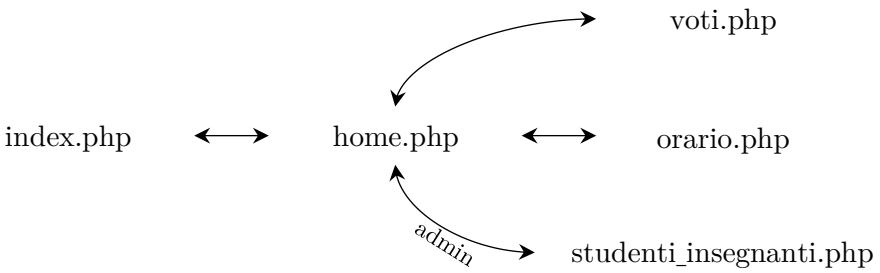
Federico Brandini (mat. 317271)
Federico Putamorsi (mat. 320041)

Server HTTPS con Registro Elettronico

1. Descrizione del sito

Partiamo introducendo la struttura del sito; per farlo possiamo utilizzare un grafo, dove ogni nodo corrisponde ad una pagina web, gli archi entranti costituiscono gli inlinks e quelli uscenti gli outlinks. Assunto ciò, la struttura del sito web può essere schematizzata come segue.

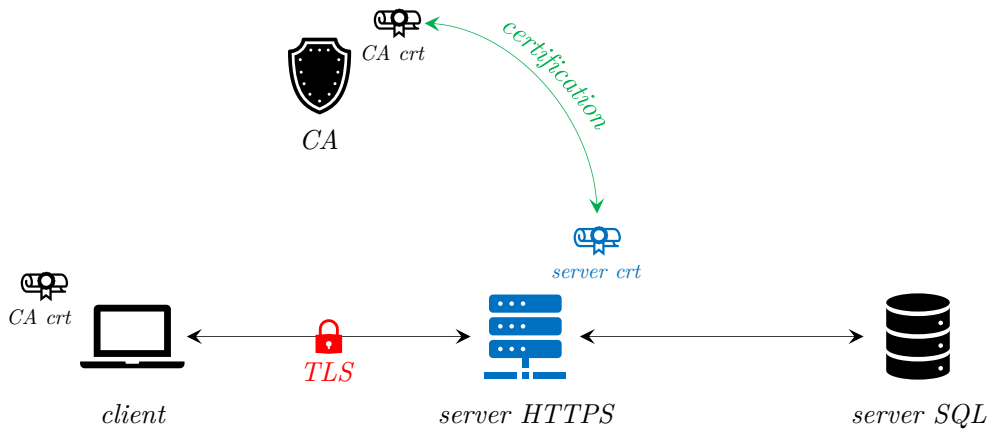


Il sito web inizia dal documento web di default *index.php*, con la schermata di login, il quale, una volta effettuato, porta a *home.php*, dove sarà possibile scegliere fra più funzionalità (due nel caso di studenti e semplici docenti e tre per docenti amministratori) attraverso un menù:

- **voti** (*voti.php*), che permette agli studenti di visualizzare i propri voti e ai docenti di inserirne di nuovi (oltre a visualizzarli e cancellarli);
- **orario** (*orario.php*), che consente a studenti e docenti di visualizzare il loro orario settimanale rispettivamente di lezione e di lavoro;
- **studenti e insegnanti** (*studenti_insegnanti.php*), che consente ai soli docenti amministratori di visualizzare tutti gli studenti e i docenti della scuola (oltre a eliminarli, inserirne di nuovi e promuovere ad amministratore i semplici docenti).

2. Architettura del sistema

Riassumiamo l'architettura dell'intero sistema realizzato nello schema seguente.



Tra le diverse entità troviamo

- la *Certification Authority* (*CA*), che emette certificati attendibili;
- il *client*, che riconosce come attendibile la CA, e quindi anche tutti i certificati firmati da quest'ultima (che non siano in stato di *revoked* oppure di *hold*);
- il *server HTTPS*, con all'interno un virtual host identificato dal nome di dominio **registroelettronico.local** e il relativo certificato emesso (e firmato) dalla CA;
- il *server SQL*, dove risiede l'intera base di dati del registro elettronico della scuola.

3. Strumenti utilizzati e Configurazione per testing locale

Per quanto concerne gli strumenti utilizzati, come server HTTP è stato impiegato Apache v2.4 e come server SQL-based, un surrogato offerto dalla piattaforma multiservizio XAMPP, MariaDB, quasi totalmente indistinguibile da un comune server SQL.

Non avendo acquistato alcun dominio, è necessario inserire la corrispondenza del fittizio **registroelettronico.local** con l'indirizzo IP di loopback 127.0.0.1 all'interno del file hosts del dispositivo locale attraverso la configurazione seguente.

```
127.0.0.1       registroelettronico.local
```

In questo modo, essendo che le name resolutions da tale file hanno la precedenza su quelle verso i server DNS, all'inserimento di tale dominio nella barra degli indirizzi del browser, si otterrà una redirection verso l'host locale.

Non essendoci nemmeno una reale CA e un reale certificato per il server, è necessario "simularli". A tal proposito è stato impiegato OpenSSL, un toolkit che consente di generare certificati a chiave pubblica e di firmarli attraverso una CA. In primo luogo, è necessario generare un certificato auto firmato per la CA (anche chiamato *certificato radice*, in quanto il primo nella catena di certificazioni) in formato X.509, con le seguenti due istruzioni (**Example-Root-CA** è il nome dell'autorità e **RootCA.crt** il suo certificato).

```
C:\Apache24\bin> openssl req -x509 -nodes -new -sha256 -days 1024 -newkey rsa:2048 -keyout
RootCA.key -out RootCA.pem -subj "/C=US/CN=Example-Root-CA"

C:\Apache24\bin> openssl x509 -outform pem -in RootCA.pem -out RootCA.crt
```

Fatto ciò, è necessario installare il certificato fittizio **RootCA.crt** sull'host locale nella sezione *Trusted Root Certificate Authorities* attraverso un inserimento manuale, oppure cliccando direttamente sul file del certificato stesso; quest'ultima opzione è disponibile soltanto per i certificati in formato X.509. A questo punto è possibile generare il certificato per il virtual host sul server HTTP, confermando l'uso di HTTPS; a tal proposito creiamo il file *domains.ext*, contenente l'elenco di tutti i nomi di dominio alternativi del nostro sito (solo uno, in questo caso).

```
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
```

keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
subjectAltName = @alt_names
[alt_names]
DNS.1 = **registroelettronico.local**

Generiamo, quindi, il certificato **regele.crt** per il virtual host.

C:\Apache24\bin> openssl req -new -nodes -newkey rsa:2048 -keyout regele.key -out regele.csr -subj "/C=US/ST=YourState/L=YourCity/O=Example-Certificates/CN="**registroelettronico.local**"

C:\Apache24\bin> openssl x509 -req -sha256 -days 1024 -in regele.csr -CA RootCA.pem -CAkey RootCA.key -CAcreateserial -extfile domains.ext -out **regele.crt**

Infine, configuriamo opportunamente l'host virtuale (da notare che il certificato generato pocanzi risiede in ${SRVROOT}/ssl/**regele.crt**).

```
Listen 443
<VirtualHost _default_:443>
        DocumentRoot "${SRVROOT}/htdocs/registroelettronico"
        ServerName registroelettronico.local:443
        SSLEngine on
        SSLCertificateFile "${SRVROOT}/ssl/regele.crt"
        SSLCertificateKeyFile "${SRVROOT}/ssl/regele.key"
</VirtualHost>
```

4. Estensione del progetto: server SMTP

Eventualmente...