

# Tuscany by POI

Progetto per il corso di Laboratorio Progettazione  
web

**Studente: Elisabetta Triolo**

**Numero Matricola: 477836**

**Corso di laurea: Informatica Umanistica**

## 1. **Che cos' è?**

**Tuscany By POI** è un progetto che ha lo scopo di guidare l'utente nella scelta dell'alloggio durante la sua permanenza in Toscana, tenendo, però, fortemente conto dei Punti di Interesse (POI, abbreviazione dell'inglese *Point of Interest*) della regione.

Il progetto, infatti, copre tutti i comuni della Toscana e permette di scegliere tra una serie di categorie di POI (ristoranti, pizzerie, monumenti, teatri, cinema, librerie, ecc.), quella interessata, di cercare i POI appartenenti a quella categoria e visualizzarli su una mappa, di trovare le tipologie di strutture ricettive (alberghi, affittacamere, case vacanza, agriturismi, ecc.) che si trovano nel raggio di 250 metri da un particolare POI scelto. Per ogni tipologia di struttura trovata l'applicazione permette, inoltre, di visualizzare le informazioni sulle strutture ricettive a essa appartenenti e le ordina in base alla distanza a piedi crescente dal POI scelto. Infine, per ogni struttura ricettiva è possibile osservare sulla mappa il percorso a piedi che la separa dal POI selezionato.

## 2. **Lo stato dell'arte**

Il campo del web riservato ai viaggi e, in particolare, alla ricerca dell'alloggio, di Punti di Interesse e di attività da svolgere quando ci si trova o si sta per arrivare in un dato territorio è molto ampio. Esistono, infatti, tanti siti web che svolgono queste e molte altre funzioni. Basta citare il famosissimo *Trip Advisor*, ma la lista comprende anche piattaforme come *Expedia*, *Hotel Travel*, *Hotels*, *Hotel Booking*, *Venere*, *Kayak*, *Trivago*.

Inserendosi all'interno di questo vasto ambito, il progetto *Tuscany By POI*, trova il suo principale elemento di differenziazione nella sua immediata focalizzazione non sulla struttura destinata all'alloggio, ma sul tipo e sulle caratteristiche dei Punti di Interesse del territorio. L'utente è messo nella condizione di ricercare i Punti di Interesse e solo dopo aver scelto ciò che gli interessa vedere o conoscere potrà visualizzare anche le strutture ricettive nei pressi. Oltre al suo essere incentrato sui POI, il progetto presenta almeno altri due elementi di differenziazione. Il primo è che esso è incentrato su un singolo territorio amministrativo, cioè la Regione Toscana con tutti i suoi 279 comuni. Il secondo è che esso mostra le strutture ricettive in ordine di distanza a piedi crescente da un particolare POI e che permette di visualizzare il percorso a piedi tra ogni POI e ogni struttura. Anche altri siti danno informazioni sulla distanza tra una struttura ricettiva e un Punto di Interesse ma, a parte il fatto che essi di solito partono dalla ricerca della struttura e non del POI, i POI interessati non sono molti e non sono ordinati per categoria. Inoltre, altre piattaforme simili non permettono di visualizzare direttamente sul sito il percorso a piedi che separa ogni struttura da un dato POI.

Il progetto *Tuscany By POI* ha un altro elemento di differenziazione che può anche essere un punto di forza e cioè esso è basato sulle "piccole" distanze, infatti, per ogni POI vengono mostrate le strutture ricettive che si trovano nel raggio di 250 metri da esso e il percorso tra un POI e una struttura visualizzato è un percorso a piedi, tutte cose che ne fanno un'applicazione principalmente indirizzata a chi ha intenzione di muoversi a piedi.

## 3. **La raccolta e la normalizzazione dei dati**

Per realizzare questo progetto sono state necessarie principalmente tre tipologie di dati, cioè quelli riguardanti le coordinate geografiche e i confini amministrativi dei comuni della Regione Toscana, quelli riguardanti i Punti di Interesse della regione e quelli riguardanti le strutture ricettive.

Per le coordinate geografiche dei comuni toscani ho scaricato il file *Comuni-Geolocalizzati-Provincie-Regioni.sql* all'indirizzo <http://www.loosecode.com/blog/development/elenco-comuni-italiani-con-le-coordinate-geografiche-con-provincie-e-regione.html> e poiché esso conteneva informazioni sulle coordinate di tutti i comuni italiani, è stato necessario estrapolare solo quelle riguardanti i comuni toscani. Per fare questo, ho scritto un breve programma php che, basandosi sui codici delle province Istat (che per la Toscana vanno da 045 a 053 più il codice 100), e su un'altra tabella scaricata dal sito dell'Istat che conteneva informazioni sui comuni italiani, è servito a creare la nuova tabella *comunitoscanageolocalizzati* formata solo da tutti i comuni toscani geolocalizzati.

Alla fine ho aggiunto un id a ogni comune.

Per i confini amministrativi dei comuni toscani, la fonte usata è stata *MAPZEN*

```
<?php
$conn=mysql_connect("localhost","root","");
if(!$conn){
    die ("Errore di connessione con il server localhost.".mysql_errno());
}

mysql_query("SET character_set_results = 'utf8', character_set_client = 'utf8', character_set_connection = 'utf8',
character_set_database = 'utf8', character_set_server = 'utf8'", $conn);

$comunigeoloc=mysql_select_db("cittatoscana");
if(!$comunigeoloc){
    die("Impossibile selezionare il database cittatoscana.".mysql_errno());
}

$comuni=file_get_contents("confinicomunitoscanadef.geojson");
$comunid=json_decode($comuni,true);
foreach($comunid["features"] as $value){
    $count=0;
    $idcomune=$value["id"];
    $nome=addslashes($value["name"]);
    echo($nome."\t");
    echo(count($value["geometry"]["coordinates"])."\n");
    foreach($value["geometry"]["coordinates"] as $val){
        $count++;
        $arrayCl="arraycoord".$count.".lng";
        echo($arrayCl."\n");
        $arrayCla="arraycoord".$count.".lat";
        echo($arrayCla."\n\n");
        $query="ALTER TABLE confinicomuni ADD $arrayCl double";
        mysql_query($query);
        $query="ALTER TABLE confinicomuni ADD $arrayCla double";
        mysql_query($query);
        foreach($val[0] as $valu){

            $lng=$valu[0];
            $lat=$valu[1];

            $query="Insert into confinicomuni (id_comune,Nome,$arrayCl,$arrayCla) Values
('$idcomune','$nome','$lng','$lat)";
            mysql_query($query);
        }
    }
}
?>
```

Fig.1: Programma php usato per creare la tabella *confinicomuni*

(<https://mapzen.com/data/borders>), una risorsa web che fornisce in formato geojson dati sui confini amministrativi tratti da *OpenStreetMap*. Il file geojson scaricato da *MAPZEN* conteneva informazioni sui confini amministrativi di tutti i comuni italiani (circa 8000). È stato necessario estrapolare dal file solo le informazioni sui confini amministrativi dei comuni toscani e creare un nuovo file che contenesse solo queste informazioni attraverso un programma javascript e uno php.

Le informazioni contenute sul nuovo file, sempre in formato geojson, sono state trasferite sul server grazie a un altro programma php per mezzo del quale è stata creata la tabella *confinicomuni*. (Vedi Fig.1)

Le informazioni sui Punti di interesse della Toscana provengono in parte dal sito degli open data della Regione Toscana (<http://open.toscana.it/>) e in parte sono stati estrapolati grazie al servizio online di geocoding di *ArcGIS*, un sistema informatico geografico di *Esri*, all'indirizzo <http://geocode.arcgis.com/arcgis/index.html>.

In particolare, dal sito degli open data della Regione Toscana, provengono i dati riguardanti i luoghi dello spettacolo, i musei, i ristoranti e le botteghe, mentre il servizio messo a disposizione da *ArcGIS* è stato usato per estrapolare i dati sulle altre categorie di POI, cioè Chiese, Caffetterie, Bar e Pub, Alimentari e supermarket, Ristoranti cinesi, Ristoranti giapponesi, Pizzerie, Centri commerciali, Negozi di abbigliamento, Librerie, Giardini, Parchi, Fast Food, Stazioni ferroviarie, Aeroporti.

I dati estrapolati dal sito degli open data della Toscana, però, non tenevano conto di alcuni cambiamenti nei confini amministrativi dei comuni avvenuti a partire dal 2010 e soprattutto nel 2014, e ciò avrebbe reso problematico far corrispondere questi dati con quelli dei confini amministrativi, in modo che ad esempio, per un comune come *Casciana Terme Lari* (formatosi nel 2014), sarebbe stato impossibile mostrare i POI presenti sul suo territorio, visto che nelle tabelle non è presente questo comune ma vi sono singolarmente i due comuni che lo hanno formato, cioè *Casciana Terme* e *Lari*.

Per ovviare a questo problema, ho scritto un programma php che sulla base dei nomi dei comuni e dei codici Istat delle province, usando la tabella precedentemente creata *comunitoscanageolocalizzati*, ha assegnato a ogni riga delle tabelle dei POI un id corrispondente all'id del comune toscano in cui ognuno di essi si trova e che è riportato proprio nella tabella *comunitoscanageolocalizzati*. (Vedi Fig.2)

Per conformità dei dati, l'id è stato aggiunto non solo alle tabelle provenienti dal sito degli open data della Toscana, ma anche alla tabella costruita con le informazioni provenienti da *ArcGIS*.

```

<?php
$conn=mysql_connect("localhost","root","");
if(!$conn){
    die ("Errore di connessione con il server localhost.".mysql_errno());
}
mysql_query("SET character_set_results = 'utf8', character_set_client = 'utf8', character_set_connection = 'utf8',
character_set_database = 'utf8', character_set_server = 'utf8'", $conn);
$db=mysql_select_db("poi");
if(!$db){
    die ("Impossibile selezionare il database POI.".mysql_errno());
}
$query1="SELECT citta,provincia FROM strutturericettivetoscana";
$result1=mysql_query($query1);
if(!$result1){
    die ("query1 non riuscita.".mysql_errno());
}
$query2="SELECT id_comune,Provincia,codice_provincia,Comune FROM comunitoscanageolocalizzati";
$result2=mysql_query($query2);
if(!$result2){
    die ("query2 non riuscita.".mysql_errno());
}
while ($row1=mysql_fetch_assoc($result1)){
    $comune=$row1["citta"];
    $comune1=addslashes($comune);
    $provincia=$row1["provincia"];
    while ($row2=mysql_fetch_assoc($result2)){
        $comune2=$row2["Comune"];
        $provincia2=$row2["Provincia"];
        $provincia2l=addslashes($provincia2);
        $idcomune=$row2["id_comune"];
        $codiceprovincia=$row2["codice_provincia"];
        if($comune==$comune2 and $provincia==$codiceprovincia){
            $query="UPDATE strutturericettivetoscana SET
id_comune='$idcomune',NomeProvincia='$provincia2l' WHERE citta='$comune1' and provincia='$provincia'";
            $do_query=mysql_query($query);
            if(!$do_query){
                die("query fallita ".mysql_errno());
            }
            break;
        }
    }
}
?>

```

Fig.2: Porzione del programma php usato per assegnare un id a ogni comune delle tabelle dei POI e della tabella delle strutture ricettive

I dati provenienti dal servizio online di geocoding di *ArcGIS* sono stati restituiti in formato json ed è stato, dunque, necessario elaborare un altro programma php che li trasferisse sul server, nella tabella *poiarctoscana*. (Vedi Fig.3)

```

<?php
$conn=mysql_connect("localhost","root","");
if(!$conn){
    die ("Errore di connessione con il server localhost.".mysql_errno());
}

mysql_query("SET character_set_results = 'utf8', character_set_client = 'utf8', character_set_connection = 'utf8',
character_set_database = 'utf8', character_set_server = 'utf8'", $conn);
$db=mysql_select_db("poi");
$file="cittatoscanaN.json";
$filer=file_get_contents($file);
$filed=json_decode($filer);
for ($j=120;$j<count($filed);$j++){
    //print("$number\n");*/
    $ch = curl_init();
    $urlR="http://geocode.arcgis.com/arcgis/rest/services/World/GeocodeServer/find?text=Airport+in+".$filed[$j]."&source
Country=ITA&outFields=PlaceName%2CCity%2CCountry%2CRegion,Place_addr,StAddr,Postal,Phone,URL,Nbrhd,Subregion&maxLoc
ations=50&forStorage=true&token=gtZfv1RQOsawytqIZDeb1n2jvKfxUoa-Yxd7hLc1gnATroWGQqyA7H-7JdRllr6xOnR-
Tl6khTNWduRW0liITCEL9EkADKTYTeSuaKZPf3grHe2VcgZl5jWoeKnHQSpfgeS0ElzcG8vEyJy9uadRQ..&f=pjson";
    print("$urlR\n");
    curl_setopt($ch, CURLOPT_URL, $urlR);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $exec=curl_exec($ch);
    $results=json_decode($exec,true);

    curl_close($ch);
    var_dump($results);
    $locations=$results["locations"];
    if(!empty($locations)){
        for($i=0;$i<count($locations);$i++){
            $paese=addslashes($locations[$i]["feature"]["attributes"]["Country"]);
            $regione=addslashes($locations[$i]["feature"]["attributes"]["Region"]);
            $provincia=addslashes($locations[$i]["feature"]["attributes"]["Subregion"]);
            $localita=addslashes($locations[$i]["feature"]["attributes"]["Nbrhd"]);
            $citta=addslashes($locations[$i]["feature"]["attributes"]["City"]);
            $nome=addslashes($locations[$i]["feature"]["attributes"]["PlaceName"]);
            $indirizzoEX=addslashes($locations[$i]["feature"]["attributes"]["Place_addr"]);
            $indirizzo=addslashes($locations[$i]["feature"]["attributes"]["StAddr"]);
            $postal=addslashes($locations[$i]["feature"]["attributes"]["Postal"]);
            $lat=$locations[$i]["feature"]["geometry"]["y"];
            $lng=$locations[$i]["feature"]["geometry"]["x"];
            $tel=addslashes($locations[$i]["feature"]["attributes"]["Phone"]);
            $url=addslashes($locations[$i]["feature"]["attributes"]["URL"]);
            $query="INSERT INTO
poiartoscana
(Tipologia,Nazione,Regione,Provincia,Citta,Localita,Nome,IndirizzoEx,Indirizzo,Cap,Latitudine,Longitudine,Telefono,Url) VALUES
('Aeroporti','$paese','$regione','$provincia','$citta','$localita','$nome','$indirizzoEX','$indirizzo','$postal','$lat','$lng','$tel','$url)";
            mysql_query($query);
        }
        echo"FINITO CITTA: ".$filed[$j].$j."\n";
    }
    else{
        echo "Nessun POI trovato per questa citta: ".$filed[$j].$j."\n";
        //exit();
    }
}
echo"FINITO TUTTO";
?>

```

Fig.3: Programma php usato per trasferire i POI trovati con ArcGIS sul server

Le informazioni sulle strutture ricettive della Toscana, infine, provengono dal sito degli open data della Toscana e anch'esse, come quelle sui Punti di Interesse, presentavano una suddivisione dei confini amministrativi datata. Per questo, anch'esse sono state rielaborate con lo stesso programma php usato per i POI. (Vedi Fig.2)

#### 4. Gli strumenti per la resa dei dati

L'API di *Google Maps* mi ha permesso di inserire nell'applicazione una mappa di Google e di usare alcune delle sue funzionalità.

Per mostrare sulla mappa i confini amministrativi dei diversi comuni toscani, ad esempio, ho usato il *Google Maps Data Layer* che permette appunto, di mostrare dati in formato geojson sulla mappa. (Vedi Fig.4)

```
var path={"type":"FeatureCollection","geocoding":{"creation_date":"2016-01-09","generator":{"author":{"name":"Mapzen"},"package":"fences-builder","version":"0.1.2"},"license":"ODbL (see http://www.openstreetmap.org/copyright)"},"features":[{"id":"idcomune","osm_type":"relation","type":"Feature","name":"comune","properties":{"name":"comune"},"geometry":{"type":"MultiPolygon","coordinates":data[comune]}}]};
newMap.data.addGeoJson(path);
newMap.data.setStyle(function (feature) {
return {
fillColor: 'rgb(255,228,225)',
strokeColor: 'rgb(205,0,0)',
strokeWeight: 1,
strokeOpacity:0.5
});
});
newMap.data.setMap(newMap);
```

Fig.4: Porzione del programma javascript che permette di mostrare i confini di un comune sulla mappa

La libreria *InfoBox*, invece, un'estensione dell'API di *Google Maps*, mi ha permesso di modificare la resa grafica delle infoWindow di *Google Maps* usate per la visualizzazione delle informazioni sui POI, ad esempio permettendo di eliminare il bottone di chiusura, di cambiare il colore di sfondo, la grandezza del font, la posizione dell'infoWindow rispetto al marker, la resa del bordo dell'infoWindow.(vedi Fig.5)

```
var infowindow = new InfoBox({
content:contenuto,
closeBoxURL:"",
maxWidth:0,
enableEventPropagation :false,
alignBottom:true
});
```

Fig.5: Porzione del programma javascript che crea un' infoBox

L'oggetto *DirectionService* dell'API non solo mi ha permesso di calcolare la distanza a piedi tra un POI e una struttura ricettiva, ma ha anche restituito il percorso a piedi tra di essi permettendomi di mostrarlo sulla mappa. (Vedi Fig.6)

Altrettanto utile è stata la libreria *jQuery UI* che mi ha fornito gli strumenti per costruire l'*Autocomplete* per la ricerca dei comuni toscani, per trasformare una semplice select html in un più gradevole *Select Menu* e per inserire una finestra (*Dialog*) personalizzabile che si apre ogni volta che si clicca su un POI e mostra alcune informazioni aggiuntive e una serie di

bottoni, ognuno corrispondente a una tipologia di struttura ricettiva che si trova nel raggio di 250 metri da quel POI.

Un notevole aiuto per la realizzazione di alcuni aspetti grafici importanti come il layout e i bottoni mi è stato fornito dalla libreria *Bootstrap*.

Infine, per l'elaborazione del codice css, del codice html e anche per l'uso della libreria *Bootstrap* è stata fondamentale la consultazione del sito *W3CSchools* (<http://www.w3schools.com/>).

```
function calcoladistanza(arrayStr,nextStr,clickedPoi,directionsService,delay,idcomune){
    var origlat=clickedPoi.lat;
    var origlng=clickedPoi.lng;
    var origin=new google.maps.LatLng(origlat,origlng);
    var destin=arrayStr[nextStr];
    var destination=new google.maps.LatLng(destin.structure.lat,destin.structure.lng);
    var options={
        origin:origin,
        destination:destination,
        travelMode:google.maps.TravelMode.WALKING,
        unitSystem:google.maps.UnitSystem.METRIC,
        provideRouteAlternatives:false,
        region:'IT'
    }
    directionsService.route(options,callback);
    function callback(response, status){
        if (status==google.maps.DirectionsStatus.OK){
            var distance=response.routes[0].legs[0].distance.value;
            var startpoint=response.routes[0].legs[0].start_location;
            var endpoint=response.routes[0].legs[0].end_location;
            var startaddress=response.routes[0].legs[0].start_address;
            var endaddress=response.routes[0].legs[0].end_address;
            var route=response.routes[0].overview_path;
            var rbounds=response.routes[0].bounds;
            arrayStr[nextStr]["distancefrom"]=distance;
            arrayStr[nextStr]["startpoint"]=startpoint;
            arrayStr[nextStr]["endpoint"]=endpoint;
            arrayStr[nextStr]["startaddress"]=startaddress;
            arrayStr[nextStr]["endaddress"]=endaddress;
            arrayStr[nextStr]["route"]=route;
            arrayStr[nextStr]["rbounds"]=rbounds;
            nextStr++;
        }else if(status==google.maps.DirectionsStatus.OVER_QUERY_LIMIT){
            delay++;
        }
        theNext(nextStr,arrayStr,clickedPoi,delay,idcomune);
    }
}
```

Fig.6: Porzione del programma javascript che usa DirectionService per calcolare la distanza e estrapolare il percorso