

Università degli Studi di Napoli Federico II

**Corso di Laurea in Ingegneria Informatica**

**Corso di Ingegneria del Software**

**Prof. S. Russo - A.A. 2020 - 21**

***Progetto***

KeepFit

Gruppo: libre

Studenti:

Vittorio Libretti M63/1123 v.libretti@studenti.unina.it

Versione 1 del *16/12/2020*

**Indice**

1. Requisiti 3

1.1 Specifiche informali d’utente 3

1.2 Requisiti software 4

2. Analisi e specifica dei requisiti 5

2.1 Modellazione dei casi d’uso 5

2.2 Diagramma delle classi 5

2.3 Diagrammi di sequenza 5

3. Stima dei costi 6

4. Piano di test funzionale 7

5. Progettazione 8

5.1 Diagramma delle classi 8

5.2 Diagrammi di sequenza 8

6. Implementazione 9

7. Testing 10

7.1 Test funzionale 10

7.2 Test strutturale 12

7.2.1 Complessità ciclomatica 12

7.2.2 Test di unità 12

7.3 Test di unità con JUnit 12

# Requisiti

## Specifiche informali d’utente

Si vuole realizzare un’applicazione per gestire gli abbonati di una palestra e per offrire loro un elenco di esercizi da consultare durante la loro sessione di allenamento.

Un utente amministratore, accedendo all’applicazione con username e password, può visualizzare un elenco di abbonati iscritti alla palestra. L’admin può aggiungere nuovi abbonati o modificare o cancellare abbonati precedentemente registrati. Di ogni abbonato si vuole visualizzare nome, cognome, data di nascita, stato dell’abbonamento (“attivo”, “scaduto”). Inoltre, selezionando un abbonato dalla lista, l’amministratore può visualizzare una scheda di dettaglio della persona: nome, cognome, codice fiscale, data di nascita, residenza e/o domicilio, data di iscrizione, telefono fisso e/o mobile, data di inizio e fine abbonamento.

Quando l’abbonamento della palestra scade, esso viene segnato come scaduto; l’amministratore provvederà a riattivarlo.

L’amministratore può promuovere gli abbonati a premium, in seguito al pagamento di un supplemento mensile. L’abbonato premium ha diritto ad avere un account personale nell’applicazione, al quale può eccedere tramite username e password. L’amministratore fornirà all’abbonato le credenziali di accesso. L’abbonato premium può visualizzare protocolli predefiniti di allenamento.

L’amministratore può visualizzare un elenco di account visualizzando username, nome e cognome dell’abbonato, data di nascita, stato dell’abbonamento premium (“in corso”, “scaduto”). L’account viene cancellato automaticamente quando scade l’abbonamento premium o manualmente dall’admin, e può essere ricreato dall’admin con il rinnovo.

Un protocollo di allenamento è un elenco di esercizi suddivisi per sessioni di allenamento. Ogni protocollo appartiene ad una specifica categoria determinata dall’obiettivo che si vuole raggiungere (“restare in forma”, “dimagrire”, “aumenta massa muscolare”, ecc).

Gli esercizi sono raggruppati per gruppo muscolare; ogni esercizio è caratterizzato da un nome, da una descrizione dei movimenti da eseguire, da un contenuto multimediale (immagine o video) che mostra visivamente l’esecuzione dell’esercizio.

Mediante delle postazioni dislocate in vari punti della palestra, l’abbonato ha la possibilità di utilizzare l’applicazione; le postazioni consistono in monitor touchscreen agganciati a colonnine o supporto a muro.

## Requisiti software

Riportare la traccia revisionata a seguito di chiarimenti, disambiguazione di termini, ristrutturazione delle frasi che esprimono più requisiti, separazione di requisiti funzionali e requisiti sui dati, etc.

E’ possibile riportare più viste sui requisiti (es.: in una pagina una vista di separazione requisiti funzionali e requisiti sui dati; in altra pagina riportare l’analisi nomi-verbi, eventualmente anche fatta con VisualParadigm).

Se rilevante (cioè se non banale), riportare un glossario dei termini in una tabella termine-significato-sinonimi.

Alla fine, riportare una tabella con i requisiti numerati secondo una codifica (es: RF01, RF02, RFn: requisiti funzionali; RD01, RD02, RDm: requisiti sui dati).

# Analisi e specifica dei requisiti

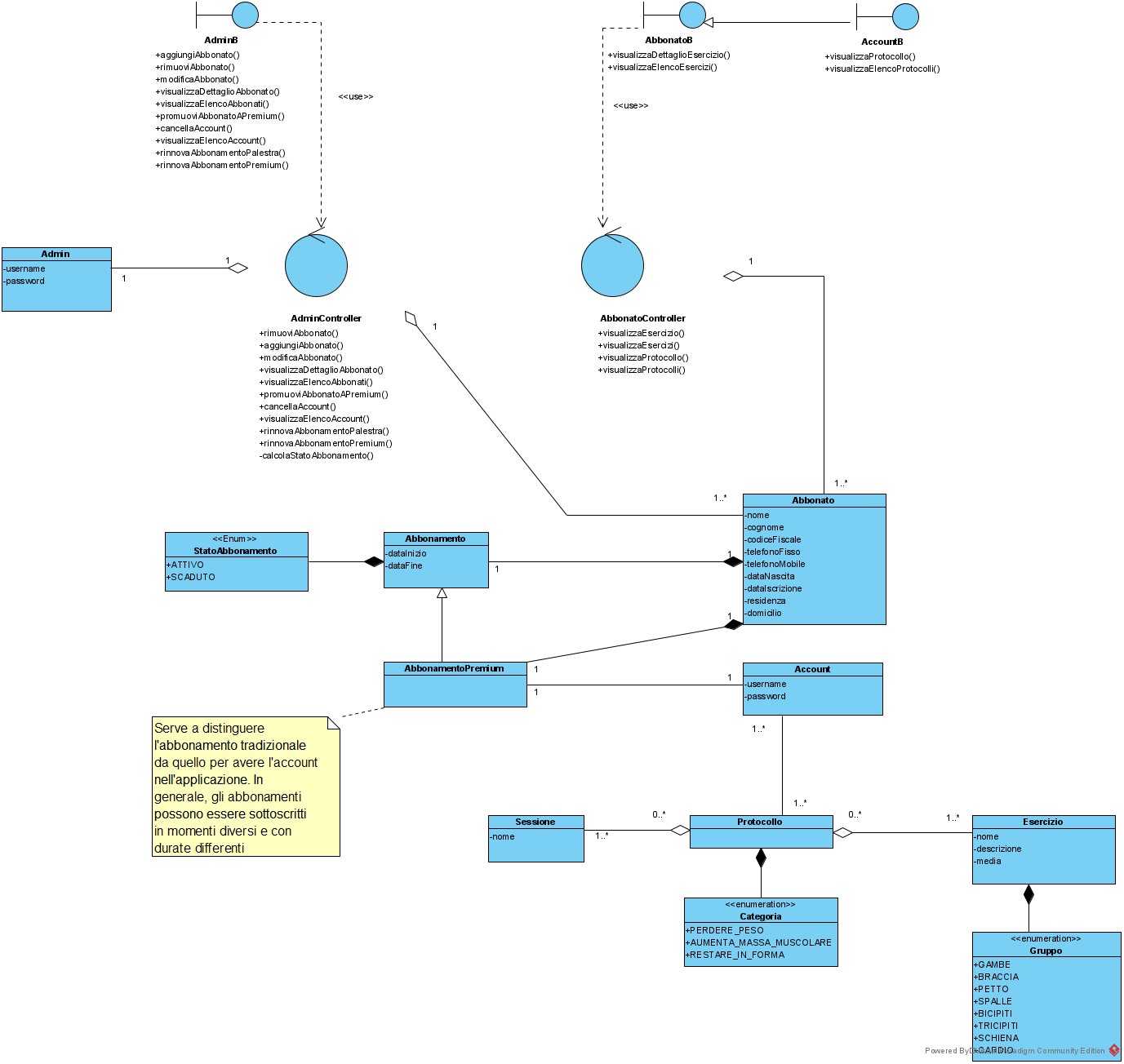
## Modellazione dei casi d’uso

Elencare attori primari, attori secondari, casi d’uso, casi d’uso di inclusione e di estensione.

Riportare il diagramma dei casi d’uso.

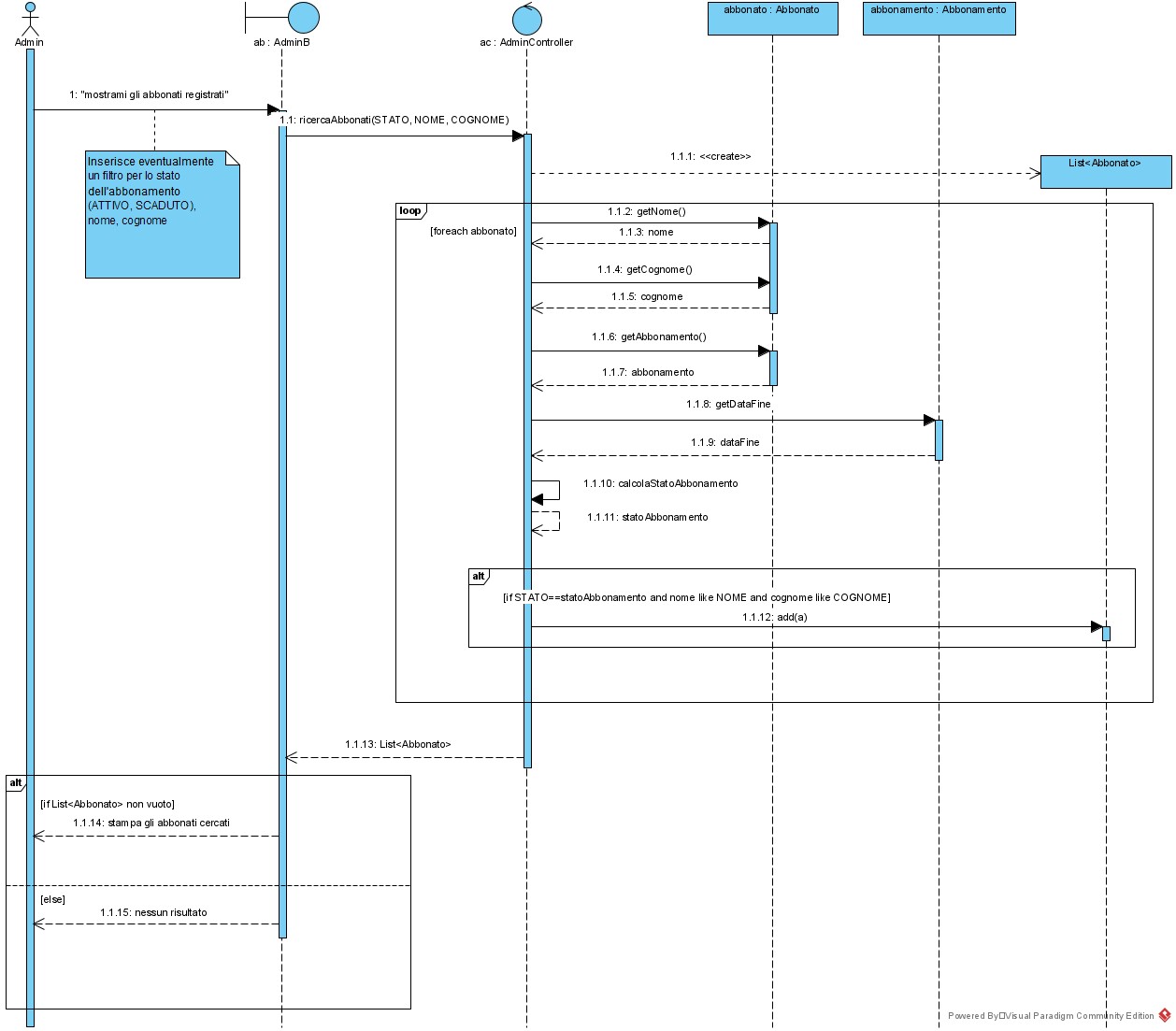
Riportare lo scenario principale per ogni casi d’uso sviluppato fino alla codifica in Java (uno per membro del gruppo).

## Diagramma delle classi

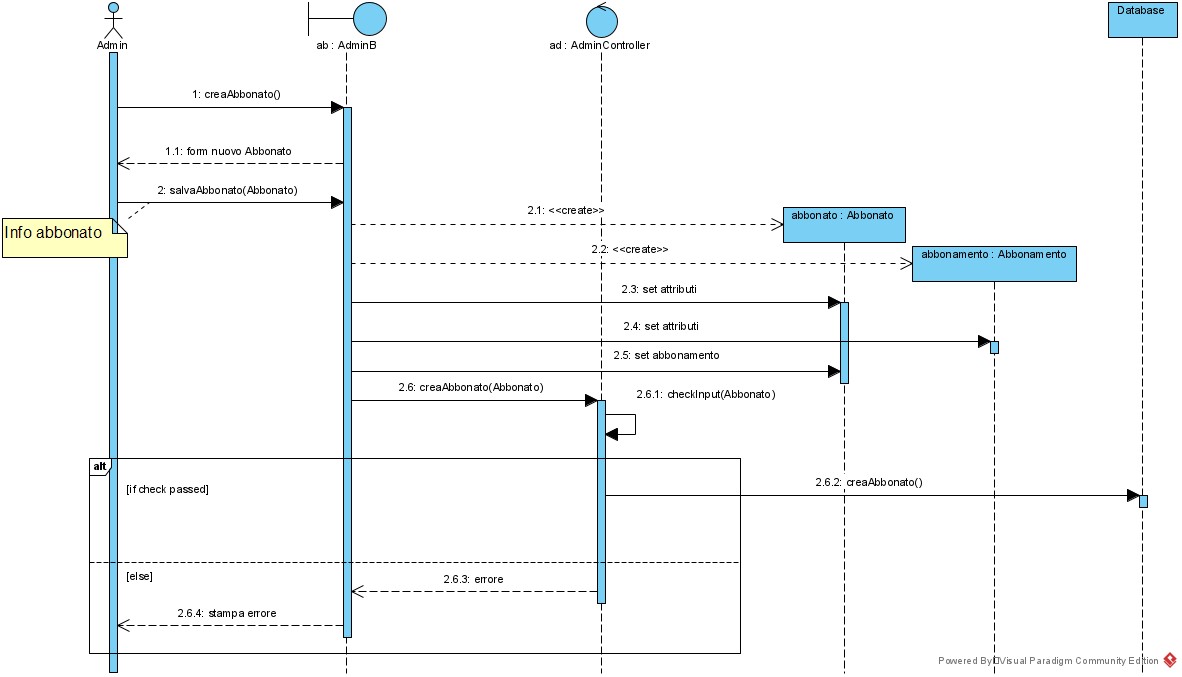


## Diagrammi di sequenza

### Visualizza elenco abbonati



### Crea abbonato



# Stima dei costi

## Valutazione degli Unadjusted Function Point (UFP)

### Ambito di conteggio

Sviluppo di progetto

### Identificazione funzioni dati e transazionali

Non sono presenti External Interface Files (EIF) poichè il sistema non coopera con altri sistemi esterni, quindi NEIF=0. È presente il database che è un Internal Logical File (ILF), quindi NILF = 1.

La complessità di ILF è 10 perché RET=8 (8 tabelle) e DET=19 (input diversi).

-EQ-

Le funzionalità che eseguono solo query sono 7. Per ognuno di esse FTR=1.

Per i DET consideriamo la funzionalità più complessa:

*Visualizza elenco abbonati*: ha tre parametri di input(nome, cognome, stato abbonamento). Le combinazioni degli input possibili per la ricerca sono 7; aggiungiamo 1 input per applicare il filtro e 1 input per eventuale messaggio di risposta. Quindi, DET=9. Il peso assegnato è 3.

-EI-

Le funzionalità che modificano L’ILF sono 8. Per ognuno di esse FTR=1.

Per i DET consideriamo la funzionalità più complessa:

*Crea abbonato*: bisogna fornire nome, cognome, data di nascita, codice fiscale, residenza, domicilio, data di iscrizione, data inizio e fine abbonamento. Aggiungiamo 1 input per il bottone di invio dati e 1 input per eventuale messaggio di risposta. Quindi, DET=11. Il peso assegnato è 3.

|  |  |  |  |
| --- | --- | --- | --- |
| INDICE | VALORE | PESO | VPI |
| NILF | 1 | 10 | 10 |
| NEIF | 0 | 5 | 0 |
| NEI | 8 | 3 | 24 |
| NEO | 0 | 4 | 0 |
| NEQ | 7 | 3 | 21 |
|  |  | totale | 55 |

## Valutazione dei FP

UFP = 55

|  |  |
| --- | --- |
| CARATTERISTICHE GENERALI | VALORE |
| COMUNICAZIONE DATI | 1 |
| DISTRIBUZIONE ELABORAZIONE | 1 |
| PRESTAZIONI | 0 |
| UTILIZZO INTENSIVO CONFIGURAZIONE | 0 |
| FREQUENZA DELLE TRANSAZIONI | 2 |
| AGGIORNAMENTO INTERATTIVO | 1 |
| EFFICIENZA PER L’UTENTE FINALE | 1 |
| INSERIMENTO DATI INTERATTIVO | 1 |
| COMPLESSITA’ ELABORATIVA | 1 |
| RIUSABILITA’ | 1 |
| FACILITA’ INSTALLAZIONE | 2 |
| FACILITA’ GESTIONE OPERATIVA | 2 |
| MOLTEPLICITA’ DI SITI | 0 |
| FACILITA’ DI MODIFICA | 2 |

AFP = 0.65 + 0.01 \*(3\*0+7\*1+4\*2) = 0.8

FP = UFP x AFP = 55 \* 0.8 = 44

##### Valutazione LLOC

JAVA LLOC = 44\*53 = 2332

# Piano di test funzionale

**PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* PER LA FUNZIONALITÀ DI “*Visualizza elenco abbonati*”.**

|  |  |  |  |
| --- | --- | --- | --- |
| Categoria Tipo di ricerca | Categoria Numero abbonati su DB | Categoria Stato abbonamento | Categoria Esito ricerca |
| * Per nome [SINGLE] * Per cognome * Per stato abbonamento * Per nome e cognome [SINGLE] * Per nome e stato abbonamento [SINGLE] * Per cognome e stato abbonamento [SINGLE] * Per nome, cognome e stato abbonamento [SINGLE] * Nessun nome, cognome, stato[SINGLE] | * 0 [SINGLE] * 1 [SINGLE] * > 1 | * Solo attivi [SINGLE] * Solo scaduti [SINGLE] * Entrambi | * 0 * 1 * > 1 |

**TEST SUITE**

La funzionalità ha 4 categorie. La prima categoria ha 8 classi di valori, la seconda 3 classi di valori, la terza 3 classi di valori, la quarta 3 classi di valori. Il numero totale di test senza vincoli è:

8x3x3x3=216

Vincoli error: 0

Vincoli single: 10

Vincoli property: 0

Il numero di test con i vincoli single ed error sono:

2x1x1x3=6

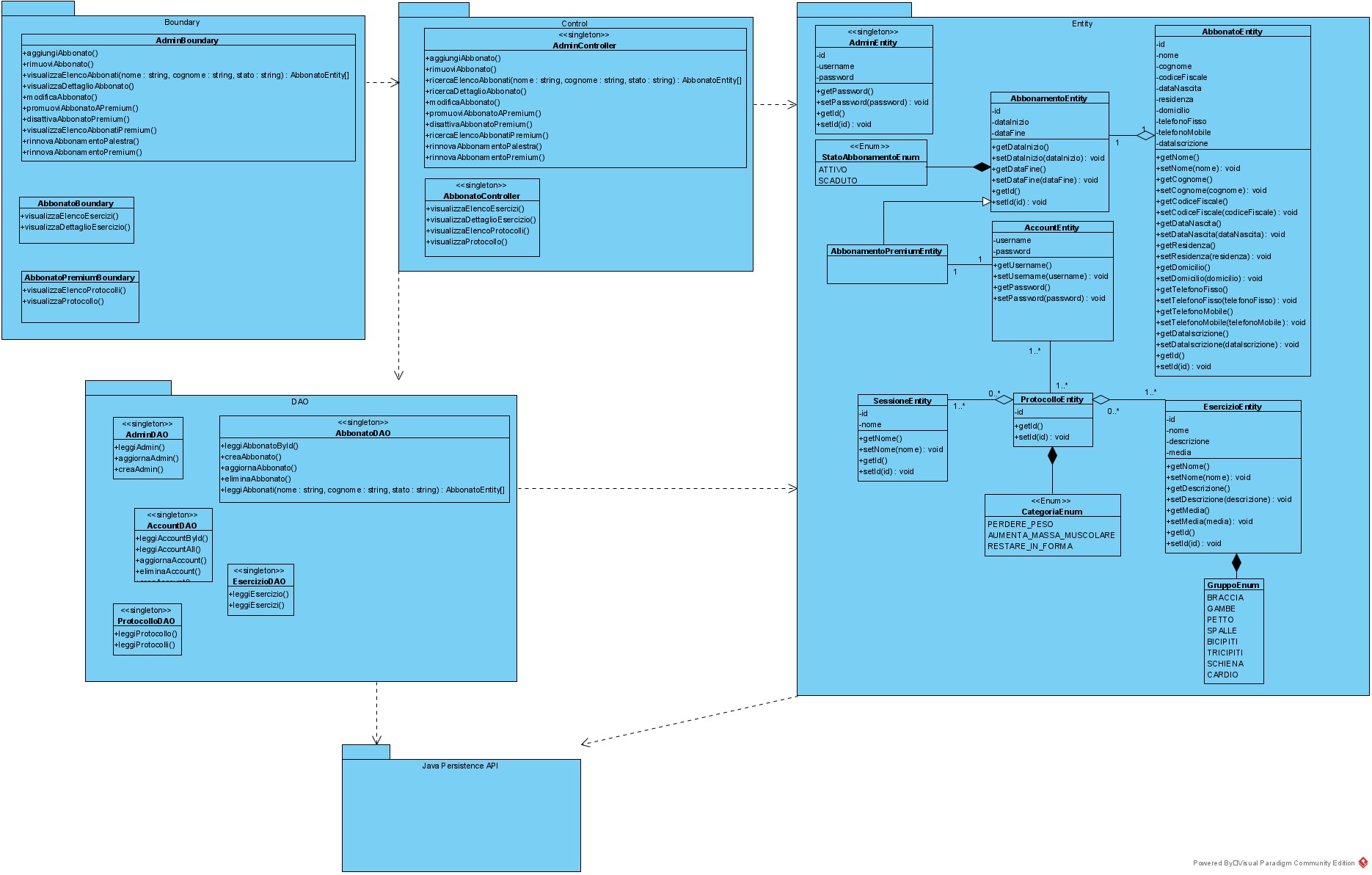
Quindi il numero di test da eseguire è:

6 + 10(single) =16

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** |
| 1 |  | Tipo ricerca: COGNOME  Numero abbonati:3  Stato abbonamento: ENTRAMBI  Esito ricerca:0 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, SCADUTO]  } | Ricerca( “”, ” Esposito”, “”) | 0 risultati | N/A |
| 2 |  | Tipo ricerca: STATO ABBONAMENTO  Numero abbonati:3  Stato abbonamento: ENTRAMBI  Esito ricerca:0 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, ATTIVO]  } | Ricerca(“”, “”, “SCADUTO” ) | 0 risultati | N/A |
| 3 |  | Tipo ricerca: COGNOME  Numero abbonati:3  Stato abbonamento: ENTRAMBI  Esito ricerca:1 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, SCADUTO]  } | Ricerca( “”, “ Rossi”, “”) | 1 risultati | N/A |
| 4 |  | Tipo ricerca: STATO ABBONAMENTO  Numero abbonati:3  Stato abbonamento: ENTRAMBI  Esito ricerca:1 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, SCADUTO]  } | Ricerca(“”, “”, “ SCADUTO” ) | 1 risultati | N/A |
| 5 |  | Tipo ricerca: COGNOME  Numero abbonati:3  Stato abbonamento: ENTRAMBI  Esito ricerca:>1 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Rossi , ATTIVO],  [Andrea, Giallo, SCADUTO]  } | Ricerca(“”, “ Rossi”, “” ) | 2 risultati | N/A |
| 6 |  | Tipo ricerca: STATO ABBONAMENTO  Numero abbonati:4  Stato abbonamento: ENTRAMBI  Esito ricerca: >1 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, SCADUTO]  } | Ricerca(“”, “”, “ATTIVO” ) | 2 risultati | N/A |
| 7 |  | Tipo ricerca: STATO ABBONAMENTO  Numero abbonati:3  Stato abbonamento: ATTIVO  Esito ricerca:>1 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, ATTIVO]  } | Ricerca(“”, “”, “ ATTIVO ” ) | 3 risultati | N/A |
| 8 |  | Tipo ricerca: QUALSIASI  Numero abbonati:3  Stato abbonamento: SCADUTO  Esito ricerca:>1 | DB = {  [Mario, Rossi, SCADUTO ],  [Luca, Verde, SCADUTO ],  [Andrea, Giallo, SCADUTO ]  } | Ricerca(“”, “”, “ SCADUTO” ) | 3 risultati | N/A |
| 9 |  | Tipo ricerca: COGNOME  Numero abbonati:1  Stato abbonamento: ATTIVO  Esito ricerca:1 | DB = {  [Mario, Rossi, ATTIVO]  } | Ricerca( “”, “ Rossi”, “”) | 1 risultati | N/A |
| 10 |  | Tipo ricerca: STATO ABBONAMENTO  Numero abbonati:0  Stato abbonamento: N/A  Esito ricerca:0 | DB = {  } | Ricerca( “”, “”, “ ATTIVO”) | 0 risultati | N/A |
| 11 |  | Tipo ricerca: TUTTI I CAMPI VUOTI  Numero abbonati:2  Stato abbonamento: ENTRAMBI  Esito ricerca:2 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, SCADUTO],  } | Ricerca( “”, “”, “ ”) | 2 risultati | N/A |
| 12 |  | Tipo ricerca: NOME  Numero abbonati:3  Stato abbonamento:ENTRAMBI  Esito ricerca:1 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, SCADUTO]  } | Ricerca( “ Luca”, “”, “”) | 1 risultati | N/A |
| 13 |  | Tipo ricerca: NOME E COGNOME  Numero abbonati:3  Stato abbonamento:ENTRAMBI  Esito ricerca:1 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, SCADUTO]  } | Ricerca( “Mario”, “Rossi”, “”) | 1 risultati | N/A |
| 14 |  | Tipo ricerca: NOME E STATO ABBONAMENTO  Numero abbonati:3  Stato abbonamento:ENTRAMBI  Esito ricerca:0 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, SCADUTO]  } | Ricerca( “Andrea”, “”, “ATTIVO”) | 0 risultati | N/A |
| 15 |  | Tipo ricerca: COGNOME E STATO ABBONAMENTO  Numero abbonati:3  Stato abbonamento: ENTRAMBI  Esito ricerca:0 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, SCADUTO]  } | Ricerca( “”, “Verde”, “SCADUTO”) | 0 risultati | N/A |
| 16 |  | Tipo ricerca:NOME, COGNOME, STATO  Numero abbonati:3  Stato abbonamento: ENTRAMBI  Esito ricerca:0 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, SCADUTO]  } | Ricerca(“Mario”, “Esposito”, “ATTIVO” ) | 0 risultati | N/A |

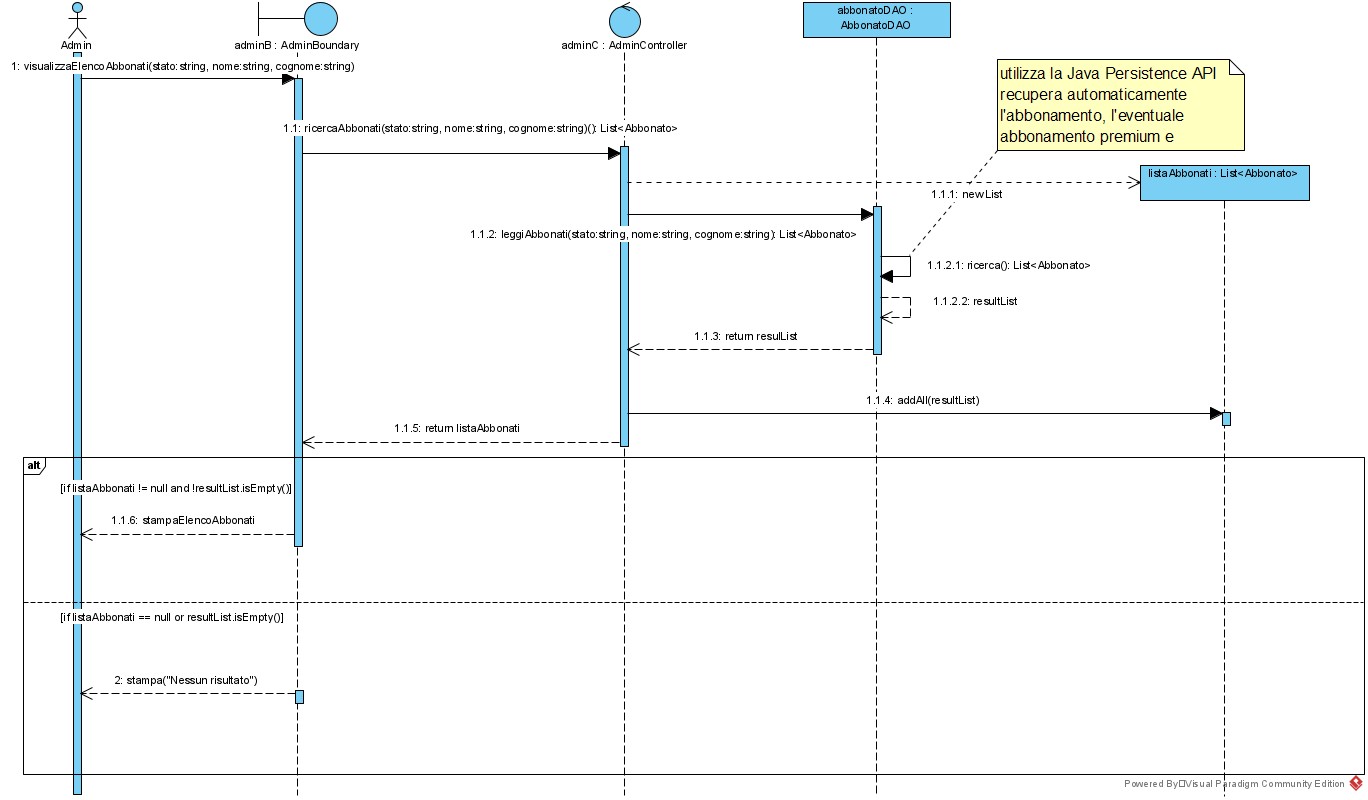
# Progettazione

## Diagramma delle classi

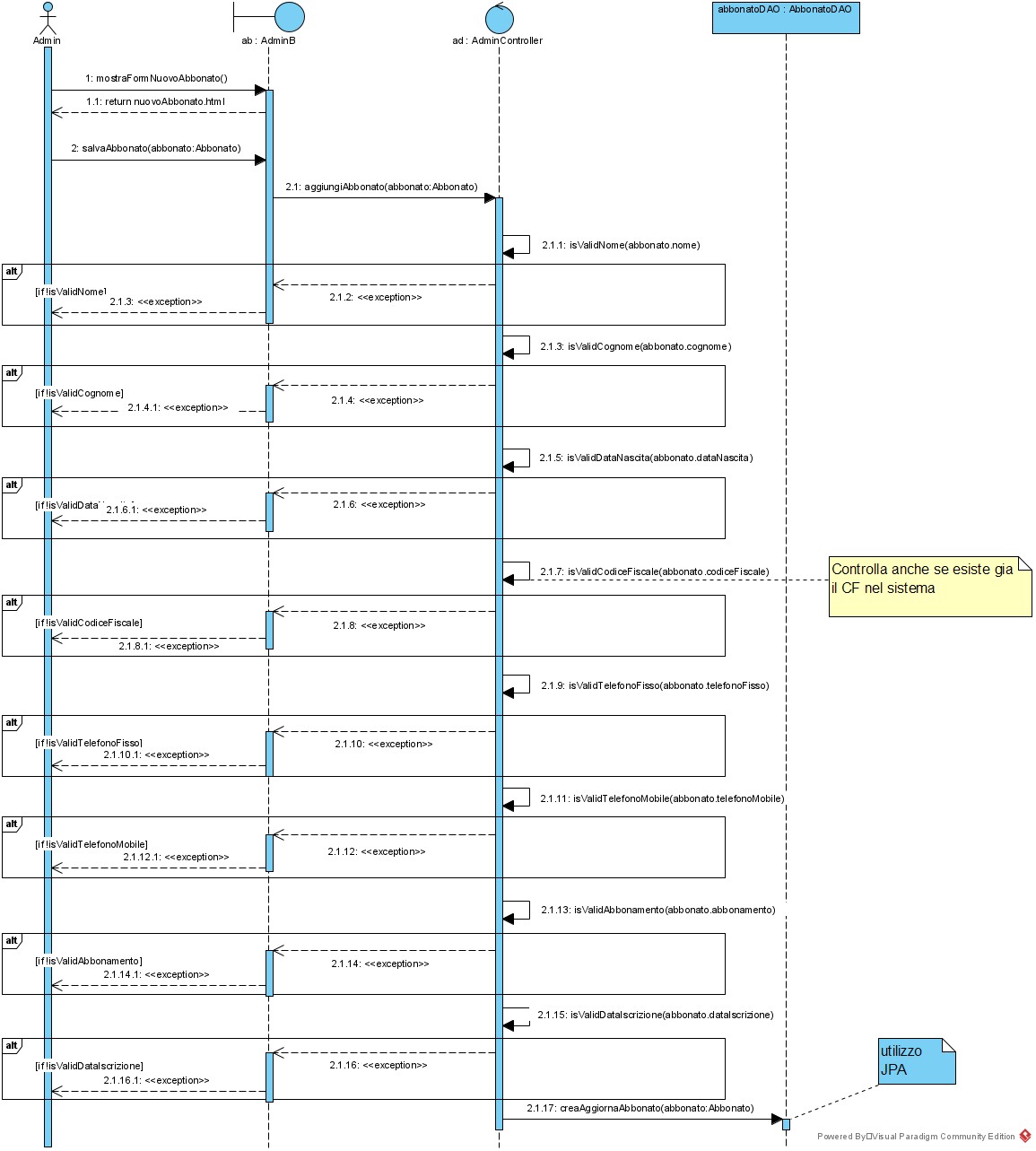


## Diagrammi di sequenza

Visualizza elenco abbonati



Crea abbonato



# Implementazione

Non includere il codice sorgente, ma descrivere l’implementazione in Java, descrivendo gli artefatti di codifica:

* elencare i package, le classi, i tipi di eccezione definiti
* elencare gli artefatti necessari per l’installazione ed esecuzione del programma, senza ovviamente l’ambiente di sviluppo come Eclipse (DB h2, eventuali librerie e versioni di Java che l’utilizzatore deve avere installati, file .class, .jar, …)
* produrre un eventuale diagramma di deployment
* inserire la documentazione del codice prodotta con Javadoc (relativamente alle funzionalità implementate)

## Elenco package

* [it.keepfit](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\package-frame.html)
* [it.keepfit.boundary](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\boundary\package-frame.html)
* [it.keepfit.boundary.impl](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\boundary\impl\package-frame.html)
* [it.keepfit.constants](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\constants\package-frame.html)
* [it.keepfit.control](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\control\package-frame.html)
* [it.keepfit.control.impl](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\control\impl\package-frame.html)
* [it.keepfit.dao](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\dao\package-frame.html)
* [it.keepfit.dao.impl](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\dao\impl\package-frame.html)
* [it.keepfit.entity](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\entity\package-frame.html)
* [it.keepfit.exception](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\exception\package-frame.html)

## Classi

* [Abbonamento](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\entity\Abbonamento.html)
* [AbbonamentoPremium](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\entity\AbbonamentoPremium.html)
* [Abbonato](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\entity\Abbonato.html)
* [*AbbonatoBoundary*](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\boundary\AbbonatoBoundary.html)
* [AbbonatoBoundaryImpl](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\boundary\impl\AbbonatoBoundaryImpl.html)
* [*AbbonatoController*](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\control\AbbonatoController.html)
* [AbbonatoControllerImpl](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\control\impl\AbbonatoControllerImpl.html)
* [*AbbonatoDAO*](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\dao\AbbonatoDAO.html)
* [AbbonatoDAOImpl](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\dao\impl\AbbonatoDAOImpl.html)
* [AbbonatoException](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\exception\AbbonatoException.html)
* [Account](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\entity\Account.html)
* [*AccountBoundary*](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\boundary\AccountBoundary.html)
* [AccountBoundaryImpl](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\boundary\impl\AccountBoundaryImpl.html)
* [*AccountDAO*](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\dao\AccountDAO.html)
* [AccountDAOImpl](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\dao\impl\AccountDAOImpl.html)
* [Admin](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\entity\Admin.html)
* [*AdminBoundary*](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\boundary\AdminBoundary.html)
* [AdminBoundaryImpl](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\boundary\impl\AdminBoundaryImpl.html)
* [*AdminController*](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\control\AdminController.html)
* [AdminControllerImpl](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\control\impl\AdminControllerImpl.html)
* [*AdminDAO*](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\dao\AdminDAO.html)
* [Categoria](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\entity\Categoria.html)
* [Esercizio](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\entity\Esercizio.html)
* [*EsercizioDAO*](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\dao\EsercizioDAO.html)
* [EsercizioDAOImpl](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\dao\impl\EsercizioDAOImpl.html)
* [GruppoMuscolare](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\entity\GruppoMuscolare.html)
* [KeepfitWebappApplication](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\KeepfitWebappApplication.html)
* [Protocollo](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\entity\Protocollo.html)
* [*ProtocolloDAO*](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\dao\ProtocolloDAO.html)
* [ProtocolloDAOImpl](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\dao\impl\ProtocolloDAOImpl.html)
* [Sessione](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\entity\Sessione.html)
* [StatoAbbonamento](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\constants\StatoAbbonamento.html)

## Elenco artefatti

* spring-boot-starter-data-jpa v2.4.0
* spring-boot-starter-thymeleaf v2.4.0
* spring-boot-starter-web v2.4.0
* h2 v1.4.200
* spring-boot-starter-test v2.4.0
* jdk 1.8

## Javadoc funzionalità implementate

AdminBoundary

#### visualizzaElencoAbbonati

@GetMapping(value="/admin/abbonati")

java.lang.String visualizzaElencoAbbonati(@RequestParam(value="nome")

java.lang.String nome,

@RequestParam(value="cognome")

java.lang.String cognome,

@RequestParam(value="stato")

java.lang.String stato,

org.springframework.ui.Model model)

Visualizza elenco abbonati.

**Parameters:**

nome - nome

cognome - cognome

stato - stato abbonamento

model - model

**Returns:**

il nome della pagina html da renderizzare

AdminController

#### ricercaAbbonati

java.util.List<[Abbonato](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\entity\Abbonato.html)> ricercaAbbonati(java.lang.String nome,

java.lang.String cognome,

java.lang.String stato)

Ricerca abbonati.

**Parameters:**

nome - nome

cognome - cognome

stato - stato abbonamento

**Returns:**

the list

AbbonatoDAO

#### leggiAbbonati

java.util.List<[Abbonato](file:///D:\Users\libre\INGEGNERIA%20DEL%20SOFTWARE\canale_j_z-libre_team\software\source\keepfit\doc\it\keepfit\entity\Abbonato.html)> leggiAbbonati(java.lang.String nome,

java.lang.String cognome,

java.lang.String stato)

Leggi abbonati.

**Parameters:**

nome - nome

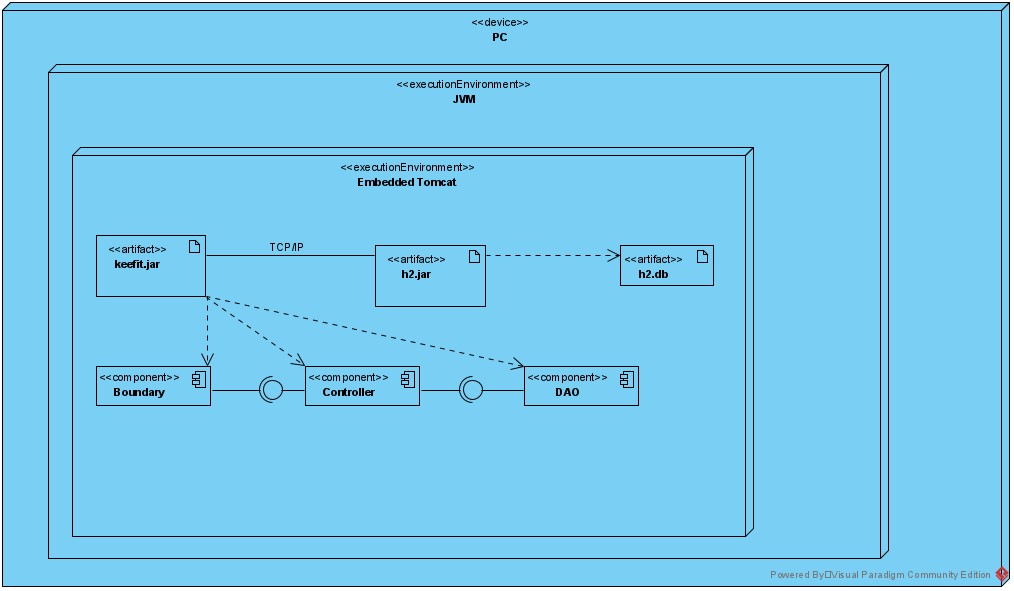
cognome - cognome

stato - stato abbonamento

**Returns:**

the list

## Diagramma di deploy



# Testing

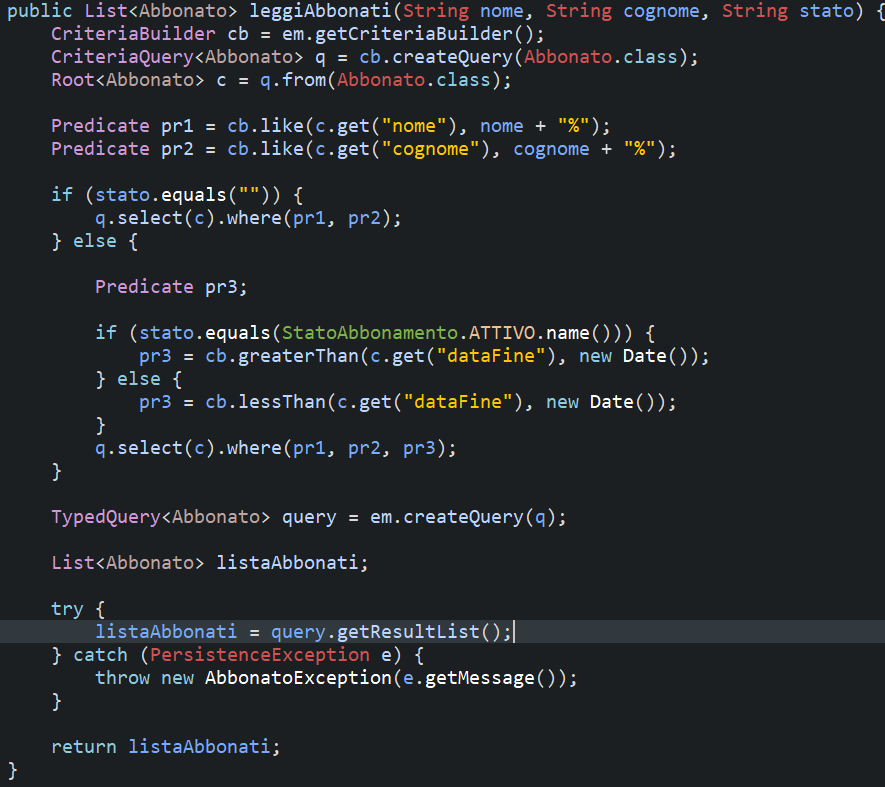
## Test funzionale

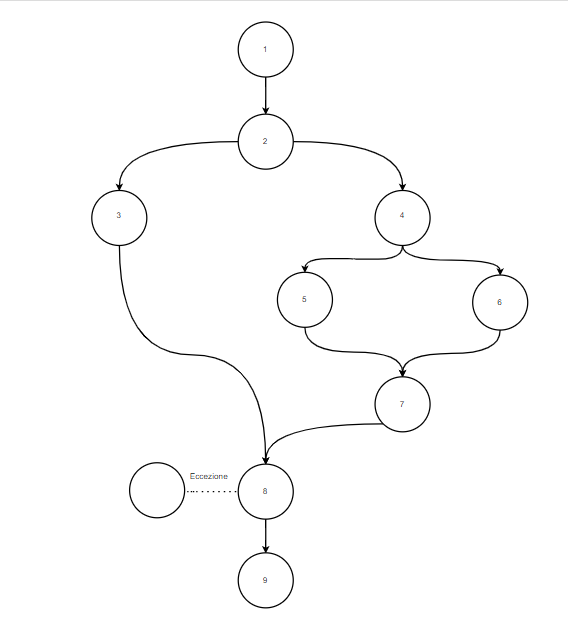
Descrivere i risultati dell’esecuzione dei test funzionali precedentemente pianificati come da tabella seguente, e le eventuali azioni di debugging.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** | **Output Ottenuti** | **Post-condizioni Ottenute** | **Esito**  **(*FAIL*, *PASS*)** |
| 1 |  | Tipo ricerca: COGNOME  Numero abbonati:3  Stato abbonamento: ENTRAMBI  Esito ricerca:0 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, SCADUTO]  } | Ricerca( “”, ” Esposito”, “”) | 0 risultati | N/A |  | N/A |  |
| 2 |  | Tipo ricerca: STATO ABBONAMENTO  Numero abbonati:3  Stato abbonamento: ENTRAMBI  Esito ricerca:0 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, ATTIVO]  } | Ricerca(“”, “”, “SCADUTO” ) | 0 risultati | N/A |  | N/A |  |
| 3 |  | Tipo ricerca: COGNOME  Numero abbonati:3  Stato abbonamento: ENTRAMBI  Esito ricerca:1 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, SCADUTO]  } | Ricerca( “”, “ Rossi”, “”) | 1 risultati | N/A |  | N/A |  |
| 4 |  | Tipo ricerca: STATO ABBONAMENTO  Numero abbonati:3  Stato abbonamento: ENTRAMBI  Esito ricerca:1 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, SCADUTO]  } | Ricerca(“”, “”, “ SCADUTO” ) | 1 risultati | N/A |  | N/A |  |
| 5 |  | Tipo ricerca: COGNOME  Numero abbonati:3  Stato abbonamento: ENTRAMBI  Esito ricerca:>1 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Rossi , ATTIVO],  [Andrea, Giallo, SCADUTO]  } | Ricerca(“”, “ Rossi”, “” ) | 2 risultati | N/A |  | N/A |  |
| 6 |  | Tipo ricerca: STATO ABBONAMENTO  Numero abbonati:4  Stato abbonamento: ENTRAMBI  Esito ricerca: >1 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, SCADUTO]  } | Ricerca(“”, “”, “ATTIVO” ) | 2 risultati | N/A |  | N/A |  |
| 7 |  | Tipo ricerca: STATO ABBONAMENTO  Numero abbonati:3  Stato abbonamento: ATTIVO  Esito ricerca:>1 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, ATTIVO]  } | Ricerca(“”, “”, “ ATTIVO ” ) | 3 risultati | N/A |  | N/A |  |
| 8 |  | Tipo ricerca: QUALSIASI  Numero abbonati:3  Stato abbonamento: SCADUTO  Esito ricerca:>1 | DB = {  [Mario, Rossi, SCADUTO ],  [Luca, Verde, SCADUTO ],  [Andrea, Giallo, SCADUTO ]  } | Ricerca(“”, “”, “ SCADUTO” ) | 3 risultati | N/A |  | N/A |  |
| 9 |  | Tipo ricerca: COGNOME  Numero abbonati:1  Stato abbonamento: ATTIVO  Esito ricerca:1 | DB = {  [Mario, Rossi, ATTIVO]  } | Ricerca( “”, “ Rossi”, “”) | 1 risultati | N/A |  | N/A |  |
| 10 |  | Tipo ricerca: STATO ABBONAMENTO  Numero abbonati:0  Stato abbonamento: N/A  Esito ricerca:0 | DB = {  } | Ricerca( “”, “”, “ ATTIVO”) | 0 risultati | N/A |  | N/A |  |
| 11 |  | Tipo ricerca: TUTTI I CAMPI VUOTI  Numero abbonati:2  Stato abbonamento: ENTRAMBI  Esito ricerca:2 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, SCADUTO],  } | Ricerca( “”, “”, “ ”) | 2 risultati | N/A |  | N/A |  |
| 12 |  | Tipo ricerca: NOME  Numero abbonati:3  Stato abbonamento:ENTRAMBI  Esito ricerca:1 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, SCADUTO]  } | Ricerca( “ Luca”, “”, “”) | 1 risultati | N/A |  | N/A |  |
| 13 |  | Tipo ricerca: NOME E COGNOME  Numero abbonati:3  Stato abbonamento:ENTRAMBI  Esito ricerca:1 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, SCADUTO]  } | Ricerca( “Mario”, “Rossi”, “”) | 1 risultati | N/A |  | N/A |  |
| 14 |  | Tipo ricerca: NOME E STATO ABBONAMENTO  Numero abbonati:3  Stato abbonamento:ENTRAMBI  Esito ricerca:0 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, SCADUTO]  } | Ricerca( “Andrea”, “”, “ATTIVO”) | 0 risultati | N/A |  | N/A |  |
| 15 |  | Tipo ricerca: COGNOME E STATO ABBONAMENTO  Numero abbonati:3  Stato abbonamento: ENTRAMBI  Esito ricerca:0 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, SCADUTO]  } | Ricerca( “”, “Verde”, “SCADUTO”) | 0 risultati | N/A |  | N/A |  |
| 16 |  | Tipo ricerca:NOME, COGNOME, STATO  Numero abbonati:3  Stato abbonamento: ENTRAMBI  Esito ricerca:0 | DB = {  [Mario, Rossi, ATTIVO],  [Luca, Verde, ATTIVO],  [Andrea, Giallo, SCADUTO]  } | Ricerca(“Mario”, “Esposito”, “ATTIVO” ) | 0 risultati | N/A |  | N/A |  |

## Test strutturale

## Complessità ciclomatica





Complessità ciclomatica: numero regioni del grafo = 3

I cammini indipendenti sono:

1. 1-2-3-8-9
2. 1-2-4-5-7-8-9
3. 1-2-4-6-7-8-9

## Test di unità

Test case cammino 1:

valore(stato) = “”

Test case cammino 2:

valore(stato) = “attivo”

Test case cammino 3:

valore(stato) = “scaduto”

## Test di unità con JUnit

Presentare i test di unità condotti con JUnit per una delle classi Java implementate (riportare la percentuale di copertura; utilizzare Javadoc per documentare la classe di test)