**Classroom Scanner AR**

**Titolo progetto:** Classroom Scanner AR

**Alunno/i:** Julian Sprugasci, Lorenzo Piazza,

Pierpaolo Casati e Claudio Engeler

**Classe:**  Info 3

**Anno scolastico:** 2019/2020

**Responsabile:** Geo Petrini

1. **Introduzione**
   1. **Informazioni sul progetto**

Allievi coinvolti: Julian Sprugasci, Lorenzo Piazza, Pierpaolo Casati e Claudio Engeler

Classe: Informatica 3AC presso la SAM Trevano

Docente responsabile: Geo Petrini

Data inizio: 2019/01/17

Data fine: definire

* 1. **Abstract**

*A school needs a system capable of managing lesson times and various classrooms. This project aims to create an augmented reality system, which through the QR codes present in the classrooms will make the time of day of that single classroom appear to improve research and facilitate people. It will also be possible to search for a classroom through a GPS function. To do this we will use the notions learned in our school using the following languages:*

*sql, php, css, js and html.*

* 1. **Scopo**

Questo progetto ha lo scopo didattico principale di imparare a gestire in team un progetto IT e di prepararci per l’esame di fine tirocinio. Per fare ciò dobbiamo ricorrere a tutte nozioni apprese durante la nostra formazione, utilizzando molti aspetti visti in varie materie, come per esempio la creazione e gestione di un Gantt oppure l’utilizzo di vari linguaggi di programmazione. L'altro scopo di questo progetto sarà quello di creare un applicativo che permetta di gestire l’orario scolastico della nostra sezione attraverso la realtà aumentata.

1. **Analisi**
   1. **Analisi del dominio**

Il docente supervisore che ci ha stipulato le specifiche del progetto sono dalla parte del cliente e ci ha assegnato un diario dei compiti. Il risultato finale, come spiegato nelle specifiche dovrà essere un applicativo web che dovrà gestire gli orari della nostra sezione e dare la possibilità alle persone di guardare l’orario in maniera immediata e più semplice attraverso la loro telecamera del telefono. L’applicativo sarà implementato in html, css, js e python. Il tutto sarà hostato su un web server di Github e quindi sarà facilmente accessibile da tutti che hanno un sito web adeguato (Firefox, Chrome e Safari).

Questo prodotto non esiste ancora sul mercato ma non verrà progettato a scopo di lucro ma solamente a scopo didattico per consolidare le nostre nozioni. Ogni piano della scuola avrà un codice QR che corrisponderà al piano corrente. Poi in ogni aula del piano sarà presente un codice QR con un pattern univoco per distinguerle, che una volta scannerizzato mostrerà a schermo l’orario della lezione in corso. Mentre se si volessero avere più informazioni riguardanti quell’aula si potrà premere sul popup a schermo che porterà direttamente ad una pagina web con tutte le informazioni necessarie. Tutte le specifiche e la guida su come utilizzare il prodotto sarà presente sul nostro sito.

* 1. **Analisi e specifica dei requisiti**

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Realizzare un sito web per l’applicativo |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Categoria** | Linguaggio |
| **Sotto Requisiti** | |
| **001** | Il sito deve essere adatto per qualsiasi dispositivo |
| **002** | Deve esserci una guida utente |
| **003** | Il sito dovrà essere indipendente |
| **004** | L’interfaccia e i colori di sfondo dovranno essere adatti a qualsiasi utente |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Dovranno essere presenti dei QR Code per ogni piano |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Categoria** | Sistema |
| **Sotto Requisiti** | |
| **001** | Ogni QR Code corrisponde ad un colore per identificare il piano attuale |
| **002** | Ogni QR Code dovrà portare direttamente al nostro sito web |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Ogni aula dovrà avere un pattern univoco per essere riconosciuta |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Categoria** | Sistema |
| **Sotto Requisiti** | |
| **001** | Il pattern non dovrà avere delle forme complesse |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | L’applicativo dovrà avere la funzione di lettura dei pattern |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Categoria** | Sistema e Linguaggio |
| **Sotto Requisiti** | |
| **001** | Lo scanner dovrà essere eseguito attraverso la fotocamera del dispositivo |
| **002** | Dovrà essere in grado di riconoscere tutti i pattern assegnati alle aule |
| **003** | Una volta proiettato il risultato dovrà essere possibile cliccare sul popup per mostrare delle informazioni in più riguardanti l’orario |
| **004** | L’interfaccia e i colori di sfondo dovranno essere adatti a qualsiasi utente |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | L’applicativo dovrà mostrare gli orari in AR |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Categoria** | Linguaggio |
| **Sotto Requisiti** | |
| **001** | Bisogna creare un finestra popup uguale per tutte le aule |
| **002** | Dovrà essere possibile leggere il database e estrarre l’orario corrente |
| **003** | Dovrà esserci la funzione “Maggiori informazioni” |
| **004** | Bisogna mostrare gli orari da qualsiasi angolazione si legge il QR Code |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | L’applicativo dovrà avere un database con tutti gli orari |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Categoria** | Linguaggio e Sistema |
| **Sotto Requisiti** | |
| **001** | Il database dovrà essere aggiornato una volta ogni anno |
| **002** | Il database deve contenere tutti i dati riguardanti gli orari |
| **003** | Il database dovrà essere sempre attivo durante i giorni lavorati |
| **004** | Il database sarà sviluppato il MySQL e hostato su un server |

|  |  |
| --- | --- |
| **ID: REQ-007** | |
| **Nome** | Bisogna creare uno script in grado di estrarre i dati dal sito dell’orario |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Categoria** | Linguaggio |
| **Sotto Requisiti** | |
| **001** | Lo script verrà sviluppato il Python e Flasc |
| **002** | Dovrà essere in grado di estrarre i dati e inserirli nel database |
| **003** | Dovrà essere eseguito una volta all’anno per aggiornare i dati |

|  |  |
| --- | --- |
| **ID: REQ-008** | |
| **Nome** | L’applicativo dovrà avere un sistema per ricercare le aule |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Categoria** | Linguaggio |
| **Sotto Requisiti** | |
| **001** | Dovrà avere la possibilità di avere un filtro per la ricerca(docente, aula o classe) |
| **002** | La funzione dovrà tenere conto della posizione corrente dell’utente |
| **003** | La funzione dovrà mostrare a schermo o indicativamente il percorso da svolgere per arrivare alla destinazione |
| **004** | La funzione dovrà funzionare tramite il gps integrato del dispositivo senza dover installare niente di esterno. |

* 1. **Use case**



* 1. **Pianificazione**

Immagine che contiene screenshot, testo

Descrizione generata automaticamente

* 1. **Analisi dei mezzi**
     1. **Software**

Per la realizzazione del progetto abbiamo utilizzato le seguenti librerie:

* **AR.js – 2.2.2:** Questa libreria l’abbiamo utilizzata per la realtà aumentata, precisamente per leggere i codici QR in modo da visualizzare gli orari attraverso degli oggetti virtuali.
* **Plugin Firefox Selenium IDE – 3.16.1:** Questo programma l’abbiamo utilizzato per ricavare i dati dell’orario scolastico attraverso il sito della scuola. Selenium permette di registrare delle interazioni con dei siti web per aiutare a generare e mantenere l’automazione del sito, facendo dei test.
* **Bootstrap - 4.4.1:** Questa strumento è stato utilizzato per tutto quello che concerne la parte grafica del nostro sito web, dando a disposizione degli strumenti veramente performanti e puliti con un’ottima documentazione intuitiva.
* **Selenium webdriver - 3.141.59:** Permette di definire quale driver browser dovrà essere utilizzato per eseguire i test e le varie operazioni sul sito in modo automatizzato. Il web browser che utilizziamo è Chrome.
* **BeautifulSoup – 4.4.0:** Permette di ricavare i dati dal sito. Precisamente è un parser di linguaggi xml e html.
* **Flask – 1.1.1:** L’abbiamo utilizzato per visualizzare il risultato, praticamente permette di creare un web framework.
* **Re:** libreria per le espressioni regolari
* **Time:** libreria che contiene dei metodi per il tempo.
* **Datetime**: libreria che contiene dei metodi per le date.
* **Selenium**: Package di selenium che contiene dverse librerie.
* **Webdriver**: Libreria che permette di determinare il browser da utilizzare per realizzare i test.
* **Options**: Libreria che contiene diverse opzioni per webdriver.
* **Bs4**: Package che contiene diversi metodi della libreria beautiful soup.
* **Beautiful soup:** Libreria che permette di parsare del codice HTML.
  + 1. **Hardware**

Visto che il nostro progetto è interamente lato web, non abbiamo dovuto utilizzare delle macchine particolari per lo sviluppo. Abbiamo utilizzato semplicemente i nostri laptop (Windows, Linux, Mac).

Per ora, finché non ci daranno a disposizione un server dedicato abbiamo caricato il nostro sito web sulla VPN di Claudio.

1. **Progettazione e Implementazione**
   1. **Location**

L’utente entrerà dalla porta di entrata della scuola. All’inizio di ogni piano sarà presente un codice QR per entrare nel nostro sito web. Nel menu principale del nostro sito ci sarà una breve guida utente per guidarlo nell’utilizzo dell’applicativo. Una volta capito il funzionamento potrà scegliere quale funzione utilizzare.

* 1. **Pattern**

Il codice QR è il marchio di fabbrica di un tipo di codice a barre a matrice (o codice a barre bidimensionale). Un codice a barre è un'etichetta ottica leggibile dalla macchina che contiene informazioni sull'elemento a cui è allegato. In pratica, un codice QR spesso contiene dati per un localizzatore, identificatore o tracker che puntano a un sito Web o un'applicazione. Un codice QR è costituito da quadrati neri disposti su una griglia quadrata su uno sfondo bianco, che possono essere letti da un dispositivo di imaging come una fotocamera ed elaborati utilizzando la correzione dell'errore Reed-Solomon fino a quando l'immagine non può essere interpretata in modo appropriato. I dati richiesti vengono quindi estratti dai motivi presenti in entrambi i componenti orizzontali e verticali dell'immagine.

Tutti i codici QR hanno una forma quadrata e includono tre contorni quadrati negli angoli in basso a sinistra, in alto a sinistra e in alto a destra. Questi contorni quadrati definiscono l'orientamento del codice. I punti all'interno del codice QR contengono informazioni sul formato e sulla versione, nonché il contenuto stesso. I codici QR includono anche un certo livello di correzione degli errori, definito come L, M, Q o H. Una bassa quantità di correzione degli errori (L) consente al codice QR di contenere più contenuti, mentre una maggiore correzione degli errori (H) rende il codice più facile da scansionare. I codici QR hanno due vantaggi significativi rispetto agli UPC tradizionali: i codici a barre comunemente utilizzati negli imballaggi al dettaglio. Innanzitutto, poiché i codici QR sono bidimensionali, possono contenere significativamente più dati di un UPC monodimensionale. Mentre un UPC può includere fino a 25 caratteri diversi, un codice QR 33x33 (versione 4), può contenere 640 bit o 114 caratteri alfanumerici. Un codice QR 177x177 (versione 40) può contenere fino a 23.648 bit o 4.296 caratteri. Un altro vantaggio dei codici QR è che possono essere scansionati da uno schermo.

# Come creare un codice QR

Per creare un codice QR non è difficile. Non lo creeremo manualmente ma utilizzeremo strumenti disponibili su Internet. Il sito Web per la creazione del codice QR è qr-code-monkey.com. Un semplice sito Web consente di creare codici QR modificando lo stile per renderlo più bello esteticamente.

# QR Code Floors

Noi avremo 4 codici differenti per ogni piano:



Il rettangolo verde rappresenta quello del primo piano.



Il rettangolo giallo rappresenta quello del secondo piano.

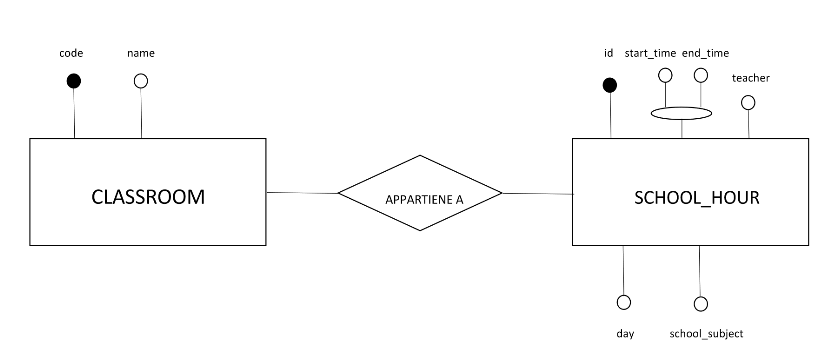


Il rettangolo rosso rappresenta quello del terzo piano.



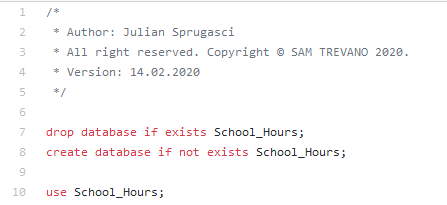
Il rettangolo blu rappresenta infine quello del quarto piano.

* 1. **Database**



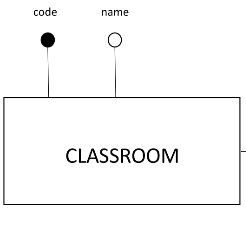
* 1. **Creazione del database**
     1. **Introduzione**

Come prima cosa, bisogna ovviamente creare il database **School\_Hours** per riuscire ad inserire tutti gli orari dell’anno scolastico. Per farlo è molto semplice.

* + 1. **Codice**
    2. **Entità Classroom**
       1. **Introduzione**

Questa entità ha il compito di racchiudere tutte le classi del quarto piano attraverso il nome dell’aula e il codice. Successivamente questa entità verrà poi collegata a **School\_Hour** per assegnare all’aula corrispondente il proprio orario scolastico.

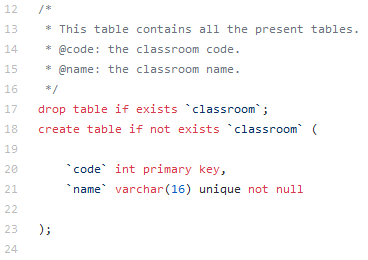
* + - 1. **Progettazione**



**Code:** Codice univoco che rappresenta e distingue ogni aula dalle altre.

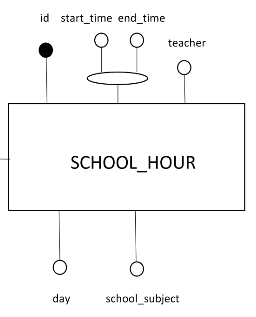
**Name:** Nome dell’aula in questione.

* + - 1. **Codice**



* + 1. **Entità School\_Hour**
       1. **Introduzione**

Questa entità ha il compito di racchiudere tutti gli orari del quarto piano attraverso vari parametri. Successivamente questa entità assegnerà alla entità **Classroom** l’orario corrispondente.

* + - 1. **Progettazione**

**Id:** Id univoco per identificare l’orario scolastico.

**Start\_time:** Orario di inizio della lezione in questione.

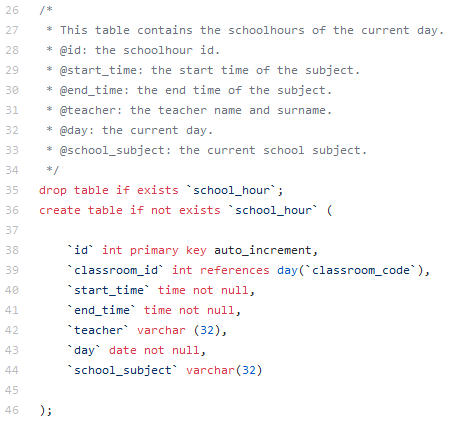
**End\_time:** Orario di fine della lezione in questione.

**Teacher:** Docente assegnato.

**Day:** Giorno della lezione.

**School\_subject:** Materia scolastica.

* + - 1. **Codice**



* 1. **Selenium**

Selenium è un Framework di test sviluppato in Java che permette di interagire con diversi browser Web. In effetti permette di realizzare dei test dinamici che facilitano dei test funzionali e quindi non richiedono l’esecuzione di un software. Per realizzare i nostri test abbiamo utilizzato l’estensione che si chiama Selenium IDE che può essere installata su qualsiasi browser.

* + 1. **Estrapolazione dati (Scraper)**

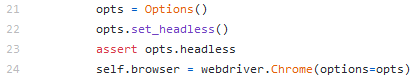
Per estrapolare i dati dal sito abbiamo creato uno script chiamato Scraper. Questo è uno script molto potente che permette in generale di estrapolare qualsiasi dato desiderato da qualsiasi sito possibile immaginabile. Per realizzarlo ci siamo basati su un esempio fornito dal nostro professore, però modificandone alcune parti.

* + 1. **Assegnazione del browser Chrome**

Il browser che abbiamo utilizzato per effettuare i vari test è Google Chrome, quindi nello script abbiamo dovuto specificare quale driver utilizzare. Per farlo abbiamo solo dovuto scrivere la seguente riga di codice:



* + 1. **Mascheramento del browser**

Ogni volta che avviamo il nostro Scraper questo aprirà il browser Google Chrome, andrà sul sito dell'orario della scuola ed eseguirà i vari test. Per nascondere la finestra del browser e evitare di mostrare all’utente le procedure eseguite dallo Scraper, è possibile mascherare il tutto aggiungendo i seguenti parametri:

* + 1. **Aggiunta di un sito web**

Per raggiungere un sito web dobbiamo fare una richiesta GET con webdriver di Selenium.

* + 1. **Verifica caricamento degli elementi**

Questo comando permette di verificare che un determinato elemento sia caricato correttamente. All’interno del metodo loaded\_check bisogna mettere l’id dell’elemento che disideriamo caricare. In questo caso l’id corrisponde al link interno “corsi”.

* + 1. **Ricavare elementi HTML**

Per potere ricavare un elemento HTML con Selenium bisogna utilizzare il metodo: **find\_element\_by\_id()** che permette di trovare un elemento con un determinato id.

Per cliccare diversi elementi di un sito con Selenium bisogna utilizzare il metodo click().

Ad esempio questa linea di codice permette di cliccare il link interno “corsi”.

* + 1. **Scrittura nome della classe**

Per potere scrivere il nome della classe all’interno della barra della ricerca abbiamo utilizzato il seguente codice. Esso permette di selezionare la barra di ricerca e di cercare l’orario di una determinata classe.

* + 1. **Cancellazione barra di ricerca**

Il metodo clear() permette di cancellare il contenuto presente all’interno della barra di ricerca (ricerche precedenti rimaste nella barra).

* + 1. **Scrittura all’interno della barra**

Il metodo send\_keys() permette di scrivere all’interno della barra di ricerca.

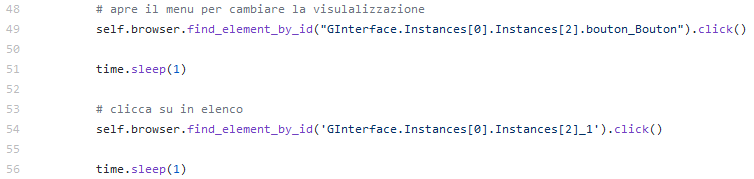
* + 1. **Avvio della ricerca**

Questa linea di codice permette allo Scraper di eseguire la ricerca attraverso i parametri inseriti nella barra di navigazione del sito degli orari scolastici.

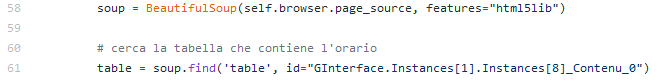
* + 1. **Recupero orario settimana successiva**

Durante la ricerca di una classe possiamo anche definire se ricavare l’orario della settimana successiva a quella corrente. Per farlo abbiamo realizzato nello Scraper un metodo che si chiama **clicca\_numero\_settimana\_dopo( ).** Ogni volta che vogliamo cercare possiamo impostare al metodo **cerca\_orario\_aule( )** un parametro a true o a false. Quest’ultimo ci permette di attivare un metodo (**clicca\_numero\_settimana\_dopo**) che sarà quello che andrà a cliccare in cima il numero della settimana successiva.

Per potere cambiare la visualizzazione dei dati dell’orario nella modalità elenco abbiamo aggiunto queste linee. Nel codice abbiamo aggiunto uno sleep, questo perché altrimenti non riuscirebbe ad eseguire l’animazione del menu a tendina che scende.



* + 1. **Parsing dei dati**

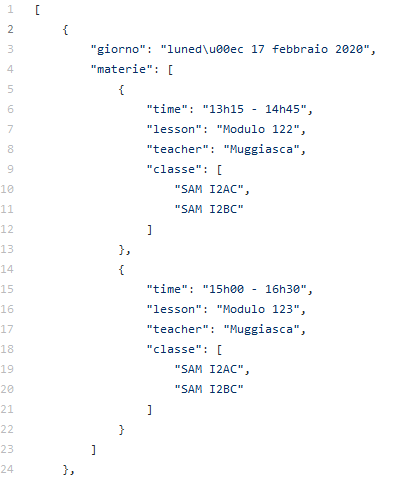
Una volta che abbiamo scelto la classe dovevamo selezionare la tabella dell’orario e iniziare a parsare i dati. Per farlo abbiamo aggiunto queste linee di codice dove utilizziamo la libreria beautiful soup.

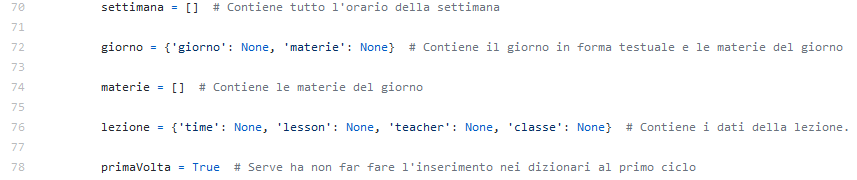
All’interno del metodo BautifulSoup dobbiamo definire su quale pagina web vogliamo parsare i dati. Con il metodo find possiamo andare a cercare nella pagina web un determinato tag con un determinato id.

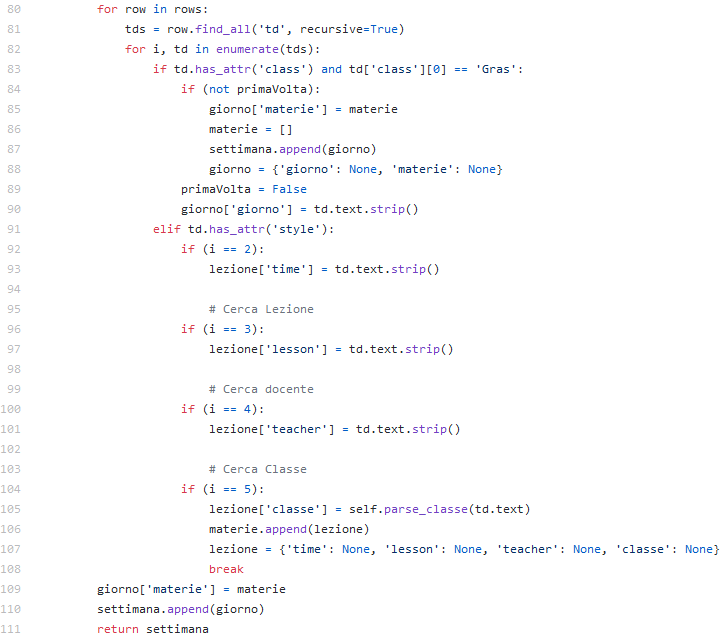
Prima di parsare i dati bisogna controllare se la tabella cercato contiene dei dati, questo perché potrebbe essere una settimana di vacanza dove non sono presenti lezioni. Inseguito per potere selezionare gli elementi della tabella dobbiamo cercare tutti tag tr che contengono dei tag td o th. Per farlo abbiamo aggiunto la seguente linea di codice.

Con il metodo find\_all possiamo catturare un tag che presente più volte all’interno della pagina web. Inseguito abbiamo pensato anche in quale modo salvare le varie informazioni all’interno di un file JSON.

* + 1. **Struttura del nostro JSON**

Nella immagine potete vedere la struttura del file JSON che abbiamo adottato:

Per creare la seguente struttura abbiamo creato e utilizzato le seguenti variabili.

* + 1. **Parte codice importante**

**Caso 1:** Il primo ciclo corrisponde ad ogni elemento **tr** della tabella.

**Caso 2:** Il secondo ciclo corrisponde ad ogni elemento **td** della tabella.

**Caso 3:** Al primo ciclo dobbiamo inserire all’interno dell’array settimana il giorno della settimana.

**Caso 4:** Invece per i cicli successivi fino al giorno successivo si deve inserire in ogni dizionario i dati.

**Alcuni metodi utilizzati per questo codice sono i seguenti:**

* **has\_attr** 🡺 permette selezionare un tag con un determinato attributo (class, id, style, …)
* **text.Strip()** 🡺permette di selezionare il contenuto di un tag.
* **append** 🡺permette aggiungere degli elementi ad un array.
* **parse\_classe** 🡺 metodo creato da noi per potere formattare nel modo corretto la classe. All’interno di questo codice abbiamo realizzato un’espressione regolare che corrisponde al testo che vogliamo scrivere all’interno del dizionario classe.
  1. **Flask**

Flask è un mini-framework in Python che viene utilizzato per lo sviluppo web e permette di creare dei propri siti web in modo dinamico e interattivo.

Abbiamo creato un file che si chiama app.py che sarà l’applicazione **flask**. All’interno per definire che si tratta di un **flask** dobbiamo importare la libreria **flask** e aggiungere la seguente linea di codice.

app = Flask(\_\_name\_\_)

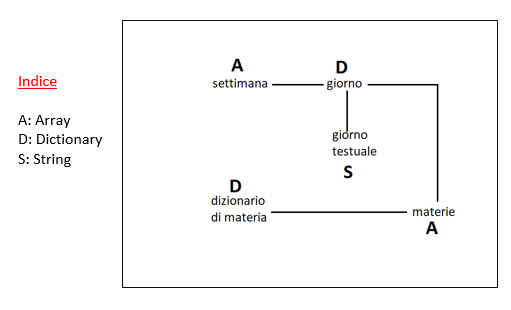
Inseguito bisogna definire la app routing che viene utilizzato per mappare l’URL specifico con la funzione associata che intende svolgere alcune attività (struttura simile al MVC) .

Nel nostro caso l’URL è associato alla funzione aule che permette di stampare l’orario di una determinata classe. Sì può anche aggiungere come parametro il tipo di richiesta che si vuole fare. In questo caso utilizziamo il metodo GET che consiste nell’accordare all’indirizzo della pagina web i diversi parametri contenenti i dati che si vogliono trasmettere. In questo caso vogliamo trasmettere se mostrare i dati della settimana corrente o quella successiva.

@app.route('/aule', methods=['GET'])

Per ricavare i parametri che sono presenti nel URL bisogna usare il metodo **request.args.get**.

* 1. **Struttura dei dati dell’orario**



1. **Test**

|  |  |  |  |
| --- | --- | --- | --- |
| **Numero Test** | **Descrizione** | **Aspettative** | **Risultato** |
| **1** | Inserire correttamente i dati all’interno delle tabelle **classroom** e **school\_hour**. | Si dovrebbe poter visualizzare i dati inerenti alla lezione attuale. | **Funziona** |

**Codice di test:**

