



Università di Pisa

Corso di Laurea in Informatica Umanistica

RELAZIONE

Personalytics

Un'applicazione web per esplorare i propri dati personali

Candidato: *Cono Stabile*

Relatori: *Dr. Anna Monreale, Dott. Riccardo Guidotti*

Correlatore: *Dott.ssa Beatrice Rapisarda*

Anno Accademico 2015-2016



Indice

Introduzione.....	1
1. Il database Unicoop.....	2
1.1 Concetti generali.....	2
1.2 Modello concettuale UML	3
1.3 PostgreSQL.....	4
1.4 Il progetto LivLab.....	5
1.4.1 Schema UML di Unicoop	5
1.4.2 PostgreSQL: creazione di Unicoop	7
1.4.3 Il popolamento del database	10
1.4.4 Descrizione delle tabelle Unicoop.....	11
2. La struttura web.....	15
2.1 HTML.....	15
2.2 CSS.....	16
2.3 Personalytics	17
2.3.1 Architettura di Personalytics	17
2.3.2 Il design	20
3. I dati come infografici.....	22
3.1 PHP	22
3.2 Apache http server.....	23
3.2.1 XAMPP	23
3.3 I dati in JSON.....	24
3.4 Data Visualization.....	26
2.3.1 D3.JS	26
2.3.2 Raphaël.JS e Morris.JS.....	27
3.5 I dati da visualizzare.....	27
3.2.1 I grafici di Personalytics.....	28
3.3 Entropia.....	34
3.2.1 L'entropia per misurare la sistematicità del cliente.....	35

4. La mobilità	38
4.1 Google Maps	38
4.1.1 Personalitics e Google Maps	39
4.1.2 L'inserimento dei marker	41
4.1.3 La clusterizzazione dei marker.....	43
4.1.4 Il centro di massa	44
4.2 Radius of gyration	46
4.3 Java.....	47
4.3.1 Trajectory builder	48
4.3.2 Trajectory builder: la tabella Traiettoria	49
5. Shopping list	51
5.1 La frequenza come suggerimento	51
5.2 Da suggerimento a prodotto da acquistare	53
6. Caso di studio	55
6.1 Comportamento in base agli acquisti	55
6.2 Comportamento in base alla sistematicità.....	56
6.3 Comportamento in base alla mobilità	58
7. Conclusioni	60
8. Bibliografia e sitografia	61
9. Appendice delle tabelle Unicoop	63

Introduzione

L'obiettivo di questa tesi è quello di descrivere in maniera dettagliata il motivo e il modo in cui ho realizzato Personalytics. L'idea della suddetta web application nasce con lo scopo di facilitare il controllo dei movimenti relativi al singolo utente. Tali movimenti riguardano principalmente i propri dati di consumo e di mobilità. Fondamentale per lo svolgimento del mio lavoro è stato lo studio del database LivLab all'interno del DBMS Oracle, realizzato dal team Living Laboratory di Livorno. Tale database è costituito da diversi dati messi a disposizione da Unicoop Tirreno attraverso il consenso dei propri soci.

A partire da un modello concettuale UML, il database che costituisce la base di partenza dell'intera applicazione web è stato realizzato all'interno del DBMS PostgreSQL e successivamente popolato attraverso il linguaggio di programmazione Python.

Dopo aver progettato la struttura web dell'applicazione, i dati all'interno del database sono stati recuperati attraverso una serie di interrogazioni mediante i linguaggi di programmazione PHP e SQL, e proiettati graficamente attraverso l'utilizzo di alcune librerie realizzate in JavaScript. Per alcuni dati sono stati effettuati calcoli statistici e non, al fine di individuare caratteristiche ben precise per ciascun cliente.

1. Il database Unicoop

Partendo dall'analisi del database LivLab sono state esaminate alcune tabelle che hanno portato alla creazione dello modello concettuale del database Unicoop. Tale schema concettuale è stato tradotto poi in modello relazionale e implementato nel DBMS PostgreSQL e successivamente popolato mediante il linguaggio di programmazione Python. Di seguito verranno introdotti prima alcuni concetti di base sui DBMS e la progettazione di una base di dati e poi verrà illustrata la progettazione del database Unicoop.

1.1 Concetti generali

Un Database è una collezione di uno o più archivi che vengono organizzati e strutturati per facilitare e rendere ottimali le operazioni di interrogazione, creazione e cancellazione. Per gestire una base di dati vengono utilizzati i DBMS (*database management system*). Questi ultimi sono software che consentono di costruire e gestire una base di dati realizzandola su una memoria di massa. I DBMS offrono diverse funzioni quali:

- Implementazione del modello logico sul sistema di elaborazione;
- Manipolazione e interrogazione;
- Sicurezza e protezione;
- Controllo dell'integrità dei dati.

Tali funzioni che i DBMS offrono vengono attivate attraverso dei semplici linguaggi con il quale l'utente entra in comunicazione con il software. Essi sono:

- DDL (*Data Definition Language*) permette di definire la struttura delle relazioni del database.
- DML (*Data Manipulation Language*) permette di modificare i dati contenuti nel database, con le operazioni di inserimento, modifica e cancellazione.
- DCL (*Data Control Language*) permette di gestire il controllo degli accessi e i permessi per gli utenti.

- SQL (*Structured Query Language*) permette di effettuare interrogazioni alla base di dati.

La progettazione di una base di dati si basa su dei processi decisionali che coinvolgono essenzialmente tre modelli fondamentali:

1. Modello concettuale dei dati: rappresenta, attraverso uno schema grafico, la realtà dei dati e le relazioni tra di essi.
2. Modello logico: traduzione del modello concettuale in modello relazionale a seconda del DBMS adottato.
3. Modello fisico: rappresenta l'effettiva installazione degli archivi elettronici nelle memorie di massa.

Fondamentale è il concetto di tabella. Questa non è altro che la struttura dati fondamentale di un database relazionale. Attraverso le tabelle si rappresentano le entità e le relazioni dello schema concettuale. Una tabella è composta da *record* (righe o tuple) e da campi (colonne o attributi): ogni record rappresenta una istanza (o occorrenza o tupla) dell'entità/relazione, ogni campo, invece, rappresenta un attributo dell'entità/relazione. Per ogni campo viene individuato un suo dominio (tipo di dato) che può essere alfanumerico, numerico, data, booleano, etc [1].

1.2 Modello concettuale UML

L'*Undefined Modeling Language*, comunemente chiamato *UML* è un metodo per descrivere l'architettura di un sistema in dettaglio. Esso fornisce una vista strutturale, ovviamente statica, del sistema in termini di classi (attributi e operazioni) e relazioni tra classi (associazioni, generalizzazioni, etc.).

L'utilizzo di UML è fondamentale prima che la fase progettuale venga realizzata in quanto permette di conoscere in anticipo il risultato del progetto su cui si sta lavorando. Inoltre, la fase del disegno progettuale precede la fase della scrittura del codice da parte degli sviluppatori, di conseguenza si abbassano in maniera notevole anche i costi di sviluppo ed è più facile evidenziare e rimediare a possibili errori piuttosto che nella fase sviluppo.

Utilizzando i diagrammi UML chiunque sia coinvolto nello sviluppo del progetto, può avere un'idea concreta di tutto l'insieme che costituisce il sistema. In questo

modo, la comunicazione e l'interazione tra le diverse risorse umane che prendono parte al progetto è molto più semplice e diretta.

1.3 PostgreSQL

PostgreSQL¹ è un ORDBMS (*Object Relational Database Management System*) di classe enterprise. Esso è stato sviluppato nel 1986, all'University of California nel dipartimento di informatica di Berkeley, dall'illustre Professore Michael Stonebraker e dal suo team sulla base del progetto Ingres sviluppato precedentemente. La prima release del software fu rilasciata nel 1995. A partire da quel momento PostgreSQL ha assistito a diverse versioni fino ad arrivare alla 9.6.1 rilasciata il 27 ottobre 2016.

PostgreSQL, come suggerisce già di per sé il nome, sfrutta il linguaggio SQL e rappresenta un'ottima alternativa ai DBMS liberi come MySQL, che a quelli a codice chiuso come Oracle. PostgreSQL offre diverse peculiarità, tra le principali:

- software open source: gli utenti possono installare, usare e distribuire PostgreSQL senza pagare grazie a una licenza molto permissiva;
- elevata aderenza agli standard SQL;
- supporto completo per Java, Python, Perl, PHP e molti altri linguaggi di programmazione, sia per le procedure interne al server di database che per l'accesso da parte di client;
- elevata affidabilità;
- possibilità di aggiungere l'estensione PostGIS che consente di memorizzare sulla base di dati oggetti di tipo GIS (*Geographic Information System*) e di interrogarli;
- possibilità di definire tipi di dati personalizzati;
- possibilità di gestire più Database contemporaneamente;
- la dimensione di ogni database può essere illimitata;
- ogni tabella può avere una dimensione massima di 32 TeraByte;

¹ www.postgresql.org

- il numero di righe per ogni tabella è illimitato;
- numero massimo di colonne per tabella varia da 250 a 1600 a seconda del tipo di ciascuna colonna.

1.4 Il progetto LivLab

Il Living Laboratory (LivLab), situato nei pressi di Livorno, è un laboratorio di sperimentazione per i servizi dedicati alla spesa che nasce dalla collaborazione tra partner industriali (Telecom Italia e Unicoop Tirreno) e accademici (ISTI-CNR² e FBK – Fondazione Bruno Kessler³) [11].

Il progetto LivLab, iniziato nell'Aprile 2015 e tutt'ora in corso, prevede il coinvolgimento di cento soci della Unicoop Tirreno che sono diventati volontariamente protagonisti di un esperimento sociale incentrato sui Big Data: le loro spese e i loro movimenti all'interno delle sedi Unicoop sono stati tracciati, memorizzati e poi elaborati, in modo da provare a migliorare le esperienze di acquisto. Tali soci-sperimentatori hanno reso fruibili i propri dati di consumo e mobilità attraverso l'uso delle tessere della catena nel corso degli acquisti, dall'utilizzo dello smartphone e specifiche applicazioni sviluppate nel Joint Open Lab di Telecom Italia [12].

1.4.1 Schema UML Unicoop

Dopo aver effettuato un'analisi esplorativa, e alcune fasi di processing per la pulizia delle tabelle disponibili nel database LivLab, la fase di data preparation ha portato alla creazione di tredici tabelle ritenute utili al fine dell'analisi a differenza delle trentadue tabelle presenti in LivLab. Le tabelle prese in considerazione sono: Cliente Device, Location_Point, Sms, Chiamata, Negozio, Articolo, Venduto, Pagamento, Marketing, Promo, Promodettaglio e Promomeccanica.

Tali tabelle sono state sottoposte ad una successiva analisi al fine di estrapolare gli attributi necessari per l'obiettivo da realizzare. A tal proposito è lecito introdurre alcuni concetti quali: *tabella* che è la rappresentazione grafica normalmente accettata

² www.isti.cnr.it

³ www.fbk.eu

1.4.2 PostgreSQL: creazione di Unicoop

Il modello relazionale rappresenta il modo attraverso il quale i dati sono organizzati all'interno degli archivi elettronici: descrive la composizione e il formato dei dati nell'aspetto di struttura logica dei dati.

Si definisce modello relazionale perché è fondato sul concetto matematico di relazione.

Dati $n > 0$ insiemi D_1, D_2, \dots, D_n , non necessariamente distinti, il prodotto cartesiano di D_1, D_2, \dots, D_n , indicato con $D_1 \times D_2 \times \dots \times D_n$, è costituito dall'insieme delle n -uple (v_1, v_2, \dots, v_n) tali che v_i appartiene a D_i , per $1 \leq i \leq n$. Una relazione matematica sui domini D_1, D_2, \dots, D_n è un sottoinsieme del prodotto cartesiano $D_1 \times D_2 \times \dots \times D_n$. Il numero n delle componenti del prodotto cartesiano (e quindi di ogni n -upla) viene detto *grado* del prodotto cartesiano e della relazione. Il numero di elementi (cioè di n -uple) della relazione è chiamato *cardinalità* della relazione⁵[1].

Ogni relazione, viene rappresentata come una tabella costituita da tante colonne quanti sono i domini, e da tante righe quante sono le n -uple. I nomi dei domini corrispondono ai nomi delle singole colonne, e i valori che compaiono all'interno delle colonne devono essere omogenei tra di loro.

Il modello relazionale viene derivato dal modello concettuale, nel caso di Unicoop dall'UML, secondo alcune regole standard di derivazioni:

1. ogni entità diventa tabella;
2. ogni attributo diventa campo della tabella;
3. ogni colonna della tabella eredita le caratteristiche dell'attributo dell'entità da cui deriva;
4. l'identificatore univoco di ogni entità diventa chiave primaria (*primary key*) della tabella derivata;
5. l'associazione uno a uno diventa un'unica tabella che contiene gli attributi della prima e della seconda entità;

⁵ P. Atzeni et. al., *Basi di dati: Modelli e linguaggi di Interrogazione*, McGraw-Hill Libri Italia, Terza edizione, 2009. Cit. p. 19

6. l'identificatore univoco dell'entità di partenza nell'associazione uno a molti diventa chiave esterna (*foreign key*) dell'entità di arrivo associata, cioè i suoi attributi - identificatori univoci - diventano colonne della seconda relazione;
7. l'associazione molti a molti diventa una nuova tabella (in aggiunta alle tabelle derivate dalle entità) composta dagli identificatori univoci delle due entità e dagli eventuali attributi dell'associazione.

Unicoop è stato creato in PostgreSQL v.9.5.4 seguendo tali regole di derivazione, utilizzando come sistema di codifica UTF-8. Qui di seguito è riportato un estratto del SQL codice che mostra la creazione del database Unicoop in PostgreSQL.

```
CREATE DATABASE unicoop
  WITH OWNER = postgres
      ENCODING = 'UTF8'
      TABLESPACE = pg_default
      LC_COLLATE = 'Italian_Italy.1252'
      LC_CTYPE = 'Italian_Italy.1252'
      CONNECTION LIMIT = -1;
```

Ogni tabella di Unicoop è stata creata seguendo i vincoli di integrità dei dati. Essi si suddividono in due tipologie:

1. intrarelazionali: che riguardano la singola tabella;
2. extrarelazionali: che riguardano legami tra due o più tabelle.

Tra i vincoli Intrarelazionali vi sono:

- chiave Primaria (*Primary Key*): identifica univocamente le singole righe della tabella dove essa è definita;
- *unique*: assume le stesse caratteristiche della Primary Key con la possibilità di definire nullo l'attributo coinvolto;
- vincoli di tupla: agiscono sul singolo record indipendentemente dagli altri e permettono di imporre una restrizione che può essere dichiarata attraverso espressioni algebriche, operatori logici, *NOT NULL*, etc.

Tra i vincoli extrarelazionali vi è il vincolo di integrità referenziale, comunemente conosciuto come vincolo di chiave esterna o *foreign key*. Un vincolo di integrità referenziale fra un insieme di attributi X di uno schema di relazione $R1$ e uno schema di relazione $R2$ impone ai valori su X di ciascuna tupla di $R1$ di comparire come valori della chiave primaria dell'istanza di $R2$ [1].

Qui di seguito è riportato un estratto del codice SQL utilizzato per la creazione della tabella “Negozio” con i vincoli di integrità precedentemente citati.

```
...  
  
CREATE TABLE public.negozio  
  
(  
  
    id_negozio bigint NOT NULL DEFAULT  
nextval('negozio_id_negozio_seq'::regclass),  
  
    cod_negozio integer NOT NULL,  
  
    nome character varying(100) NOT NULL,  
  
    indirizzo character varying(100) NOT NULL,  
  
    ... ,  
  
    cod_negozio_padre integer,  
  
    CONSTRAINT negozio_pkey PRIMARY KEY (id_negozio),  
  
    CONSTRAINT negozio_canale_check CHECK (canale::text = 'S'::text OR  
canale::text = 'I'::text OR canale::text = 'G'::text OR canale::text =  
'E'::text)  
  
)  
  
...
```

1.4.3 Il popolamento del database

Dopo aver creato Unicoop con le relative relazioni in PostgreSQL, le tabelle del database LivLab prese in considerazione sono state esportate dall'ambiente Oracle⁶ in formato CSV (Comma Separated Value) e successivamente sottoposte ad una ripulitura mediante Google Refine⁷. Quest'ultimo è un potentissimo tool che permette di effettuare diverse operazioni su tabelle di dati. Le principali operazioni utilizzate per la pulizia delle tabelle del database LivLab sono:

- *filtraggio e partizionamento* dei dati mediante l'uso di espressioni regolari.

Lo step successivo riguarda la fase di popolamento del database Unicoop con le tabelle aggiornate e la scelta di Python come linguaggio di programmazione per effettuare tale popolamento.

Python⁸, ideato da Guido van Rossum all'inizio degli anni Novanta, è un linguaggio di programmazione dinamico e ad alto livello, orientato agli oggetti utilizzabile per molti tipi di sviluppo software. Tra le caratteristiche principali di questo linguaggio di programmazione sono le variabili non tipizzate e l'uso dell'indentazione per la definizione dei blocchi. Attraverso tale linguaggio è possibile definire velocemente degli script per effettuare semplici compiti, come per esempio quello di trasferire dei dati da un database all'altro. Esso è un forte supporto all'integrazione con altri linguaggi e programmi, ed è fornito di una libreria standard, a sua volta estendibile con altri moduli, che lo rende adatto a molti impieghi. Le librerie utilizzate per la fase di popolamento sono:

- *psycopg*: libreria utilizzata per avviare la connessione con il database Unicoop con il DBMS PostgreSQL;
- *pandas*: libreria utilizzata per leggere e manipolare dati multidimensionali.

La realizzazione del codice è stata effettuata sull'ambiente di sviluppo PyCharm⁹. Si tratta di un tool che presenta grande velocità di editing testuale data dalla possibilità

⁶ www.oracle.com

⁷ www.openrefine.org

⁸ www.python.org

⁹ www.jetbrains.com

di riconoscere errori di sintassi e di indentazione che vengono prontamente sottolineati dal software.

Qui di seguito è riportato un estratto del codice Python che sfrutta la libreria `psycopg`:

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import psycopg2

try:
    con = psycopg2.connect(database='unicoop', user='postgres')
    cur = con.cursor()
    cur.execute('SELECT version()')
    ver = cur.fetchone()
    print "Connessione abilitata"

except psycopg2.DatabaseError, e:
    print 'Error %s' % e
    sys.exit(1)
```

1.4.4 Descrizione delle tabelle di Unicoop

Le tabelle riportano in maniera dettagliata le caratteristiche di ogni relazione in cui viene indicato il nome dell'attributo, la funzione del medesimo, il tipo di dato scelto e la sua descrizione. Qui di seguito è riportata la descrizione di alcune tabelle significative: Cliente, Negozio e Venduto.

Cliente

ATTRIBUTO	FUNZIONE	TIPO DI DATO	DESCRIZIONE
Id_cliente	Primary Key	Incremental	Incrementale progressivo e chiave primaria della tabella
Nro_cliente	Unique	Numeric	Numero identificativo cliente
Categoria		Character varying	Specifica la cooperativa di appartenenza del cliente (Unicoop Tirreno, Non socio, etc.)
Nro_carta		Integer	Numero carta del cliente se socio
Nome		Character varying	Nome del cliente
Cognome		Character varying	Cognome del cliente
Data_Nascita		Date	Data di nascita del cliente
Sesso		Character	Sesso del cliente (M/F)
Stato_civile		Character varying	Stato civile del cliente (celibe, nubile, etc.)
Professione		Character varying	Professione del cliente
Titolo_studio		Character varying	Titolo di studio del cliente
Indirizzo		Character varying	Indirizzo del cliente
CAP		Integer	Codice di avviamento postale del cliente
Città		Character varying	Città del cliente
Provincia		Character	Provincia del cliente
Regione		Character varying	Regione in cui risiede il cliente
Telefono		Character varying	Numero di telefono del cliente
Email		Character varying	Indirizzo e-mail
Id_negozio	ForeignKey	Bigint	Chiave esterna che punta alla tabella negozio
Id_device	ForeignKey	Bigint	Chiave esterna che punta alla tabella device

Tabella 1. Cliente

Negozio

ATTRIBUTO	FUNZIONE	TIPO DI DATO	DESCRIZIONE
Id_negozio	PrimaryKey	Incremental	Incrementale progressivo e chiave primaria della tabella
Cod_negozio	Unique	Integer	Codice identificativo del negozio
Nome		Character varying	Nome del negozio
Indirizzo		Character varying	Indirizzo del negozio
Località		Character varying	Località del negozio
Provincia		Character	Provincia della località del negozio
Regione		Character varying	Regione del negozio
CAP		Integer	Codice di avviamento postale del negozio
Canale		Character varying	Canale del negozio (S per super, I per Iper, G per GestInCoop, ed E per E-commerce)
Cod_negozio_padre		Integer	Codice del negozio padre (negozio che è stato chiuso e riaperto sottoforma di un nuovo canale)
Canale_padre		Character varying	Canale del negozio padre (S per super, I per Iper, G per GestInCoop, ed E per E-commerce)
Padre_nome		Character varying	Nome del negozio padre

Tabella 2. Negozio

Venduto

ATTRIBUTO	FUNZIONE	TIPO DI DATO	DESCRIZIONE
id_venduto	PrimaryKey	Incremental	Incrementale progressivo e chiave primaria della tabella venduto
id_cliente	ForeignKey	Incremental	Chiave esterna che punta alla tabella cliente
id_negozio	ForeignKey	Incremental	Chiave esterna che punta alla tabella venduto
id_Articolo	ForeignKey	Incremental	Chiave esterna che punta alla tabella articolo
Cassa		Integer	Numero della cassa dove è avvenuto il fatto di vendita
Scontrino		Integer	Numero scontrino
Importo		Money	Importo della vendita
Sconto		Decimal	Sconto applicato alla vendita
Altri_sconti		Decimal	Ulteriori sconti applicati alla vendita
QTA_pezzi		Integer	Quantità dei pezzi venduti
QTA_peso		Decimal	Quantità di peso dei pezzi venduti
Data_ora		Numeric	Data e ora di vendita in formato Unix

Tabella 3. Venduto

2. La struttura web

Dopo aver scelto il nome dell'applicazione, ovvero Personalitics, è stata realizzata la struttura di quest'ultima mediante il linguaggio HTML. Successivamente è stata definita la grafica utilizzando il fogli di stile CSS.

2.1 HTML

HTML¹⁰ (*HyperText Markup Language*) è un linguaggio di markup utilizzato per trasformare documenti testuali in pagine web. Con il termine markup, originario dell'ambiente tipografico, si allude ad un insieme di regole che descrivono i meccanismi di rappresentazione, strutturali, semantici o presentazionali di un testo. Esistono due tipologie di markup:

1. *procedurale*: specificano quali sono le procedure di trattamento del testo e indicano le istruzioni da eseguire affinché la porzione di testo referenziata possa essere visualizzata;
2. *descrittivo*: lascia che sia il software a scegliere quale rappresentazione debba essere applicata al testo.

HTML è un linguaggio di markup descrittivo, e una tra le più importanti caratteristiche di quest'ultimo è che assicura che tra la struttura e la visualizzazione del testo vi sia una corretta separazione.

L'HTML, come precedentemente detto, viene impiegato per formattare e per impaginare i documenti ipertestuali. Esso è stato sviluppato da Tim Berners-Lee alla fine degli anni '80. Questo linguaggio di formattazione descrive le modalità di visualizzazione grafica – quello che viene definito come layout – del contenuto di una pagina web (contenuto che può essere sia in forma testuale che in forma non testuale) tramite dei tag di formattazione.

È importante non confondere l'HTML con un linguaggio di programmazione, visto che si tratta di due realtà ben diverse. L'HTML non può essere ritenuto un linguaggio di programmazione perché, anche se supporta l'inserimento di oggetti esterni come filmati e immagini, non presuppone alcuna definizione di strutture di controllo, di funzioni, di strutture dati o di variabili in grado di realizzare programmi.

¹⁰ www.html.it

Una pagina web realizzata con HTML può presentare diverse strutture, ma quella basilare è costituita da un' intestazione e dal corpo del documento; ogni documento HTML inizia e termina con il tag `<HTML></HTML>`.

Nel corso degli anni si è assistito a diverse versioni di questo linguaggio di markup, l'ultima, utilizzata anche per creare la struttura della web application, è la 5.

2.2 CSS

CSS (*Cascading Style Sheets*), chiamati anche semplicemente fogli di stile, è un linguaggio utilizzato per descrivere la presentazione visuale di un documento HTML. Esso è stato introdotto nel 1996 dal World Wide Web Consortium¹¹, anche conosciuto come W3C, un'organizzazione che ha come scopo quello di sviluppare tutte le potenzialità del World Wide Web.

Grazie ai fogli di stile CSS è possibile definire, attraverso delle semplicissime regole, il modo in cui debbano essere presentati alcuni elementi di un documento HTML come il colore, la dimensione e lo stile del testo, ma anche elementi più complessi come il layout di pagina, le sfumature, etc. Una regola CSS è del tipo:

```
selettore{proprietà: valore};  
body{background-color: black};
```

I fogli di stile possono essere applicati ad un documento HTML in tre modi diversi:

1. Stili inline: lo stile viene applicato direttamente all'elemento HTML;
2. Fogli di stili incorporati: il foglio di stile viene implementato all'inizio documento HTML mediante l'elemento `style` che deve comparire all'interno dell'intestazione;
3. Fogli di stile esterni: gli stili vengono specificati su un file distinto. Tale procedimento risulta essere più vantaggioso in quanto è possibile applicare lo stesso stile a tutte le pagine che fanno riferimento a esso con una sola operazione.

Per la realizzazione dello stile delle pagine HTML della web application si è utilizzato CSS3 (l'ultima versione) su un file separato, collegando quest'ultimo a tutte le pagine del progetto.

¹¹ www.w3schools.com

2.3 Personalitics

Peronalytics è il nome scelto per la web application. Esso, come è possibile dedurre dal termine, è composto da due parole:

1. Persona: dal latino *persōna*, indica un individuo della specie umana, senza distinzione di sesso, età, condizione sociale e sim., considerato sia come elemento a sé stante, sia come facente parte di un gruppo o di una collettività¹².
2. Analisi: dal greco *anàlysis*, allude ad un metodo di studio che consiste nella scomposizione di un tutto, concreto o astratto, nelle parti [...] allo scopo di renderlo più razionale ed efficiente¹³.

La scelta di unire questi due sostativi è legata a quella che è la funzione del progetto: la persona, ovvero il cliente, può effettuare una serie di analisi. Queste ultime non sono altro che il controllo e il confronto sui propri dati di consumo e di mobilità.

2.3.1 Architettura di Personalitics

La struttura web di Personalitics è suddivisa in sei pagine HTML distinte, a ciascuna delle quali è collegato lo stesso foglio di stile CSS:

- Login
- Index
- Acquisti
- Profilo
- Mobilità
- Spesa

Qui di seguito è riportato uno schema logico della struttura web.

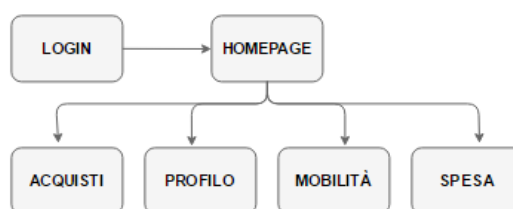


Figura 2. Schema logico della struttura web di Personalitics.

¹² www.treccani.it/vocabolario/persona

¹³ www.treccani.it/vocabolario/analisi

La pagina “Login” (vedi Figura 3) è costituita da un logo realizzato con il software Adobe Illustrator¹⁴ dal quale, con un semplice click, è possibile avviare un form che permette all’utente di loggarsi all’interno del sito mediante il proprio numero di carta socio COOP.



Designed by
C. Stabile

Figura 3. Login.html

Dopo aver effettuato l’accesso attraverso il numero carta, è possibile accedere all’ “Homepage” (vedi Figura 4) della web application caratterizzata dalla presenza di un’animazione, messa a disposizione dalla piattaforma codepen¹⁵, che allude alla funzione di Personalytics. L’homepage presenta il titolo Personalytics posto al centro, e un header posto in basso: a destra è presente il logo dell’applicazione dal quale è possibile accedere ad un modal dialog che in maniera sintetica descrive il contenuto di Personalytics, mentre alla sinistra è presente un menù dal quale è possibile accedere alle altre pagine della web application.



Figura 4. Index.html

¹⁴ www.adobe.com

¹⁵ www.codepen.io

La pagine “Acquisti” (vedi Figura 5), “Profilo” e “Mobilità” permettono al cliente loggatosi di effettuare una serie di confronti e di controlli sui propri dati di consumo e di mobilità attraverso l’interazioni tra grafici che verranno spiegati nel capitolo successivo.

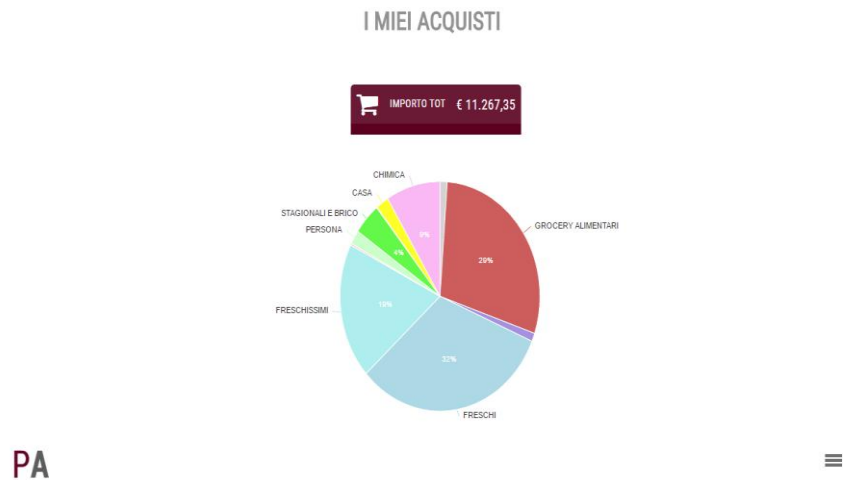


Figura 5. Acquisti.html

L’ultima pagina di Personalitics è “Spesa” (vedi Figura 6). In quest’ultima è possibile compilare una vera e propria lista della spesa che viene generata mediante dei suggerimenti. Questi non sono altro che i prodotti che il cliente acquista con più frequenza. Cliccando sul suggerimento è possibile eliminarlo dalla lista dei suggerimenti e aggiungerlo nel blocco “lista della spesa”. Inoltre, è possibile editare un prodotto dalla tastiera e cliccando sul bottone “+” aggiungerlo alla lista.



Figura 6. Spesa.html

2.3.2 Il design

Dopo aver analizzato diversi siti, si è optato per la realizzazione di una struttura HTML che fosse tanto simile a quella dei siti di tendenza dell'ultimo anno. Il motivo principale è quello di permettere agli utenti un facile riconoscimento delle varie parti che compongono la web application, come per esempio l'utilizzo di un hamburger che raccoglie tutte le voci del menù. Importante è stata invece la scelta dello stile di Personalytics poiché l'idea di base è stata quella di realizzare una struttura web minimalista.

Il minimalismo, conosciuto anche come *minimal art*, è una tendenza artistica sviluppatasi nel corso degli anni sessanta caratterizzata da un processo di riduzione della realtà, dall'antiespressività, dall'impersonalità, dalla freddezza emozionale, dall'enfasi sull'oggettualità e fisicità dell'opera, dalla riduzione alle strutture elementari geometriche [8].

Uno dei principali aspetti del minimalismo in ambito del Word Wide Web è quello di ridurre il design ai soli elementi essenziali. Nel mettere in atto tale principio è stato utile fare alcune considerazioni su tre requisiti essenziali:

1. Oggetto: individuare qual è lo scopo del sito;
2. Usabilità: riguarda l'esperienza dell'utente con il sito;
3. Equilibrio: verificare che vi sia una perfetta gerarchia visiva.

Una volta fatte le precedenti considerazioni, si è passati a quella che è la scelta del colore per il sito. Quest'ultima è una conseguenza di diversi fattori, il più importante è sicuramente legato a quella che è la funzione del sito. I colori scelti per la realizzazione estetica del sito sono quattro:

- Bianco utilizzato per lo sfondo del sito.
- Bordeaux utilizzato per il logo, per l'animazione e alcuni blocchi.
- Grigio utilizzato per il logo, per i titoli e per il menù.
- Nero utilizzato per il logo e il testo.

Qui di seguito è riportata la paletta di colori, con il relativo codice in esadecimale, utilizzata per realizzare la grafica di Personalytics.

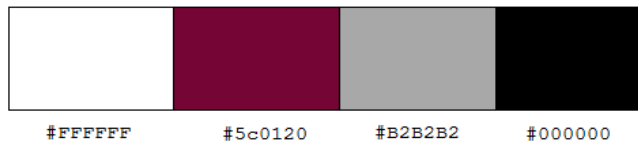


Figura 7. Paletta dei colori

Di fondamentale importanza è stata la scelta del font. Per font s'intende l'insieme dei caratteri tipografici caratterizzati e accomunati da un certo stile grafico.

Nella realizzazione dei contenuti di Personalytics, la scelta del font è stata oggetto di svariate analisi per far sì che i contenuti testuali si armonizzassero in modo aggraziato con la grafica delle pagina della web application.

Per la scelta del font di Personalytics sono state fatte diverse considerazioni sui seguenti fattori: leggibilità, dimensione e fascino.

La leggibilità è uno degli aspetti più importanti per quanto riguarda la scelta di un font. Un carattere tipografico poco leggibile non permette la completa navigazione all'utente. La leggibilità è legata al fattore dimensione: scegliere la dimensione adatta favorisce la lettura dei contenuti. Ultimo fattore è il fascino poiché l'utilizzo di un certo font piuttosto che un altro influenza, e non di poco, la qualità estetica di un progetto web.

Il font utilizzato per la parte contenutistica di Personalytics è Roboto (vedi Figura 7) messo a disposizione da Google Fonts¹⁶. Tale font è caratterizzato da forme ampiamente geometriche, ma allo stesso tempo il carattere presenta curve amichevoli che favoriscono un ritmo più naturale di lettura.

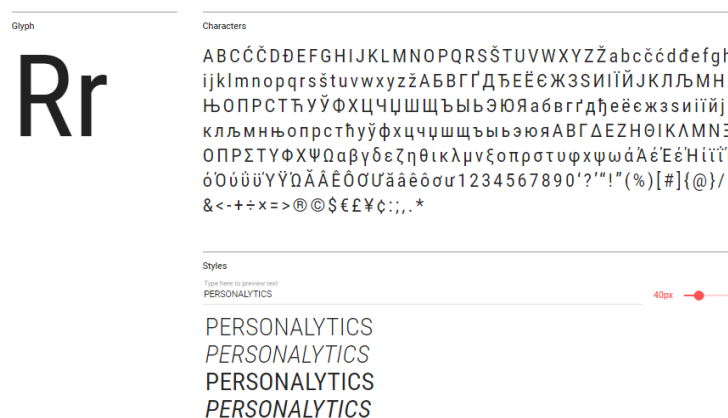


Figura 8. Google Fonts, Roboto

¹⁶ www.fonts.google.com

3. I dati come infografici

Dopo aver avviato la connessione con il DBMS PostgreSQL, si è passati all'analisi dei tipi di dati da rappresentare. Una volta effettuata la scelta di questi ultimi, Unicoop è stato interrogato attraverso il linguaggio di programmazione PHP. Terminata ogni singola interrogazione si è passati allo studio di quale infografico potesse rappresentare al meglio il tipo di dato scelto, per poi concludere con l'effettiva implementazione dei grafici utilizzando le librerie JavaScript.

3.1 PHP

PHP¹⁷ (*PHP: Hypertext Preprocessor*) è un linguaggio di programmazione “general-purpose” che viene utilizzato in maggioranza per la realizzazione di siti web dinamici in quanto è generatore di HTML. Si tratta di un linguaggio interpretato, non compilato al quale possono essere aggiunte estensioni per eseguire diverse funzionalità come per esempio sicurezza/crittografia, interazione con altri siti web, etc.

PHP è onnipresente ed è compatibile con la maggior parte dei sistemi operativi come Linux, molte varianti di Unix (compresi HP-UX, Solaris e OpenBSD), Microsoft Windows, MacOS X, MacOS Xserver, RISC OS, e probabilmente altri. Inoltre supporta anche la maggior parte dei server web esistenti, come per esempio Apache (vedi paragrafo 3.2). Tra le caratteristiche più salienti di PHP è la possibilità di supportare una completa gamma di database. Scrivere una pagina web collegata ad un database è incredibilmente semplice utilizzando una delle specifiche estensioni del database (*pg* per PostgreSQL) [10].

Qui di seguito è riportato un estratto del codice utilizzato per avviare la connessione tra PHP e il database Unicoop in PostgreSQL.

```
...
$conn="host='$host' port='$port' dbname='$db' user='$user' password='$psw'";
$conn1=pg_connect($conn) or die ("Errore di connessione. ".pg_last_error());
echo "Connessione avvenuta con successo";
pg_close($conn1);
...
```

¹⁷ www.php.net

3.2 Apache http server

Apache¹⁸ http server, più comunemente conosciuto come Apache, è un server web sviluppato nel corso del 1995 dall'Apache Software Foundation (ASF), una fondazione no-profit che si occupa dello sviluppo di progetti software.

Apache ha il compito di tradurre le richieste dei browser in pagine web e comprendere come elaborare il codice PHP, poiché quest'ultimo senza un server web non permetterebbe agli utenti web di raggiungere i documenti che contengono il suo codice. Apache non è l'unico server web esistente, ma a differenza di altri presenta il vantaggio di essere gratuito e di fornire tutto il codice esistente. Inoltre presenta la caratteristica di essere eseguibile su sistemi operativi differenti.

3.2.1 XAMPP

Fondamentale per lo sviluppo di Personalytics è stato XAMPP¹⁹. Quest'ultimo è una distribuzione Apache multi piattaforma, ciò significa che funziona su ambienti Linux, Mac e Windows. XAMPP è un pacchetto costituito da diverse componenti:

- Apache che è web server application che elabora e risponde le richieste restituendo i contenuti verso il computer richiedente (client);
- MySQL in quanto ogni applicazione web si appoggia a un database per memorizzare i dati. Esso è il più popolare tra i DMBS ed è Open Source e gratuito;
- PHP come linguaggio di script lato server per realizzare siti di ogni genere;
- altre componenti che non sono state trattate per la realizzazione del progetto.

¹⁸ www.apache.org

¹⁹ www.apachefriends.org

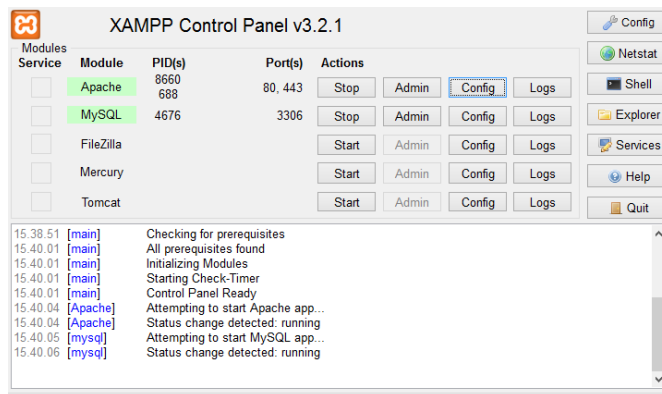


Figura 9. XAMPP, Pannello di controllo

È importante dire che prima di procedere con le interrogazioni al database Unicoop, è stato necessario integrare il DBMS PostgreSQL con XAMPP.

A tal proposito è lecito mostrare tre passaggi che hanno permesso tale integrazione:

1. Dal pannello di controllo XAMPP (vedi Figura 9) è stata selezionata la voce *config* di Apache che ha permesso di aprire il file *php.ini*.
2. Dal file *php.ini* sono stati eliminati i commenti alle righe:

```
extension=php_pdo_pgsql.dll
extension=php_pgsql.dll
```

3. Alla fine del file *php.ini* alla voce *httpd.conf* è stata aggiunta la seguente riga che completa la configurazione tra DBMS e server web.

```
LoadFile "C:\xampp\php\libpq.dll"
```

3.3 I dati in JSON

JSON (*JavaScript Object Notation*) è un formato di testo per lo scambio di dati che si adatta a diversi tipi di linguaggi come Python, C++, Perl, Javascript, PHP, etc.

Il formato JSON viene costruito su due strutture:

- Una raccolta di coppie nome/valore realizzata dai diversi linguaggi di programmazione come un oggetto, un dizionario, una tabella hash o come un array associativo.
- Un elenco di valori che nella maggior parte dei linguaggi si realizza con un array, una sequenza o un elenco [3].

Per la realizzazione del progetto si è scelto di costruire il JSON, tramite PHP, come una raccolta di coppie nome/valore attraverso un array associativo.

Qui di seguito è riportato un estratto del codice PHP, che dopo aver interrogato il database memorizza i valori in un array associativo restituendolo, grazie alla funzione `json_encode`, sottoforma di rappresentazione JSON (vedi Figura 10).

```
<?php
...
    $records3 = array(
        array(
            "label" => "BIO",
            "value" => $bio
        ),
        ...
        array(
            "label" => "DOCG",
            "value" => $docg
        )
    );
echo json_encode($records3);
pg_close($conn1);
?>
```

```
[
  - {
    label: "BIO",
    value: 46
  },
  - {
    label: "CELIACI",
    value: 19
  },
]
```

Figura 10. Formato JSON

3.4 Data Visualization

La vista è, per la maggior parte degli esseri umani il senso più importante, ed è il senso più importante anche per chi progetta infografiche. Si parla a tal proposito di Data Visualization. Con questo termine s'intende una disciplina della statistica descrittiva, con i dati mostrati sotto forma di immagini, grafici e schemi.

La visualizzazione dei dati viene sfruttata per catturare l'attenzione dell'utente mediante rappresentazioni grafiche, che fungono a supporto della presentazione dei dati di un database, senza dilungarsi troppo nel documento scritto che potrebbero distogliere l'attenzione dell'utente.

In ambito della Data Visualization è fondamentale il modo in cui i dati vengano visualizzati, per esempio i grafici a torta risultano essere utili per analizzare sondaggi, statistiche e per la distribuzione del denaro, ma allo stesso è sconsigliabile il loro uso per confrontare dei valori tra diverse categorie. Nella creazione di un grafico è importante:

- scegliere quale grafico può rappresentare in maniera più adeguata il tipo di dato;
- individuare il target di utenti a cui il grafico è destinato;
- tenere conto dei colori e dei contrasti per garantire l'accesso al grafico anche agli utenti ipovedenti.

3.4.1 D3.JS

D3.JS²⁰ (*Data Driven Document*) è una libreria scritta in Javascript utilizzabile per la visualizzazione interattiva dei dati e totalmente integrabile all'interno di pagine Web. Tale libreria è stata realizzata da Michael Bostock, Jeffrey Heer e da Vadim Ogievetsky nel 2011, e nasce come evoluzione del precedente progetto Protovis sviluppato nell'università di Stanford.

D3.JS è molto usata nell'ambito Data Visualization:

- Si tratta di una libreria scritta in Javascript. Quest'ultimo è un linguaggio di programmazione orientato agli oggetti e agli eventi diffuso sia a lato client che a lato server.
- E' compatibile con gli standard più diffusi nel web: SVG, HTML5, etc.

²⁰ www.d3js.org

- Consente di creare visualizzazioni sia statiche che dinamiche.
- Consente di implementare dati strutturati in differenti formati quali per esempio CSV e JSON.
- È una libreria notevolmente documentata con una vasta gamma di tutoria ed esempi da cui attingere.
- Funziona sui browser più recenti che includono il formato SVG, quindi Chrome, Safari, Firefox, ma allo stesso tempo può presentare delle problematiche relative al fattore compatibilità con Internet Explorer.

3.4.2 Raphaël.JS e Morris.JS

Raphaël²¹ è una piccola libreria JavaScript che semplifica il lavoro con la grafica vettoriale sul web. Tale libreria utilizza SVG come base per la creazione grafica. L'obiettivo di questa libreria è quello di fornire un adattatore che renderà il disegno vettoriale compatibile con la maggior parte dei browser. Raphaël attualmente è supportata da Firefox, Safari, Chrome, Opera e le versioni più recenti di Internet Explorer. Morris²² è una libreria leggera basata su jQuery e Raphaël, che fornisce semplicissimi grafici a linea, ad area, a bar e grafici ad anello.

3.5 I dati da visualizzare

Per lo sviluppo di Personalytics di fondamentale importanza sono state le tabelle Cliente, Venduto, Pagamento, Negozio, Articolo e Marketing. Attraverso queste è stato possibile realizzare le diverse tipologie di grafici all'interno delle pagine della web application. I grafici presenti in Personalytics sono stati realizzati con D3.JS e Morris.JS.

²¹ www.dmitrybaranovskiy.github.io

²² www.morrisjs.github.io

3.5.1 I grafici di Personalytics

In Personalytics sono state utilizzate diverse tipologie di grafici quali torte, ciambelle, barre, linee e un radar chart.

I grafici a torta (vedi Figura 11) utilizzati nella web app sono due e sono stati realizzati con D3.JS. Ambedue i grafici assumono in maniera simile le stesse caratteristiche: mostrano la distribuzione in percentuale della somma totale dell'importo speso dal cliente suddiviso per settore di marketing. Fondamentale è stato lo studio per il colore del grafico: ad ogni etichetta del grafico corrisponde una specifica fetta colorata secondo specifiche linee guida che rimandano a quella che è la psicologia dei colori del marketing [9].

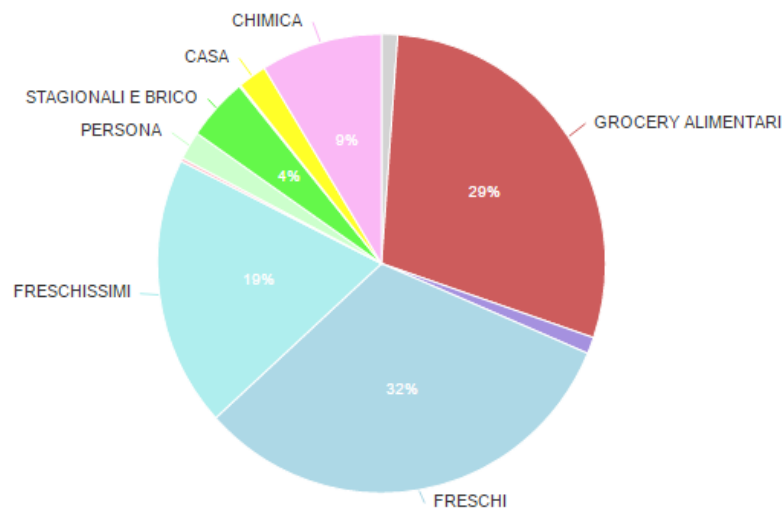


Figura 11. Torta dello storico del cliente

Prima di procedere con la realizzazione dell'infografico a torta, il database Unicoop presente in PostgreSQL è stato interrogato mediante il linguaggio di programmazione PHP.

Qui di seguito un estratto del codice PHP che ha permesso l'estrazione dei dati dal database Unicoop per la realizzazione del grafico a torta (vedi Figura 11).


```

<?php
session_start();

include("config.php");
$cliente = $_SESSION['id_cliente'];

$sql = "SELECT settore, sum(importo) AS Totale_per_settore FROM venduto JOIN
articolo ON venduto.id_articolo = articolo.id_articolo JOIN marketing on
articolo.id_marketing = marketing.id_mkt WHERE ID_CLIENTE = ".$cliente."
GROUP BY (settore)" ;

$result = pg_query($conn1, $sql);
$records3 = array();

while($row = pg_fetch_row($result)){
    $settore=$row[0];
    $totale= ($row[1]);
    ... ;
    array_push($records3, array("label" => $settore, "value" =>$totale));
}

echo json_encode($records3);
pg_close($conn1);
?>

```

Per la realizzazione del grafico a torta (vedi Figura 11), come detto precedentemente, è stata utilizzata la libreria D3.JS scritta in Javascript. Fondamentale per la creazione è stato l'utilizzo della chiamata `jQuery.ajax()`²³. Quest'ultima è una tecnica di sviluppo web che si basa sullo scambio di dati in background fra web browser e server e che consente l'aggiornamento dinamico di una pagina web. La sintassi di una chiamata AJAX è la seguente:

```
$.ajax({ name:value, name:value, ... })
```

Una chiamata AJAX è costituita da diverse componenti che creano le diverse coppie nome/valore. Per la creazione della torta sono state utilizzate le seguenti:

url: indica il percorso per inviare la richiesta;

method: specifica il tipo di richiesta (GET o POST);

data type: tipo di dato previsto della risposta del server;

²³ www.api.jquery.com

success: funzione da eseguire quando la richiesta ha esito positivo [3].

Qui di seguito è riportato un estratto del codice JavaScript utilizzato, incluso il funzionamento della chiamata AJAX, per la realizzazione della torta (vedi Figura 11)

```
function torta() {
    $.ajax({
        url: "php/totale.php",
        method: "POST",
        dataType: "JSON",
        success: function(result) {
            var j=0;
            var data = [];
            for (i in result) {
                ...
            }
            "labels": {
                ...
                "truncation": {
                    "enabled": true
                }
                "tooltips": {
                    "enabled": true,
                    "type": "placeholder",
                    "string": "{label}: {value} \u20AC"
                },
                "callbacks": {},
                "resize": true,
            });
        }
    });
};
```

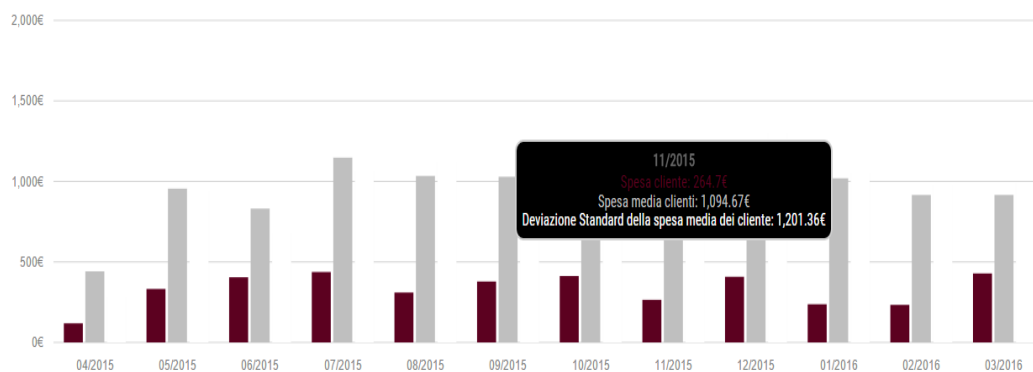


Figura 12. Bar Chart

Il grafico a barre (vedi Figura 12) mostra l'andamento annuale dell'importo mensile speso dal cliente confrontato con la media della spesa mensile di tutti gli altri soci Unicoop. Inoltre è stata calcolata per questi ultimi la deviazione standard.

La deviazione standard, chiamata anche scarto quadratico, è un indice di dispersione statistico, vale a dire una stima della variabilità di una popolazione di dati o di una variabile casuale. Essa è una misura di distanza dalla media e quindi ha sempre un valore positivo.

La deviazione Standard può essere calcolata sulla popolazione o sul campione: la popolazione è l'insieme degli elementi a cui si riferisce l'indagine statistica, il campione è un sottoinsieme di misurazioni selezionate dalla popolazione. [2]

Per la realizzazione del bar chart è stato effettuato il calcolo sulla popolazione degli utenti che fanno parte del database Unicoop.

Qui segue la formula per il calcolo della deviazione standard sulla popolazione:

$$\sigma = \sqrt{\frac{1}{(N-1)} \sum_{i=1}^N (x_i - \bar{x})^2}$$

N è il numero di osservazioni dell'intera popolazione, \bar{x} è la media della popolazione e x_i è lo i -esimo dato statistico osservato.

Qui di seguito un estratto del codice PHP che mostra sfrutta la funzione STDDEV di PostgreSQL per calcolare lo squarto quadratico.

```
<?php
...
$sql2 ="SELECT STDDEV(importo::numeric)as Media_DS,pagamento.data_p FROM
pagamento WHERE pagamento.id_cliente <> ".$cliente." AND
pagamento.data_p>(DATE '".$ultimadata.'" - Interval '12') GROUP BY
pagamento.data_p ORDER BY data_p ASC";
$result2 = pg_query($conn1, $sql2);
if(!$result2){
    echo pg_last_error($conn1);
    exit;
}
while($row2 = pg_fetch_row($result2)){
...
}
echo json_encode($records3);
pg_close($conn1);
php?>
```

Il bar chart (vedi Figura 12) è stato realizzato con la libreria Morris. Qui è riportato un estratto del codice JavaScript utilizzato per la realizzazione del grafico a barre.

```
...
Morris.Bar({
    element: 'bar_chart',
    data: data,
    xkey: 'y',
    ykeys: ['a','b','c'],
    postUnits: '€',
    labels: ['Spesa cliente', 'Spesa media clienti', 'Deviazione
Standard della spesa media dei cliente'],
    barColors:['#5c0120', '#bfbfbf', 'rgba(255,255,255,0)'],
    resize: true,
});
```

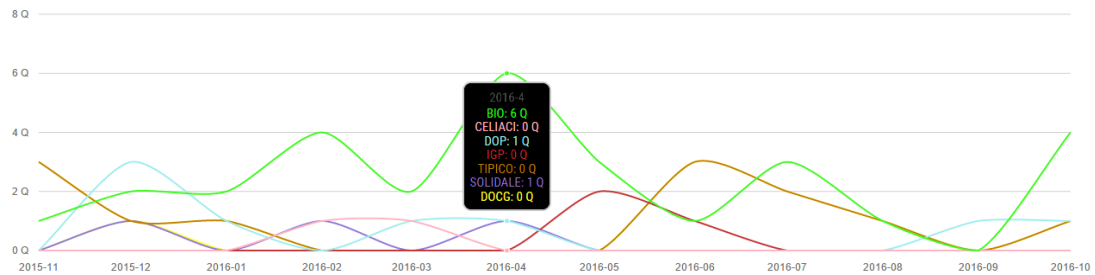


Figura 13, Grafico a linee

Il grafico a linee (vedi Figura 13) mostra la tendenza annuale suddivisa per mesi del consumatore. Per tendenza s'intende la sua abitudine nell'acquisto di prodotti che possono essere:

- 1) Biologici (BIO)
- 2) Gluten Free (Celiaci)
- 3) Indicazione Geografica Protetta (IGP)
- 4) Origine controllata e garantita (DOCG)
- 5) Origine protetta (DOP)
- 6) Locali/Tipici (TIPICO)
- 7) Solidali (SOLIDALE)

Qui di seguito un estratto del codice PHP che ha permesso l'estrazione dei dati dal database Unicoop.

```
<?php

$sql = "SELECT articolo.fl_coop, articolo.fl_bio, articolo.fl_celiaci,
articolo.fl_dop, articolo.fl_igp, articolo.fl_tipico, articolo.fl_solidale,
articolo.fl_docg, venduto.data_v

FROM venduto JOIN articolo ON venduto.id_articolo = articolo.id_articolo

WHERE venduto.id_cliente = ".$cliente." AND venduto.data_v >(DATE
'".$ultimadata."' -INTERVAL '12 months')

ORDER BY data_v DESC";

?>
```

Il grafico a linee è stato realizzato con la libreria Morris.js seguendo lo stesso procedimento già utilizzato nel caso del grafico a barre (vedi Figura 12).

3.6 Entropia

“L’entropia nasce come misura del valore informativo di una classe di eventi mutuamente esclusivi. [...] Gli eventi rari contengono maggiore informazione rispetto agli eventi più probabili. [...] Quanto più piccola è la probabilità tanto più grande è dunque l’informazione”²⁴_[2].

Qui di seguito la formula per calcolare l’entropia:

$$H = - \sum p(x) \log p(x)$$

H è detta entropia e si misura in *bit*, corrisponde alla sommatoria della probabilità di x moltiplicata per il logaritmo in base 2 della stessa. Per ottenere i risultati con valori positivi la formula è preceduta dal segno $-$ (negativo).

Per definizione, l’entropia è sempre un numero compreso tra zero ed uno: quando il valore entropico è pari a 0 ciò significa che la portata informativa dell’elemento non ha rilevanza, quando invece il valore è pari ad uno, l’elemento ha un carico informativo rilevante.

In parole più semplici, si definisce entropia la quantità di informazione “trasportata” dall’ elemento di un insieme; allo scopo di questa tesi l’entropia è servita a calcolare la sistematicità del cliente, ovvero la frequenza con cui esso compie determinate azioni e quanto le azioni di quest’ultimo incidano sull’ambiente COOP.

Il calcolo dell’entropia si basa sulla probabilità degli elementi chiamati in causa, che altro non sono che il frutto dello studio delle frequenze degli stessi elementi. La

probabilità di un evento, oltre alla sua definizione classica $P(a) = \frac{a}{n}$ (a numero di casi favorevoli e n il numero di casi possibili), dispone anche di una definizione frequentista:

$$P = \lim_{n \rightarrow \infty} \frac{f}{n}$$

²⁴ A.Lenci et. al., *Testo e computer Elementi di linguistica computazionale*, Roma, Carocci, 2005. Cit. p.172

Grazie a questa proprietà, che definisce la probabilità come il limite per n che tende all'infinito della frequenza relativa, è stato possibile giungere alla conclusione che la sistematicità del cliente alluda alle frequenze delle sue azioni che si ripercuotono sull'utilizzo del sistema COOP.

3.6.1 L'Entropia per misurare la sistematicità del cliente

In Personalytics di fondamentale importanza è stata l'entropia, che come illustrato nel paragrafo precedente, è stata utilizzata per misurare la sistematicità di ciascun cliente. Con il termine sistematicità si allude alla frequenza con cui il cliente compie determinate azioni che rientrano nel suo abituale utilizzo del negozio COOP. Il calcolo entropico per misurare la sistematicità del cliente verte essenzialmente su cinque variabili:

1. Orario (O)
2. Giorno della settimana (G)
3. Importo (I)
4. Negozio (N)
5. Quantità di articolo presenti nel carrello (A)

Per alcune variabili, orario, importo e quantità di articoli, si è scelto di suddividere in valori in classi, ovvero un raggruppamento in fasce per esemplificare al meglio l'insieme dei valori [7].

Per l'orario si è scelto di creare tre fasce orarie:

MATTINA	Orario < 13:00
POMERIGGIO	Orario \geq 14:00 e Orario \leq 16:00
SERA	Orario \geq 17:00

Per l'importo sono stati individuati cinque intervalli:

Importo \leq 10,00 €
Importo > 10,00 € e Importo \leq 25,00 €
Importo > 25,00 € e Importo \leq 50,00 €
Importo > 50,00 € e Importo \leq 100,00 €
Importo > 100,00 €

Per la quantità di articoli presenti nel carrello sono state individuate quattro fasce:

Qnt \leq 10
Qnt > 10 e Qnt \leq 20
Qnt > 20 e Qnt \leq 50
Qnt \geq 50

I valori ottenuti dal calcolo entropico di ogni variabile sono stati rappresentati mediante un grafico radar a cinque punte realizzato con D3.JS (vedi Figura 13).

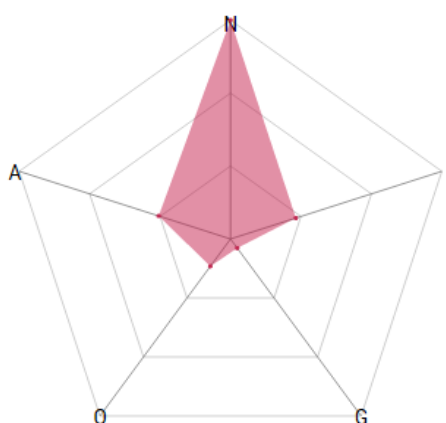


Figura 14. Radar chart

Per facilitare la comprensione del radar chart è necessario spiegare alcune sue caratteristiche: al centro troviamo il valore minimo dell'entropia (0), mentre ai cinque apici è posto il valore massimo (1). Fatta questa premessa, il grafico (vedi Figura 14) mostra che il cliente frequenta lo stesso negozio, ma non è prevedibile per

quanto riguarda il giorno della settimana, la quantità di articoli presenti nel suo carrello, l'importo e l'orario.

Qui un estratto del codice PHP utilizzato per calcolare l'entropia sull'orario contenente la suddivisione di quest'ultimo in fasce:

```
//ENTROPIA FASCE ORARIE
$sql4 = "SELECT EXTRACT(HOUR FROM orario) as ora, COUNT (*) AS numero
        FROM venduto
        WHERE id_cliente = ".$cliente."
        GROUP BY ora";
$result4 = pg_query($conn1, $sql4);
$somma_h=0;
$k=0;
$mattina=0;
$sera=0;
$pomeriggio=0;
$countmat=0;
$countpom=0;
$countser=0;
while($row = pg_fetch_row($result4)){
    if($row[0]<13){
        $mattina++;
        $countmat=$countmat+$row[1];}
    else if($row[0]>=13 and $row[0]<=16){
        $pomeriggio++;
        $countpom=$countpom + $row[1];}
    else if($row[0]>=17){
        $sera++;
        $countser = $countser + $row[1];}}

$somma_h = $countmat + $countpom + $countser;
$entropia_h = $entropia_h + (($countmat/$somma_h) *
(log(($countmat/$somma_h),2))) + (($countpom/$somma_h) *
(log(($countpom/$somma_h),2))) + (($countser/$somma_h) *
(log(($countser/$somma_h),2)));
$entropia_h = - ($entropia_h);
$entropia_h = number_format(($entropia_h / log(3, 2)),2);
...
```

4. La mobilità

Sfruttando le API di Google Maps, sono state estrapolate le coordinate di ciascun socio COOP e inserite come marcatori all'interno della mappa e poi clusterizzati. Successivamente è stato calcolato il centro di massa e il raggio d'inerzia. Inoltre, è stato utilizzato un applicativo Java che ha permesso la ricostruzione delle traiettorie di ogni singolo cliente.

4.1 Google Maps

Google Maps²⁵ è un servizio, messo a disposizione da Google, accessibile dal relativo sito web che consente la ricerca e la visualizzazione di mappe geografiche di buona parte della terra. Si tratta di un servizio gratuito attraverso il quale è possibile:

- spostarsi fra e sulle mappe in maniera semplice e intuitiva;
- scegliere la modalità di visualizzazione del terreno (satellitare, earth, etc.);
- localizzare luoghi peculiari, esempio monumenti, ristoranti, alberghi, etc.;
- visualizzare foto satellitari o immagini fotografiche nelle quali si possono distinguere strade, case, giardini, etc;
- ottenere il percorso stradale tra due punti a scelta fornendo tempo stimato e distanza;
- utilizzare il servizio per implementare mappe nei propri siti web.

Per inserire una mappa all'interno delle proprie pagine web è necessario inserire un comando che si collega all' API di Google Maps.

L'API, acronimo di *Application Programming Interface*, indica un insieme di procedure disponibili al programmatore per lo svolgimento di un dato compito all'interno di una certa procedura.

²⁵ www.maps.google.it

4.1.1 Personalytics e Google Maps

Personalytics, come già detto nei capitoli precedenti, oltre a tenere traccia dei dati di consumo del cliente, tiene conto anche della sua mobilità e nel farlo sfrutta Google Maps.

Per inserire la mappa all'interno della web application è stato necessario creare il collegamento con l'API di Google Maps che è stato inserito all'interno della pagina "mobilità.html".

```
<script src="https://maps.googleapis.com/maps/api/js?key=***"async defer>
</script>
```

È stato fondamentale, ai fini del collegamento, richiedere una chiave di autenticazione (API keys) che è stata inserita dopo la clausola *key* all'interno del link.

Dopo aver effettuato il collegamento è stato creato un div (cioè un blocco) nella stessa pagina all'interno del quale verrà posizionata la mappa.

```
<div id="map_canvas"></div>
```

Lo step successivo è stato quello di creare un file JavaScript (maps.js) in cui è stato necessario effettuare due passaggi essenziali:

1. segnalare le coordinate del centro della mappa espresse in latitudine e longitudine;
2. scegliere il livello di zoom con il quale mostrare la mappa. Tale livello va da 0 a 21.

Qui di seguito un estratto del codice JavaScript utilizzato per realizzare i due passaggi sopra citati.

```
function initialize() {
    map = new google.maps.Map(document.getElementById('map_canvas'), {
        zoom: 20,
        center: {lat: 43.72333, lng: 10.40167}
    });
}
(...)
```

Dopo aver collegato il precedente file maps.js alla pagina "Mobilità" si è passati a definire le dimensione della mappa mediante i fogli di stile CSS.

La seguente immagine mostra la prima apparizione della mappa nella pagina “mobilità.html”

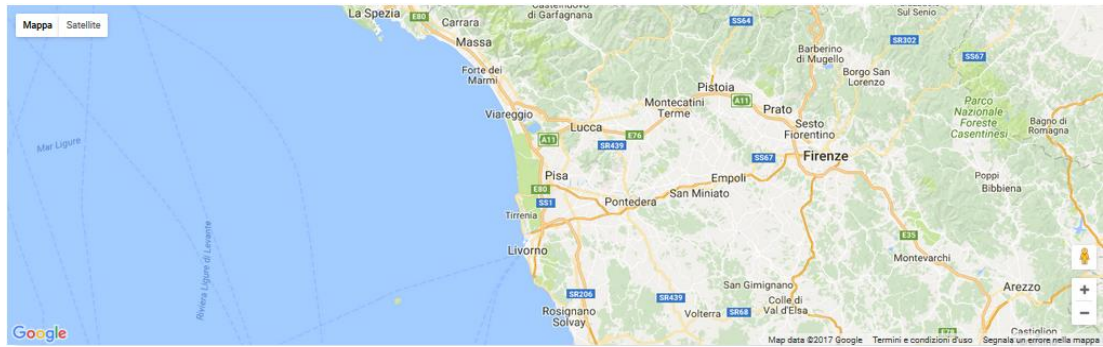


Figura 15. Mappa

4.1.2 L'inserimento dei marker

Dopo aver implementato Google Maps in Personalytic la fase successiva riguarda l'inserimento dei marker relativi agli spostamenti GPS del cliente presenti nella tabella "Location_Point" (vedi Tabella 4).

Location_Point

ATTRIBUTO	FUNZIONE	TIPO DI DATO	DESCRIZIONE
Id_location	PrimaryKey	Incremental	Incrementale progressivo e chiave primaria della tabella
Id_device	ForeignKey	Bigint	Chiave esterna che punta alla tabella Device
Id_cliente	ForeignKey	Bigint	Chiave esterna che punta alla tabella Cliente
Latitudine		Decimal	Latitudine
Longitudine		Decimal	Logitudine
Speed		Decimal	Velocità
Dwell_time		Decimal	Tempo di fermata in un punto
Provider		Character Varying	Fornitore del servizio di localizzazione (GPS, network, etc.)
BSSID		Character Varying	Indirizzo MAC
Weekday		Integer	Valore compreso tra 0 e 6 che corrisponde al giorno della settimana: 0 a Lunedì, 1 a Martedì, ..., 6 a Domenica.
Time_stamp		Numeric	Data e ora in formato Unix

Tabella 4. Location_Point

Gli spostamenti del singolo cliente dalla tabella sono stati estrapolati mediante un'interrogazione PHP. I dati estrapolati sono stati successivamente inseriti all'interno di un array associativo in JSON costituito da tante coordinate quante sono gli spostamenti del cliente. Qui di seguito un estratto di interrogazione PHP.

```
$sql = "SELECT DISTINCT latitudine, longitudine
      FROM location_point
      WHERE ID_CLIENTE = ".$cliente." ;
```

I dati estrapolati sono stati successivamente implementati nello stesso file JavaScript in cui è stata inizializzata la mappa mediante una chiamata AJAX. Qui di seguito un estratto del codice JavaScript utilizzato per l'inserimento dei marker.

```
(...)
var mcOptions = {
  gridSize: 50,
  maxZoom: 15,
  imagePath: '../img/m'
};

var lista = result;
var markers = [];
for (var i in lista) {
  var myLatLng = {lat: lista[i]['lat'], lng: lista[i]['long']};
  var marker = new google.maps.Marker({position: myLatLng});
  markers.push(marker);
}
(...)
```

La seguente immagine mostra la mappa di Google popolata con i marker relativi alle posizioni GPS del cliente.

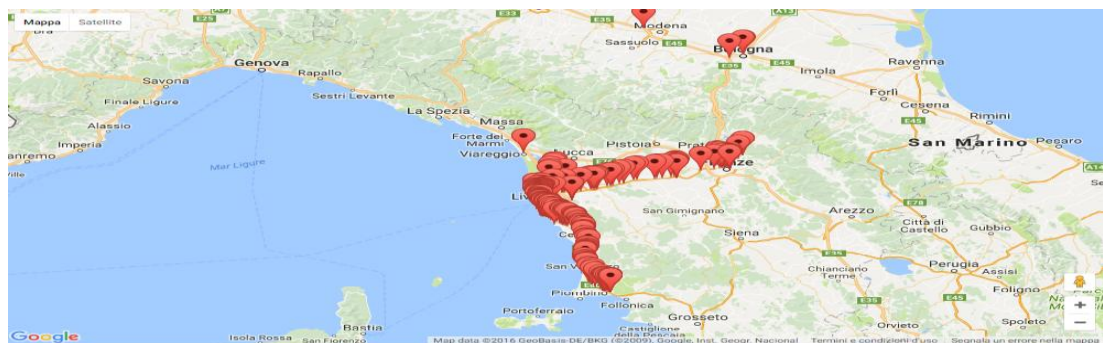


Figura 16. Mappa con i marker relativi alle coordinate del cliente

4.1.3 La clusterizzazione dei marker

Prima di esemplificare il modo in cui i marker sono stati clusterizzati è lecito fare chiarezza su cosa sia il clustering.

Il clustering innanzitutto è una tecnica di Data Mining. Quest'ultimo è l'insieme delle metodologie e delle tecniche che hanno per oggetto l'estrazione di un sapere o di una conoscenza a partire da grandi quantità di dati [6].

Il clustering, come detto precedentemente, è una tecnica che consiste nel collocare oggetti che possono essere prodotti, segmenti di mercato, locazione geografiche, etc., all'interno di raggruppamenti omogenei, ovvero dello stesso tipo, chiamati *cluster*, in base a dei criteri di valutazione delle similarità fra tali oggetti.

Il servizio Google Maps permette di usufruire di una libreria "MarkerClusterer" che permette la clusterizzazione dei marcatori posti in prossimità con lo scopo di semplificare la visualizzazione di questi ultimi all'interno della mappa. Tale libreria è stata utilizzata per clusterizzare le coordinate del cliente e per rendere leggibile la quantità dei suoi spostamenti.

Il primo passaggio per eseguire il clustering è stato quello di collegare la libreria JavaScript alla pagina "mobilità.html" inserendo il link all'interno di un tag script.

```
<script https://developers.google.com/maps/documentation/javascript/exam  
ples/markerclusterer/markerclusterer.js/>
```

Successivamente si è passati all'implementazione del clustering all'interno del precedente file maps.js. Qui di seguito un estratto del codice JavaScript che realizza la clusterizzazione.

```
(...)  
var markerCluster = new MarkerClusterer(map, markers,  
{ imagePath: 'https://developers.google.com/maps/documentation/javascript/exam  
ples/markerclusterer/m' });  
(...)
```

La seguente figura mostra il modo in cui i marcatori sono stati clusterizzati. Ogni cluster è contrassegnato da un numero che indica il numero di marker presenti in quella determinata zona.



Figura 17. Mappa con clustering dei marker

4.1.4 Il centro di massa

Il centro di massa, chiamato anche baricentro, è il punto geometrico corrispondente al valore medio della distribuzione della massa del sistema nello spazio.

Si definisce centro di massa di un sistema (discreto) di N punti materiali il punto geometrico le cui coordinate, in un dato sistema di riferimento, sono date da:

$$\mathbf{R} = \frac{m_1 \mathbf{r}_1 + m_2 \mathbf{r}_2 + \dots + m_N \mathbf{r}_N}{M} = \frac{\sum_{i=1}^N m_i \mathbf{r}_i}{\sum_{i=1}^N m_i}$$

dove $M = m_1 + m_2 + \dots + m_N$ è la massa totale del sistema e le quantità r_i sono i raggi vettori dei punti materiali rispetto al sistema di riferimento usato.

Assumendo che tutti i punti materiali, di generiche coordinate (x_i, y_i) , abbiano la stessa massa ($m_i = m$ per $i=1, \dots, N$), il centro di massa delle coordinate (x_{cm}, y_{cm}) è dato dalle formule:

$$x_{cm} = \frac{\sum x_i}{N} \quad y_{cm} = \frac{\sum y_i}{N}$$

Attraverso l'applicazione di queste ultime due formule è stato possibile calcolare il centro di massa rispetto alle coordinate del cliente. Qui di seguito un estratto del codice PHP che ha permesso il calcolo del baricentro.

```
sql = "SELECT DISTINCT AVG(replace(latitudine,',','.')::numeric) as cmx,
      AVG(replace(longitudine,',','.')::numeric) as cmy

      FROM location_point

      WHERE id_cliente=".$cliente."";
```

Dopo aver calcolato il centro di massa, le nuove coordinate x_{cm} e y_{cm} sono state inserite all'interno di un array associativo in formato JSON e successivamente richiamato all'interno del file maps.js mediante un chiamata AJAX al file PHP. Qui di seguito è riportato un estratto del codice JavaScript che mostra l'inserimento delle coordinate relative al baricentro nella mappa.

```
...
$.ajax({
  dataType: "JSON",
  url: "php/centro_medio.php",
  method: "POST",
  success: function(result) {
    var lista2 = result;
    var myLatLng = {lat: lista2[0]['lat'], lng: lista2[0]['long']};
    var marker = new google.maps.Marker({
      position: myLatLng,
      icon: "http://maps.google.com/.../yellow-dot.png"
    });
    marker.setMap(map);
  }
});
```

La seguente figura mostra il modo in cui viene inserito, all'interno della mappa, il centro di massa utilizzando un marker diverso.



Figura 18. Mappa con il posizionamento del centro di massa

4.2 Radius of gyration

Il raggio d'inerzia, in inglese *radius of gyration*, descrive come le componenti di un oggetto sono distribuite attorno al suo asse di rotazione. Può essere definito come la distanza da un polo alla quale dovrebbe trovarsi tutta la massa di un corpo rigido affinché il momento di inerzia resti invariato.

$$R_i^2 = \frac{I}{M}$$

Il raggio d'inerzia si calcola quindi come la radice quadrata del momento di inerzia (I) fratto la massa totale del corpo (M).

Il *radius of gyration* è stato calcolato in Personalytics per ciascun cliente come la sua distanza media dalla posizione preferita, o centro di massa [5]. Qui di seguito la formula utilizzata:

$$r_g = \sqrt{\frac{1}{n} \sum_{i \in L} (r_i - r_{cm})^2}$$

dove L è l'insieme dei luoghi visitati dal singolo cliente, r_i è un vettore a due dimensioni che descrive le coordinate geografiche della posizione i ; n è il numero totale di visite o tempo trascorso e RCM è il centro di massa del singolo individuo.

Qui è riportato un estratto del codice PHP con la relativa funzione utilizzata per calcolare del *radius of gyration*.

```
...
function haversineGreatCircleDistance(
    $latitudeFrom, $longitudeFrom, $latitudeTo, $longitudeTo, $earthRadius =
6371000)
{
    $latFrom = deg2rad($latitudeFrom);
    $lonFrom = deg2rad($longitudeFrom);
    $latTo = deg2rad($latitudeTo);
    $lonTo = deg2rad($longitudeTo);

    $latDelta = $latTo - $latFrom;
    $lonDelta = $lonTo - $lonFrom;

    $angle = 2 * asin(sqrt(pow(sin($latDelta / 2), 2) +
        cos($latFrom) * cos($latTo) * pow(sin($lonDelta / 2), 2)));
    return $angle * $earthRadius;
}
...
```

Il risultato ottenuto è stato successivamente convertito in metri e inserito all'interno di un blocco HTML. Qui di seguito un'immagine che mostra il raggio d'inerzia calcolato per un cliente x.

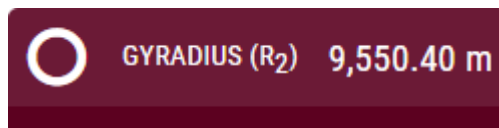


Figura 19. Radius of gyration del cliente analizzato. Quest'ultimo gravita in un'area che dista 10 km dal centro di massa.

4.3 Java

Java²⁶ è un linguaggio di programmazione sviluppato alla Stanford University agli inizi degli anni novanta da Sun Microsystems e altri esperti. Si tratta di un linguaggio orientato agli oggetti specificatamente progettato per essere il più possibile indipendente dalla piattaforma di esecuzione.

4.3.1 Trajectory Builder

Trajectory_Builder.jar [5] è un applicativo in Java sviluppato da alcuni membri del KDDLab. Tale programma permette, attraverso il setting di alcuni parametri che vengono inseriti all'interno di un file di testo (.txt) e successivamente richiamati all'interno del terminale con il file Java, la ricostruzione delle traiettorie GPS del cliente.

Tutte le osservazioni delle coordinate dello stesso cliente vengono concatenate in ordine temporale crescente in modo da ottenere una traiettoria globale unica. La ricostruzione della traiettoria avviene secondo alcuni criteri stabiliti dallo sviluppatore: si è scelto di suddividere la traiettoria globale in diverse sotto-traiettorie, corrispondenti a viaggi o spostamenti del cliente, utilizzando un intervallo di 30 minuti. Se l'intervallo di tempo tra due osservazioni consecutive è maggiore di 30 minuti, la prima osservazione è considerata come la fine di un viaggio e la seconda come l'inizio di un altro viaggio [5].

I parametri necessari sono:

- URL= Connessione url: jdbc:postgresql://<host>:<port>/<database_name>
- USER = Username
- PASSWORD = Password
- INPUT_TABLE = La tabella contenente le coordinate GPS prese in analisi.
Le informazioni richieste sono: <user_id>,<lat>,<lon>,<timestamp>
- USER_ID = Il nome della colonna contenente l'id del cliente <user_id>
- LON = Il nome della colonna contenente la longitudine <longitude>
- LAT = Il nome della colonna contenente la latitudine <latitude>
- TIME = Il nome della colonna contenente il timestap <timestamp>

²⁶ www.java.com

- OUTPUT_TABLE = Nome della tabella di output
- IGNORE_NOISE = Punti che devono essere ignorati
- IGNORE_ONE_POINT = Traiettorie composte da un unico punto che verranno rimosse
- MAX_TIME_GAP = Massimo tempo tra due punti (s)
- MAX_SPACE_GAP = Massimo spazio tra due punti (m)
- MAX_SPEED = Velocità massima (m/s)
- MIN_SPEED = Velocità minima (m/s)
- REMOVE_REDUNDANCY = Rimozione dei punti ridondanti considerando una dispanza massima
- MIN_SPACE_GAP = Minimo spazio tra due punti (m)

Dopo aver eseguito nel terminale il file Trajectory_builder.jar, con i relativi parametri settati, il programma si collega in automatico al DBMS PostgreSQL all'interno del quale verrà generata una tabella nel database di destinazione definito all'interno dei parametri.

4.3.2 Trajectory builder: la tabella Traiettoria

Nello corso dello sviluppo di Personalytics, grazie all'utilizzo del programma Trajectory_builder.jar è stato possibile generare una nuova tabella chiamata "Traiettoria" per distinguerla da "Location_point".

Essa nasce con lo scopo di fare alcuni calcoli specifici sugli spostamenti GPS del cliente, quali:

- i km totali percorsi;
- la Media dei km percorsi;
- la deviazione standard dei km percorsi rispetto alla media dei km percorsi dal cliente.

I parametri richiesti dall'applicativo Java sono stati presi dalla tabella "Location_Point".

Qui di seguito è riportato il file di testo (.txt) con il settaggio dei parametri utilizzato.

```

URL = jdbc:postgresql://localhost:5432/unicoop

USER = postgres

PASSWORD = ****

INPUT_TABLE = public.traiettoria

USER_ID = cane

LON = lon

LAT = lat

TIME = time_stamp

OUTPUT_TABLE = public.traiettoria

MAX_TIME_GAP = 1200

MAX_SPACE_GAP = 50

```

Dopo l'esecuzione del file Java, all'interno del database Unicoop è stata aggiunta la nuova tabella. Qui di seguito è riportata un'immagine che mostra la tabella traiettoria in PostgreSQL.

id text	object geometry	duration numeric	length numeric	uid bigint	start_date text
1	2 2	0.0	404.95503968429495	2	2015-04-03
2	3	0.0	178.3075337549504	2	2015-04-04
3	5	0.0	120.3010342986639	2	2015-04-06
4	6	0.0	95.6080474308261	2	2015-04-07
5	7	0.0	77.78179180028716	2	2015-04-08
6	3 177	0.0	99.9765996082333	3	2015-11-14
7	3 178	0.0	35.099161359057774	3	2015-11-16
8	2 8	0.0	606.3160362543798	2	2015-04-10
9	2 9	0.0	763.5250463803421	2	2015-04-11
10	2 10	0.0	860.3761042986654	2	2015-04-12
11	2 11	0.0	343.78698257745054	2	2015-04-13
12	2 12	0.0	467.81739945838217	2	2015-04-14
13	3 179	0.0	74.38804469716347	3	2015-11-17
14	2 13	0.0	342.2830770115602	2	2015-04-15
15	2 14	0.0	492.32152676517046	2	2015-04-16
16	2 15	0.0	632.7679990814308	2	2015-04-17
17	2 16	0.0	569.4237515748038	2	2015-04-18
18	2 17	0.0	1026.276536171888	2	2015-04-19
19	3 180	0.0	307.7978265540314	3	2015-11-18
20	2 18	0.0	507.5147604647537	2	2015-04-20

Figura 20. Unicoop, Traiettoria

5. Shopping list

Dopo aver effettuato un particolare studio sul modo in cui rappresentare i cento prodotti acquistati dal cliente con frequenza maggiore rispetto agli altri, si è giunti alla realizzazione di una shopping list dinamica realizzata attraverso la combinazione dell'HTML, del CSS e del linguaggio di programmazione JavaScript.

5.1 La frequenza come suggerimento

Di fondamentale importanza per la realizzazione della shopping list presente all'interno della pagina "spesa" (vedi capitolo 2, paragrafo 2.3, sezione 2.3.1) è stata la modalità con la quale si è determinata la frequenza relativa agli acquisti, da parte del cliente, dei prodotti all'interno della struttura COOP. Attraverso la frequenza è stato possibile determinare quali e quanti sono i prodotti che il socio acquista abitualmente.

Qui di seguito è riportato un estratto del codice PHP che ha permesso l'estrazione dei cento prodotti più acquistati dai clienti ordinati in maniera decrescente di frequenza.

```
$sql = "SELECT m.categoria, count(*) as occorrenza
      FROM articolo AS a JOIN venduto AS v on a.id_articolo=v.id_articolo
      JOIN marketing AS m on a.id_marketing=m.id_mkt
      WHERE id_cliente=".$cliente."
      GROUP BY m.categoria
      ORDER BY count(*) DESC
      LIMIT 100;"
```

Il risultato ottenuto dopo aver effettuato l'interrogazione al database è stata una lista contenente i cento prodotti più acquistati dal cliente ordinati in maniera decrescente rispetto al numero di acquisto. Qui di seguito un esempio:

LATTE	250
PASTA DI SEMOLA	190
...	...
BIRRA	20

La lista ottenuta corrisponde alla lista dei suggerimenti presenti all'interno della pagina "spesa". Tali suggerimenti, recuperati attraverso una chiamata AJAX, sono stati creati attraverso l'utilizzo del linguaggio di programmazione Javascript combinato con l'HTML.

```
$(document).ready(function () {  
  
    $.ajax({  
  
        url: "php/suggerimenti.php",  
  
        method: "POST",  
  
        dataType: "JSON",  
  
        success: function(result) {  
  
            var html=''  
  
            for (i in result) {  
  
                html += '<li id="prodotti">'+result[i]["Categoria"]+'</li>'  
  
                }  
  
                $("#lista_prod").html(html);  
  
            }  
  
        });  
  
    });  
});
```

Ogni suggerimento è stato inserito dal JavaScript all'interno di un lista HTML ``. Successivamente a quest'ultima sono stati applicati i fogli di stile CSS per quanto riguarda la resa grafica.



Figura 21. Lista dei suggerimenti

5.2 Da suggerimento a prodotto da acquistare

Attraverso l'utilizzo del linguaggio di programmazione Javascript è stato possibile creare una shopping list dinamica (vedi Figura 22). Quest'ultima è resa tale attraverso l'utilizzo dei metodi. Questi non sono altro che le attività che un oggetto può compiere. Alcuni dei metodi utilizzati per la realizzazione della lista della spesa sono:

- *addClass* permette di aggiungere una classe agli elementi selezionati;
- *remove* permette di rimuovere un elemento da un elenco in cascata;
- *append* permette di inserire un contenuto specificato alla fine degli elementi selezionati.

Il funzionamento della lista avviene mediante la selezione di un suggerimento che verrà eliminato dalla lista, ma verrà aggiunto in nuova lista che corrisponde alla lista della spesa del cliente. Qui di seguito un estratto del codice Javascript che mostra il modo in cui un suggerimento viene selezionato e eliminato dalla prima lista, e viene aggiunto nella seconda lista.

```
(...)  
  
if(event.target.localName === 'li'){  
    $this.addClass('remove');  
  
    setTimeout(function(){  
        $this.remove(); //e rimuovi l'elemento  
    }, 420);  
}  
});  
  
$list_old.bind('click', 'li', function(event){  
    var $this = $(event.target);  
  
    if(event.target.localName === 'li'){  
        $list.append('<li id="new_prod" class="ele">'+$($this).html()+'</li>');  
        $this.addClass('remove');  
    }  
});  
  
(...)
```

Qui di seguito è riportata un'immagine che mostra l'esecuzione della shopping list.

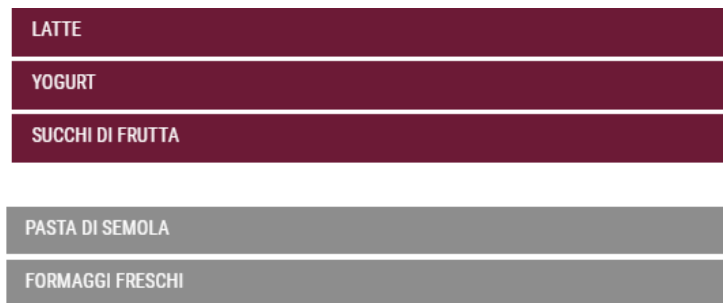


Figura 22. Lista della spesa che viene compilata mediante la selezione di un suggerimento.

6. Caso di studio

In questo capitolo vengono mostrati i dati relativi a tre diversi clienti che si comportano in modo diverso rispetto agli acquisti, alla sistematicità e alla loro mobilità.

6.1 Comportamento sulla base degli acquisti

Prendendo in esame tre clienti x, y, z è stato possibile evidenziare alcuni comportamenti in merito ai loro dati di consumo all'interno dell'ambiente COOP. Per tutti e tre i clienti sono state recuperate le informazioni riguardo lo storico dell'importo speso e come quest'ultimo si suddivide in percentuale a seconda della categoria di marketing.

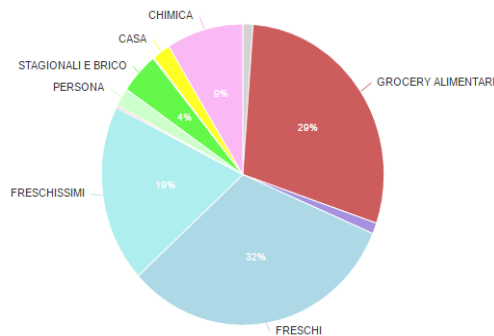


Figura 33. Torta cliente x

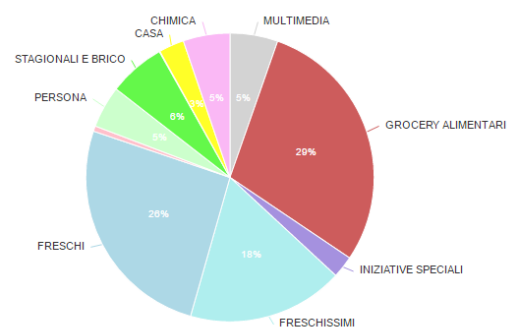


Figura 24. Torta del cliente y

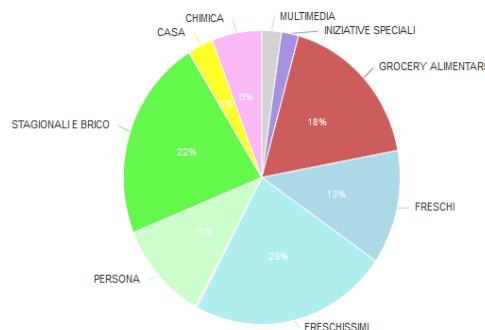


Figura 25. Torta del cliente z

Il cliente x ha speso un importo totale di € 11.267,35. Dal grafico (vedi Figura 23) è possibile dedurre che si tratta di un socio che acquista maggiormente prodotti che appartengono alla categoria di marketing freschi (32%), seguita dalla categoria grocery alimentari (29%) e freschissimi (19%). Acquista invece in minoranza prodotti appartenenti al settore salute, multimedia e iniziative speciali dove la percentuale risulta essere ≤ 2 .

Il cliente y ha speso un importo totale di € 14.033,99. Dal secondo grafico (vedi Figura 24) è possibile dedurre che si tratta di un socio che acquista maggiormente prodotti che appartengono alla categoria di marketing grocery alimentari (29%), seguita dalla categoria freschi (26%) e freschissimi (19%). Acquista invece in minoranza prodotti appartenenti al settore salute e iniziative speciali dove la percentuale risulta essere ≤ 2 .

Il cliente z ha speso un importo totale di € 3.774,39. Dal terzo grafico (vedi Figura 25) è possibile dedurre che si tratta di un socio che acquista maggiormente prodotti che appartengono alla categoria di marketing stagionali e brico (22%), seguita dalla categoria freschissimi (26%) e freschissimi (13%). Acquista invece in minoranza prodotti appartenenti al settore salute, iniziative speciali e multimedia dove la percentuale risulta essere ≤ 2 .

6.2 Comportamento in base alla sistematicità

Prendendo in analisi gli stessi clienti x, y, e z è stato possibile confrontare anche il loro comportamento in merito alla sistematicità calcolata, come detto precedentemente, attraverso l'entropia (vedi capitolo 3, paragrafo 3.6, sezione 3.6.1).

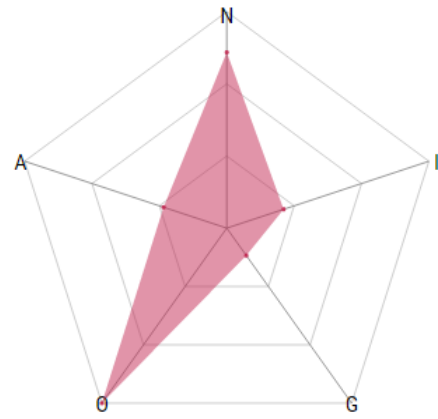
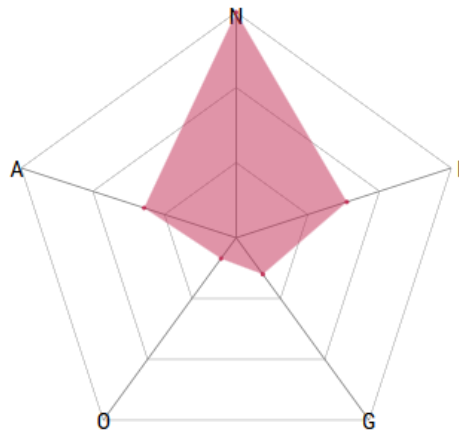


Figura 26. Sistematicità del cliente x Figura 27. Sistematicità del cliente y

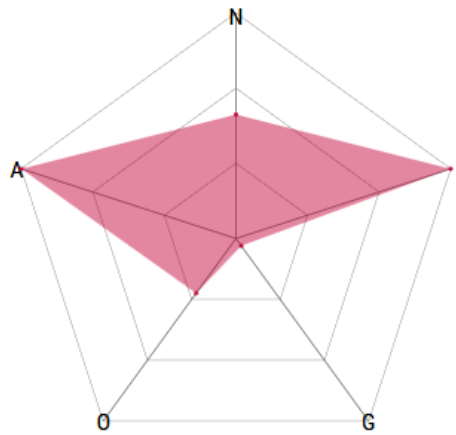


Figura 28. Sistematicità del cliente z

Il cliente x, secondo il grafico (vedi Figura 26) risulta essere sistematico per quanto riguarda il negozio. Quindi si tratta di un socio COOP che tendenzialmente frequenta la solita struttura. Risulta essere meno sistematico in merito alle altre quattro variabili quali l'importo, la quantità di articoli presenti nel carrello, l'orario e il giorno della settimana.

Il cliente y, secondo il grafico (vedi Figura 27) risulta essere sistematico per quanto riguarda l'orario e quasi prevedibile per quanto riguarda il negozio. Quindi si tratta di un socio COOP che tendenzialmente frequenta la stessa struttura nel solito orario. Risulta essere meno sistematico in merito all'importo speso, l'orario e la quantità di articoli presenti nel carrello.

Il cliente z, secondo il grafico (vedi Figura 28) risulta essere sistematico per quanto riguarda la quantità di articoli presenti nel carrello e per l'importo speso. Quindi si tratta di un socio COOP che tendenzialmente acquista la stessa quantità di articoli e spende quasi sempre lo stesso importo. Risulta essere meno sistematico in merito al giorno della settimana, l'orario e il negozio.

6.3 Comportamento in base alla mobilità

Confrontando i dati di mobilità degli stessi clienti x, y e z, è stato possibile mettere in risalto alcune caratteristiche che si differenziano tra di loro in merito al centro di massa (vedi capitolo 4, paragrafo 4.1, sezione 4.1.4), al raggio d'inerzia (vedi capitolo 4, paragrafo 4.2) e ai punti di localizzazione.

Ai fini di una maggiore comprensione del confronto tra le diverse mobilità dei clienti è opportuno tenere conto che maggiore è il *radius of gyration* e maggiore sarà la distanza tra il centro di massa e i punti localizzati.



Figura 29. Mobilità del cliente x



Figura 30. Mobilità del cliente y



Figura 31. Mobilità del cliente z

Il cliente x presenta un centro di massa con latitudine 43.914736591856 e longitudine 10.497166429288 (vedi Figura 29). Il raggio d'inerzia calcolato rispetto

al baricentro risulta essere 9,550.40 m. Il cliente gravita in un'area che dista circa 10 km rispetto al centro di massa.

Il cliente y presenta un centro di massa con latitudine 42.819887923997 e longitudine 10.916636673753 (vedi Figura 30). Il raggio d'inerzia, calcolato rispetto al baricentro, risulta essere 12,538.18 m. Il cliente gravita in un'area che dista circa 13 km rispetto al centro di massa

Il cliente z presenta un centro di massa con una latitudine di 43.154876972862 e una longitudine di 11.525547234802 (vedi Figura 31). Il raggio d'inerzia, calcolato rispetto al baricentro, risulta essere 14,308.57 m. Il cliente gravita in un'area che dista circa 14 km rispetto al centro di massa.

7. Conclusioni

L'obiettivo del mio lavoro è stato quello di creare una web application con lo scopo di aiutare ogni socio Unicoop a monitorare con estrema facilità i propri dati di consumo e di mobilità.

Durante la realizzazione di questo progetto sono state riscontrate delle difficoltà inerenti all'uso dei vari linguaggi di programmazione, tra cui l'utilizzo della libreria D3.JS per la visualizzazione e la rappresentazione dei dati sotto forma di grafici.

È fondamentale sottolineare che si tratta di un prototipo e che alcuni aspetti, come per esempio la resa grafica, verranno rivisti e, come tutte le cose, migliorati.

8. Bibliografia e Sitografia

Bibliografia

- [1] Paolo Atzeni, Stefano Ceri, Stefano Paraboschi, Riccardo Torlone “*Basi di dati: Modelli e linguaggi di Interrogazione*”, McGraw-Hill Libri Italia, Terza edizione, 2009
- [2] Alessandro Lenci, Simonetta Montemagni, Vito Pirrelli “*Testo e computer. Elementi di linguistica computazionale*”, Roma, Carocci, 2005
- [3] Andrea Romagnoli, Pasquale Salerno, Andrea Guidi “*AJAX per applicazioni web*”, Apogeo Editore, 2007
- [4] Susi Dulli, Sara Furini, Edmondo Peron “*Data Mining: Metodi e strategie*”, Springer Editore, Milano, 2007
- [5] Fosca Giannotti, Mirco Nanni, Dino Pedreschi , Fabio Pinelli , Chiara Renso , Salvatore Rinzivillo , Roberto Trasarti “*Unveiling the complexity of human mobility by querying and mining massive trajectory data*”
- [6] Michael Steinbach, Vipin Kumar, Pang-Ning Tan “*Introduction to Data Mining*”, Edizione 2, Addison-Wesley, 2013
- [7] Riccardo Guidotti, Michele Coscia, Dino Pedreschi, Diego Pennacchioli “*Behavioral Entropy and Profitability in Retail*”, IEEE International Conferene on. IEEE, 2015
- [8] Francesco Poli, “*Minimalismo, Arte Povera, Arte Concettuale*”, Bari, Laterza, 2009

Sitografia

- [9] Vanessa Vidale: La Psicologia nel colore del marketing (<http://www.vanessa-vidale.it/psicologia-del-colore-nel-marketing/>), data di consultazione 17/11/2016.
- [10] Documentazione PostgreSQL (<http://php.net/manual/en/book.pgsql.php>), data di consultazione 08/05/2016
- [11] LivLab, i Big data nel carrello della spesa e le opportunità per acquisti più sostenibili, Luca Aterini (<http://www.greenreport.it/news/consumi/i-big-data-nel-carrello-della-spesa-e-le-opportunita-per-acquisti-piu-sostenibili/>), data di consultazione 03/01/2017.
- [12] LivLab Livorno (<http://kdd.isti.cnr.it/project/LivLab-livorno>), data di consultazione 03/01/2017.

9. Appendice delle tabelle Unicoop

Pagamento

ATTRIBUTO	FUNZIONE	TIPO DI DATO	DESCRIZIONE
Id_pagamento		Incremental	Incrementale progressivo
Cod_pagamento	PrimaryKey	Integer	Codice del pagamento e chiave primaria della tabella
Orario		Time	Orario pagamento
Id_cliente	ForeignKey	Bigint	Chiave esterna che punta alla tabella Cliente
Id_negozio	ForeingKey	Bigint	Chiave esterna che punta alla tabella negozio
Cassa		Integer	Cassa dove è avvenuto il pagamento
Scontrino		Integer	Numero scontrino
Importo		Money	Importo del pagamento
Data		Date	Data del pagamento

Tabella 5. Pagamento

Mod_pagamento

ATTRIBUTO	FUNZIONE	TIPO DI DATO	DESCRIZIONE
Id_mod_pagamento	PrimaryKey	Incremental	Incrementale progressivo e chiave primaria della tabella modalità pagamenti
Descrizione		Varchar	Descrizione relativa alla modalità di pagamento (bancomat, postepay, etc.)
Id_pagamento	ForeignKey	Integer	Chiave esterna che punta alla tabella pagamenti

Tabella 6. Mod_pagamento

Articolo

ATTRIBUTO	FUNZIONE	TIPO DI DATO	DESCRIZIONE
Id_Articolo	PrimaryKey	Incremental	Chiave primaria della tabella
Cod_art	PrimaryKey	Numeric	Codice identificativo dell'articolo
ID_mkt	ForeignKey	Bigint	Chiave esterna che punta alla tabella marketing
Descrizione		Character varying	Descrizione dell'articolo
Marca		Character varying	Marca dell'articolo
fl_coop		Boolean	Articolo coop
fl_celiaci		Boolean	Articolo celiaci (gluten free)
fl_dop		Boolean	Articolo DOOP
fl_igp		Boolean	Articolo IGP
fl_tipico		Boolean	Articolo tipico
fl_solidale		Boolean	Articolo equo-solidale
fl_docg		Boolean	Articolo DOCG
fl_carrello		Boolean	Articolo carrello
Rilevanza		Boolean	Rilevanza dell'articolo ("true" è rilevante, "false" è irrilevante)

Tabella 7. Articolo

Device

ATTRIBUTO	FUNZIONE	TIPO DI DATO	DESCRIZIONE
Id_device	PrimaryKey	Incremental	Incrementale progressivo
DeviceID		Bigint	Codice IMEI del telefono, o di qualunque altro dispositivo.

Tabella 8. Device

Marketing

ATTRIBUTO	FUNZIONE	TIPO DI DATO	DESCRIZIONE
Id_mkt	PrimaryKey	Incremental	Incrementale progressivo e chiave primaria della tabella
Cod_mkt	PrimaryKey	Integer	Codice univoco di marketing
Area		Character varying	Area di marketing
Cod_area		Integer	Codice area di marketing
Macrosettore		Character varying	Macrosettore di marketing
Cod_macroset		Integer	Codice macrosettore
Settore		Character varying	Settore di marketing
Cod_settore		Integer	Codice settore
Reparto		Character varying	Reparto di marketing
Cod_reparto		Integer	Codice reparto
Categoria		Character varying	Categoria di marketing
Cod_categoria		Integer	Codice categoria
Sottocategoria		Character varying	Sottocategoria della categoria di marketing
Cod_subcat		Integer	Codice sottocategoria
Segmento		Character varying	Segmento di marketing
Cod_segmento		Integer	Codice segmento

Tabella 9. Marketing

Chiamata

ATTRIBUTO	FUNZIONE	TIPO DI DATO	DESCRIZIONE
Id_chiamata	PrimaryKey	Incremental	Incrementale progressivo e chiave primaria della tabella
Id_device	ForeignKey	Bigint	Chiave esterna che punta alla tabella Device
Hashed_name		Character varying	Codice criptato
Id_cliente	ForeignKey	Bigint	Chiave esterna che punta alla tabella cliente
Durata		Time	Durata della chiamata
Tipologia		Char	In entrata o in Uscita, I/O
Timestamp		Numeric	Data e ora in formato Unix

Tabella 10. Chiamata

Promo

ATTRIBUTO	FUNZIONE	TIPO DI DATO	DESCRIZIONE
Id_promo	PrimaryKey	Incremental	Chiave primaria della tabella
Pgr_promo		Bigint	Codice della promozione
Canale		Character	Tipo di promozione di un negozio (S per super, I per Iper, G per GestInCoop, ed E per E-commerce)
Cod_promo_ut		Bigint	chiave operativa offerta, canale non iper
Cod_promo_ip		Character Varying	chiave operativa offerta, canale Iper
Descrizione		Character Varying	Descrizione dell'offerta
Tipo		Character	Locale, distrettuale, nazionale e non specificato (L, D, N)
Data_inizio		Date	Data inizio promozione
Data_fine		Date	Data fine promozione

Tabella 11. Promo

SMS

ATTRIBUTO	FUNZIONE	TIPO DI DATO	DESCRIZIONE
Id_sms	PrimaryKey	Incremental	Incrementale progressivo e chiave primaria della tabella
Id_device	ForeignKey	Bigint	Chiave esterna che punta alla tabella Device
Id_cliente	ForeignKey	Bigint	Chiave esterna che punta alla tabella cliente
Hashed_name		Character varying	Codice criptato
Tipologia		Char	Sms in entrata o in Uscita, I/O
Time_stamp		Numeric	Data e ora in formato Unix

Tabella 12. SMS

Promodettaglio

ATTRIBUTO	FUNZIONE	TIPO DI DATO	DESCRIZIONE
Id_promo_dett	PrimaryKey	Incremental	Incrementale progressivo e chiave primaria della tabella promodettaglio
Id_promo	ForeignKey	Bigint	Chiave esterna che punta alla tabella promo
id_articolo	ForeignKey	Bigint	Chiave esterna che punta alla tabella articolo
id_negozio	ForeignKey	Bigint	Chiave esterna che punta alla tabella negozio
Descrizione_promo		Character varying	Descrizione promozione in percentuale
Data_inizio		Date	Data inizio promozione
Data_fine		Date	Data fine promozione
fl_volantino		Boolean	Articolo presente sul volantino
fl_distrettuale		Boolean	Articolo in promozione presente nel distretto
Target		Character varying	Indica il target a cui si rivolge la promozione: tutti, riservata ai soci, etc.

Tabella 13. Promodettaglio

Promomeccanica

ATTRIBUTO	FUNZIONE	TIPO DI DATO	DESCRIZIONE
Id_pgr_meccanica_offerta	PrimaryKey	Incremental	Chiave primaria della tabella promo meccanica
Id_promo	ForeignKey	Bingint	Chiave esterna che punta alla tabella promo
Descrizione		Character varying	Descrizione meccanica
Tipo_sconto		Character varying	Tipologia sconto: "Valore", "Percentuale", "MxN", "Erogazione punti", "Taglio prezzo", "Non disponibile"
Percentuale_sconto		Integer	Percentuale sconto
Pezzi_presi		Integer	Numero di pezzi presi
Pezzi_pagati		Integer	Numero di pezzi pagati
Sconto_valore		Decimal	Sconto del valore
Punti_jolly		Integer	Eventuali punti jolly
Punti_aggiuntivi		Integer	Eventuali punti aggiuntivi

Tabella 14. Promomeccanica