

# *CENTRO MUSICALE*

*Progetto esame per il corso di Progettazione e Laboratorio web*

**Studente: Giorgio Mauro**

**Matricola: 508043**

**Corso di laurea: informatica umanistica**

**anno: 2016/2017**

# INTRODUZIONE

## 1.1 L'idea

L'idea era quella di poter riuscire a fare un'applicazione per tutti gli amanti della musica, e di qualsiasi genere musicale, per cercare la propria band preferita. All'inizio pensavo di poter

riuscire a fare un'applicazione molto più completa con una ricerca dei concerti di una band attraverso una linea temporale, cioè una ricerca attraverso dei trimestri prefissati.

Alla fine ho optato per una ricerca più semplice in modo tale che all'utente restituisca ogni singolo concerto annunciato dalla band.

## 1.2 Mockup del sito

Prima di iniziare, ho progettato il sito su carta in modo tale di trovare la disposizione più corretta all'utente. Ho iniziato a cercare l'immagine più ricorrente alla mia applicazione e attraverso delle ricerche su google maps l'ho trovata e subito dopo ho iniziato a programmare.

# STATO DELL'ARTE

Il campo del web riservato agli eventi da cercare, in particolare, la band richiesta da te. Mi sono ispirato un po' a metalitalia.com che ti mostra la zona dell'evento mostrato da loro.

Il progetto “**Centro Musicale**” s'intende come un sito web per gli eventi, in modo tale che ti possa mostrare tutto attraverso le infowindow nei marker.

Comprende tutte le band possibili contenuti da bandsintown che abbiano comunicato il proprio concerto o tour.

L'utente è messo in condizione da cercare la propria band d'interesse e visualizzare il luogo allargando o restringendo il cerchio inserito nella mappa, portandolo a vedere cosa preferisce di più. Ci sono band che non esistono più, come ad esempio i “Led Zeppelin” e per risolvere questo problema ho deciso di mettere nella box sotto la ricerca un messaggio in cui dice che se il termine è sbagliato, o non ha eventi annunciati, gli eventi non sono stati trovati, così ho potuto presupporre più casi in un unico solo messaggio.

**Centro musicale** è un progetto concentrato molto sulla musica e sull'informazione per le persone di eventi della propria band preferita, cercando di fornire più dettagli possibili.

## NORMALIZZAZIONE DEI DATI

Per la realizzazione di questo progetto sono state necessarie delle tipologie di dati: titolo, latitudine, longitudine, data e giorno, città, regione e luogo.

Questi dati sono stati pubblici, e possono essere presi da chiunque, basta fare la chiamata delle API e verranno restituite. Sono stati presi da <https://app.swaggerhub.com/api/Bandsintown/PublicAPI/3.0.0> che mi ha permesso di estrarre ed estrapolare i dati. Per fare questo ho fatto scritto un programma in php.

Ad ogni marker che si visualizzerà nella mappa, cliccandoci, spunterà un infowindow con tutte le informazioni necessarie. Qui sotto riporto lo screenshot del codice in php scritto per questo progetto e per la connessione al database ho fatto un altro foglio in notepad includendolo al codice di php.

```

//Se sono maggiori di 0 (il termine è già stato cercato);
if($contatore>0){

    //rifaccio la query;
    $valoridb="SELECT * FROM 'eventi' WHERE 'titolo' LIKE '$valoretesto'";
    $eseguirecupo=mysqli_query($connessione,$valoridb);

    //mi creo una stringa JSON con i dati del mio database;
    $jsonstringa="[";

    while($row=mysqli_fetch_array($eseguirecupo, MYSQLI_ASSOC)){ //per ogni riga prendo le info che mi servono
        $jsonstringa= $jsonstringa."{
            \"id\":".".$row['id'].",
            \"titolo\":".".$row['titolo'].",
            \"lat\":".".$row['lat'].",
            \"lng\":".".$row['lng'].",
            \"luogo\":".".$row['luogo'].",
            \"citta\":".".$row['citta'].",
            \"regione\":".".$row['regione'].",
            \"data\":".".$row['data'].",
            \"data_f\":".".$row['data_f'].",
        },";
    };
    $jsonstringa=$jsonstringa."]";
    $json_completo= str_replace(",",""),"",$jsonstringa);
    echo("$json_completo");
}

//Caso in cui il termine cercato non sia presente nel database;
date:

```

L'immagine riportata sopra è il caso in cui ci sia qualcosa nel database e va a prendere le informazioni...

```

$dati = file_get_contents("http://api.bandsintown.com/artists/".$valoretesto."/events.json?api_version=2.0&app_id=YOUR_APP_ID"); //richiesta alle API;
$dati=json_decode($dati); // decodifica dei dati ricevuti;

//Inserisco i dati ottenuti nel mio database;

foreach($dati as $value){
    $latitudine=$value->venue->latitude;
    $longitudine=$value->venue->longitude;
    $luogo=$value->venue->place;
    $citta=$value->venue->city;
    $regione=$value->venue->region;
    $query="INSERT INTO eventi VALUES('$value->id','$value->title','$latitudine','$longitudine','$luogo','$citta','$regione','$value->datetime','$value->formatted_datetime')";
    $eseguire=mysqli_query($connessione,$query);
    if (!$eseguire){
        die(mysqli_error($connessione));
    }
}

//Selezione i dati appena inseriti e li passo a JS;
$query_take_lat="SELECT * FROM 'eventi' WHERE 'titolo' LIKE '$valoretesto'";
$eseguire_due=mysqli_query($connessione,$query_take_lat);
$json="["; //creo la stringa JSON con i dati del database;

while($row=mysqli_fetch_array($eseguire_due, MYSQLI_ASSOC)){
    $json= $json."{
        \"id\":".".$row['id'].",
        \"titolo\":".".$row['titolo'].",
        \"lat\":".".$row['lat'].",
        \"lng\":".".$row['lng'].",
        \"luogo\":".".$row['luogo'].",
        \"citta\":".".$row['citta'].",
        \"regione\":".".$row['regione'].",
        \"data\":".".$row['data'].",
        \"data_f\":".".$row['data_f'].",
    },";
}
$json=$json."]";
$json_completo= str_replace(",",""),"",$json);
echo($json_completo);

```

... E in questo ELSE(immagine riportata qui sopra) inserisce i dati nel database, nel caso in cui il database sia vuoto, e crea la stringa JSON con i dati nel database.

## Connessione al DataBase

La connessione al database è stato fatto attraverso un altro foglio in notepad ed è stato collegato all'altro foglio in php.

```
<?php

//connessione al database;
$nomehost = "localhost";
$nomeuser = "root";
$connessione = mysqli_connect($nomehost,$nomeuser,'',"centro_musicale");

?>
```

## LA RESTITUZIONE DEI DATI

Grazie alle API di *Google Maps* sono riuscito a mettere la mappa sulla mia applicazione e usarla per i miei obiettivi, raggiunti come creare il cerchio, spostare il cerchio al click

```
//creo la mappa;
var map = new google.maps.Map(document.getElementById('mappa'), {
  center: {lat: 41.895676, lng: 12.496850},
  scrollwheel: true,
  zoom: 2
});

//creo il contenitore per le info dei marker;
var infowindow = new google.maps.InfoWindow({
  content: ""
});

//creo il cerchio all'avvio;
var circle = new google.maps.Circle({
  map: map,
  radius: 2000000,
  fillColor: '#AA0000',
  draggable:true,
  editable:true,
  center: new google.maps.LatLng(41.900569, 12.488787)
});

//al click su mappa sposto il cerchio nel punto cliccato;
google.maps.event.addListener(map, 'click', function(e) {
  circle.setMap(null);
  circle = new google.maps.Circle({
    map: map,
    radius: 2000000,
    fillColor: '#AA0000',
    draggable:true,
    editable:true,
    center: new google.maps.LatLng(e.latLng.lat(), e.latLng.lng())
  });
});
```

Per i marker e per le infowindow ho creato una funzione apposita dove i marker dentro il cerchio li colora di verde, che sono quelli più accessibili a te, e quelli fuori vengono colorati di rosso.

Giustamente puoi modificare il raggio del cerchio per indicare la zona a te accessibile. Nell'immagine di sotto mostrerò la parte di codice per creare i marker e le infowindow.

```
function CreaMarker(map,obj,collezione,circle, infowindow){
    function bindInfoWindow(marker, map, infowindow, description) { //funzione per settare il contenuto delle infowindow per ogni marker;
        marker.addListener('click', function() {
            infowindow.setContent(description);
            infowindow.open(map, this);
        });
    }
    var circleLat= circle.getCenter().lat(); //prendo i valori di raggio e coordinate del cerchio per il confronto con i punti degli eventi;
    var circleLng= circle.getCenter().lng();
    var radius = circle.getRadius();

    for(i in obj){ //per ogni evento;
        var distanzapuntodacerchio= getDistance(obj[i].lat,obj[i].lng,circleLat,circleLng); //controllo la distanza tra l'evento e il centro del cerchio;
        var stringa = "<b>Nome Evento:"+" "+obj[i].titolo+"</b></br><p>Luogo:"+" "+obj[i].luogo+"<br>Citt&agrave;:"+" "+obj[i].citta+"<br>Regione:"+" "+obj[i].regione+"<br>Data e O
        if((distanzapuntodacerchio<radius)){ // se i punti sono dentro il raggio del cerchio;
            var markers = new google.maps.Marker({
                position: new google.maps.LatLng(obj[i].lat, obj[i].lng),
                map: map,
                icon:'http://maps.google.com/mapfiles/ms/icons/green-dot.png' // li stampi in verde;
            });
            collezione.push(markers); //metto i punti dentro il contenitore "collezione"
            bindInfoWindow(markers, map, infowindow, stringa); // collego ogni marker all'infowindow corrispondente;
        }else{ // se i punti sono fuori dal cerchio;
            var markers = new google.maps.Marker({
                position: new google.maps.LatLng(obj[i].lat, obj[i].lng),
                map: map,
                icon:'http://maps.google.com/mapfiles/ms/icons/red-dot.png' //li stampi in rosso
            });
            collezione.push(markers); //metto i punti dentro il contenitore "collezione"
            bindInfoWindow(markers, map, infowindow, stringa);
            $("#messaggio").html("ATTENZIONE: Sono presenti altri eventi fuori dalla zona selezionata"); //avviso in cui siano presenti eventi anche fuori dalla zona selezionata;
        }
    }
}
```

Var stringa non si legge molto bene e ve la riporto qui sotto

```
var stringa = "<b>Nome Evento:"+" "+obj[i].titolo+"</b></br><p>Luogo:"+"
"+obj[i].luogo+"<br>Citt&agrave;:"+" "+obj[i].citta+"<br>Regione:"+"
"+obj[i].regione+"<br>Data e Ora:"+" "+obj[i].data_f+"</p>"
```

**VAR STRINGA** prende il contenuto da visualizzare nelle infowindow.

Utile è stata la libreria Geometry, messa all'interno dell'HTML (<script src="http://maps.googleapis.com/maps/api/js?key=AIzaSyCV5ZUMji1GO4uEfabSnHkdKUKjJW0mXTU&libraries=geometry"></script>), che mi ha permesso di calcolare la latitudine e la longitudine per metterli all'interno del raggio del cerchio.

```
var rad = function(x) { //funzione matematica per il calcolo della distanza;
    return x * Math.PI / 180;
};

// funzione per la distanza tra due punti; (CENTRO DEL CERCHIO E PUNTO DELL'EVENTO)
var getDistance = function(latp, lngp, latc, lngc) {
    var R = 6378137; // raggio della terra;
    var dLat = rad(latc - latp);
    var dLong = rad(lngc - lngp);
    var a = Math.sin(dLat / 2) * Math.sin(dLat / 2) +
    Math.cos(rad(latp)) * Math.cos(rad(latc)) *
    Math.sin(dLong / 2) * Math.sin(dLong / 2);
    var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
    var d = R * c;
    return d; //restituisce la distanza;
};
```

## **TECNOLOGIE USATE**

Per creare il sito ho usufruito di alcune tecnologie web quali:

- HTML;
- CSS;
- JS;
- JQuery;
- PHP;
- MySql;

Un grande aiuto è stato consultare W3C per il CSS e per l'ajax (qui sotto viene riportata l'immagine per l'avvio alla ricerca dei valori inseriti).

(<http://www.w3schools.com/>).

Per il **PHP** mi è stato molto utile consultare il sito e le slide del professore.

(<http://php.net/manual/it/>)

```
//avvio della funzione alla ricerca;
$("#bottone").click(function(){
    deleteMarkers(); //cancello i marker presenti nell'array;
    var valoretesto = $("#testarea").val(); // prendo il valore inserito dall'utente;
    if(valoretesto==""){ //se l'utente lascia bianco;
        $("#messaggio").html("<p> Assicurati di scrivere qualcosa nella casella di testo</p>");
    }else{ //se l'utente scrive allora esegui la chiamata ajax e spedisce il valore scritto al php;
        $("#messaggio").html("<p> </p>");
        $.ajax({
            type: 'GET',
            url: './php/dati.php',
            contentType: "application/json",
            dataType: 'text',
            data: {'valoretesto': valoretesto},
            success: function(resp){ //se la chiamata ha successo...
                try{
                    var obj= JSON.parse(resp); //prova a farmi il parse della stringa JSON
                    if (obj.length==0){ // se non ci sono concerti per l'artista cercato;
                        $("#messaggio").html("Non sono stati trovati eventi per i termini ricercati");
                    }
                }catch(e){ //Nel caso il parse dia errore per termini sbagliati inseriti;
                    e=$("#messaggio").html("Non sono stati trovati eventi per i termini ricercati");
                }
                CreaMarker(map,obj,collezione,circle,infoWindow);
            },
            error: function(resp){ //in caso di errore della chiamata;
                console.log(resp);
            }
        });
    }
});
```