

Un'App per scoprire l'Italia



Studenti:
Luca Peppe
Vincenzo Nastro

Docente:
Gennaro Costagliola

Introduzione

L'Italia è uno dei paesi più ricchi al mondo sia per la sua storia che per la sua cultura

Molte bellezze che offre purtroppo sono sconosciute ai turisti



DiscoverItaly è nata per valorizzare il territorio ed essere una guida per esplorare l'Italia.

Introduzione

DiscoverItaly è stata sviluppata come applicazione ibrida, utilizzando il framework Ionic 3.

In questo modo è possibile utilizzare DiscoverItaly su iOS, Android e Windows Phone, avendo in ogni caso un'UX pari a quella di un'applicazione nativa.



Introduzione

Per ottimizzare le performance dell'applicazione, è stato utilizzato il servizio di cloud computing AWS per l'hosting dei wrapper e del mediator.

Questa soluzione ha permesso di distribuire il calcolo e di rendere il backend flessibile a possibili picchi di carico.



Fonti

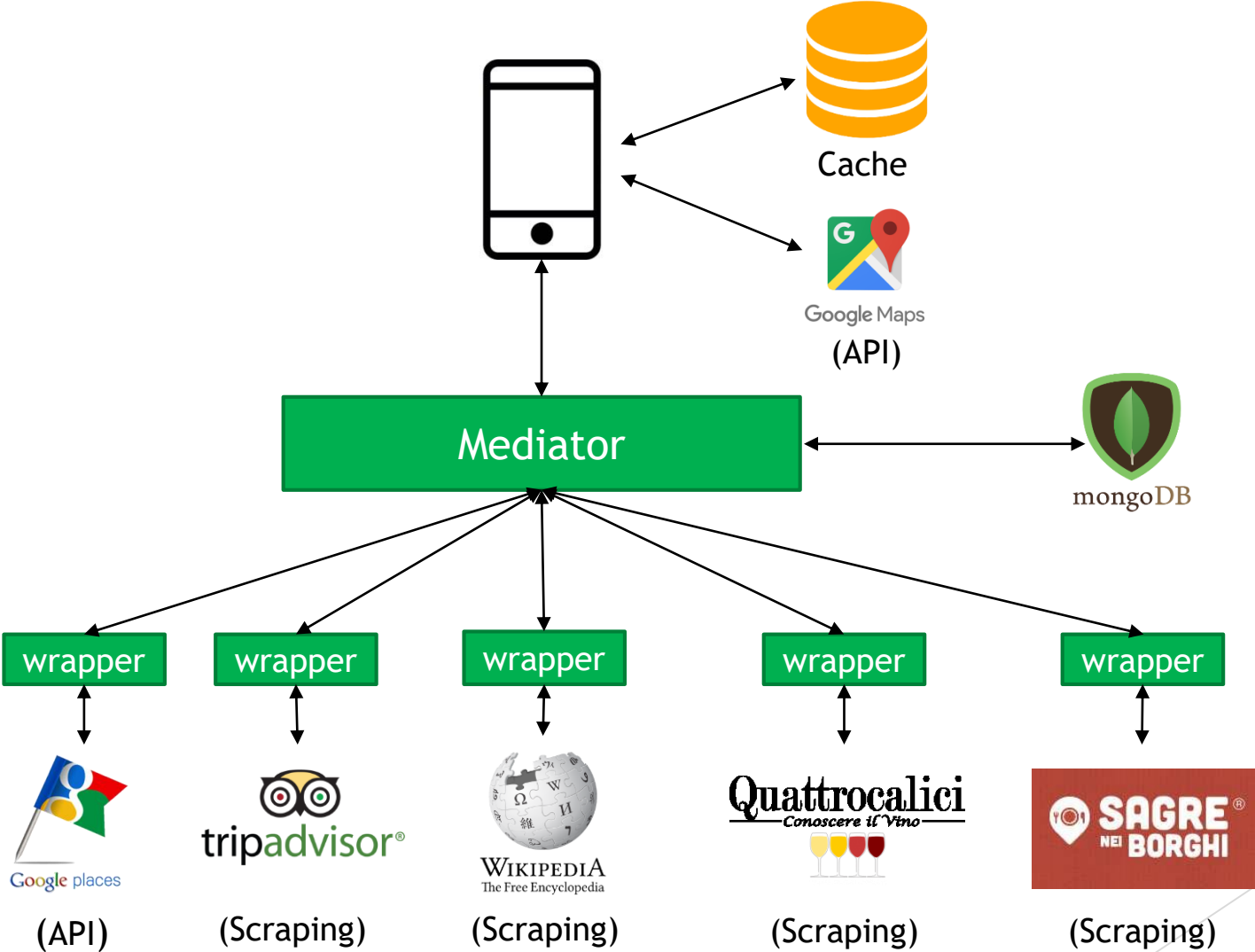
Luca Peppe:



Vincenzo Nastro:



Architettura



Fonti



Google Maps

Utilizzata attraverso il seguente plugin nativo di Ionic:

<https://ionicframework.com/docs/native/google-maps/>

Il plugin ha permesso di utilizzare le API di Google Maps in modo nativo su iOS e Android e ha offerto la funzionalità di geolocalizzazione nella schermata principale dell'applicazione.

Fonti



Le API di Google Places sono state utilizzate per offrire le seguenti funzionalità:

- autocompletamento dei luoghi inseriti dall'utente
- reverse geocoding (ricerca dei dettagli di un luogo partendo dalle coordinate)
- ricerca di ristoranti nelle vicinanze
- dettagli dei ristoranti

Volatilità: [aggiornamento mensile](#)

Fonti



Utilizzata per estrapolare informazioni riguardanti:

- Luoghi di interesse

Volatilità: *aggiornamento mensile*

- Recensioni dei ristoranti

Volatilità: *aggiornamento giornaliero*



Utilizzata per estrapolare informazioni riguardanti i prodotti DOP e IGP

Volatilità: *aggiornamento mensile*

Fonti



WIKIPEDIA
The Free Encyclopedia

Utilizzata per fornire una descrizione dettagliata dei luoghi di interesse trovati tramite Tripadvisor.

Volatilità: **aggiornamento mensile**



Utilizzata per estrapolare informazioni riguardanti eventi e le sagre

Volatilità: **aggiornamento giornaliero**

Wrapper sviluppati



Per estrapolare le informazioni da questa fonte abbiamo lavorato sulla struttura del DOM

Xpath = `//*[@id="centrale"]/div[2]/div/div[2]`
(restituisce la lista di tutti gli eventi trovati)




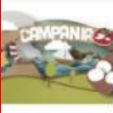
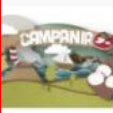
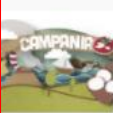

Xpath = `//*[@id="content_centrale"]`
(restituisce la descrizione di un evento)

Wrapper sviluppati

INFO ITALIA ZUCCHERI ACCEDI
MAPPA CALENDARIO SAGRE

SAGRE
NEI BORGHI

by *Italia Zuccheri*

	Sagra del Mare Flegrea	Dal 4 al 7 agosto 2016	Monte di Procida (NA)	Campania	RECENSIONE
	Fiordilatte Fiordifesta	Dal 6 al 8 agosto 2016	Agerola (NA)	Campania	RECENSIONE
	Sagra Del Fiordilatte Di Agerola	Dal 6 all'8 Agosto 2016	Agerola (NA)	Campania	NUMERATA DA NOI
	In Vico Veritas	27 agosto 2016	Vico Di Palma (NA)	Campania	
	Sagra della porchetta	Dal 14 al 21 gennaio 2017	Sant'antonio abate (NA)	Campania	
	Tombola Vajassa: una scostumatissima tombola napoletana	28 gennaio 2017	Napoli (NA)	Campania	
	Non è vero ma ci credo	4 febbraio 2017	Napoli (NA)	Campania	

//*[@id="centrale"]
/div[2]/div/div[2]

Wrapper sviluppati

1)



2)

Sagra del Mare
Flegrea

3)

Dal 4 al 7
agosto 2016

4)

Monte di Procida
(NA)

5)

Campania

RECENSIONE

1. `//*[@id="centrale"]/div[2]/div/div[2]/a[1]/div/div/div[2]`

2. `//*[@id="centrale"]/div[2]/div/div[2]/a[1]/div/div/div[3]`

3. `//*[@id="centrale"]/div[2]/div/div[2]/a[1]/div/div/div[4]`

4. `//*[@id="centrale"]/div[2]/div/div[2]/a[1]/div/div/div[5]`

5. `//*[@id="centrale"]/div[2]/div/div[2]/a[1]/div/div/div[6]`

Wrapper sviluppati


SAGRE
BORGH

INFO ITALIA ZUCCHERI CALENDARIO SAGRE MAPPA ACCEDI

Del 4 al 7 agosto 2016 MI piace 55

A Monte di Procida si festeggia il mare!

Sagra del Mare Flegrea
Monte di Procida (NA)



Tranquilli non avete letto male la Sagra del Mare di Flegrea è proprio a Monte di Procida.

Sì, perché a dispetto del nome che trae in inganno i più ignari, in questo ridente borgo campano si festeggia una delle sagre più belle mai dedicate alle prelibatezze del mare.

Nella incantevole cornice della splendida insenatura di Acquamorta al vostro arrivo troverete ad attendervi la migliore selezione di prelibatezze della gastronomia flegrea.

Il più goloso inizio a bagnare le papille con l'acqua di mare, perché tra ristoratori locali che propongono menù a tema e gli stand delle degustazioni i vostri palati verranno estasiati da piatti dal gusto inconfondibile che profumano di mare.

Primi piatti a base di pesce, pepata di cozze, bruschetta al sapore di mare, pizzette d'alghie e la proverbiale frittura mista in cui potrete trovare tutto il meglio del pesce fresco locale.

Non vi resterà che brindare al buongusto con i migliori vini DOC delle cantine campane, fare incetta di frutta di stagione e dolci tipici per chiudere la vostra espericolista da sagra

E SAPEVATE CHE...?

Le origini della frittura arrivano da molto lontano.

La cottura a base di fritto è infatti antichissimo, pare che fosse già diffusa addirittura nella società egiziana intorno al 2500 a.C.

Pare inoltre che questa pratica fosse particolarmente diffusa anche nell'antica Roma, dove le pietanze fritte erano consumate come cibo da strada.

Prodotto da
Italia Zuccheri
100% zucchero bianco

← `//*[@id="content_centrale"]`

Wrapper sviluppati



WIKIPEDIA
The Free Encyclopedia

La pagina di wikipedia relativa al monumento viene estrapolata dal motore di ricerca di Google usando come query:

Luogo_di_interesse «wikipedia» città

Xpath = `//*[@id="rso"]/div/div/div[1]/div/div/h3/a`
(restituisce le ancore di tutti gli elementi di Google)

Xpath = `//*[@id="mw-content-text"]`
(restituisce la descrizione relativa al luogo di interesse)

Wrapper sviluppati



Per lo scraping su Quattroclici è stato sviluppato un bot utilizzando selenium-webdriver e il browser headless PhantomJS.

L'esigenza di utilizzare un bot è nata dal fatto che i dati presenti all'interno della pagina non sono visualizzabili per intero, ma sono inseriti all'interno di una tabella con un numero limitato di righe, che deve essere necessariamente scorsa.

Per ottenere i prodotti tipici, il bot interagisce con la pagina web fingendosi un utente e, una volta raggiunti i dati, utilizza il seguente selettore CSS per estrapolarli:

```
#td-outer-wrap > div.td-main-content-wrap.td-container-wrap > div > div.td-pb-row.body-content > div.td-pb-span8.td-main-content > div.wpb_raw_code.wpb_content_element.wpb_raw_html.vc_custom_1493908674521.wine-list > div > div.bootstrap-table > div.fixed-table-container > div.fixed-table-body > table > tbody > tr'
```


Wrapper sviluppati



Anche per effettuare lo scraping dei luoghi di interesse e delle recensioni dei ristoranti sono stati sviluppati dei bot.

Una volta giunti alla pagina contenente le informazioni volute, sono stati utilizzati i seguenti selettori xpath per ottenere i div con i dati da estrapolare:

Luoghi di interesse:

```
//div[@class='listing_details' and  
not(div[@class='photo_booking']//div[@class='noImageBorder']//img[@class='npp']) and  
div[@class='listing_info']/div[@class='tag_line']/div/a]
```

Recensioni dei ristoranti:

```
#taplc_location_reviews_list_0 > div.review-  
container
```

Wrapper sviluppati

Luoghi di interesse -
Selettori CSS

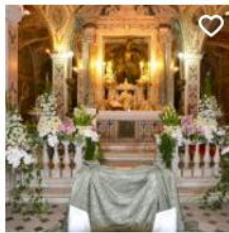


div.listing_title

div.listing_rating
span.ui_bubble_rating

div.photo_booking img

div.tag_line span



Duomo di Salerno

1.498 recensioni
N. 1 di 100 Cose da fare a Salerno
Chiese e cattedrali

Altre info

2 opzioni di
prenotazione



Giardino della Minerva

625 recensioni
N. 2 di 100 Cose da fare a Salerno
Giardini

Altre info



Lungomare di Salerno

1.350 recensioni
N. 3 di 100 Cose da fare a Salerno
Luoghi e punti d'interesse

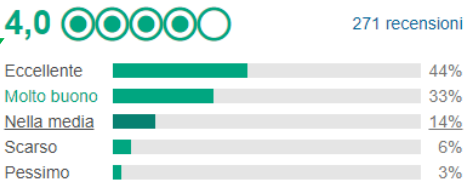
Altre info

Wrapper sviluppati



Recensioni dei ristoranti - Selettori CSS

```
#taplc_location_detail_overview_restaurant_0 > div.block_wrap > div.overviewContent > div.ui_columns.is-multiline.is-mobile.reviewsAndDetails > div.ui_column.is-6.reviews > div.rating > span
```







-   1 settimana fa
"Ottima scelta"
-   2 settimane fa
"Placevole e consigliato"

Diagram illustrating CSS selectors for restaurant reviews:

- `img.centeredImg` points to the profile picture of the reviewer.
- `span.ui_bubble_rating` points to the star rating.
- `span.ratingDate` points to the date of the review.
- `span.noQuotes` points to the review text.
- `div.ui_column.is-2` points to the entire review entry.
- `span.expand_inline.scrname` points to the expand/collapse icon.
- `p.partial_entry` points to the review text.

Example reviews shown:

- Alida S:  Recensito ieri  tramite dispositivo mobile
Eccellente
Rappresentate il migliore ristorante di Avellino per l'eccellenza del cibo, la scelta dei vini, la varietà infinita, il servizio celere e attento. Braviiiiii!!!
Grazie, Alida S
- Marta D:  Recensito 2 giorni fa  tramite dispositivo mobile
All perfect!
Uno dei ristoranti più importanti di Avellino! Servizio gentile, simpatico e disponibile
Piatti deliziosi, pizza buonissima. Prezzi altucci ma ne vale la pena!! Perfect!!
Grazie, Marta D

Wrapper sviluppati



Google places

La funzione di reverse geocoding è stata ottenuta sfruttando le API di Google Places. In particolare, è stata utilizzata la seguente richiesta HTTP:

https://maps.googleapis.com/maps/api/geocode/json?language=it&key=API_KEY&latlng=coordinate

I dati ottenuti dalla richiesta GET sono stati poi filtrati e ridotti ad un JSON.

Wrapper sviluppati



Per la ricerca dei ristoranti è stata utilizzata la seguente richiesta HTTP:

https://maps.googleapis.com/maps/api/place/nearbysearch/json?key=API_KEY&language=it&location=coordinate&radius=raggio&type=restaurant

Il parametro radius permette di specificare la distanza massima dei ristoranti da restituire. Anche in questo caso i dati sono stati filtrati e ridotti ad un JSON.

Wrapper sviluppati



Per la ricerca dei dettagli dei ristoranti è stata utilizzata la seguente richiesta HTTP:

https://maps.googleapis.com/maps/api/place/details/json?key=API_KEY&language=it&placeid=place_id

Con tale richiesta è stato possibile ottenere le seguenti informazioni:

- Apertura del ristorante
- Numero di telefono
- Coordinate del ristorante
- Recensioni
- Foto

Descrizione schemi - fonti locali



Google places

GP_place_det(placeid, città, provincia, regione, coordinate)

GP_ristorante(placeid, nome_ristorante, città, voto_medio, coordinate, num_telefono, foto, apertura)

GP_recensione(placeid, nome_utente, img_utente, voto, data, recensione)



QC_prodotto(prodotto, certificazione, descrizione, regione)

Descrizione schemi - fonti locali



TA_attrazione(nome, città, tipologia)

TA_recensione(nome_ristorante, città, nome_utente, img_utente, voto, data, titolo, recensione)

TA_ristorante_det(nome_ristorante, città, voto_medio)



WIKIPEDIA
The Free Encyclopedia

WIKI_descr(nome, città, descrizione)

Descrizione schemi - fonti locali



SB_sagra(nome, città, provincia, regione, data_inizio, data_fine, descrizione)

Descrizione schemi - schema globale



Dettagli_Posti(placeid, città, provincia, regione, coordinate)

Prodotti_Tipici(prodotto, certificazione, descrizione, regione)

Luoghi_di_Interesse(nome, città, tipologia, descrizione)

Sagre(nome, città, provincia, regione, data_inizio, data_fine, descrizione)

Ristoranti(placeid, nome_ristorante, città, coordinate, num_telefono, foto, apertura, voto_Google, voto_TripAdvisor)

Recensioni(placeid, nome_ristorante, città, nome_utente, img_utente, voto, data, recensione, titolo)

Mapping GAV

Dettagli_Posti(placeid, città, provincia, regione, coordinate) :-

GP_place_det(placeid, città, provincia, regione, coordinate)

Prodotti_Tipici(prodotto, certificazione, descrizione, regione) :-

QC_prodotto(prodotto, certificazione, descrizione, regione)

Luoghi di Interesse(nome, città, tipologia, descrizione) :-

TA_attrazione(nome, città, tipologia), **WIKI_descr**(nome, città, descrizione)

Sagre(nome, città, provincia, regione, data_inizio, data_fine,

descrizione) :- **SB_sagra**(nome, città, provincia, regione, data_inizio, data_fine, descrizione)

Mapping GAV

Ristoranti(placeid, nome_ristorante, città, coordinate, num_telefono, foto, apertura, voto_Google, voto_TripAdvisor) :-
GP_ristorante(placeid, nome_ristorante, città, voto_Google, coordinate, num_telefono, foto, apertura), **TA_ristorante_det**(nome_ristorante, città, voto_TripAdvisor)

Recensioni(placeid, nome_ristorante, città, nome_utente, img_utente, voto, data, recensione, titolo) :-
GP_recensione(placeid, nome_utente, img_utente, voto, data, recensione), **GP_ristorante**(placeid, nome_ristorante, città, voto_medio, coordinate, num_telefono, foto, apertura), titolo = 'null'

Recensioni(placeid, nome_ristorante, città, nome_utente, img_utente, voto, data, recensione, titolo) :-
TA_recensione(nome_ristorante, città, nome_utente, img_utente, voto, data, titolo, recensione), placeid='null'

Mapping LAV

GP_place_det(placeid, città, provincia, regione, coordinate) :-
Dettagli_Posti(placeid, città, provincia, regione, coordinate)

GP_ristorante(placeid, nome_ristorante, città, voto_medio, coordinate, num_telefono, foto, apertura) :-
Ristoranti(placeid, nome_ristorante, città, coordinate, num_telefono, foto, apertura, voto_medio, voto_TripAdvisor)

GP_recensione(placeid, nome_utente, img_utente, voto, data, recensione) :-
Recensioni(placeid, nome_ristorante, città, nome_utente, img_utente, voto, data, recensione, titolo), titolo = 'null'

QC_prodotto(prodotto, certificazione, descrizione, regione) :-
Prodotti_Tipici(prodotto, certificazione, descrizione, regione)

Mapping LAV

TA_attrazione(nome, città, tipologia) :-

Luoghi_di_Interesse(nome, città, tipologia, descrizione)

TA_recensione(nome_ristorante, città, nome_utente, img_utente, voto, data, titolo, recensione) :-

Recensioni(placeid, nome_ristorante, città, nome_utente, img_utente, voto, data, recensione, titolo), placeid = 'null'

TA_ristorante_det(nome_ristorante, città, voto_medio) :-

Ristoranti(placeid, nome_ristorante, città, coordinate, num_telefono, foto, apertura, voto_Google, voto_medio)

WIKI_descr(nome, città, descrizione) :-

Luoghi_di_Interesse(nome, città, tipologia, descrizione)

SB_sagra(nome, città, provincia, regione, data_inizio, data_fine, descrizione) :-

Sagre(nome, città, provincia, regione, data_inizio, data_fine, descrizione)

Query in datalog e SQL

RicercaLuoghiDiInteresse(coordinate, nome, città, tipologia)

:- Dettagli_Posti(_ , città, _ , _ , coordinate), **Luoghi_di_Interesse**(nome, città, tipologia, _)

```
SELECT Luoghi_di_Interesse.nome,  
Luoghi_di_Interesse.città,  
Luoghi_di_Interesse.tipologia  
FROM Luoghi_di_Interesse, Dettagli_Posti  
WHERE Dettagli_Posti.città = Luoghi_di_Interesse.città and  
Dettagli_Posti.coordinate = '40.9221644,14.7776341';
```

RicercaDescrizioneLuogoDiInteresse(nome, città, descrizione)

:- Luoghi_di_Interesse(nome, città, _ , descrizione)

```
SELECT descrizione  
FROM Luoghi_di_Interesse  
WHERE Luoghi_di_interesse.città = 'Roma' and  
Luoghi_di_interesse.nome = 'Colosseo';
```

Query in datalog e SQL

RicercaRistoranti(placeid, placeidRis, nome_ristorante, coordinate, foto, voto_Google) :- **Dettagli_Posti**(placeid , _ , _ , _ , coordinate), **Ristoranti**(placeidRis, nome_ristorante, _ , coordinate, _ , foto, _ , voto_Google, _)

```
SELECT Ristoranti.placeid as placeidRis,  
Ristoranti.nome_ristorante as nome_ristorante,  
Ristoranti.coordinate as coordinate,  
Ristoranti.foto as foto,  
Ristoranti.voto_Google as voto_Google  
FROM Ristoranti, Dettagli_Posti  
WHERE Dettagli_Posti.coordinate = Ristoranti.coordinate and  
Dettagli_Posti.placeid = 'ChIJ3dhaEIXMOxMR2k5gul0noQA';
```


Query in datalog e SQL

RicercaDettagliRistorante(placeid, città, num_telefono, foto, apertura, voto_TripAdvisor) :- **Ristoranti**(placeid, _, città, _, num_telefono, foto, apertura, _, voto_TripAdvisor)

```
SELECT città, num_telefono, foto, apertura, voto_TripAdvisor  
FROM Ristoranti  
WHERE Ristoranti.placeid = 'ChIJSxEHTVHMOxMRiVCgnJrl0FA';
```

Query in datalog e SQL

RicercaProdottiTipici(coordinate, prodotto, certificazione, regione) :- **Dettagli_Posti**(_, _, _, regione, coordinate),
Prodotti_Tipici(prodotto, certificazione, _, regione)

```
SELECT Prodotti_Tipici.prodotto,  
Prodotti_Tipici.certificazione,  
Prodotti_Tipici.regione  
FROM Prodotti_Tipici, Dettagli_Posti  
WHERE Dettagli_Posti.regione = Prodotti_Tipici.regione and  
Dettagli_Posti.coordinate = '40.9221644,14.7776341';
```

Query in datalog e SQL

RicercaDettagliProdottoTipico(prodotto, certificazione, regione, descrizione) :- **Prodotti_Tipici**(prodotto, certificazione, descrizione, regione)

```
SELECT descrizione  
FROM Prodotti_Tipici  
WHERE Prodotti_Tipici.prodotto='Arancia del Gargano IGP' and  
Prodotti_Tipici.certificazione='IGP' and  
Prodotti_Tipici.regione='Puglia'
```

Query in datalog e SQL

RicercaRecensioniRistoranteByPlaceid(placeid, nome_utente, img_utente, voto, data, recensione, titolo) :-
Recensioni(placeid, nome_ristorante, città, nome_utente, img_utente, voto, data, recensione, titolo)

```
SELECT nome_utente, img_utente, voto, data, recensione, titolo  
FROM Recensioni  
WHERE placeid = 'ChiJSxEHTVHMOxMRiVCgnJrl0FA'
```

RicercaRecensioniRistoranteByNome(nome_ristorante, città, nome_utente, img_utente, voto, data, recensione, titolo) :-
Recensioni(placeid, nome_ristorante, città, nome_utente, img_utente, voto, data, recensione, titolo)

```
SELECT nome_utente, img_utente, voto, data, recensione, titolo  
FROM Recensioni  
WHERE nome_ristorante = 'Da Giorgio' AND città = 'Avellino'
```

Query in datalog e SQL

RicercaSagre(placeid, nome, città, provincia, regione, data_inizio, data_fine) :- **Dettagli_Posti**(placeid, città, provincia, regione, _), **Sagre**(nome, città, provincia, regione, data_inizio, data_fine, _)

```
SELECT Sagre.nome,  
Sagre.città,  
Sagre.provincia,  
Sagre.data_inizio,  
Sagre.data_fine  
FROM Sagre, Dettagli_Posti  
WHERE Dettagli_Posti.città = Sagre.città and  
Dettagli_Posti.provincia = Sagre.provincia and  
Dettagli_Posti.regione = Sagre.regione and  
Dettagli_Posti.placeid = 'ChIJ3dhaEIXMOxMR2k5gul0noQA';
```

Query in datalog e SQL

RicercaDettagliSagra(nome, città, provincia, regione, descrizione) :- **Sagre**(nome, città, provincia, regione, _ , _ , descrizione)

```
SELECT descrizione
FROM Sagre
WHERE Sagre.nome='Sagra dell'uva di Rutigliano'
Sagre.città='Rutigliano',
Sagre.provincia='Bari',
Sagre.regione='Puglia'
```

Unfolding Algorithm

RicercaLuoghiDiInteresse(coordinate, nome, città, tipologia) :-
Dettagli_Posti(v1, città, v2, v3, coordinate), **Luoghi_di_Interesse**(nome,
città, tipologia, v4)

1

RicercaLuoghiDiInteresse'(coordinate, nome, città, tipologia) :-
GP_place_det(v1, città, v2, v3, coordinate), **Luoghi_di_Interesse**(nome,
città, tipologia, v4)

2

RicercaLuoghiDiInteresse'(coordinate, nome, città, tipologia) :-
GP_place_det(v1, città, v2, v3, coordinate), **TA_attrazione**(nome, città,
tipologia), **WIKI_descr**(nome, città, v4)

Bucket Algorithm

RicercaRistoranti(placeid, placeidRis, nome_ristorante, coordinate, foto, voto_Google) :- **Dettagli_Posti**(placeid, città, provincia, regione, coordinate), **Ristoranti**(placeidRis, nome_ristorante, città, coordinate, num_telefono, foto, apertura, voto_Google, voto_TripAdvisor')

Bucket filling

Dettagli_Posti

GP_place_det(placeid, città, provincia, regione, coordinate)

Ristoranti

GP_ristorante(placeid, nome_ristorante, città, voto_medio, coordinate, num_telefono, foto, apertura)

TA_ristorante_det (nome_ristorante, città, voto_medio)

Bucket Algorithm

RicercaRistoranti(placeid, placeidRis, nome_ristorante, coordinate, foto, voto_Google) :-
Dettagli_Posti(placeid , città', provincia', regione', coordinate),
Ristoranti(placeidRis, nome_ristorante, città'', coordinate, num_telefono', foto, apertura',
voto_Google, voto_TripAdvisor')

Dettagli_Posti

GP_place_det(placeid, città,
provincia, regione, coordinate)

Ristoranti

GP_ristorante(placeid,
nome_ristorante, città, voto_medio,
coordinate, num_telefono, foto,
apertura)

TA_ristorante_det (nome_ristorante,
città, voto_medio)

RicercaRistoranti'(placeid, placeidRis, nome_ristorante, coordinate, foto, voto_Google) :-
GP_place_det(placeid , città', provincia', regione', coordinate),
GP_ristorante(placeidRis, nome_ristorante, città'', voto_Google, coordinate,
num_telefono', foto, apertura')

Bucket Algorithm

Checking containment

RicercaRistoranti'(placeid, placeidRis, nome_ristorante, coordinate, foto, voto_Google) :-
GP_place_det(placeid , città', provincia', regione', coordinate),
GP_ristorante(placeidRis, nome_ristorante, città'', voto_Google, coordinate,
num_telefono', foto, apertura')

RicercaRistoranti'(placeid, placeidRis, nome_ristorante, coordinate, foto, voto_Google) :-
Dettagli_Posti(placeid, città', provincia', regione', coordinate) ,
GP_ristorante(placeidRis, nome_ristorante, città'', voto_Google, coordinate,
num_telefono', foto, apertura')

RicercaRistoranti'(placeid, placeidRis, nome_ristorante, coordinate, foto, voto_Google) :-
Dettagli_Posti(placeid, città', provincia', regione', coordinate) ,
Ristoranti(placeidRis, nome_ristorante, città'', coordinate, num_telefono', foto, apertura',
voto_Google, voto_TripAdvisor')

RicercaRistoranti' \subseteq **RicercaRistoranti**

RicercaRistoranti' è una riscrittura massimalmente contenuta di **RicercaRistoranti**

Tecnologie utilizzate



ANGULARJS



APACHE
CORDOVA™



mongoDB



ionic



{JSON}