

# PERCORSO PER LE COMPETENZE TRASVERSALI E L'ORIENTAMENTO



**MIIDM**

La storia del passato, attraverso le tecnologie del futuro

*Giuseppe Bocci*  
*5Ai*  
*a.s. 2019/2020*

1° anno – 2017/2018

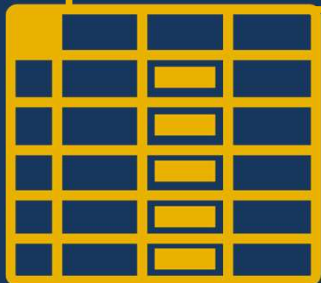


*Sicurezza*

*Visita e studio  
dei musei*



*Studio delle tecniche  
di catalogazione  
ed esposizione*



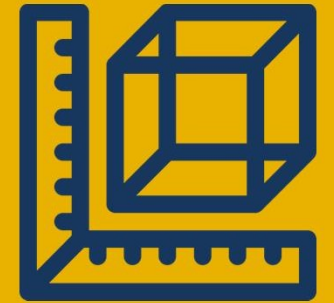
*Studio dei nostri reperti  
e catalogazione*



2° anno – 2018/2019







# IMPLEMENTAZIONE SOFTWARE

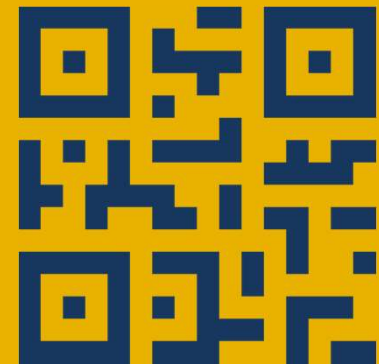
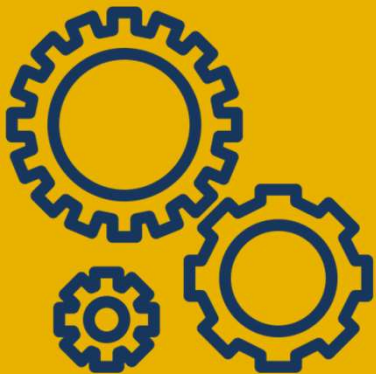
3° anno – 2019/2020

*Tutor interno: Prof. Michelangelo De Bonis*

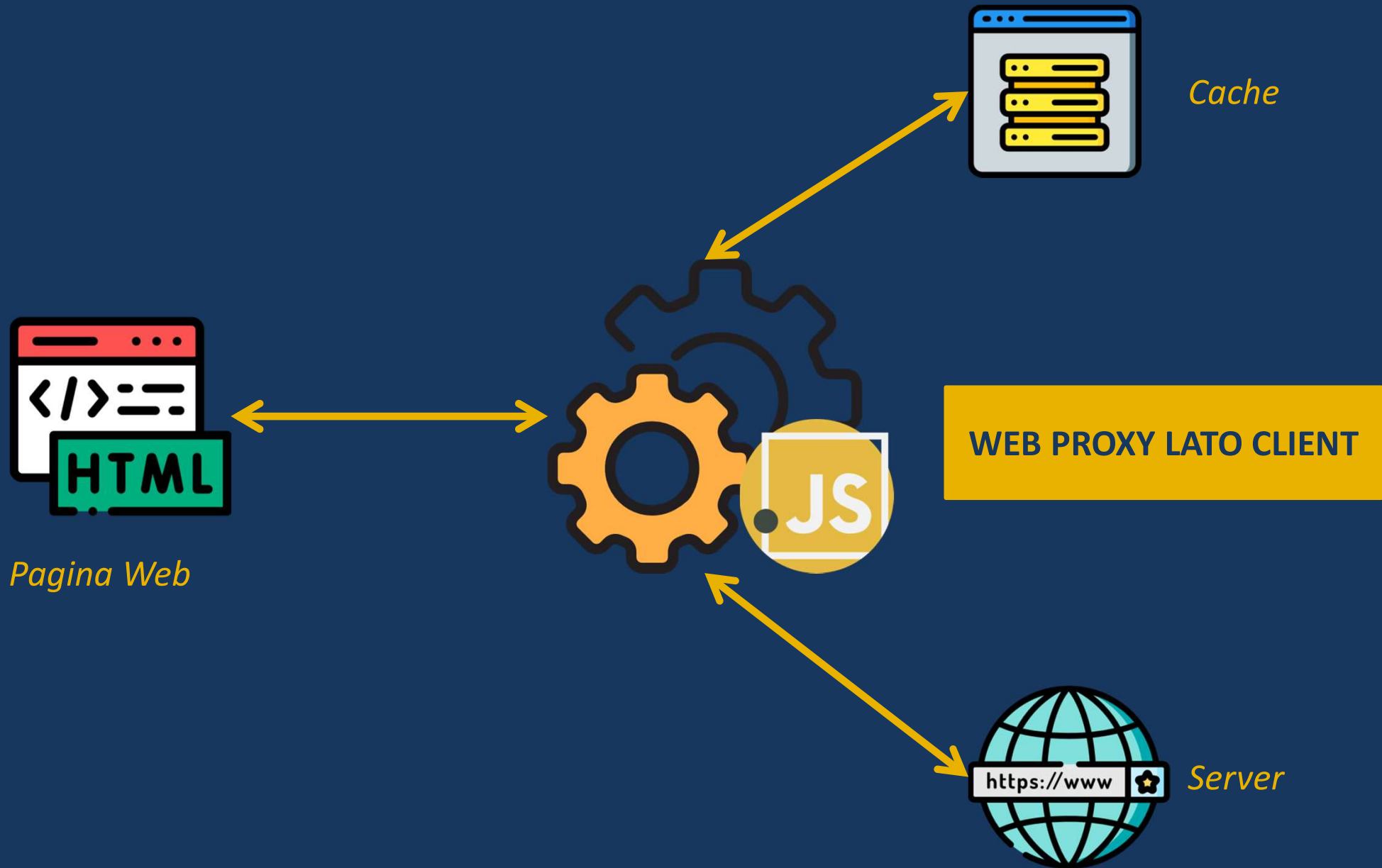
*Tutor esterno: Dott. Alfredo De Biase*



# IL MIO RUOLO

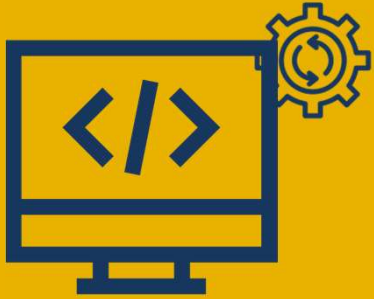


# SERVICE WORKER





# SERVICE WORKER



*Tecnologia ancora in sviluppo*

*Gestione asincrona in background*



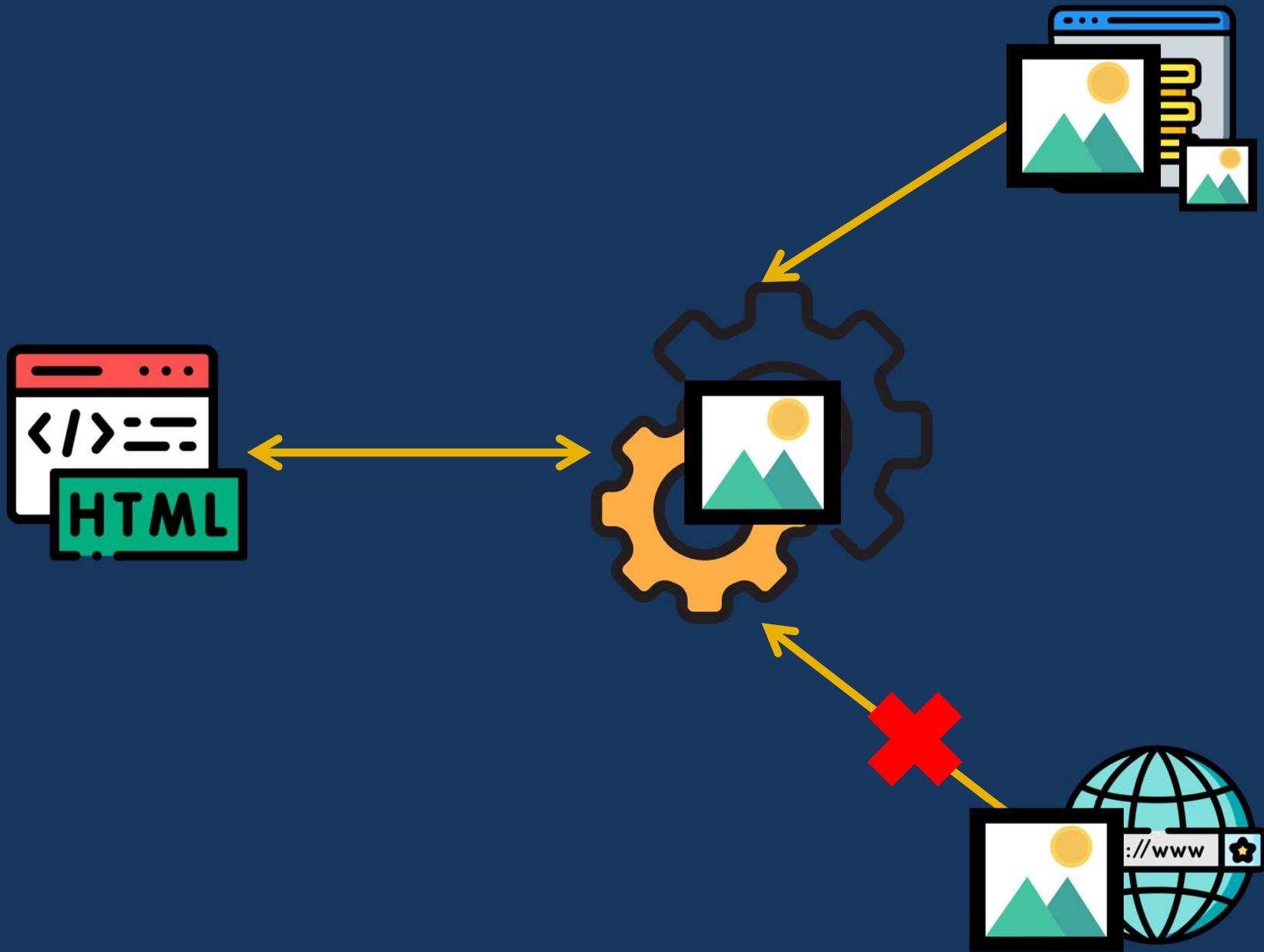
*Notifiche push*

*Geofencing*





# NETWORK THAN CACHE

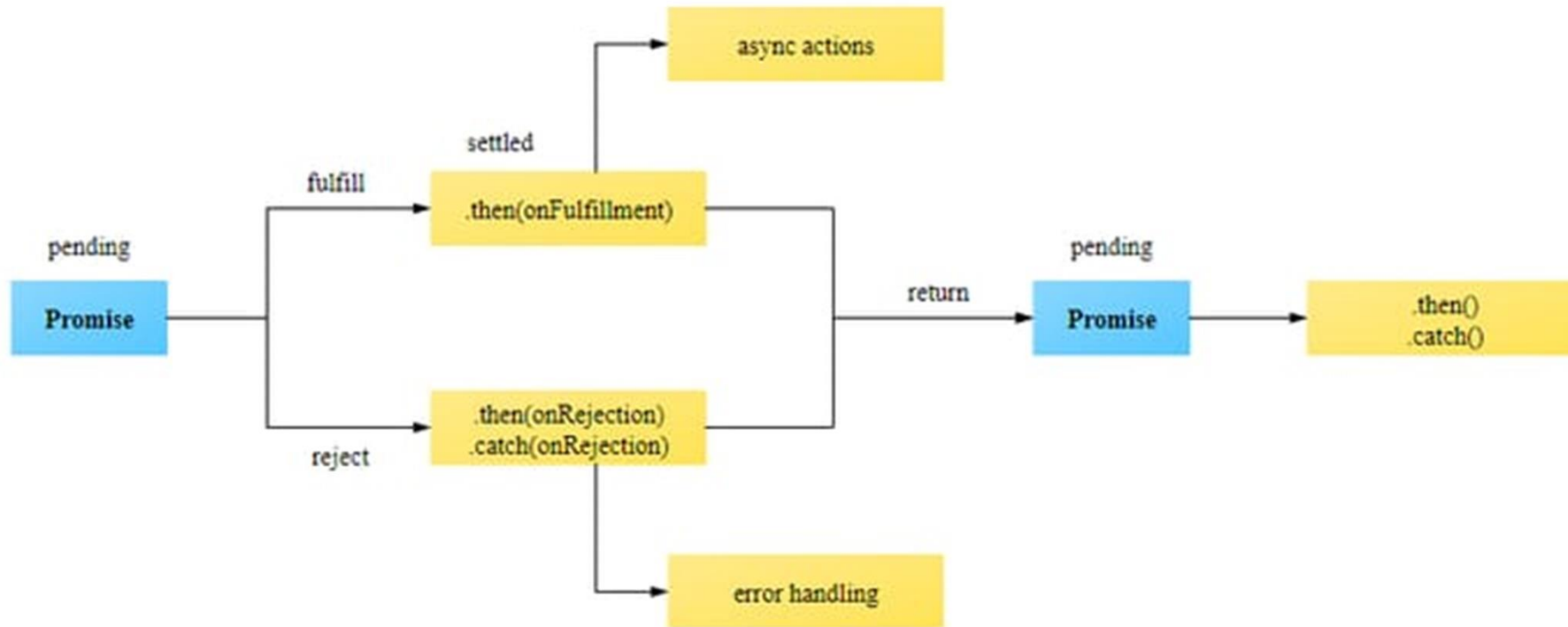


# INSTALLAZIONE SERVICE WORKER

```
1  if ('serviceWorker' in navigator) {  
2    // Path che contiene il service worker  
3    navigator.serviceWorker.register('sw.js').then(function(registration) {  
4      console.log('Service worker installato correttamente, ecco lo scope:', registration.scope);  
5    }).catch(function(error) {  
6      console.log('Installazione service worker fallita:', error);  
7    });  
8  }
```

```
// Evento fetch
self.addEventListener('fetch', event => {
  console.log("Richiesta URL: "+event.request.url);
  event.respondWith(
    //provo a fare il fetch della richiesta
    fetch(event.request).then(response => {
      //se va a buon fine:
      //1) faccio il caching della request
      var r1 = response.clone();
      caches.open(cacheName('attuale', config)).then(cache => {
        cache.put(event.request, r1);
      });
      //2) la Promise va a fulfillment con il return della response
      return response;
    }).catch(() => {
      //se il fetch ha fallito la Promise sarà Pending
      //provo a fare un match nella mia cache
      return caches.open(cacheName('attuale', config)).then(cache => {
        var cr= cache.match(event.request);
        console.log("cache.match:",cr);
        return cr;
      }).catch( error => {
        //se il match nella cache fallisce genererà un errore e la Promise sarà rejected
        console.error("Fetch error:",error);
        error.reject();
      });
    });
  );
});
```

# PROMISE





**GRAZIE PER L'ATTENZIONE**