

# 1.2 Iterazione 1, Analisi

## 1.2.1 Introduzione

Per l'iterazione 1 sono stati scelti i seguenti requisiti:

- lo scenario principale di successo del caso d'uso UC2 (Inserimento nuovo tipo di Porta per garage);
- lo scenario principale di successo del caso d'uso UC3 (Vendita Porta per garage);
- il calcolo del totale di una vendita è basato sulla regola di dominio R1;
- caso d'uso d'avviamento.

## 1.2.2 Caso d'uso UC2: Inserimento nuovo tipo di Porta per garage

### Scenario principale

1. L'Amministratore inizia l'immissione di un nuovo tipo di Porta per Garage.
2. L'Amministratore inserisce il nome e il codice identificativo della Porta per Garage.
3. L'Amministratore inserisce il nome e il codice identificativo di un Componente e specifica la quantità. Il Sistema aggiunge quel Componente, in quella quantità, al nuovo tipo di Porta per Garage.

*Il passo 3 viene ripetuto finché serve.*

4. L'Amministratore inserisce codice del tipo di lamiera per il nuovo tipo di Porta per Garage. Il Sistema calcola il prezzo base e mostra i dati del nuovo tipo di Porta inserito.
5. L'Amministratore conferma i dati inseriti. Il Sistema registra le informazioni sul nuovo tipo di Porta per Garage.

Dallo scenario principale del caso d'uso UC2 è possibile identificare le seguenti classi concettuali:

- **Amministratore**: attore primario, colui che usa il sistema.
- **GASH**: rappresenta il sistema Garage Shutter.
- **TipoPorta**: un tipo di porta che è possibile ordinare.
- **ComponentePorta**: un componente del tipo di porta con quantità variabile.
- **Componente**: la descrizione dei componenti che si possono ordinare
- **TipoLamiera**: Tipo di lamiera che è possibile usare per la nuova porta.

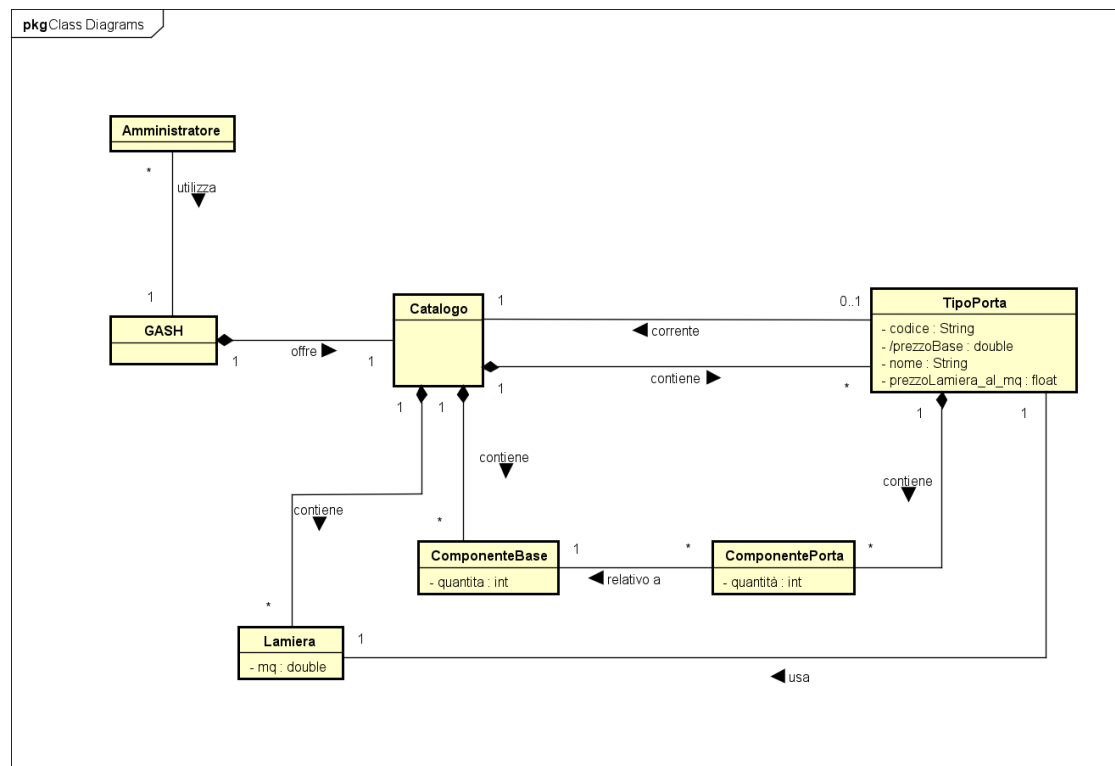
Un TipoPorta contiene vari componenti in diverse quantità e utilizza un solo tipo di Lamiera.

Ad esempio, una *PortaA* può essere composta da:

- Un tubo.
- Due guide.
- Un motore SPLIT 60.
- 4 molle.

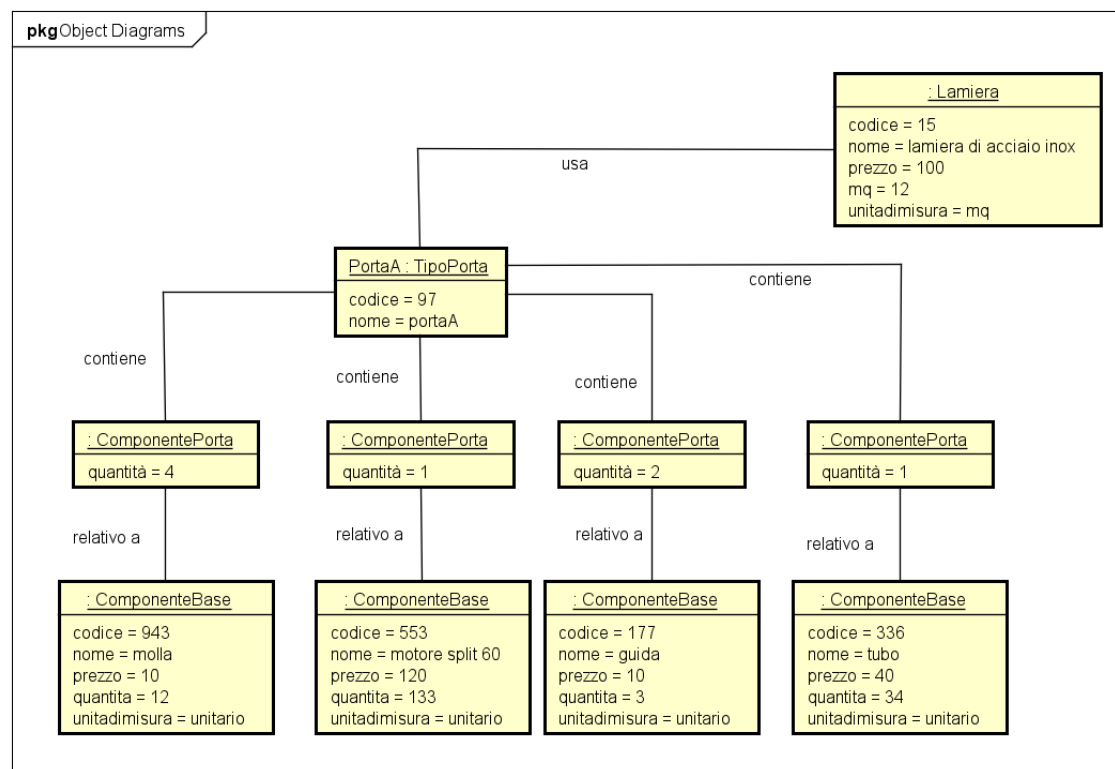
Ed utilizza come materiale un tipo di lamiera, ad esempio quella da 9mm di spessore.

## Modello di dominio



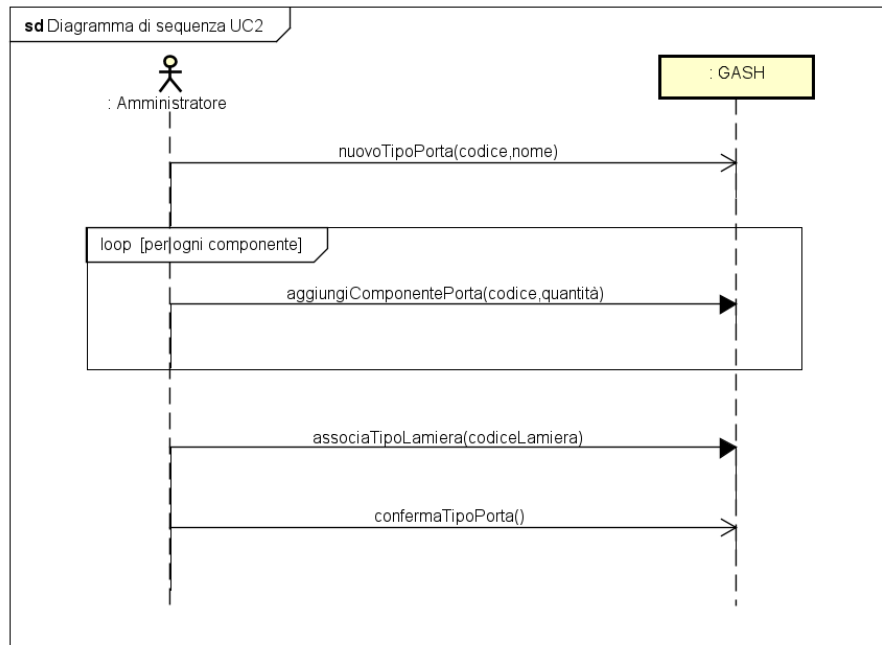
## Modello degli oggetti

Per rappresentare la *PortaA* si possono utilizzare gli oggetti mostrati dal seguente diagramma di oggetti di dominio.



### 1.2.3 Caso d'uso UC2, Diagramma di sequenza di sistema

Il diagramma di sequenza di sistema per il caso d'uso UC2 è il seguente:



### 1.2.4. Caso d'uso UC2, Contratti delle operazioni

#### Contratti delle operazioni

<b>Operazione</b>	aggiungiComponentePorta(nome:String, codice:String, quantità:Int)
<b>Riferimenti</b>	Caso d'uso: <i>Inserimento nuovo tipo di Porta per Garage</i>
<b>Pre-condizioni</b>	-è in corso la definizione di TipoPorta <i>tp</i>
<b>Post-condizioni</b>	-è stata creata una nuova istanza <i>cp</i> di ComponentePorta -gli attributi di <i>cp</i> sono stati inizializzati; - <i>cp</i> è stata associata a <i>tp</i> tramite l'associazione "contiente"; - <i>cp</i> è stata associata a un Componente <i>c</i> , basandosi sul codice tramite l'associazione "relativo a".

<b>Operazione</b>	associaTipoLamiera(codiceLamiera:String)
<b>Riferimenti</b>	Caso d'uso: <i>Inserimento nuovo tipo di Porta per Garage</i>
<b>Pre-condizioni</b>	-è in corso la definizione di TipoPorta <i>tp</i>
<b>Post-condizioni</b>	-è stata ricavata tramite una ricerca l'istanza <i>l</i> di Lamiera basandosi sul codiceLamiera - <i>l</i> è stata associa a <i>tp</i> tramite l'associazione " <i>usa</i> ";

## 1.2.5. Caso d'uso UC3: Nuova Vendita porta per garage

### Scenario principale

1. Il Cliente si reca in sede comprare una porta per garage.
2. L'Addetto seleziona i dati del cliente, effettuando una ricerca nella anagrafica dei clienti.
3. Il Cliente comunica il tipo di porta per garage che vuole ordinare e le relative dimensioni.  
L'addetto inserisce il codice identificativo della porta necessaria e specifica le dimensioni. Il Sistema calcola i mq di lamiera necessaria e il prezzo della porta. Il sistema registra la porta richiesta.

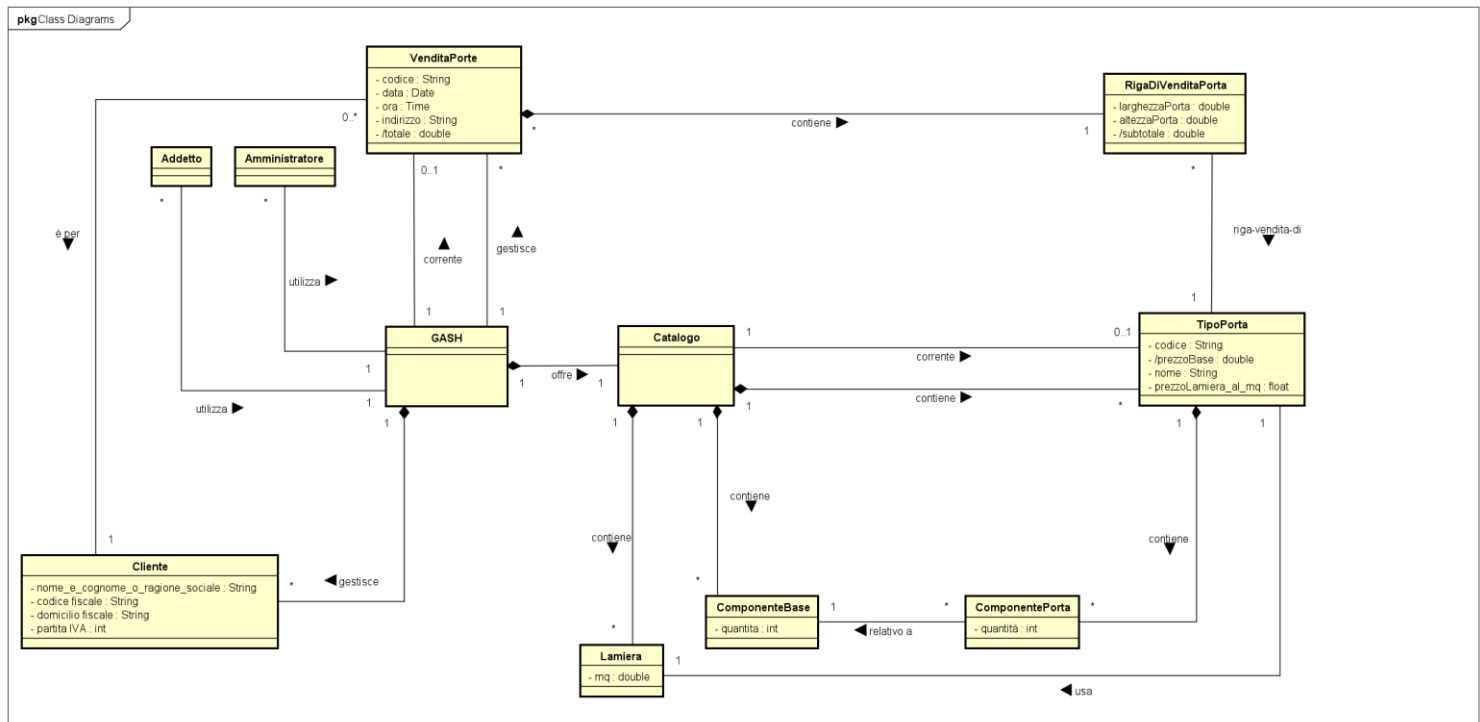
*Il passo 3 viene ripetuto finché serve*

4. Il Cliente comunica l'indirizzo in cui verrà effettuato il lavoro. L'Addetto registra questa informazione nel Sistema. Il Sistema mostra il riepilogo della vendita. L'Addetto mostra il riepilogo al cliente digitalmente.
5. L'Addetto conferma la vendita. Il Sistema registra la vendita della porta per garage.
6. Il Cliente decide di pagare subito. L'Addetto registra l'avvenuto pagamento e il tipo di pagamento. Il Sistema registra il pagamento.

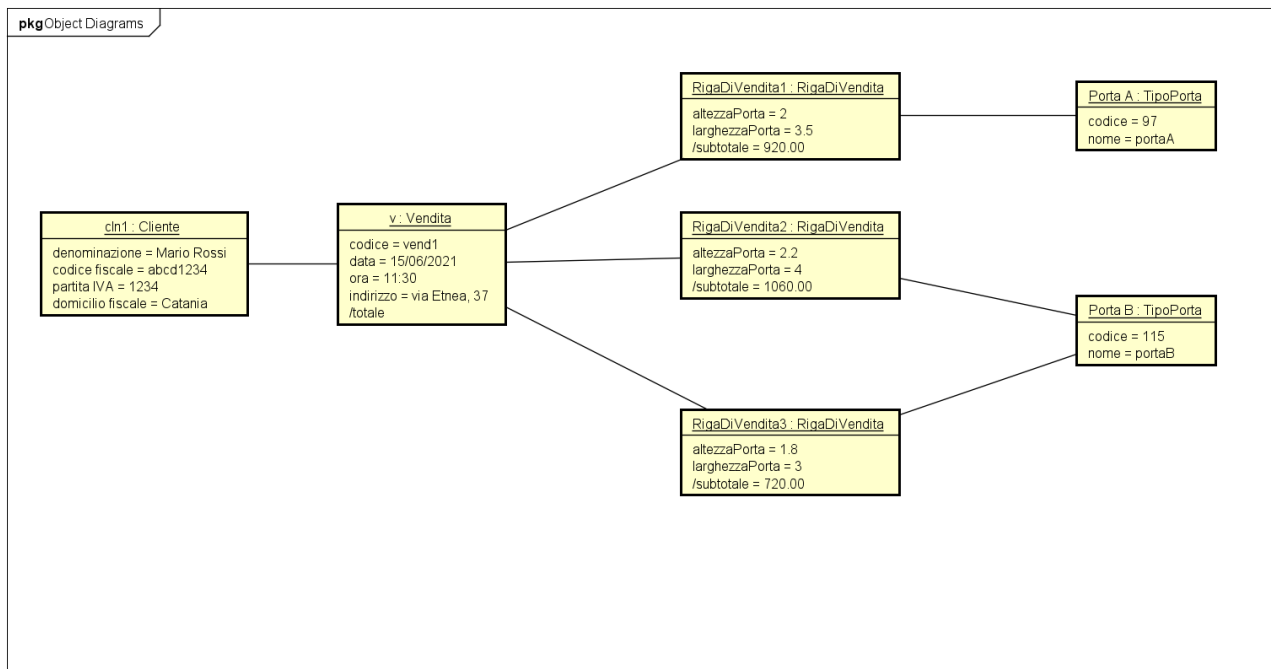
In questa prima iterazione del caso d'uso UC3 viene analizzato interamente lo scenario principale di successo. Da esso possono essere ricavate le seguenti classi, non ancora identificate nel caso d'uso precedente:

- **Cliente:** un cliente di GASH. Può ordinare porte per garage.
- **Addetto:** si occupa della vendita di porte per garage.
- **Vendita:** una vendita ad un cliente, composta da una o più porte per garage.
- **RigaDiVendita:** contiene le informazioni riguardanti la vendita di un elemento, corrispondente ad una porta per garage.

## Modello di dominio



## Modello degli oggetti



Un oggetto vendita rappresenta la vendita di una o più porte per garage ed è effettuata dall'Addetto per un cliente. Una vendita potrebbe essere registrata su un apposito modulo, come mostrato nella figura seguente:

### Vendita n. 256

---

1 Porta A

2 Porte B

totale ordine: €200.00

cliente:

Mario Rossi

consegna:

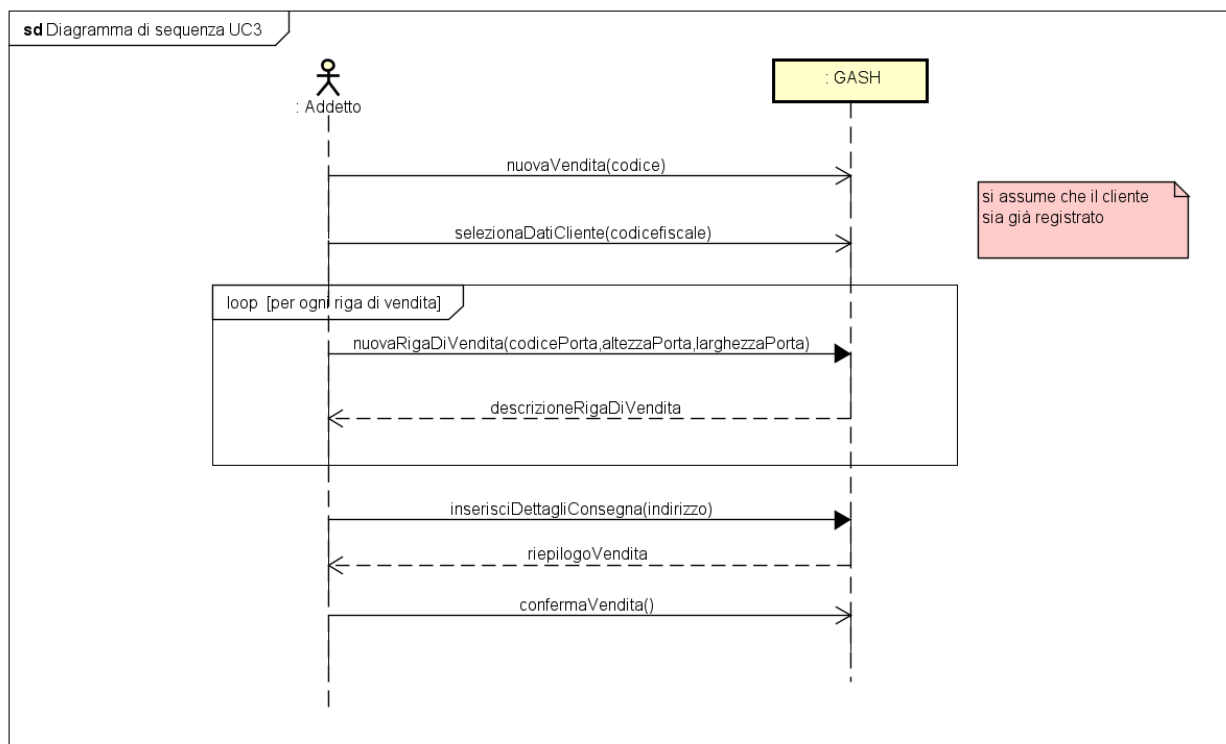
via Etnea, 37

15 giugno 2021, ore 11:30

L'oggetto `v:Vendita` rappresenta la Vendita. Esso è composto da righe, che specificano le porte per garage richieste; queste sono rappresentate dagli oggetti `RigaDiVendita`. Gli altri elementi della vendita sono rappresentati da altri oggetti, associazioni ed attributi del modello di dominio, come mostrato nel seguente diagramma di oggetti di dominio, che rappresenta la vendita in considerazione.

### 1.2.6 Caso d'uso UC3, Diagramma di sequenza di sistema

Il diagramma di sequenza di sistema per lo scenario principale di successo del caso d'uso UC3 per questa iterazione è il seguente:



## 1.2.7 Caso d'uso UC3, Contratti delle operazioni

### 1.2.7.1 Nuova riga di vendita

L'operazione di sistema `nuovaRigaDiVendita` consente di aggiungere una nuova porta per garage alla vendita corrente.

<b>Operazione</b>	<code>nuovaRigaDiVendita(codicePorta: int, altezzaPorta: Float, larghezzaPorta: Float)</code>
<b>Riferimenti</b>	Caso d'uso: <i>Nuova Vendita porta per garage</i>
<b>Pre-condizioni</b>	- è in corso l'inserimento di una vendita <i>v</i>
<b>Post-condizioni</b>	<ul style="list-style-type: none"><li>- è stata creata una nuova istanza <i>rv</i> di <code>RigaDiVendita</code>;</li><li>- la riga di vendita <i>rv</i> è stata associata a un <code>TipoPorta tp</code>, sulla base di <code>codicePorta</code>, tramite l'associazione "registra-vendita-di";</li><li>- gli attributi di <i>rv</i> sono stati inizializzati (l'attributo <code>/subtotale</code> viene calcolato come segue: <math>\text{subtotale} = \text{prezzoBase} + \text{altezzaPorta} \times \text{larghezzaPorta} \times \text{prezzoLamiera\_al\_mq}</math>);</li><li>- la riga di vendita <i>rv</i> è stata associata alla Vendita corrente tramite l'associazione "contiene".</li></ul>

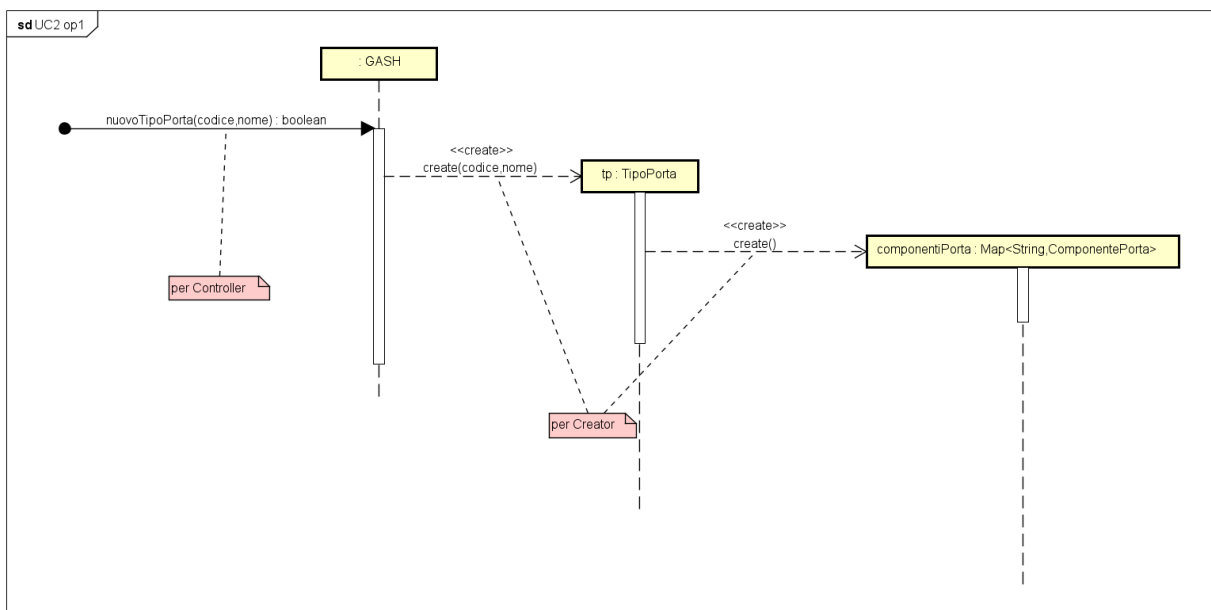
# 1.3 Iterazione 1, Progettazione

## 1.3.1 Introduzione

La prima scelta di progetto da fare è sul controller per i vari casi d'uso. Si è deciso di utilizzare la classe GASH come facade controller.

## 1.3.2 Caso d'uso UC2, Diagrammi di interazione

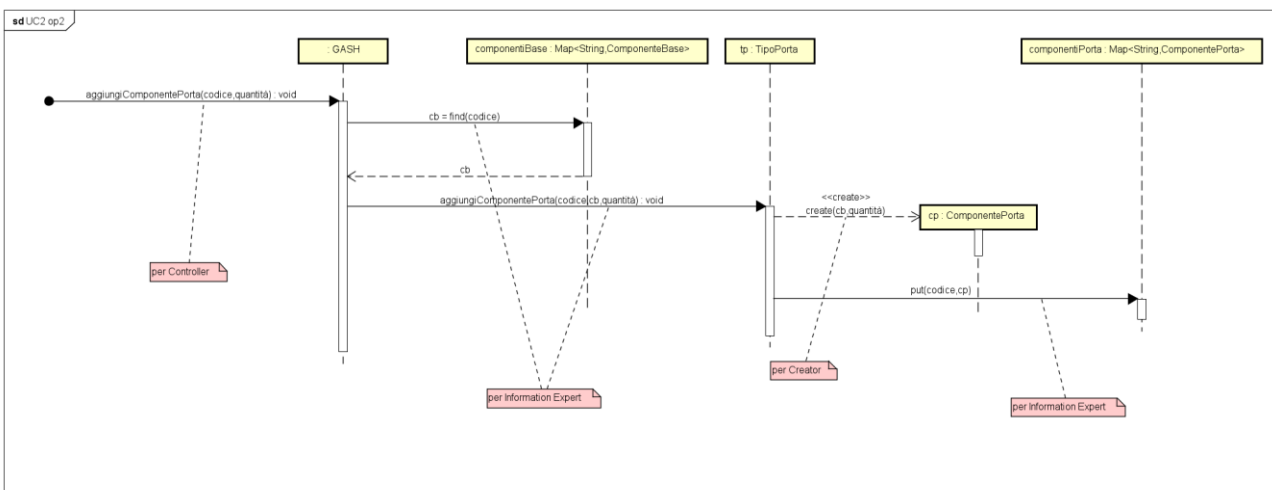
### 1.3.2.1 nuovoTipoPorta(codice:String,nome:String):Boolean



L'operazione `nuovoTipoPorta` ritorna una variabile Booleana: se l'oggetto di `TipoPorta` `tp` viene istanziato correttamente ritorna `true`, altrimenti false.

Durante la fase di implementazione verranno effettuati dei controlli sul codice e sul nome inseriti, ad esempio un controllo sull'esistenza di un tipo porta con quel codice nel sistema.

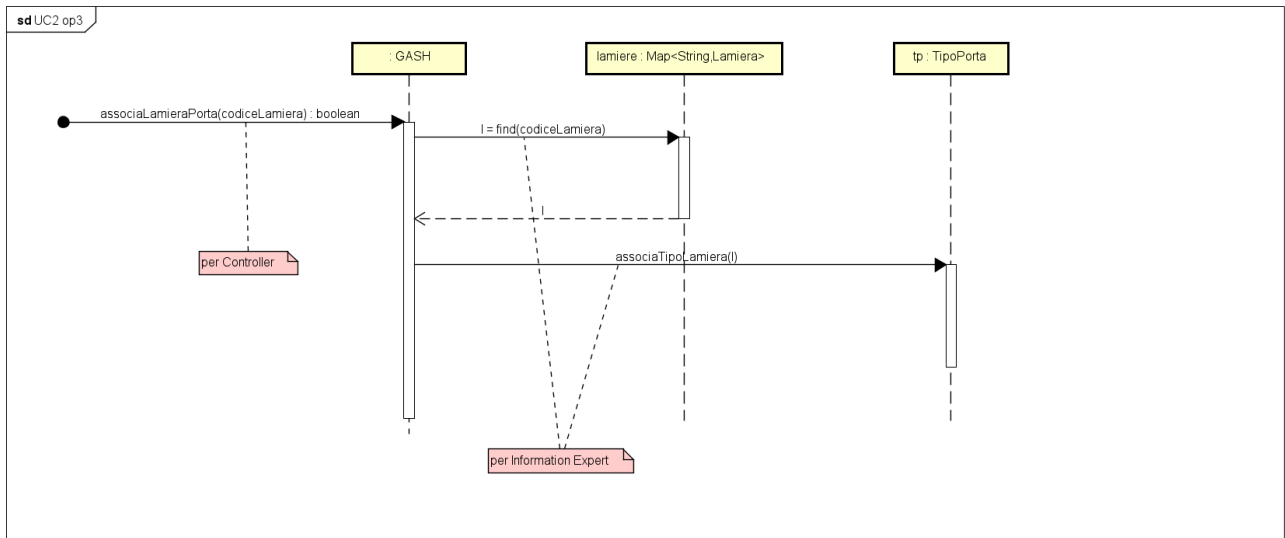
### 1.3.2.2 aggiungiComponentePorta(codice:String,quantita:Int)



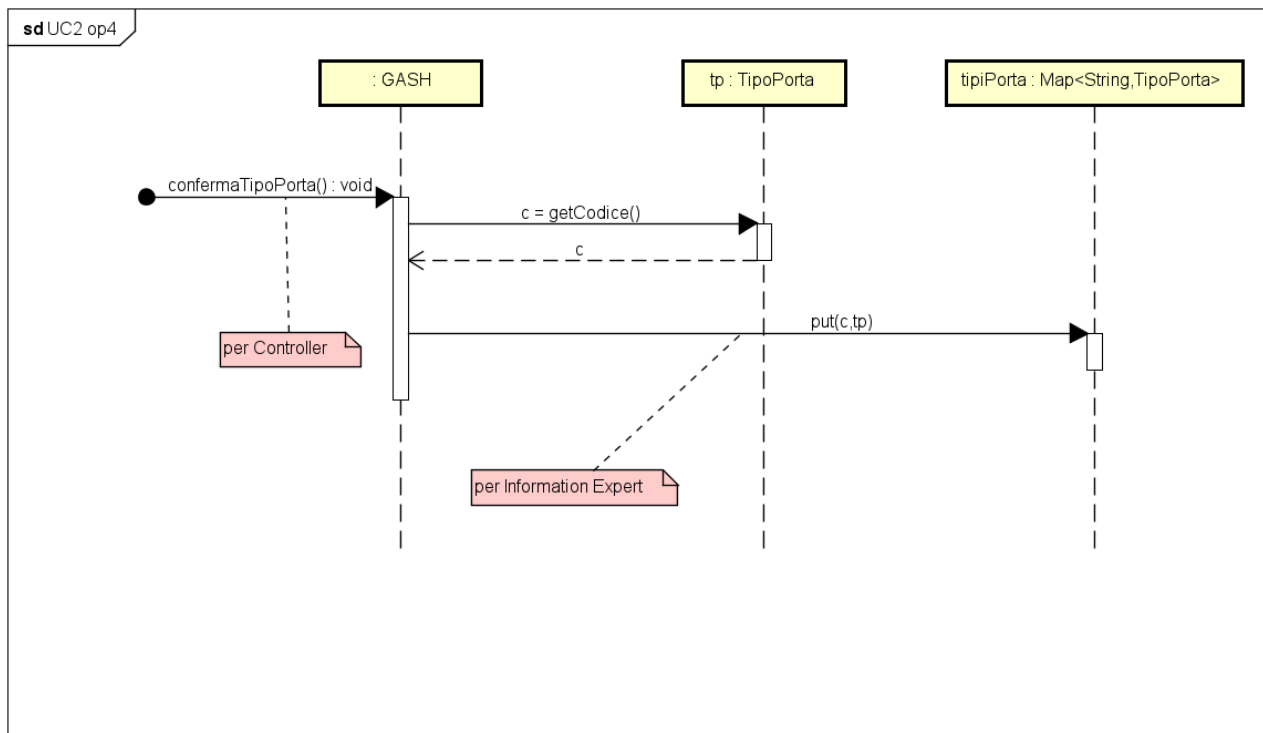


Questa operazione permette di aggiungere un ComponenteBase alla Mappa di componenti porta con chiave il codice.

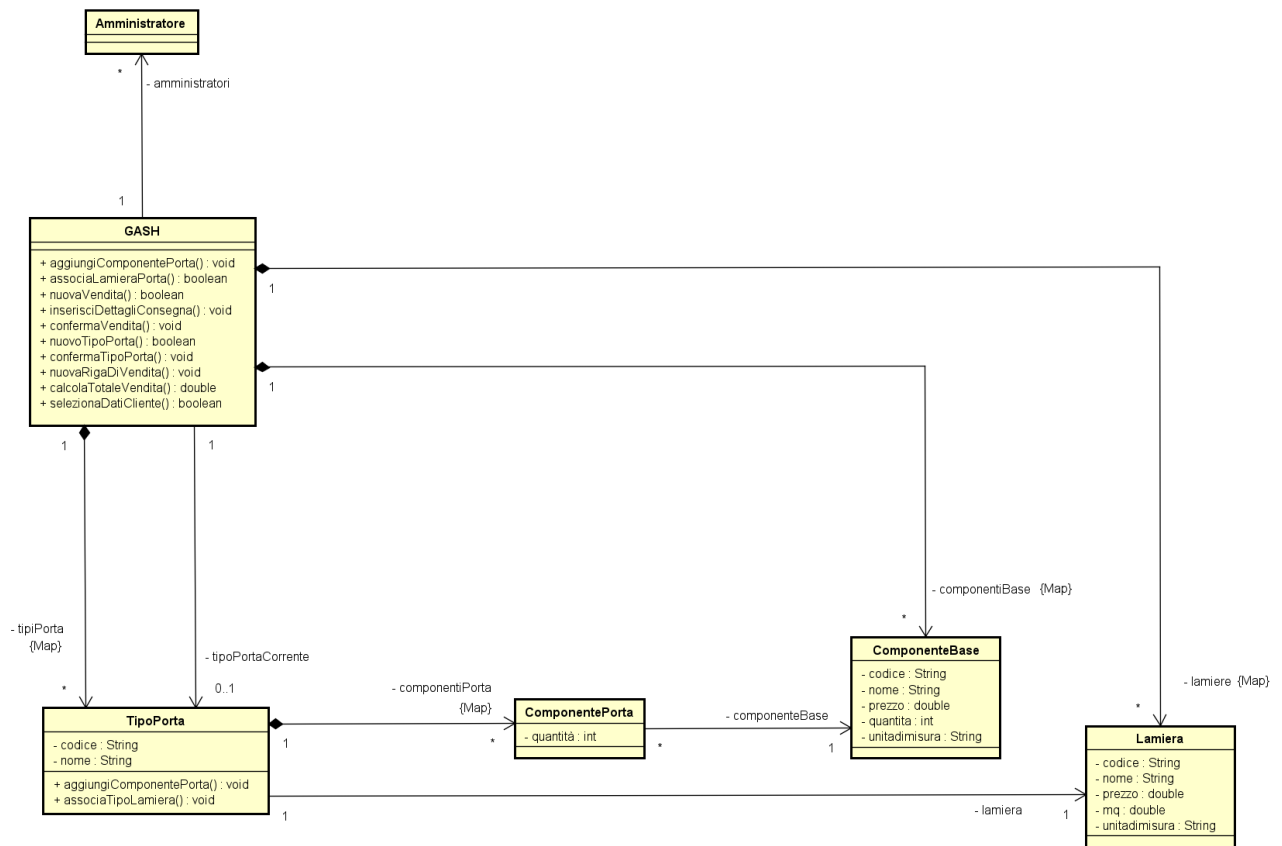
### 1.3.2.3 associaLamieraPorta(codiceLamiera:String):Boolean



### 1.3.2.4 confermaTipoPorta()

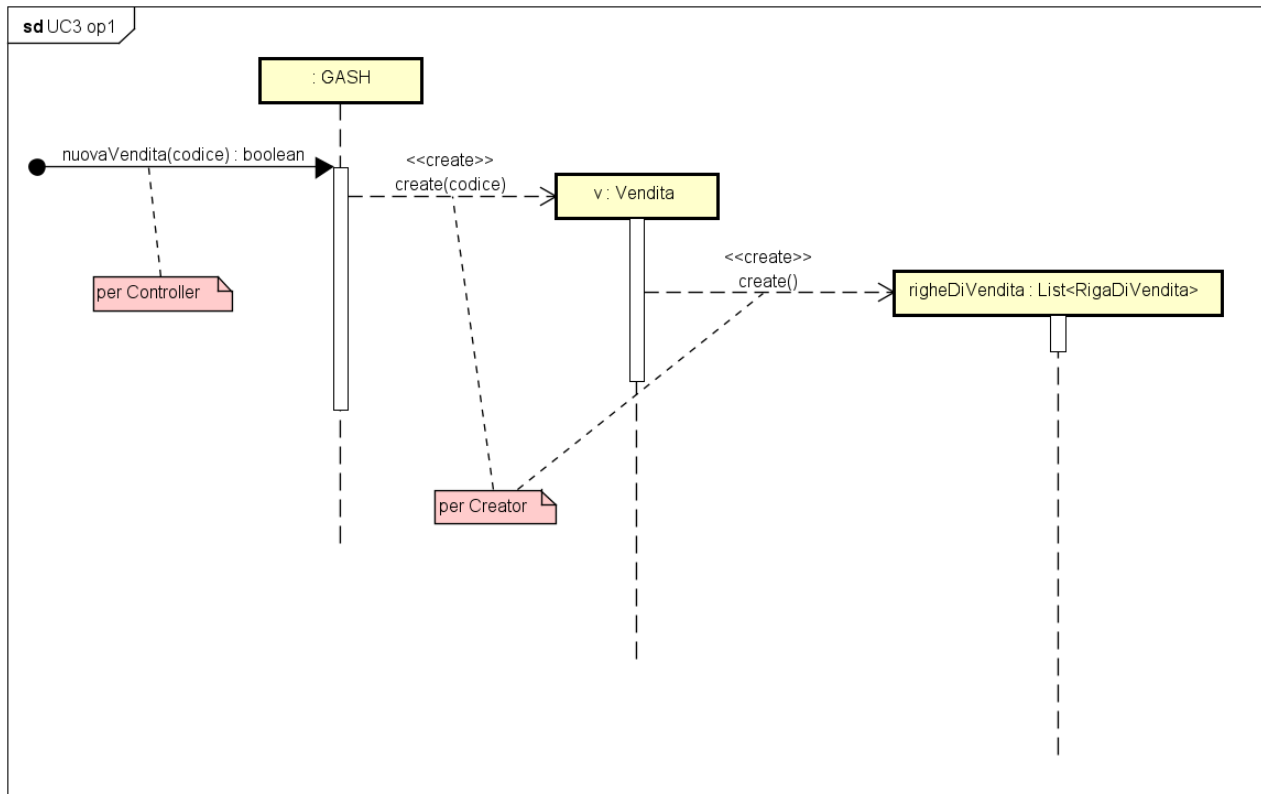


### 1.3.3 Caso d'uso UC2, Diagramma delle classi di progetto



## 1.3.4 Caso d'uso UC3, Diagrammi di interazione

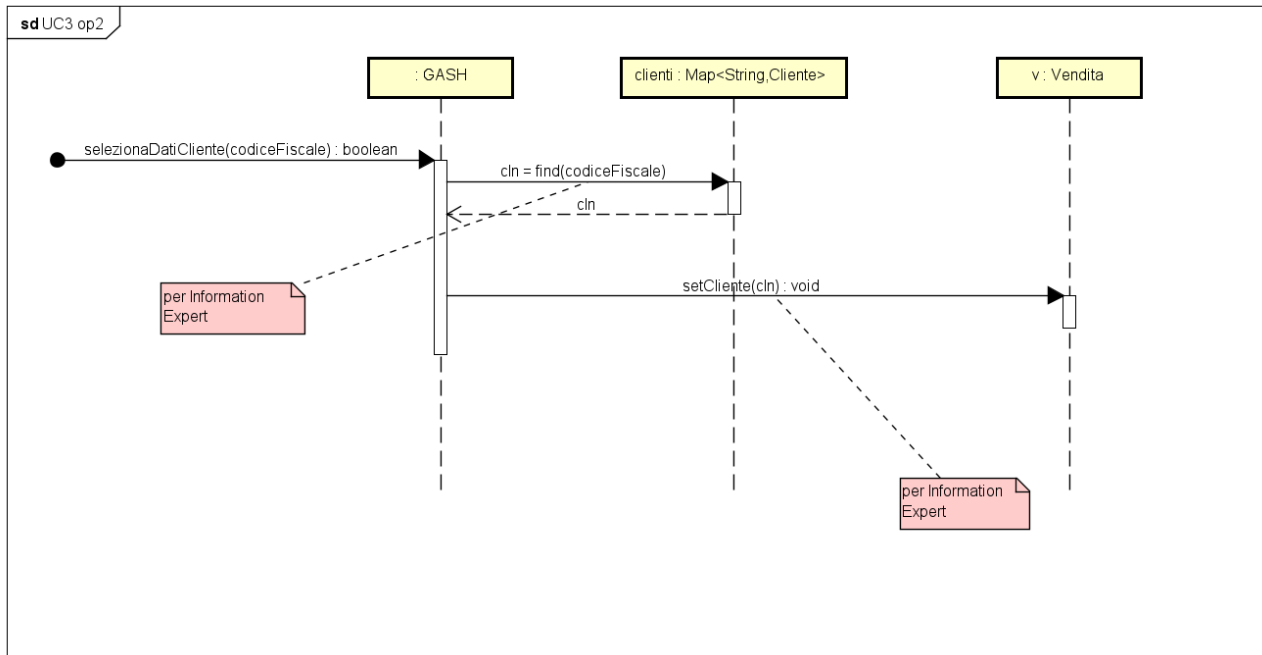
### 1.3.4.1 nuovaVendita(codice:String):Boolean



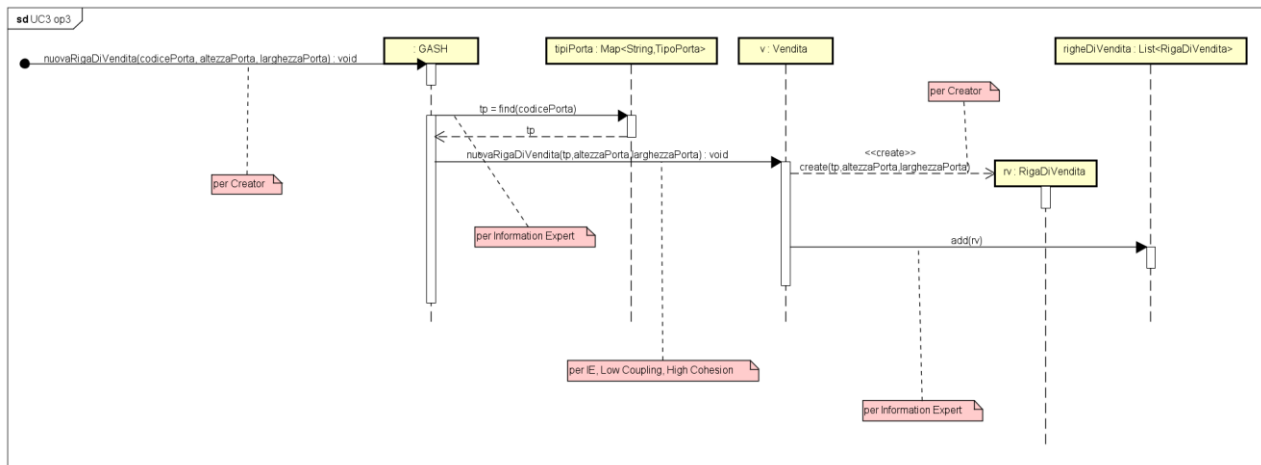
L'operazione `nuovaVendita` crea un'istanza `v` di `Vendita` che sarà la vendita corrente, se esiste una `Vendita` con quel codice l'operazione non istanzierà l'oggetto e ritornerà `false`.

Il ritorno dell'operazione di tipo booleano servirà all'interfaccia grafica per mostrare un errore opportuno e annullare l'inserimento della nuova vendita.

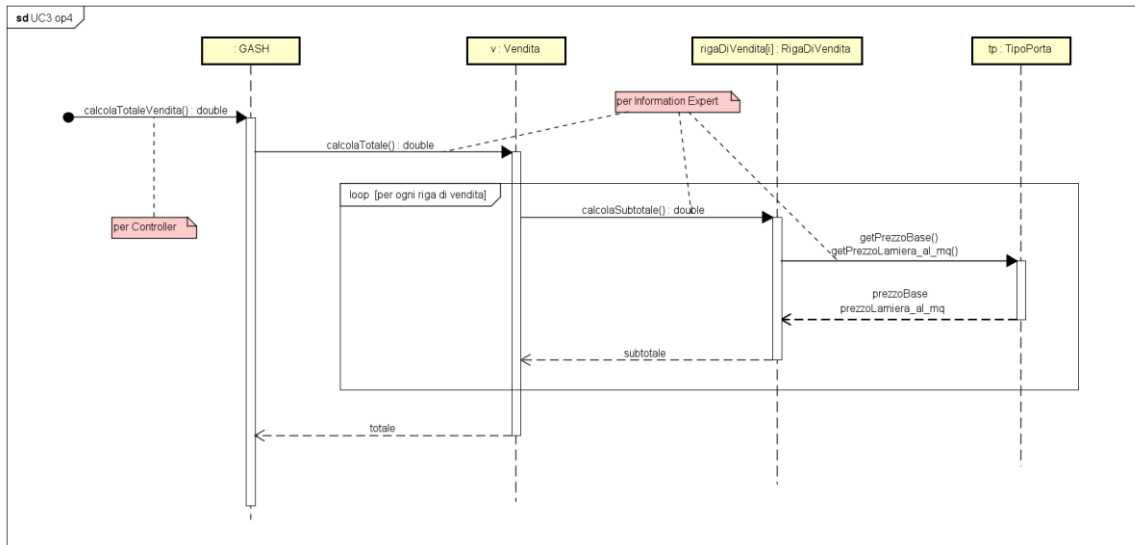
### 1.3.4.2 selezionaDatiCliente(codiceFiscale:String):Boolean



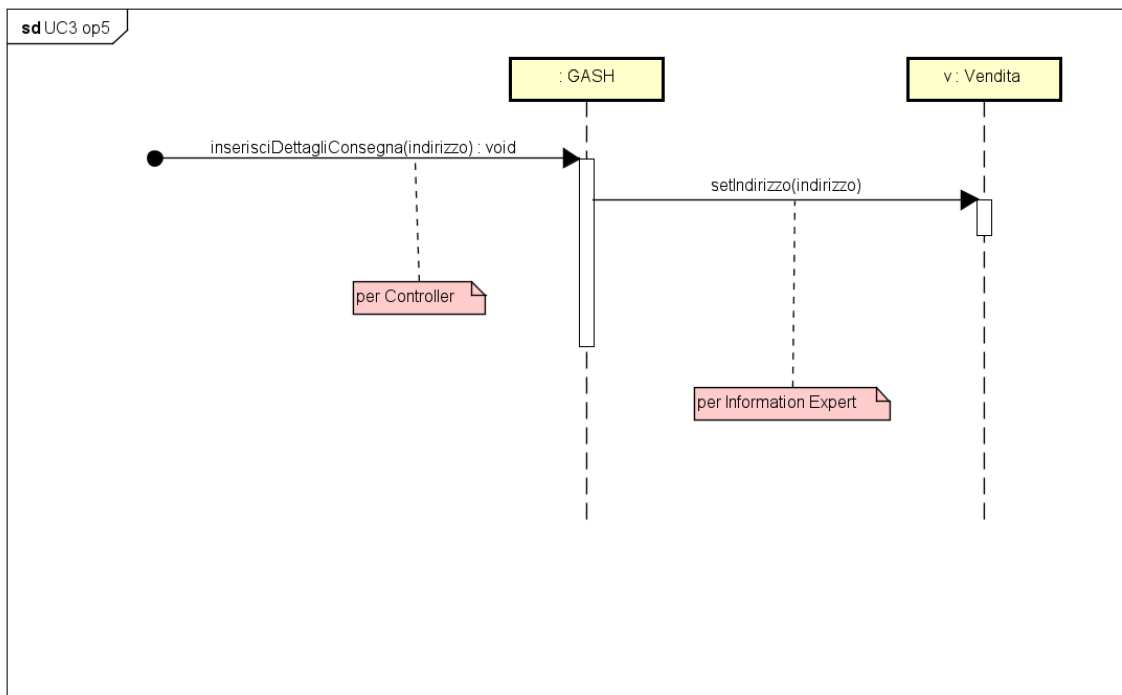
### 1.3.4.3 nuovaRigaDiVendita(codicePorta:String, altezzaPorta:Float, larghezzaPorta:Float)



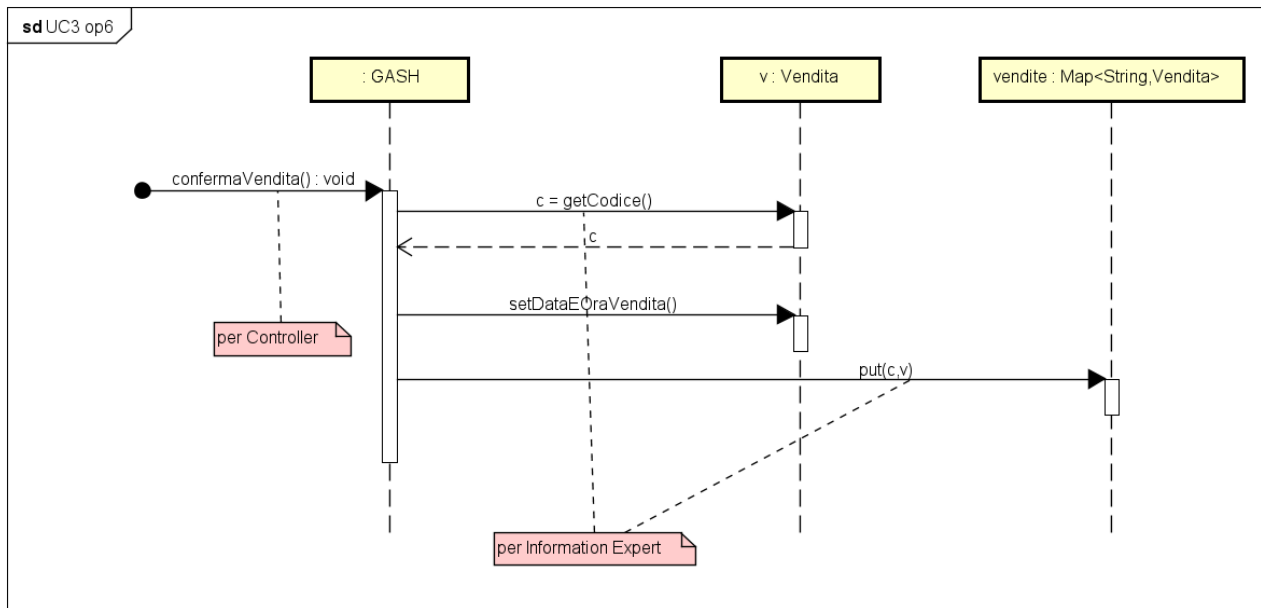
### 1.3.4.4 calcolaTotaleVendita()



### 1.3.4.5 inserisciDettagliConsegna(indirizzo:String)



### 1.3.4.6 confermaVendita()

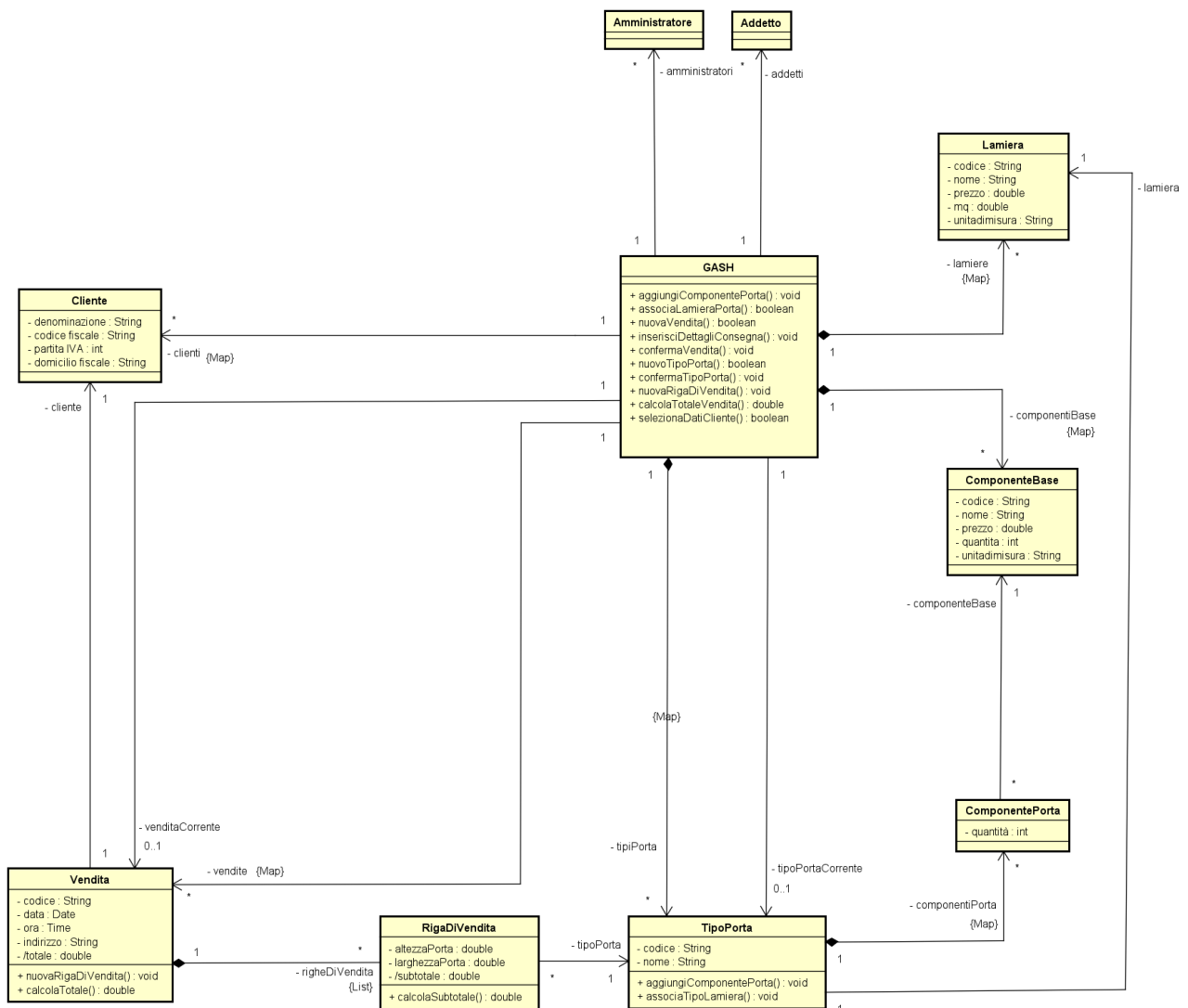


Con l'esecuzione di `confermaVendita()` verranno aggiornati gli attributi quantità dei Componenti utilizzati per costruire le Porte di garage vendute. I dettagli di quest'operazione verranno trattati nel codice delle successive iterazioni.

Il metodo `setDataEOraVendita()` imposterà la data e ora attuale della vendita mediante una funzione di una libreria java (`LocalDateTime`).

Tutte le vendite verranno registrate in un file di testo.

### 1.3.5 Caso d'uso UC3, Diagramma delle classi di progetto



# 1.4 Iterazione 1, Refactoring

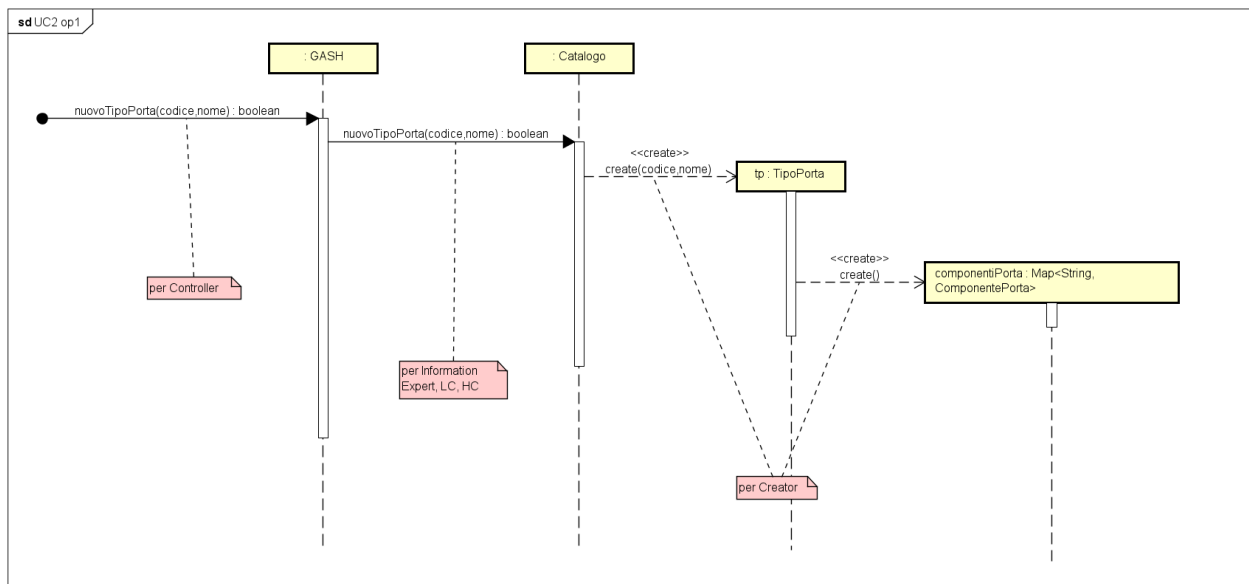
## 1.4.1 Introduzione

Dall'analisi del codice del progetto, si evince che la classe controller GASH ha responsabilità eccessive, che possono essere risolte introducendo una classe Catalogo, assegnando a quest'ultima la responsabilità di conoscere i tipi di porta per garage e i componenti. Quest'attività viene svolta nel codice, attraverso una successione di *refactoring*, con l'obiettivo di cambiare l'organizzazione del codice senza alterarne il comportamento complessivo. Nello svolgere quest'attività, viene anche scelto di applicare il design pattern Singleton (GoF) al controller, GASH.

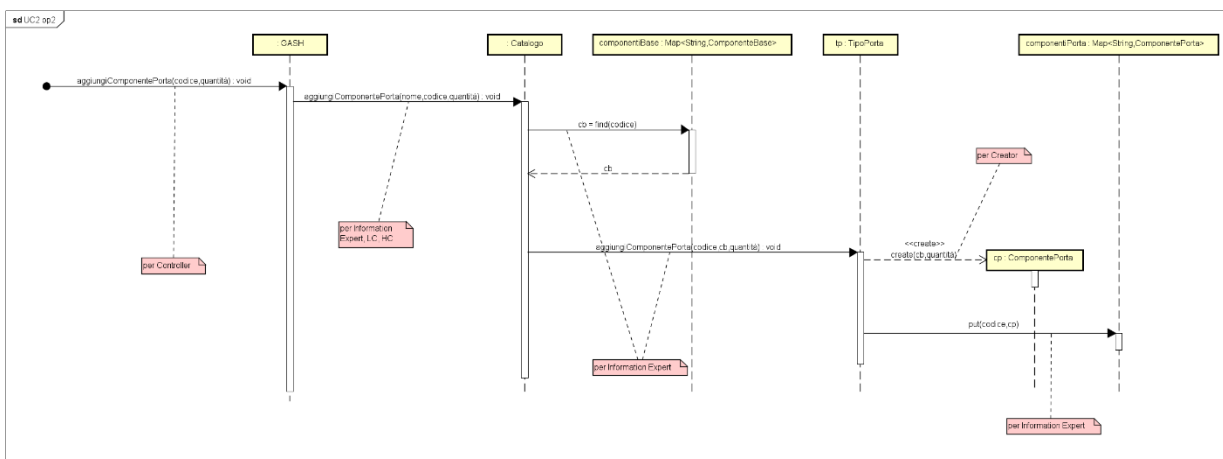
## 1.4.2 Caso d'uso UC2

L'aggiunta di una classe Catalogo, che gestisce le associazioni verso le classi TipoPorta, ComponenteBase e Lamiera, modifica tutti i diagrammi di interazione relativi al caso d'uso UC2.

### 1.4.2.1 nuovoTipoPorta(codice:String,nome:String): boolean

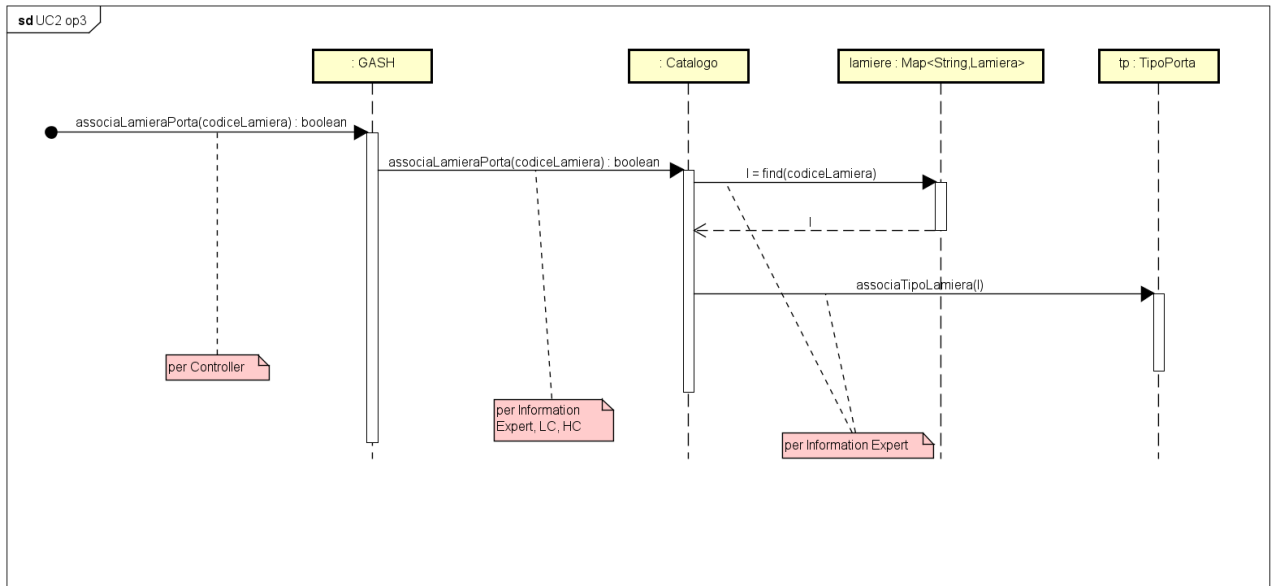


### 1.4.2.2 aggiungiComponentePorta(codice:String,quantita:int)

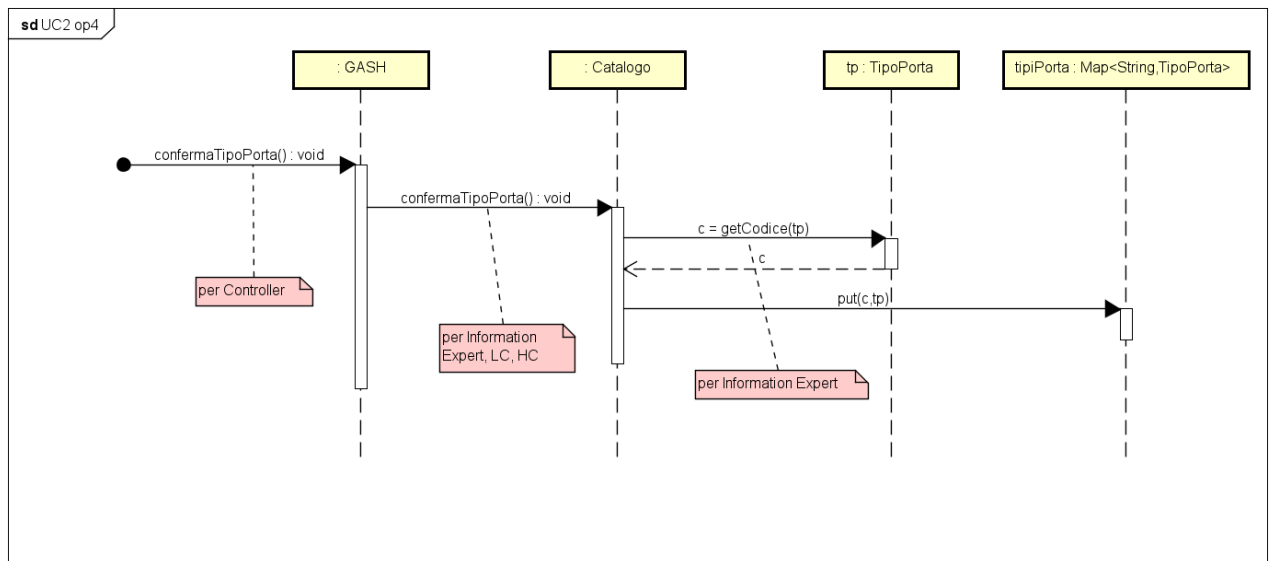




### 1.4.2.3 associaLamieraPorta(codice:Lamiera): boolean



### 1.4.2.4 confermaTipoPorta()

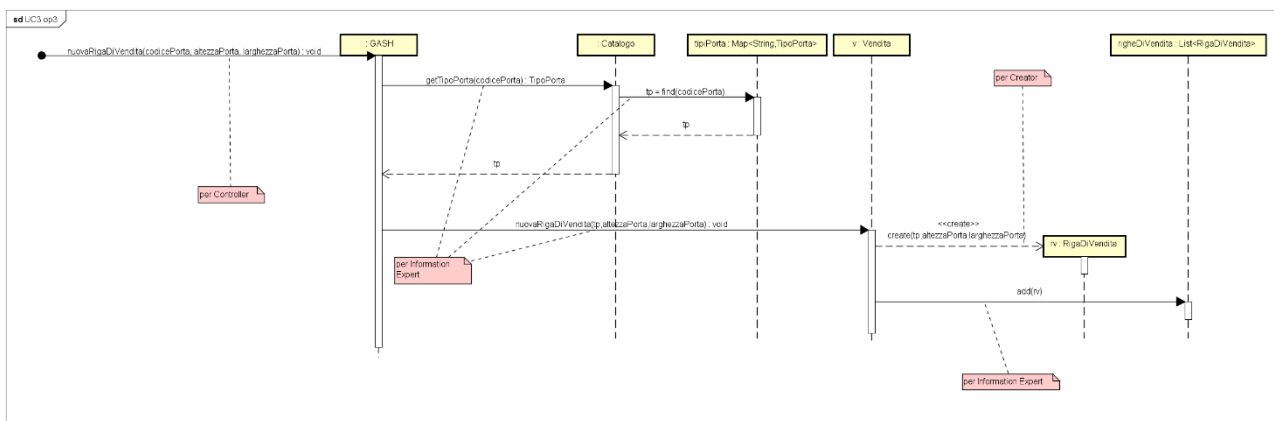


### 1.4.3 Caso d'uso UC3

L'inserimento della classe Catalogo influenza in maniera minore le operazioni di sistema relative al caso d'uso UC3. L'unico diagramma di interazione che subisce dei cambiamenti è quello relativo all'operazione di sistema nuovaRigaDiVendita.

#### 1.4.3.1

#### nuovaRigaDiVendita(codicePorta:String,altezzaPorta:Float,larghezzaPorta:Float)



## 1.4.4 Diagramma delle classi di progetto

La relazione tra GASH e Catalogo è mostrata come una dipendenza e non come un'associazione, in quanto Catalogo viene acceduto come Singleton.

