

# **Progetto Reti di Calcolatori**

## **-GreenPass-**

### **2021/2022**



*A cura di :*

*Federico Prisco 0124001574*

*Mario Scognamiglio 0124002032*

## TRACCIA

Progettare ed implementare un servizio di gestione dei green pass secondo le seguenti specifiche. Un utente, una volta effettuata la vaccinazione, tramite un client si collega ad un centro vaccinale e comunica il codice della propria tessera sanitaria. Il centro vaccinale comunica al ServerV il codice ricevuto dal client ed il periodo di validità del green pass. Un ClientS, per verificare se un green pass è valido, invia il codice di una tessera sanitaria al ServerG il quale richiede al ServerV il controllo della validità. Un ClientT, inoltre, può invalidare o ripristinare la validità di un green pass comunicando al ServerG il contagio o la guarigione di una persona attraverso il codice della tessera sanitaria.

### 1.Descrizione del progetto

-Il client(utente) una volta vaccinato si collega al centro vaccinale per inviare il proprio ID della tessera sanitaria.  
Esso riceverà dal Centro Vaccinale uno dei seguenti messaggi :”Green pass creato con successo”oppure “Green pass già creato”. Il secondo messaggio si visualizzerà solo se sarà inserito lo stesso ID della tessera sanitaria più volte .

-Il centro vaccinale comunica al ServerV il codice ricevuto dal client ed il periodo di validità del Green pass che è di 6 mesi .

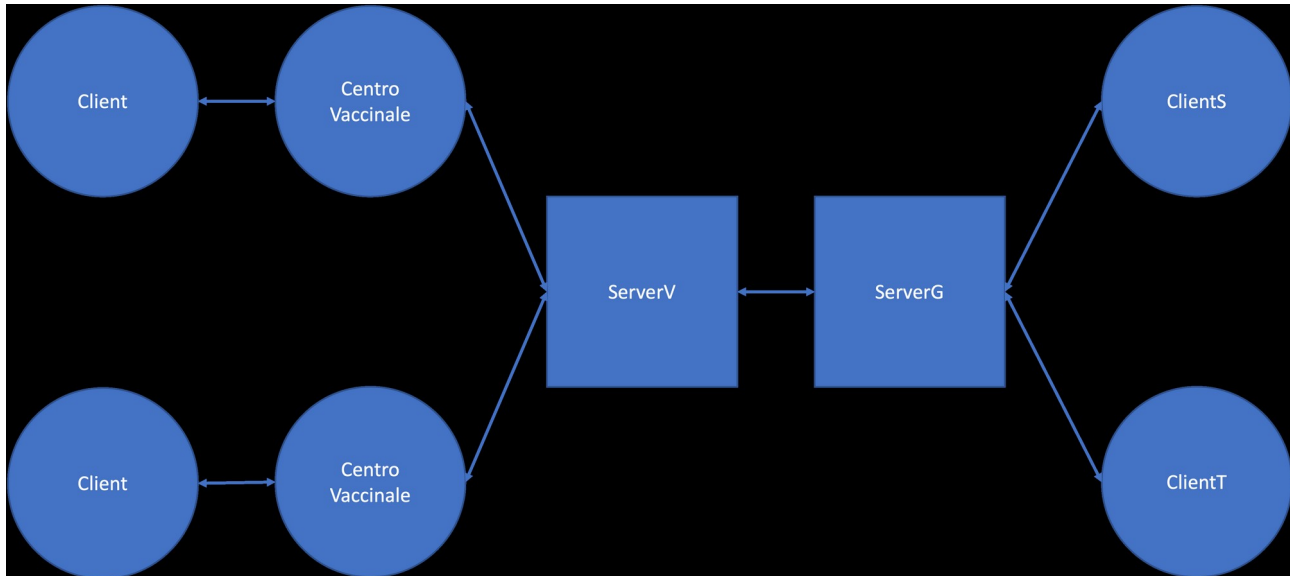
-Un ClientS può verificare se il Green pass è valido. Invia il codice della tessera sanitaria al ServerG che richiede a sua volta il controllo di validità al ServerV .  
Il serverG riceverà dal serverV l’esito della validità del GreenPass e a sua volta lo inoltrerà al clientS .

-Il ClientT può invalidare o ripristinare la validità del Green pass comunicando al ServerG il contagio o la guarigione dell’utente attraverso il codice della tessera sanitaria .

L’invio e la ricezione dei messaggi vengono effettuati attraverso la chiamata ‘FullRead’ e ‘FullWrite’.

Per migliorare la leggibilità del codice si definisce un file ‘wrapper.h’ in cui sono state inserite le funzioni .

## 2. Descrizione e schemi dell'architettura



- Il client effettua la richiesta di inserimento dell'ID tessera
- Il centro vaccinale è un peer perché rimane in ascolto delle richieste client, ed effettua richieste al ServerV
- Il clientS effettua la richiesta per verificare un ID tessera
- Il clientT effettua la richiesta per invalidare o ripristinare un ID tessera
- ServerG è un peer perché rimane in ascolto delle richieste clientS e clientT, ed effettua richieste al ServerV

## 3. Descrizione e schemi del protocollo applicazione

Sono state utilizzate molte comunicazioni di un solo carattere. L'informazione sul significato del carattere è memorizzata nell'entità ricevente.

Ad esempio:

Quando il ServerV riceve una richiesta dal ServerG, il serverV riceve un solo carattere come primo messaggio ovvero 'g' esso capirà che si tratta di una richiesta del ServerG, oppure se riceve 's' esso capirà che si tratta di una richiesta di validità del green pass dal ClientS e stesso modo per 't' per una richiesta di modifica del green pass dal clientT.

Tale scelta ha il vantaggio di saturare al minimo la banda in fase di utilizzo.

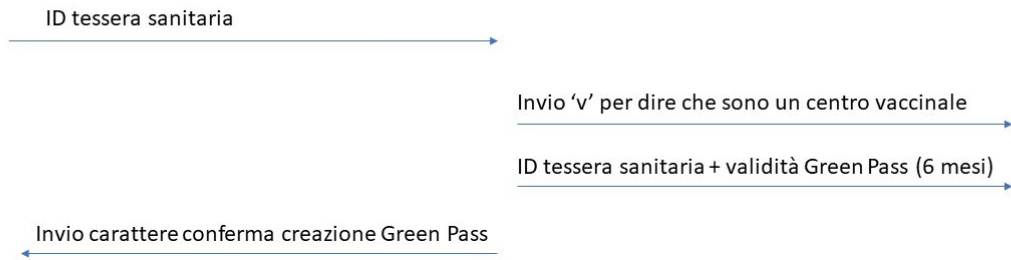
Di seguito le funzionalità che richiedono la comunicazione tra entità.

## -Rilascio Green Pass

Client

Centro Vaccinale

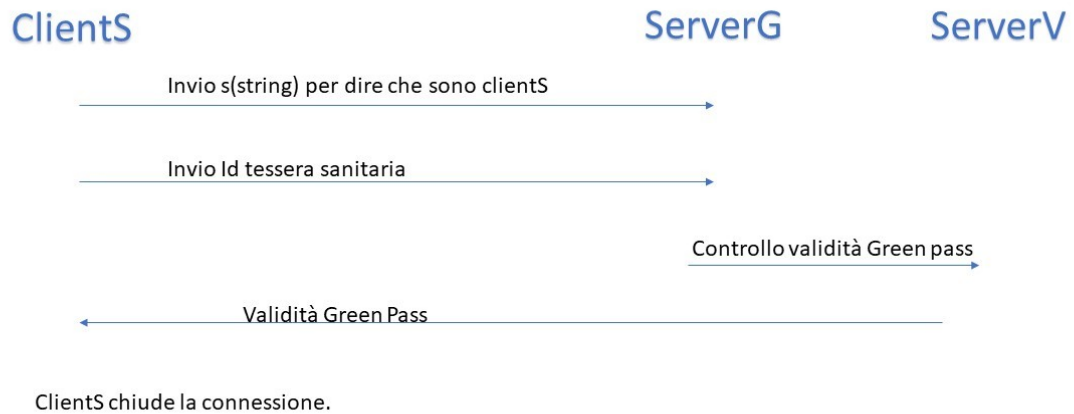
ServerV



Client chiude la connessione

Il client si collega al centro vaccinale . Una volta avvenuta la connessione il client invia l' ID della tessera sanitaria .Il centro vaccinale invia l'ID della tessera sanitaria e la durata del Green pass al serverV , infine risponde al client con la creazione del Green Pass se tutto è avvenuto in maniera corretta.

## Controllo validità Green Pass(ClientS)



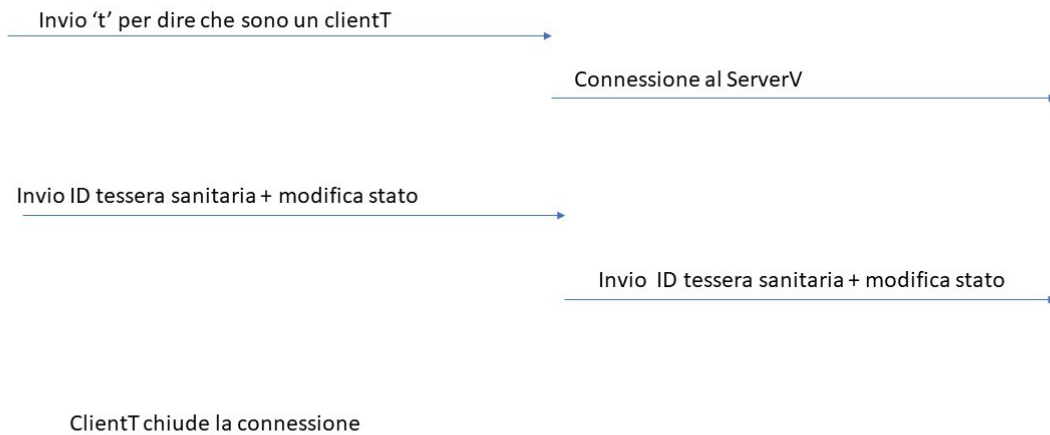
Dopo aver inserito l' ID della tessera sanitaria dei vari utenti vaccinati che hanno ottenuto il green pass, per controllare la validità di un green pass e vedere se l'utente è in quarantena oppure no, il clientS invia l'ID della tessera sanitaria al serverG il quale inoltrerà al serverV. Il serverV inoltrerà l'esito al ServerG il quale a sua volta inoltrerà l'esito al clientS. Dopo di che il clientS termina.

## Modifica Stato(ClientT)

ClientT

ServerG

ServerV



Infine per modificare lo stato di un utente in “guarigione”o “quarantena” il clientT si collega al ServerG e gli comunica l’eventuale scelta . Il serverG a sua volta si collega al serverV per comunicargli l’ID della tessera sanitaria e l’eventuale modifica di stato . Il clientT termina .

## **Implementazione :**

Il servizio richiesto , ovvero la gestione del Green Pass è un servizio dove lo scambio di informazioni è relativamente ridotto, si richiede che non ci sia perdita delle informazioni . Per questo motivo e per la sua affidabilità è stato utilizzato il protocollo TCP .

E' stata utilizzata, inoltre, la libreria pthread che permette di generare più thread in grado di effettuare lavori diversi ed indipendenti .

## **4.Dettagli implementativi**

### **-Client**

Il client stabilisce una connessione con il centro vaccinale(server)sulla porta 8080 , successivamente invia il codice della tessera sanitaria preso in input e lo invia al centro vaccinale . Se il centro vaccinale risponde con 'y' allora il Green pass è stato creato, nel caso contrario, invece, è stato già creato in precedenza .

### **-ClientS**

Il clientS stabilisce una connessione con il ServerG sulla porta 8082 . Il clientS prende in input l' ID della tessera sanitaria che si vuole verificare , lo invia al serverG , il quale poi risponderà al clientS dopo aver chiesto al serverV . Il clientS riceverà uno dei seguenti messaggi:

- l' ID della tessera sanitaria è errato
- il Green pass è invalidato
- il Green pass è valido

### **-ClientT**

Il clientT stabilisce una connessione con il ServerG sulla porta 8082 .

Prende in input l'ID della tessera sanitaria e la relativa modifica di stato e la invia al ServerG .Successivamente i messaggi vengono inoltrati al serverV .

Infine, il clientT riceverà uno dei seguenti messaggi :

- la modifica è avvenuta con successo
- il codice della tessera sanitaria non è valido
- Se si effettua una modifica di stato del tipo guarigione quando il Green pass è ancora valido, allora si riceverà il messaggio: "il paziente non è stato contagiato"
- Se si inserisce una modifica di stato "quarantena" quando il paziente è già in quarantena, si riceverà il seguente messaggio:"il paziente è già in quarantena".

### **-Centro Vaccinale**

Il centro vaccinale è un peer che rimane in ascolto delle richieste del client sulla porta 8080 ed effettua le richieste al serverV sulla porta 8081 .Una volta che il centro vaccinale è stato riconosciuto dal ServerV , invia l' ID della tessera sanitaria e la durata del Green pass(ovvero 6 mesi). Viene creato un thread indipendente per ogni client connesso .

### **-ServerV**

Il ServerV rimane in ascolto del ServerG e del centro vaccinale sulla porta 8081 . Si crea un thread indipendente per ogni entità connessa. Il ServerV ha funzione principale rispetto al ServerG siccome tutte le comunicazioni arrivano a lui . In base alle entità il server soddisfa la richiesta . Per riconoscere le varie entità che si collegano , il serverV legge il primo messaggio , se è 'g' allora si è connesso il ServerG altrimenti se è 'v' si è connesso il centro vaccinale . Quando il server riceve 'g' fa un'ulteriore distinzione per capire quale client ha inoltrato la richiesta al serverG (clientS,clientT) .Per poi soddisfare tale richiesta.

### **-ServerG**

Il ServerG è un peer che rimane in ascolto delle entità clientS e clientT sulla porta 8082 .Effettua anche richieste al ServerV collegandosi sulla porta 8081 . Stabilisce una connessione al ServerV per poi mettersi in ascolto per eventuali connessioni da parte del clientT e il clientS. Viene creato un thread indipendente per ogni client connesso . Il serverG fa da tramite ed invia tutti i messaggi dei vari client al ServerV .

### **MANUALE UTENTE**

La cartella principale contiene più sottocartelle :

-Centro\_vaccinale

-Client

-Client\_S

-Client\_T

-Server\_G

-Server\_V

Tutte queste cartelle contengono file.c(codici sorgenti), file wrapper.h(si riferiscono all'header) , file.out(eseguibili precompilati) ed infine il makefile .

Utilizzare un sistema Linux dotato di compilatore gcc.



## Come compilare ed eseguire

Aprire il terminale direttamente nelle cartelle , o spostarsi attraverso le cartelle .  
Il processi sono da eseguire in ordine cronologico.E' stato inserito all'interno di ogni cartella un makefile che ci permette di velocizzare le operazioni di compilazione da terminale. Digitare e inviare i comandi da terminale come segue.

```
ServerV:-make  
        -./server_v
```

```
Centro_Vaccinale : -make  
                   -./centro_v
```

```
ServerG:-make  
        -./server_g
```

```
Client: -make  
        -./client  
ClientT: -make  
        -./client_t
```

```
ClientS: -make  
        -./client_s
```

Il serverV risponde sulla porta 8081, e non richiede alcun argomento in input.

```
-./server_v
```

Per quanto riguarda il ServerG ,non si riporta nessun indirizzo IP in input come argomento dal momento che è stato implementato nel codice l'indirizzo ip (127.0.0.1) se si vuole utilizzare la stessa macchina . Qualora si volesse eseguire su macchine separate si dovrebbe andare a cambiare all'interno del codice l'indirizzo ip .

```
-./server_g
```

Il centro vaccinale non richiede nessun argomento in input come argomento dal momento che è stato implementato nel codice l'indirizzo ip (127.0.0.1) se si vuole utilizzare la stessa macchina . Qualora si volesse eseguire su macchine separate si dovrebbe andare a cambiare all'interno del codice l'indirizzo ip .

```
-./centro_v
```

Nei vari client è stato implementato l'indirizzo ip(127.0.0.1) se si vuole utilizzare la stessa macchina .

```
-./client  
-./client_t  
-./client_s
```

## Istruzioni di utilizzo:

Quando tutte le entità sono online è possibile utilizzare il software in maniera corretta .