

# Programmazione 3 e Laboratorio di Programmazione 3

Distributore automatico di bevande

**Vittorio Fones 0124/1384**

# Descrizione progetto

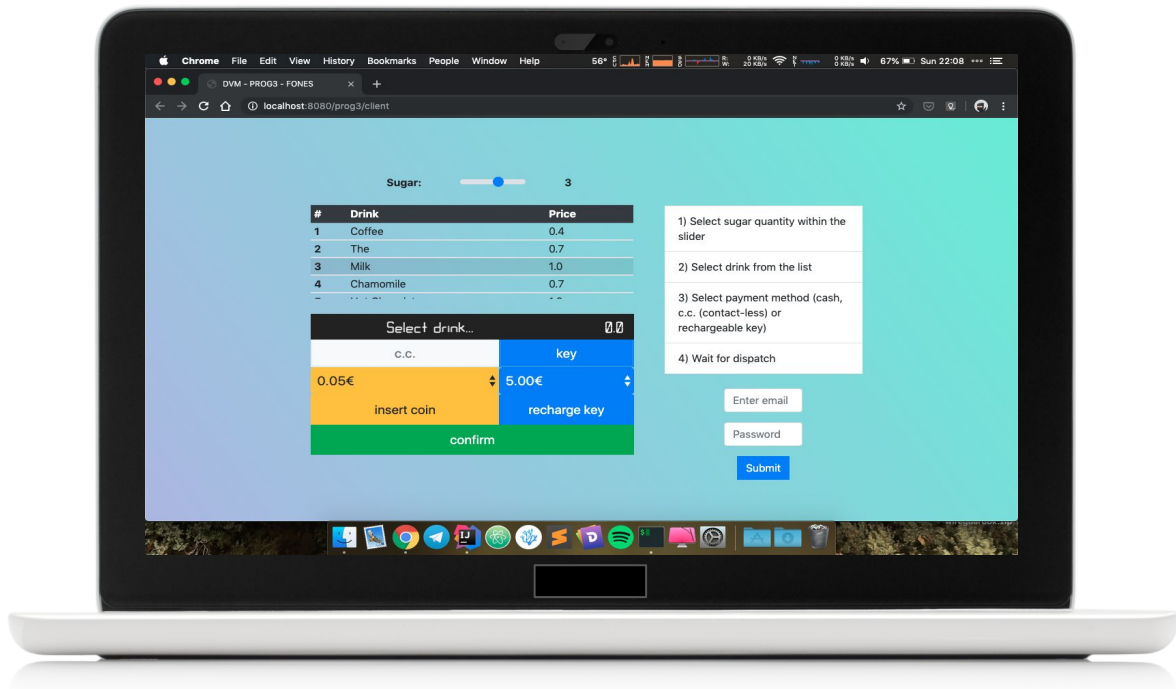
---

Si vuole simulare un distributore automatico di bevande. Il distributore ha 6 tipologie di bevande: caffè, the, latte, camomilla, cioccolata e acqua calda. L'accesso deve avvenire sia in modalità amministratore che in modalità utente.

# Modalità utente

## Utente non registrato può

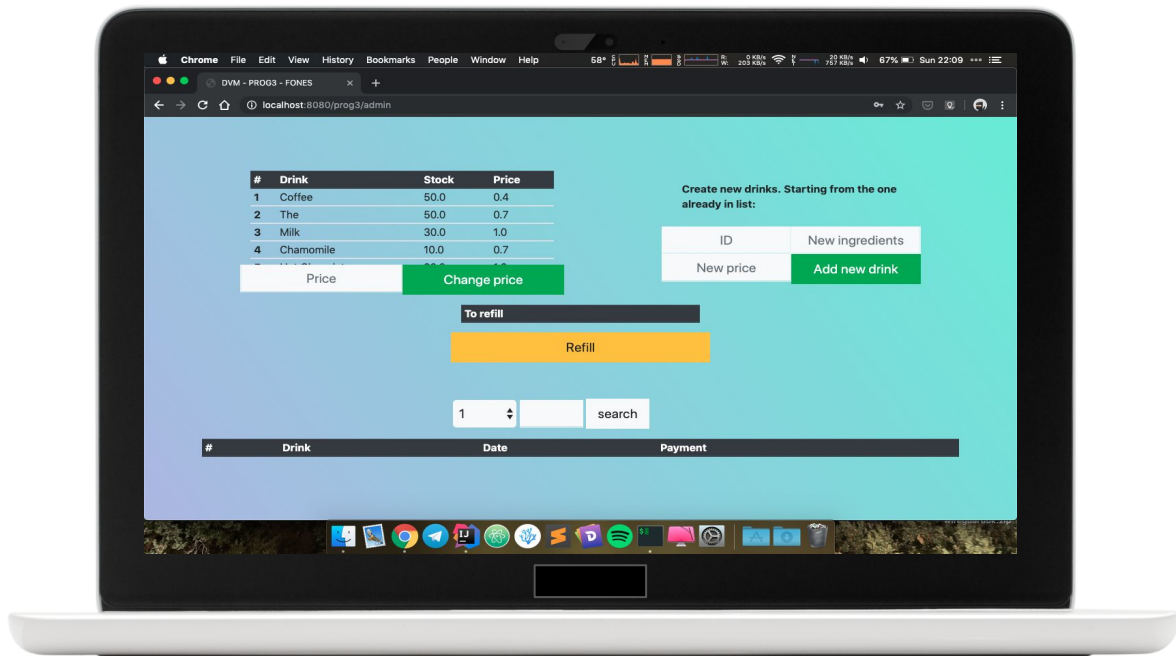
- scegliere, prelevare e pagare una bevanda. Il pagamento può avvenire secondo le modalità:
  - contanti
  - chiavetta ricaricabile
  - carta di credito
- ricaricare una chiavetta inserendo banconote



# Modalità amministratore

## Amministratore può

- periodicamente aggiungere bevande alla scorta
- definire il prezzo per ogni tipo di bevanda
- fare un report sui consumi mensili delle diverse tipologie di bevande
- aggiungere una nuova tipologia di bevanda partendo da quelle già esistenti



# Tecnologie usate:

**Java SDK 11**

**PostgreSQL 11 RDBMS**

**Servlet / Java Server Pages**

Software scritto in java per la gestione di richieste HTTP. / Software per creare pagine web dinamiche.

**Hibernate ORM**

Un framework per il mapping tra oggetti e relazionali. Estende le funzionalità di JDBC.

— — —

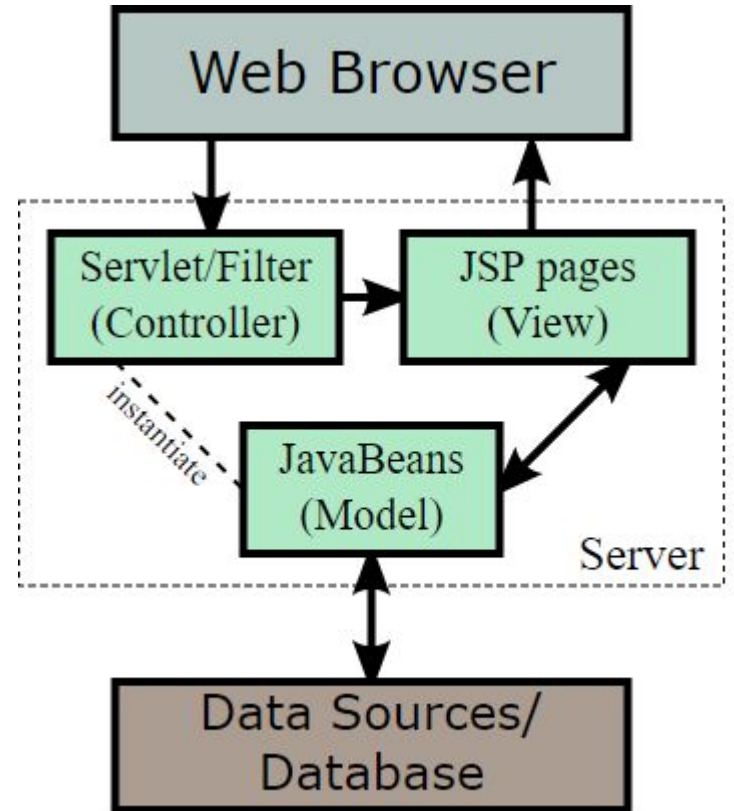
## Model Controller View

Pattern architetturale più usato nello sviluppo di applicativi web. Usato al posto di Observer e altri pattern comportamentali.

## Data Access Object

Pattern architetturale usato per gestire la persistenza dei dati in un RDBMS. Hibernate interagisce con il DB e “mappa” i risultati nelle Java Beans.

# Pattern architetturali





# → Template Method

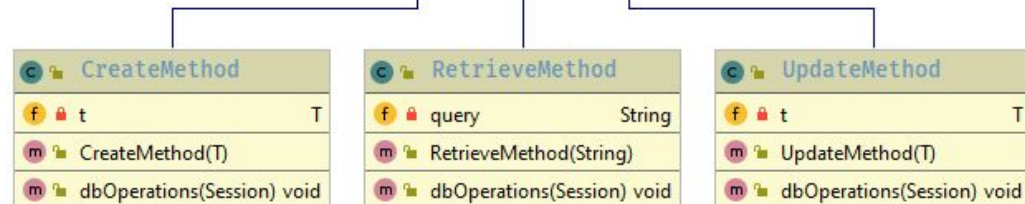
# → Singleton

dbOperations		
f	crudMethod	ACrudMethod<T>
m	dbOperations(ACrudMethod<T>)	
m	run()	void
m	changeMethod(ACrudMethod<T>)	void
p	t	T
p	list	List<T>

GenericDao		
f	selectStarFrom	String
m	GenericDao()	
m	GenericDao(String)	
m	save(T)	void
m	update(T)	void
m	query(String)	List<T>
m	queryBean(String)	T
p	all	List<T>

ACrudMethod		
m	dbOperations(Session)	void
m	run()	void
p	t	T
p	list	List<T>

Hibernate		
f	factory	SessionFactory
m	Hibernate()	
m	getSessionFactory()	SessionFactory



Purchase	
purchase_id	int
date	Date
product	Product
cash	boolean
cc_number	String
key	Key
Purchase()	
Purchase(Date, Product, boolean, String, Key)	
getPurchase_id()	int
setPurchase_id(int)	void
setDate(Timestamp)	void
getDate()	Date
setProduct(Product)	void
getProduct()	Product
setCash(boolean)	void
isCash()	boolean
getCc_number()	String
setCc_number(String)	void
getKey()	Key
setKey(Key)	void

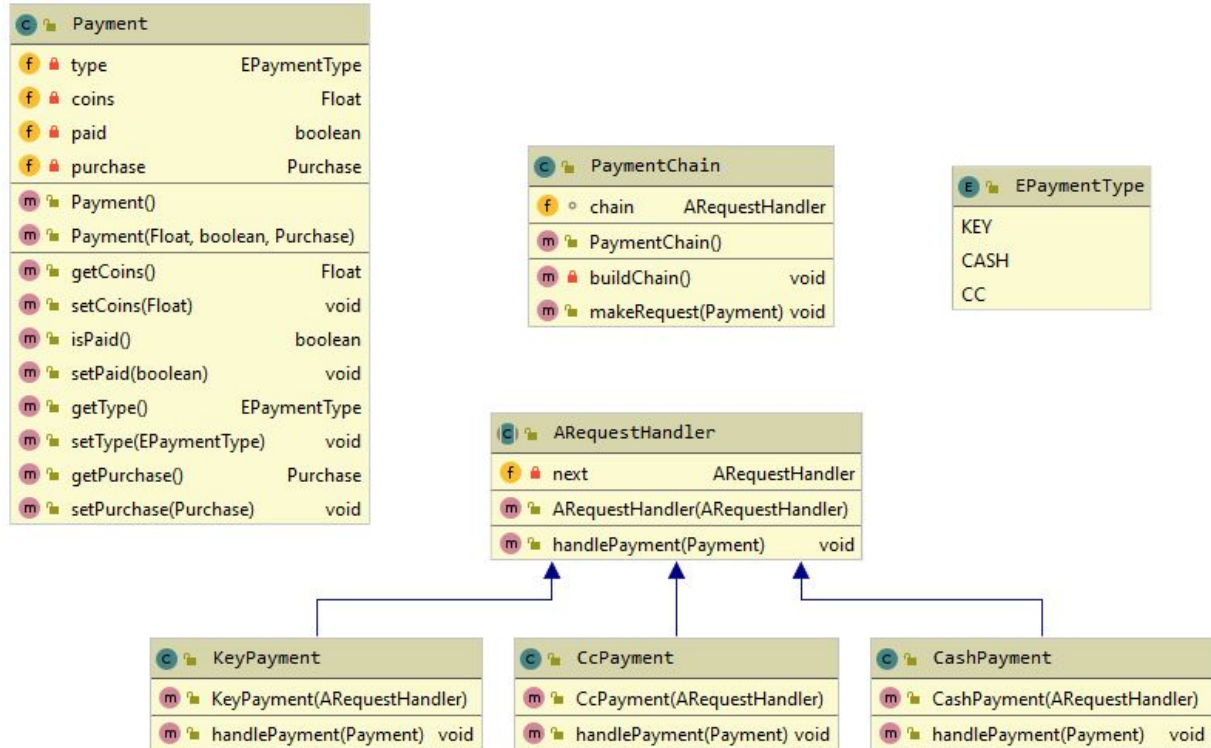
Product	
prod_id	int
price	float
stock	float
name	String
Product()	
Product(float, float, String)	
getPrice()	float
setPrice(float)	void
getStock()	float
setStock(float)	void
getName()	String
setName(String)	void
getProd_id()	int
setProd_id(int)	void

Admin	
admin_id	int
email	String
password	String
Admin()	
Admin(String, String)	
getAdmin_id()	int
setAdmin_id(int)	void
setEmail(String)	void
getEmail()	String
setPassword(String)	void
getPassword()	String

Key	
id_key	int
balance	float
Key()	
Key(float)	
getId_key()	int
setId_key(int)	void
getBalance()	float
setBalance(float)	void

# Java beans

## → Chain of Responsibility



**Grazie per l'attenzione**

**Vittorio Fones**

0124001384