



SVILUPPO SOFTWARE DI UN CALL-CENTER

S.F.C. GROUP

ELABORATO INGEGNERIA DEL
SOFTWARE AA. 2022-2023

Balassone Francesco
Bellotti Carmine
Cecere Simone

N46005679
N46005710
N56005669

f.balassone@studenti.unina.it
ca.bellotti@studenti.unina.it
simo.cecere@studenti.unina.it

Indice

1. Specifiche Informali	3
2. Analisi e specifica dei requisiti.....	4
2.1 Sistema di gestione call center.....	4
2.2 Revisione dei requisiti.....	5
2.3 Glossario dei termini	6
2.4.1 Requisiti funzionali	6
2.5 Modellazione dei casi d'uso	8
2.5.1 Attori e Casi d'uso	9
2.5.2 Diagramma dei casi d'uso	11
2.6 Diagramma Classi	18
Diagramma delle Classi di Analisi.....	18
2.7 Diagrammi di Sequenza	20
2.8 Verifica e completezza dei requisiti.....	26
3. Stima dei costi.....	27
4. Piano di test funzionale	32
5. Progettazione.....	40
5.1 Diagramma delle Classi.....	40
6. Implementazione	45
7. Testing.....	47
7.1 Test strutturale.....	47
7.1.1 Complessità ciclomatica	47
7.1.2 Test Funzionale	50

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 e email 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. Analisi e specifica dei requisiti

2.1 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 e email** 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.

Legenda

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

2.2 Revisione dei requisiti

1. Il sistema deve consentire agli amministratori di creare liste di numeri telefonici.
2. Il sistema deve consentire agli amministratori di aggiungere numeri telefonici dalle liste.
3. Il sistema deve consentire agli amministratori di rimuovere numeri telefonici dalle liste.
4. Ogni numero telefonico è costituito da 10 cifre.
5. Di ogni centralinista si vuole memorizzare id, nome, cognome e email.
6. I centralinisti sono organizzati in gruppi.
7. Il sistema deve consentire agli amministratori di raggruppare i centralinisti.
8. La creazione di un gruppo consiste nell'inserimento di almeno un centralinista.
9. Il sistema deve consentire agli amministratori di eliminare i raggruppamenti dei centralinisti.
10. Di ogni gruppo di centralinisti si vuole memorizzare id e descrizione.
11. Il sistema deve consentire agli amministratori di assegnare ciascuna lista di numeri al più a un gruppo di centralinisti.
12. Il sistema deve consentire ai centralinisti di richiedere al sistema il prossimo numero da chiamare.
13. Il sistema deve consentire ai centralinisti di registrare l'esito della telefonata.
14. Di ogni telefonata si vuole memorizzare l'id, la data, l'ora, le note e l'esito
15. Gli esiti possono essere:
 - a. occupato,
 - b. senza risposta,
 - c. da richiamare,
 - d. non interessato,
 - e. appuntamento fissato
16. Il sistema deve consentire, nell'eventualità di "appuntamento fissato", ai centralinisti di creare un nuovo appuntamento tra uno degli agenti vendita del call center e l'utenza chiamata
17. Di ogni agente di vendita del call center si vuole memorizzare il Codice Fiscale
18. Di ogni appuntamento si vuole memorizzare l'id, data, ora e note relative all'incontro.
19. Il sistema deve consentire ad un agente di vendita di ottenere informazioni relative a tutti gli appuntamenti fissati.
20. Il sistema deve consentire ad un agente di vendita di modificare le note degli appuntamenti fissati.
21. Il sistema deve consentire ad un agente di vendita di risalire alle note di un centralinista al termine della chiamata conclusa con un appuntamento fissato.
22. Il sistema deve inoltre consentire ai centralinisti di collegare un nuovo appuntamento ad un altro nell'eventualità in cui quest'ultimo fallisca.
23. Il sistema deve implementare le politiche sulla privacy degli utenti
24. La disponibilità del sistema deve essere garantita durante tutta la fascia oraria di lavoro di un call-center.

25. Il sistema deve prevedere misure di backup e recupero in caso di interruzioni e guasti.

2.3 Glossario dei termini

Termine	Descrizione	Sinonimi
Appuntamento	Incontro/chiamata prefissato di comune accordo tra un agente di vendita e un utente	
Appuntamento fissato	Esito di una chiamata tra un centralinista e un utente.	
Appuntamento fallito	es. l'utente non era disponibile al momento dell'appuntamento con l'agente	
Utente	Generico utente che viene chiamato	Utenza chiamata
Amministratore	Personale autorizzato con accesso al sistema ed è responsabile della creazione e gestione delle liste di numeri telefonici, dei gruppi di centralinisti e dell'assegnazione delle liste ai gruppi.	
Centralinista	Operatore del call center che effettua chiamate ai numeri telefonici loro assegnati, è responsabile di registrare gli esiti delle chiamate e di creare appuntamenti con gli agenti di vendita.	
Agente di vendita	Operatore del call center che gestisce gli appuntamenti fissati con gli utenti chiamati. Accede alle informazioni sugli appuntamenti e può modificarne le note.	

2.4 Classificazione dei requisiti

2.4.1 Requisiti funzionali

ID	Requisito	Origine
RF01	Il sistema deve consentire agli amministratori di creare liste di numeri telefonici.	1
RF02	Il sistema deve consentire agli amministratori di aggiungere numeri telefonici dalle liste.	2
RF03	Il sistema deve consentire agli amministratori di rimuovere numeri telefonici	3

	dalle liste.	
RF04	Il sistema deve consentire agli amministratori di raggruppare i centralinisti.	7
RF05	Il sistema deve consentire agli amministratori di eliminare i raggruppamenti dei centralinisti.	9
RF06	Il sistema deve consentire agli amministratori di assegnare ciascuna lista di numeri al più a un gruppo di centralinisti.	11
RF07	Il sistema deve consentire ai centralinisti di richiedere al sistemail prossimo numero da chiamare.	12
RF08	Il sistema deve consentire ai centralinisti di registrare l'esito della telefonata.	13
RF09	Il sistema deve consentire, nell'eventualità di "appuntamento fissato", ai centralinisti di creare un nuovo appuntamento tra uno degli agenti vendita del call center e l'utenza chiamata	16
RF10	Il sistema deve consentire ad un agente di vendita di ottenere informazioni relative a tutti gli appuntamenti fissati.	19
RF11	Il sistema deve consentire ad un agente di vendita di modificare le note degli appuntamenti fissati.	20
RF12	Il sistema deve consentire ad un agente di vendita di risalire alle note di un centralinista al termine della chiamata conclusasi con un appuntamento fissato.	21
RF13	Il sistema deve inoltre consentire ai centralinisti di collegare un nuovo appuntamento ad un altro nell'eventualità in cui quest'ultimo fallisca.	22

2.4.2 Requisiti sui dati

ID	Requisito	Origine
RD01	Ogni lista telefonica è composta da numeri telefonici.	1,2,3
RD02	Ogni numero telefonico è costituito da 10 cifre.	4
RD03	Ogni centralinista è identificato da id, nome, cognome e email.	5
RD04	I centralinisti devono essere organizzati in gruppi.	6
RD05	Ogni gruppo di centralinisti è identificato da un ID e una descrizione.	10
RD06	Ogni telefonata è identificata da ID, data, ora, note e esito.	14
RD07	Gli esiti possono essere: occupato, senza risposta, da richiamare, non interessato, appuntamento fissato	15
RD08	Ogni agente di vendita è identificato dal codice fiscale.	17
RD09	Ogni appuntamento è identificato da ID, data, ora e note relative all'incontro.	18

2.4.3 Vincoli / Altri requisiti

ID	Requisito	Origine
V01	La creazione di un gruppo consiste nell'inserimento di almeno un centralinista.	8
RNF01	Il sistema deve implementare le politiche sulla privacy degli utenti	23
RNF02	Il sistema deve essere disponibile e accessibile durante tutta la fascia oraria di lavoro di un call-center.	24
RNF03	Il sistema deve prevedere misure di backup e recupero in caso di interruzioni o guasti.	25

2.5 Modellazione dei Casi

d'uso

2.5.1 Attori e Casi d'uso

Attori Primari

- Amministratore
- Centralinista
- Agente Vendita

Attori Secondari

- -

Casi d'uso

- UC1: Creare lista numeri telefonici
- UC2: Aggiungere numero telefonico alla lista
- UC3: Rimuovere numero telefonico dalla lista
- UC4: Assegnare lista numeri telefonici a gruppo di centralinisti
- UC5: Rimuovere gruppo di centralinisti
- UC6: Creare gruppo di centralinisti
- UC7: Visualizzare note chiamate
- UC8: Visualizzare dettagli appuntamento
- UC9: Registrare esito chiamata
- UC10: Collegare appuntamento al precedente

Casi d'uso di inclusione

- UC11: Trovare lista numeri telefonici
- UC12: Trovare gruppo centralinisti
- UC13: Richiedere numero da chiamare

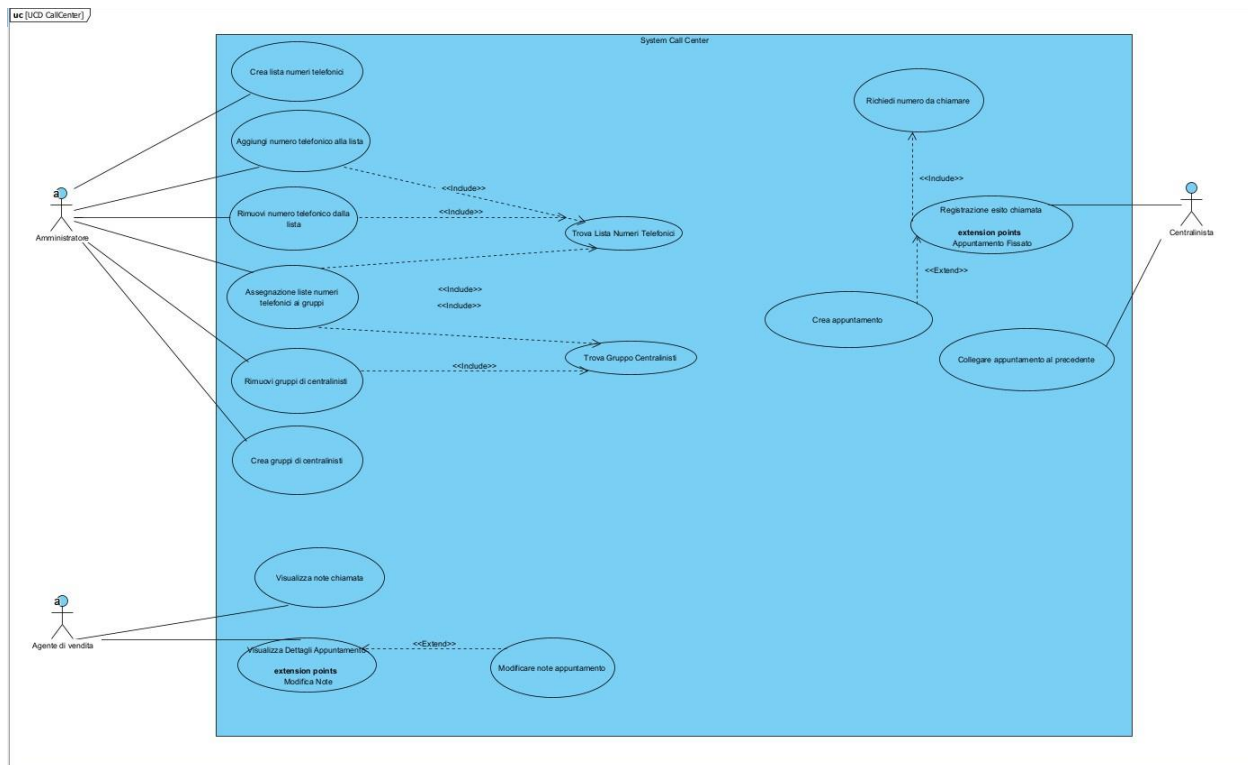
Casi d'uso di estensione

- UC14: Modificare note appuntamento
- UC15: Creare appuntamento

Caso d'Uso	Attori Primari	Attori Secondari	Incl. / Ext.	Requisiti Corrispondenti
UC1: Creare lista numeri telefonici	Amministratore	-	-	RF01
UC2: Aggiungere numero telefonico alla lista	Amministratore	-	Include UC10	RF02
UC3: Rimuovere numero telefonico dalla lista	Amministratore	-	Include UC10	RF03
UC4: Assegnare lista di numeri a un gruppo di centralinisti	Amministratore	-	Include UC10, Include UC11	RF06
UC5: Rimozione gruppo	Amministratore	-	Include UC11	RF05
UC6: Creare gruppo di centralinisti	Amministratore	-	-	RF04
UC7: Visualizzare note chiamata	Agenti di vendita	-	-	RF13
UC8: Visualizzare dettagli appuntamento	Agente di Vendita	-	-	RF11
UC9: Registrare esito telefonata	Centralinista	-	Include UC13	RF08
UC10: Collegare al appuntamento precedente	Centralinista	-	-	RF09
UC11: Trova lista numeri telefonici	Amministratore	-	-	RF02, RF03
UC12: Trova gruppo centralinisti	Amministratore	-	-	RF05
UC13: Richiedere numero da chiamare	Centralinista	-	-	RF07

UC14: Modificare note dell'appuntam ento	Agenti di Vendita	-	Estende UC8	RF12
UC15: Creare appuntame nto	Centralinista	-	Estende UC9	RF09

2.5.2 Diagramma dei casi d'uso



2.5.3 Scenari

Nome Caso d'uso	UC1: Creare lista numeri telefonici
Attore Primario	Amministratore
Attore Secondario	-
Descrizione	L'amministratore crea una nuova lista di numeri telefonici inserendo i numeri desiderati.
Pre-condizioni	Il sistema deve risultare accessibile.
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. L'amministratore accede al sistema. 2. L'amministratore crea una lista di numeri telefonici <ol style="list-style-type: none"> 2.1 Il sistema crea una lista vuota 3. Il sistema conferma l'avvenuta creazione, mostrando l'ID della lista appena creata.
Post-condizioni	La lista numeri telefonici è stata aggiunta al database di sistema.
Casi d'uso correlati	-
Sequenza di eventi alternativi	2.1

Nome Caso d'uso	UC2: Aggiungi numero telefonico alla lista
Attore Primario	Amministratore
Attore Secondario	-
Descrizione	L'amministratore aggiunge un numero telefonico ad una lista esistente.
Pre-condizioni	Il sistema possiede almeno una lista di numeri telefonici al suo interno.
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. L'amministratore accede al sistema. 2. L'amministratore aggiunge un numero, composto da 10 cifre, ad una lista selezionata. 3. Include (TrovaLista). 4. Il sistema cerca la lista. <ol style="list-style-type: none"> 4.1 Se la lista non esiste, il sistema non consente l'inserimento del numero. 4.2 Se il numero aggiunto non è di 10 cifre, il sistema riporta un messaggio di errore. 4.3 Se il numero di telefono è già presente nella lista selezionata, il sistema invia un messaggio di avviso. 5. Il sistema conferma l'aggiunta del numero alla lista.
Post-condizioni	La lista di numeri telefonici scelta presenta un numero telefonico in più.
Casi d'uso correlati	UC11
Sequenza di eventi alternativi	4.1, 4.2, 4.3

Nome Caso d'uso	UC3: Rimuovere numero telefonico dalla lista
Attore Primario	Amministratore
Attore Secondario	-
Descrizione	L'amministratore elimina un numero di telefono da una lista di numeri telefonici.
Pre-condizioni	La lista scelta per la rimozione del numero di telefono deve risultare non vuota.

Sequenza di eventi principale	<ol style="list-style-type: none"> 1. L'amministratore accede al sistema. 2. L'amministratore elimina un numero telefonico da una lista selezionata. 3. Include (TrovaLista). 4. Il sistema cerca la lista. 4.1 Se la lista non esiste, il sistema non consente la scelta del numero. 4.2 Se il numero di telefono non è presente nella lista, il sistema invia un messaggio di errore. 4. Il sistema conferma la rimozione del numero telefonico dalla lista.
Post-condizioni	La lista di numeri telefonici scelta presenta un numero telefonico in meno.
Casi d'uso correlati	UC11
Sequenza di eventi alternativi	4.1, 4.2

Nome Caso d'uso	UC4: Assegnare lista di numeri a un gruppo di centralinisti
Attore Primario	Amministratore
Attore Secondario	-
Descrizione	L'amministratore assegna una lista di numeri telefonici a un gruppo di centralinisti presente nel sistema.
Pre-condizioni	Nel sistema è presente almeno una lista di numeri telefonici.
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. L'amministratore accede al sistema. 2. L'amministratore seleziona una lista di numeri telefonici e un gruppo di centralinisti. 3. Include (TrovaLista). 4. Il sistema cerca la lista. 4.1 Se la lista non esiste, il sistema non consente l'inserimento del gruppo. 5. Include (TrovaGruppo). 6. Il sistema cerca il gruppo. 6.1 Se il gruppo di centralinisti non esiste, il sistema restituisce un messaggio di errore. 7. Il sistema assegna la lista selezionata al gruppo selezionato. 8. Il sistema conferma l'avvenuta assegnazione.
Post-condizioni	Il numero di liste telefoniche assegnabili è diminuito.
Casi d'uso correlati	UC11, UC12
Sequenza di eventi alternativi	4.1, 6.1

Nome Caso d'uso	UC5: Rimozione gruppo
Attore Primario	Amministratore
Attore Secondario	-
Descrizione	Un amministratore rimuove un gruppo di centralinisti dal sistema.
Pre-condizioni	Il sistema deve risultare accessibile.

Sequenza di eventi principale	<ol style="list-style-type: none"> 1. L'amministratore accede al sistema. 2. L'amministratore seleziona il gruppo di centralinisti darimuovere. 3. Include (TrovaGruppo). 4. Il sistema cerca il gruppo. 4.1 Se non vi fosse presente alcun gruppo di centralinisti all'interno del sistema, quest'ultimo invia un messaggio di avviso. 5. Il sistema libera i centralinisti di tale gruppo. 6. Il sistema conferma l'avvenuta eliminazione.
Post-condizioni	Il gruppo di centralinisti è stato rimosso dal database del sistema.
Casi d'uso correlati	UC12
Sequenza di eventi alternativi	4.1

Nome Caso d'uso	UC6: Creare gruppo di centralinisti
Attore Primario	Amministratore
Attore Secondario	-
Descrizione	Un amministratore crea un nuovo gruppo di centralinisti nel sistema.
Pre-condizioni	Il sistema deve risultare accessibile.
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. L'amministratore accede al sistema. 2. L'amministratore inserisce la descrizione del nuovo gruppo. 3. Il sistema conferma l'avvenuta creazione, restituendo l'ID del nuovo gruppo. 4. Il sistema richiede l'inserimento di almeno un centralinista al gruppo. 5. L'amministratore assegna i centralinisti. 6. Il sistema conferma l'avvenuta assegnazione.
Post-condizioni	Il gruppo di centralinisti è stato aggiunto al database di sistema.
Casi d'uso correlati	-
Sequenza di eventi alternativi	-

Nome Caso d'uso	UC7: Visualizzare note chiamate
Attore Primario	Agente di vendita
Attore Secondario	-
Descrizione	L'agente di vendita consulta le note legate ad una chiamata tra un centralinista e un cliente.
Pre-condizioni	L'agente di vendita possiede i privilegi necessari.
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. L'agente di vendita effettua l'autenticazione al sistema. 2. L'agente di vendita seleziona una chiamata presente nel database di sistema 2.1 Se l'input risultasse errato e non vi fosse riscontro nel database di sistema, quest'ultimo invia un messaggio di errore che invita l'agente di vendita alla correzione dell'input. 2.2 Se la telefonata non è relativa ad un appuntamento assegnato all'agente, il sistema

	restituisce un messaggio d'errore. 3. L'agente di vendita richiede di visualizzare le note relative allachiamata selezionata. 4. Il sistema permette la visualizzazione delle note associate alla chiamata selezionata.
Post-condizioni	-
Casi d'uso correlati	-
Sequenza di eventi alternativi	2.1, 2.2

Nome Caso d'uso	UC8: Visualizzare dettagli appuntamento
Attore Primario	Agente di vendita
Attore Secondario	-
Descrizione	L'agente di vendita richiede ed ottiene dal sistemainformazioni riguardati gli appuntamenti fissati.
Pre-condizioni	L'agente di vendita possiede i privilegi necessari.
Sequenza di eventi principale	1. L'agente di vendita effettua l'autenticazione al sistema. 2. L'agente di vendita richiede la visualizzazione delle informazioni relative agli appuntamenti fissati. 3. Il sistema fornisce l'elenco degli appuntamenti fissati con le relative informazioni.punto di estensione: modifica note.
Post-condizioni	UC14
Casi d'uso correlati	-
Sequenza di eventi alternativi	-

Nome Caso d'uso	UC9: Registrare esito della telefonata
Attore Primario	Centralinista
Attore Secondario	-
Descrizione	Dopo aver effettuato una chiamata, il centralinista registra le informazioni della telefonata nel databasedi sistema.
Pre-condizioni	Il centralinista ha effettuato una telefonata.
Sequenza di eventi principale	1. Include (Richiedere prossimo numero) 2. Dopo aver effettuato una telefonata, il centralinista ne registra ID, data, ora, note e esito. Punto di estensione: Crea Appuntamento 3. Il sistema conferma la registrazione della telefonata.
Post-condizioni	Le informazioni relative alla chiamata sono stateaggiunte al sistema.
Casi d'uso correlati	UC13, UC15
Sequenza di eventi alternativi	-

Nome Caso d'uso	UC10: Collegare appuntamento al precedente
Attore Primario	Centralinista
Attore Secondario	-

Descrizione	Un centralinista collega un nuovo appuntamento a un appuntamento precedente fallito.
Pre-condizioni	Il centralinista possiede i privilegi necessari a collegare gli appuntamenti.
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il centralinista effettua l'identificazione nel sistema. 2. Il centralinista identifica un appuntamento precedente fallito relativo ad un utente. 3. Il centralinista crea un nuovo appuntamento per l'utente in questione e lo collega al suo appuntamento precedente tramite id e id dell'appuntamento precedentemente fallito. 4. Il sistema registra il collegamento effettuato.
Post-condizioni	Il nuovo appuntamento inserito riferisce l'appuntamento precedente fallito dell'utente.
Casi d'uso correlati	UC15
Sequenza di eventi alternativi	-

Nome Caso d'uso	UC11: Trova Lista Numeri Telefonici
Attore Primario	Amministratore
Attore Secondario	-
Descrizione	L'amministratore ricerca una determinata lista presente all'interno del database di sistema.
Pre-condizioni	La lista ricercata deve essere presente nel database di sistema.
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. L'amministratore inserisce l'id della lista richiesta 2. Viene controllata la presenza della lista nel database, e nel caso vi fosse viene visualizzato un messaggio di conferma 2.1 Nel caso non vi fosse viene visualizzato un messaggio di errore 3. Viene riportata la lista cercata
Post-condizioni	-
Casi d'uso correlati	-
Sequenza di eventi alternativi	2.1

Nome Caso d'uso	UC12: Trova Gruppo Centralinisti
Attore Primario	Amministratore
Attore Secondario	-
Descrizione	L'amministratore ricerca un determinato gruppo all'interno del database di sistema.
Pre-condizioni	Il gruppo ricercato deve essere presente nel database di sistema.
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. L'amministratore inserisce l'id del gruppo richiesto 2. Viene controllata la presenza del gruppo nel database, e nel caso vi fosse viene visualizzato un messaggio di conferma 2.1 Nel caso non vi fosse viene visualizzato un messaggio di errore 3. Viene riportato il gruppo cercato

Post-condizioni	-
Casi d'uso correlati	-
Sequenza di eventi alternativi	2.1

Nome Caso d'uso	UC13: Richiedere il prossimo numero da chiamare
Attore Primario	Centralinista
Attore Secondario	-
Descrizione	Un centralinista richiede il numero successivo da chiamare dalla lista di numeri assegnatagli.
Pre-condizioni	Il centralinista appartiene ad un gruppo al quale è stata assegnata una lista di numeri telefonici.
Sequenza di eventi principale	1. Il centralinista richiede un numero dalla lista assegnatagli, a seconda del gruppo di appartenenza. 1.1 Se non è presente alcun numero di telefono nella lista, il sistema restituisce un messaggio di avviso. 2. Il sistema fornisce il numero di telefono.
Post-condizioni	-
Casi d'uso correlati	-
Sequenza di eventi alternativi	1.1

Nome Caso d'uso	UC14: Modificare note dell'appuntamento
Attore Primario	Agente di vendita
Attore Secondario	-
Descrizione	L'agente di vendita modifica le note relative ad un appuntamento fissato.
Pre-condizioni	L'agente di vendita deve avere accesso all'appuntamento di cui intende modificare le note.
Sequenza di eventi principale	1. L'agente di vendita effettua l'identificazione nel sistema. 2. L'agente di vendita seleziona l'appuntamento. 2.1 Se l'appuntamento non viene trovato, il sistema invia un messaggio di errore. 3. L'agente di vendita modifica le note dell'appuntamento. 4. Il sistema salva le modifiche effettuate.
Post-condizioni	-
Casi d'uso correlati	UC8
Sequenza di eventi alternativi	2.1

Nome Caso d'uso	UC15: Creare appuntamento
Attore Primario	Centralinista
Attore Secondario	-
Descrizione	Un centralinista crea un nuovo appuntamento tra un agente vendita e l'utente chiamato.
Pre-condizioni	Il centralinista possiede i privilegi necessari alla creazione di un

	appuntamento.
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il centralinista effettua l'identificazione nel sistema. 2. Il centralinista seleziona una chiamata il cui esito è "appuntamento fissato". <ol style="list-style-type: none"> 2.1 Se non vi fosse presente alcuna chiamata dall'esito "appuntamento fissato" all'interno del sistema, quest'ultimo invia un messaggio di avviso. 3. Il sistema crea un nuovo appuntamento inserendo ID, data, ora e note relative fornite dal centralinista. punto di estensione: appuntamento fallito 4. Il sistema registra l'appuntamento nel database del sistema.
Post-condizioni	L'appuntamento è stato aggiunto nel database del sistema.
Casi d'uso correlati	UC9
Sequenza di eventi alternativi	2.1

2.6 Diagramma Classi

Diagramma delle Classi di Analisi

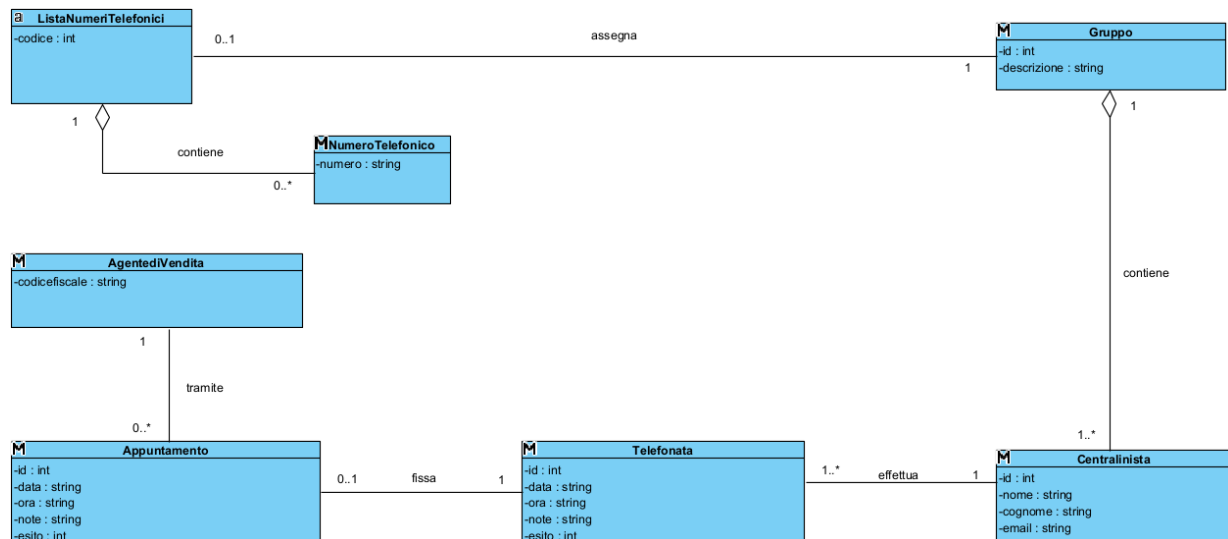
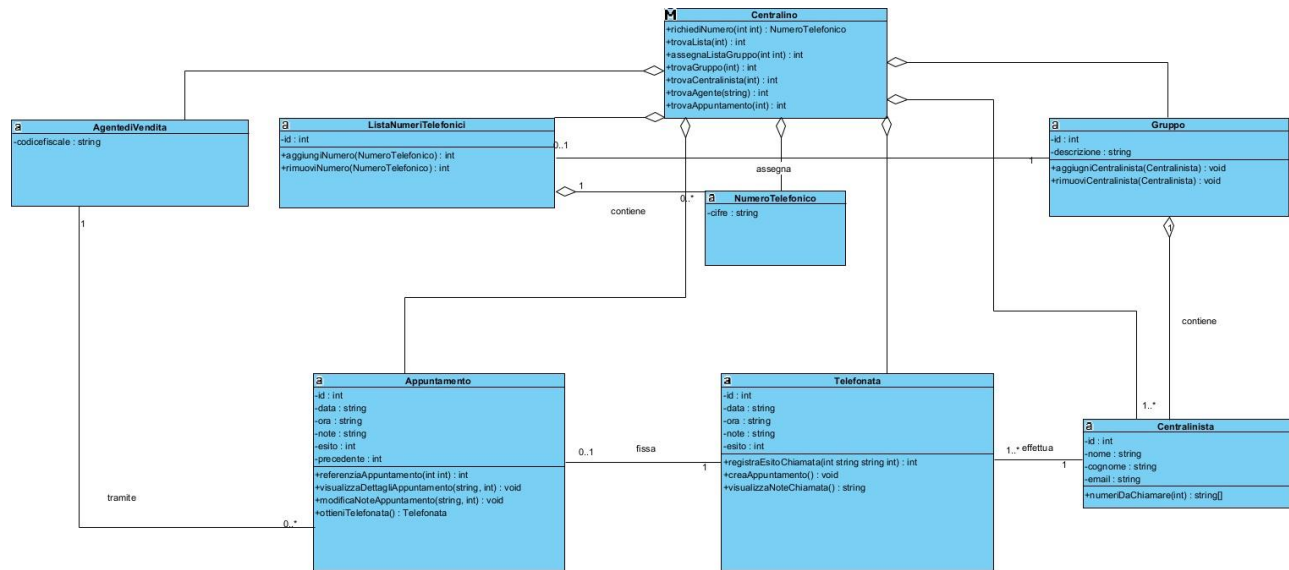
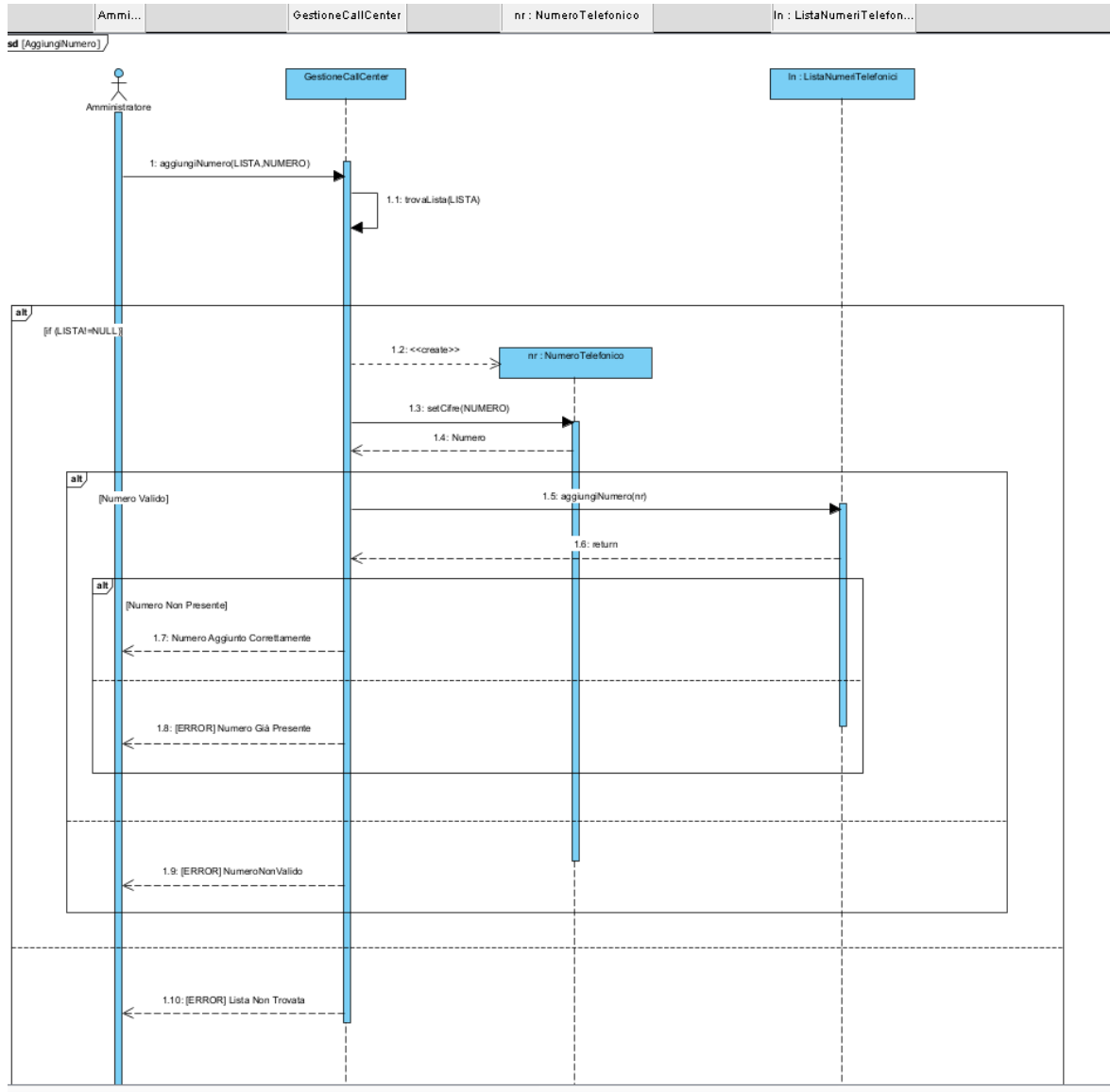


Diagramma delle Classi Raffinato

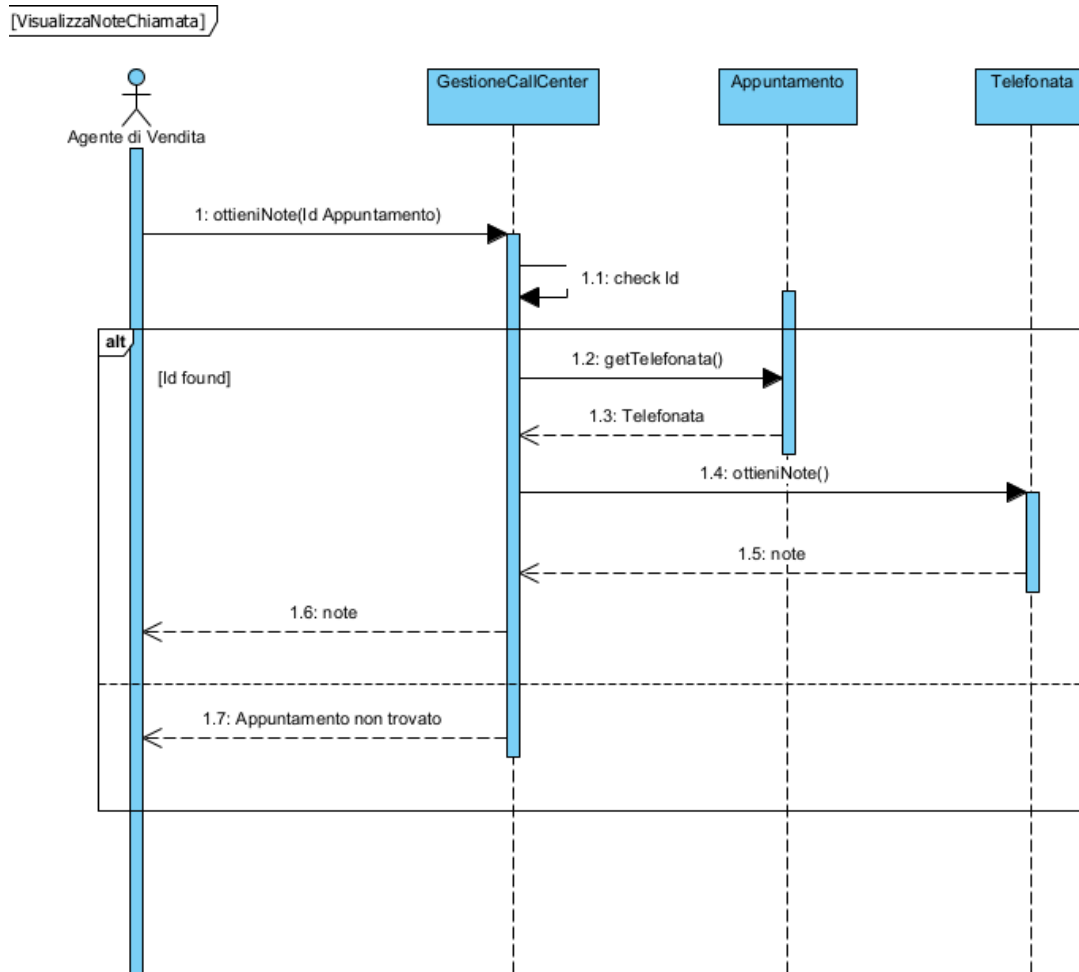


2.7 Diagrammi di Sequenza

- Caso d'uso: *AggiungiNumero(UC2)*

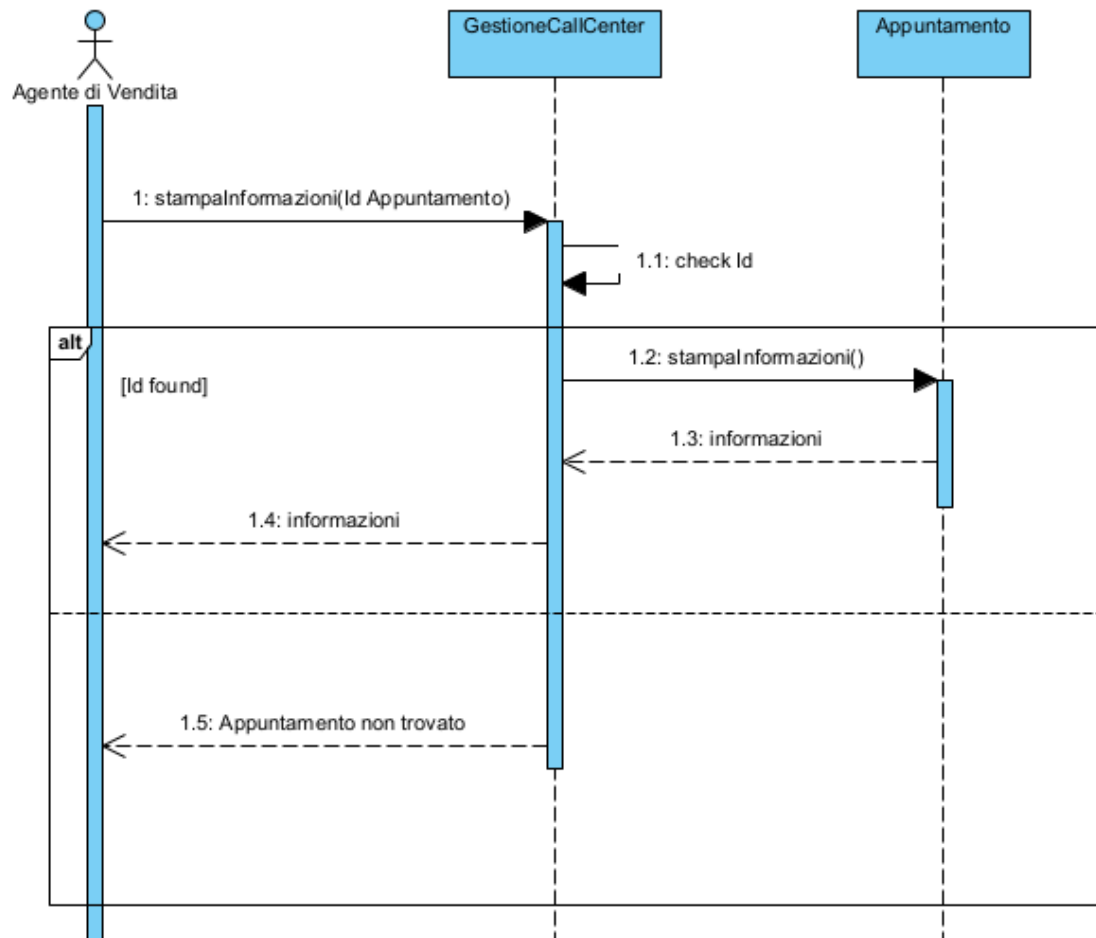


- **Caso d'uso:** *Visualizza Note Chiamata (UC7)*

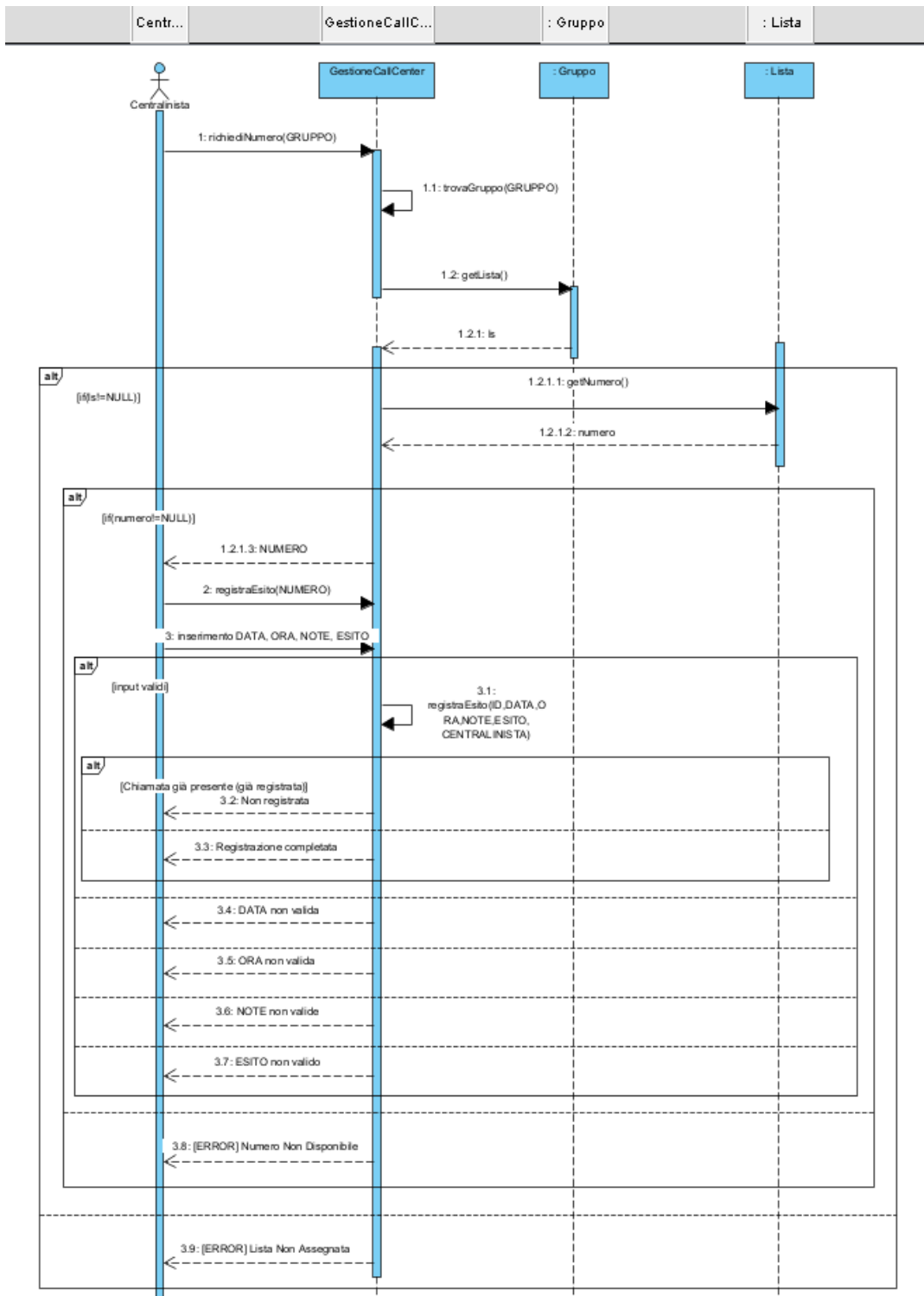


- **Caso d'uso:** *Visualizzare Dettagli Appuntamento (UC8)*

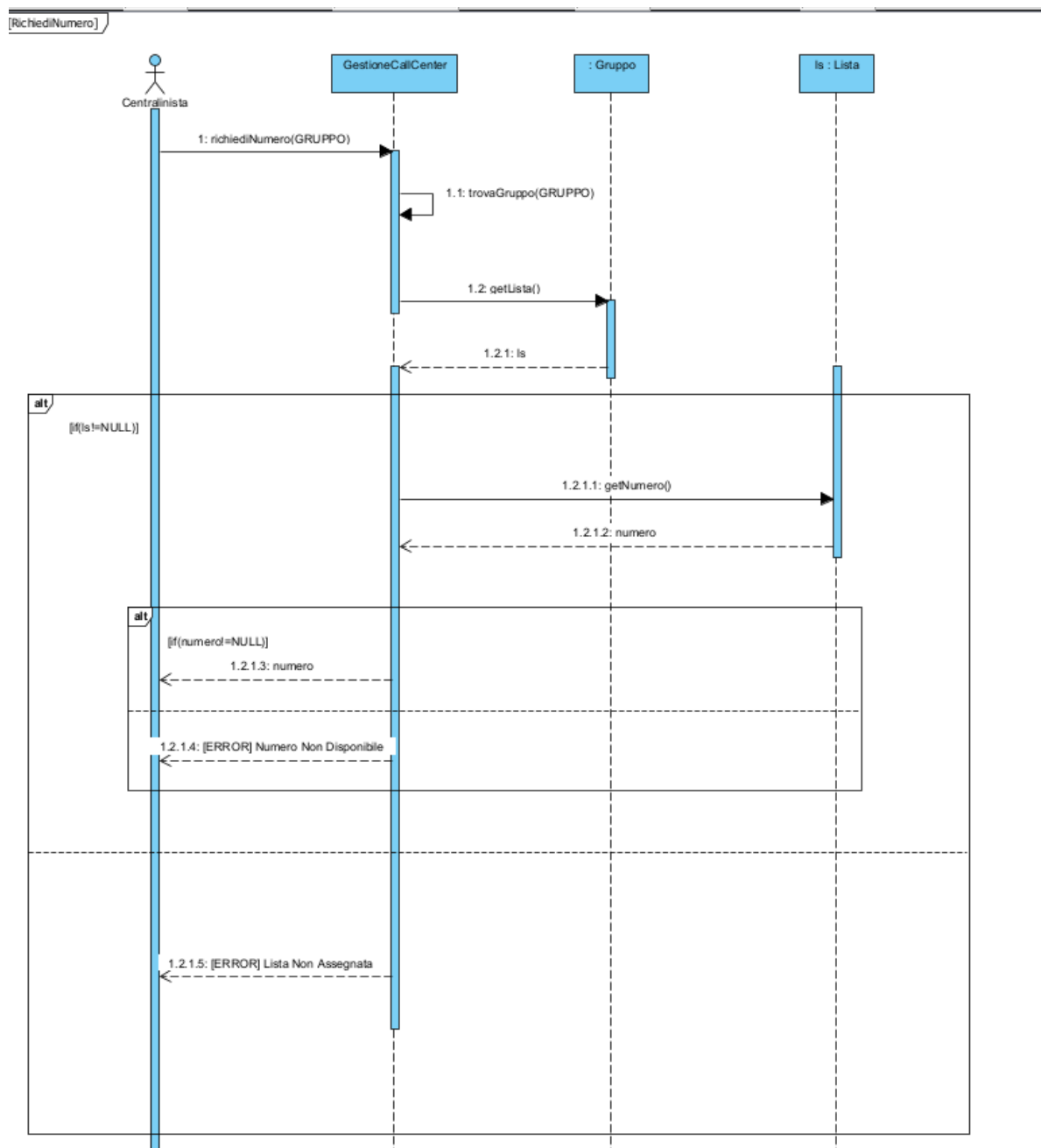
[VisualizzaDettagliAppuntamento]



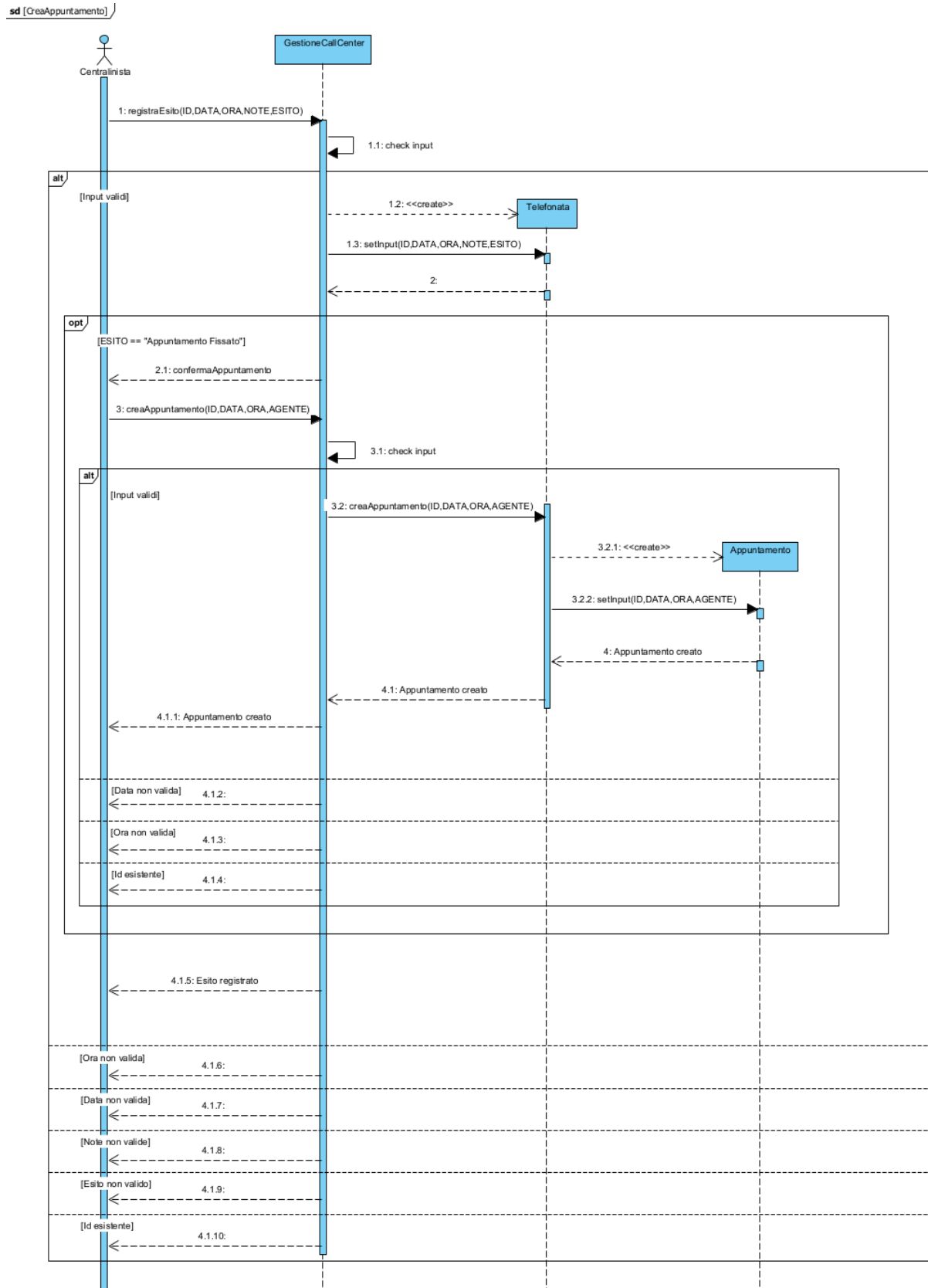
- **Caso d'uso: Registra Esito (UC9)**



- **Caso d'uso: Richiedi Numero (UC13)**



- **Caso d'uso: Creare Appuntamento (UC15)**



2.8 Verifica e completezza dei requisiti

Legenda: UCD= Use Case Diagram, UC= *Class Diagram*, SD= Sequence Diagram

- **RF01** è modellato nell'UCD con l'attore **Amministratore** e con il caso d'uso UC1
- **RF02** è modellato nell'UCD con l'attore **Amministratore** e con il caso d'uso UC2
- **RF03** è modellato nell'UCD con l'attore **Amministratore** e con il caso d'uso UC3
- **RF04** è modellato nell'UCD con l'attore **Amministratore** e con il caso d'uso UC6
- **RF05** è modellato nell'UCD con l'attore **Amministratore** e con il caso d'uso UC5
- **RF06** è modellato nell'UCD con l'attore **Amministratore** e con il caso d'uso UC4
- **RF07** è modellato nell'UCD con l'attore **Centralinista** e con il caso d'uso UC12
- **RF08** è modellato nell'UCD con l'attore **Centralinista** e con il caso d'uso UC9
- **RF09** è modellato nell'UCD con l'attore **Centralinista** e con il caso d'uso UC14
- **RF10** è modellato nell'UCD con l'attore **Agente di Vendita** e con il caso d'uso UC8
- **RF11** è modellato nell'UCD con l'attore **Agente di Vendita** e con il caso d'uso UC13
- **RF12** è modellato nell'UCD con l'attore **Agente di Vendita** e con il caso d'uso UC7
- **RF13** è modellato nell'UCD con l'attore **Centralinista** e con il caso d'uso UC15
- **RD01** è modellato nel CD con la classe **ListaNumeriTelefonici**
- **RD02** è modellato nel CD con la classe **NumeroTelefonico**
- **RD03** è modellato nel CD con la classe **Centralinista**
- **RD04, RD05** sono modellati nel CD con la classe **Gruppo**
- **RD06, RD07** è modellato nel CD con la classe **Telefonata**
- **RD08** è modellato nel CD con la classe **AgentediVendita**
- **RD09** è modellato nel CD con la classe **Appuntamento**

3. Stima dei costi

- Tabella di riferimento per le complessità di dati e transazioni

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

- Tabella elenco dei fattori correttivi (valore da 0 a 5)

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

Aggiungi numero telefonico alla lista

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	2		4	6	10
NEIF	0				
NEI	2	3			6
NEO	0				
NEQ	1		7		7

NILF: Numeri telefonico creato dal sistema [1 medio], Lista di numeri presente nel sistema [1 complesso]

NEI: Numero fornito dall'amministratore [1 semplice], Id lista fornito dall'amministratore [1 semplice]

NEQ: Interrogazioni sulla tabella ListaNumeriTelefonici [1 medio]

UFP = 23
LLOC/FP = 1.219

FATTORI CORRETTIVI	
COMUNICAZIONE DATI	3
DISTRIBUZIONE ELABORAZIONE	3
PRESTAZIONI	3
UTILIZZO INTENSIVO CONFIGURAZIONE	1
FREQUENZA DELLE TRANSAZIONI	3
INSERIMENTO DATI INTERATTIVO	5
EFFICIENZA PER L'UTENTE FINALE	5
AGGIORNAMENTO INTERATTIVO	5
COMPLESSITA' ELABORATIVA	1
RIUSABILITA'	1
FACILITA' INSTALLAZIONE	1
FACILITA' GESTIONE OPERATIVA	5
MOLTEPLICITA' DI SITI	1
FACILITA' DI MODIFICA	5
	42

FP = 24,61
JAVA = 1.304

1. Comunicazione dei dati: Media. È necessario che il sistema consenta l'aggiunta del numero telefonico alla lista in modo che sia disponibile per gli amministratori.
2. Distribuzione elaborazione: Media. Il sistema dovrebbe gestire l'aggiunta del numero telefonico in modo da distribuire l'elaborazione in modo efficiente tra i componenti del sistema.
3. Prestazioni: Media. Il sistema dovrebbe garantire tempi di risposta accettabili durante l'aggiunta del numero telefonico e l'accesso alla lista dei numeri telefonici per gli amministratori.
4. Utilizzo intensivo configurazione: Bassa. L'aggiunta di un numero telefonico alla lista di numeri telefonici di solito non richiede una configurazione intensiva.
5. Frequenza delle transazioni: Media. Dipende dalla frequenza con cui vengono aggiunti nuovi numeri telefonici alla lista.
6. Inserimento dati interattivo: Alta. Gli amministratori dovrebbero poter inserire i numeri telefonici in modo interattivo nel sistema.
7. Efficienza per l'utente finale: Alta. Il sistema dovrebbe essere progettato per semplificare l'aggiunta dei numeri telefonici per gli amministratori.
8. Aggiornamento interattivo: Alta. Dovrebbe essere possibile aggiornare la lista dei numeri telefonici in tempo reale in risposta ai cambiamenti.
9. Complessità elaborativa: Bassa. L'aggiunta di un numero telefonico alla lista di numeri telefonici è generalmente un'operazione semplice.
10. Riusabilità: Bassa. La funzione è utile per l'aggiunta di numeri telefonici, pertanto la sua riusabilità è limitata.
11. Facilità installazione: Bassa. L'aggiunta di un numero telefonico richiede solitamente solo una modifica del sistema esistente, quindi l'installazione è relativamente semplice.

12. Facilità gestione operativa: Alta. Il sistema dovrebbe consentire una gestione agevole della lista dei numeri telefonici per i centralinisti.
13. Molteplicità di siti: Bassa. Se il sistema del call center è centralizzato, la molteplicità dei siti non influisce significativamente sull'aggiunta di un numero telefonico alla lista.
14. Facilità di modifica: Alta. Il sistema dovrebbe essere progettato per consentire modifiche facili e rapide alla lista dei numeri telefonici.

Visualizzare dettagli appuntamento

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	1		4		4
NEIF	0				
NEI	1	3			3
NEO	1		10		10
NEQ	1		7		7

NILF: Appuntamenti presenti nel sistema. NEI: Id fornito dall'amministratore NEO: Informazioni appuntamento ricevute dall'agente di vendita.

NEQ: Interrogazione della tabella Appuntamenti.

UFP = 24
LLOC/FP =

FATTORI CORRETTIVI	
COMUNICAZIONE DATI	5
DISTRIBUZIONE ELABORAZIONE	3
PRESTAZIONI	5
UTILIZZO INTENSIVO CONFIGURAZIONE	1
FREQUENZA DELLE TRANSAZIONI	3
INSERIMENTO DATI INTERATTIVO	1
EFFICIENZA PER L'UTENTE FINALE	5
AGGIORNAMENTO INTERATTIVO	5
COMPLESSITA' ELABORATIVA	1
RIUSABILITA'	3
FACILITA' INSTALLAZIONE	3
FACILITA' GESTIONE OPERATIVA	5
MOLTEPLICITA' DI SITI	1
FACILITA' DI MODIFICA	5
	46

FP = 24,61
JAVA = 1.304

1. Comunicazione dei dati: Alta. È essenziale che il sistema consenta la comunicazione dei dettagli degli appuntamenti.
2. Distribuzione elaborazione: Media. Il sistema dovrebbe gestire la distribuzione dei dettagli degli appuntamenti in modo efficiente tra i vari componenti.

3. Prestazioni: Alta. Il sistema dovrebbe garantire tempi di risposta rapidi nella visualizzazione dei dettagli degli appuntamenti.
4. Utilizzo intensivo configurazione: Bassa. La visualizzazione dei dettagli degli appuntamenti non richiede solitamente una configurazione intensiva.
5. Frequenza delle transazioni: Media. Dipende dalla frequenza con cui vengono fissati nuovi appuntamenti e dalla necessità di visualizzare i dettagli degli appuntamenti correnti.
6. Inserimento dati interattivo: Bassa. Gli agenti di vendita non inseriscono manualmente i dettagli degli appuntamenti, ma visualizzano le informazioni esistenti.
7. Efficienza per l'utente finale: Alta. Il sistema dovrebbe essere progettato per facilitare l'accesso e la visualizzazione dei dettagli degli appuntamenti per gli agenti di vendita.
8. Aggiornamento interattivo: Alta. Il sistema dovrebbe consentire l'aggiornamento in tempo reale dei dettagli degli appuntamenti quando necessario (nel nostro caso riguarda solo le note).
9. Complessità elaborativa: Bassa. La visualizzazione dei dettagli degli appuntamenti è di solito un'operazione relativamente semplice.
10. Riutilizzabilità: Media. La funzione può essere riutilizzata per la visualizzazione di appuntamenti anche in altre realtà di interesse.
11. Facilità installazione: Media. La funzione richiede integrazione con il sistema esistente del call center, quindi l'installazione può richiedere un po' di lavoro.
12. Facilità gestione operativa: Alta. Il sistema dovrebbe consentire una gestione agevole dei dettagli degli appuntamenti per gli agenti di vendita.
13. Molteplicità di siti: Bassa. Se il sistema del call center è centralizzato, la molteplicità dei siti non influisce significativamente sulla visualizzazione dei dettagli degli appuntamenti.
14. Facilità di modifica: Alta. Il sistema dovrebbe consentire modifiche facili e rapide ai dettagli degli appuntamenti quando necessario (note appuntamento).

Assegnare lista di numeri a un gruppo di centralinisti

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	2		4		8
NEIF	0				
NEI	2	3			6
NEO	0				
NEQ	2		7		14

NILF: Lista di numeri telefonici e Gruppi di centralinisti presenti nel sistema. NEI: Id gruppo e Id lista forniti dall'amministratore.

NEQ: Interrogazioni sulle tabelle ListaNumeriTelefonici [1 medio] e Gruppo [1 medio].

UFP = 28
LLOC/FP =

FATTORI CORRETTIVI	
COMUNICAZIONE DATI	5
DISTRIBUZIONE ELABORAZIONE	3
PRESTAZIONI	5
UTILIZZO INTENSIVO CONFIGURAZIONE	3
FREQUENZA DELLE TRANSAZIONI	3
INSERIMENTO DATI INTERATTIVO	3
EFFICIENZA PER L'UTENTE FINALE	5
AGGIORNAMENTO INTERATTIVO	1
COMPLESSITA' ELABORATIVA	3
RIUSABILITA'	1
FACILITA' INSTALLAZIONE	3
FACILITA' GESTIONE OPERATIVA	5
MOLTEPLICITA' DI SITI	1
FACILITA' DI MODIFICA	3

FP = 30,52

JAVA = 1.617

1. Comunicazione dei dati: Alta. È fondamentale che il sistema consenta l'assegnazione delle liste di numeri telefonici ai gruppi dicentralinisti
2. Distribuzione elaborazione: Media. Il sistema dovrebbe gestire in modo efficiente la distribuzione delle liste di numeri telefonici tra i gruppi di centralinisti.
3. Prestazioni: Alta. Il sistema dovrebbe garantire tempi di risposta rapidi nell'assegnazione delle liste di numeri telefonici ai gruppi di centralinisti.
4. Utilizzo intensivo configurazione: Media. L'assegnazione delle liste di numeri telefonici richiede una configurazione iniziale per definire i gruppi e le relative assegnazioni.
5. Frequenza delle transazioni: Media. Dipende dalla frequenza con cui avvengono assegnazioni di liste di numeri telefonici ai gruppi di centralinisti.
6. Inserimento dati interattivo: Media. Potrebbe essere necessario un inserimento interattivo dei gruppi e delle assegnazioni delle liste di numeri telefonici nel sistema.
7. Efficienza per l'utente finale: Alta. Il sistema dovrebbe essere progettato per semplificare l'assegnazione delle liste di numeri telefonici ai gruppi di centralinisti e rendere l'utilizzo efficiente.
8. Aggiornamento interattivo: Bassa. Il sistema non prevede l'aggiornamento delle assegnazioni delle liste di numeri telefonici ai gruppi di centralinisti.
9. Complessità elaborativa: Media. L'assegnazione delle liste di numeri telefonici richiede una gestione di molteplici entità ambo ilati.
10. Riusabilità: Bassa. La funzione può essere scarsamente riutilizzata in altri contesti applicativi.
11. Facilità installazione: Media. L'assegnazione delle liste di numeri telefonici richiede un'integrazione con il sistema esistente del call center.
12. Facilità gestione operativa: Alta. Il sistema dovrebbe consentire una gestione agevole delle assegnazioni delle liste di numeri telefonici ai gruppi di centralinisti.

13. Molteplicità di siti: Bassa. Se il call center ha più siti, l'assegnazione delle liste di numeri telefonici richiede una gestione coerente tra i diversi siti, ma nel nostro caso si tratta di un sistema centralizzato.
14. Facilità di modifica: Media. Il sistema non prevede la riassegnazione delle liste di numeri telefonici ad altri gruppi di centralinisti, a meno che questi non vengano rimossi.

4. Piano di test funzionale

PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* *“RegistraEsitoTelefonata”*

DATA	ORA	NOTE	ESITO
- Data con formato valido (gg/mm/aaaa) - Data con formato non valido [ERROR]	- Ora con formato valido [hh:mm] - Ora con formato non valido [ERROR]	- Stringa di caratteri di lunghezza ≤ 1000 - Stringa di caratteri di lunghezza > 1000 [ERROR]	- Valore >0 e ≤ 5 : - 1: occupato; - 2: senza risposta; - 3: da richiamare; - 4: non interessato; - 5: appuntamento fissato; - Valore ≤ 0 [ERROR] - Valore >5 [ERROR]

Il numero di test da effettuarsi senza particolari vincoli

è $2 \times 2 \times 2 \times 3 = 24$ Introduciamo i vincoli [ERROR]

Il numero di test da eseguire per testare singolarmente i vincoli è 5 (1 per Data, 1 per Ora, 1 per Note, 2 per Esito). Il numero di test risultante è: $(1 \times 1 \times 1 \times 1) + 5 = 6$

- **Test Suite**

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti input validi	Data valida Ora valida Stringa di caratteri di lunghezza ≤ 1000 Valore > 0 e ≤ 5	Nessuna	{Data: "11/12/2022", Ora: "15:30", Note: "Telefonata Effettuata", Esito: 2}	Chiamata registrata con successo!	Sono salvati nel sistema i dettagli della telefonata.

2	Data con formato non valido.	Formato data non valido [ERROR] Ora valida Stringa di caratteri di lunghezza ≤ 1000 Valore > 0 e ≤ 5	Nessuna	{Data: "11/13/2022", Ora: "15:30", Note: "Telefonata Effettuata", Esito: 2}	Messaggi o di errore - Data non valida!	
3	Ora con formato non valido.	Data valida Formato ora non valido [ERROR] Stringa di caratteri di lunghezza ≤ 1000 Valore > 0 e ≤ 5	Nessuna	{Data: "11/12/2022", Ora: "25:30", Note: "Telefonata Effettuata", Esito: 2}	Messaggio di errore - Ora non valida!	

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
4	Stringa di caratteri di lunghezza > 1000.	Data valida Oravalida Stringa di caratteri di lunghezza > 1000 [ERROR] Valore > 0 e ≤ 5	Nessuna	{Data: "11/12/2022", Ora: "12:30", Note: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam tristique risus nec ipsum cursus, sit amet tincidunt ligula facilisis. Sed fringilla sem ac est congue bibendum. Sed dapibus justo auctor, consequat felis nec, dignissim velit. Sed tincidunt ultrices diam. Mauris eu enim non justo lobortis volutpat at vitae lectus. Cras eu orci at sapien faucibus feugiat. Nulla vulputate eleifend mi, ut ullamcorper lacus varius at. Mauris id massa porta, tristique libero eu, lacinia du. Sed vitae diam dignissim, pretium sapien vel, ultricies nulla. Etiam efficitur viverra nulla, at facilisis lorem scelerisque sit amet. Nulla volutpat felis sed tortor rutrum vestibulum. Cras cursus mauris quis ante dapibus feugiat. Sed rhoncus congue leo, eget posuere risus tristique eu. Proin at orci nunc. Mauris id massa porta, tristique libero eu, lacinia du. Sed vitae diam dignissim, pretium sapien vel, ultricies nulla. Etiam efficitur viverra nulla, at facilisis lorem scelerisque sit	Messaggi o di errore - Note Non Valide!	

				amet. Nulla volutpat felis sed tortor rutrum vestibulum. Cras cursus mauris quis ante dapibus feugiat. Sed rhoncus congue leo, eget posuere risus tristique eu. Proin at orci nunc.", Esito: 2}		
5	Esito < 0.	Data valida Oravalida Stringa di caratteri di lunghezza ≤ 1000 Esito < 0 [ERROR]	Nessuna	{Data: "11/12/2022", Ora: "11:30", Note: "Telefonata Effettuata", Esito: -1}	Messaggi o di errore - Esito Non Valido!	
6	Esito > 5	Data valida Oravalida Stringa di caratteri di lunghezza ≤ 1000 Valore > 5 [ERROR]	Nessuna	{Data: "11/12/2022", Ora: "11:30", Note: "Telefonata Effettuata", Esito: 6}	Messaggi o di errore - Esito Non Valido!	

PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* “ModificaNoteAppuntamento”

NOTE	ESITO
- Stringa di caratteri di lunghezza ≤ 1000 - Stringa di caratteri di lunghezza > 1000 [ERROR]	- Valore >0 e ≤ 1 : - 0: appuntamento riuscito; - 1: appuntamento fallito; - Valore <0 [ERROR] - Valore >1 [ERROR]

Il numero di test da effettuarsi senza particolari

vincoli è $2 \times 3 = 6$ Introduciamo i vincoli [ERROR]

Il numero di test da eseguire per testare singolarmente i vincoli è 3 (1 per Note, 2 per Esito). Il numero di test risultante è: $(1 \times 1) + 3 = 4$

- **Test Suite**

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti input validi	- Stringa di caratteri di lunghezza ≤ 1000 - Esito >0 e ≤ 1	Nessuna	{Note: "Appuntamenti" Esito: 0}	Note modificate con successo!	Sono salvati nel sistema i dettagli della telefonata.
2	Stringa di caratteri di lunghezza > 1000 [ERROR]	- Stringa di caratteri di lunghezza > 1000 [ERROR] - Esito >0 e ≤ 1	Nessuna	{Note: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam tristique risus nec ipsum cursus, sit amet tincidunt ligula facilisis. Sed fringilla sem ac est congue bibendum. Sed dapibus justo auctor, consequat felis nec, dignissim velit. Sed tincidunt ultrices diam. Mauris eu enim non justo lobortis volutpat at vitae lectus. Cras eu orci at sapien faucibus feugiat. Nulla vulputate eleifend mi, ut	Messaggio di errore - Testo troppo lungo!	

				<p>ullamcorper lacus varius at. Mauris id massa porta, tristique libero eu, lacinia dui. Sed vitae diam dignissim, pretium sapien vel, ultrices nulla. Etiam efficitur viverra nulla, at facilisis lorem scelerisque sit amet. Nulla volutpatfelis sed tortor rutrum vestibulum. Cras cursus mauris quis ante dapibus feugiat. Sed rhoncus congue leo, eget posuere risus tristique eu. Proin at orci nunc.", Esito: 0}</p>		
3	Esito < 0.	Stringa di caratteri di lunghezza ≤ 1000 Esito < 0 [ERROR]	Nessuna	{Note: "Appuntame ntos ", Esito: -1}	Messaggi o di errore - Valore non valido!	
4	Esito > 1	Stringa di caratteri di lunghezza \leq 1000 Valore > 1 [ERROR]	Nessuna	{Note: "Appuntame ntos ", Esito: 3}	Messaggi o di errore - Valore non valido!	

PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* “CreaAppuntamento”

DATA	ORA
- Data con formato valido (gg/mm/aaaa) - Data con formato non valido[ERROR]	- Ora con formato valido [hh:mm] - Ora con formato non valido[ERROR]

Il numero di test da effettuarsi senza particolari vincoli

è $2 \times 2 \times 3 = 24$ Introduciamo i vincoli [ERROR]

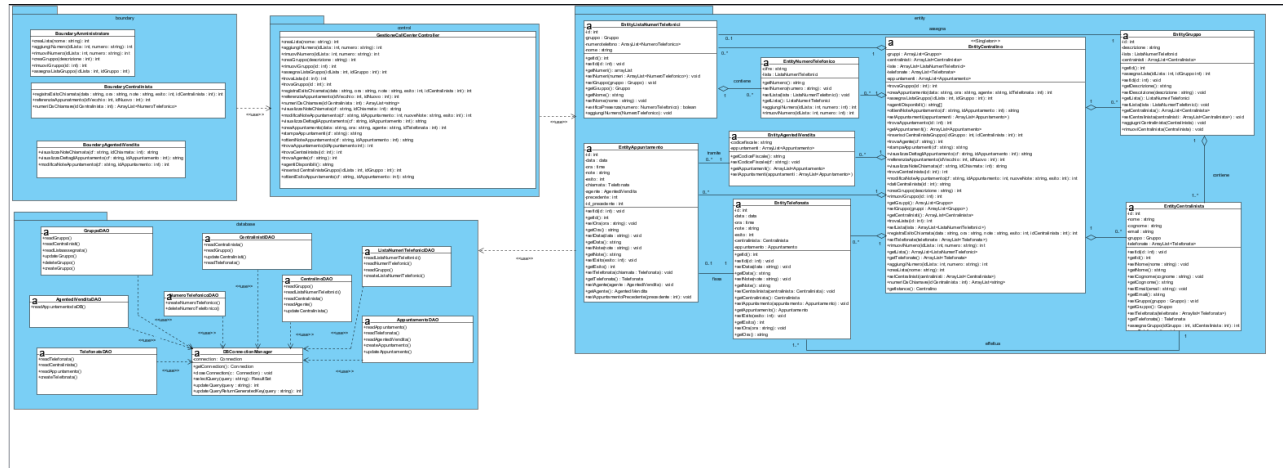
Il numero di test da eseguire per testare singolarmente i vincoli è 5 (1 per Data, 1 per Ora, 1 per Note, 2 per Esito). Il numero di test risultante è: $(1 \times 1 \times 1 \times 1 \times 1) + 5 = 6$

• *Test Suite*

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti input validi	Data valida Ora valida Stringa di caratteri di lunghezza ≤ 1000 Valore esito > 0 e ≤ 1	Nessuna	{Data: "12/12/2022", Ora: "12:30"}	Appuntamento inserito con successo!	Appuntamento salvato nel sistema.
2	Data con formato non valido.	Formato data non valido [ERROR] Ora valida Stringa di caratteri di lunghezza ≤ 1000 Valore esito > 0 e ≤ 1	Nessuna	{Data: "33/13/2022", Ora: "10:30"}	Messaggio di errore - Data non valida!	
3	Ora con formato non valido.	Data valida Formato ora non valido [ERROR] Stringa di caratteri di lunghezza ≤ 1000 Valore esito > 0 e ≤ 1	Nessuna	{Data: "12/12/2022", Ora: "25:30"}	Messaggio di errore - Ora non valida!	

5. Progettazione

5.1 Diagramma delle Classi

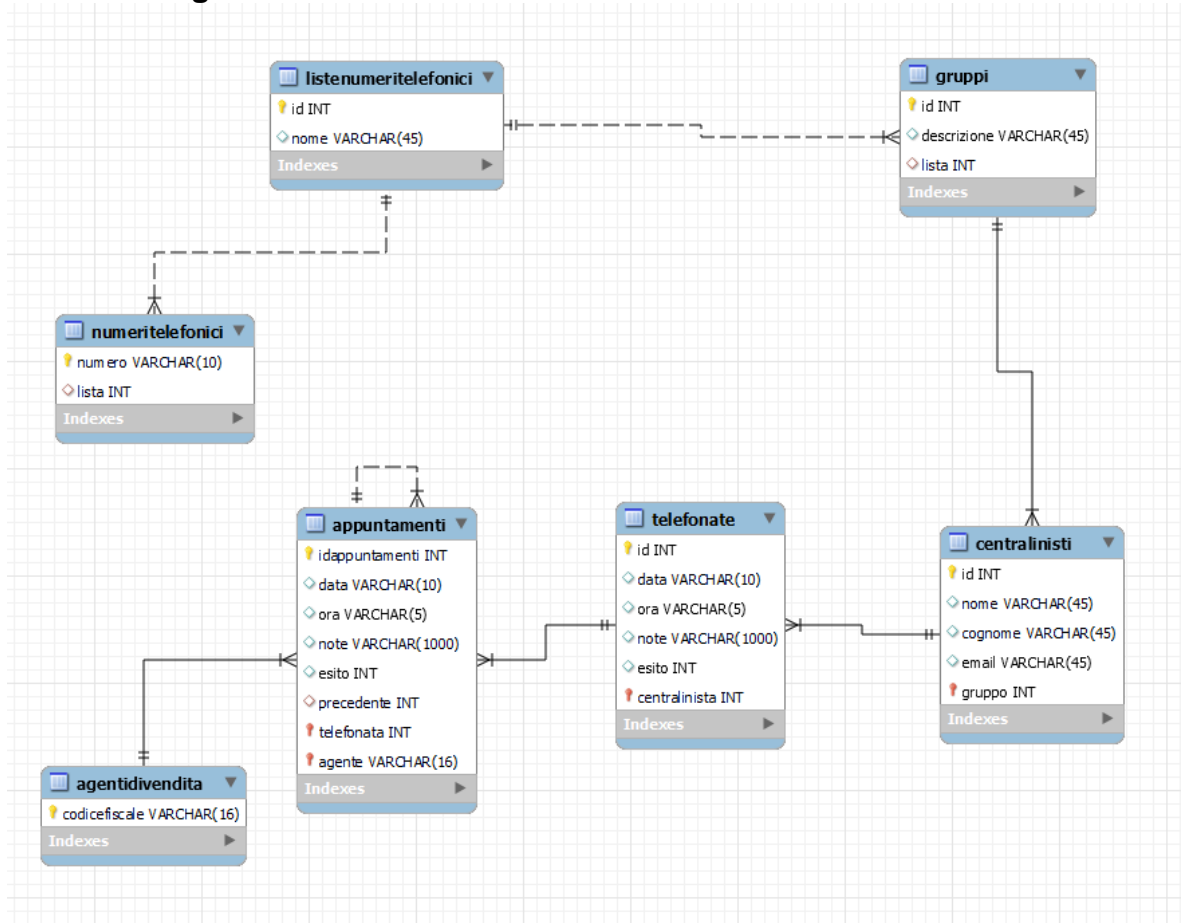


- *Aggiungi Numero (UC2)*



- **Caso d'uso:** *Registra Esito Telefonata (UC9)*

5.3 Modello Logico Dati



6. Implementazione

1. Package

- **Boundary, Control, Entity e Database** che realizzano il Pattern BCED.

2. Classi:

- **Classi del Package Boundary**
 - *MainFrame* rappresenta la finestra principale dell'interfaccia grafica
 - *AdminView*, *CentralinistaView*, *AgenteView* responsabili della logica di presentazione delle funzionalità disponibili rispettivamente di Amministratori, Centralinisti e Agenti;
- **Classi del Package Control**
 - *Controller* controlla l'esecuzione del processo.
- **Classi del Package Entity**
 - *EntityCentralino*: funge da façade per il Package ed è implementata come Singleton

- *EntityAgenteDiVendita*: è responsabile di gestire e conservare i dati relativi ad un agente di vendita
- *EntityCentralinista*: è responsabile di gestire e conservare i dati relativi ad un centralinista
- *EntityAppuntamento*: è responsabile di gestire e conservare i dati relativi ad un appuntamento
- *EntityGruppo*: è responsabile di gestire e conservare i dati relativi ad un gruppo
- *EntityListaNumeriTelefonici*: è responsabile di gestire e conservare i dati relativi ad una lista di numeri telefonici
- *EntityNumeriTelefonici*: è responsabile di gestire e conservare i dati relativi ad un numero di telefono
- **Classi del Package *Database***
 - *DBConnectionManager*: è responsabile di gestire la connessione col Database
 - *DBCentralino*: è responsabile dell'estrazione, lettura e aggiornamento dei dati relativi ad un centralino dal Database
 - *DBCentralinista*: è responsabile dell'estrazione, lettura e aggiornamento dei dati relativi ad un centralinista
 - *DBAgenteDiVendita*: è responsabile dell'estrazione, lettura e aggiornamento dei dati relativi ad un agente di vendita
 - *DBAppuntamento*: è responsabile dell'estrazione, lettura e aggiornamento dei dati relativi ad un appuntamento
 - *DBListaNumeriTelefonici*: è responsabile dell'estrazione, lettura e aggiornamento dei dati relativi ad una lista di numeritelefonici
 - *DBNumeriTelefonici*: è responsabile dell'estrazione, lettura e aggiornamento dei dati relativi ad un numero telefonico
 - *DBGruppo*: è responsabile dell'estrazione, lettura e aggiornamento dei dati relativi ad un gruppo di centralinisti

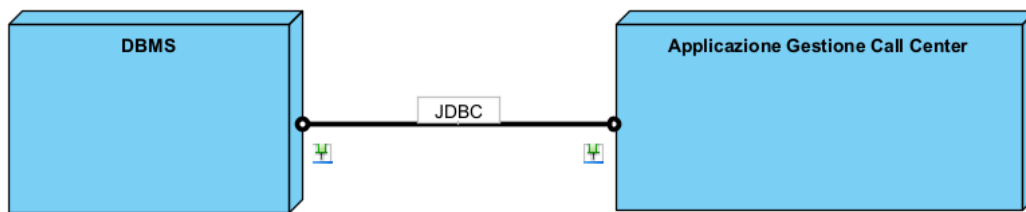
3. Eccezioni:

- ***ListaNonTrovata***
- ***GruppoNonTrovato***
- ***DataNonValida***
- ***OraNonValida***
- ***NumeroNonValido***
- ***NoteNonValide***
- ***EsitoTelefonataNonValido***
- ***EsitoAppuntamentoNonValido***

4. Artefatti necessari per l'installazione ed esecuzione del programma

- Per l'installazione ed esecuzione del programma è necessario includere il file `mysql-connector-java-8.0.29.jar` per realizzare la connessione con il Database creato con MySQLServer.

5. Deployment Diagram



6. Confronto con la stima dei costi:

- Numero di LLOC in Java: 1162
- La stima dei costi aveva previsto un numero di LLOC pari a 1.304, quindi lo scostamento in positivo indica che la stima effettuata in fase di analisi non è stata ecceduta

7. Testing

7.1 Test strutturale

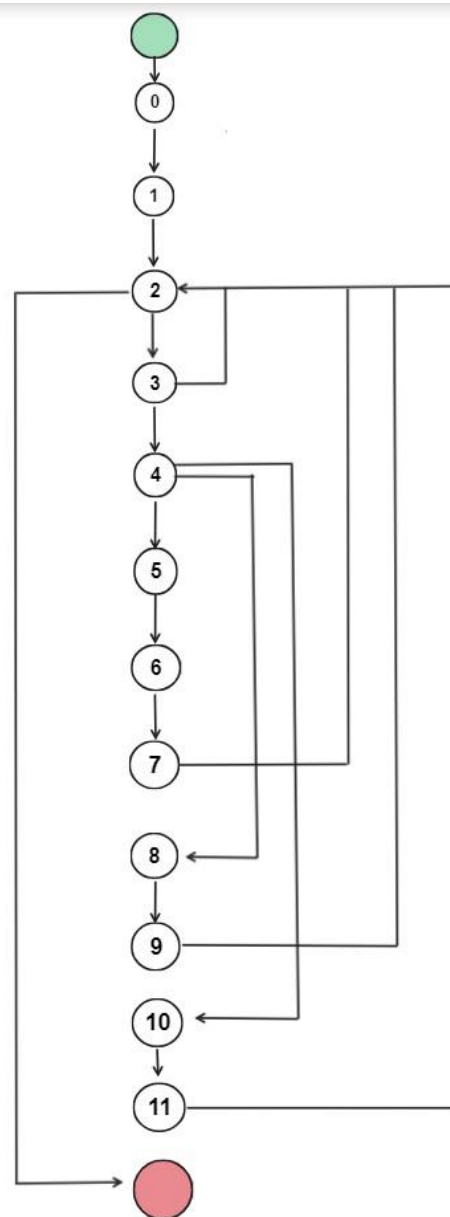
7.1.1 Complessità ciclomatica

```
1. //modificaNoteAppuntamento
2. public int modificaNoteAppuntamento(String cf, int idAppuntamento, String
nuoveNote, int esito) {
3.     int ret=0;
4.     ArrayList<EntityAppuntamento> l = ottieniAppuntamenti(cf);
5.     for(int i=0; i<l.size(); i++) {
6.         if(l.get(i).getId()==idAppuntamento) {
7.             try {
8.                 l.get(i).setEsito(esito);
9.                 l.get(i).setNote(nuoveNote);
10.                ret = l.get(i).aggiornaSuDB();
11.            } catch (NoteNonValide n) {
12.                System.out.println(n);
13.                ret = -1;
14.            } catch (EsitoAppuntamentoNonValido e) {
15.                System.out.println(e);
16.                ret = -1;
17.            }
18.        }
19.    }
20.
21.    return ret;
22. }
23.
```

• **Numero Ciclomático di McCabe:**
 $17(\#nArchi) - 14(\#nNodi) + 2 = 5$

• **Cammini Indipendenti**

1. 0-1-2
2. 0-1-2-3
3. 0-1-2-3-4-10-11-2
4. 0-1-2-3-4-5-6-7-2
5. 0-1-2-3-8-9-2




```

1. // Crea Appuntamento
2. public void actionPerformed(ActionEvent e) {
3.
4. int ret=0;
5. String data = textField.getText();
6. String ora = textField_1.getText();
7. String agente = (String) comboBox.getSelectedItemAt();
8.
9.
10. try {
11.     Controller.isDataValida(data);
12.
13.     Controller.isOraValida(ora);
14.
15.     if(agente.isEmpty() || !(Controller.trovaAgente(agente)>0)){
16. JOptionPane.showMessageDialog(btnNewButton, "Agente Non Trovato", "Error",
JOptionPane.PLAIN_MESSAGE);
17.     } else {
18.         ret = Controller.creaAppuntamento(data, ora, agente,callID);
19.         if(ret>0) {
20.JOptionPane.showMessageDialog(btnNewButton, "Appuntamento inserito correttamente con
id="+ret, "Plain Text", JOptionPane.PLAIN_MESSAGE);
21.         } else {
22. JOptionPane.showMessageDialog(btnNewButton, "Appuntamento non inserita", "Error",
JOptionPane.PLAIN_MESSAGE);
23. }
24. }
25.
26. } catch (OraNonValida ex) {
27. JOptionPane.showMessageDialog(btnNewButton, "Ora Non Valid", "Error",
JOptionPane.PLAIN_MESSAGE);
28.
29.     } catch (DataNonValida ex) {
30.JOptionPane.showMessageDialog(btnNewButton, "Data Non Valid", "Error",
JOptionPane.PLAIN_MESSAGE);
31.     }
32.     }
33.     });
34.

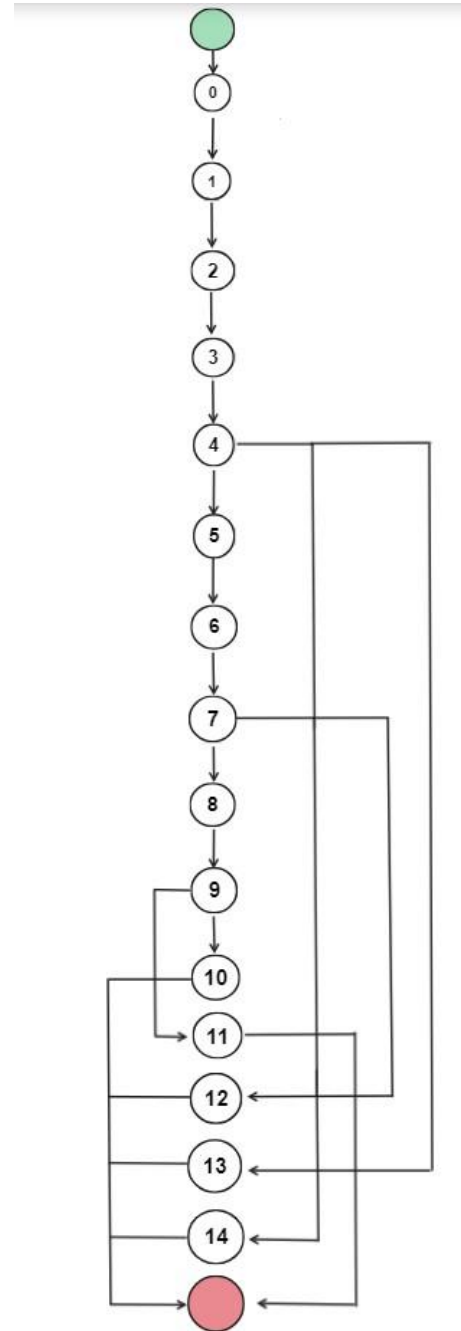
```

• **Numero Ciclomatico di McCabe:**

$$20(\#nArchi)-17(\#nNodi)+2=5$$

• **Cammini Indipendenti:**

1. 0-1-2-3-4-13
2. 0-1-2-3-4-14
3. 0-1-2-3-4-5-6-7-12
4. 0-1-2-3-4-5-6-7-8-9-11
5. 0-1-2-3-4-5-6-7-8-9-10



7.1.2 Test Funzionale

- **RegistraEsitoTelefonata**

Test Case Id	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Output Ottenuti	Post-condizioni Ottenute	Esito (PASS/FAIL)
1	Tutti input validi	Data valida Ora valida Stringa di caratteri di lunghezza ≤ 1000 Valore > 0 e	Nessuna	{Data: "11/12/2022", Ora: "15:30", Note: "Telefonata Effettuata", Esito: 2}	Chiamata registrata con successo !	Sono salvati nel sistema i dettagli della telefonata.	Chiamata registrata con successo!	Sono salvati nel sistema i dettagli della telefonata.	PASS
2	Data con formato non valido.	Formato data non valido [ERROR] Ora valida Stringa di caratteri di lunghezza ≤ 1000 Valore > 0 e	Nessuna	{Data: "11/13/2022", Ora: "15:30", Note: "Telefonata Effettuata", Esito: 2}	Messaggi o di errore - Data non valida!	-	[ECCEZIONE] Data Non Valida!	-	PASS
3	Ora con formato non valido.	Data valida Formato ora non valido [ERROR] Stringa di caratteri di lunghezza ≤ 1000 Valore > 0 e	Nessuna	{Data: "11/12/2022", Ora: "25:30", Note: "Telefonata Effettuata", Esito: 2}	Messaggi o di errore - Ora non valida!	-	[ECCEZIONE] Ora Non Valida!	-	PASS
4	Stringa di caratteri di lunghezza > 1000 .	Data valida Ora valida Stringa di caratteri di lunghezza > 1000 [ERROR] Valore > 0 e	Nessuna	{Data: "11/12/2022", Ora: "12:30", Note: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam tristique risus nec ipsum cursus, sit amet tincidunt ligula facilisis. Sed fringilla sem ac est congue bibendum. Sed dapibus justo auctor, consequat felis	Messaggi o di errore - Note Non Valide!	-	[ECCEZIONE] Note Non Valide!	-	Esito (PASS/FAIL)

				<p>nec, dignissim velit. Sed tincidunt ultrices diam. Mauris eu enim non justo lobortis volutpat at vitae lectus. Cras eu orci at sapien faucibus feugiat. Nulla vulputate eleifend mi, ut ullamcorper lacus varius at. Mauris id massa porta, tristique libero eu, lacinia dui. Sed vitae diam dignissim, pretium sapien vel, ultricies nulla. Etiam efficitur viverra nulla, at facilisis lorem scelerisque sit amet. Nulla volutpat felis sed tortor rutrum vestibulum. Cras cursus mauris quis ante dapibus feugiat. Sed rhoncus congue leo, eget posuere risus tristique eu. Proin at orci nunc. Mauris id massa porta, tristique libero eu, lacinia dui. Sed vitae diam dignissim, pretium sapien vel, ultricies nulla. Etiam efficitur viverra nulla, at facilisis lorem scelerisque sit amet. Nulla volutpat felis sed tortor rutrum vestibulum. Cras cursus mauris quis</p>					
--	--	--	--	--	--	--	--	--	--

				ante dapibus feugiat. Sed rhoncus congue leo, eget posuere risus tristique eu. Proin at orci nunc.", Esito: 2}					
5	Esito < 0.	Data valida Ora valida Stringa di caratteri di lunghezza ≤ 1000 Esito < 0 [ERROR]	Nessuna	{Data: "11/12/2022", Ora: "11:30", Note: "Telefonata Effettuata", Esito: -1}	Messaggi o di errore - Esito Non Valido!	-	[ECCEZIONI] Esito Non Valido!	-	PASS
6	Esito > 5	Data valida Ora valida Stringa di caratteri di lunghezza ≤ 1000 Valore > 5 [ERR	Nessuna	{Data: "11/12/2022", Ora: "11:30", Note: "Telefonata Effettuata", Esito: 6}	Messaggi o di errore - Esito Non Valido!	-	[ECCEZIONI] Esito Non Valido!	-	PASS

- CreaAppuntamento

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Output Ottenuti	Post-Condizioni Ottenute	Esito (PASS/FAIL)
1	Tutti input validi	Data valida Ora valida Stringa di caratteri di lunghezza ≤ 1000 Valore esito > 0 e ≤ 1	Nessuna	{Data: "12/12/2022", Ora: "12:30"}	Appuntamento inserito con successo!	Appuntamento salvato nel sistema.	Appuntamento inserito con successo!	Sono salvati nel sistema i dettagli dell'appuntamento	PASS
2	Data con formato non valido.	Formato data non valido [ERROR] Ora valida Stringa di caratteri di lunghezza ≤ 1000 Valore esito > 0 e ≤ 1	Nessuna	{Data: "33/13/2022", Ora: "10:30"}	Messaggio di errore - Data non valida!	-	[ECCEZIONE] Data Non Valida!	-	PASS
3	Ora con formato non valido.	Data valida Formato ora non valido [ERROR] Stringa di caratteri di lunghezza ≤ 1000	Nessuna	{Data: "12/12/2022", Ora: "25:30"}	Messaggio di errore - Ora non valida!	-	[ECCEZIONE] Ora Non Valida!	-	PASS

		Valore esito > 0 e ≤ 1							
--	--	-----------------------------------	--	--	--	--	--	--	--

- **ModificaNoteAppuntamento**

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Output Ottenuti	Post-Condizioni Ottenute	Esito (PASS/FAIL)
1	Tutti input validi	- Stringa di caratteri di lunghezza ≤ 1000 - Esito >0 e ≤ 1	Nessuna	{Note: "Appuntamentos" Esito: 0}	Note modificate con successo!	Sono salvati nel sistema i dettagli della telefonata.	Note modificate con successo!	Le modifiche effettuate alle note relative ad un dato appuntamento vengono salvate nel database	PASS
2	Stringa di caratteri di lunghezza > 1000 [ERROR]	- Stringa di caratteri di lunghezza > 1000 [ERROR] - Esito >0 e ≤ 1	Nessuna	{Note: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam tristique risus nec ipsum cursus, sit amet tincidunt ligula facilisis. Sed fringilla sem ac est congue bibendum. Sed dapibus justo auctor, consequat felis nec, dignissim velit. Sed tincidunt ultrices diam. Mauris eu enim non justo lobortis volutpat at vitae lectus. Cras eu orci at sapien faucibus feugiat. Nulla vulputate	Messaggio di errore - Testo troppo lungo!	-	[ECCEZIONI] Testo troppo lungo!	-	PASS

				eleifend mi, ut ullamcorper lacus varius at. Mauris id massa porta, tristique libero eu, lacinia dui. Sed vitae diam dignissim, pretium sapien vel, ultricies nulla. Etiam efficitur viverra nulla, at facilisis lorem scelerisque sit amet. Nulla volutpat felis sed tortor rutrum vestibulum. Cras cursus mauris quis ante dapibus feugiat. Sed rhoncus congue leo, eget posuere risus tristique eu. Proin at orci nunc.", Esito: 0}					
3	Esito < 0.	Stringa di caratteri di lunghezza ≤ 1000 Esito < 0 [ERROR]	Nessuna	{Note: "Appuntamentos", Esito: -1}	Messaggio di errore - Valore non valido!	-	[ECCEZIONI] Valore non valido!	-	PASS
4	Esito > 1	Stringa di caratteri di lunghezza ≤ 1000 Valore > 1	Nessuna	{Note: "Appuntamentos", Esito: 3}	Messaggio di errore - Valore non valido!	-	[ECCEZIONI] Valore non valido!	-	PASS

		[ERROR]							
--	--	---------	--	--	--	--	--	--	--