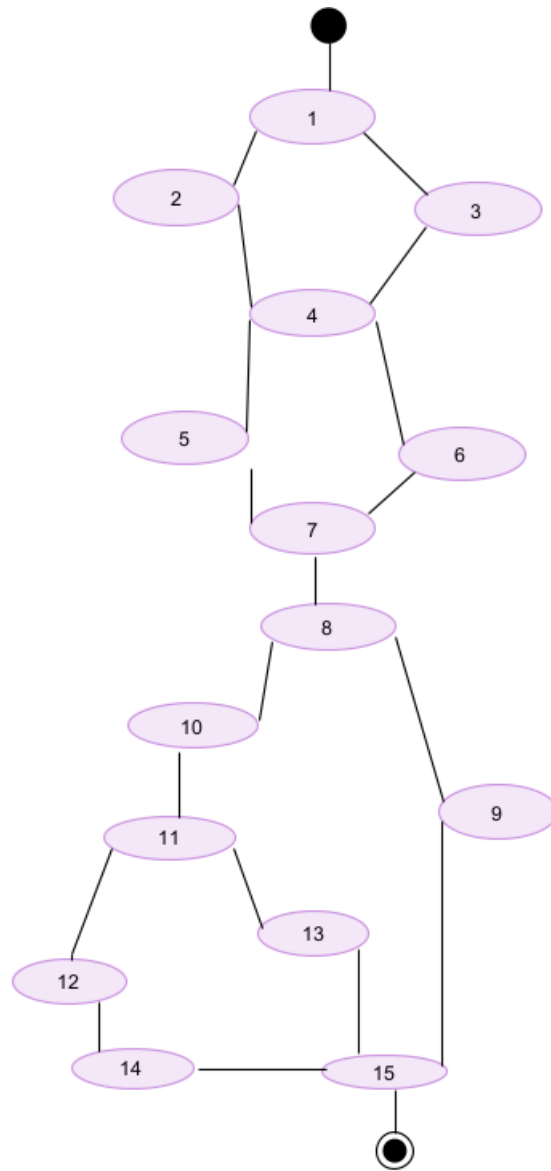
    String query = new String ("UPDATE Appuntamenti SET note ='"+ note+"'
WHERE IdAppuntamento ='"+id+"'") ;

    try {
        int risultato = DBConnectionManager.updateQuery(query) ;

        if(risultato>0) {
            System.out.println("NOTE MODIFICATE:");
            System.out.println("ID: " + id + ", note: " + note);
        }else {
            System.out.println(" l'id scelto è sbagliato , riprovare
!") ;
        }

    }catch(ClassNotFoundException |SQLException e) {
        e.printStackTrace();
    }
}
```





### NUMERO CICLOMATICO

il numero dei cammini linearmente indipendenti di un programma è pari al numero cicломatico di

#### McCabe:

- $V(G) = \text{Numero di regioni chiuse del grafo} + 1 = 4 + 1 = 5$
- $V(G) = \text{Numero di nodi predicato (1-4-8-11)} + 1 = 4 + 1 = 5$
- $V(G) = \# \text{archi} - \# \text{nodi} + 2 = 18 - 15 + 2 = 5$

#### CAMMINI :

- 1) 1-3-4-6-7-8-9-15
- 2) 1-2-4-5-7-8-10-11-12-14-15
- 3) 1-2-4-5-7-8-10-11-13-15
- 4) 1-2-4-6-7-8-9-15

5) 1-3-4-5-7-8-10-11-13-15

## 7.2 Test Funzionale

### 7.2.1 Assegnazione lista

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Output ottenuti	Post-condizioni ottenute	Esito
1	Tutti input validi	ID lista valido, ID gruppo valido	L'ID della lista e l'ID del gruppo centralinist i sono validi e presenti nel database.	{ ID lista : "11111" , ID gruppo : "22222" }	Assegnazione e effettuata	Il gruppo di centralinisti ora possiede una lista	"Lista assegnata con successo"	Sono soddisfatti tutti i requisiti, la lista è assegnata	PASS
2	Lunghezza ID lista non valida, lunghezza ID gruppo valida	Lunghezza ID lista > 5 [ERROR], Lunghezza ID gruppo = 5	L'ID della lista non è valido in quanto le cifre sono maggiori di 5, l'ID del gruppo centralinist i è sia valido che presente nel database	{ID lista: "100113", ID gruppo: "1111" }	Assegnazione e non effettuata perché la lunghezza dell'ID lista è maggiore di 5	L'assegnazione della lista non è avvenuta e l'amministratore riprova digitando un ID lista di 5 cifre	"Lista non assegnata, ID della lista troppo lungo"	Non sono soddisfatti tutti i requisiti, si ha un'avvertenza riguardo l'ID lista sbagliato	FAIL
3	Lunghezza ID lista non valida, lunghezza ID gruppo valida	Lunghezza ID lista < 5 [ERROR], Lunghezza ID gruppo = 5	L'ID della lista non è valido in quanto le cifre sono minori di 5, l'ID del gruppo centralinist i è sia valido che presente nel database	{ID lista: "1001", ID gruppo: "1111" }	Assegnazione e non effettuata perché la lunghezza dell'ID lista è minore di 5	L'assegnazione della lista non è avvenuta e l'amministratore riprova digitando un ID lista di 5 cifre	"Lista non assegnata, ID della lista troppo corto"	Non sono soddisfatti tutti i requisiti, si ha un'avvertenza riguardo l'ID lista sbagliato	FAIL
4	ID lista non presente nel database	ID lista non presente nel database [ERROR], ID gruppo valido	L'ID della lista non è presente nel database, l'id del gruppo centralinist i è sia valido che presente	{ ID lista : "11122", ID gruppo: "11143" }	Assegnazione e non effettuata perché l'ID della lista non è presente nel database	L'assegnazione della lista non è avvenuta e l'amministratore riprova controllando se nel database è presente l'id della lista inserito	"Lista non assegnata, ID della lista non trovato"	Non sono soddisfatti tutti i requisiti, si ha un'avvertenza riguardo l'ID lista non disponibile	FAIL

			nel database						
5	ID gruppo non presente nel database	ID gruppo non presente nel database [ERROR], ID lista valido	L'ID del gruppo non è presente nel database, l'id della lista è sia valido che presente nel database	{ ID lista : "11143", ID gruppo: "11122" }	Assegnazione e non effettuata perché l'ID del gruppo non è presente nel database	L'assegnazione della lista non è avvenuta e l'amministratore riprova controllando se nel database è presente l'id del gruppo inserito	"Lista non assegnata, ID troppo lungo"	Non sono soddisfatti tutti i requisiti, si ha un'avvertenza riguardo l'ID gruppo non disponibile	FAIL
6	Lunghezza ID lista valida, lunghezza ID gruppo non valida	Lunghezza ID lista = 5, Lunghezza ID gruppo > 5 [ERROR]	L'ID della lista è valido, l'ID del gruppo centralinisti non è valido in quanto le cifre sono maggiori di 5	{ID lista: "10011", ID gruppo: "11114" }	Assegnazione e non effettuata perché la lunghezza dell'ID gruppo è maggiore di 5	L'assegnazione e della lista non è avvenuta e l'amministratore riprova digitando un ID gruppo di 5 cifre	"Lista non assegnata, ID del gruppo troppo lungo"	Non sono soddisfatti tutti i requisiti, si ha un'avvertenza riguardo l'ID gruppo sbagliato	FAIL
7	Lunghezza ID lista valida, lunghezza ID gruppo non valida	Lunghezza ID lista = 5, Lunghezza ID gruppo < 5 [ERROR]	L'ID della lista è valido, l'ID del gruppo centralinisti non è valido in quanto le cifre sono minori di 5	{ID lista: "10011", ID gruppo: "11" }	Assegnazione e non effettuata perché la lunghezza dell'ID gruppo è minore di 5	L'assegnazione e della lista non è avvenuta e l'amministratore riprova digitando un ID gruppo di 5 cifre	"Lista non assegnata, ID troppo corto"	Non sono soddisfatti tutti i requisiti, si ha un'avvertenza riguardo l'ID gruppo sbagliato	FAIL

## 7.2.2 Modifica Note Appuntamenti

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre - condizioni	Input	Output Attesi	Post-condizioni Attese	Output ottenuti	Post-condizioni ottenute	Esito
1	Tutti input validi	ID appuntamento corretto, lunghezza note valida	L'ID dell' appuntamento è valido, in quanto le cifre sono 5, le note sono valide, in quanto non superano i 45 caratteri	{ID appuntamento: "16511", Note: "Appuntamento stabilito secondo delle direttive" }	Appuntamento riconosciuto e note modificabili	L'appuntamento è identificato e l'agente vendita ha modificato le note	"Note modificate con successo"	Sono soddisfatti tutti i requisiti, le note sono modificate	PASS
2	Lunghezza ID appuntamento corretta, lunghezza note non corretta	Lunghezza ID appuntamento corretta, lunghezza note > 45 [ERROR]	L'ID dell' appuntamento è valido, in quanto le cifre sono 5, le note non sono valide, in quanto superano i 45 caratteri	{ID appuntamento: "12331", Note: "Appuntamento stabilito secondo delle direttive [...]"} }	Note appuntamento non modificate poiché la lunghezza è maggiore di 45 caratteri	L'appuntamento è stato identificato, l'agente vendita riprova controllando che la lunghezza delle note non superi i 45 caratteri	"Note non modificate, il numero di caratteri supera i 45 elementi"	Non sono soddisfatti tutti i requisiti, si ha un'avvertenza riguardo la lunghezza note	FAIL
3	Lunghezza ID appuntamento non corretta, lunghezza note corretta	Lunghezza ID appuntamento > 5 [ERROR], lunghezza note < 45	L'ID dell' appuntamento non è valido, in quanto le cifre sono maggiori di 5, le note sono valide, in quanto non superano i 45 caratteri	{ ID appuntamento: "112633", Note: "Appuntamento stabilito secondo delle direttive" }	Appuntamento non riconosciuto poiché la lunghezza dell'ID è maggiore di 5 caratteri	L'appuntamento non è stato identificato e l'agente vendita riprova digitando un ID appuntamento di 5 cifre	"Note non modificate, l'ID è troppo lungo"	Non sono soddisfatti tutti i requisiti, si ha un'avvertenza riguardo l'ID lista sbagliato	FAIL
4	Lunghezza ID appuntamento non corretta, lunghezza	Lunghezza ID appuntamento < 5 [ERROR],	L'ID dell' appuntamento non è valido, in quanto le cifre	{ ID appuntamento: "1633", Note: "Appunta	Appuntamento non riconosciuto poiché la lunghezza dell'ID è	L'appuntamento non è stato identificato e	"Note non modificate, l'ID è troppo corto"	Non sono soddisfatti tutti i requisiti, si ha un'avvertenza	FAIL

	note corretta	lunghezza note < 45	sono minori di 5, le note sono valide, in quanto non superano i 45 caratteri	mento stabilito secondo delle direttive” }	minore di 5 caratteri	l’agente vendita riprova digitando un ID appuntam ento di 5 cifre		za riguardo l’ID lista sbagliato	
5	ID appuntame nto non corretto, lunghezza note < 45	ID telefonata non presente nel database [ERROR], lunghezza note < 45	L’ID dell’ appuntam ento non è valido, in quanto non è stato trovato nel database, le note sono valide, in quanto non superano i 45 caratteri	{ ID appuntam ento : “11421”, Note: “Appunta mento stabilito secondo delle direttive” }	Appuntame nto non trovato	L’appunta mento non è stato identificat o e l’agente vendita riprova digitando un ID tra quelli riconosciu ti nel database	“Note non modificate, ID non trovato”	Non sono soddisfatti tutti i requisiti, si ha un’avverten za riguardo l’ID gruppo sbagliato	FAIL