UNIVERSITÀ DEGLI STUDI DI TORINO

DIPARTIMENTO DI INFORMATICA

SCUOLA DI SCIENZE DELLA NATURA

Corso di Laurea Magistrale in Informatica

Progetto Modellazione Concettuale del Web Semantico

Lorenzo Sciandra

Stefano Vittorio Porta

ANNO ACCADEMICO
2020/2021

1 Motivazioni

Il dominio che abbiamo deciso di modellare riguarda la ristorazione. In particolare la gestione delle pietanze proposte dagli esercizi alimentari e le varie attività di consegna, recensione, ordine ed amministrazione ad essi collegati. L'idea è nata dal fatto che soprattutto nell'ultimo periodo, non solo nelle grandi città, ma anche nei più centri urbani di modeste dimensioni sempre più imprese di ristorazione sono ricorse a servizi di consegne, date le restrizioni sanitarie vigenti. Tra le varie applicazioni, anche molto rinomate, che mediano tra la proposta del locale e la richiesta dei clienti non tutte fanno affidamento ad una gestione semantica della conoscenza, preferendo piuttosto una classica gestione dei dati. L'idea di creare un'ontologia comune che modelli correttamente questo dominio potrebbe quindi risultare utile per condividere uno strumento univoco e potente in grado di facilitarne il compito. C'è anche da aggiungere che tra le varie piattaforme di delivery già esistenti da noi provate, nessuna permette di filtrare i locali anche in base alla gestione o meno di disturbi alimentari. Questo criterio di ricerca è stato da noi tenuto in considerazione nella modellazione.

2 Requisiti

2.1 Finalità

Il fine è quello della creazione di una ipotetica piattaforma di delivery che possa permettere ad un locale di esporsi facilmente con le proprie proposte a degli utenti che vi si rivolgono per saziare le proprie esigenze alimentari. Il tutto attraverso un reasoner che riesca facilmente ad inferire pietanze incompatibili o meno con diete come la vegana e la vegetariana o con intolleranze ed allergie come la celiachia e il favismo.
Gli utenti potranno infatti filtrare i piatti non solo in base al luogo geografico o alla tipologia di locale, ma anche a seconda della presenza o meno di piatti conformi a diete e disturbi o facendo riferimento a precedenti recensioni.

Seguiranno un paio di mockup dell'ipotetica piattaforma di delivery immaginata.

Figura 1: Pagina iniziale di ricerca

Figura 2: Risultati della ricerca locali a Torino

2.2 Task specifici e contesto

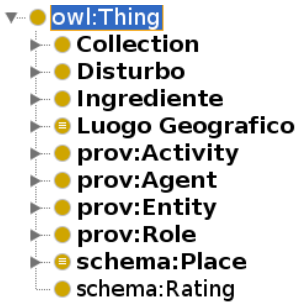
L'ontologia permette la modellazione di task come: la ricerca di un locale attraverso i filtri, l'ordine di una pietanza, la recensione di un esercizio alimentare a seguito di una consumazione, l'attività di consegna da parte di addetti e la gestione degli ordini e delle consegne da parte del proprietario dell'attività.

2.3 Utenti target

- Proprietari di esercizi alimentari che vogliono gestire il traffico lavorativo del loro locale;
- Persone che lavorano nel settore delle consegne e che usano l'applicativo per trovare nuovi ordini da recapitare;
- Utenti generici che vogliono effettuare un ordine da un locale o recensire un posto dopo una consumazione.

3 Descrizione del Dominio

Il dominio dell'ontologia spazia da argomenti alimentari a sanitari, passando attraverso la gestione delle azioni disponibili in una classica piattaforma di delivery e alla localizzazione geografica di un posto. Diamo uno sguardo alle classi discendenti direttamente da **owl:Thing**.



Come classi top-level troviamo: **Collection** importata dal pattern ODP Set, da noi usato per modellare le pietanze come insiemi di ingredienti; **Disturbo** allineato con i concetti della International Classification of Diseases; **Ingrediente**; **Luogo Geografico** allineato con Place di Schema.org; **schema:Rating** per classificare le recensioni dei locali e le 4 classi di Provenance usate per modellare le azioni.

Usando le classi di Provenance **Activity, Entity, Agent** e **Role** abbiamo classificato le azioni *Recensire, Ordinare, Consegnare* e *Avviare Impresa* ponendole come rdfs:subClassOf di prov:Activity al centro della rappresentazione. Attorno all'attività infatti ruotano i concetti di: agente, colui che

fa l'azione; ruolo, il compito svolto dalla persona nell'activity ed entità che identifica elementi creati o usati nel corso dell'azione.
Le sottoclasssi di **Disturbo**, allineate con owl:equivalentTo con i codici delle classi dell'ontologia ICD, ci permettono di introdurre le più note allergie e disturbi alimentari per classificare le pietanze come incompatibili con determinati problemi di salute.
Troviamo poi **Ingrediente** la cui gerarchia molto articolata permette di classificare gli ingredienti come semplici, derivati, vegetariani, vegani o non vegetariani, che verranno usati per comporre le pietanze.
Luogo geografico e le varie sottoclassi, spesso allineate con i concetti di Schema.org, sono state invece introdotte per permettere di dare una localizzazione geografica precisa agli esercizi alimentari.
Troviamo infine la classe **schema:Rating** che viene però leggermente modificata e semplificata rispetto alla classe di Schema.org, qui dotata semplicemente di una data property *Possiede Stelle* che permette di assegnare un punteggio da 1 a 5, come nella maggior parte delle recensioni.
L'idea di base da noi usata per allineare la nostra ontologia con risorse esterne è stata: usare la rdfs:subClassOf quando il nostro dominio doveva estendere un concetto top-level di una ontologia esterna in modo da poter sfruttare le object properties ad alto livello già definite dell'ontologia ed estenderle con quelle più specifiche per le classi sottostanti.
Quando invece l'ontologia esterna presenta nomi delle classi composti da codici (come quella dei disturbi), di difficile comprensione, o quando il nostro dominio si discosta leggermente dalla risorsa esterna e non abbiamo bisogno di usare delle properties già definite allora abbiamo effettuato l'allineamento solo di singole classi con owl:equivalentTo.
Per quanto riguarda le object properties le uniche importate sono quelle usate dal pattern Set e quelle di Provenance. Quest'ultime connettono le corrispettive classi importate e dalle quali, se necessario, sono state definite delle sotto proprietà che collegano direttamente le classi più specifiche. Tutte le altre object properties sono state invece definite ex novo.
Dando invece uno sguardo alle data properties notiamo che tra le importate vi sono **size** proviente dal pattern List che indica la dimensione dell'insieme modellato e **schema:ratingValue** che associa ad una classe Rating il valore double corrispondente. Questa è stato ulteriormente definita con una sotto data property *Possiede Stelle* che associa invece uno starRating, un datatype da noi definito che non nient'altro che una restrizione sui double che va da 1 a 5.

4 Documentazione del Dominio

5 LODE

6 Visualizzazione Ontologia

Vediamo ora una versione grafica dell'ontologia che ne espliciti le relazioni tra le varie classi mediante sussunzione.

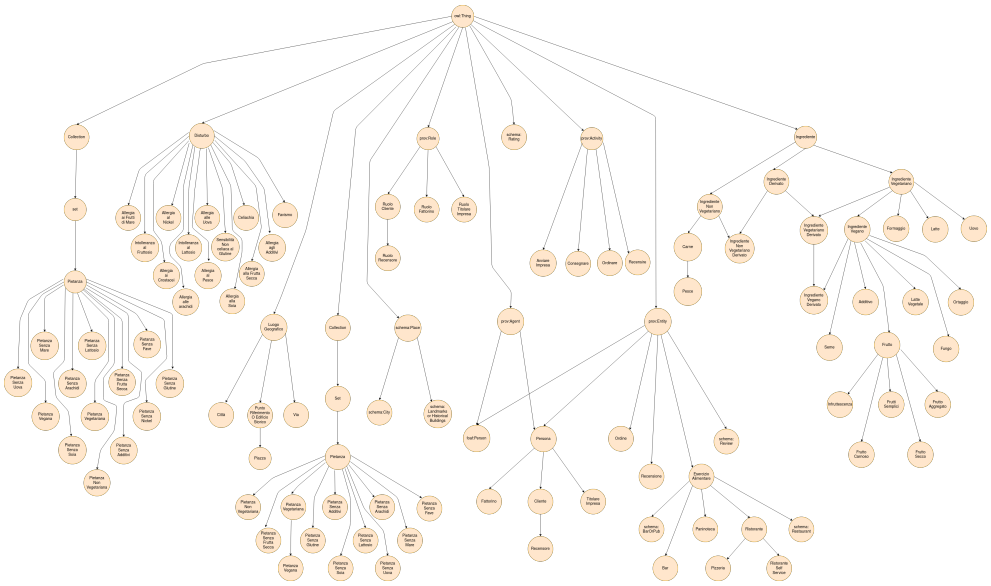


Figura 3: Gerarchia delle classi

Come si può osservare dall'immagine appena presentata per quanto riguarda la modellazione delle azioni abbiamo usato Provenance e in particolare il PROV Model, dal quale abbiamo definito delle sottoclassi più specifiche e delle sottoproprietà che collegassero le nuovi classi:

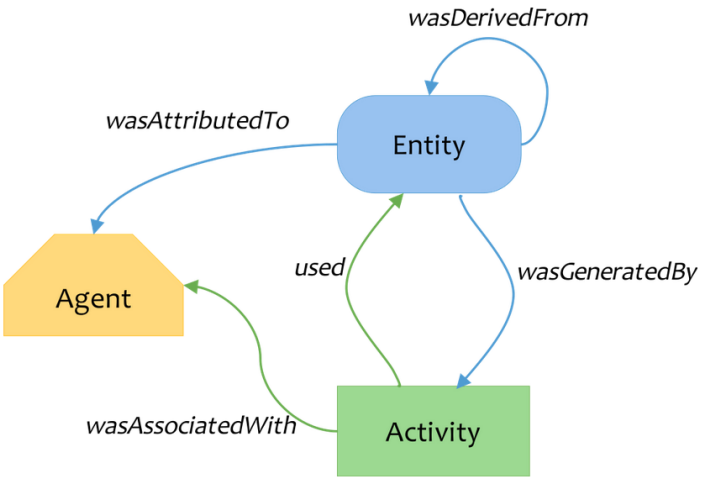


Figura 4: PROV Model

Per quanto riguarda invece la modellazione delle pietanze come insieme di ingredienti che le compongono abbiamo usato l' Ontology Design Pattern Set:

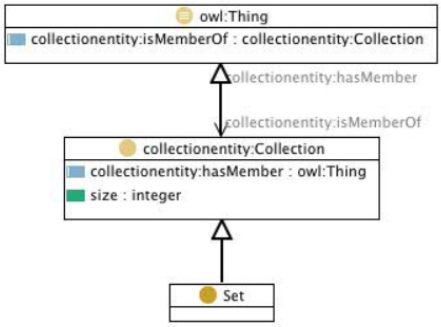


Figura 5: ODP Set

Abbiamo quindi definito la Pietanza come sussunta a Set ed usato la object property ereditata "hasMember" per la definizione delle nuove proprietà "utilizza ingrediente" e "utilizza indirettamente l'ingrediente" che collegano direttamente la pietanza con gli ingredienti di cui è composta.
Mostriamo ora alcune rappresentazioni grafiche estratte da GraphDB riguardo alle dipendenze tra le classi nella gerarchia.

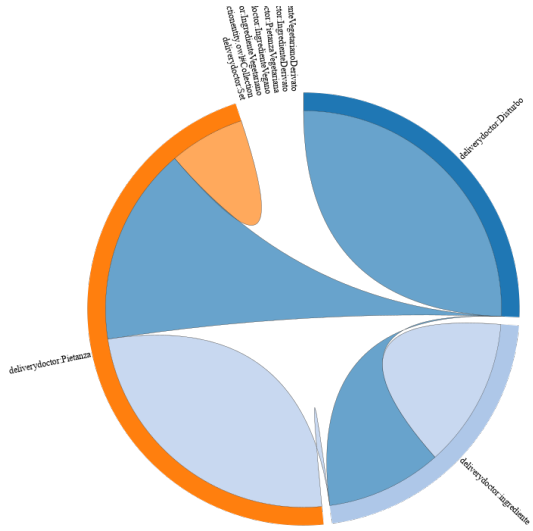


Figura 6: Dipendenze in entrata

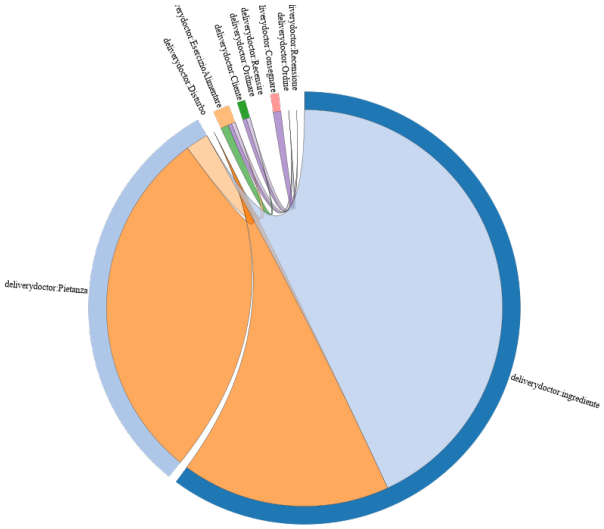


Figura 7: Dipendenze in uscita

7 Queries SPARQL

Suddivideremo questa sezione in 4 parti: la prima contenente l'interazione e quindi le queries di un untente normale, la seconda per l'utente proprietario di un esercizio alimentare, la terza riguardano invece l'interazione di un fattorino e l'ultima parte sarà dedicata alle queries dirette verso risorse ontologiche esterne.

7.1 Utente Normale

Il flusso d'interazione dell'utente generico Luca riguarderà l'iniziale ricerca dei locali nella sua città, Torino, per poi filtrare i risultati sulla base degli esercizi alimentari che dispongono di almeno una pietanza compatibile con la celiachia. Terminerà l'interazione visionando tutti gli ordini compiuti.

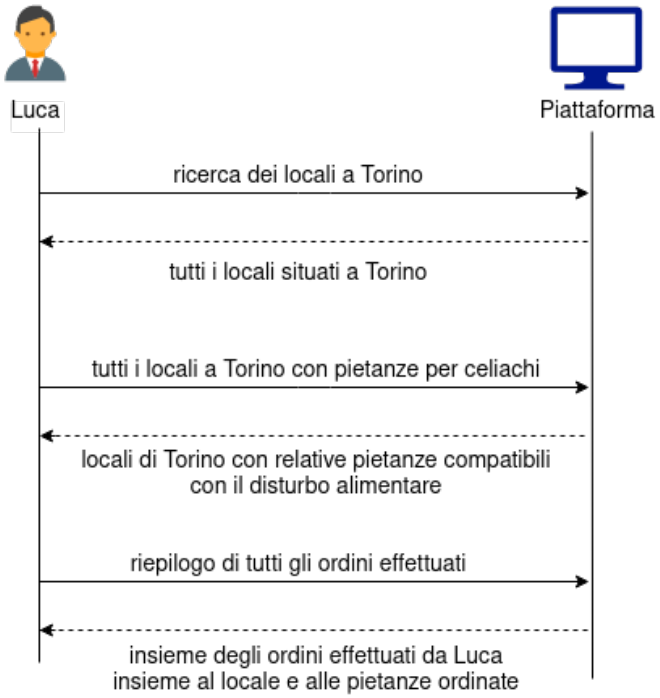


Figura 8: Interazione tra Luca e la piattaforma

Iniziamo con la Query che permette all'utente di ottenere tutti i locali situati a Torino.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX deliverydr:
<http://www.semanticweb.org/progettomodsem2020/deliverydoctor#>

SELECT ?localeNome WHERE{
    ?locale deliverydr:localeSituatoInCitta ?citta;
        deliverydr:haNome ?localeNome.
    ?citta deliverydr:haNome "Torino".
}
```

Con risultato:

?localeNome
Dubai Coffee Lounge
Napples

L'utente ora filtrerà i risultati ottenuti cercando i locali e le rispettive pietanze servite compatibili con il disturbo celiachia.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX deliverydr:
<http://www.semanticweb.org/progettomodsem2020/deliverydoctor#>

SELECT ?localeNome ?pietanza WHERE {
    ?locale deliverydr:localeSituatoInCitta ?citta;
        deliverydr:localeServePietanza ?pietanza;
        deliverydr:haNome ?localeNome.
    ?citta deliverydr:haNome "Torino".
    ?pietanza deliverydr:pietanzaCompatibileConDisturbo
                                    deliverydr:Celiachia,
                                    deliverydr:AllergiaGlutine.
}
```

Ottenendo come risultato:

?localeNome	?pietanza
Dubai Coffee Lounge	*deliverydr:CappuccinoSpeciale*
Dubai Coffee Lounge	*deliverydr:RisottoSugo*

Come possiamo notare dall'insieme dei locali verrà escluso Napples che nella nostra ontologia propone solo pizze e in maniera analoga tra le pietanze proposte dal Dubai verranno filtrate solo quelle compatibili con la celiachia/allergia al glutine.
Supponiamo ora che il nostro utente Luca voglia vedere lo storico degli ordini che ha effettuato.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX deliverydr:
<http://www.semanticweb.org/progettomodsem2020/deliverydoctor#>

SELECT ?ordine ?localeNome ?pietanza WHERE{
    ?ordinare deliverydr:ordinatoDa ?persona;
        deliverydr:ordinareDalLocale ?locale.
    ?ordine deliverydr:ordineCreatoDaAzione ?ordinare;
        deliverydr:ordineCompostoDaPietanza ?pietanza.
    ?persona deliverydr:haNome "Luca".
    ?locale deliverydr:haNome ?localeNome.
}
```

?ordine	?locale	?pietanza
deliverydr: CappuccinoECroissant	*Dubai Coffee Lounge*	*deliverydr: CappuccinoSpeciale*
deliverydr: CappuccinoECroissant	*Dubai Coffee Lounge*	*deliverydr:Croissant*

7.2 Utente Proprietario

Nel caso dell'Utente Proprietario, il flusso d'interazione è più frammentato, dato che i compiti che deve assolvere sono molteplici, e come esempio può:

1. Controllare gli ordini che la sua attività ha ricevuto
2. Controllare le recensioni lasciate dagli utenti sulla sua attività
3. Visionare il menù proposto dall'attività per valutare eventuali cambiamenti

Supponendo che il Proprietario diriga il locale "Dubai Coffee Lounge", è piuttosto immediata la ricerca di tutti gli ordini che gestisce e delle pietanze richieste:

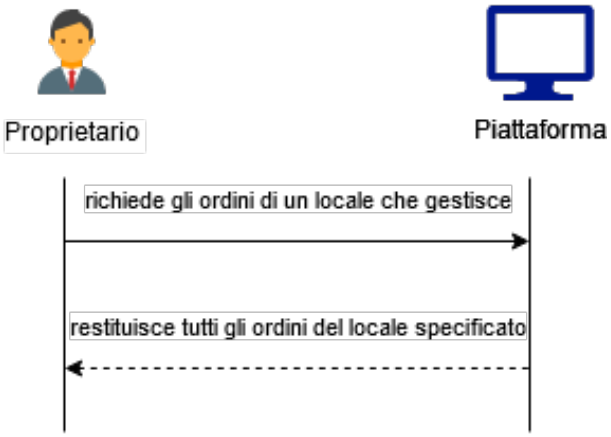


Figura 9: Il Proprietario richiede gli ordini gestiti per uno specifico locale

```
1  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3  PREFIX deliverydr:
4  <http://www.semanticweb.org/progettomodsem2020/deliverydoctor#>
5
6  SELECT ?ordine ?persona ?pietanza WHERE {
7      ?locale deliverydr:haNome "Dubai Coffee Lounge".
8      ?ordine deliverydr:ordineCompostoDaPietanza ?pietanza;
9              deliverydr:ordineCreatoDaAzione ?azione.
10     ?azione deliverydr:ordinatoDa ?persona;
11             deliverydr:ordinareDalLocale ?locale.
12 }
```

?ordine	?persona	?pietanza
deliverydr: CappuccinoECroissant	*deliverydr:Luca*	*deliverydr:Croissant*
deliverydr: CappuccinoECroissant	*deliverydr:Luca*	*deliverydr: CappuccinoSpeciale*

Assieme alla visualizzazione degli ordini gestiti, potrebbe essere utile sapere cosa pensano i clienti del locale e del servizio che offre. Per far ciò, è necessario recuperare alcuni dati:

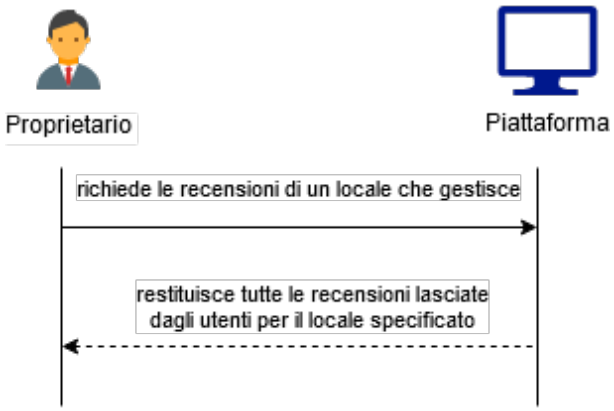


Figura 10: Il Proprietario richiede le recensioni pubblicate per uno specifico locale

```
1  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
2  PREFIX schema: <http://schema.org/>
3  PREFIX deliverydr:
4  <http://www.semanticweb.org/progettomodsem2020/deliverydoctor#>
5
6  SELECT ?persona ?testo ?stelle WHERE {
7      ?locale deliverydr:haNome "Dubai Coffee Lounge".
8      ?azione deliverydr:recensireLocale ?locale;
9        deliverydr:recensitoDa ?persona;
10       deliverydr:scrivereRecensione ?recensione.
11     ?recensione schema:reviewBody ?testo;
12                 schema:reviewRating ?rating.
13     ?rating deliverydr:possiedeStelle ?stelle
14 }
```

?persona	?testo	?stelle
deliverydr:Luca	*Servono un'ottima cheesecake e dei...*	*5*

Infine, se volesse vedere le pietanze che il locale serve al momento, per variare il menù (tenendo conto del numero di ingredienti che le compongono):

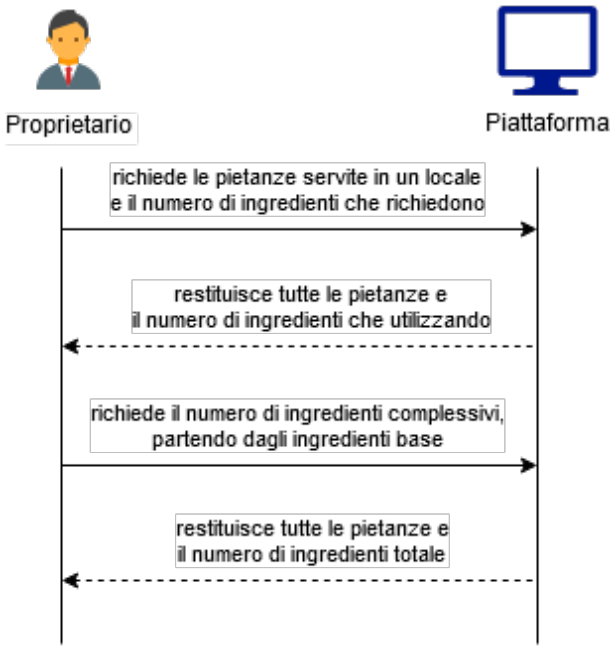


Figura 11: Il Proprietario richiede le pietanze servite in uno specifico locale, richiedendo successivamente anche il numero complessivo di ingredienti utilizzati

```
1  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
2  PREFIX deliverydr:
3  <http://www.semanticweb.org/progettomodsem2020/deliverydoctor#>
4
5  SELECT ?pietanza
6  (count(DISTINCT ?ingrediente) AS ?numero_ingredienti)
7  WHERE {
8      ?locale deliverydr:haNome "Dubai Coffee Lounge".
9      ?locale deliverydr:localeServePietanza ?pietanza.
10     ?pietanza deliverydr:utilizzaIngrediente ?ingrediente.
11 }
```

```
12 GROUP BY ?pietanza
```

?pietanza	?numero_ingredienti
deliverydr:PolpetteAlSugo	*3*
deliverydr:CappuccinoSpeciale	*2*
deliverydr:LasagneAlForno	*6*
deliverydr:Croissant	*4*
deliverydr:RisottoSugo	*2*

Potrebbe essere utile conoscere anche il numero di ingredienti non direttamente utilizzati dalla pietanza. In tal caso la query richiede anche una UNION:

```
1  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
2  PREFIX deliverydr:
3  <http://www.semanticweb.org/progettomodsem2020/deliverydoctor#>
4
5  SELECT ?pietanza
6  (count(DISTINCT ?ingrediente) AS ?numero_ingredienti)
7  WHERE {
8      ?locale deliverydr:haNome "Dubai Coffee Lounge".
9      ?locale deliverydr:localeServePietanza ?pietanza.
10     {?pietanza deliverydr:utilizzaIngrediente ?ingrediente}
11      UNION
12     {?pietanza deliverydr:utilizzaIngredienteIndirettamente
13        ?ingrediente}
14 }
15 GROUP BY ?pietanza
```

In questo caso i valori contenuti nella colonna del numero di ingredienti saranno leggermente differenti:

?pietanza	?numero_ingredienti
deliverydr:PolpetteAlSugo	*14*
deliverydr:CappuccinoSpeciale	*4*
deliverydr:LasagneAlForno	*15*
deliverydr:Croissant	*6*
deliverydr:RisottoSugo	*3*

7.3 Utente Fattorino

Per l'interazione con l'utente adibito alle consegne degli ordini, vediamo una tipica azione che potrebbe fare il fattorino Gianni, come controllare tutte le consegne che ha portato a termine.

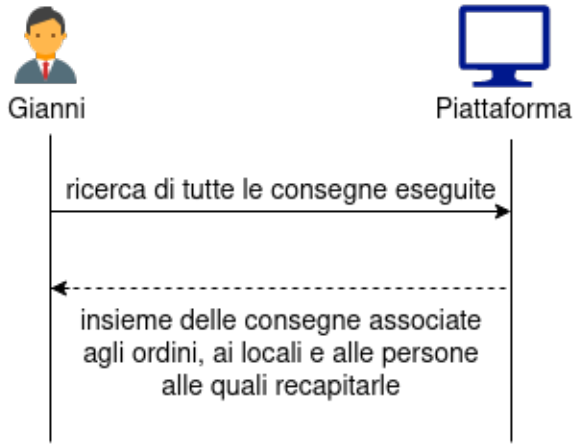


Figura 12: Interazione tra Gianni e la piattaforma

```
1  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2  PREFIX deliverydr:
3  <http://www.semanticweb.org/progettomodsem2020/deliverydoctor#>
4
5  SELECT ?consegna ?ordine ?localeNome ?destinatario WHERE{
6      ?consegna deliverydr:consegnareAlCliente ?destinatario;
7          deliverydr:consegnatoDa ?persona;
8          deliverydr:consegnareOrdine ?ordine.
9      ?persona deliverydr:haNome "Gianni".
10     ?ordine deliverydr:ordineCreatoDaAzione ?ordinare.
```

```
11      ?ordinare deliverydr:ordinareDalLocale ?locale.
12      ?locale deliverydr:haNome ?localeNome.
13  }
```

?consegna	?ordine	?localeNome	destinatario
deliverydr: ConsegnaOrdine Luca	*deliverydr: CappuccinoE Croissant*	*deliverydr: Dubai Coffee Lounge*	*deliverydr: Luca*

7.4 Queries verso risorse esterne

Verrà prima presentata un flowchart dell'interazione per poi soffermarci sulle due queries.

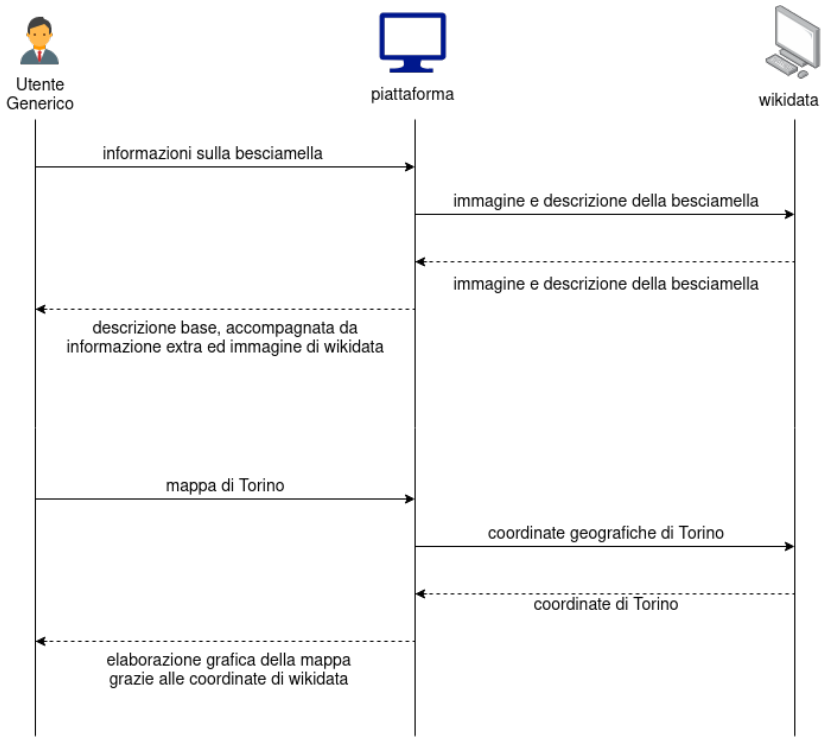


Figura 13: Schema interazione utente-piattaforma-wikidata

Come primo esempio di query indirizzata verso risorse esterne supponiamo la situazione in cui un utente, dopo aver visto gli ingredienti che compongono una pietanza, sia interessato ad ottenere informazioni su uno di essi. Supponiamo ad esempio che un utente, osservi le lasagne servite dal *Dubai Coffee Lounge* e voglia avere più informazioni riguardo la besciamella che le compone:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX schema: <http://schema.org/>
PREFIX deliverydr:
<http://www.semanticweb.org/progettomodsem2020/deliverydoctor#>

SELECT ?immagine ?descrizioneBase ?informazioneExtra WHERE {
    deliverydr:Besciamella rdfs:comment ?descrizioneBase.
    SERVICE <https://query.wikidata.org/sparql> {
        <http://www.wikidata.org/entity/Q209974>
        <http://www.wikidata.org/prop/direct/P18> ?immagine;
            schema:description ?informazioneExtra.
    }
    FILTER ( lang(?descrizioneBase) = "it" )
    FILTER ( lang(?informazioneExtra) = "it" )
}
```

Come risultato otteniamo l'immagine di Wikidata della besciamella, assieme alla descrizione base presente nell'ontologia e la descrizione dell'ingrediente estratta sempre da wikidata.

?immagine	?descrizioneBase	?informazioneExtra
https://upload.wikimedia.org/wikipedia/...	*Salsa base della cucina italiana, composta da farina, burro, latte e aromatizzata con noce moscata*	*salsa di base utilizzata per ricette più complesse*

Figura 14: Immagine referenziata dal link ricevuto

Come secondo caso immaginiamo invece un utente straniero in visita alla città di Torino. Volendo trovare dei locali presso i quali poter ordinare, potrebbe essere interessato ad ottenere una mappa della città su cui poterli collocare per una maggiore comodità nella ricerca. A tale scopo interroghiamo Wikidata per ottenere le coordinate geografiche della città, in modo poi da visualizzarli su una mappa di un'ipotetica applicazione finale centrata su di essa.

```
1 SELECT ?coordinate WHERE {
2   SERVICE <https://query.wikidata.org/sparql> {
3     <http://www.wikidata.org/entity/Q495>
4     <http://www.wikidata.org/prop/direct/P625> ?coordinate .
5     }
6 }
```

?coordinate
"Point(7.7 45.066666666)"

8 Applicazione Client

Il client da noi creato è stato realizzato usando HTML5, Bootstrap 5, Vue.js e Ajax. Per quanto riguarda la grafica abbiamo preso ispirazione dalle figure 1 e 2 in cui imaginavamo un mockup dell'ipotetica applicazione finale.
Il client si presenta quindi con una prima pagina html che funziona da smistamento verso le 3 pagine che daranno la possibilità di eseguire le queries introdotte nella selezione precedente, suddivise in base all'utente.

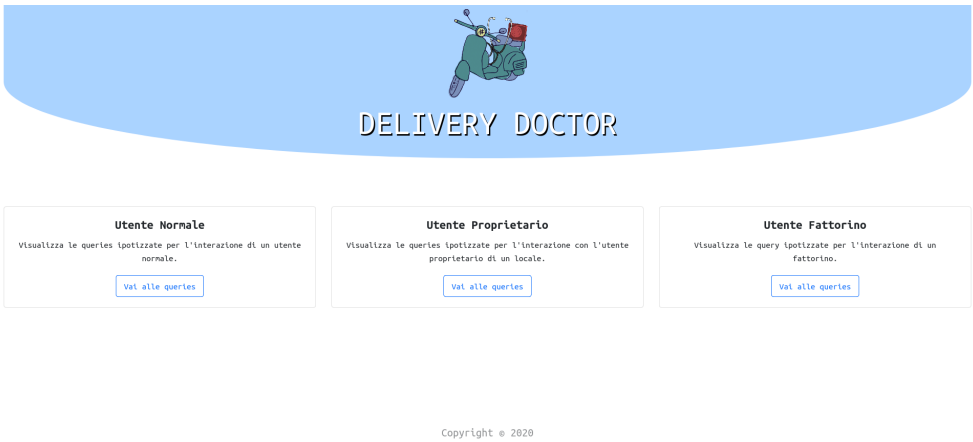


Figura 15: Pagina iniziale del client

Prendendo come esempio la sezione dell'utente normale, la nuova pagina presenterà una selezione tra le 3 queries introdotte. Una volta scelta la query e premuto il pulsante esegui la richiesta http verrà indirizzata a graphDB usando Ajax e il risultato ottenuto sarà gestito da Vue.js che ci permetterà di generare una tabella dinamica in base all'output ricevuto.

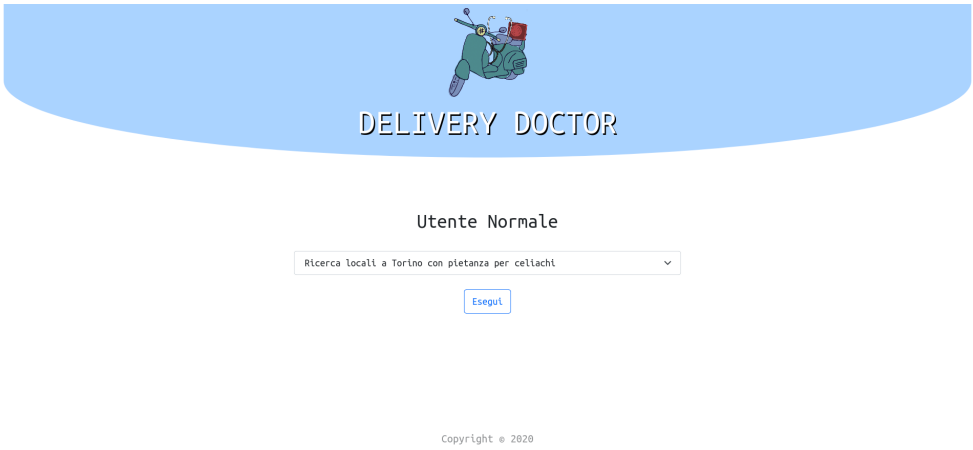


Figura 16: Pagina per eseguire le query di un utente normale

9 Allegati Full Size

Saranno presentate, in questa sezione finale della documentazione, alcune delle immagini ingrandite inserite nelle sezioni precedenti che a causa dello spazio limitato potevano risultare di difficile lettura.

owl:Thing
Collection
set
Pietanza
Pietanza Senza Uova
Pietanza Senza Mare
Pietanza Vegana
Pietanza Senza Arachidi
Pietanza Vegetariana
Pietanza Senza Soia
Pietanza Senza Lattosio
Pietanza Senza Frutta Secca
Pietanza Senza Additivi
Pietanza Non Vegetariana
Pietanza Senza Fave
Pietanza Senza Nickel
Pietanza Senza Glutine
Disturbo
Allergia ai Frutti di Mare
Allergia al Nickel
Intolleranza al Fruttosio
Allergia ai Crostacei
Allergia alle arachidi
Intolleranza al Lattosio
Allergia alle Uova
Allergia al Pesce
Sensibilità Non celiaca al Glutine
Allergia alla Frutta Secca
Allergia alla Soia
Celiachia
Allergia agli Additivi
Favismo
Luogo Geografico
Città
Punto Riferimento O Edificio Storico
Piazza
Via
Collection
Set
Pietanza
Pietanza Non Vegetariana
Pietanza Vegetariana
Pietanza Senza Frutta Secca
Pietanza Vegana
Pietanza Senza Glutine
Pietanza Senza Additivi
Pietanza Senza Soia
Pietanza Senza Lattosio
Pietanza Senza Arachidi
Pietanza Senza Uova
Pietanza Senza Mare
Pietanza Senza Fave
prov:Role
Ruolo Cliente
Ruolo Recensore
Ruolo Fattorino
Ruolo Titolare Impresa
schema: Rating
schema:Place
schema:City
schema: Landmarks or Historical Buildings
prov:Activity
Avviare Impresa
Consegnare
Ordinare
Recensire
prov:Agent
foaf:Person
Persona
Fattorino
Cliente
Recensore
Titolare Impresa
prov:Entity
Ordine
Recensione
schema: Review
Esercizio Alimentare
schema: BarOrPub
Paninoteca
Ristorante
schema: Restaurant
Bar
Pizzeria
Ristorante Self Service
Ingrediente
Ingrediente Derivato
Ingrediente Non Vegetariano
Carne
Pesce
Ingrediente Non Vegetariano Derivato
Ingrediente Vegetariano Derivato
Ingrediente Vegetariano
Ingrediente Vegano
Formaggio
Latte
Uovo
Ingrediente Vegano Derivato
Additivo
Latte Vegetale
Ortaggio
Seme
Frutto
Fungo
Infruttescenza
Frutti Semplici
Frutto Aggregato
Frutto Carnoso
Frutto Secco