

# Università degli studi di Salerno

Corso di Ingegneria del Software

## Core Shirt Object Design Document



CORE SHIRT  
The Best T-Shirt Series

Data: 14/10/2016

### Partecipanti:

Nome	Matricola
Ranieri Salvatore	0512103286
Russo Gennaro	0512103160
Turi Pasquale	0512102860
Urlo Mario Pio	0512103133

# Indice

1. Introduzione
  - 1.1. Obiettivi di design trade-off
  - 1.2. Linee guida per l'interfaccia
  - 1.3. Definizioni, acronimi e abbreviazioni
  - 1.4. Riferimenti
2. Packages
  - 2.1 Package model
  - 2.2 Package control
3. Interfacce delle classi
  - 3.1 Classe Dipendente
  - 3.2 Classe Articolo
  - 3.3 Classe TShirt
  - 3.4 Classe Ordine
  - 3.5 Classe Fattura
  - 3.6 Classe Spesa
  - 3.7 Classe Carrello
  - 3.8 Classe Cliente
  - 3.9 Classe ManageArticolo
  - 3.10 Classe ManageCarrello
  - 3.11 Classe MangeDipendente
  - 3.12 Classe ManageFattura
  - 3.13 Classe ManageOrdine
  - 3.14 Classe ManageRifornimento
  - 3.15 Classe ManageSpesa
  - 3.16 Classe ManageTShirt
  - 3.17 Classe ManageStipendi
  - 3.18 Classe ManageTotale
  - 3.19 Classe Controller
  - 3.20 Classe FlussoEconomico
  - 3.21 Classe GestisciArticolo
  - 3.22 Classe ServletTshirt
  - 3.23 Classe GestisciDipendente
  - 3.24 Classe ServletLogin
  - 3.25 Classe DbConnect
  - 3.26 Classe ServletInsertSpesa
  - 3.27 Classe GestisciOrdine
  - 3.28 Classe ServletRifornimento

- 3.29 Classe ServletInsertArticolo
- 3.30 Classe ServletInsertDipendente
- 3.31 Classe PagaStipendi

# 1. Introduzione

## 1.1 Obiettivi di design Trade-off

Dopo la realizzazione del documento RAD (Requirement Analysis Document) e SDD (System Design Document), abbiamo descritto in linea di massima, quello che sarà il nostro sistema e quindi i nostri obiettivi, tralasciando gli aspetti implementativi.

### Prestazioni vs. Costi

Considerando il sistema che stiamo realizzando, possiamo dire che il non eccessivo budget a nostra disposizione ci ha consentito di realizzare il prodotto utilizzando materiale open source partendo da zero minimizzando così i costi e rendendo il sistema più che soddisfacente.

### Interfaccia vs. Tempo di risposta

Il tempo di risposta tra server e interfaccia sono più che sufficienti (rapidi) a soddisfare le esigenze dei vari dipendenti e clienti collegati al Sistema.

Ovviamente maggiore sarà la grandezza del database e maggiore sarà il tempo di risposta e ricerca nel database.

### Interfaccia vs. Easy-use

L'interfaccia, grazie a form e ad una impostazione semplice e intuitiva, permette un facile utilizzo del sistema.

### Costi vs. Mantenimento

Grazie a un uso di materiale open source e l'utilizzo del linguaggio javadoc il sistema può essere facilmente modificato, implementato con nuove funzioni o corretto in presenza di errori.

## 1.2 Linee guida per l'interfaccia

Il sistema è multi-utente (può accedervi chiunque, sia un semplice cliente o un dipendente).

Al cliente, il sistema nasconde la logica delle operazioni, fornendogli solamente la consultazione del catalogo dei prodotti e la possibilità di acquistarli.

Il dipendente invece deve avere accesso all'interfaccia magazzino, contabilità e gestione dipendenti.

Il dipendente riceve i risultati delle richieste effettuate dal sistema, senza preoccuparsi delle operazioni e di come queste sono state effettuate. Avvenuta l'operazione di vendita di uno o più prodotti il sistema decrementa dal DataBase totale il numero delle quantità vendute del prodotto acquistato (aggiorna quindi il DataBase totale).

## **1.3 Definizioni, acronimi e abbreviazioni**

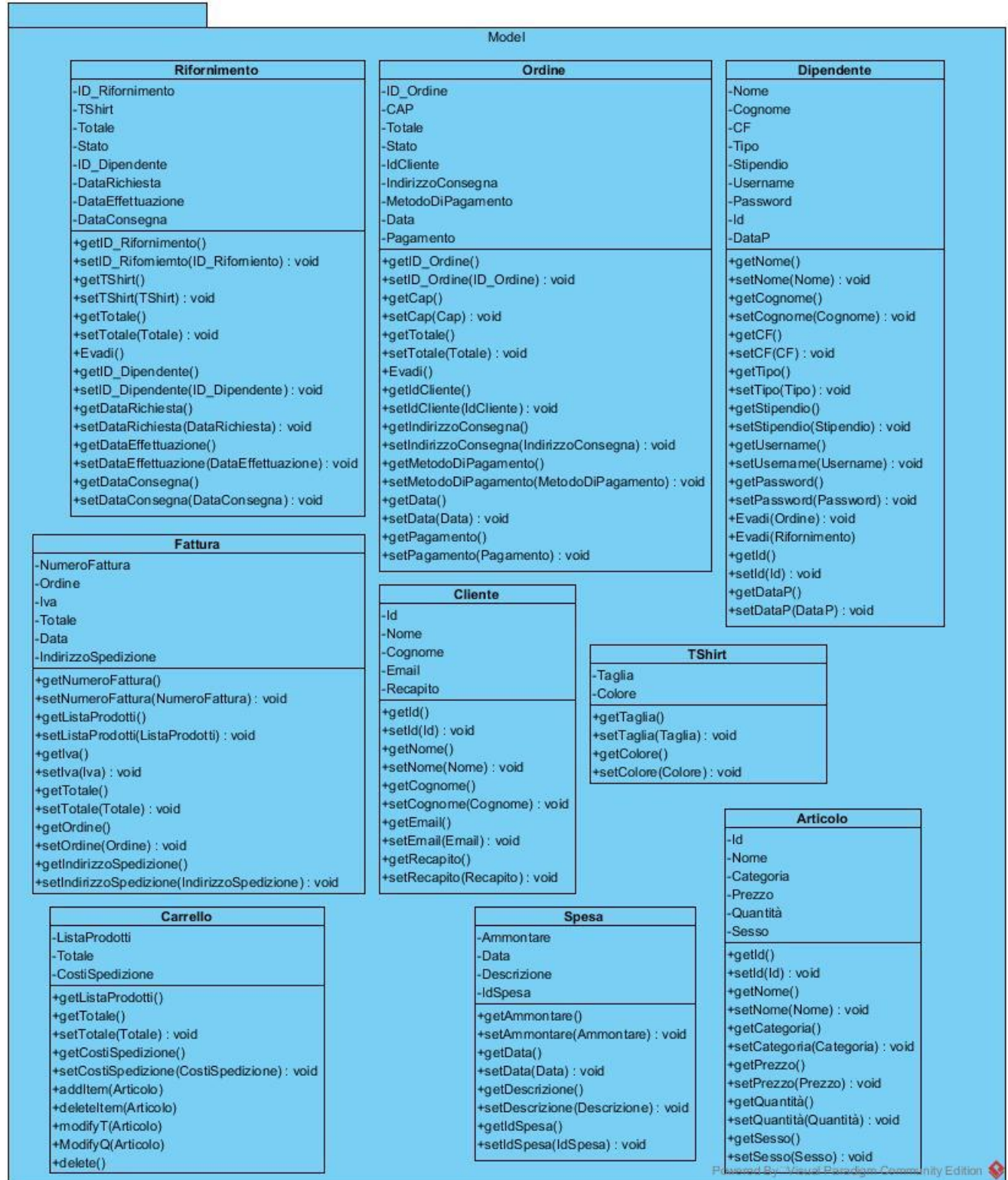
Non ci sono acronimi, definizioni e abbreviazioni.

## **1.4 Riferimenti**

- Sommerville, Software Engineering (6th edition, 2001), Addison Wesley.
- Roger S. Pressman, Principi di Ingegneria del Software (terza edizione, 2000), Mc Graw Hill Italia.

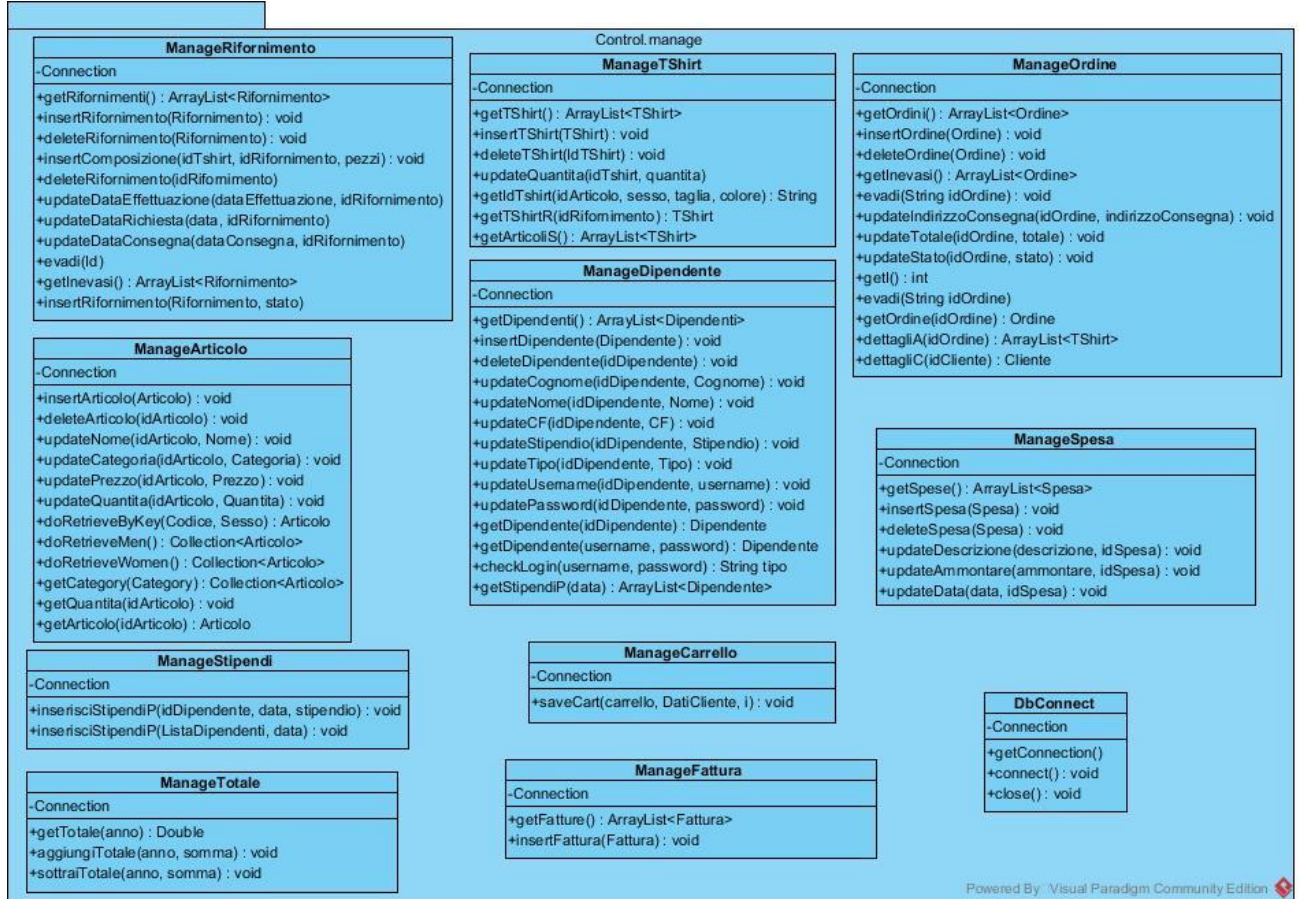
## 2. Packages

### 2.1 Package Model

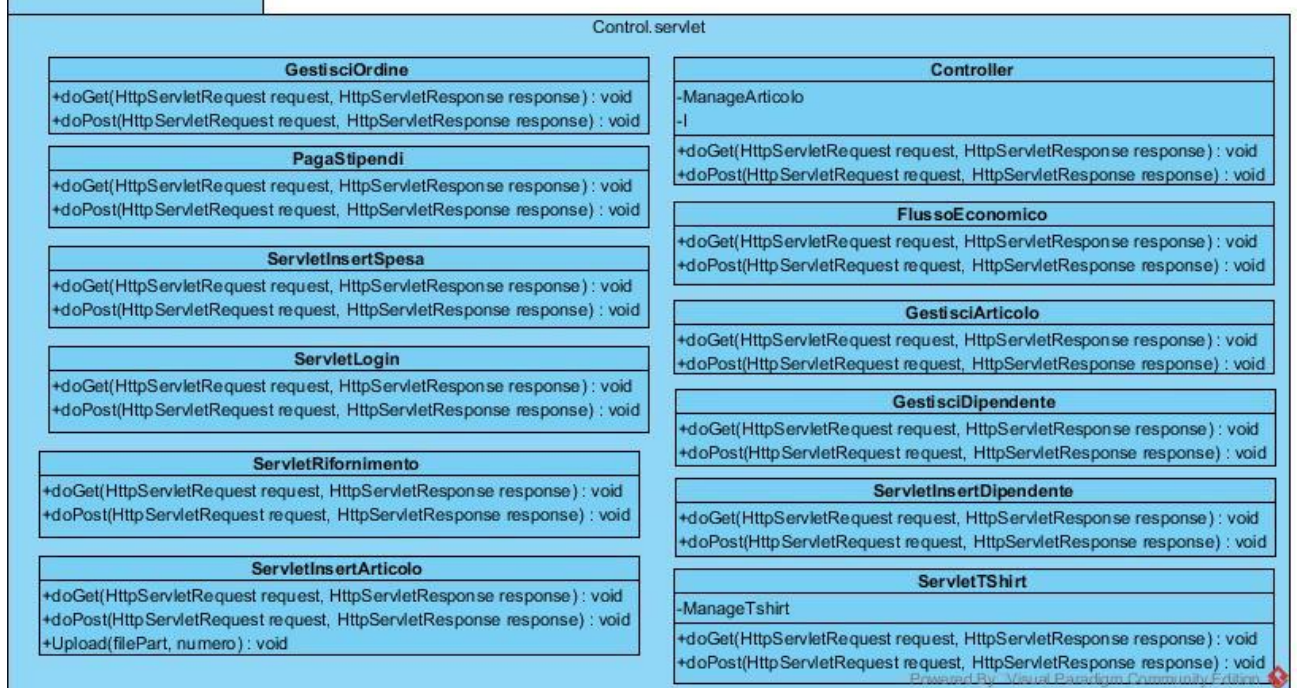




## 2.2 Package Control.Manage



## 2.3 Package Control.Servlet



### 3. Interfacce delle classi

#### 3.1 Classe Dipendente

Nome Classe	Dipendente
Descrizione	Questa classe rappresenta l'astrazione di un dipendente
Pre-Condizioni	<p><b>context</b> Dipendente : : getNome() pre: Nome != null</p> <p><b>context</b> Dipendente : : setNome(Nome) pre: Nome != null</p> <p><b>context</b> Dipendente : : getCognome() pre: Cognome != null</p> <p><b>context</b> Dipendente : : setCognome(Cognome) pre: Cognome != null</p> <p><b>context</b> Dipendente : : getCodiceFiscale() pre: CodiceFiscale != null</p> <p><b>context</b> Dipendente : : setCodiceFiscale(CodiceFiscale) pre: CodiceFiscale.lenght()=16</p> <p><b>context</b> Dipendente : : getId() pre: Id &gt; 0</p> <p><b>context</b> Dipendente : : setId(Id) pre: Id &gt; 0</p> <p><b>context</b> Dipendente : : getStipendio() pre: Stipendio != null</p> <p><b>context</b> Dipendente : : setStipendio(Stipendio) pre: 1000 &lt; Stipendio &lt; 2000</p> <p><b>context</b> Dipendente : : getUsername() pre: Username != null</p> <p><b>context</b> Dipendente : : setUsername(Username) pre: Username != null</p> <p><b>context</b> Dipendente : : getPassword() pre: Password != null</p> <p><b>context</b> Dipendente : : setPassword&gt;Password) pre: Password != null</p> <p><b>context</b> Dipendente : : getDataP() pre: dataP != null</p> <p><b>context</b> Dipendente : : setDataP(data) pre: data != null</p> <p><b>context</b> Dipendente : : toString2() pre:</p>



Post-Condizioni	<p><b>context</b> Dipendente : : setName(Nome) post: Nome != null</p> <p><b>context</b> Dipendente : : setCognome(Cognome) post: Cognome != null</p> <p><b>context</b> Dipendente : : setCodiceFiscale(CodiceFiscale) post: CodiceFiscale != null</p> <p><b>context</b> Dipendente : : setId(Id) post: Id &gt; 0</p> <p><b>context</b> Dipendente : : setStipendio(Stipendio) post: 1000 &lt; Stipendio &lt; 2000</p> <p><b>context</b> Dipendente : : setUsername(Username) post: Username != null</p> <p><b>context</b> Dipendente : : setPassword&gt;Password) post: Password != null</p> <p><b>context</b> Dipendente : : setDataP(data) post: dataP = data</p> <p><b>context</b> Dipendente : : toString2() post: risultato != null</p>
Invarianti	<p><b>context</b> Dipendente <b>inv</b>: Nome != null and Cognome != null and CodiceFiscale != null and Id &gt; 0 and 1000 &lt; Stipendio &lt; 2000 and Username != null and Password != null</p>

### 3.2 Classe Articolo

Nome Classe	Articolo
Descrizione	Questa classe rappresenta l'astrazione di un articolo.
Pre-Condizioni	<p><b>context</b> Articolo : : getName() pre: Nome != null</p> <p><b>context</b> Articolo : : setName(Nome) pre: Nome != null</p> <p><b>context</b> Articolo : : getCategory() pre: Categoria != null</p> <p><b>context</b> Articolo : : setCategoria(Categoria) pre: Categoria != null</p> <p><b>context</b> Articolo : : getSesso() pre: Sesso != null</p> <p><b>context</b> Articolo : : setSesso(Sesso) pre: Sesso != null</p> <p><b>context</b> Articolo : : getIdArticolo() pre: IdArticolo &gt; 0</p> <p><b>context</b> Articolo : : setIdArticolo(IdArticolo) pre: IdArticolo &gt; 0</p> <p><b>context</b> Articolo : : getPrezzo() pre: Prezzo &gt; 0</p> <p><b>context</b> Articolo : : setPrezzo(Prezzo) pre: Prezzo &gt; 0</p>

	<b>context</b> Articolo : : getQuantità() pre: Quantità > 0 <b>context</b> Articolo : : setQuantità(Quantità) pre: Quantità > 0
Post-Condizioni	<b>context</b> Articolo : : setNome(Nome) post: Nome != null <b>context</b> Articolo : : setCategoria(Categoria) post: Categoria != null <b>context</b> Articolo : : setSesso(Sesso) post: Sesso != null <b>context</b> Articolo : : setIdArticolo(IdArticolo) post: IdArticolo > 0 <b>context</b> Articolo : : setPrezzo(Prezzo) post: Prezzo > 0 <b>context</b> Articolo : : setQuantità(Quantità) post: Quantità > 0
Invarianti	<b>context</b> Articolo <b>inv:</b> Nome != null and Categoria != null and Sesso != null and IdArticolo > 0 and Prezzo > 0 and Quantità > 0

### 3.3 Classe TShirt

Nome Classe	TShirt
Descrizione	Questa classe rappresenta l'astrazione di una t-shirt.
Pre-Condizioni	<b>context</b> TShirt : : getTaglia() pre: Taglia != null <b>context</b> TShirt : : setTaglia(Taglia) pre: Taglia != null <b>context</b> TShirt : : getColore() pre: Colore != null <b>context</b> TShirt : : setColore(Colore) pre: Colore != null
Post-Condizioni	<b>context</b> TShirt : : setTaglia(Taglia) post: Taglia != null <b>context</b> TShirt : : setColore(Colore) post: Colore != null
Invarianti	<b>context</b> TShirt <b>inv:</b> Nome != null and Categoria != null and Sesso != null and IdArticolo > 0 and Prezzo > 0 and Quantità > 0 and Taglia != null and Colore != null

### 3.4 Classe Ordine

Nome Classe	Ordine
Descrizione	Questa classe rappresenta l'astrazione di un ordine.
Pre-Condizioni	<b>context</b> Ordine : : getData() pre: Data != null <b>context</b> Ordine : : setData(Data) pre: Data != null

	<p><b>context</b> Ordine : : getMetodoPagamento() pre: MetodoPagamento != null</p> <p><b>context</b> Ordine : : setMetodoPagamento(MetodoPagamento) pre: MetodoPagamento != null</p> <p><b>context</b> Ordine : : getStato() pre: Stato != null</p> <p><b>context</b> Ordine : : setStato(Stato) pre: Stato != null</p> <p><b>context</b> Ordine : : getIdOrdine() pre: IdOrdine &gt; 0</p> <p><b>context</b> Ordine : : setIdOrdine(IdOrdine) pre: IdOrdine &gt; 0</p> <p><b>context</b> Ordine : : getTotale() pre: Totale &gt; 0</p> <p><b>context</b> Ordine : : setTotale(Totale) pre: Totale &gt; 0</p> <p><b>context</b> Ordine : : getIdCliente() pre: IdCliente &gt; 0</p> <p><b>context</b> Ordine : : setIdCliente(IdCliente) pre: IdCliente &gt; 0</p> <p><b>context</b> Ordine : : getCap() pre: Cap != null</p> <p><b>context</b> Ordine : : setCap(Cap) pre: Cap != null</p> <p><b>context</b> Ordine : : getIndirizzoConsegna() pre: IndirizzoConsegna != null</p> <p><b>context</b> Ordine : : setIndirizzoConsegna(IndirizzoConsegna) pre: IndirizzoConsegna != null</p>
Post-Condizioni	<p><b>context</b> Ordine : : setData(Data) post: Data != null</p> <p><b>context</b> Ordine : : setMetodoPagamento(MetodoPagamento) post: MetodoPagamento != null</p> <p><b>context</b> Ordine : : setStato(Stato) post: Stato != null</p> <p><b>context</b> Ordine : : setIdOrdine(IdOrdine) post: IdOrdine &gt; 0</p> <p><b>context</b> Ordine : : setTotale(Totale) post: Totale &gt; 0</p> <p><b>context</b> Ordine : : setIdCliente(IdCliente) post: IdCliente &gt; 0</p> <p><b>context</b> Ordine : : setCap(Cap) post: Cap != null</p> <p><b>context</b> Ordine : : setIndirizzoConsegna(IndirizzoConsegna) post: IndirizzoConsegna != null</p>

Invarianti	<b>context</b> Ordine <b>inv:</b> Data != null and MetodoPagamento != null and Stato != null and IdOrdine > 0 and Totale > 0 and IdCliente > 0 and IndirizzoConsegna != null and Cap != null
------------	--

### 3.5 Classe Fattura

Nome Classe	Fattura
Descrizione	Questa classe rappresenta l'astrazione di una fattura.
Pre-Condizioni	<b>context</b> Fattura : : getData() pre: Data != null <b>context</b> Fattura : : setData(Data) pre: Data != null <b>context</b> Fattura : : getOrdine() pre: Ordine != null <b>context</b> Fattura : : setOrdine(Ordine) pre: Ordine != null <b>context</b> Fattura : : getIva() pre: Iva > 0 <b>context</b> Fattura : : setIva(Iva) pre: Iva > 0 <b>context</b> Fattura : : getTotale() pre: Totale > 0 <b>context</b> Fattura : : setTotale(Totale) pre: Totale > 0 <b>context</b> Fattura : : getIdFattura() pre: IdFattura > 0 <b>context</b> Fattura : : setIdFattura(IdFattura) pre: IdFattura > 0 <b>context</b> Fattura : : getIndirizzoSpedizione() pre: IndirizzoSpedizione != null <b>context</b> Ordine : : setIndirizzoSpedizione(IndirizzoSpedizione) pre: IndirizzoSpedizione != null
Post-Condizioni	<b>context</b> Fattura : : setData(Data) pre: Data != null <b>context</b> Fattura : : setOrdine(Ordine) pre: Ordine != null <b>context</b> Fattura : : setIva(Iva) pre: Iva > 0 <b>context</b> Fattura : : setTotale(Totale) pre: Totale > 0 <b>context</b> Fattura : : setIdFattura(IdFattura) pre: IdFattura > 0 <b>context</b> Ordine : : setIndirizzoSpedizione(IndirizzoSpedizione) pre: IndirizzoSpedizione != null
Invarianti	<b>context</b> Fattura <b>inv:</b> Data != null and Ordine != null and Iva > 0 and Totale > 0 and IdFattura > 0 and IndirizzoSpedizione != null

### 3.6 Classe Spesa

Nome Classe	Spesa
Descrizione	Questa classe rappresenta l'astrazione di una spesa.
Pre-Condizioni	<p><b>context</b> Spesa : : getData() pre: Data != null</p> <p><b>context</b> Spesa : : setData(Data) pre: Data != null</p> <p><b>context</b> Spesa : : getIdSpesa() pre: IdSpesa &gt; 0</p> <p><b>context</b> Spesa : : setIdSpesa(IdSpesa) pre: IdSpesa &gt; 0</p> <p><b>context</b> Spesa : : getDescrizione() pre: Descrizione != null</p> <p><b>context</b> Spesa : : setDescription(Descrizione) pre: Descrizione != null</p> <p><b>context</b> Spesa : : getAmmontare() pre: Ammontare &gt; 0</p> <p><b>context</b> Spesa : : setAmmontare(Ammontare) pre: Ammontare &gt; 0</p> <p><b>context</b> Spesa : : toString2() pre: Data != null and Descrizione != null and Ammontare &gt; 0 and IdSpesa &gt; 0</p>
Post-Condizioni	<p><b>context</b> Spesa : : setData(Data) post: Data != null</p> <p><b>context</b> Spesa : : setDescription(Descrizione) post: Descrizione != null</p> <p><b>context</b> Spesa : : setIdSpesa(IdSpesa) post: IdSpesa &gt; 0</p> <p><b>context</b> Spesa : : setAmmontare(Ammontare) post: Ammontare &gt; 0</p> <p><b>context</b> Spesa : : toString2() post: Stringa != null</p>
Invarianti	<b>context</b> Spesa <b>inv</b> : Data != null and Descrizione != null and Ammontare > 0 and IdSpesa > 0

### 3.7 Classe Carrello

Nome Classe	Carrello
Descrizione	Questa classe rappresenta l'astrazione di una carrello.

Pre-Condizioni	<p><b>context</b> Carrello : : getSpedizione() pre: Spedizione != null</p> <p><b>context</b> Carrello : : setSpedizione(Spedizione) pre: Spedizione != null</p> <p><b>context</b> Carrello : : getTotal() pre: Totale != null</p> <p><b>context</b> Carrello : : setTotale(Totale) pre: Totale != null</p> <p><b>context</b> Carrello : : getListArticoli() pre: ListaArticoli != null</p> <p><b>context</b> Carrello : : delete() pre:</p> <p><b>context</b> Carrello : : addItem(TShirt) pre: TShirt != null and ListaArticoli != null</p> <p><b>context</b> Carrello : : deleteItem(TShirt) pre: TShirt != null and ListaArticoli != null</p> <p><b>context</b> Carrello : : ModifyT(TShirt) pre: TShirt != null and ListaArticoli != null</p> <p><b>context</b> Carrello : : ModifyQ(TShirt) pre: TShirt != null and ListaArticoli != null</p>
Post-Condizioni	<p><b>context</b> Carrello : : setSpedizione(Spedizione) post: Spedizione != null</p> <p><b>context</b> Carrello : : setTotale(Totale) post: Totale != null</p> <p><b>context</b> Carrello : : delete() post: ListaArticoli vuota and Totale=0</p> <p><b>context</b> Carrello : : addItem(TShirt) post: TShirt in ListaArticoli</p> <p><b>context</b> Carrello : : deleteItem(TShirt) post: TShirt not in ListaArticoli</p>
Invarianti	<p><b>context</b> Carrello <b>inv:</b> ListaArticoli != null and Totale != null and Spedizione != null</p>

### 3.8 Classe Cliente

Nome Classe	Cliente
Descrizione	Questa classe rappresenta l'astrazione di un cliente.
Pre-Condizioni	<p><b>context</b> Cliente : : getNome() pre: Nome != null</p> <p><b>context</b> Cliente : : setNome(Nome) pre: Nome != null</p> <p><b>context</b> Cliente : : getCognome() pre: Cognome != null</p>

	<p><b>context</b> Cliente : : setCognome(Cognome) pre: Cognome != null</p> <p><b>context</b> Cliente : : getIdCliente() pre: IdCliente != null</p> <p><b>context</b> Cliente : : setIdCliente(IdCliente) pre: IdCliente != null</p> <p><b>context</b> Cliente : : getEmail() pre: Email != null</p> <p><b>context</b> Cliente : : setEmail(Email) pre: Email != null</p> <p><b>context</b> Cliente : : getRecapito() pre: Recapito != null</p> <p><b>context</b> Cliente : : setRecapito(Recapito) pre: Recapito != null</p>
Post-Condizioni	<p><b>context</b> Cliente : : setName(Nome) post: Nome != null</p> <p><b>context</b> Cliente : : setCognome(Cognome) post: Cognome != null</p> <p><b>context</b> Cliente : : setIdCliente(IdCliente) post: IdCliente != null</p> <p><b>context</b> Cliente : : setEmail(Email) post: Email != null</p> <p><b>context</b> Cliente : : setRecapito(Recapito) post: Recapito != null</p>
Invarianti	<p><b>context</b> Cliente <b>inv:</b> Nome != null and Cognome != null and IdCliente != null and Email != null and Recapito != null</p>

### 3.9 Classe ManageArticolo

Nome Classe	ManageArticolo
Descrizione	Questa classe rappresenta l'astrazione della gestione degli articoli.
Pre-Condizioni	<p><b>context</b> ManageArticolo : : insertArticolo(Articolo) pre: Articolo != null</p> <p><b>context</b> ManageArticolo : : deleteArticolo(idArticolo) pre: idArticolo != null</p> <p><b>context</b> ManageArticolo : : updateNome(idArticolo, Nome) pre: idArticolo != null and Nome != null</p> <p><b>context</b> ManageArticolo : : updateCategoria(idArticolo, Categoria) pre: idArticolo != null and Categoria != null</p> <p><b>context</b> ManageArticolo : : updatePrezzo(idArticolo, Prezzo) pre: idArticolo != null and Prezzo != null</p> <p><b>context</b> ManageArticolo : : updateQuantita(idArticolo, Quantita) pre: idArticolo != null and Quantita != null</p>



	<p><b>context</b> ManageArticolo : : doRetrieveByKey(codice, sesso) pre: codice != null and sesso != null</p> <p><b>context</b> ManageArticolo : : doRetrieveMen() pre:</p> <p><b>context</b> ManageArticolo : : doRetrieveWomen() pre:</p> <p><b>context</b> ManageArticolo : : getCategory(category) pre: category != null</p> <p><b>context</b> ManageArticolo : : getQuantita(idArticolo) pre: idArticolo != null</p> <p><b>context</b> ManageArticolo : : getArticolo(idArticolo) pre: idArticolo != null</p>
Post-Condizioni	
Invarianti	<b>context</b> ManageArticolo <b>inv</b> :

### 3.10 Classe ManageCarrello

Nome Classe	ManageCarrello
Descrizione	Questa classe rappresenta l'astrazione della gestione di un carrello.
Pre-Condizioni	<b>context</b> ManageCarrello : : saveCart(carrello, DatiCliente, i) pre: carrello != null and DatiCliente != null and i != null
Post-Condizioni	
Invarianti	<b>context</b> ManageCarrello <b>inv</b> :

### 3.11 Classe MangeDipendente

Nome Classe	ManageDipendente
Descrizione	Questa classe rappresenta l'astrazione della gestione dei dipendenti.
Pre-Condizioni	<p><b>context</b> ManageDipendente : : getDipendenti() pre:</p> <p><b>context</b> ManageDipendente : : insertDipendente(Dipendente) pre: Dipendente != null</p> <p><b>context</b> ManageDipendente : : deleteDipendente(idDipendente) pre: idDipendente != null</p> <p><b>context</b> ManageDipendente : : updateNome(idDipendente, Nome) pre: idDipendente != null and Nome != null</p>

	<p><b>context</b> ManageDipendente : : updateCognome(idDipendente, Cognome) pre: idDipendente != null and Cognome != null</p> <p><b>context</b> ManageDipendente : updateCodiceFiscale(idDipendente,CodiceFiscale) pre: Dipendente != null and CF!= null</p> <p><b>context</b> ManageDipendente : : updateStipendio(idDipendente, Stipendio) pre: idDipendente != null and Stipendio != null</p> <p><b>context</b> ManageDipendente : : updateTipo(idDipendente, Tipo) pre: idDipendente != null and Tipo != null</p> <p><b>context</b> ManageDipendente : : updateUsername(idDipendente, Username) pre: idDipendente != null and Username != null</p> <p><b>context</b> ManageDipendente : : updatePassword(idDipendente, Password) pre: idDipendente != null and Password != null</p> <p><b>context</b> ManageDipendente : : getDipendente(idDipendente) pre: idDipendente != null</p> <p><b>context</b> ManageDipendente : : getDipendente(Username, Password) pre: Username != null and Password != null</p> <p><b>context</b> ManageDipendente : : checkLogin(Username, Password) pre: Username != null and Password != null</p> <p><b>context</b> ManageDipendente : : getStipendiP(Data) pre: Data != null</p>
Post-Condizioni	<p><b>context</b> ManageDipendente : : getDipendente(idDipendente) post: Dipendente != null</p> <p><b>context</b> ManageDipendente : : getDipendente(Username, Password) post: Dipendente != null</p> <p><b>context</b> ManageDipendente : : checkLogin(Username, Password) post: tipo != null</p> <p><b>context</b> ManageDipendente : : getStipendiP(Data) post: listaDipendenti != null</p> <p><b>context</b> ManageDipendente : : getDipendenti() post: ListaDipendenti != null</p>
Invarianti	<p><b>context</b> ManageDipendente <b>inv</b>:</p>

### 3.12 Classe ManageFattura

Nome Classe	ManageFattura
Descrizione	Questa classe rappresenta l'astrazione della gestione delle fatture.
Pre-Condizioni	<b>context</b> ManageFattura : : getFatture() pre: <b>context</b> ManageFattura : : insertFattura(Fattura) pre: Fattura != null
Post-Condizioni	
Invarianti	<b>context</b> ManageFattura <b>inv</b> :

### 3.13 Classe ManageOrdine

Nome Classe	ManageOrdine
Descrizione	Questa classe rappresenta l'astrazione della gestione degli ordini.
Pre-Condizioni	<b>context</b> ManageOrdine : : getOrdini() pre: <b>context</b> ManageOrdine : : getInevasi() pre: <b>context</b> ManageOrdine : : deleteOrdine(Ordine) pre: Ordine != null <b>context</b> ManageOrdine : updateIdCliente(idOrdine, idCliente) pre: idOrdine != null and idCliente != null <b>context</b> ManageOrdine : updateIndirizzoConsegna(idOrdine, IndirizzoConsegna) pre: idOrdine != null and IndirizzoConsegna != null <b>context</b> ManageOrdine : updateTotale(idOrdine, Totale) pre: idOrdine != null and Totale != null <b>context</b> ManageOrdine : updateMetodoPagamento(idOrdine, MetodoPagamento) pre: idOrdine != null and MetodoPagamento != null <b>context</b> ManageOrdine : updateStato(idOrdine, Stato) pre: idOrdine != null and Stato != null <b>context</b> ManageOrdine :getI() pre: <b>context</b> ManageOrdine : evadi(idOrdine) pre: idOrdine != null <b>context</b> ManageOrdine : getOrdine(idOrdine) pre: idOrdine != null <b>context</b> ManageOrdine : dettagliA(idOrdine) pre: idOrdine != null

	<b>context</b> ManageOrdine : dettagliC(idCliente) pre: idCliente != null
Post-Condizioni	<b>context</b> ManageOrdine : getOrdine(idOrdine) pre: Ordine != null  <b>context</b> ManageOrdine : dettagliA(idOrdine) pre: ListaArticoli != null  <b>context</b> ManageOrdine : dettagliC(idCliente) pre: Cliente != null  <b>context</b> ManageOrdine : getI() post: I != null  <b>context</b> ManageOrdine : : getOrdini() post: ListaOrdini != null  <b>context</b> ManageOrdine : : getInevasi() post: ListaInevasi != null
Invarianti	<b>context</b> ManageOrdine <b>inv:</b>

### 3.14 Classe ManageRifornimento

Nome Classe	ManageRifornimento
Descrizione	Questa classe rappresenta l'astrazione della gestione delle richieste di rifornimento.
Pre-Condizioni	<b>context</b> ManageRifornimento : : getRifornimenti() pre:  <b>context</b> ManageRifornimento : : insertRifornimento(Rifornimento) pre: Rifornimento != null  <b>context</b> ManageRifornimento : : insertComposizione(idRifornimento, Idtshirt, pezzi) pre: idRifornimento != null and Idtshirt != null and pezzi > 0  <b>context</b> ManageRifornimento : : deleteRifornimento(Rifornimento) pre: Rifornimento != null  <b>context</b> ManageRifornimento : : updateDataEffettuazione(dataEffettuazione, idRifornimento)pre: idRifornimento != null and dataEffettuazione != null  <b>context</b> ManageRifornimento : : updateDataRichiesta(dataRichiesta, idRifornimento)pre: idRifornimento != null and dataRichiesta != null  <b>context</b> ManageRifornimento : : updateDataConsegna(dataConsegna, idRifornimento)pre: idRifornimento != null and dataConsegna!= null  <b>context</b> ManageRifornimento : : evadi(idRifornimento) pre: idRifornimento != null  <b>context</b> ManageRifornimento : : getInevasi() pre:

	<b>context</b> ManageRifornimento : : insertRifornimento(Rifornimento,Stato) pre: Rifornimento != null and Stato != null
Post-Condizioni	<b>context</b> ManageRifornimento : : getInevasi() post: ListaInevasi!=null  <b>context</b> ManageRifornimento : : getRifornimenti() post: ListaRifornimenti!=null
Invarianti	<b>context</b> ManageRifornimento <b>inv</b> :

### 3.15 Classe ManageSpesa

Nome Classe	ManageSpesa
Descrizione	Questa classe rappresenta l'astrazione della gestione delle spese.
Pre-Condizioni	<b>context</b> ManageSpesa : : getSpese() pre:  <b>context</b> ManageSpesa : : insertSpesa(Spesa) pre:Spesa != null  <b>context</b> ManageSpesa : : deleteSpesa(idSpesa) pre: idSpesa != null  <b>context</b> ManageSpesa : : updateDescrizione(idSpesa, Descrizione) pre: idSpesa != null and Descrizione != null  <b>context</b> ManageSpesa : : updateAmmontare(idSpesa, ammontare) pre: idSpesa != null and ammontare != null  <b>context</b> ManageSpesa : : updateData(idSpesa, Data) pre: idSpesa != null and Data != null
Post-Condizioni	<b>context</b> ManageSpesa : : getSpese() post: ListaSpese != null
Invarianti	<b>context</b> ManageSpesa <b>inv</b> :

### 3.16 Classe ManageTShirt

Nome Classe	ManageTShirt
Descrizione	Questa classe rappresenta l'astrazione della gestione delle t-shirt.
Pre-Condizioni	<b>context</b> ManageTShirt : : getTShirts() pre:  <b>context</b> ManageTShirt : : getArticoliS() pre:  <b>context</b> ManageTShirt : : insertTShirt(TShirt) pre: TShirt != null

	<p><b>context</b> ManageTShirt : : deleteTShirt(idTShirt) pre: idTShirt != null</p> <p><b>context</b> ManageTShirt : : updateSesso(idTShirt, sesso) pre: idTshirt != null and sesso != null</p> <p><b>context</b> ManageTShirt : : updateTaglia(idTShirt, taglia) pre: idTshirt != null and taglia != null</p> <p><b>context</b> ManageTShirt : : updateColore(idTShirt, colore) pre: idTshirt != null and colore != null</p> <p><b>context</b> ManageTShirt : : updateQuantita(idTShirt, quantita) pre: idTshirt != null and quantita != null</p> <p><b>context</b> ManageTShirt : : getIdTshirt(idTShirt, sesso, taglia, colore) pre: idTshirt != null and sesso != null and taglia != null and colore != null</p> <p><b>context</b> ManageTShirt : : getTshirtR(idTShirt) pre: idTshirt != null</p>
Post-Condizioni	<p><b>context</b> ManageTShirt : : getTShirts() post: ListaTShirt != null</p> <p><b>context</b> ManageTShirt : : getArticoliS() post: Lista != null</p> <p><b>context</b> ManageTShirt : : getIdTshirt(idTShirt, sesso, taglia, colore) pre: IdTshirt != null</p> <p><b>context</b> ManageTShirt : : getTshirtR(idTShirt) pre: Tshirt != null</p>
Invarianti	<b>context</b> ManageTShirt <b>inv:</b>

### 3.17 Classe ManageStipendi

Nome Classe	ManageStipendi
Descrizione	Questa classe si occupa della gestione degli stipendi.
Pre-Condizioni	<p><b>context</b> ManageStipendi : : inserisciStipendiP(idDipendente, data, stipendio) pre: idDipendente != null and data != null and stipendio != null</p> <p><b>context</b> ManageStipendi : : inserisciStipendiP(ListaDipendenti, data) pre: ListaDipendenti != null and data != null</p>
Post-Condizioni	
Invarianti	<b>context</b> ManageStipendi <b>inv:</b>

### 3.18 Classe ManageTotale

Nome Classe	ManageTotale
Descrizione	Questa classe si occupa della gestione del capitale.
Pre-Condizioni	<b>context</b> ManageTotale : : getTotal(anno) pre: anno != null  <b>context</b> ManageTotale : : aggiungiTotale(anno, somma) pre: anno != null and somma != null  <b>context</b> ManageTotale : : sottraiTotale(anno, somma) pre: anno != null and somma != null
Post-Condizioni	
Invarianti	<b>context</b> ManageTotale <b>inv</b> :

### 3.19 Classe Controller

Nome Classe	Controller
Descrizione	Questa classe si occupa della gestione dei dati degli articoli, della gestione del carrello.
Pre-Condizioni	<b>context</b> Controller : : doGet(request, response) pre: request != null and response != null  <b>context</b> Controller : : doPost(request, response) pre: request != null and response != null
Post-Condizioni	<b>context</b> Controller : : doGet(request, response) post: request != null and response != null  <b>context</b> Controller : : doPost(request, response) post: request != null and response != null
Invarianti	<b>context</b> Controller <b>inv</b> :

### 3.20 Classe FlussoEconomico

Nome Classe	FlussoEconomico
Descrizione	Questa classe si occupa della gestione delle entrate e delle uscite.
Pre-Condizioni	<b>context</b> FlussoEconomico : : doGet(request, response) pre: request != null and response != null  <b>context</b> FlussoEconomico : : doPost(request, response) pre: request != null and response != null
Post-Condizioni	<b>context</b> FlussoEconomico : : doGet(request, response) post: request



	<p>!= null and response != null</p> <p><b>context</b> FlussoEconomico : : doPost(request, response) post: request != null and response != null</p>
Invarianti	<b>context</b> FlussoEconomico <b>inv</b> :

### 3.21 Classe GestisciArticolo

Nome Classe	GestisciArticolo
Descrizione	Questa classe si occupa della gestione degli articoli.
Pre-Condizioni	<p><b>context</b> GestisciArticolo : : doGet(request, response) pre: request != null and response != null</p> <p><b>context</b> GestisciArticolo : : doPost(request, response) pre: request != null and response != null</p>
Post-Condizioni	<p><b>context</b> GestisciArticolo : : doGet(request, response) post: request != null and response != null</p> <p><b>context</b> GestisciArticolo : : doPost(request, response) post: request != null and response != null</p>
Invarianti	<b>context</b> GestisciArticolo <b>inv</b> :

### 3.22 Classe ServletTshirt

Nome Classe	ServletTshirt
Descrizione	Questa classe si occupa della gestione della Tshirt.
Pre-Condizioni	<p><b>context</b> ServletTshirt : : doGet(request, response) pre: request != null and response != null</p> <p><b>context</b> ServletTshirt : : doPost(request, response) pre: request != null and response != null</p>
Post-Condizioni	<p><b>context</b> ServletTshirt : : doGet(request, response) post: request != null and response != null</p> <p><b>context</b> ServletTshirt : : doPost(request, response) post: request != null and response != null</p>
Invarianti	<b>context</b> ServletTshirt <b>inv</b> :

### 3.23 Classe GestisciDipendente

Nome Classe	GestisciDipendente
Descrizione	Questa classe si occupa della gestione dei dipendenti.
Pre-Condizioni	<b>context</b> GestisciDipendente : : doGet(request, response) pre: request != null and response != null  <b>context</b> GestisciDipendente : : doPost(request, response) pre: request != null and response != null
Post-Condizioni	<b>context</b> GestisciDipendente : : doGet(request, response) post: request != null and response != null  <b>context</b> GestisciDipendente : : doPost(request, response) post: request != null and response != null
Invarianti	<b>context</b> GestisciDipendente <b>inv</b> :

### 3.24 Classe ServletLogin

Nome Classe	ServletLogin
Descrizione	Questa classe si occupa della gestione degli accessi.
Pre-Condizioni	<b>context</b> ServletLogin : : doGet(request, response) pre: request != null and response != null  <b>context</b> ServletLogin : : doPost(request, response) pre: request != null and response != null
Post-Condizioni	<b>context</b> ServletLogin : : doGet(request, response) post: request != null and response != null  <b>context</b> ServletLogin : : doPost(request, response) post:            request != null and response != null
Invarianti	<b>context</b> ServletLogin <b>inv</b> :

### 3.25 Classe DbConnect

Nome Classe	DbConnect
Descrizione	Questa classe si occupa della connessione con il database.
Pre-Condizioni	<b>context</b> DbConnect : : connect() pre:  <b>context</b> DbConnect : : close() pre: Connection != null  <b>context</b> DbConnect : : getConnection() pre: Connection != null

Post-Condizioni	<b>context</b> DbConnect : : getConnection() post: Connection != null
Invarianti	<b>context</b> DbConnect <b>inv</b> :

### 3.26 Classe ServletInsertSpesa

Nome Classe	ServletInsertSpesa
Descrizione	Questa classe si occupa dell'inserimento delle spese.
Pre-Condizioni	<b>context</b> ServletInsertSpesa : : doGet(request, response) pre: request != null and response != null  <b>context</b> ServletInsertSpesa : : doPost(request, response) pre: request != null and response != null
Post-Condizioni	<b>context</b> ServletInsertSpesa : : doGet(request, response) post: request != null and response != null  <b>context</b> ServletInsertSpesa : : doPost(request, response) post: request != null and response != null
Invarianti	<b>context</b> ServletInsertSpesa <b>inv</b> :

### 3.27 Classe GestisciOrdine

Nome Classe	GestisciOrdine
Descrizione	Questa classe si occupa della gestione degli ordini.
Pre-Condizioni	<b>context</b> GestisciOrdine : : doGet(request, response) pre: request != null and response != null  <b>context</b> GestisciOrdine : : doPost(request, response) pre: request != null and response != null
Post-Condizioni	<b>context</b> GestisciOrdine : : doGet(request, response) post: request != null and response != null  <b>context</b> GestisciOrdine : : doPost(request, response) post: request != null and response != null
Invarianti	<b>context</b> GestisciOrdine <b>inv</b> :

### 3.28 Classe ServletRifornimento

Nome Classe	ServletRifornimenti
Descrizione	Questa classe si occupa della gestione dei rifornimenti.
Pre-Condizioni	<b>context</b> ServletRifornimenti : : doGet(request, response) pre:

	request != null and response != null  <b>context</b> ServletRifornimenti : : doPost(request, response) pre: request != null and response != null
Post-Condizioni	<b>context</b> ServletRifornimenti : : doGet(request, response) post: request != null and response != null  <b>context</b> ServletRifornimenti : : doPost(request, response) post: request != null and response != null
Invarianti	<b>context</b> ServletRifornimenti <b>inv:</b> GestoreRifornimenti != null

### 3.29 Classe ServletInsertArticolo

Nome Classe	ServletInsertArticolo
Descrizione	Questa classe si occupa dell'inserimento degli articoli.
Pre-Condizioni	<b>context</b> ServletInsertArticolo : : doGet(request, response) pre: request != null and response != null  <b>context</b> ServletInsertArticolo : : doPost(request, response) pre: request != null and response != null
Post-Condizioni	<b>context</b> ServletInsertArticolo : : doGet(request, response) post: request != null and response != null  <b>context</b> ServletInsertArticolo : : doPost(request, response) post: request != null and response != null
Invarianti	<b>context</b> ServletInsertArticolo <b>inv:</b>

### 3.30 Classe ServletInserisciDipendente

Nome Classe	ServletInserisciDipendente
Descrizione	Questa classe si occupa dell'inserimento dei dipendenti.
Pre-Condizioni	<b>context</b> ServletInserisciDipendente : : doGet(request, response) pre: request != null and response != null  <b>context</b> ServletInserisciDipendente : : doPost(request, response) pre: request != null and response != null
Post-Condizioni	<b>context</b> ServletInserisciDipendente : : doGet(request, response) post: request != null and response != null  <b>context</b> ServletInserisciDipendente : : doPost(request, response) post: request != null and response != null

Invarianti	<b>context</b> ServletInserisciDipendente <b>inv:</b>
------------	---

### 3.31 Classe PagaStipendi

Nome Classe	PagaStipendi
Descrizione	Questa classe si occupa di pagare gli stipendi.
Pre-Condizioni	<b>context</b> PagaStipendi : : doGet(request, response) pre: request != null and response != null  <b>context</b> PagaStipendi : : doPost(request, response) pre: request != null and response != null
Post-Condizioni	<b>context</b> PagaStipendi : : doGet(request, response) post: request != null and response != null  <b>context</b> PagaStipendi : : doPost(request, response) post:            request != null and response != null
Invarianti	<b>context</b> PagaStipendi <b>inv:</b>