



SAPIENZA  
UNIVERSITÀ DI ROMA

## Una applicazione web su cloud per il supporto al Banco Alimentare

Facoltà Ingegneria dell'informazione, informatica e statistica  
Corso di Laurea in Informatica

Candidato

Alessio Civica  
Matricola 1916744

Relatore

Andrea Sterbini

Estensione Anno Accademico 2021/2022

---

**Una applicazione web su cloud per il supporto al Banco Alimentare**  
Tesi di Laurea. Sapienza – Università di Roma

© 2022 Alessio Civica. Tutti i diritti riservati

Questa tesi è stata composta con L<sup>A</sup>T<sub>E</sub>X e la classe Sapthesis.

Email dell'autore: [civica.1916744@studenti.uniroma1.it](mailto:civica.1916744@studenti.uniroma1.it)

*A mio nonno Armando*

# Indice

|  |           |
|--|-----------|
| <b>1 Introduzione</b>  | <b>2</b>  |
| 1.1 Obiettivo della relazione . . . . .  | 2         |
| 1.2 Scopo della Web App . . . . .  | 2         |
| 1.2.1 Statistiche riguardante le donazioni alimentari online e il loro bisogno . . . . .         | 3         |
| 1.3 Tecnologie utilizzate . . . . .  | 4         |
| 1.4 Vantaggi del Cloud . . . . .   | 5         |
| 1.4.1 I vari modelli di business e quale è stato scelto . . . . .                                | 5         |
| 1.4.2 Vantaggi del non avere una struttura fisica . . . . .                                      | 5         |
| 1.4.3 Sicurezza . . . . .  | 5         |
| 1.4.4 Continuità del servizio . . . . .  | 6         |
| 1.4.5 Risparmio energetico e sostenibilità . . . . .   | 6         |
| <b>2 Descrizione della Base di dati iniziale e Analisi concettuale</b>                           | <b>7</b>  |
| 2.1 Base di dati iniziale . . . . .  | 7         |
| 2.2 Raffinamento dei Requisiti per la ristrutturazione . . . . .                                 | 7         |
| 2.3 Diagramma ER . . . . .   | 9         |
| 2.4 Glossario dei domini . . . . .   | 10        |
| 2.5 Definizione informale dei vincoli esterni . . . . .  | 11        |
| 2.6 Dizionario dei dati . . . . .  | 12        |
| <b>3 Analisi degli Use-Case</b>  | <b>14</b> |
| 3.1 Diagramma UML degli Use-Case . . . . .   | 14        |
| 3.2 Specifiche concettuali informali degli Use-Case . . . . .                                    | 15        |
| <b>4 Schema Relazionale della Base Dati</b>  | <b>18</b> |
| 4.1 Diagramma ER ristrutturato . . . . .   | 18        |
| 4.2 Definizione delle Relazioni Derivanti da Entità e Relationship Accorpate ad Entità . . . . . | 20        |
| 4.3 Progettazione dei vincoli esterni . . . . .  | 21        |
| 4.4 Realizzazione degli use-case . . . . .   | 22        |
| <b>5 Descrizione del sito con screenshot</b>   | <b>24</b> |
| 5.1 Dona . . . . .   | 24        |
| 5.2 Cronologia . . . . .   | 27        |
| 5.3 Home Section . . . . .   | 27        |
| 5.4 Login, Registrazione, Modifica e Disconnessione . . . . .                                    | 31        |

|   |           |
|---|-----------|
| <b>6 Problematiche affrontate per implementare il sito su Microsoft Azure</b>                               | <b>34</b> |
| 6.1 Studio del portale . . . . .  | 34        |
| 6.2 sistema per rilasciare aggiornamenti nel Web App immediati e come collegare i vari componenti . . . . . | 35        |
| 6.3 Riepilogo dei costi mensili . . . . .   | 36        |
| <b>7 Conclusioni</b>  | <b>37</b> |
| 7.1 Tecnologie interessanti implementate . . . . .  | 37        |
| 7.1.1 Implementazione dei COOKIES . . . . .   | 37        |
| 7.1.2 SQL injection . . . . .   | 37        |
| 7.1.3 Responsiveness . . . . .  | 38        |
| 7.1.4 Validazione form utente lato client . . . . .   | 38        |
| 7.1.5 Implementazione dei Meta tags . . . . .   | 38        |
| 7.2 Tecnologie future . . . . .   | 39        |
| 7.2.1 Implementazione della verifica in 2 passaggi . . . . .  | 39        |
| 7.2.2 Recupero della password . . . . .   | 39        |
| 7.2.3 Algoritmo per il calcolo della destinazione migliore . . . . .  | 39        |
| 7.2.4 Algoritmo per il calcolo del locker più vicino . . . . .  | 39        |
| 7.2.5 implementazione CAPTCHA . . . . .   | 40        |
| 7.2.6 pop-up per la privacy e la cookies policy . . . . .   | 40        |
| 7.2.7 accessibilità . . . . .   | 40        |
| 7.2.8 Lingua . . . . .  | 41        |
| <b>Sitografia</b>   | <b>42</b> |

# Elenco delle figure

|      |  |    |
|------|--|----|
| 2.1  | Base di dati iniziale.                                 | 7  |
| 2.2  | Diagramma ER   | 10 |
| 4.1  | Diagramma ER ristrutturato                             | 19 |
| 5.1  | Sezione inserimento prodotto.                          | 26 |
| 5.2  | Sezione indirizzo di consegna.                         | 26 |
| 5.3  | Cronologia dei prodotti Alimentari donati dall'utente. | 27 |
| 5.4  | Header del sito.                                       | 28 |
| 5.5  | Sezione statistiche.                                   | 29 |
| 5.6  | Visualizzazione di una Sezione                         | 30 |
| 5.7  | Sezione in formato Card                                | 30 |
| 5.8  | Sezione I nostri obiettivi tramite semi-donut.         | 31 |
| 5.9  | Login.   | 32 |
| 5.10 | Modifica credenziali dell'utente.                      | 32 |
| 6.1  | Sezione inserimento prodotto.                          | 36 |

# Capitolo 1

## Introduzione

### 1.1 Obiettivo della relazione

L'obiettivo della seguente relazione è la realizzazione e la presentazione di una Web App in Cloud.

L'utente dovrà registrarsi, inserendo informazioni personali, ed effettuare il login, per poter in un secondo momento donare prodotti alimentari. Rimane comunque possibile modificare i propri dati personali in qualunque momento.

Il primo passo per effettuare una donazione è quello di scrivere il nome, selezionare la data di scadenza e la tipologia dell'alimento che si intende devolvere.

Successivamente è possibile impostare il relativo indirizzo di ritiro (presso un Locker o presso un indirizzo privato) e la relativa data di ritiro. È possibile inoltre visualizzare la propria cronologia delle donazioni effettuate o già programmate.

Nella Home sono presenti, per utenti registrati e non, i dati statistici della Web App:

- Quantità di donazioni effettuate.
- numero di donazioni effettuate.
- numero di utenti attivi.
- la quantità di prodotti alimentari che sono stati donati rispetto agli obiettivi preposti dai gestori della Web App.

### 1.2 Scopo della Web App

Negli ultimi anni, sempre più persone si sono rivolte al mondo online per effettuare donazioni alimentari. Questo è dovuto alla necessità di fornire cibo a coloro che ne hanno bisogno, ma anche al desiderio di utilizzare la tecnologia per raggiungere un pubblico più ampio.

Le donazioni alimentari online sono diventate particolarmente importanti durante la pandemia da COVID-19, quando molte persone hanno perso il lavoro o sono state costrette a restare a casa a causa delle restrizioni. In molte parti del mondo, le organizzazioni che si occupano di distribuire cibo ai bisognosi hanno dovuto aumentare la propria attività, per rispondere all'incremento della domanda. Le donazioni alimentari online sono state un modo per far fronte a questa emergenza, fornendo cibo a chi ne ha bisogno, gestendo il tutto anche da remoto.

Lo scopo della Web App è l'incremento dell'efficienza e alla sostenibilità per la donazione di alimenti, anche in eccesso da parte di: ristoranti, supermercati, e

privati. Lo scopo è quindi quello di indirizzare l'attenzione sulla questione dello spreco alimentare e supportare l'obiettivo di una società più sostenibile riguardo lo spreco alimentare e solidale verso le zone più bisognose.

Con questa Web App non si andrebbero ovviamente ad eliminare i problemi dello spreco alimentare e della fame nel mondo, ma anche se in minima parte, diminuirebbero tali problemi. Inoltre, non è da sottovalutare, il miglioramento che può apportare la categorizzazione degli alimenti fatta direttamente dagli utenti e non eseguita in un secondo momento da un addetto.

Per il momento i prodotti inseriti dagli utenti nella Web App possono essere spediti da tutta Italia in un unico punto di interesse prestabilito dal gestore del sito, che accordatosi con una associazione no profit (es. Croce Rossa, Banco Alimentare, Caritas, ecc...) potrà ricevere i prodotti alimentari in tutto il continente.

In futuro, a seguito di un eventuale crescita di donatori in questa Web App, si potrebbe pensare di distribuire alimenti agli indigenti in ogni città; per il momento i gestori della Web App affidano tale distribuzione alle grandi associazioni no profit.

### **1.2.1 Statistiche riguardante le donazioni alimentari online e il loro bisogno**

Le donazioni di cibo online sono una tendenza relativamente nuova e ci sono poche statistiche disponibili su di esse. Tuttavia, ci sono alcune informazioni generali che possono aiutare a capire il loro impatto:

- Secondo una ricerca del 2020 condotta dalla piattaforma di raccolta fondi online "Classy", il numero di campagne di raccolta fondi per la distribuzione di cibo è aumentato del 42% rispetto all'anno precedente.
- Una ricerca del 2019 condotta dal "Digital Giving Index" ha rilevato che il 49% dei donatori negli Stati Uniti aveva donato cibo online, mentre il 45% aveva donato in modo tradizionale.
- In Australia, l'organizzazione "OzHarvest", che si occupa di distribuire cibo agli indigenti, ha lanciato una piattaforma di donazione alimentare online chiamata "Rescue Food". Nel 2020, la piattaforma ha raccolto circa 1,5 milioni di pasti.
- In generale, le donazioni di cibo online sembrano essere particolarmente utili in situazioni di emergenza, come la pandemia da COVID-19. Ad esempio, negli Stati Uniti, l'organizzazione "Feeding America" ha lanciato una piattaforma di donazione alimentare online chiamata "COVID-19 Response Fund", che ha raccolto oltre 100 milioni di dollari.
- Secondo una relazione del 2021 del Programma delle Nazioni Unite per lo sviluppo (UNDP), la pandemia da COVID-19 ha fatto aumentare la povertà estrema nel mondo del 7% nel 2020, portando a circa 120 milioni di persone in più che vivono in povertà rispetto all'anno precedente.
- In Europa, secondo una relazione del 2021 dell'Organizzazione per la cooperazione e lo sviluppo economico (OCSE), il tasso di povertà relativa è aumentato dal 15,7% nel 2019 al 17,1% nel 2020. Tuttavia, ci sono stati grandi differenze tra i paesi, con un aumento maggiore nei paesi del sud Europa.

In Italia, secondo una relazione del 2022 dell'ISTAT, l'istituto nazionale di statistica italiano, sono in condizione di povertà assoluta poco più di 1,9 milioni di famiglie e circa 5,6 milioni di individui (9,4% come l'anno precedente).

Per la povertà relativa l'incidenza sale all'11,1% (da 10,1% del 2020) e le famiglie sotto la soglia sono circa 2,9 milioni (2,6 milioni nel 2020). In particolare, l'aumento della povertà è stato maggiore nelle regioni del sud Italia. Alcune organizzazioni come il Banco Alimentare hanno avviato iniziative per accettare donazioni di cibo online ma si è ancora agli inizi. Tuttavia, non sono ancora disponibili dati precisi sulle quantità raccolte attraverso questi canali.

### 1.3 Tecnologie utilizzate

Linguaggi e Framework utilizzati:

- **HTML (HyperText Markup Language):**

linguaggio di markup utilizzato per la creazione di pagine web. Si tratta di un linguaggio di programmazione a livello di testo che definisce la struttura e il contenuto delle pagine web ovvero tramite sistema di etichette o tag che vengono inseriti nel codice sorgente per definire la struttura e il contenuto delle pagine web.

- **CSS(Cascading Style Sheet):**

linguaggio di styling utilizzato per separare la struttura del contenuto delle pagine web (definita da HTML) dal loro design e stile. Questo rende il processo di sviluppo e manutenzione del sito web più efficiente e flessibile.

- **Bootstrap:**

framework front-end per la creazione di siti web e applicazioni web. Si tratta di un insieme di strumenti, componenti e modelli pre-costruiti che possono essere utilizzati per creare rapidamente una struttura di base per un sito web, è basato su HTML, CSS e JavaScript.

- **PHP(Hypertext Preprocessor):**

linguaggio di programmazione server-side utilizzato per lo sviluppo di siti web dinamici. A differenza dei linguaggi di programmazione client-side come HTML, CSS e JavaScript che vengono eseguiti sul browser dell'utente, il codice PHP viene eseguito sul server e la sua visualizzazione viene inviata al browser sotto forma di HTML.

- **JavaScript:**

linguaggio di programmazione client-side utilizzato per creare siti web interattivi e applicazioni web. JavaScript viene eseguito sul browser dell'utente, il che significa che tutto il codice JavaScript viene eseguito sul computer dell'utente e non sul server.

- **PostgreSQL:**

sistema di gestione di database relazionali open source. È stato progettato per fornire una solida architettura di database per applicazioni aziendali, scientifiche e web, ed è utilizzato da molte aziende e organizzazioni in tutto il mondo.

Piattaforme utilizzate:

- Microsoft Azure:

piattaforma cloud che offre una vasta gamma di servizi per la creazione, la gestione e lo sviluppo di applicazioni, tra cui la virtualizzazione delle macchine, l'archiviazione dei dati, gestione di webapps e molto altro

- GitHub:

piattaforma web per la gestione del codice sorgente e la collaborazione sul software. È stato progettato per aiutare gli sviluppatori a gestire il loro codice anche condividerlo con altri

## 1.4 Vantaggi del Cloud

### 1.4.1 I vari modelli di business e quale è stato scelto

Le tipologie di modelli più comuni e utilizzate nel Cloud Computing sono:

- IaaS (Infrastructure as a Service):

Modello di servizi cloud che offre accesso a infrastrutture informatiche, come server virtuali, storage, rete e sicurezza.

- PaaS (Platform as a Service):

Modello di servizi cloud che offre una piattaforma per lo sviluppo e la distribuzione di applicazioni, senza dover gestire l'infrastruttura sottostante.

- SaaS (Software as a Service):

Modello di servizi cloud che offre accesso a software o applicazioni su internet. L'utente può utilizzare le applicazioni di un provider di servizi cloud senza doverle installare sul loro computer o gestirne la manutenzione.

Il modello che ho utilizzato è PAAS (Platform as a Service), in quanto impostando i requisiti hardware che software del server e della web-app ho voluto occuparmi principalmente dello sviluppo del Database e del codice per la web-app al fine di avere i vantaggi offerti da Microsoft Azure che elencherò in sottoparagrafi di seguito.

### 1.4.2 Vantaggi del non avere una struttura fisica

Non necessita di particolari conoscenza della struttura hardware e software utilizzata. In breve, con il Cloud, non si ha la necessità di acquistare, mantenere o risolvere problematiche hardware, come server o unità di archiviazione, poiché queste risorse sono fornite da Microsoft Azure sempre aggiornate e sicure.

### 1.4.3 Sicurezza

La sicurezza che si ha tramite Microsoft Azure sono le seguenti:

- Controllo degli accessi:

Microsoft Azure utilizza l'autenticazione a più fattori e le autorizzazioni basate su ruoli per garantire che solo gli utenti autorizzati possano accedere ai dati e alle risorse.

- **Protezione dei dati:**

Microsoft Azure utilizza tecnologie avanzate di crittografia per proteggere i dati in riposo, in transito e durante l'elaborazione.

- **Monitoring e rilevamento delle minacce:**

Microsoft Azure utilizza tecnologie avanzate di rilevamento delle minacce e di sicurezza per monitorare continuamente l'ambiente alla ricerca di attività sospette e per prevenire eventuali violazioni della sicurezza.

#### **1.4.4 Continuità del servizio**

La continuità del servizio è un aspetto critico nell'utilizzo del cloud, in quanto garantisce la disponibilità continua delle applicazioni e dei dati. In Microsoft Azure, la continuità del servizio è garantita attraverso le seguenti misure:

- **Disaster Recovery:**

Microsoft Azure offre soluzioni di Disaster Recovery per aiutare a proteggere i dati e le applicazioni da eventuali interruzioni del servizio o perdite di dati. Queste soluzioni includono opzioni come la replicazione dei dati in una seconda regione e la creazione di immagini di ripristino per le applicazioni.

- **Monitoraggio e manutenzione proattiva:**

Microsoft Azure utilizza tecnologie avanzate di monitoraggio e manutenzione proattiva per identificare e correggere eventuali problemi prima che influiscano sulla disponibilità del servizio.

- **Architettura a livello di servizio:**

Microsoft Azure è progettato per gestire automaticamente la distribuzione e il ripristino dei servizi in caso di problemi, garantendo la continuità del servizio per le applicazioni e i dati.

#### **1.4.5 Risparmio energetico e sostenibilità**

- **Energia rinnovabile:**

Microsoft Azure si impegna a raggiungere il 100% di energia rinnovabile per la sua rete globale di data center entro il 2025.

- **Recupero di calore:**

Microsoft Azure utilizza il recupero di calore dai data center per alimentare il riscaldamento e il raffreddamento degli edifici circostanti.

- **Ottimizzazione dell'utilizzo dei server:**

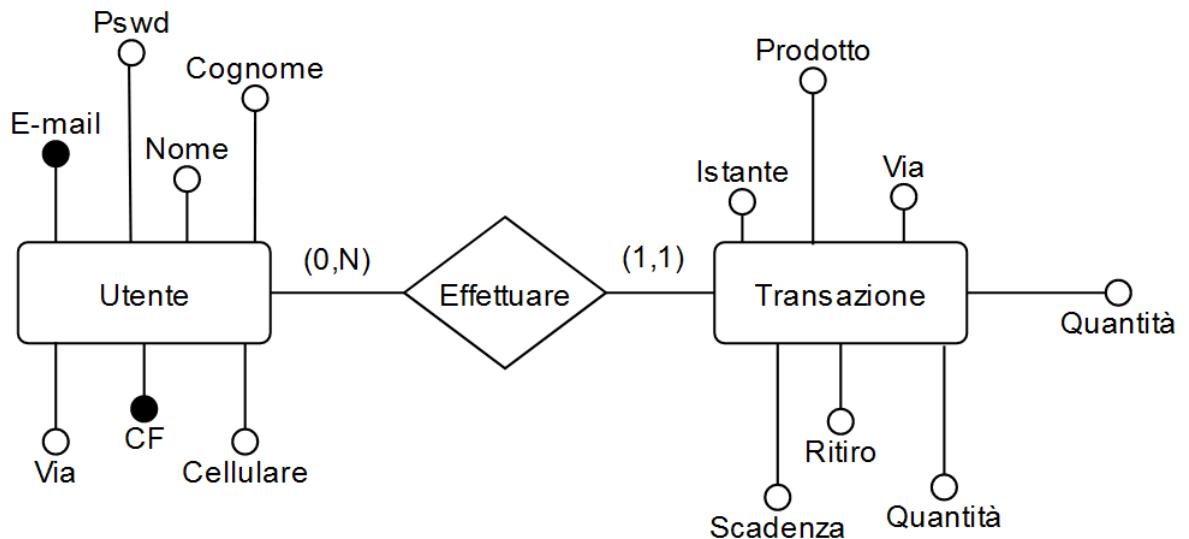
I provider di cloud utilizzano tecnologie di virtualizzazione e di gestione della capacità per ottimizzare l'utilizzo dei server e ridurre la quantità di energia necessaria per eseguire le applicazioni.

## Capitolo 2

# Descrizione della Base di dati iniziale e Analisi concettuale

### 2.1 Base di dati iniziale

Di seguito viene riportata la base di dati iniziale molto primitiva, con le entità principali ma con grande ridondanza dei dati.



**Figura 2.1.** Base di dati iniziale.

### 2.2 Raffinamento dei Requisiti per la ristrutturazione

Raccolta iniziale dei requisiti che descrivono le esigenze degli utenti e gli obiettivi del sistema tramite una lista ordinata.

1. Dell' Utente interessa:

- 1.1 L'E-mail
  - 1.2 La Password
  - 1.3 Il Nome
  - 1.4 Il Cognome
  - 1.5 Il Cellulare
  - 1.6 Il Codice Fiscale
  - 1.7 La Regione di residenza
  - 1.8 La Città di residenza
  - 1.9 La Via di residenza
  - 1.10 Il Cap di residenza
2. Della Transazione interessa:
- 2.1 Il cliente che effettua la transazione
  - 2.2 I prodotti che intende donare
  - 2.3 L'istante della donazione
  - 2.4 La data di ritiro della donazione
  - 2.5 Via di ritiro dei prodotti
3. Del Prodotto interessa:
- 3.1 La tipologia
  - 3.2 Il nome
  - 3.3 La Scadenza
  - 3.4 La Quantità (in Chilogrammi)
  - 3.5 I Tags per specificare alcune caratteristiche del prodotto
4. Della Tipologia interessa:
- 4.1 La categoria {Pasta, Verdura, Frutta, Altro}
5. Della Regione interessa:
- 5.1 Il Nome
6. Del Cap interessa:
- 6.1 Il codice
7. Dell'Identificativo interessa:
- 7.1 Il codice
  - 7.2 Il cliente
8. Della Città interessa:
- 8.1 Il nome

## 8.2 La regione

### LE FUNZIONALITÀ PRINCIPALI:

1. Dato un utente è possibile registrarsi
2. Dato un utente è possibile effettuare il login
3. Dato un utente della piattaforma deve essere possibile effettuare una transazione se la data di scadenza è maggiore della data odierna e la data del ritiro è minore della scadenza del prodotto e maggiore di oggi
4. Dato un utente è possibile avere lo storico delle transazioni effettuate
5. Dato un utente è possibile modificare i suoi dati personali
6. Dato un utente è possibile visualizzare il numero di donazioni totali
7. Dato un utente è possibile visualizzare la quantità (in Chilogrammi) delle donazioni totali
8. Dato un utente è possibile visualizzare la quantità (in Chilogrammi) delle donazioni per tipologia

## 2.3 Diagramma ER

Il Diagramma Entità-Relazione (ER) è uno strumento utilizzato nella progettazione di database per rappresentare le entità (oggetti o concetti) all'interno di un sistema e le relazioni tra di esse. Il diagramma ER visualizza graficamente le entità del database e le relazioni tra di esse.

Il diagramma ER è composto da entità, relazioni e attributi. Le entità rappresentano i concetti o gli oggetti all'interno del sistema, mentre le relazioni rappresentano le connessioni tra di esse. Gli attributi descrivono le proprietà delle entità.

Le entità vengono rappresentate come rettangoli, le relazioni come linee che collegano le entità, e gli attributi come ovali attaccati alle entità. Il tipo di relazione tra le entità può essere uno-a-uno, uno-a-molti o molti-a-molti.

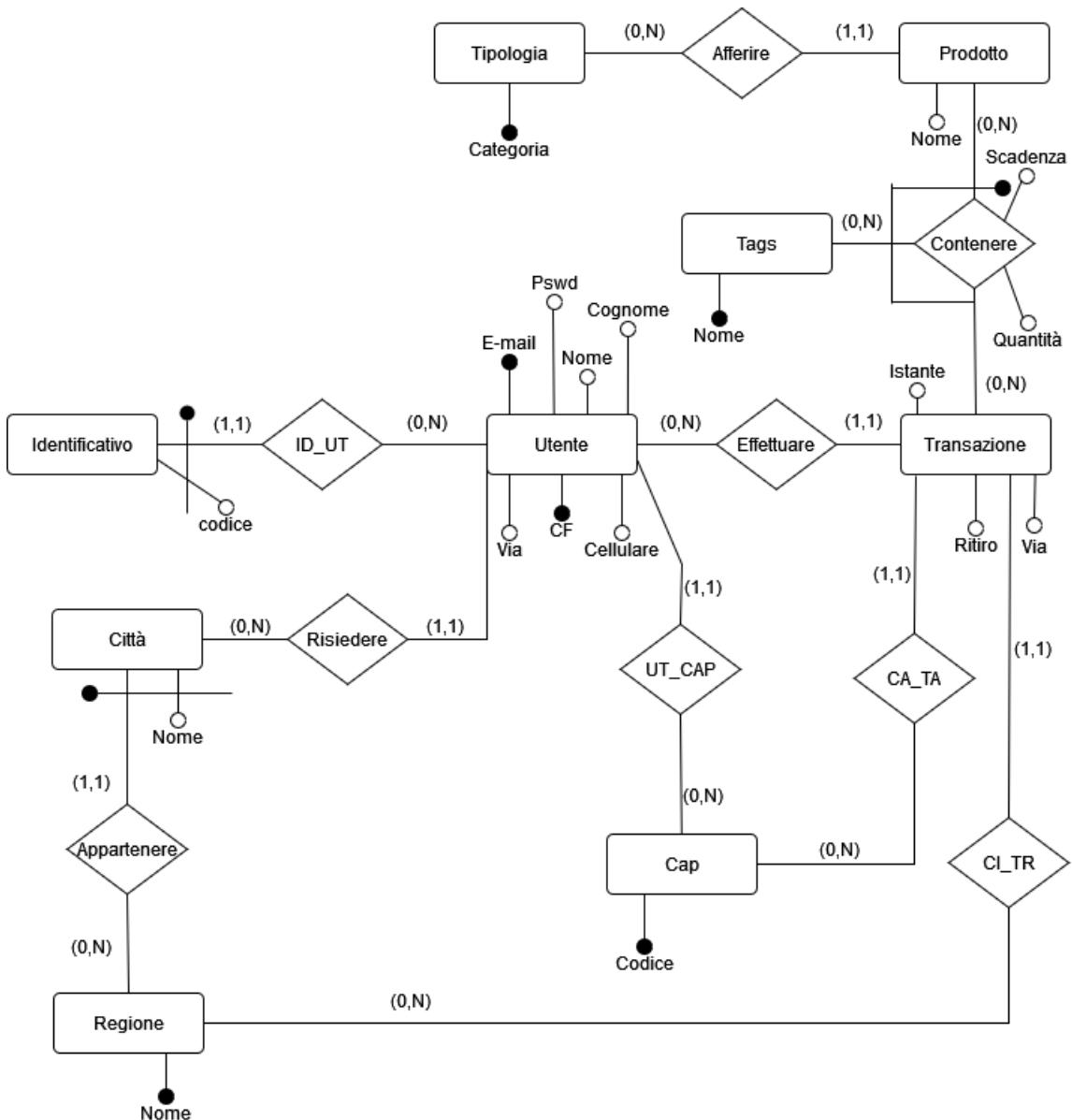


Figura 2.2. Diagramma ER

## 2.4 Glossario dei domini

Alcuni domini degli attributi sono stati personalizzati per avere un controllo maggiore sui dati inseriti nel sistema.

Istante

Il dominio è formato dalla data e orario

Dcap

Il dominio è una stringa composta da 5 cifre numeriche .

Indirizzo di posta

Il dominio è una stringa il formato stringa@stringa.stringa.

Password

Il dominio è una stringa avente almeno:

1. Lettera minuscola;
2. Lettera maiuscola;
3. Numero ;
4. Carattere speciale;
5. Compresa tra 8 e 16 caratteri;

Telefono

Il dominio è un intero avente 10 cifre numeriche.

CodiceFiscale

Il dominio è una stringa avente la sequenza:

- 6 lettere maiuscole/minuscole;
- 2 cifre numeriche;
- 1 lettere maiuscole/minuscole;
- 2 cifre numeriche;
- 1 lettere maiuscole/minuscole;
- 3 cifre numeriche;
- 1 lettere maiuscole/minuscole;

## 2.5 Definizione informale dei vincoli esterni

I vincoli esterni, anche noti come vincoli di integrità referenziale esterna, sono regole che vengono imposte su una tabella di database per garantire che i dati inseriti in una colonna di tale tabella siano coerenti con i dati presenti in un'altra tabella.

[Vincolo.transazione.momento]

Per ogni transazione l'istante della transazione deve essere minore della data del ritiro.

$$\forall t, d, i, r \quad Transazione(t) \wedge Instante(t, d) \wedge Data(d, i) \wedge ritiro(t, r) \rightarrow i < r$$

[Vincolo.ritiro.scadenza]

Per ogni transazione tutti i suoi prodotti devo avere la scadenza maggiore del ritiro.

$$\forall t, p, r, s$$

$$\text{Transazione}(t) \wedge \text{Prodotto}(p) \wedge \text{Contenere}(t, p) \wedge \text{Ritiro}(t, r) \wedge \text{Scadenza}(t, p, s) \rightarrow s > r$$

[Vincolo.transazione.instante]

Non possono esistere transazioni nello stesso istante effettuate dallo stesso utente

$$\forall t, t', u, i, i' \quad \text{Transazione}(t) \wedge \text{Transazione}(t') \wedge \text{Utente}(u) \wedge \text{Effettuare}(u, t) \wedge \text{Effettuare}(u, t') \wedge \text{Istante}(t, i) \wedge \text{Istante}(t', i') \wedge i = i' \rightarrow t = t'$$

## 2.6 Dizionario dei dati

Il dizionario dei dati, anche noto come data dictionary, è una raccolta di informazioni sulle entità, gli attributi, i vincoli e altre informazioni utili relativi ai dati presenti in una base di dati. Il dizionario dei dati viene utilizzato come strumento di riferimento per aiutare i programmati, gli amministratori del database e altri membri del team di sviluppo del software a comprendere e utilizzare correttamente i dati contenuti in una base di dati.

### Entità Utente

Ogni istanza di questa entità rappresenta un utente (Req. 1.)

| attributo | dominio            | molteplicità | descrizione                    |
|-----------|--------------------|--------------|--------------------------------|
| email     | Indirizzo di posta |              | L'email dell'utente            |
| pswd      | password           |              | La password scelta dall'utente |
| Nome      | stringa            |              | Il nome dell'utente            |
| cognome   | stringa            |              | Il cognome dell'utente         |
| cellulare | Telefono           |              | Il cellulare dell'utente       |
| Cf        | CodiceFiscale      |              | Il codice fiscale dell'utente  |
| via       | stringa            |              | La via dove risiede l'utente   |

### Entità Transazione

Ogni istanza di questa entità rappresenta una transazione (Req. 2.)

| attributo | dominio          | molteplicità | descrizione   |
|-----------|------------------|--------------|---|
| Codice    | GeneratoreIntero |              | Il codice per distinguere univocamente la transazione     |
| Istante   | Istante          |              | L'istante di tempo in cui viene effettuata la transazione |
| Via       | Stringa          |              | La via dove il pacco verrà ritirato                       |
| ritiro    | data             |              | La data di ritiro del pacco                               |

### Relazione Contenere

Ogni istanza di questa relazione lega una transazione ad un prodotto

| attributo | dominio    | molteplicità | descrizione              |
|-----------|------------|--------------|--------------------------|
| Scadenza  | Data       |              | La scadenza del prodotto |
| quantità  | Intero > 0 |              | La quantità del prodotto |

[Vincolo.di.integrità]: La data di scadenza deve essere maggiore della data odierna.

### Entità Prodotto

Ogni istanza di questa entità rappresenta un prodotto(Req.3)

| attributo | dominio          | molteplicità | descrizione  |
|-----------|------------------|--------------|--|
| Codice    | GeneratoreIntero |              | Il codice per distinguere univocamente un prodotto |
| Nome      | stringa          |              | Il nome del prodotto                               |

### Entità Tipologia

Ogni istanza di questa entità rappresenta una tipologia(Req.4)

| attributo | dominio                      | molteplicità | descrizione               |
|-----------|------------------------------|--------------|---------------------------|
| categoria | {pasta,verdura,frutta,altro} |              | La categoria del prodotto |

### Entità Regione

Ogni istanza di questa entità rappresenta una tipologia(Req.5)

| attributo | dominio | molteplicità | descrizione             |
|-----------|---------|--------------|-------------------------|
| nome      | stringa |              | La regione di residenza |

### Entità Cap

Ogni istanza di questa entità rappresenta una tipologia(Req.6)

| attributo | dominio | molteplicità | descrizione                               |
|-----------|---------|--------------|---|
| codice    | Dcap    |              | Il cap del luogo di residenza dell'utente |

### Entità Città

Ogni istanza di questa entità rappresenta una tipologia(Req.8)

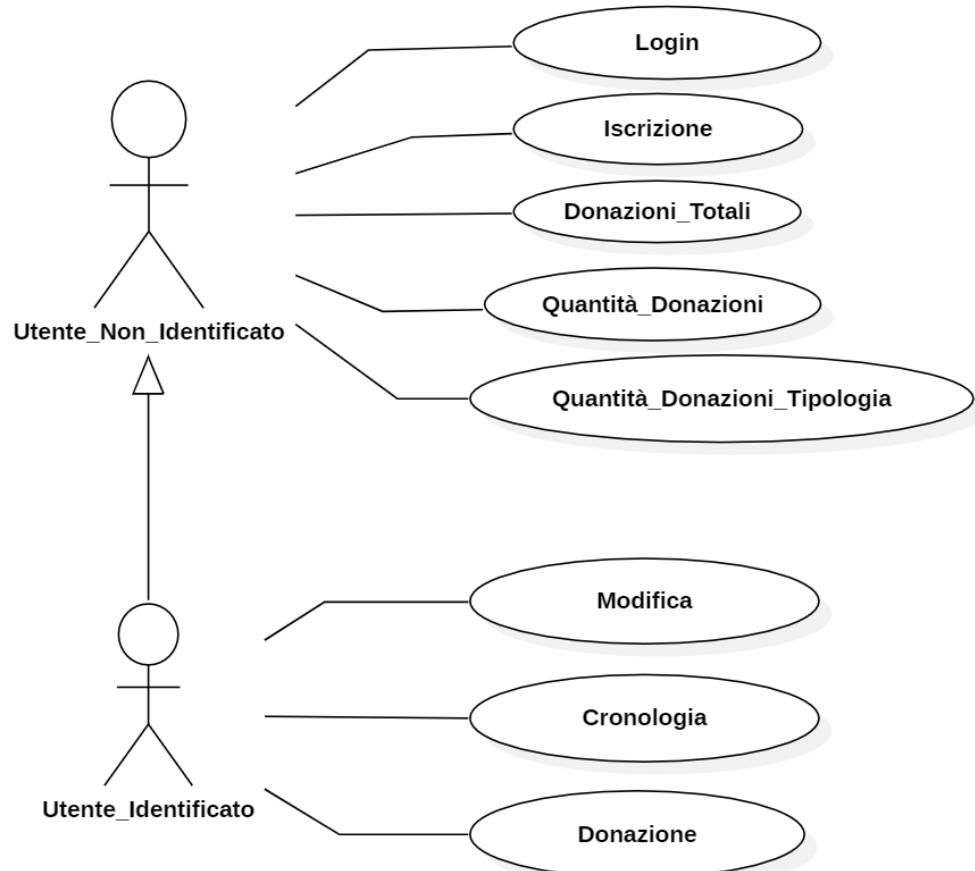
| attributo | dominio | molteplicità | descrizione           |
|-----------|---------|--------------|-----------------------|
| nome      | stringa |              | La città di residenza |

# Capitolo 3

## Analisi degli Use-Case

### 3.1 Diagramma UML degli Use-Case

UML (Unified Modeling Language) è un linguaggio di modellizzazione standard utilizzato per la progettazione e la documentazione di sistemi software e di altri tipi di sistemi. Il diagramma UML è una rappresentazione grafica dei componenti di un sistema e delle relazioni tra di essi.



## 3.2 Specifiche concettuali informali degli Use-Case

Le specifiche concettuali informali degli Use-Case sono un modo per descrivere in modo chiaro e non tecnico i requisiti e le funzionalità di un sistema software. Queste specifiche descrivono i casi d'uso (Use-Case) del sistema, ovvero le attività o le funzionalità che il sistema deve eseguire per soddisfare le esigenze dell'utente.

- SPECIFICHE USE-CASE: Utente\_Non\_Registrato

**ISCRIZIONE**( No:nome, Co:cognome, Em:email, Ps:password, Cell:cellulare, Cf:CodiceFiscale, Via:indirizzo, Ci:città, Re:regione, C:cap ) : Utente (Req. 1 delle Funzionalità Principali )

- **Precondizioni:** La seguente funzionalità non richiede precondizioni.
- **Postcondizioni:** Variazione del livello estensionale dei dati: Alla fine dell'esecuzione della funzione, il livello estensionale dei dati cambia rispetto a quello iniziale, come descritto di seguito:
  - **Variazioni nel dominio di interpretazione:** un nuovo elemento  $\alpha$ .
  - **Variazioni nelle ennuple di predicati:** Utente( $\alpha$ ), nome( $\alpha$ , No), cognome( $\alpha$ , Co), email( $\alpha$ , Em), password( $\alpha$ , ps), cellulare( $\alpha$ , cell), CodiceFiscale( $\alpha$ , Cf), via( $\alpha$ , Via), risiedere( $\alpha$ , Ci), appartenere(Ci, Re), ut\_cap( $\alpha$ , C)

**LOGIN**(Em:email,Ps:password): Utente (Req. 2 delle Funzionalità Principali)

- **Precondizioni:**  $\exists u \text{ Utente}(u) \wedge \text{email}(u, Em) \wedge \text{password}(u, ps)$
- **Postcondizioni:** Variazione del livello estensionale dei dati: Nessuna
- **Valore di ritorno:** Sia  $u$  tale da soddisfare la formula seguente:  $\text{Utente}(u) \wedge \text{email}(u, Em) \wedge \text{password}(u, ps)$

**Donazioni\_Totali()** : Intero  $\geq 0$  (Req. 6 delle Funzionalità Principali )

- **Precondizioni:** La seguente funzionalità non richiede precondizioni.
- **Postcondizioni:** Nessuna
- **Valore di ritorno:**  $|D|$  tale che:  

$$D = \{ t \mid \text{Transazione}(t) \}$$

**Quantità\_Donazioni()** : Reale $\geq 0$  (Req. 7 delle Funzionalità Principali )

- **Precondizioni:** La seguente funzionalità non richiede precondizioni.

- **Postcondizioni:** Nessuna

- **Valore di ritorno:**

$$Q = \{ (t,p,q) \mid \text{Transazione}(t) \wedge \text{Prodotto}(p) \wedge \text{Contenere}(t,p) \wedge \text{quantità}(t,p,q) \}$$

$$\text{Risultato} = \sum_{(t,p,q) \in Q} q$$

**Quantità\_Donazioni\_Tipologia(Tipo: Stringa)** : Reale $\geq 0$  (Req. 8 delle Funzionalità Principali )

- **Precondizioni:** La seguente funzionalità non richiede precondizioni.

- **Postcondizioni:** Nessuna

- **Valore di ritorno:**

$$Q = \{ (t,p,q) \mid \exists \text{ Cat } \text{Transazione}(t) \wedge \text{Prodotto}(p) \wedge \text{Contenere}(t,p) \wedge \text{quantità}(t,p,q) \wedge \text{Afferisce}(p,\text{Tipo}) \wedge \text{Categoria}(\text{Tipo},\text{Cat}) \wedge \text{Cat}=\text{Tipo} \}$$

$$\text{Risultato} = \sum_{(t,p,q) \in Q} q$$

- SPECIFICHE USE-CASE: Utente\_Non\_Registrato

**Cronologia(em:E-mail)** : {T:transazione,P:prodotto} (0,N) (Req. 4 delle Funzionalità Principali )

- **Precondizioni:** La seguente funzionalità non richiede precondizioni.

- **Postcondizioni:** Nessuna

- **Valore di ritorno:**

$$Q = \{ (t,p) \mid \exists u,e \text{ Transazione}(t) \wedge \text{Prodotto}(p) \wedge \text{Contenere}(t,p) \wedge \text{Utente}(u) \wedge \text{effettuare}(u,t) \wedge \text{Contenere}(t,p) \wedge \text{E-mail}(u,e) \wedge e=em \}$$

**Donazione**(em:email, L:p:prodotto,s:data,q:reale,ta: tags(0,N)(1,N), v:stringa, r:data ):Transazione (Req. 3 delle Funzionalità Principali )

- **Precondizioni:** per ogni s app L r > adesso and s > adesso and s > r .
- **Postcondizioni:** Variazione del livello estensionale dei dati: Alla fine dell'esecuzione della funzione, il livello estensionale dei dati cambia rispetto a quello iniziale, come descritto di seguito:
  - **Variazioni nel dominio di interpretazione:** inserimento dell' elemento  $\alpha$ .
  - **Variazioni nelle ennuple di predicati:** Transazione( $\alpha$ ), istante( $\alpha$ , adesso), via( $\alpha$ , v), ritiro( $\alpha$ ,d), effettuare (U,  $\alpha$ ), contenere( $\alpha$ ,p)  $\forall p \in L$ , scadenza( $\alpha$ ,p,s) $\forall s \in L$ , quantità( $\alpha$ ,s,q)  $\forall q \in L$ , tags( $\alpha$ ,ta)  $\forall ta \in L$ .
- Valore di ritorno:  $\alpha$

## Capitolo 4

# Schema Relazionale della Base Dati

Questo passaggio viene eseguito per:

- Migliorare la comprensione: la ristrutturazione può aiutare a semplificare la struttura del diagramma e renderlo più facile da comprendere per gli utenti.
- Ottimizzare le prestazioni: la ristrutturazione può aiutare a migliorare le prestazioni del database, rendendo più efficiente l'accesso ai dati e la gestione delle informazioni.
- Rimuovere le ridondanze: la ristrutturazione può aiutare a identificare e rimuovere le informazioni ridondanti, che possono rallentare il funzionamento del database.
- Supportare la scalabilità: la ristrutturazione può aiutare a preparare il database per una maggiore quantità di dati e una maggiore complessità, rendendolo più scalabile e pronto per futuri sviluppi.

### 4.1 Diagramma ER ristrutturato

Di seguito viene riportato il Diagramma ER ristrutturato al fine di rispettare le seguenti proprietà:

- Analisi delle ridondanze (ed eventuale loro eliminazione)
- Eliminazione delle generalizzazioni
- Partizionamento/accorpamento delle entità
- Eliminazione degli attributi multivaleure e degli attributi composti
- Scelta degli identificatori primari

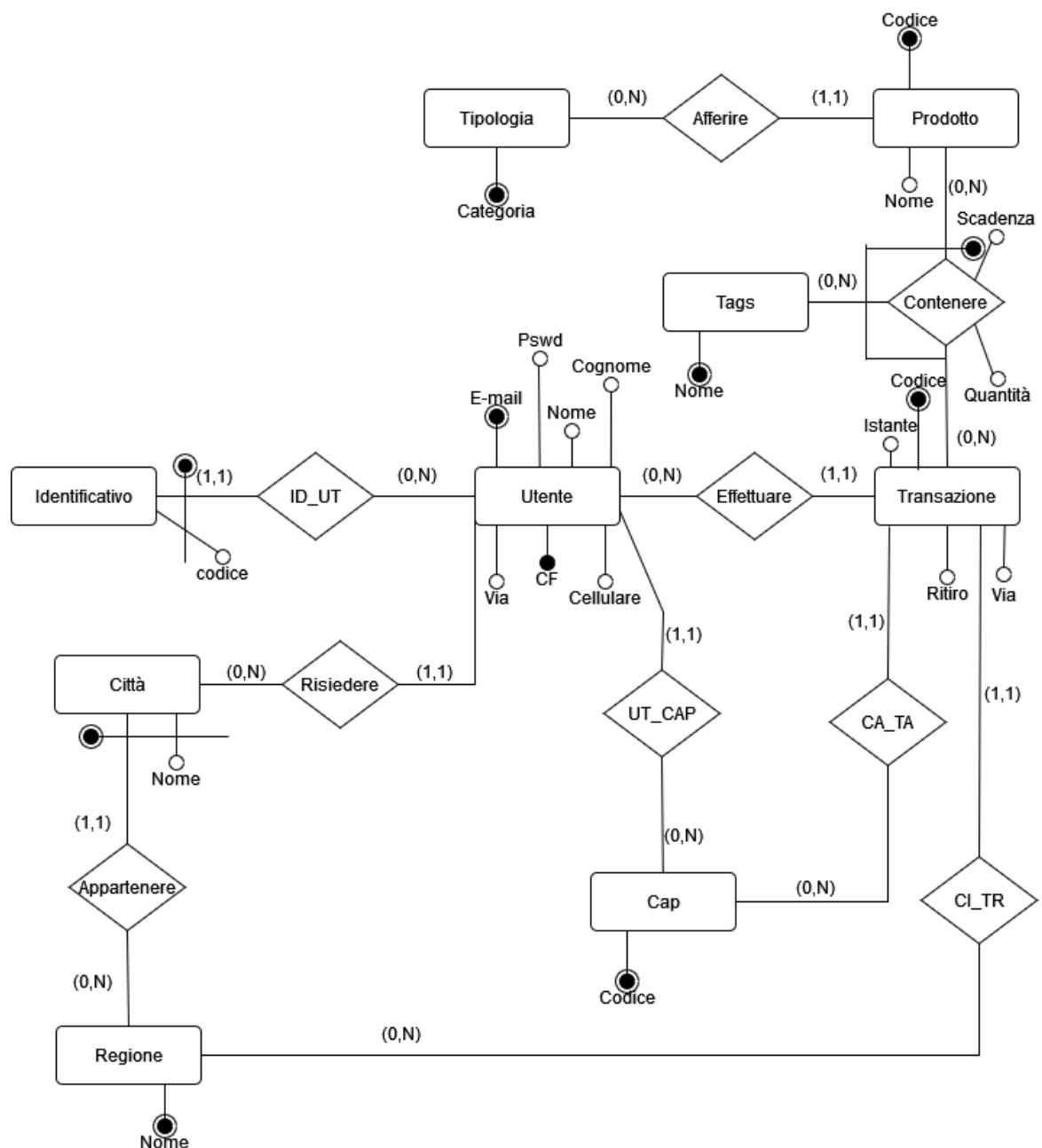


Figura 4.1. Diagramma ER ristrutturato

## **4.2 Definizione delle Relazioni Derivanti da Entità e Relationship Accorpate ad Entità**

- **Identificativo** (codice:stringM, codUtente:stringM)  
foreign key: (codUtente) references utente(email)

La relazione accorda la relationship ID\_UT.

- **Utente** (email:stringM, pswd:password, nome:stringS, cognomen:stringS, via:string, cf:CodiceFiscale, cell:cellulare, codCittà:stringS, codRegione:stringS codCap:Cap)  
chiave: cf  
foreign key: (codCittà,codRegione) references città(nome,codRegione)  
foreign key: codCap references cap(codice)

La relazione accorda le relationship: Risiedere, UT\_CAP.

- **Città** (nome:stringS, codRegione:stringS)  
foreign key codRegione references regione(nome)

La relazione accorda la relationship: Appartenere

- **Regione**(nome:stringS)
- **Cap**(codice:cap)
- **Transazione**(codice:integer, istante: timestamp with timezone, via: stringM, ritiro:date, codUtente:stringM, codCap:Cap, codCittà:stringS, codRegione:stringS)  
serial: i valori del campo codice sono generati automaticamente dal DBMS  
foreign key: codUtente references utente(nome)  
foreign key: codCap references Cap(codice)  
foreign key: (codCittà,codRegione) references città(nome,codRegione)

La relazione accorda le relationship: CI\_TR,CA e Effettuare

- **Contenere**(Scadenza:date, quantità:real, codTransazione:integer, codProdotto:stringM, codTags:stringM)  
foreign key: codTransazione references Transazione(codice)  
foreign key: codTags references Tags(nome)  
foreign key: codProdotto references Prodotto(codice)
- **Tags**(Nome:stringM)
- **Prodotto**(Codice:integer, nome:stringM, codTipologia:stringS)  
serial: i valori del campo codice sono generati automaticamente dal DBMS  
foreign key: codTipologia references Tipologia(categoria)

La relazione accoppi le relationship: afferire

- **Tipologia**(categoria:stringS)

### 4.3 Progettazione dei vincoli esterni

#### [Vincolo.transazione.momento] Req.1

Il vincolo impone che per ogni transazione, il campo istante deve essere minore della data del ritiro

**La funzione sarà invocata al momento dell'inserimento di una transazione punto di richiesta :** dopo l'operazione riscontrata

- DB.Vincolo.transazione.momento(i:Timestamp , r:date)  

$$Q \leftarrow \text{If } (i < r) \text{ then}$$

$$\text{return True;}$$

$$\text{else:}$$

$$\text{return False;}$$

if  $Q == \text{False}$  then  
 blocca l'operazione intercettata

#### [Vincolo.ritiro.scadenza] Req.2

Il vincolo impone che la data del ritiro del pacco sia necessariamente minore alla scadenza dei prodotti in quanto altrimenti non si avrebbe tempo per utilizzare i prodotti inviati dall'utente

Isoliamo le operazioni che potrebbero causare una violazione del vincolo logico specifico nella base dati legale ovvero:

L'Inserimento di una transazione da parte di un utente

- DB.Vincolo.ritiro.scadenza ( P:set(p:stringM), r:date ):Transazione

Creazione di una tabella temporanea in cui inserisco tutti i codice dei prodotti che poi verrà eliminata

Create Table TMP as (codice stringsM primary key)

Vado a inserire tutti i valori dei prodotti

**La funzione sarà invocata al momento dell'inserimento di una transazione punto di richiesta :** dopo l'operazione riscontrata

$Q \leftarrow \text{Select * From TMP Where TMP.scadenza} < :r$

If  $|Q| > 0$  then  
 blocca l'operazione altrimenti permettila

#### [Vincolo.transazione.instante] Req.3

Il vincolo impone che non possano esistere transazioni nello stesso istante effettuate dallo stesso utente

- DB.Vincolo.transazione.instante(u:stringM, i: timestamp)

**La funzione sarà invocata al momento dell'inserimento di una transazione**

**punto di richiesta :** dopo l'operazione riscontrata

$Q \leftarrow \text{Select } * \text{ From transazione t Where t.codUtente} = :u \text{ and t.istante} == :i$

If  $|Q| > 0$  then  
 blocca l'operazione altrimenti permettila

## 4.4 Realizzazione degli use-case

- Utente\_non\_identificato

– **ISCRIZIONE( No:stringS, Co: stringS, Em:email, Ps:password, Cell:cellulare, Cf:CodiceFiscale, Via:stringM, Ci:stringS, Re:stringS, C:cap ) : StringM**

Insert into Utente(nome, cognome, email, pswd, cell, cf, via, codCittà, CodRegione,codCapp) values (:No, :Co, :Em, :Ps, :Cell, :Cf, :Via, :Ci, :Re, :C)

– **Donazioni\_Totali(): integerZ**

Select count (T.codice) as Risultato  
 From Transazione T

– **Quantità\_Donazioni(): reale>= 0**

Select sum(C.quantità) as Risultato  
 From Transazione T, Contenere C, Prodotto P

Where C.codTransazione = T.codice and C.codProdotto=O.codice

- **Donazioni\_Donazioni\_tipologia(Tip:StringS): reale>= 0**

```
Select sum(C.quantità) as Risultato
From Transazione T, Contenere C, Prodotto P , Tipologia Tipo
Where C.codTransazione = T.codice and C.codProdotto=O.codice and
Tipo.categoria = :Tip
```

- **Utente\_identificato**

- **Cronologia(em:email):set<integerZ,Timestamp,date,stringM>**

```
select T.codice, T.istante, T.ritiro, T.via
from Transazione
where codUtente = :em
```

- **Donazione(em:email, setp:integerZ,s:date,q:Real,ta, set(nome:stringS), v:stringS, r:date ):IntegerZ**

```
Insert into Transazione (istante,via,ritiro)
Values (Current_timestamp(), :v , :r)
Verrà generato un codice da tale query → Codice
```

Per ogni prodotto e tags viene eseguita la seguente query:

```
Insert into contenere (codTransazione,CodTags,CodProdotto,scadenza,quantità)
Values (:codice, p, :s, :q, :nome)
```

## Capitolo 5

# Descrizione del sito con screenshot

### 5.1 Dona

La parte dinamica del sito è gestita in questa sezione, attraverso una combinazione della struttura definita all'interno del file “index.php” della cartella “Dona” e una serie di file JS (raggruppati dentro l'opportuna sottocartella “Dona\_Script”) che ne definiscono il comportamento:

- Index.JS: questo è lo script principale richiamato dalla pagina “index.php”. Lo scopo primario di questo script, è quello di effettuare un setting iniziale di una serie di variabili utili per la processazione dei dati che verranno inseriti dall'utente all'interno del form del dona (variabili che conterranno i tag, l'indirizzo, l'oggetto table che verrà usato per il riepilogo, variabili di controllo etc...) e soprattutto di richiamare a sé tutti gli altri JS contenuti nella opportuna cartella.  
Tale richiamo viene eseguito grazie ad una chiamata AJAX ad un opportuno “Loader.PHP” che passerà all’index.JS la lista dei file che sono contenuti nella directory, dopodiché mediante un ciclo FOR sarà sufficiente richiamare per ogni elemento la funzione definita da JQuery “getScript”, per prendere ed eseguire tutti i file JS scansionati e passati dallo “scanddir()” del Loader.PHP (ovviamente dovremo stare attenti a non rieseguire lo stesso “index.JS”).
- DinamicTable.JS: questo è lo script che genera dinamicamente la tabella di riepilogo.  
tramite l'oggetto HTML “table” contenuto in una opportuna variabile (definita in “index.JS”), sarà modificato ad ogni inserimento di un prodotto, al click del bottone di “inserisci”, vi sono delle condizioni che devono però risultare rispettate:
  - Data di Scadenza strettamente maggiore della currentDate().
  - Almeno una Tipologia di prodotto selezionata.
  - Nome del prodotto settato come una stringa testuale.

Se tutti i controlli vengono rispettati allora viene inserita una nuova entry generata dinamicamente come una riga con delle colonne che verrà appesa alla

table, al contempo un array mi terrà traccia dei dati inseriti dall'utente per ogni specifico prodotto.

Al contrario al click del bottone di “Rimuovi l'ultimo inserimento” verrà generata un nuovo oggetto table che scorrerà l'array descritto precedentemente, meno l'ultimo index (ogni index dell'array corrisponde ad una riga della tabella) e per ognuno di essi mi rigenera le stesse righe e le stesse colonne.

Ora che si ha questo nuovo oggetto table (meno l'ultima entry), lo sostituisco al principale e lo ricarico nella opportuna Div della pagina di ordinazione.

- GestioneGenerale.JS: questo script si occupa di una serie di operazioni di gestione generale della pagina di Donazione.

- Controllare la size del Viewport per fare degli opportuni Hide e Show.
  - Gestione randomica del contenuto testuale delle carte, implementate mediante bootstrap inerenti al “Lo sapevi che...”, alla selezione di un bottone corrispondente alla tipologia di prodotto inserito.
  - Di gestire l'invio all'atto di Conferma della Donazione, cioè quando sono nella sezione di “Consegna tramite Locker” o “Ritiro presso Domicilio”, tramite una chiamata al file Elaborazione.PHP che si prenderà i dati tramite l'opportuna GET e li processerà per inserire il tutto nell'apposito DB.

- Map.JS: questo script si occupa della gestione dei 3 iframe implementati nella opportuna DivMap della pagina “index.php”. In particolare lo script con una funzione preliminare, gestisce la scelta della città dell'utente tramite un menù a tendina associato a degli Switch Case, e ognuno di essi (per città) richiamerà una relativa funzione di generaOrario(), quest'ultima è una semplice funzione, che ritorna una stringa al cui interno vi è presente un numero randomico stabilito come apertura e chiusura (l'apertura è un numero compreso tra 8-11; mentre la chiusura tra 19-21).

Tale funzione viene richiamata solo una volta, questo per evitare che ogni volta che torni sulla medesima città mediante il select-option che mi attiva lo Switch-Case alla scelta, mi rigeneri magari per un Locker un NUOVO orario da quello che avevo visto precedentemente.

Successivamente per ogni città selezionata, vengono generati 3 link (Google Maps) corrispettivi ai suoi 3 Locker (indirizzi scelti dall'amministratore dell'applicazione Web) che verranno settati opportunamente come attributo “src” dei 3 Iframe della pagina del sito.

- Tag.JS: infine questo script si occupa sostanzialmente della gestione dei vari TAG che l'utente opzionalmente inserisce per ogni prodotto.

In particolare ad ogni pressione del tasto invio, o all'evento blur da parte dell'utente sull'area d'inserimento, un opportuno array di “tag” sarà aggiornato e usato per generare poi una opportuna lista di visualizzazione dei tag scritti (ovviamente tale array sarà modificato anche ad ogni eliminazione e usato nuovamente per rigenerare la lista dei tag aggiornata in corrispondenza delle eliminazioni effettuate).

“Elaborazione.PHP”:

Quest'ultimo si occupa di prendere i dati relativi a tutti i prodotti inseriti dall'utente, e di inserirli nell'apposita table “Transazioni” del nostro DB.

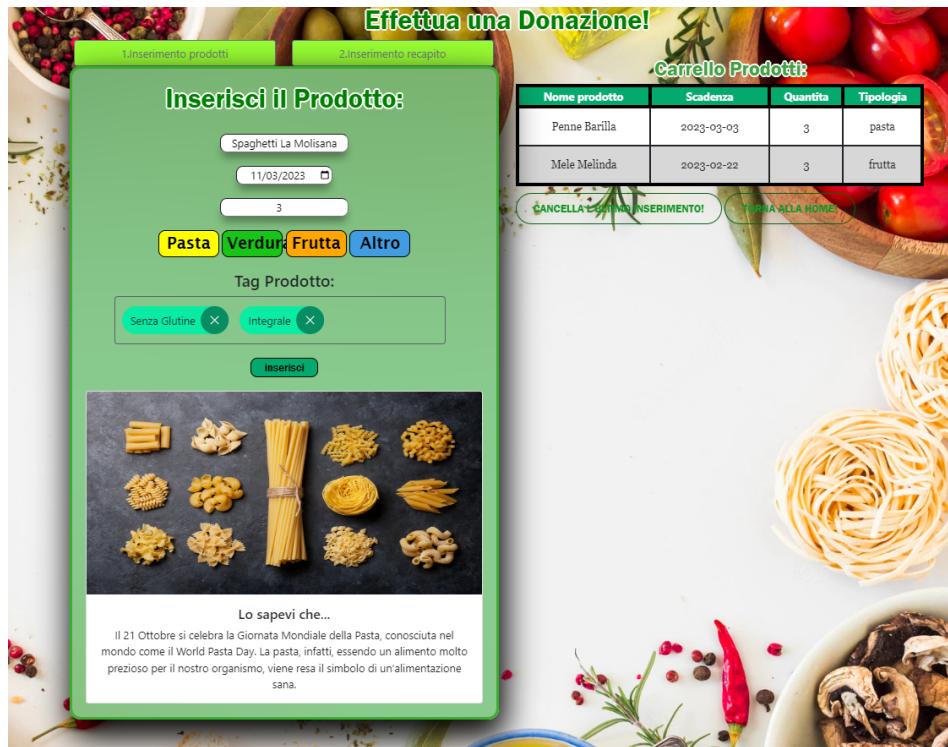


Figura 5.1. Sezione inserimento prodotto.

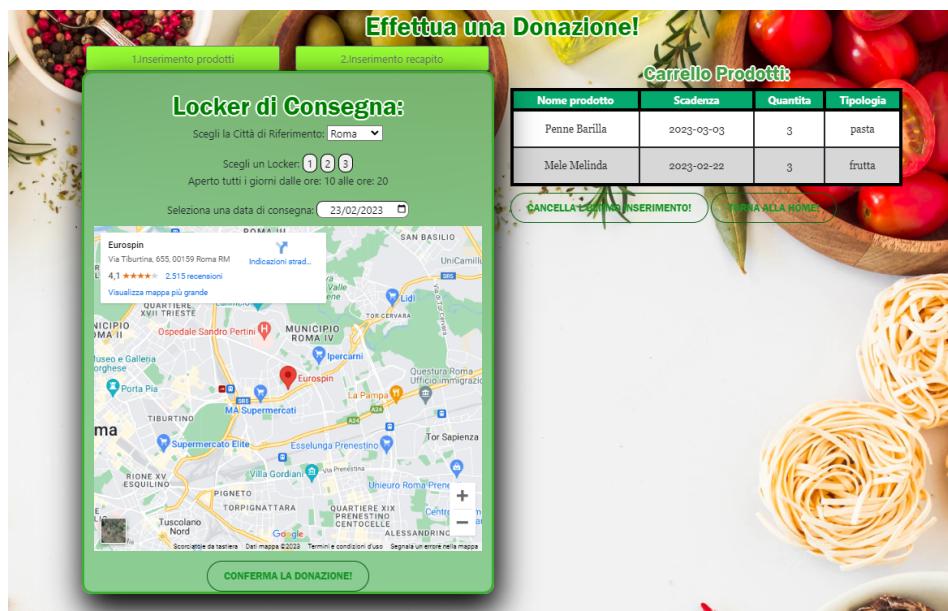


Figura 5.2. Sezione indirizzo di consegna.

## 5.2 Cronologia

La table all'interno della pagina cronologia viene generata tramite una richiesta in PHP in corrispettiva dell'utente loggato, successivamente ogni dato passatoci dal PHP verrà inserito all'interno della table in modo dinamico grazie al file index.js associato. Avremmo poi delle query di riepilogo degli ordini effettuati. È presente una leggenda che spiega l'eventuale colore della data scelta (per il recapito o la consegna presso LOCKER) anch'essa generata in modo dinamico. La responsive è stata gestita tramite delle medie queries dove dopo una determinata soglia la tabella viene modificata in modo da avere una presentazione in modo verticale supponendo che si sta visionando il sito web in una modalità telefono.

| CRONOLOGIA ORDINI     |                     |            |                  |                 |  |            |              |
|-----------------------|---------------------|------------|------------------|-----------------|--|------------|--------------|
| Nome Prodotto:        | Tipologia Prodotto: | Scadenza:  | Quantità Donata: | Data Donazione: | Indirizzo Scelto:  |            | Data Scelta: |
| Zucchine              | verdura             | 2023-03-05 | 2                | 2023-02-27      | Consenza presso Locker: Eurospin Via Tiburtina 655       | 2023-03-01 |              |
| Mele                  | frutta              | 2023-03-10 | 3                | 2023-02-27      | Consenza presso Locker: CONAD Piazza dei Redi Roma<br>15 | 2023-02-27 |              |
| Banane                | frutta              | 2023-03-11 | 2                | 2023-02-27      | Consenza presso Locker: CONAD Piazza dei Redi Roma<br>15 | 2023-02-27 |              |
| Broccoli              | verdura             | 2023-04-06 | 3                | 2023-02-27      | Consenza presso Locker: CONAD Piazza dei Redi Roma<br>15 | 2023-02-27 |              |
| Barilla<br>maccheroni | pasta               | 2023-03-04 | 3                | 2023-02-21      | Consenza presso Locker: CONAD Piazza dei Redi Roma<br>15 | 2023-02-23 |              |

Quantità di frutta donata ,complessivamente, pari a: 5 KG

Quantità di pasta donata ,complessivamente, pari a: 3 KG

Quantità di verdura donata ,complessivamente, pari a: 5 KG

Legenda per le date di consegna:

- Il tuo ordine sarà ritirato nei giorni successivi
- Il tuo ordine sarà ritirato entro oggi
- Il tuo ordine è stato ritirato nei giorni precedenti

[HOME](#)

**Figura 5.3.** Cronologia dei prodotti Alimentari donati dall'utente.

## 5.3 Home Section

- Header

La struttura dell'Header è stata realizzata tramite la libreria Bootstrap, mentre il suo contenuto varia in corrispondenza di una funzione di controllo PHP. In particolare:

Se la Sessione non è stata aperta da un utente registrato, o meglio se quest'ultimo non vi ha effettuato ancora un Login, l'Header si presenterà con un semplice button di "ACCEDI" in alto a destra della nostra Home Page; alternativamente vi sarà presente un menù dropdown (sempre implementato mediante Bootstrap), con tutte le opzioni disponibili per l'utente:

- Una modifica dei dati inseriti in fase di registrazione.
- Una propria cronologia delle donazioni effettuate.
- E l'opzione di Disconnetti che come da nome chiude la sessione dell'utente fino a nuovo login.

L'Header viene modificato mediante un codice PHP, il quale ne muterà il comportamento a seconda del contenuto nella SESSION (gestita dinamicamente con l'apertura di una sessione dotata di un proprio identificativo salvato in un'apposita table del nostro DB, e l'ennupla corrispettiva verrà eliminata ad ogni Disconnessione).

Ho voluto gestire i dati personali dell'utente all'interno della SESSION, così da migliorare l'efficienza da parte delle richieste ricevute dal DB, dato che così facendo la richiesta dei dati verrà effettuata una sola volta per utente, e i dati persistono fin tanto che la sessione non verrà chiusa e la relativa session\_id eliminata dalla ennupla corrispettiva della table.

- Video e Slogan

Immediatamente dopo l'header vi è presente una sezione per la gestione di un Video, mandato in Loop, subito dopo vi è lo slogan anche qui messo in un'apposita Div che cambierà comportamento a seconda di un avvenuta donazione oppure no.



*"Quello che facciamo è soltanto una goccia nell'oceano. Ma se non ci fosse quella goccia all'oceano mancherebbe."*

**Figura 5.4.** Header del sito.

- **Carousel**

Successivamente sempre grazie a Bootstrap, ho implementato un Carousel dinamico, dove quindi l'utente, tramite delle frecce gestite a livello JS (mediante il file "index.js" presente nella cartella della Home\_Section) è libero di muovere delle schermate con delle piccole informazioni preliminari sul nostro sito.

- **I nostri Donatori**

Qui vi è presente una semplice rappresentazione di alcune delle informazioni estrapolate dal nostro DB in merito alle transazioni effettuate sulla nostra piattaforma.

Il numero di Utenti attivi è gestito tramite una funzione inserita all'atto di caricamento del documento; la quale ogni 4 secondi setta randomicamente tale numero per dare più interattività (questo ovviamente a scopo illustrativo per evitare di dover segnare solo una sola sessione aperta sul sito in fase di

esecuzione).



## I nostri Donatori:

| QUANTITA' DI DONAZIONI (Kg): | NUMERO DI DONAZIONI EFFETTUATE: | NUMERO DI UTENTI ATTIVI: |
|------------------------------|---------------------------------|--------------------------|
| 18.1                         | 7                               | 9725                     |

**Figura 5.5.** Sezione statistiche.

- Sezioni

Successivamente, sono state implementate le sezioni del sito con le loro rispettive informazioni che verranno aggiornate eventualmente dal gestore del sito (“Il nostro programma, le nostre collaborazioni, News”), le sezioni sono state inserite con una disposizione su griglia descritta nell’apposito file “style.css”.

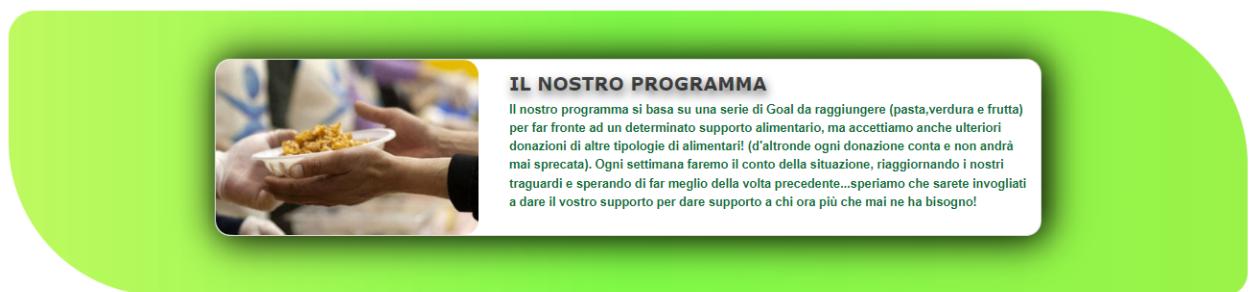
Le informazioni presenti nelle sezioni vengono caricate a livello JS dal file “index.js”, da dei file txt messi appositamente dentro la cartella “descrizioni\_card”, file che potranno essere modificati in un qualsiasi momento avendo un corrispettivo aggiornamento anche sulle apposite sezioni in modo dinamico. Le Sezioni vengono rappresentate in 2 modalità diverse e speculari tra loro, a seconda delle dimensioni del Viewport e gestite mediante mediaquery, infatti avremo una visualizzazione:

- In modo “Normale”: Dove al loro interno vi sarà presente una prima Div/Div1, che formerà lo sfondo della sezione di griglia su cui stiamo inserendo la sezione, una seconda Div/Div2 che conterrà la corrispettiva immagine in un certo modo e ricoprendo un certo numero di righe e colonne della zona disegnata, e successivamente altre Div che seguendo il medesimo ragionamento della Griglia conterranno il titolo della sezione e il testo che verrà caricato mediante Script.
- In modo “Card-implemented”: Dove il ragionamento rimane abbastanza il medesimo con la differenza che ora avremo le Sezioni rappresentate mediante carte di Bootstrap dinamiche.

Laddove venissero visualizzate le sezioni in modalità “Normale” le Card presenti nella modalità “Card-implemented” saranno nascoste (hide) o viceversa.



**Figura 5.6.** Visualizzazione di una Sezione



**Figura 5.7.** Sezione in formato Card

- I nostri Obiettivi e il Footer:

Implementati mediante CSS puro (classe=semi-donut), questi grafici si predispongono a variare il proprio fill in corrispettiva di un codice PHP pronto a calcolare la percentuale, a seconda degli obiettivi inseriti dall'amministratore del sito.

Successivamente vi è presente un oggetto paragraph, contenente i dati segnati nell'anello in una forma di legenda (in particolare stamperà la quantità donata per ciascuna tipologia e l'obiettivo prefissato dall'amministratore).

Gli anelli vengono “nascosti” mediante media query dello style.css, laddove il viewport si andasse a restringere eccessivamente lasciando spazio solo alla legenda descritta precedentemente; questo perché gli anelli dopo una certa soglia cominciano ad essere tagliati e diminuire la loro dimensione per farli entrare nella schermata sarebbe risultato illeggibile.

Infine vi è presente la Div inerente alle informazioni social, con annesso anche una piccola list-inline con vari HREF alle sezioni del sito.



**Figura 5.8.** Sezione I nostri obiettivi tramite semi-donut.

## 5.4 Login, Registrazione, Modifica e Disconnessione

Ogni input type viene associato ad una label mediante css, dove al focus di un determinato inputType la Label associata effettuerà una transazione verso l'alto per lasciar spazio all'inserimento del dato da parte dell'utente, se il dato fornитогli non rispetterà la regex assegnatavi allora la label ritornerà al posto di partenza, se invece il dato rispetterà la regex associata il contenuto e il colore del testo all'interno della label cambierà; questa implementazione si trova sia in login che registrazione.

- **Login**

Nel login si effettua un controllo dell' email se già esistente nel database, successivamente se l'email esisterà andremo a verificare che anche la password inserita dall'utente sia quella corrispondente all'interno del database, altrimenti ritornerà un messaggio in output e chiederà all'utente di modificare l'email o la password inserita.

Il ricordami è stato gestito tramite la SESSION ovvero la nostra Session-id viene inserita all'interno di un database se solo se il checkbox del ricordami viene checkato così tutte le volte che l'utente andrà a collegarsi al nostro sito web si effettuerà un controllo della sua session-id, se è già presente all'interno del nostro database verranno ricaricati tutti i dati dell'utente all'interno della nostra session.

- **Registrazione**

Oltre ad avere ulteriori regex, il form di "registrazione/index.php" avrà all'interno un datalist contenente tutte le regioni italiane così da agevolare l'inserimento da parte dell'utente.

Una volta che i dati avranno rispettato tutte le regex per ogni inputType associato, verrà inviato il form e controllato eventualmente se il codice fiscale sia univoco, stessa cosa per l'email.

Dopodiché se l'email e la relativa password associata sono già esistenti all'interno del nostro database, allora l'utente sarà riportato nel medesimo form con i dati inseriti precedentemente ancora settati, dovendo però cambiare ovviamente mail.

- **Modifica**

Il form viene ricompilato tramite i dati estratti dalla session e inseriti negli appositi input Type al fine di ridurre il numero di query richieste al DB.

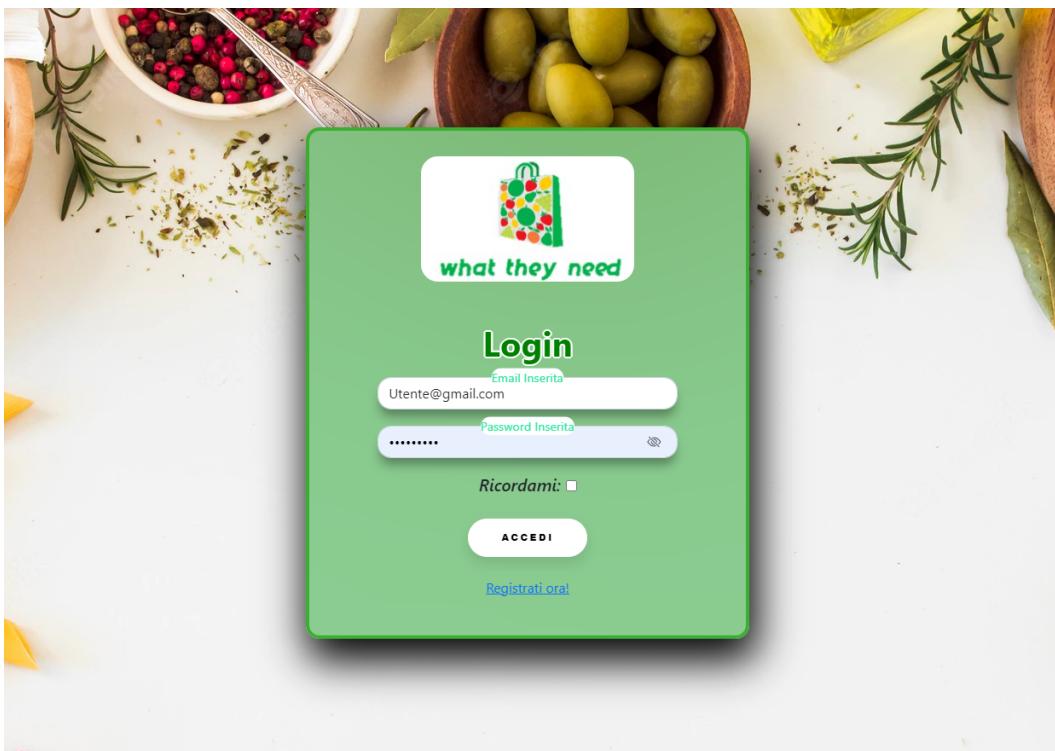


Figura 5.9. Login.

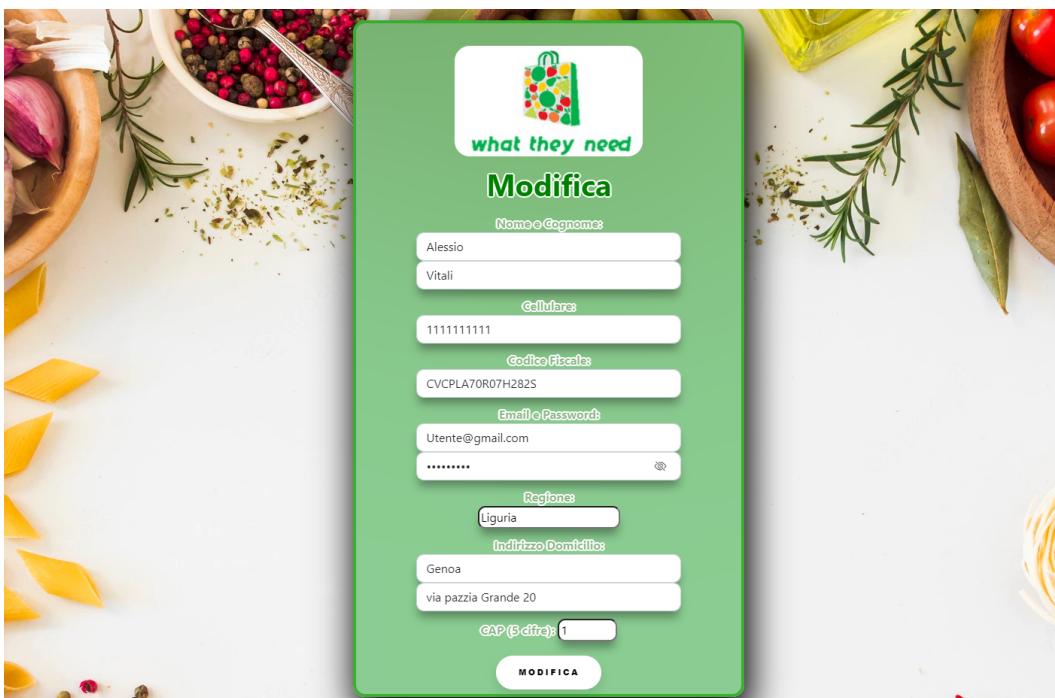


Figura 5.10. Modifica credenziali dell'utente.

- Disconnessione

La disconnessione del profilo viene effettuata tramite una chiamata alla pagina logout.PHP che essenzialmente va a mettere all'interno della nostra session: [“loggato”]=0, la session corrispettiva all'utente verrà svuotata e distrutta, rendendo di nuovo disponibile quel session id, cancellando anche l'assegnazione che avevamo dato all'utente all'interno del nostro database.

## Capitolo 6

# Problematiche affrontate per implementare il sito su Microsoft Azure

### 6.1 Studio del portale

La prima problematica affrontata in Microsoft Azure è stata quella di non aver mai utilizzato Microsoft Azure e quindi ho dovuto spendere diverso tempo nello studio del portale con le eventuali tariffe per l'affitto delle componenti del sito. Essendo un progetto a fini universitari ho potuto anche usufruire della convenzione che Microsoft Azure rende disponibile per gli studenti ovvero quella di: AZURE FOR STUDENT partire con un Budget di 100\$ così da avere il tempo necessario per implementare tutta la struttura della mia WebApp  
Dopo uno studio più approfondito del portale ho deciso di affittare un:

- Server Postgres singolo :  
Il nome del server di riferimento è: localhost12  
con sede in: East US.  
La configurazione a livello hardware che ho scelto nella fase di configurazione è la: Basic, con 1 vCore e 5 GB di memoria.  
Il sistema operativo che ho scelto nella fase di configurazione è: Linux  
La versione Postgres scelta nella fase di configurazione è stata la Versione PostgreSQL 11  
quello più adatto alla mie esigenze.  
Tutti i restanti parametri sono stati lasciati in modalità Default in quanto erano già configurati alle mie esigenze.
- Servizio App:  
Il nome del servizio di riferimento è: NomeSito  
con sede in: Central US.  
Il linguaggio di programmazione che è stato scelto nella fase di configurazione è: PHP con versione 8.2  
Di default è stato abilitato:
  - TLS (Transport Layer Security)  
è un protocollo di sicurezza che viene utilizzato per garantire la trasmissione sicura di informazioni su Internet, viene utilizzato per crittografare i dati in transito tra due sistemi, come ad esempio tra un browser web e

un server web.

Il protocollo funziona attraverso l'utilizzo di crittografia a chiave pubblica e privata, dove la chiave pubblica viene utilizzata per crittografare i dati trasmessi, mentre la chiave privata viene utilizzata per decriptare i dati. In questo modo, anche se un malintenzionato riesce ad intercettare i dati trasmessi, non sarà in grado di leggerli senza possedere la chiave privata corrispondente.

- per aumentare il monitoraggio della web App sono stati attivati i log di diagnostica dove è possibile avere un costante report su tutti i componenti del Servizio App.

## **6.2 sistema per rilasciare aggiornamenti nel Web App immediati e come collegare i vari componenti**

L'obiettivo era trovare un metodo semplice e veloce per rilasciare continui aggiornamenti della Web App, tra le varie opzioni ho deciso di utilizzare i seguenti servizi GitHub e Visual Studio Code;

- Visual Studio Code

Tramite questo editor di codice sorgente è stato possibile:

- Scrivere il codice sorgente per poi eseguire i vari commit direttamente nella repository di GitHub.
- Aggiungere l'estensione di Microsoft Azure così da gestire tutte le componenti del sito direttamente dall'editor e aggiornare la Web App tramite i vari Deploy.

- GitHub

Tramite il servizio GitHub è stato possibile collegare una repository direttamente alla relativa Web App tramite il portale di Microsoft Azure condividendo il link della repository in GitHub.

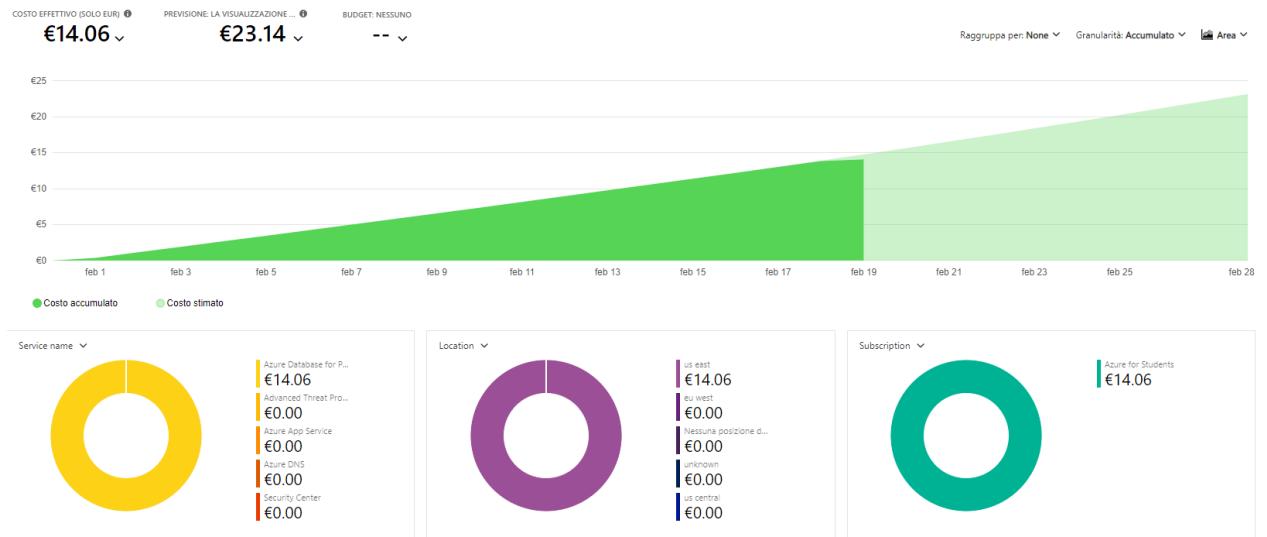
Per quanto riguarda la connessione al Server PostGres è stato possibile tramite la stringa di connessione che Microsoft Azure rende disponibile in vari linguaggi per quanto riguarda quella in PHP è:

```
host=localhost12.postgres.database.azure.com
port=5432
dbname={postgres}
user=postgres@localhost12
password={your _password}
sslmode=require
```

### 6.3 Riepilogo dei costi mensili

Un'altra problematica affrontata per Implementare la Web App in Cloud è la gestione dei costi in quanto si è dovuto trovare il giusto compromesso per avere il miglior bilancio Qualità/prezzo così da offrire un Servizio sempre disponibile ma allo stesso tempo anche conveniente. La gestione dei costi è un aspetto fondamentale del processo di gestione dei progetti, in quanto aiuta a garantire che il progetto venga completato entro il budget prestabilito e che non ci siano sorprese in termini di costi. Ecco alcuni motivi per cui è importante avere una gestione dei costi efficace:

- Budget: La gestione dei costi aiuta a stabilire un budget realistico per il progetto e a garantire che i costi siano controllati durante il suo svolgimento.
- Pianificazione: La gestione dei costi aiuta a pianificare il progetto in modo efficace, identificando le risorse necessarie e il tempo richiesto per il completamento del progetto.
- Controllo: La gestione dei costi consente di tenere traccia dei costi effettivi e di confrontarli con il budget previsto, consentendo di identificare eventuali discrepanze e di intervenire tempestivamente per correggere il corso del progetto.
- Decisioni: La gestione dei costi aiuta a prendere decisioni informate in merito al progetto, ad esempio se è possibile effettuare delle modifiche o aggiungere nuove funzionalità senza superare il budget prestabilito.
- Comunicazione: La gestione dei costi aiuta a comunicare in modo efficace con gli stakeholder del progetto, consentendo di fornire informazioni accurate e trasparenti sui costi e sullo stato di avanzamento del progetto.



**Figura 6.1.** Sezione inserimento prodotto.

# Capitolo 7

## Conclusioni

### 7.1 Tecnologie interessanti implementate

Di seguito vengono riportate alcune delle tecnologie più interessanti implementate nel sito

#### 7.1.1 Implementazione dei COOKIES

I cookies sono file di testo che vengono salvati sul computer o dispositivo mobile dell'utente quando visita un sito web. Questi file vengono utilizzati per immagazzinare informazioni sulle preferenze e attività del soggetto, in modo che il sito web possa personalizzare l'esperienza di navigazione e migliorare la tua interazione con esso.

I cookies possono essere utilizzati per una varietà di scopi, tra cui:

- Mantenere le tue preferenze, come la lingua o le impostazioni di visualizzazione, durante le visite successive al sito web.
- Tenere traccia dei tuoi comportamenti di navigazione sul sito web, in modo da fornire annunci pubblicitari pertinenti.
- Aiutare il sito web a monitorare l'utilizzo e l'affidabilità del sito stesso, in modo da migliorare la sua usabilità e le prestazioni.

Nel caso della Web App implementata da me sono stati usati per salvare i vari dati dell'utente nel caso avesse spuntato il checkBox RICORDAMI nella pagina Login.php così da non interrogare il server in continuazione.

Nella fase di progettazione ho scelto di settare la durata dei cookies di 24 ore.

#### 7.1.2 SQL injection

L'SQL Injection è un tipo di attacco informatico in cui un attaccante sfrutta una vulnerabilità presente in un'applicazione web per iniettare del codice SQL malevolo nel database dell'applicazione. Ciò può permettere all'attaccante di recuperare o modificare dati sensibili, come le credenziali degli utenti.

Per ovviare a tale problematica ho voluto inserire nella query per l'interrogazione al DataBase dei placeholder che poi verranno riempiti tramite la funzione pg\_query\_params.

```
//Apertura connessione al database

$query = 'SELECT * from utente where email=$1 AND pswd=$2';
$result = pg_query_params($dbconn, $query, array($email, $password));

//chiusura della connessione al database
```

### 7.1.3 Responsiveness

La responsività della applicazione web (responsiveness) è la proprietà per la quale un applicazione web può essere fruita correttamente sui diversi dispositivi mobili e non (e.g. PC desktop, smartphone, laptop, tablet). Per rendere la piattaforma responsiva è stato utilizzato il framework Bootstrap, che oltre a offrire diversi fogli di stile (CSS), offre strumenti che permettono di rendere responsivi gli elementi HTML che compongono le pagine dell'applicazione. Tra gli strumenti utilizzati in questo senso figurano:

- le media query: un insieme di direttive CSS che dovranno applicarsi a un dispositivo (media) di un certo tipo e/o avendo determinate caratteristiche (consentono al rendering del contenuto di adattarsi a diverse condizioni);
- alcuni componenti responsive offerti da Bootstrap: navbar, modal, list group, buttons e altri.

### 7.1.4 Validazione form utente lato client

Le regex (espressioni regolari) sono spesso utilizzate nei form per validare le informazioni immesse dall'utente, cioè per assicurarsi che l'utente abbia inserito i dati in un formato corretto.

Le regex possono essere utilizzate per validare numerosi tipi di dati immessi dagli utenti, nella Web App sono stati utilizzati per: indirizzi email, numeri di telefono, codici postali, password e altro ancora. In questo modo, le regex aiutano a prevenire l'invio di dati incompleti, errati o dannosi al server dell'applicazione, migliorando la sicurezza e l'usabilità dell'applicazione.

Inoltre, l'utilizzo di regex può anche semplificare la validazione dei dati all'interno del codice dell'applicazione, rendendo il codice più leggibile e manutenibile.

### 7.1.5 Implementazione dei Meta tags

I meta tag in un sito web sono informazioni che vengono incluse nell'intestazione (header) della pagina HTML, ma non vengono visualizzati direttamente nella pagina web essi forniscono informazioni sui contenuti della pagina web, sulla sua struttura e sui metodi di indicizzazione dei motori di ricerca.

I meta tag inseriti all'interno del sito sono di tipo:

- Il meta tag "description" fornisce una breve descrizione del contenuto della pagina web. Questa descrizione può essere utilizzata dai motori di ricerca per fornire un'anteprima della pagina nei risultati di ricerca.

- Il meta tag "keywords" fornisce una lista di parole chiave che descrivono il contenuto della pagina web. Questo tag veniva utilizzato in passato dai motori di ricerca per indicizzare le pagine, ma oggi ha meno importanza a causa della diffusa pratica di spamming.
- Il meta tag "viewport" viene utilizzato per definire la visualizzazione ottimale della pagina su dispositivi mobili. Questo tag aiuta a garantire che la pagina web sia visualizzata correttamente su diverse dimensioni di schermo e dispositivi mobili.

All'interno del sito sono stati aggiunti alcuni meta tag per facilitare la ricerca della pagina web.

## 7.2 Tecnologie future

### 7.2.1 Implementazione della verifica in 2 passaggi

L'implementazione della verifica in 2 passaggi è un modo efficace per proteggere l'accesso agli account online. In sostanza, la verifica in 2 passaggi richiede ai utenti di fornire non solo una password, ma anche un codice di verifica aggiuntivo, che viene generato da un dispositivo di sicurezza come un telefono cellulare o un'applicazione.

Ecco come funziona l'implementazione della verifica in 2 passaggi:

1. L'utente immette il proprio nome utente e la password per accedere all'account online.
2. Il sistema richiede un codice di verifica aggiuntivo, che può essere generato da un'applicazione per smartphone, un token fisico o via SMS.
3. L'utente inserisce il codice di verifica, che viene verificato dal sistema.
4. Se il codice di verifica è corretto, l'utente viene autenticato e può accedere al proprio account.

### 7.2.2 Recupero della password

L'implementazione del recupero della password è un'importante funzionalità per un'applicazione o un servizio web che richiede l'autenticazione degli utenti. Il recupero della password consente agli utenti di reimpostare la propria password in caso di dimenticanza o smarrimento.

### 7.2.3 Algoritmo per il calcolo della destinazione migliore

In futuro si potrebbe pensare di implementare un algoritmo per il calcolo della destinazione migliore, ovvero la zona più bisognosa in base ai prodotti inseriti dall'utente, quindi passare da una sola zona di interesse per volta a più zone in base alle tipologie di prodotti che si inviano.

### 7.2.4 Algoritmo per il calcolo del locker più vicino

Ulteriore Algoritmo da inserire potrebbe essere il calcolo del locker più vicino tramite l'attivazione del GPS dal proprio dispositivo su cui si effettua la donazione o il locker più vicino al proprio indirizzo di residenza.

### 7.2.5 implementazione CAPTCHA

CAPTCHA è un acronimo di "Completely Automated Public Turing test to tell Computers and Humans Apart". È un tipo di test di sicurezza utilizzato per determinare se l'utente che sta tentando di accedere a un sito web o di eseguire un'operazione su di esso è un essere umano o un programma automatizzato (ad esempio, un bot).

I CAPTCHA funzionano mostrando all'utente una sfida che può essere completata solo da un essere umano, come ad esempio:

- Inserire una serie di lettere o numeri distorti e difficile da leggere in un campo di testo.
- Selezionare tutte le immagini che corrispondono ad una specifica categoria (ad esempio, selezionare tutte le immagini che mostrano un semaforo).
- Risolvere un puzzle o un problema matematico semplice.

In genere, i CAPTCHA sono utilizzati per proteggere i siti web dagli attacchi di spam o di bot automatizzati, che possono causare danni all'integrità dei dati o alla sicurezza dell'applicazione.

### 7.2.6 pop-up per la privacy e la cookies policy

I pop-up per la privacy e i cookie sono una funzionalità comune che si trova spesso sui siti web moderni. Questi popup vengono utilizzati per informare gli utenti sulle politiche del sito riguardo alla privacy e ai cookie e per richiedere il consenso degli utenti per l'utilizzo di cookie e altre tecnologie di tracciamento.

### 7.2.7 accessibilità

L'accessibilità è un requisito importante per utilizzare e interagire con il sito web in modo efficace e completo. Ci sono molte linee guida e standard internazionali che gli sviluppatori web possono seguire per rendere i loro siti web accessibili.

Ecco alcune delle best practice per migliorare l'accessibilità di un sito web:

- Utilizzare il markup semantico: Utilizzare correttamente gli elementi HTML e associare etichette descrittive ai form elementi consentirà ai visitatori con disabilità di comprendere meglio la struttura e il contenuto del sito web.
- Fornire descrizioni alternative per le immagini: Le descrizioni alternative per le immagini devono essere fornite per le immagini di contenuto, al fine di consentire ai visitatori con disabilità visive di comprendere il contenuto delle immagini.
- Garantire un alto contrasto tra il testo e lo sfondo: Ciò aiuta i visitatori con disabilità visive a leggere il contenuto del sito web.
- Fornire una navigazione chiara e coerente: Una navigazione chiara e coerente aiuta i visitatori a trovare facilmente il contenuto del sito web. L'utilizzo di un modello di navigazione familiare, come un menu di navigazione sul lato sinistro della pagina, può aiutare i visitatori a orientarsi all'interno del sito.

- Utilizzare una sintassi semplice e facile da comprendere: Utilizzare una sintassi semplice e facile da comprendere per il contenuto del sito web, aiuta i visitatori con disabilità cognitive e linguistica a comprendere meglio il contenuto del sito.
- Garantire la compatibilità con le tecnologie assistive: L'utilizzo di tecniche web compatibili con le tecnologie assistive come lettori di schermo, dispositivi Braille, ecc., consente ai visitatori con disabilità di accedere al contenuto del sito web.
- Testare l'accessibilità: Testare il sito web con strumenti di testing dell'accessibilità come il WAVE, può aiutare gli sviluppatori a identificare e risolvere problemi di accessibilità.

### 7.2.8 Lingua

Per aumentare il più possibile la diffusione della Web App si deve avere varie traduzioni del sito almeno per le lingue principali.

# Sitografia

- [1] Cloud Storage Google Cloud, <https://cloud.google.com/storage?hl=it>
- [2] Microsoft Azure, <https://azure.microsoft.com/it-it>
- [3] Linguaggio SQL, <https://www.html.it/guide/guida-linguaggio-sql/>
- [4] PostgreSQL, <https://www.postgresql.org/>
- [5] Cloud Computing, <https://www.redhat.com/it/topics/cloud-computing/iaas-vs-paas-vs-saas>
- [6] Visual Studio code, <https://code.visualstudio.com/>
- [7] Requisiti per l'Accessibilità, <https://www.ipsoa.it/documents/quotidiano/2022/08/02/siti-web-norme-accessibilita-parte-persone-disabilita>
- [8] Requisiti per la sicurezza in un sito web, <https://blog.register.it/7-step-per-garantire-la-sicurezza-del-tuo-sito-web/>
- [9] Repository Web App su GitHub, <https://github.com/Alessiorevo1000/NomeSito>
- [10] Vantaggi del Cloud Computing, <https://www.criticalcase.com/it/blog/i-vantaggi-del-cloud-computing.html>
- [11] Framework Bootstrap, <https://getbootstrap.com/>
- [12] Gestione cookies, <https://guidaphp.it/base/variabili-superglobali/cookie>
- [13] Introduzione al linguaggio PHP, <https://www.php.net/>
- [14] Progettazione di una base di Dati, <https://www.datamaze.it/blogs/post/sviluppo-dei-database-l-importanza-della-progettazione>

*Desidero prendere un breve momento per ringraziare coloro che, con impegno e pazienza, mi sono stati vicino durante il mio percorso universitario.*

*Vorrei quindi ringraziare il mio relatore Andrea Sterbini per la sua disponibilità e per i suoi consigli preziosi che mi ha fornito durante la stesura della tesi.*

*Desidero esprimere la mia sincera gratitudine a Manuela Grande per la sua disponibilità, la sua guida e la sua pazienza durante tutto il mio percorso di tirocinio e per avermi fatto sentire parte del team.*