

**Progetto:**

## **Paninaro Next**

**Titolo del documento:**

## **Documento di Architettura**

### **INDICE**

Scopo del documento	1
1. Diagramma delle classi	2
2. Codice in Object Constraint Language	4
3. Diagramma delle classi con codice OCL	<b>Error! Bookmark not defined.</b>

### **Scopo del documento**

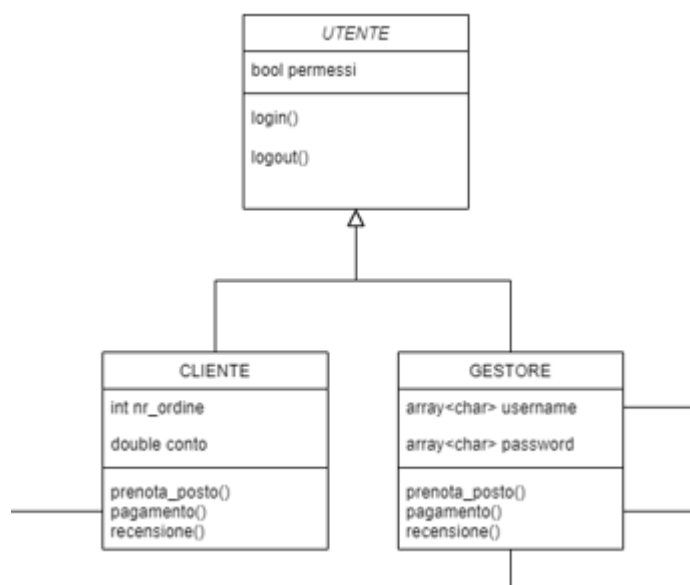
Il presente documento riporta la definizione dell'architettura del progetto Paninaro Next usando diagrammi delle classi in Unified Modeling Language (UML) e codice in Object Constraint Language (OCL). Nel precedente documento è stato presentato il diagramma degli use case, il diagramma di contesto e quello dei componenti. Ora, tenendo conto di questa progettazione, viene definita l'architettura del sistema dettagliando da un lato le classi che dovranno essere implementate a livello di codice e dall'altro la logica che regola il comportamento del software. Le classi vengono rappresentate tramite un diagramma delle classi in linguaggio UML. La logica viene descritta in OCL perché tali concetti non sono esprimibili in nessun altro modo formale nel contesto di UML.

# 1. Diagramma delle classi

Nel presente capitolo vengono presentate le classi previste nell'ambito del progetto Paninaro Next. Ogni componente presente nel diagramma dei componenti diventa una o più classi. Tutte le classi individuate sono caratterizzate da un nome, una lista di attributi che identificano i dati gestiti dalla classe e una lista di metodi che definiscono le operazioni previste all'interno della classe. Ogni classe può essere anche associata ad altre classi e, tramite questa associazione, è possibile fornire informazioni su come le classi si relazionano tra loro.

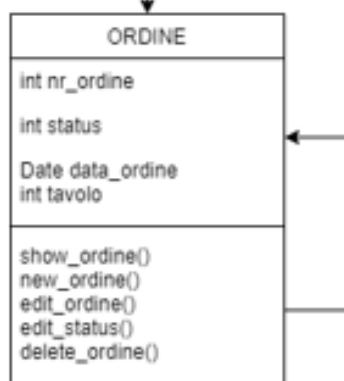
Riportiamo di seguito le classi individuate a partire dai diagrammi di contesto e dei componenti. In questo processo si è proceduto anche nel massimizzare la coesione e minimizzare l'accoppiamento tra classi.

## 1.1. Utente: cliente



Analizzando l'immagine si nota che la classe UTENTE ha 2 sottoclassi, che entrambe ereditano l'attributo permessi e le funzioni login() e logout(). Esse si distinguono in base a questo attributo. CLIENTE avrà come attributi il numero dell'ordine e il conto, mentre il gestore l'username e password per accedere all'area riservata.

## 1.2. Ordini



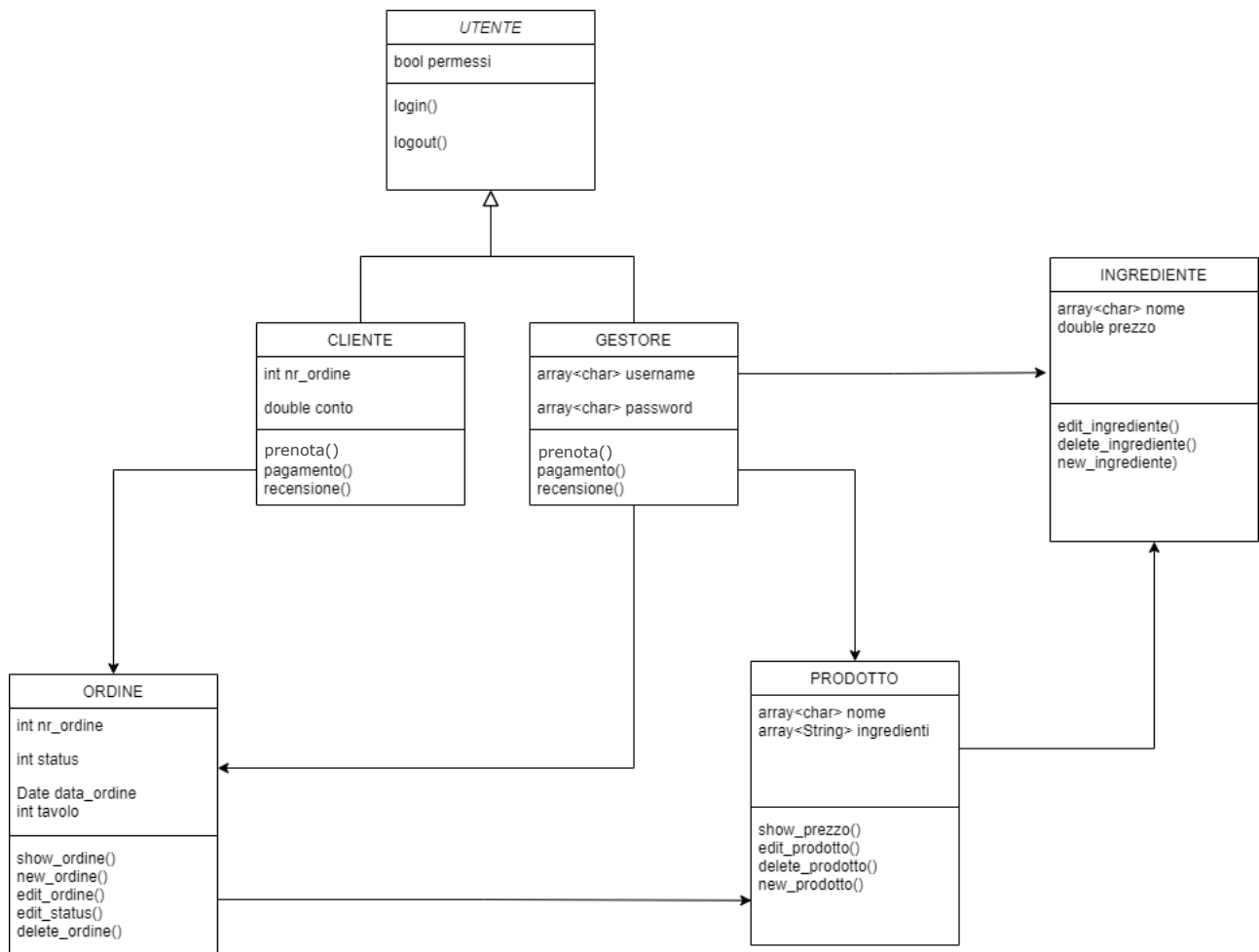
Per rappresentare gli ordini useremo la classe ORDINE, che come attributo avrà il numero dell'ordine, lo status, la data dell'ordine e il numero del tavolo.

### 1.3. Prodotti e Ingredienti



In **Ingrediente** è presente il nome e il prezzo, con dei metodi per modificare, eliminare e aggiungere gli ingredienti. In **Prodotto** è invece presente il nome e la lista degli ingredienti da cui è composto, un metodo per calcolarne il prezzo dinamicamente e altri per modificare, eliminare e aggiungere prodotti. Infine, per accedere a queste classi sono necessari i permessi da amministratore.

### 1.4. Diagramma delle classi complessivo



## 2. Codice in Object Constraint Language

In questo capitolo è descritta in modo formale la logica prevista nell'ambito di alcune operazioni di alcune classi. Tale logica viene descritta in Object Constraint Language (OCL) perché tali concetti non sono esprimibili in nessun altro modo formale nel contesto di UML.

### 1. Cliente e ordine

Quando un cliente fa un ordine questo deve avere lo stesso numero;

```
context Cliente::prenota_posto()
post : self.nr_ordine= Ordine.nr_ordine
```

### 2. Prodotto

Il prezzo di ogni prodotto deve essere strettamente maggiore di 0. E per modificare, aggiungere o cancellare un prodotto servono gli adeguati permessi;

```
context Prodotto
inv : prezzo_prodotto > 0.00
```

```
context Prodotto::new_prodotto()
pre : permessi = true
```

```
context Prodotto::edit_nome()
pre : permessi = true
```

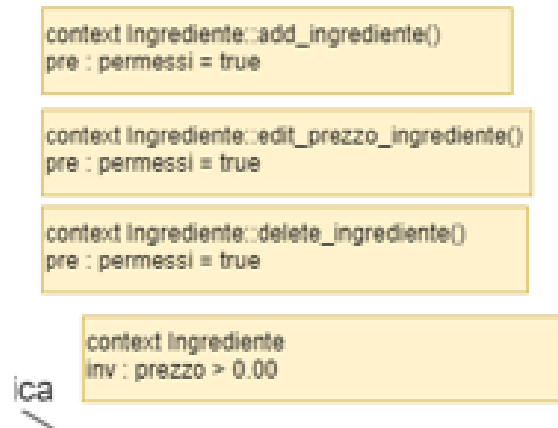
```
context Prodotto::delete_prodotto()
pre : permessi = true
```

```
context Prodotto::edit_ingredienti()
pre : permessi = true
```

```
context Prodotto::edit_prezzoProdotto()
pre : permessi = true
```

### 3. Ingredienti

Per aggiungere, modificare o cancellare un prodotto sono necessari i permessi di amministratore. In aggiunta, il prezzo di ogni ingrediente deve essere strettamente maggiore di 0;



#### 4. Gestore

È possibile vedere il resoconto dell'attività solo se si possiedono i permessi.

