

UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE  
SCUOLA INTERDIPARTIMENTALE DELLE SCIENZE, DELL'INGEGNERIA E DELLA SALUTE  
INFORMATICA

# Progetto Reti di Calcolatori



GreenPass

**Proponenti:**  
Cerqua Crescenzo 0124002171  
Salese Fabio 0124002165

**Data di Consegna:** 24/02/2023

**Anno Accademico:**  
2022 – 2023

**Docente:**  
Prof. A. Ferone

## Sommario

<b>CentroVaccinale .....</b>	<b>4</b>
<b>ServerV .....</b>	<b>4</b>
<b>ServerG .....</b>	<b>5</b>
<b>ClientS .....</b>	<b>6</b>
<b>ClientT .....</b>	<b>7</b>
<b>Pacchetti Applicazione .....</b>	<b>7</b>
<b>Descrizione del protocollo .....</b>	<b>7</b>
<b>Dettagli implementativi dei client .....</b>	<b>9</b>
<b>Dettagli implementativi dei server .....</b>	<b>11</b>
<b>Manuale Client – Guida alla compilazione ed esecuzione.....</b>	<b>12</b>

## Introduzione

La pandemia di COVID-19 ha portato alla necessità di trovare soluzioni innovative per la gestione della salute pubblica. In questo contesto, la creazione di un sistema di gestione dei green pass rappresenta una risposta importante e attuale alle esigenze di tracciabilità e monitoraggio dei casi di infezione. Nel presente progetto, abbiamo progettato ed implementato un servizio di gestione dei green pass, secondo specifiche che prevedono l'utilizzo di un client per la comunicazione con il centro vaccinale, il server di validazione dei green pass e il server di gestione delle invalidazioni e ripristini. In particolare, il nostro sistema prevede la comunicazione tra utenti e centro vaccinale, la validazione dei green pass tramite un server dedicato, e la gestione delle invalidazioni e ripristini attraverso un'apposita interfaccia. In questa relazione, forniremo una descrizione dettagliata dell'architettura del sistema, delle funzionalità implementate e dei dettagli implementativi, insieme a una guida Client per la compilazione ed esecuzione del sistema.

## Descrizione del progetto

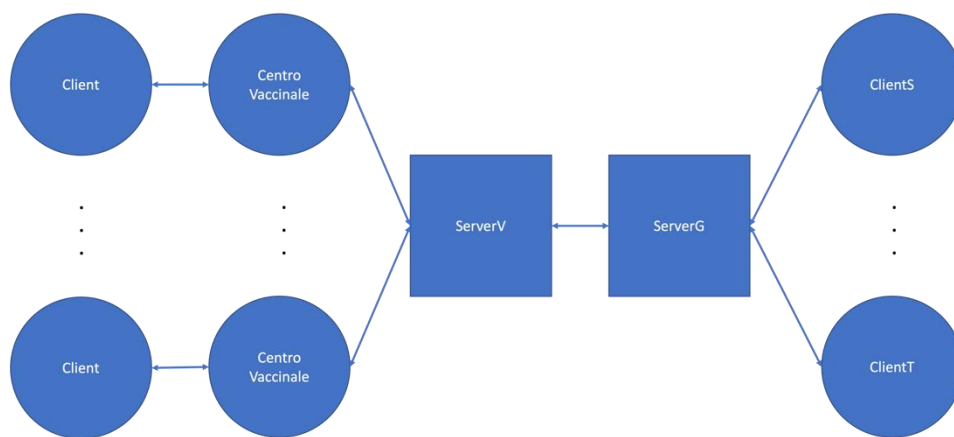
Il nostro progetto consiste nella progettazione e implementazione di un servizio di gestione dei green pass, il quale permette la verifica dell'avvenuta vaccinazione di una persona tramite un'interfaccia client-server. Il servizio prevede l'utilizzo di tre componenti principali: il client per la comunicazione con il centro vaccinale, il server di validazione dei green pass e il server di gestione delle invalidazioni e ripristini.

In particolare, l'utente, dopo aver effettuato la vaccinazione, utilizza il Client per collegarsi al centro vaccinale e comunicare il codice della propria tessera sanitaria. Il centro vaccinale, a sua volta, comunica al server di validazione dei green pass (ServerV) il codice ricevuto dal client e il periodo di validità del green pass. Successivamente, il client di verifica (ClientS) può inviare il

codice di una tessera sanitaria al server di gestione dei green pass (ServerG) per controllare la validità del green pass associato.

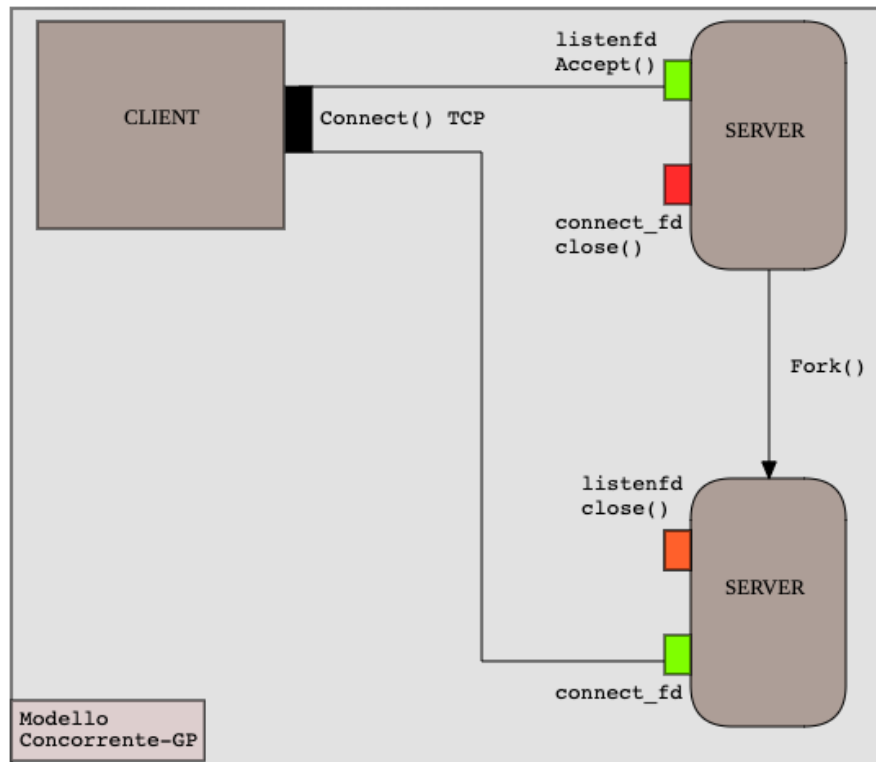
Inoltre, il server di gestione dei green pass consente anche di invalidare o ripristinare la validità di un green pass. In caso di contagio o guarigione di una persona, il client di gestione (ClientT) può comunicare al ServerG il codice della tessera sanitaria per aggiornare lo stato del green pass associato.

Il nostro sistema si basa sull'utilizzo di un protocollo di comunicazione tra i vari componenti, il quale viene descritto dettagliatamente nella sezione successiva della relazione.



## DESCRIZIONE E SCHEMI DELL'ARCHITETTURA

Il progetto è stato realizzato adottando un modello di programmazione client-server. Il server implementato è in grado di gestire le richieste provenienti da più client contemporaneamente, grazie all'utilizzo di un modello *concorrente*. Questo è stato ottenuto mediante l'utilizzo della systemcall *fork()*, che permette la creazione di un numero N di processi figli in base alle connessioni dei client accettate dal server. Il processo padre è responsabile di gestire il descrittore *listenfd()* per l'accettazione delle richieste di connessione, mentre ogni figlio si occupa della gestione del descrittore *connfd()*, ovvero della connessione attiva con il client e della fornitura dei relativi servizi.



## CentroVaccinale

CentroVaccinale è un servizio che si occupa di gestire la validità dei green pass per gli utenti che si sono vaccinati. Il servizio si compone di un client che viene utilizzato dagli utenti per comunicare al centro vaccinale il proprio codice di tessera sanitaria dopo aver effettuato la vaccinazione.

Una volta ricevuto il codice, il centro vaccinale comunica al ServerV il codice ricevuto dal client e il periodo di validità del green pass associato al Client. Il ServerV gestisce quindi la lista di tutti i green pass validi, memorizzando le informazioni sulla validità di ciascuno.

Per verificare la validità di un green pass, un ClientS può inviare il codice di una tessera sanitaria al ServerG, che richiede al ServerV il controllo della validità del green pass associato al codice della tessera sanitaria fornito. Se il green pass è valido, il ServerG lo comunica al ClientS. In caso contrario, il ServerG indica al ClientS che il green pass non è valido.

Infine, il CentroVaccinale consente anche ai ClientT di 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. Questo consente al ServerG di aggiornare lo stato del green pass associato al codice della tessera sanitaria e di comunicare ai ClientS se il green pass è ancora valido o no.

## ServerV

Il ServerV è il fulcro del progetto, poiché svolge un ruolo fondamentale nella gestione delle informazioni relative ai Green Pass degli utenti. Il suo compito principale è quello di gestire la creazione, l'aggiornamento e la memorizzazione dei certificati verdi all'interno di un database simulato attraverso un filesystem persistente.

Per fare ciò, il ServerV si mette in ascolto di connessioni da parte del CentroVaccinale e attende la ricezione dei dati relativi all'avvenuta vaccinazione di un utente. Una volta ricevuti questi dati, il ServerV crea un file univoco per ogni Green Pass, assegnando il nome del file al numero di tessera sanitaria ricevuto dal CentroVaccinale e inserendo al suo interno la data di inizio e fine validità del Green Pass. In questo modo, il ServerV svolge la funzione di database, consentendo la memorizzazione e la consultazione dei Green Pass in modo efficiente.

Il ServerG, invece, si occupa di verificare la validità di un Green Pass. Per fare ciò, contatta il ServerV e richiede il certificato verde corrispondente al numero di tessera sanitaria fornito dal client. Il ServerV effettua quindi una "query" per cercare il Green Pass richiesto utilizzando il numero di tessera sanitaria salvato. Se il file esiste, il ServerV lo invia al ServerG per consentire le opportune operazioni. In questo modo, il ServerV svolge la funzione di supporto al controllo della validità dei Green Pass.

Per gestire la comunicazione con il CentroVaccinale e con il ServerG, è stata adottata la seguente convenzione: all'apertura della connessione con il ServerV, sia il CentroVaccinale che il ServerG inviano un bit di riconoscimento, che il ServerV utilizza per determinare chi ha richiesto la connessione. Se il bit inviato ha un valore "1", il ServerV gestisce la connessione con il CentroVaccinale. Se il bit inviato ha un valore "0", il ServerV gestisce la connessione con il ServerG. Questo sistema di riconoscimento consente al ServerV di stabilire immediatamente la provenienza della richiesta e di gestire la connessione in modo corretto.

Infine, il ServerV gestisce anche le modifiche di validità dei certificati verdi. Tali modifiche avvengono a seguito delle richieste effettuate dal ClientT, identificabile come organizzazione sanitaria, come l'Cl. In pratica, il ServerV riceve dal ClientT le informazioni riguardanti la modifica di validità di un determinato Green Pass e, una volta verificate le disposizioni indicate, provvede a modificare la relativa data di validità nel filesystem in cui sono archiviati tutti i certificati verdi. In questo modo, il ServerV contribuisce in modo significativo alla gestione dei Green Pass, garantendo l'aggiornamento tempestivo delle informazioni sullo stato di salute degli utenti.

## ServerG

Il ServerG è una componente critica dell'intera infrastruttura del sistema di gestione dei Green Pass. Il suo compito principale è quello di verificare la validità dei certificati verdi, interagendo con il ClientS attraverso l'App Scansione. Dopo aver stabilito la connessione con il ClientS, il ServerG invia un messaggio di benvenuto e un form in cui l'Client dovrà inserire il numero di tessera sanitaria associato al certificato verde da verificare.

Una volta ricevuti i dati, il ServerG si connette al ServerV inviando il numero di tessera sanitaria ricevuto. Il ServerV riceve la richiesta del ServerG e verifica la presenza del certificato verde richiesto nel suo filesystem. Se il certificato verde esiste, il ServerV lo invia al ServerG per la successiva verifica.

Il ServerG effettua quindi una verifica sulla validità del Green Pass, controllando la data di inizio e fine validità all'interno del certificato verde. In base alla validità del certificato, il ServerG comunica al ClientS l'esito positivo o negativo della verifica.

In questo modo, il ServerG svolge un ruolo essenziale nell'assicurare la sicurezza e l'accuratezza delle informazioni sui certificati vaccinali. Il ServerG, inoltre, contribuisce a garantire l'integrità e la sicurezza dell'intera infrastruttura del sistema di gestione dei Green Pass.

## Client

Il Client, ovvero il Client che desidera richiedere l'emissione di un Green Pass, avvia il processo di connessione al CentroVaccinale, che si presenta con un messaggio di benvenuto.

Una volta stabilita la connessione, il CentroVaccinale richiede al 'Client di compilare un form contenente il proprio nome, cognome e numero di tessera sanitaria. È importante notare che il numero di tessera sanitaria deve essere esattamente di 10 caratteri, altrimenti il CentroVaccinale non accetterà la richiesta.

Una volta che il Client ha compilato il form con i propri dati, lo invia al CentroVaccinale che invierà un messaggio di conferma di ricezione. A questo punto, il CentroVaccinale procede ad elaborare i dati forniti dal Client, verificando la correttezza del numero di tessera sanitaria e che il richiedente soddisfi i requisiti per l'emissione del Green Pass.

Successivamente, il CentroVaccinale si connette al ServerV per generare il Green Pass richiesto. Per fare ciò, prende la data di inizio e ne aggiunge 3 mesi per definire il periodo di validità del certificato verde. Infine, il CentroVaccinale invia un pacchetto al ServerV contenente il numero di tessera sanitaria e il periodo di validità del Green Pass appena creato.

In caso di esito positivo, il CentroVaccinale crea il Green Pass per il Client e lo invia al ServerV, che si occuperà di archivarlo nel filesystem e di renderlo disponibile per le successive verifiche da parte del ServerG.

Il CentroVaccinale invia poi al Client l'esito positivo della richiesta di emissione del Green Pass, insieme alle informazioni sulla sua validità e sulle relative disposizioni normative.

In caso di esito negativo, il CentroVaccinale comunica al Client la ragione del rifiuto, ad esempio perché il numero di tessera sanitaria fornito non è valido o perché il Client non soddisfa i requisiti per l'emissione del Green Pass.

## ClientS

Il ClientS è un'applicazione per la scansione e la verifica di un Green Pass. Una volta avviata l'applicazione, essa si connette al ServerG per avviare il processo di verifica. Dopo aver stabilito la connessione, l'App riceve un messaggio di benvenuto e viene richiesto di inserire il numero di tessera sanitaria associata al Green Pass da verificare. Questo codice identificativo viene inviato al ServerG che lo inoltra al ServerV per la verifica della validità del Green Pass.

L'applicazione rimane in attesa della risposta dei due server, e in particolare del ServerG che riceverà la risposta dal ServerV. Una volta verificata la validità del GP, l'Applicazione riceve un responso dal ServerG che indica se il GP è valido o non valido. Questo risultato viene visualizzato sull'interfaccia dell'applicazione per comunicare al Client il risultato della verifica.

In questo modo, il ClientS svolge un ruolo importante nella verifica della validità dei Green Pass, fornendo un servizio rapido e preciso per garantire la sicurezza degli utenti e la conformità alle normative in vigore.

## ClientT

Il ClientT è un'Applicazione che ha il compito di gestire la validità dei Green Pass attraverso l'invio di referti di tamponi. Il processo inizia con la connessione del ClientT al ServerG, il quale richiede l'invio del referto di un tampone. Una volta ricevuto il referto, il ServerV aggiorna il campo "report" del file associato al Green Pass corrispondente in base al risultato del tampone. In particolare, se il ClientT invia un valore "1", il Green Pass viene invalidato, mentre se invia uno "0" il Green Pass viene ripristinato.

In questo processo, l'App Green Pass (ClientS) svolge un ruolo cruciale, poiché consente di verificare la validità dei certificati verdi in tempo reale, tenendosi costantemente aggiornata sulle variazioni apportate dal ClientT. In questo modo, si garantisce che i Green Pass siano sempre aggiornati e corretti, garantendo la massima sicurezza e accuratezza delle informazioni sui certificati vaccinali.

## Pacchetti Applicazione

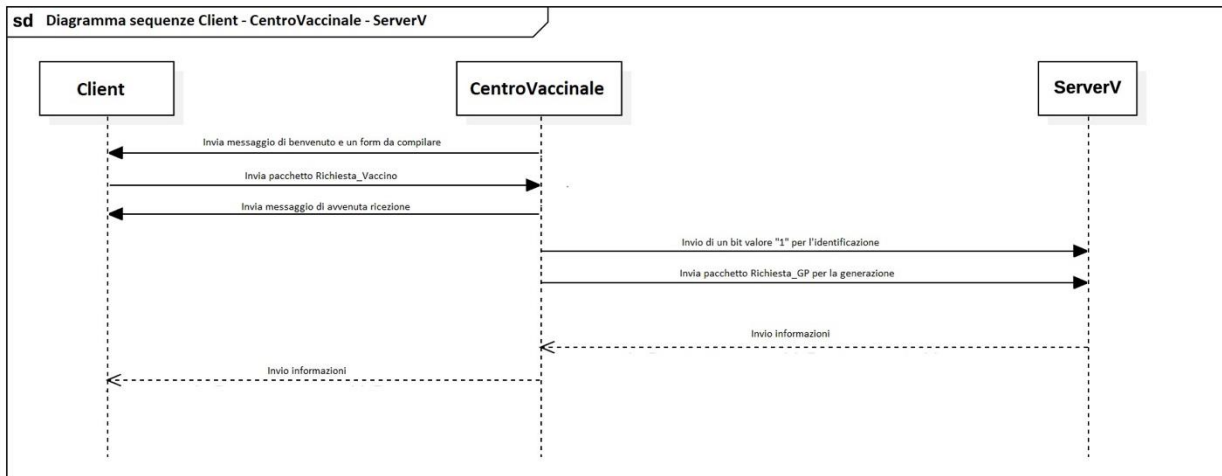
I Pacchetti Applicazione sono un insieme di pacchetti utilizzati da diverse parti interessate nella gestione del Green Pass. In particolare, il Client utilizza il pacchetto Richiesta\_Vaccino per richiedere al Centro Vaccinale la generazione di un certificato verde. Richiesta\_Vaccino contiene tre campi, ovvero nome, cognome e numero di tessera sanitaria del Client, con una dimensione massima di DIM\_MAX (1024 byte) per i campi nome e cognome, e di DIM\_ID (11 byte) per il numero di tessera sanitaria, incluso il carattere terminatore.

Il Centro Vaccinale, a sua volta, utilizza la struttura DATA per salvare la data di generazione del pacchetto di richiesta del Green Pass da inviare al ServerV. La struttura RICHIESTA\_GP contiene il numero di tessera sanitaria del client e le date di inizio e fine validità del Green Pass, rappresentate come interi e memorizzate nei campi data\_inizio e scadenza. Il campo ID, che contiene il numero di tessera sanitaria, ha una dimensione di DIM\_ID (11 byte).

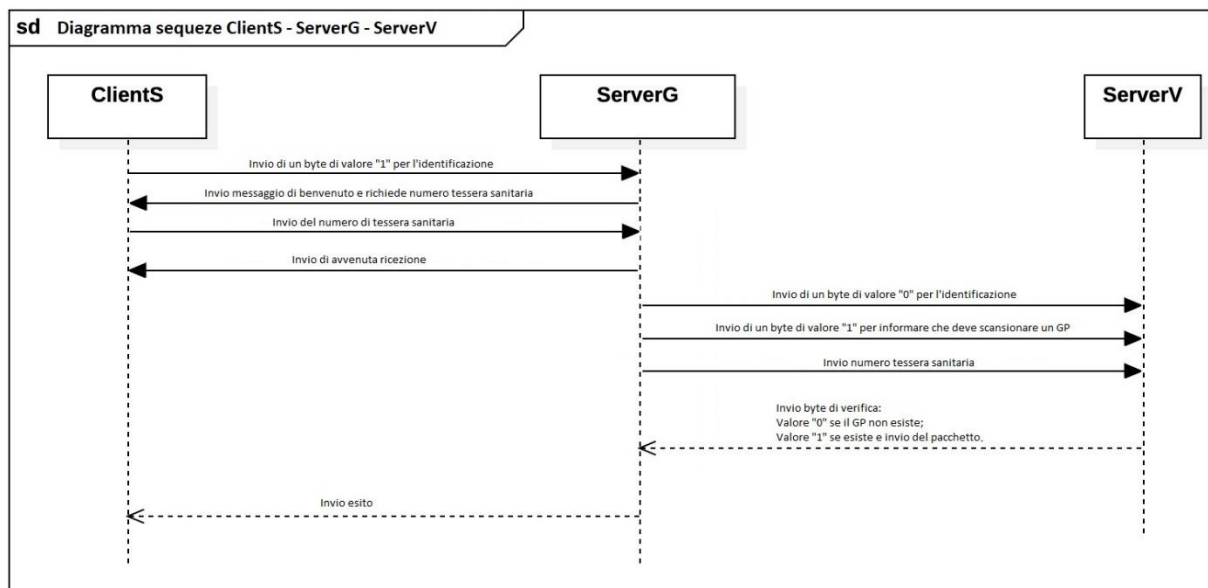
Infine, il CLIENT utilizza la struttura REPORT, che contiene il numero di tessera sanitaria associato al Green Pass e il referto del tampone effettuato sulla persona. Il campo ID ha una dimensione di DIM\_ID (11 byte), mentre il campo REPORT è di un solo byte di tipo char. Questa struttura viene utilizzata dall'CLIENTT per inviare il pacchetto al ServerG, al fine di verificare l'autenticità del Green Pass.

## Descrizione del protocollo

## Diagramma delle sequenze Client-CentroVaccinale-ServerV

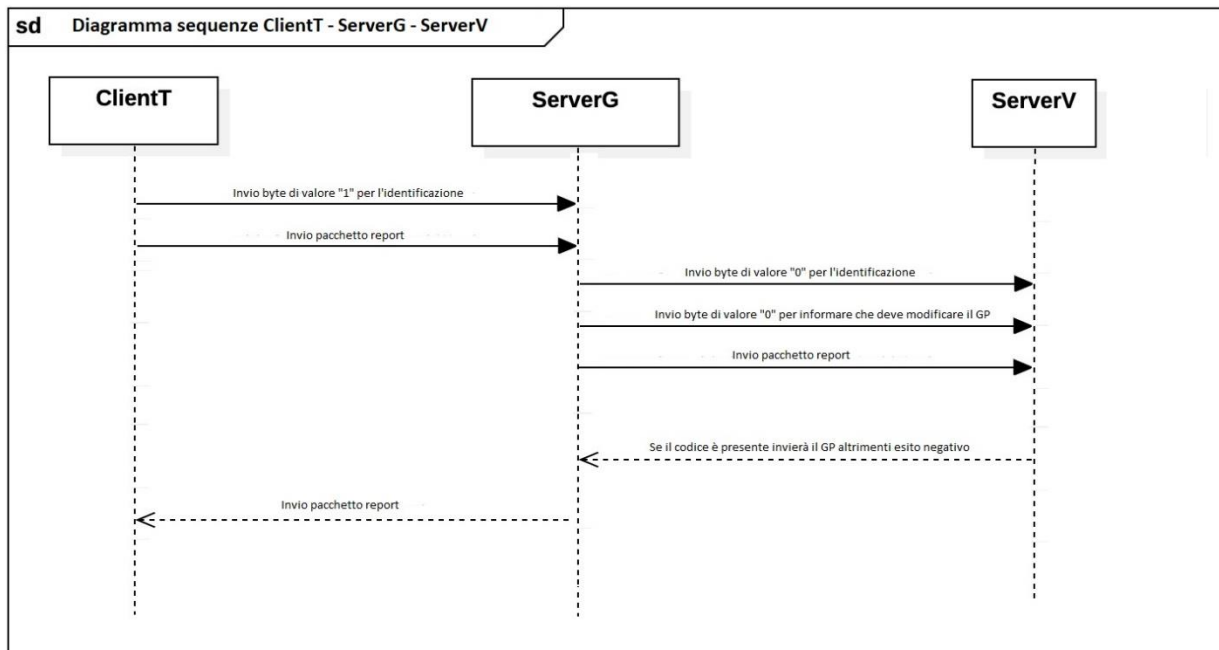


## Diagramma delle sequenze ClientS-ServerG-ServerV





## Diagramma delle sequenze ClientT-ServerG-ServerV



## Dettagli implementativi dei client

### Client

Il programma utilizza il protocollo TCP/IP per comunicare con il server del centro vaccinale tramite la libreria standard del C per la gestione delle socket.

Il programma include diverse librerie standard del C, come `stdio.h`, `stdlib.h`, `unistd.h`, `netdb.h`, `sys/types.h`, `sys/socket.h`, `arpa/inet.h` e `errno.h`, necessarie per la gestione della socket e la gestione degli errori.

In primo luogo, il programma definisce la struttura `RICHIESTA_VACCINO` che rappresenta il pacchetto applicazione da inviare al centro vaccinale. La struttura contiene tre campi: `name`, `surname` e `ID`. Il campo `name` rappresenta il nome del client che richiede la prenotazione del vaccino, il campo `surname` rappresenta il cognome del client e il campo `ID` rappresenta il codice tessera sanitaria del client.

Successivamente, il programma definisce una funzione `creazione_pacchetto()` che consente di creare un pacchetto `RICHIESTA_VACCINO` per l'Client. La funzione richiede all'Client di inserire il proprio cognome, nome e codice tessera sanitaria tramite la funzione `fgets()`, che legge una stringa da `stdin` (cioè l'input standard). La funzione `fgets()` legge una riga alla volta e la salva nel buffer specificato. La funzione `fgets()` restituisce `NULL` se si verifica un errore durante la lettura della stringa.

Dopo aver letto le informazioni dall'Client, la funzione `creazione_pacchetto()` restituisce il pacchetto `RICHIESTA_VACCINO` creato.

Il programma include anche due funzioni per la lettura e la scrittura dei dati sulla socket. La funzione `full_read()` legge un certo numero di byte in modo esatto, iterando la lettura finché non ha letto tutti i byte richiesti. Inoltre, la funzione `full_read()` è in grado di gestire l'interruzione della lettura da parte di una System Call.

La funzione `full_write()` esegue un ciclo di scritture ripetute fino a quando non sono stati scritti tutti i byte richiesti. La funzione continua a scrivere anche se viene interrotta da una System Call. Infine, il programma principale include il codice per la connessione al server del centro vaccinale e l'invio del pacchetto `RICHIESTA_VACCINO` al server. In particolare, il programma richiede all'Client di inserire l'indirizzo IP del server e la porta su cui il server è in ascolto. Successivamente, il programma utilizza la funzione `gethostbyname()` per risolvere l'indirizzo IP in un nome host e la funzione `socket()` per creare una nuova socket. Il programma utilizza quindi la funzione `connect()` per stabilire la connessione con il server.

Una volta stabilita la connessione, il programma invia il pacchetto `RICHIESTA_VACCINO` al server utilizzando la funzione `full_write()`. Il server risponde con un messaggio di `ACK`, che il programma legge dalla socket utilizzando la funzione `full_read()`.

## ***ClientS***

Il client si connette al server specificando l'indirizzo IP e la porta del server tramite una chiamata di sistema `socket()` e successivamente si connette tramite una chiamata di sistema `connect()`.

Il client invia un messaggio di un solo bit, che ha valore 0, al server per informarlo che la comunicazione successiva avverrà tra il client e l'applicazione C invece che con il ServerG.

Il client utilizza le funzioni `full_write()` e `full_read()` per inviare e ricevere i dati in modo affidabile. Queste funzioni garantiscono che il numero di byte specificato in input venga scritto o letto senza interruzioni e in modo affidabile.

Il client è in grado di comunicare con il server tramite una connessione TCP a basso livello, grazie all'utilizzo della libreria di sistema `netdb.h`, che contiene le definizioni per le operazioni del database di rete. Inoltre, il client fa uso della libreria di sistema `sys/types.h` per definire i tipi di dati utilizzati nei sistemi Unix, della libreria `sys/socket.h` per definire i socket, della libreria `arpa/inet.h` per le operazioni Internet, della libreria `unistd.h` per le funzioni e costanti standard del sistema Unix e della libreria `errno.h` per la gestione delle situazioni di errore.

Il client ha una dimensione massima del buffer definita come `DIM_MAX` pari a 1024 byte, una dimensione della risposta di conferma definita come `DIM_ACK` pari a 64 byte, una dimensione del messaggio di benvenuto dal ServerG definita come `DIM_BENVENUTO` pari a 108 byte, una dimensione dell'ACK ricevuto dal ClientS definita come `APP_ACK` pari a 39 byte e una dimensione del codice di tessera sanitaria definita come `DIM_ID` pari a 11 byte (10 byte + 1 byte del carattere terminatore).

## ***ClientT***

Il codice inizia con l'importazione delle librerie necessarie per gestire la comunicazione attraverso i socket. In particolare, sono importate le librerie standard per l'I/O, la gestione degli errori, le operazioni su Internet e le definizioni per le operazioni del database di rete.

Successivamente, sono definite alcune costanti utilizzate nel programma, tra cui la dimensione massima del buffer, la dimensione del codice di tessera sanitaria e la dimensione dell'acknowledgement ricevuto dal ServerG.

Definita una struttura `REPORT`, contenente il numero di tessera sanitaria di un green pass ed il suo referto di validità. La struttura viene utilizzata per rappresentare il pacchetto del clientT contenente il numero di tessera sanitaria di un green pass ed il suo referto di validità.

Sono definite le funzioni `full_read` e `full_write` utilizzate per leggere e scrivere esattamente count byte iterando opportunamente le letture e le scritture. Queste funzioni gestiscono anche la possibilità di essere interrotte da una System Call.

Nella funzione principale `main` si crea il descrittore di socket e si inizializza la struttura `sockaddr_in` che viene utilizzata per specificare l'indirizzo del server a cui connettersi. Viene quindi inizializzata

la struttura REPORT, il pacchetto di dati da inviare al server, e il bit di partenza start\_bit, che viene impostato a 1 per indicare al server che la comunicazione deve avvenire con il ServerG.

La stringa che rappresenta l'indirizzo IP del server viene convertita in un formato di rete, la funzione connect viene utilizzata per stabilire una connessione con il server e viene effettuato il controllo sull'avvenuta connessione.

Successivamente, la funzione full\_write viene utilizzata per inviare al server il pacchetto contenente il numero di tessera sanitaria del green pass e il suo referto di validità. Infine, viene chiusa la connessione con il server.

## Dettagli implementativi dei server

### *ServerV*

Il programma di occupa della gestione dei green pass e della comunicazione tra un client e un server. Vengono definite tre strutture: DATA, RICHIESTA\_GP e REPORT. La struttura DATA contiene giorno, mese e anno come interi; la struttura RICHIESTA\_GP contiene un ID (numero di tessera sanitaria del Client), una variabile report che rappresenta la validità del green pass, e due strutture DATA che rappresentano la data di inizio e fine validità del green pass; la struttura REPORT contiene un ID e una variabile report.

La funzione full\_read si occupa della lettura completa di un file descriptor, riceve un file descriptor, un buffer e la dimensione del buffer. La funzione cicla fino a che non ha letto tutti i dati richiesti e aggiorna il buffer e il numero di byte rimanenti da leggere.

La funzione full\_write si occupa della scrittura completa su un file descriptor, riceve un file descriptor, un buffer e la dimensione del buffer. La funzione cicla finché non ha scritto tutti i dati richiesti e aggiorna il buffer e il numero di byte rimasti da scrivere.

La funzione CV\_comunication si occupa di creare un nuovo green pass e di scriverlo su un file. Riceve un file descriptor e legge una struttura RICHIESTA\_GP tramite la funzione full\_read. In seguito, se il file descriptor è stato aperto correttamente, scrive la struttura su un file utilizzando la funzione write.

La funzione modifica\_report si occupa della modifica del report di un green pass. Riceve un file descriptor e legge una struttura REPORT tramite la funzione full\_read. In seguito, apre un file descriptor con il nome ID presente nella struttura REPORT e se il file descriptor è stato aperto correttamente legge la struttura RICHIESTA\_GP dal file. Utilizzando la funzione flock, la funzione verifica se il file è già aperto da un altro processo e lo blocca se necessario. Infine, modifica il valore della variabile report nella struttura RICHIESTA\_GP e scrive la struttura aggiornata sul file utilizzando la funzione write.

### *ServerG*

Il codice presentato è costituito da tre definizioni di struct, da due funzioni di lettura/scrittura full\_read() e full\_write() che effettuano la lettura/scrittura di una quantità di byte specifica anche in presenza di interruzioni causate da system call e da alcune funzioni che implementano la logica dell'app.

La prima struttura, "data" è utilizzata per rappresentare una data attraverso tre interi: giorno, mese e anno.

La seconda struttura, "Richiesta\_GP", è utilizzata per rappresentare una richiesta di green pass inviata dal centro vaccinale al ServerV. Essa contiene il numero di tessera sanitaria del client, la data di inizio e fine validità del green pass.

La terza struttura, "REPORT", è utilizzata per rappresentare un report di validità di un green pass inviato dall'CLIENTT. Essa contiene il numero di tessera sanitaria del possessore del green pass e un flag che indica se il green pass è valido o meno.

Il client utilizza le funzioni `full_write()` e `full_read()` per inviare e ricevere i dati in modo affidabile.

La funzione "handler" è utilizzata per catturare il segnale SIGINT (CTRL-C) e stampare un messaggio di arrivederci. In particolare, se viene catturato il segnale SIGINT, la funzione stampa "Uscita in corso.", attende 2 secondi e poi stampa "Arrivederci!".

La funzione "crea\_data\_inizio" viene utilizzata per ottenere la data corrente del sistema. Essa utilizza la funzione "time" per ottenere il numero di secondi trascorsi dal 1 gennaio 1970 e poi utilizza la funzione "localtime" per convertire questo valore in una struttura "tm" che rappresenta la data e l'ora locali. Infine, la funzione estrapola giorno, mese e anno dalla struttura "tm" e li assegna ai parametri di ritorno.

La funzione "verifica\_ID" implementa la logica dell'app. Essa riceve in input il numero di tessera sanitaria di un Client e poi richiede al ServerV il report di validità del suo green pass.

Successivamente, la funzione effettua alcune procedure di verifica per determinare se il green pass è valido o meno e comunica l'esito all'c. In particolare, se il green pass è valido, la funzione ritorna il valore "1". Altrimenti, la funzione ritorna il valore "0".

## Manuale Client – Guida alla compilazione ed esecuzione

### *Compilazione*

Per compilare i vari componenti del progetto, si deve aprire il terminale e posizionarsi nella cartella "Progetto GreenPass". Successivamente, è necessario digitare il comando di compilazione corrispondente al componente che si intende compilare.

#### **Per il Client:**

*gcc -o Client Client.c*

#### **Per il Centro Vaccinale:**

*gcc -o CentroVaccinale CentroVaccinale.c*

#### **Per il ServerV:**

*gcc -o ServerV ServerV.c*

#### **Per il ServerG:**

*gcc -o ServerG ServerG.c*

#### **Per il ClientS:**

*gcc -o ClientS ClientS.c*

#### **Per ClientT:**

*gcc -o ClientT ClientT.c*

## ***Esecuzione***

Per eseguire il software desiderato, bisogna digitare il comando corrispondente al componente che si vuole eseguire. Si consiglia di aprire un terminale e posizionarsi nella cartella "Progetto GreenPass" prima di eseguire i comandi.

### **Per il Client:**

*./Client localhost*

Per il Centro Vaccinale:

*./CentroVaccinale*

### **Per il ServerV:**

*./ServerV*

### **Per il ServerG:**

*./ServerG*

### **Per ClientS:**

*./ClientS*

### **Per ClientT:**

*./ClientT*

Per garantire una **corretta esecuzione** dei 6 codici, si raccomanda di eseguirli in sequenza:

**ServerV, CentroVaccinale, ServerG, Client, ClientS, ClientT.**

Ecco la simulazione del programma, eseguendo correttamente i passaggi indicati.

Inizialmente, avviamo CentroVaccinale, seguito da ServerG e infine ServerV, i quali rimarranno costantemente in ascolto per ricevere e gestire le richieste di connessione e servizio.

Il nostro programma Client è stato sviluppato per connettersi al CentroVaccinale, dove verrà compilato un form e successivamente inviato. Una volta ricevuto, il CentroVaccinale trasmetterà il pacchetto al ServerV, che creerà un file univoco avente come nome il codice di tessera sanitaria del client. All'interno del file saranno registrate le informazioni riguardanti la data di inizio e di scadenza del Green Pass, la cui validità è di 3 mesi.

```
fabio@fabio-VirtualBox:~/Immagini/Progetto2402_2$ ./Client localhost
Benvenuto nel centro vaccinale di Firenze
Inserisci il tuo nome, cognome e numero di tessera sanitaria per la registrazione

Inserisci cognome: Crescenzo
Inserisci nome: Cerqua
Inserire tessera sanitaria (Massimo 10 caratteri): 0123456786

Dati registrati con successo!

fabio@fabio-VirtualBox:~/Immagini/Progetto2402_2$
```

Ora il CentroVaccinale rimane in attesa di richieste di vaccinazione. Una volta ricevuta la richiesta, il CentroVaccinale invia un modulo contenente le informazioni richieste e stampa il suo contenuto. Inoltre, la finestra visualizza la data di inizio e fine validità del certificato di vaccinazione.

```
fabio@fabio-VirtualBox:~/Immagini/Progetto2402_2$ ./CentroVaccinale
In attesa di nuove richieste di vaccinazione...
In attesa di nuove richieste di vaccinazione...

Informazioni ricevute
Nome: Cerqua
Cognome: Crescenzo
Numero Tessera Sanitaria: 0123456786

Data dell'inizio validità Green Pass: 24:02:2023
Data scadenza Green Pass: 24:05:2023

```

Per verificare la validità del GreenPass appena generato, possiamo utilizzare ClientS per effettuare la scansione del documento.

```
fabio@fabio-VirtualBox:~/Immagini/Progetto2402_2$ ./ClientS
Benvenuto nel server di verifica
Inserisci il numero della tessera sanitaria per verificarne la validità.

Inserisci il codice della tessera sanitaria. (Massimo 10 caratteri): 0123456786
numero di tessera sanitaria ricevuto

Convalida in corso! Per favore attendere...

Green Pass valido, operazione conclusa
fabio@fabio-VirtualBox:~/Immagini/Progetto2402_2$
```

Il GreenPass è valido in quanto la scansione ha avuto un esito positivo.

Andiamo a invalidare il GreenPass appena creato, comunicando al ServerV un esito di tampone positivo.

```
fabio@fabio-VirtualBox:~/Immagini/Progetto2402_2$ ./ClientT
Inserire codice di tessera sanitaria e referto del tampone per invalidare o ripristinare Green Pass.
Inserisci il codice tessera sanitaria (max 10 caratteri): 0123456786
Inserisci 0 per invalidare il Green Pass o 1 per ripristinarlo: 0

Attendi. Stiamo inviando la richiesta per sospendere il Green Pass inserito...
Operazione conclusa con successo!
```

```
fabio@fabio-VirtualBox:~/Immagini/Progetto2402_2$
```

Verifichiamo se l'operazione è stata eseguita con successo, effettuando la scansione del GreenPass tramite l'applicazione di verifica.

```
fabio@fabio-VirtualBox:~/Immagini/Progetto2402_2$ ./ClientS
Benvenuto nel server di verifica
Inserisci il numero della tessera sanitaria per verificarne la validità.

Inserisci il codice della tessera sanitaria. (Massimo 10 caratteri): 0123456786
numero di tessera sanitaria ricevuto

Convalida in corso! Per favore attendere...

Il Green Pass non è valido!
```

```
fabio@fabio-VirtualBox:~/Immagini/Progetto2402_2$ █
```