

# TEST

## CARLOAN

Versione 1.0

Data di rilascio:

INGEGNERIA DEL SOFTWARE A.A. 2014-2015

**Realizzato da**

Piemonte Christopher 587662 ITPS  
[chris.piemonte@hotmail.it](mailto:chris.piemonte@hotmail.it)

Saracino Paolo 592949 ITPS  
paolo.saracino.ps@gmail.com

---

# INDICE

---

## **1. ANALISI DEL TIPO DI TEST**

- 1.1 Calcolo del numero cicломatico
- 1.2 Report "Analisi del tipo di test"

## **2. Test in piccolo**

- 2.1 GestioneAccount
- 2.2 GestioneAgenzie
- 2.3 GestioneAuto
- 2.4 GestioneClienti
- 2.5 GestioneContratti
- 2.6 GestioneFasce
- 2.7 GestioneImpiegati

## **3. Test in grande**

- 3.1 SqlQueryFactory
- 3.2 ViewsJsonParser
- 3.3 DbPropertiesReader
- 3.4 BaseEncrypter
- 3.5 CalcoloTotale

---

## 1. ANALISI DEL TIPO DI TEST

---

### 1.1 Calcolo del Numero Ciclomatico

Il calcolo del numero ciclomatico per ciascun metodo di ogni classe è stato ottenuto utilizzando l'apposito plugin per Eclipse CodePro AnalytiX, e si trova nel file ReportCyclomaticComplexity.html.

### 1.2 Report “Analisi del tipo di Test”

- Non sono state testate classi relative all'utilizzo di JavaFX
- Non sono stati effettuati test su classi entità, come i Transfer Objects, o comunque privi di elaborazioni.
- I test funzionali sono stati effettuati considerando la corrispondenza del comportamento del Software con i requisiti, contenuti nel documento di Analisi e Specifica dei Requisiti.
- Gli unit test per le classi contenute in applicationServices comprendono i test per le classi contenute in businessObjects e per le classi contenute in mySqlDao.entity.

---

## 2. TEST IN PICCOLO

---

### 2.1 GestioneAccount

#### 2.1.1 login(String username, String password)

#### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
username, password	CE1	Credenziali valide	CE2	Username o password non corretti

#### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	username = "admin", password = "admin"	CE1
TC1	username = "test", password = "errato"	CE2

#### ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
1	TC0	ImpiegatoTO	ImpiegatoTO
2	TC1	null	null

---

### 2.1.2 addNewAccount(AccountTO account)

#### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
account	CE1	Non esiste	CE2	esiste

#### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	username = "test", password = "0000", ruolo = "impiegato"	CE1
TC1	username = "admin", password = "admin", ruolo = "admin"	CE2

#### ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
3	TC0	true	true
4	TC1	False	false

### 2.1.3 getAccount(String username)

#### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
username	CE1	Esiste	CE2	Non esiste

---

## PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	username = "admin"	CE1
TC1	username = "test"	CE2

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
5	TC0	ImpiegatoTO	ImpiegatoTO
6	TC1	null	null

### 2.1.4 getPasswordOf(String username)

## CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
username	CE1	Esiste	CE2	Non esiste

## PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	username = "admin"	CE1
TC1	username = "test"	CE2

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
7	TC0	"admin"	"admin"
8	TC1	null	null

---

### 2.1.5 update(AccountTO account)

#### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
account	CE1	Esiste	CE2	Non esiste

#### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	username = "admin", password = "0000", ruolo = "impiegato"	CE1
TC1	username = "test", password = "0000", ruolo = "impiegato"	CE2

#### ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
9	TC0	true	true
10	TC1	false	false

### 2.1.6 exists(String username)

#### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
username	CE1	Esiste	CE2	Non esiste

---

## PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	username = "admin"	CE1
TC1	username = "test"	CE2

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
11	TC0	true	true
12	TC1	false	false

## 2.2 GestioneAgenzie

### 2.2.1 addNewAgenzia(AgenziaTO agenzia)

## PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	citta = "Torino", indirizzo = "via orabona", telefono = "080012345"	

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
13	TC0	true	true



---

### 2.2.2 getAgenzia(String id)

#### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
id	CE1	Esiste	CE2	Non esiste

#### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	id = "1"	CE1
TC1	id = "70"	CE2

#### ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
14	TC0	AgenziaTO	AgenziaTO
15	TC1	null	null

### 2.2.3 getManagerOf(String id)

#### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
id	CE1	Esiste	CE2	Non esiste

#### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	id = "1"	CE1
TC1	id = "70"	CE2

#### ESECUZIONE CASI DI TEST

---

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
16	TC0	ManagerTO	ManagerTO
17	TC1	null	null

#### 2.2.4 getImpiegatiOf(String id)

#### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
id	CE1	Esiste	CE2	Non esiste

#### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	id = "1"	CE1
TC1	id = "70"	CE2

#### ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
18	TC0	List<ImpiegatoTO>	List<ImpiegatoTO>
19	TC1	null	null

---

## 2.2.5 update(AgenziaTO agenzia)

### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
agenzia	CE1	Esiste	CE2	Non esiste

### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	id = "1"	CE1
TC1	id = "70"	CE2

### ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
20	TC0	true	true
21	TC1	false	false

## 2.2.6 deleteCitta(String citta)

### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
citta	CE1	Esiste	CE2	Non esiste

### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	citta = "Torino"	CE1
TC1	citta = "Londra"	CE2

---

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
20	TC0	true	true
21	TC1	false	false

### 2.2.7 exists(String id)

## CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
id	CE1	Esiste	CE2	Non esiste

## PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	id = "1"	CE1
TC1	id = "70"	CE2

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
22	TC0	true	true
23	TC1	false	false

---

## 2.3 GestioneAuto

### 2.3.1 addNewAuto(AutoTO auto)

#### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
auto	CE1	Non esiste	CE2	Esiste

#### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	targa = "qq444qq", modello = "Reanult Twingo", stato = "D", fascia= "A", km = "via orabona", manutenzioneOrdinaria = <i>Date</i> , agenzia = "1"	CE1
TC1	targa = "qq444qq", modello = "Reanult Twingo", stato = "D", fascia= "A", km = "via orabona", manutenzioneOrdinaria = <i>Date</i> , agenzia = "1"	CE2

#### ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
24	TC0	true	true
25	TC1	false	false

---

### 2.3.2 getAuto(String targa)

#### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
targa	CE1	Esiste	CE2	Non esiste

#### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	targa = "kj088fv"	CE1
TC1	targa = "qq444qq"	CE2

#### ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
26	TC0	true	true
27	TC1	False	false

### 2.3.3 update(AutoTO auto)

#### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
auto	CE1	Esiste	CE2	Non esiste

---

## PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	targa = "kj088fv", modello = "Honda Civic", stato = "D", fascia= "C", km = "870", manutenzioneOrdinaria = 2016-03-14, agenzia = "4"	CE1
TC1	targa = "qq444qq", modello = "Reanult Twingo", stato = "D", fascia= "A", km = "via orabona", manutenzioneOrdinaria = Date, agenzia = "1"	CE2

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
28	TC0	true	true
29	TC1	false	false

### 2.3.4 exists(String targa)

## CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
targa	CE1	Esiste	CE2	Non esiste

---

## PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	targa = "kj088fv"	CE1
TC1	targa = "qq444qq"	CE2

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
30	TC0	true	true
31	TC1	False	false

### 2.3.5 changeStatoOf(String targa, String newStato)

## CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
targa	CE1	Esiste	CE2	Non esiste
newStato	CE3	D	CE6	G
	CE4	M	CE7	
	CE5	N	CE8	

## PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	targa = "kj088fv", newStato = "D"	CE1, CE3
TC1	targa = "kj088fv", newStato = "G"	CE1, CE6
TC2	targa = "qq444qq", newStato = "N"	CE2, CE5
TC3	targa = "qq444qq", newStato = "G"	CE2, CE6



---

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
32	TC0	true	true
33	TC1	false	false
34	TC2	false	false
35	TC3	false	false

### 2.3.6 changeKmOf(String targa, double newKm)

## CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
targa	CE1	Esiste	CE2	Non esiste
newKm	CE3	newKm >= 0	CE4	newKm < 0

## PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	targa = "kj088fv", newKm = "37"	CE1, CE3
TC1	targa = "kj088fv", newKm = "-37"	CE1, CE4
TC2	targa = "qq444qq", newKm = "0"	CE2, CE3

---

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
36	TC0	true	true
37	TC1	false	false
38	TC2	false	false

## 2.4 GestioneClienti

### 2.4.1 addNewCliente(ClienteTO cliente)

## CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
cliente	CE1	Non esiste	CE2	Esiste

## PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	cf = "AAAAAAAAAAAAAAAA", nome = "Renato", cognome = "Renati", dataNascita = <i>Date</i> , telefono = "0801010101"	CE1
TC1	cf = "AAAAAAAAAAAAAAAA", nome = "Renato", cognome = "Renati", dataNascita = <i>Date</i> , telefono = "0801010101"	CE2

---

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
39	TC0	true	true
40	TC1	false	false

### 2.4.2 getClient(String cf)

## CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
cliente	CE1	Esiste	CE2	Non Esiste

## PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	cf = "FRANZS13A01D447S"	CE1
TC1	cf = "AAAAAAAAAAAAAAAAAA"	CE2

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
41	TC0	true	true
42	TC1	false	false

### 2.4.3 update(ClienteTO cliente)

## CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
cliente	CE1	Esiste	CE2	Non esiste

## PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	cf = "FRANZS13A01D447S", nome = "Francesca", cognome = "Neri", dataNascita = 1953-03-14, telefono = "08090111"	CE1
TC1	cf = "AAAAAAAAAAAAAAAA", nome = "Renato", cognome = "Renati", dataNascita = Date, telefono = "0801010101"	CE2

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
43	TC0	true	true
44	TC1	false	false

### 2.4.4 exists(String cf)

## CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
cf	CE1	Esiste	CE2	Non esiste

## PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	cf = "FRANZS13A01D447S"	CE1
TC1	cf = "AAAAAAAAAAAAAAAA"	CE2

---

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
45	TC0	true	true
46	TC1	false	false

## 2.5 GestioneContratti

### 2.5.1 getContratto(String id)

## CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
id	CE1	Esiste	CE2	Non esiste

## PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	cf = "1"	CE1
TC1	cf = "100"	CE2

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
47	TC0	true	true
48	TC1	false	false

---

## 2.5.2 exists(String id)

### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
id	CE1	Esiste	CE2	Non esiste

### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	cf = "1"	CE1
TC1	cf = "100"	CE2

### ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
49	TC0	true	true
50	TC1	false	false

## 2.5.3 changeModNoleggio(String id, String newModNoleggio)

### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
id	CE1	Esiste	CE2	Non esiste
newModNoleggio	CE3 CE4	giornaliero settimanale	CE5	mensile

---

## PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	Id = "1", newModNoleggio = "giornaliero"	CE1, CE3
TC1	Id = "1", newModNoleggio = "mensile"	CE1, CE5
TC2	Id = "100", newModNoleggio = "giornaliero"	CE2, CE3

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
51	TC0	true	true
52	TC1	false	false
53	TC2	false	false

### 2.5.4 changeDataFine(String id, newDataFine)

## CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
id	CE1	Esiste	CE2	Non esiste

## PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	Id = "1"	CE1
TC1	Id = "100"	CE2

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
54	TC0	true	true
55	TC1	false	false

### 2.5.5 changeAgenziaFine(String id, String newAgenziaFine)

## CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
id	CE1	Esiste	CE2	Non esiste
newAgenziaFine	CE3	Esiste	CE4	Non esiste

## PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	id = "1", newAgenziaFine = "1"	CE1, CE3
TC1	id = "1", newAgenziaFine = "70"	CE1, CE4
TC3	id = "100", newAgenziaFine = "1"	CE2, CE3

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
56	TC0	true	true
57	TC1	false	false
58	TC3	false	false



---

## 2.5.6 chiudiContratto(String id, String impiegatoFine, double kmPercorsi, double totale)

### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
id	CE1	Esiste	CE2	Non esiste
impiegatoFine	CE3	Esiste	CE4	Non esiste
kmPercorsi	CE5	kmPercorsi < 0	CE6	kmPercorsi > 0
totale	CE7	totale > 0	CE8	totale < 0

### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	Id = "1", impiegatoFine = "RBRBRS13G71F145H", kmPercorsi = 10 totale = 10	CE1, CE3, CE5, CE7
TC1	Id = "100", impiegatoFine = "RBRBRS13G71F145H", kmPercorsi = 10 totale = 10	
TC2	Id = "1", impiegatoFine = "BBBBBBBBBIBBBBBBB", kmPercorsi = 10 totale = 10	
TC3	Id = "1", impiegatoFine = "RBRBRS13G71F145H", kmPercorsi = -10 totale = 10	
TC4	Id = "1", impiegatoFine = "RBRBRS13G71F145H", kmPercorsi = 10 totale = -10	

TC5	cf = "100"Id = "1", impiegatoFine = "RBRBRS13G71F145H", kmPercorsi = 10 totale = 10	CE2
-----	--	-----

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
59	TC0	true	true
60	TC1	false	false
61	TC2	false	false
62	TC3	false	false
63	TC4	false	false
64	TC5	false	false

## 2.6 GestioneFasce

### 2.6.1 addNewFascia(String id)

## CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
id	CE1	Non esiste	CE2	Esiste

## PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	id = "Z"	CE1
TC1	id = "Z"	CE2

---

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
65	TC0	true	true
66	TC1	false	false

### 2.6.2 getFascia(String id)

## CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
id	CE1	Esiste	CE2	Non esiste

## PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	id = "A"	CE1
TC1	id = "Z"	CE2

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
67	TC0	true	true
68	TC1	false	false

---

### 2.6.3 update(FasciaTo fascia)

#### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
fascia	CE1	Esiste	CE2	Non esiste

#### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	id = "A", descrizione = "Test", tariffaGiornaliera = 10, tariffaSetimanale = 10, tariffaKm = 10	CE1
TC1	id = "Z", descrizione = "Test", tariffaGiornaliera = 10, tariffaSetimanale = 10, tariffaKm = 10	CE2

#### ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
69	TC0	true	true
70	TC1	false	false

---

## 2.6.4 exists(FasciaTo fascia)

### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
fascia	CE1	Esiste	CE2	Non esiste

### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	id = "A"	CE1
TC1	id = "Z"	CE2

### ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
71	TC0	true	true
72	TC1	false	false

## 2.7 GestioneImpiegati

### 2.7.1 addNewImpiegato(ImpiegatoTO impiegato)

### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
impiegato	CE1	Non esiste	CE2	Esiste

---

## PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	cf = "BIBIBIBIBIBIBIBI", nome = "Berto", cognome = "Bibi", dataNascita = Date, telefono = "0101010101", agenzia = "1", username = "bibì"	CE1
TC1	cf = "BIBIBIBIBIBIBIBIBI", nome = "Berto", cognome = "Bibi", dataNascita = Date, telefono = "0101010101", agenzia = "1", username = "bibì"	CE2

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
73	TC0	true	true
74	TC1	false	false

---

## 2.7.2 getImpiegato(String cf)

### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
cf	CE1	Esiste	CE2	Non esiste

### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	cf = "ABRBRS13G71F145H"	CE1
TC1	cf = "BIBIBIBIBIBIBIBI"	CE2

### ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
75	TC0	true	true
76	TC1	false	false

---

### 2.7.3 update(ImpiegatoTO impiegato)

#### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
impiegato	CE1	Esiste	CE2	Non esiste

#### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	cf = "ABRBRS13G71F145H", nome = "Alberto", cognome = "Baresi", dataNascita = Date, telefono = "08078555", agenzia = "3", username = "rol001"	CE1
TC1	cf = "BIBIBIBIBIBIBIBI", nome = "Berto", cognome = "Bibi", dataNascita = Date, telefono = "0101010101", agenzia = "1", username = "bibì"	CE2

#### ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
77	TC0	true	true
78	TC1	false	false



---

## 2.7.4 delete(String cf)

### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
cf	CE1	Esiste	CE2	Non esiste

### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	cf = "BIBIBIBIBIBIBIBI"	CE1
TC1	cf = "IKIKIKIKIKIKIKIK"	CE2

### ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
79	TC0	true	true
80	TC1	false	false

## 2.7.5 exists(String cf)

### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
cf	CE1	Esiste	CE2	Non esiste

### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	cf = "ABRBRS13G71F145H"	CE1
TC1	cf = "IKIKIKIKIKIKIKIK"	CE2

---

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
81	TC0	true	true
82	TC1	false	false

---

## 3. TEST IN GRANDE

---

### 3.1 SqlQueryFactory

#### 3.1.1 getQuery(String id)

#### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
id	CE1	Esiste	CE2	Non esiste

#### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	id = "create_agenzia"	CE1
TC1	cf = "test"	CE2

#### ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
83	TC0	INSERT INTO `carloan`.`agenzia` `(`citta`, `indirizzo`, `telefono`, `stato`) VALUES (?, ?, ?, ?);	INSERT INTO `carloan`.`agenzia` `(`citta`, `indirizzo`, `telefono`, `stato`) VALUES (?, ?, ?, ?);
84	TC1	null	null

---

## 3.2 ViewsJsonParser

### 3.2.1 getViewPath(String id)

#### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
id	CE1	Esiste	CE2	Non esiste

#### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	id = "bootstrap"	CE1
TC1	id = "test"	CE2

#### ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
85	TC0	/main/ids/presentation/view/Bootstrap.fxml	/main/ids/presentation/view/Bootstrap.fxml
86	TC1	null	null

---

### 3.3 DbPropertiesReader

#### 3.3.1 getProperty(String id)

##### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
id	CE1	Esiste	CE2	Non esiste

##### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	id = "user"	CE1
TC1	id = "test"	CE2

##### ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
87	TC0	root	root
88	TC1	null	null

### 3.4 BaseEncrypter

#### 3.4.1 decrypt(String encrypted)

##### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	encrypted = "Uf0xQnkC8ZDuKO4bMx9Oqg=="	CE1

##### ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
89	TC0	manager	manager

---

## 3.5 CalcoloTotale

### 3.5.1 setTotale(LocalDate inizio, LocalDate fine, double tariffaBase, double km, String kmNoleggio)

#### CLASSI DI EQUIVALENZA

Condizioni Esterne	Classi di Equivalenza			
	CE	Valide	CE	Non Valide
inizio	CE1	Non è null	CE2	null
fine	CE3	Non è null	CE4	null
kmNoleggio	CE5 CE6	limitato Illimitato	CE7	super

#### PROGETTAZIONE CASI DI TEST

TC	DATI	CLASSI COPERTE
TC0	inizio = 2015-11-19, fine = 2015-11-19, tariffaBase = 10, km = 30, kmNoleggio = "limitato"	CE1, CE3, CE5
TC1	inizio = null, fine = 2015-11-19, tariffaBase = 10, km = 30, kmNoleggio = "illimitato"	CE2, CE3, CE5
TC2	inizio = 2015-11-19, fine = null, tariffaBase = 10, km = 30, kmNoleggio = "illimitato"	CE1, CE4, CE5
TC3	inizio = 2015-11-19, fine = 2015-11-19, tariffaBase = 10, km = 30, kmNoleggio = "super"	CE1, CE3, CE7

---

## ESECUZIONE CASI DI TEST

ID	INPUT	VALORE ATTESO	VALORE OTTENUTO
90	TC0	300	300
91	TC1	0	0
92	TC2	0	0
93	TC3	0	0

---

### **3. TEST DI INTEGRAZIONE (NON APPLICABILE)**

---