

Corso di Tecnologie e Linguaggi per il Web

Anno 2021-2022

MyMeteo

Andrea Fumagalli, 941668

## **1 Analisi dei requisiti**

### **1.1 Destinatari**

Il progetto si propone di fornire una app semplice e intuitiva per informarsi sul meteo locale e globale, è quindi diretta a chiunque abbia questo interesse e fornisce la possibilità di salvare dei preferiti per gli utilizzatori più interessati e assidui.

L'app non richiede perciò nessuna conoscenza pregressa ed è disponibile per ogni tipo di device grazie alla struttura responsiva.

### **1.2 Modello di valore**

Il punto di forza dell'app è la scalabilità sulla base dell'interesse dell'utilizzatore: ne può essere fatto un uso basilare consultando le previsioni locali, ma si può anche ricercare il meteo attuale di una qualsiasi città e monitorarlo nei giorni successivi o ancora vedere le previsioni di qualsiasi città nel mondo.

L'app fornisce anche una dark mode

### **1.3 Flusso dei dati**

I dati riguardanti il meteo vengono ottenuti attraverso l'API OpenWeather in formato JSON e i dati significativi vengono estratti e forniti all'utente.

Le preferenze dell'utente vengono salvate in LocalStorage.

### **1.4 Aspetti tecnologici**

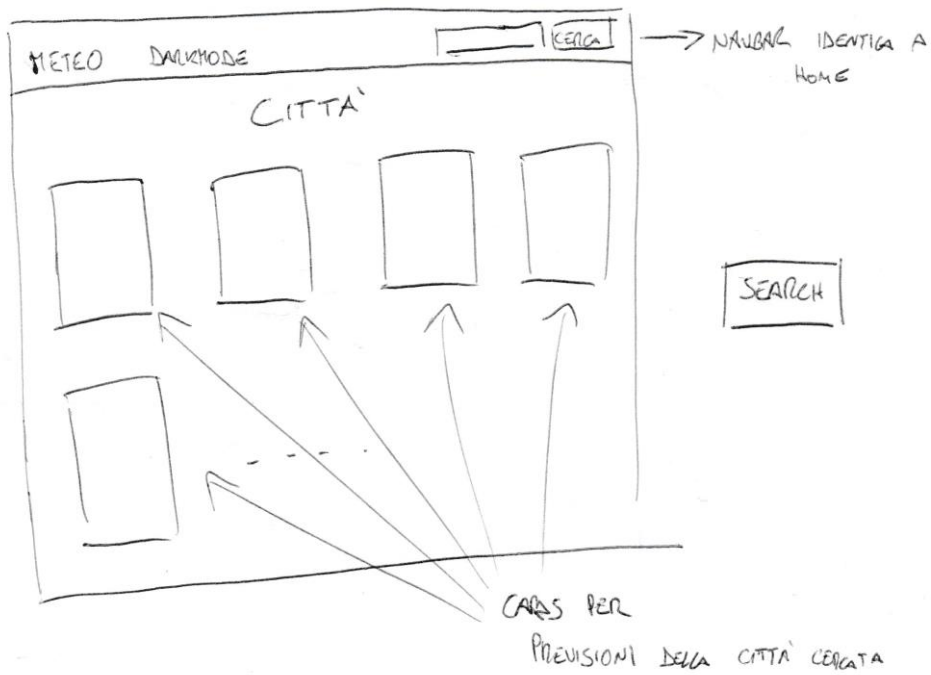
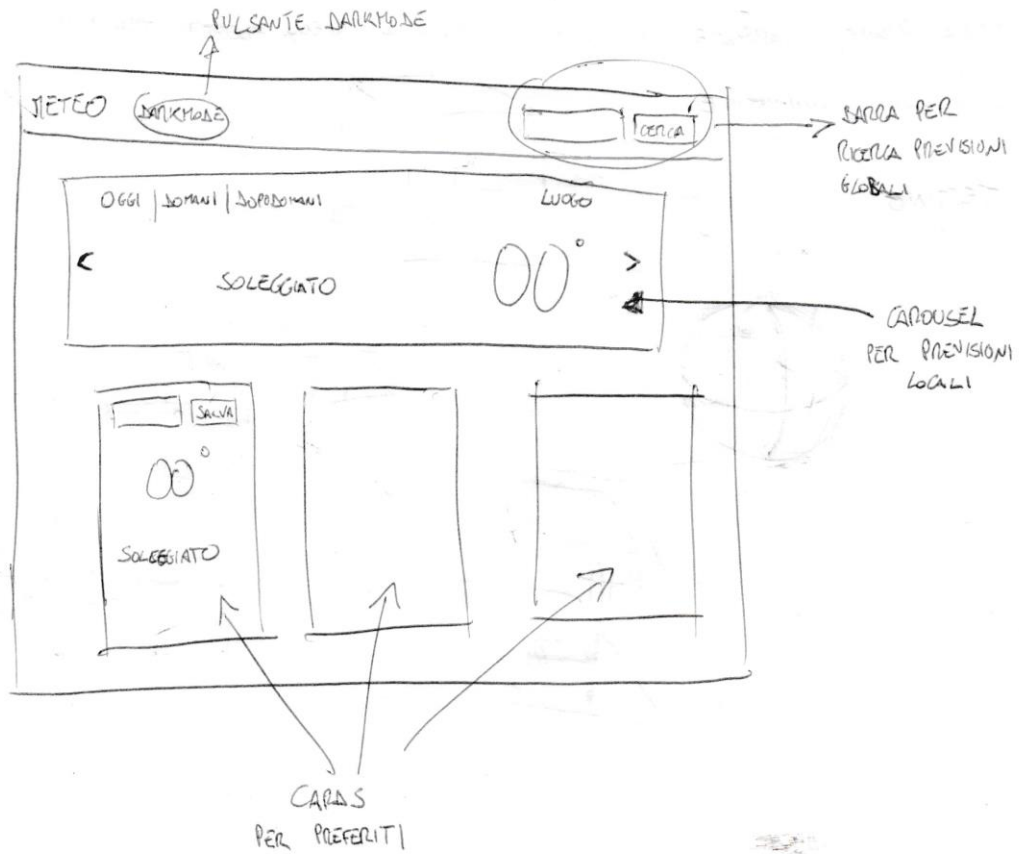
L'interfaccia si basa su un template Bootstrap responsivo e fa uso del plugin Carousel.

L'app si basa su un framework Node.js che utilizza il modulo express per la gestione del server. Vengono utilizzati anche i moduli dotenv, path, ejs e axios.

Le chiamate XMLHttpRequest sono eseguite con approccio AJAX utilizzando il modulo axios e scambiano dati in formato JSON.

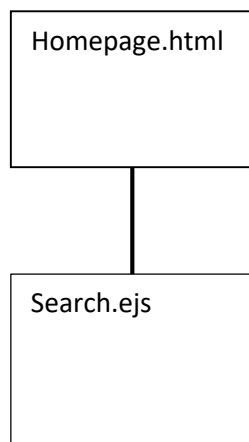
## 2 Interfacce

Homepage

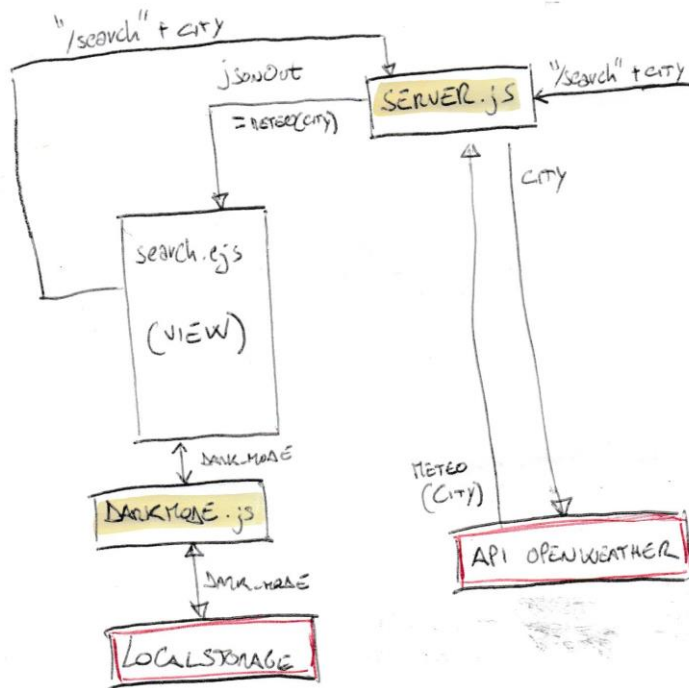
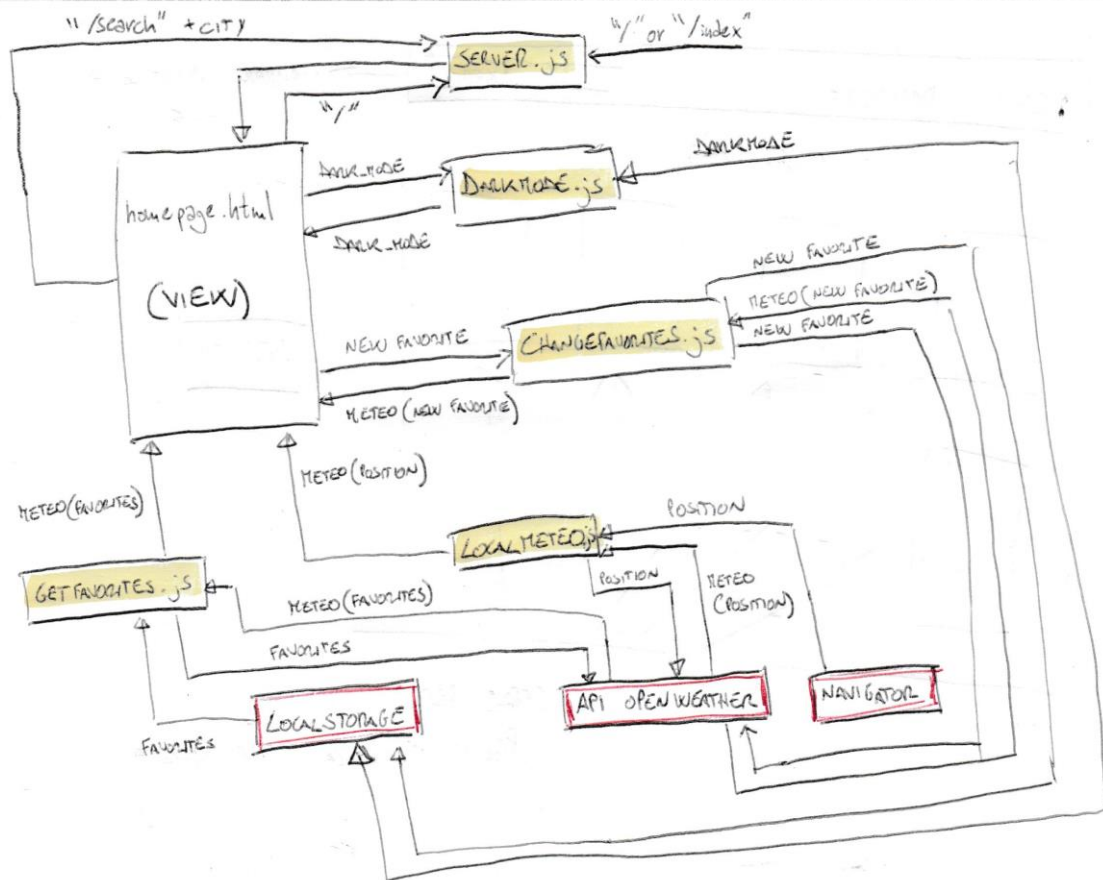


### 3 Architettura

#### 3.1 Diagramma dell'ordine gerarchico delle risorse



### 3.2 Descrizione delle risorse



## 4. Codice

### 4.1 HTML

Search form

```
<form class="d-flex" id="search-form" method="post" enctype="application/x-www-form-urlencoded" action="/search">
  <input id="search-text" name="city" class="form-control me-2" type="search" placeholder="Cerca" aria-label="Search">
  <button id="search-button" class="btn btn-outline-success" type="submit">Cerca</button>
</form>
```

### 4.2 CSS

Regola principale dark mode

```
.dark-mode {
  background-color: #343a40;
  color: white;
}
```

### 4.3 API

Dark mode

```
function darkmode(){
  var body = document.body;
  body.classList.toggle("dark-mode");
  var x = localStorage.getItem('dark-mode');
  x = Math.abs(x-1);
  localStorage.setItem("dark-mode", x);
};

const darkModeSwitch = document.getElementById("dark-mode-switch");
darkModeSwitch.addEventListener("click", darkmode);

document.addEventListener('onload', readDarkModeStorage());
function readDarkModeStorage(){
  var x = localStorage.getItem('dark-mode');
  if(x == null){
    localStorage.setItem("dark-mode", 0);
  }else{
    if(x == 1){
      var body = document.body;
      body.classList.toggle("dark-mode");
    }
  }
};
```

## Local meteo

```
const API_KEY = '893659f8e4f5dfbb6499a4c2b0e8ad50'

var request = async url => {
  const response = await fetch(url);
  return response.ok ? response.json() : Promise.reject({ error: 500 });
};

const getWeatherInfo = async (latitude, longitude) => {
  try {
    const url = `https://api.openweathermap.org/data/2.5/onecall?lat=${latitude}&lon=${longitude}&exclude=minutely,hourly&units=metric&lang=it&appid=${API_KEY}`;
    const response = await request(url);
    document.getElementById("today-degrees").textContent = Math.round(response.daily[0].temp.day) + "°C";
    document.getElementById("today-description").textContent = response.daily[0].weather[0].description;
    document.getElementById("today-img").src = `http://openweathermap.org/img/wn/${response.daily[0].weather[0].icon}@2x.png`

    document.getElementById("tomorrow-degrees").textContent = Math.round(response.daily[1].temp.day) + "°C";
    document.getElementById("tomorrow-description").textContent = response.daily[1].weather[0].description;
    document.getElementById("tomorrow-img").src = `http://openweathermap.org/img/wn/${response.daily[1].weather[0].icon}@2x.png`

    document.getElementById("day-after-degrees").textContent = Math.round(response.daily[2].temp.day) + "°C";
    document.getElementById("day-after-description").textContent = response.daily[2].weather[0].description;
    document.getElementById("day-after-img").src = `http://openweathermap.org/img/wn/${response.daily[2].weather[0].icon}@2x.png`
  } catch (err) {
    console.error(err);
  }
};

document.addEventListener('DOMContentLoaded', () => {
  navigator.geolocation.getCurrentPosition(position => {
    const { latitude, longitude } = position.coords;
    getWeatherInfo(latitude, longitude);
  });
});
```

## Get favorites

```
const cities_std = ["Londra", "Tokyo", "Roma", "Berlino"];

var request = async url => {
  const response = await fetch(url);
  return response.ok ? response.json() : Promise.reject({ error: 500 });
};

const getWeather = async (id, city_name) => {
  try {
    const url = `https://api.openweathermap.org/data/2.5/weather?q=${city_name}&units=metric&lang=it&appid=${API_KEY}`;
    const response = await request(url);
    document.getElementById("pref"+id+"-city").textContent = response.name + ", " + response.sys.country;
    document.getElementById("pref"+id+"-temp").textContent = Math.round(response.main.temp) + "°C";
    document.getElementById("pref"+id+"-info").textContent = response.weather[0].description;
    document.getElementById("pref"+id+"-icon").src = `http://openweathermap.org/img/wn/${response.weather[0].icon}@2x.png`
  } catch (err) {
    console.error(err);
  }
};

document.addEventListener('DOMContentLoaded', () => {
  var cities = new Array();
  for (let index = 0; index < 4; index++) {
    var city = localStorage.getItem("pref"+index);
    if(city == null){
      city = cities_std[index];
    }
    cities.push(city);
  };
  for (let index = 0; index < 4; index++) {
    getWeather(index, cities[index])
  };
});
```



## Change favorites

```
var request = async url => {
  const response = await fetch(url);
  return response.ok ? response.json() : Promise.reject({ error: 500 });
};

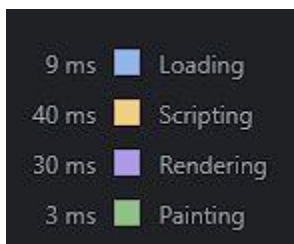
const btn0 = document.getElementById("pref0-btn");
btn0.addEventListener("click", changeFavorite);
const btn1 = document.getElementById("pref1-btn");
btn1.addEventListener("click", changeFavorite);
const btn2 = document.getElementById("pref2-btn");
btn2.addEventListener("click", changeFavorite);
const btn3 = document.getElementById("pref3-btn");
btn3.addEventListener("click", changeFavorite);

function changeFavorite(event){
  var x = event.target.id.substr(4,1);
  var newCity = document.getElementById("pref"+x+"-text").value;
  document.getElementById("pref"+x+"-text").value='';
  if(newCity !== null){
    getOneWeather(newCity,x);
  }
};

const getOneWeather = async (city_name,x) => {
  try {
    const url = `https://api.openweathermap.org/data/2.5/weather?q=${city_name}&units=metric&lang=it&appid=${API_KEY}`;
    const response = await request(url);
    localStorage.setItem("pref"+x, city_name);
    document.getElementById("pref"+x+"-city").textContent = response.name + ", " + response.sys.country;
    document.getElementById("pref"+x+"-temp").textContent = Math.round(response.main.temp) + "°C";
    document.getElementById("pref"+x+"-info").textContent = response.weather[0].description;
    document.getElementById("pref"+x+"-icon").src = `http://openweathermap.org/img/wn/${response.weather[0].icon}@2x.png`;
  } catch (err) {
    console.error(err);
  }
};
```

## 5. Performance

Load homepage



Search

