

UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

SCUOLA INTERDIPARTIMENTALE DELLE SCIENZE, DELL'INGEGNERIA E DELLA SALUTE

INFORMATICA



## Progetto Laboratorio Reti di Calcolatori

**Proponenti:**

Ferraro Dominick      0124002048

Mungari Alfredo      0124002134

**Data di Consegna:**

15/02/2022

**Anno Accademico:**

2021 – 2022

**Docente:**

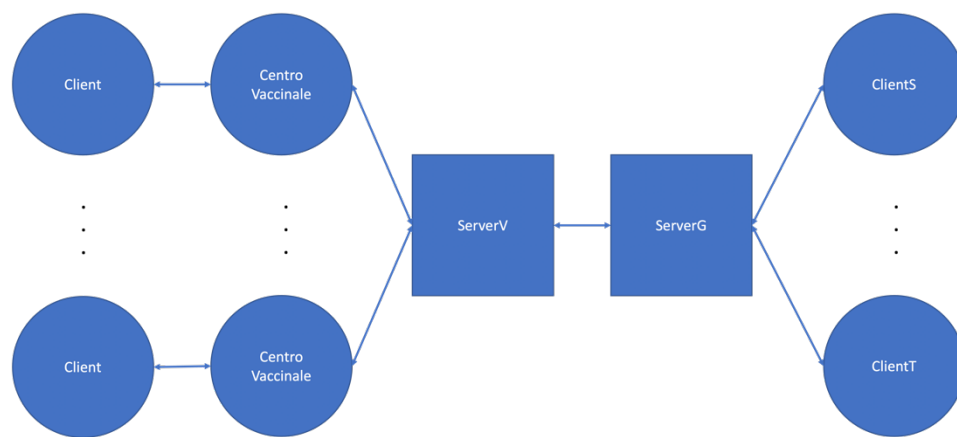
Prof. A. Ferone

# Indice

Descrizione generale del progetto .....	3
Descrizione dettagliata .....	3
Client → Utente.....	4
Client → Centro Vaccinale .....	4
ServerV → Server Vaccinale .....	6
ServerG → ServerVerifica .....	7
ClientS → App Green Pass .....	7
ClientT → ASL.....	8
Descrizione del Modello .....	9
Descrizione del Protocollo .....	10
Comunicazione User → CentroVaccinale .....	10
Comunicazione CentroVaccinale→ ServerVaccinale .....	11
Comunicazione AppVerifica → ServerVerifica .....	12
Comunicazione ServerVerifica→ ServerVaccinale .....	13
Comunicazione ASL → ServerVerifica .....	14
Manuale Utente – Guida alla compilazione ed esecuzione.....	15
Simulazione dell'applicazione .....	17
Mapping Hardware Software.....	20
Gestione dei dati persistenti e concorrenza .....	20

## Descrizione generale del progetto

Un client si connette a un Centro Vaccinale comunicando la propria tessera sanitaria. Il Centro Vaccinale associa alla tessera sanitaria una data di scadenza del Green Pass e la invia al ServerV. Dall'altro lato abbiamo un ClientS che invia un codice di tessera sanitaria a un ServerG che a sua volta lo invia al ServerV per accettarsi che un Green Pass sia valido: il ServerV salva in un filesystem tutti i Green Pass e invia un GP al ServerG quando lo richiede, il ServerG esegue l'operazione di verifica validità e lo comunica al ClientS. Infine abbiamo un ClientT che comunica col ServerG e può ripristinare o invalidare un Green Pass.



## Descrizione dettagliata

Il progetto proposto rappresenta tutto il mini-mondo per la gestione dei certificati vaccinali, cioè il Green Pass.

Un utente, dopo aver effettuato la vaccinazione, comunica i propri dati anagrafici e il numero di tessera sanitaria a un Centro Vaccinale, il quale innanzitutto comunicherà l'eventuale ricezione dei dati al cliente e invierà poi il codice della tessera sanitaria, con il periodo di validità del Green Pass a un ServerVaccinale che svolge un ruolo di database dove salva in un filesystem tutti i certificati verdi.

Abbiamo poi un ClientS che può essere visto come l'app che scansiona i Green Pass, ad esempio *Verifica C19*, che invia un codice di una tessera sanitaria al ServerG che a sua volta chiede al ServerV di inviargli un Green Pass: così facendo il ServerVerifica effettuerà l'operazione di scansione della validità per poi comunicarlo al ClientS.

Infine abbiamo un ClientT, identificabile come un'organizzazione sanitaria, come l'*ASL*, che può invalidare o ripristinare la validità di un Green Pass comunicando il contagio o la guarigione di una persona al ServerG mediante il codice della tessera sanitaria.

## Client → Utente

Il Client (Utente) si connette al CentroVaccinale. Dopo aver stabilito una connessione, l'Utente riceve un messaggio di benvenuto dal CentroVaccinale per notificargli l'avvenuta connessione. Successivamente, il CentroVaccinale chiede al Client Utente di compilare un form contenente nome, cognome e numero di tessera sanitaria che deve essere esattamente di 10 caratteri.

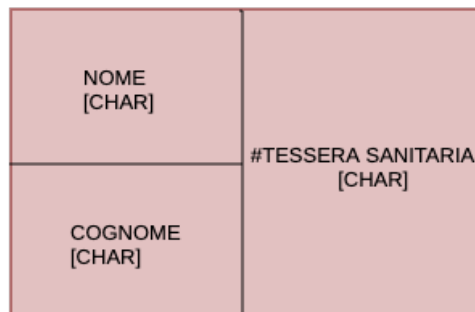
Il Client Utente compila questo form e lo invia al CentroVaccinale che invierà un altro messaggio di ricezione avvenuta.

## Pacchetto Applicazione → Utente

Andiamo a definire un pacchetto *Vax\_Request* usato dall'Utente che invierà al Centro Vaccinale per generare un certificato verde.

```
typedef struct {  
    char name[MAX_SIZE];  
    char surname[MAX_SIZE];  
    char ID[ID_SIZE];  
} VAX_REQUEST;
```

*Vax\_Request* è una struttura contenente 3 campi: nome, cognome e numero di tessera sanitaria dell'utente. La dimensione massima di nome e cognome è MAX\_SIZE (1024 byte) mentre la dimensione del numero di tessera sanitaria è ID\_SIZE(11 byte), poiché contiamo anche il byte del carattere terminatore.



---

## Client → Centro Vaccinale

Il CentroVaccinale accetta connessioni dal ClientUtente. Dopo aver stabilito una connessione, invia un messaggio di benvenuto al ClientUtente e gli chiede di compilare un form contenente nome, cognome e numero di tessera sanitaria dell'utente. Dopo aver ricevuto questi dati, invia un messaggio di avvenuta ricezione all'Utente.

Successivamente il CentroVaccinale si connette al ServerVaccinale per inviare i dati appena ricevuti e generare un Green Pass. Il CentroVaccinale, ha quindi ricevuto il

pacchetto dall'utente e deve generare un GP da inviare al ServerVaccinale. Il CentroVaccinale prende la *data\_inizio* e gli aggiunge 3 mesi, creando così un periodo di validità della certificazione verde. Successivamente il CentroVaccinale invia un pacchetto al ServerVaccinale contenente il numero della tessera sanitaria associata a una data di inizio e fine validità del Green Pass.

## Pacchetto Applicazione → Centro Vaccinale

Per prima cosa andiamo a creare una struttura *DATE* che ci permetterà di salvare una data, contenente i campi giorno, mese, anno.

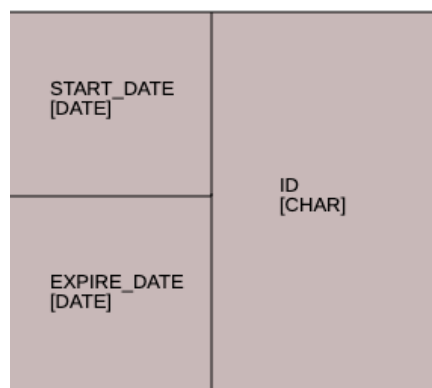
```
typedef struct {  
    int day;  
    int month;  
    int year;  
} DATE;
```

DATE sarà necessaria per creare il pacchetto applicazione del Centro Vaccinale che invierà al Server Vaccinale per generare un Green Pass, associandogli una data iniziale e di scadenza.

Definiamo quindi la struttura *GP\_REQUEST*: pacchetto inviato dal centro vaccinale al server vaccinale contenente il numero di tessera sanitaria dell'utente, la data di inizio e fine validità del GP.

```
typedef struct {  
    char ID[ID_SIZE];  
    DATE start_date;  
    DATE expire_date;  
} GP_REQUEST;
```

Il campo ID ha dimensione *ID\_SIZE* (11 byte) mentre i due campi *start\_date*, *expire\_date* sono di tipo *DATE* e sono interi.



## ServerV → Server Vaccinale

Il ServerVaccinale resta in attesa di connessioni da parte del CentroVaccinale, dopo aver accettato la connessione, attenderà la ricezione dei dati.

Una volta arrivati i dati, il ServerVaccinale creerà un file univoco per ogni Green Pass, rinominando il file col numero di tessera sanitaria ricevuto dal CentroVaccinale e contenente data inizio e fine validità del GP.

Poiché il ServerVaccinale svolge anche la funzione di database, fornisce un supporto molto importante per la scansione della validità di un Green Pass. Per scelte progettuali, si è deciso di utilizzare un filesystem per simulare una connessione al DB: i file saranno comunque persistenti allo shutdown dell'applicazione.

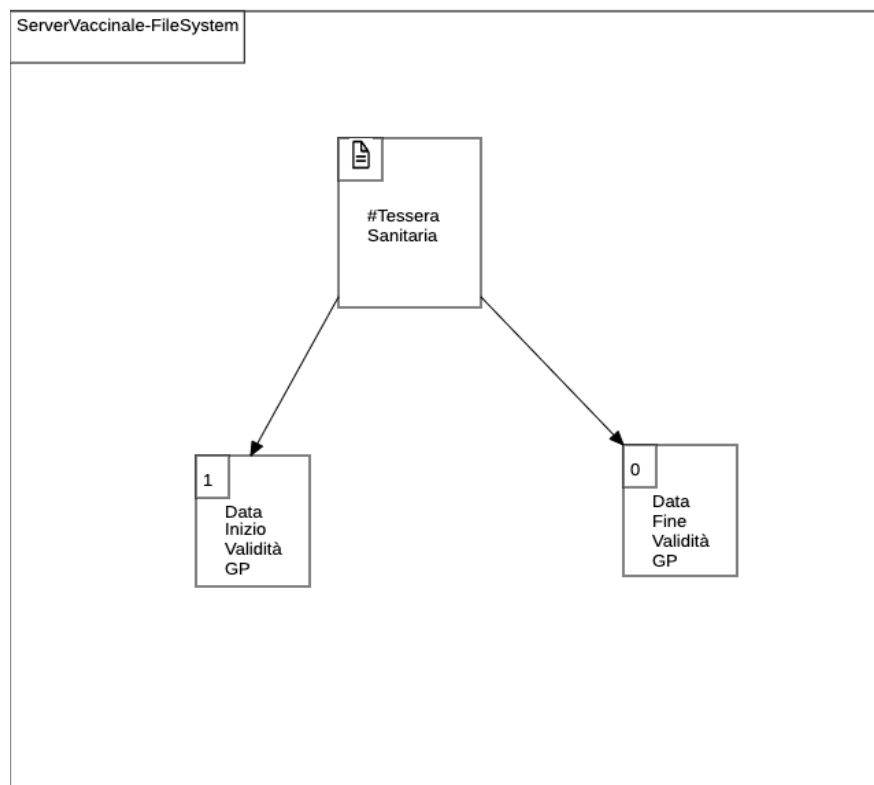
Il ServerVaccinale viene contattato dal ServerVerifica che gli richiede un GP specifico.

Il ServerVaccinale quindi effettua una “query” per trovare il GP richiesto, grazie al numero di tessera sanitario salvato. Se il file esiste, lo invierà al ServerVerifica in modo da poter svolgere le dovute operazioni.

Per gestire la comunicazione sia col CentroVaccinale che col ServerVerifica si è adottata la seguente convenzione: al momento della connessione col ServerVaccinale, sia il CentroVaccinale che il ServerVerifica inviano un bit di riconoscimento, così facendo il ServerVaccinale saprà chi ha richiesto una connessione.

Se il bit inviato ha valore “1” allora il ServerVaccinale gestirà la connessione col CentroVaccinale.

Se il bit inviato ha valore “0” allora il ServerVaccinale gestirà la connessione col ServerVerifica.



Inoltre, il ServerVaccinale eseguirà un'ulteriore funzione: quella di modificare la validità di un certificato verde seguendo le disposizioni dettate dal ClientT (ASL).

## ServerG → ServerVerifica

Il ServerVerifica resta in attesa di connessioni da parte del ClientS (App Scansione), dopo aver accettato la connessione invierà un messaggio di benvenuto e un form, dove attenderà la ricezione di un numero di tessera sanitaria associata a un certificato verde da verificare.

Una volta arrivati i dati, il ServerVerifica si collegherà al ServerVaccinale, inviandogli il numero di tessera sanitaria. Fatto ciò, il ServerVaccinale invierà il Green Pass richiesto, se esistente, al ServerVerifica in modo da poterne testare la validità: se il GP è valido invia un esito positivo al ClientS, oppure notifica al ClientS che il GP non è valido o non esiste.

---

## ClientS → App Green Pass

Il ClientS può essere visto come un'app per scansionare un GP per testarne la validità. ClientS si connette al ServerVerifica, riceve un messaggio di benvenuto dal ServerVerifica che gli chiederà di immettere un numero di tessera sanitaria associata a un Green Pass, in modo da poterne testare la validità. Dopo aver inserito il codice, l'App resta in attesa che il ServerVerifica e ServerVaccinale facciano le dovute operazioni e riceve un esito positivo o negativo da parte del ServerVerifica.



## ClientT → ASL

Il ClientT può essere visto come un ente pubblico, ad esempio una ASL.

Il ClientT ha la possibilità di invalidare o ripristinare un Green Pass a seconda se la persona associata al numero di tessera sanitaria risulta positiva o è guarita.

Il ClientT, quindi, comunica un referto di un tampone al ServerVerifica, il ServerVerifica si interfaccia col ServerVaccinale che andrà a modificare il file interessato, in particolare il campo “report”: se il ClientT invierà un “1” il Green Pass sarà invalidato, se il ClientT invierà uno “0” il Green Pass sarà ripristinato.

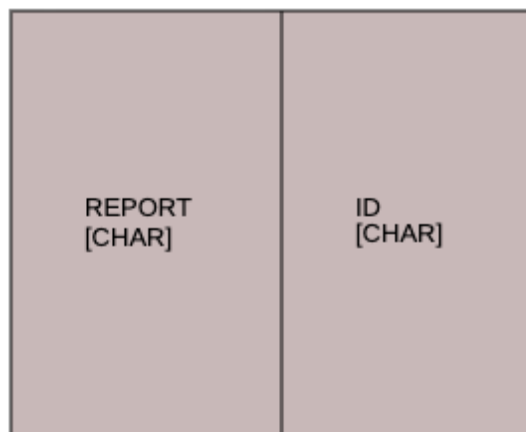
Molto importante qui è l’App Green Pass (ClientS) che permette di testare il corretto funzionamento del ClientT, che sarà sempre aggiornata delle variazioni di validità dei certificati verdi.

## Pacchetto Applicazione → ASL

Definiamo una struttura *REPORT*: è il pacchetto applicazione che l’ASL invierà al Server Verifica contenente un numero di tessera sanitarie e un referto associato a un Green Pass.

```
typedef struct {  
    char ID[ID_SIZE];  
    char report;  
} REPORT;
```

Il campo ID ha dimensione ID\_SIZE (11 byte), il campo REPORT ha dimensione 1 byte (char).



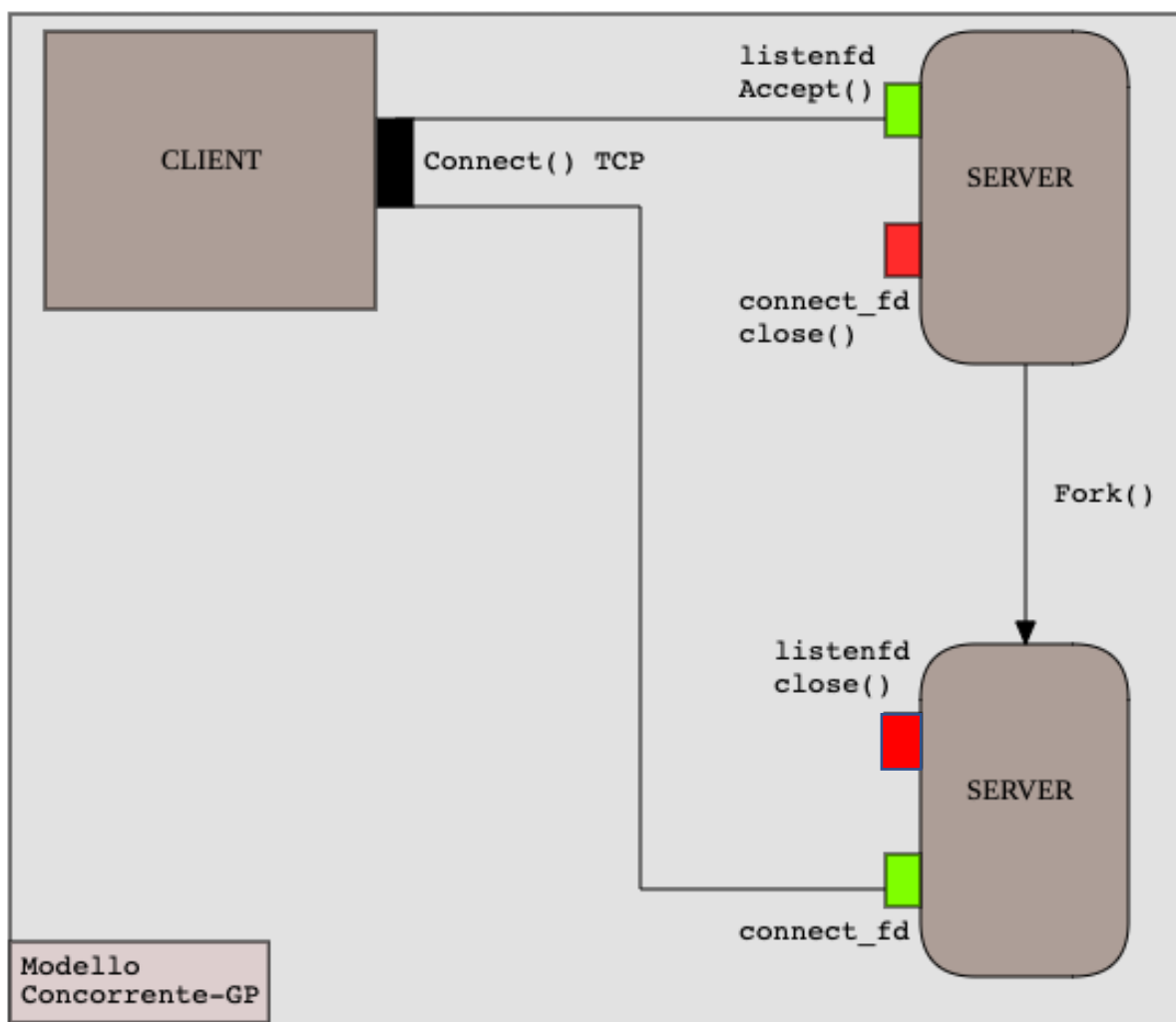


## Descrizione del Modello

Il modello di programmazione scelto per la realizzazione del progetto è quello client-server.

Il server utilizzato è **concorrente**, cioè è in grado di fornire servizi a più client contemporaneamente, questo è realizzato tramite l'uso della systemcall *fork()* che crea  $N$  processi figli tanti quanti sono le connessioni dei client che vengono accettate dai server.

Il padre gestirà il descrittore *listenfd* per accettare le richieste di connessione, mentre un figlio gestirà il descrittore restituito dalla *connect()*, cioè *connfd* per fornire servizi al client connesso.



## Descrizione del Protocollo

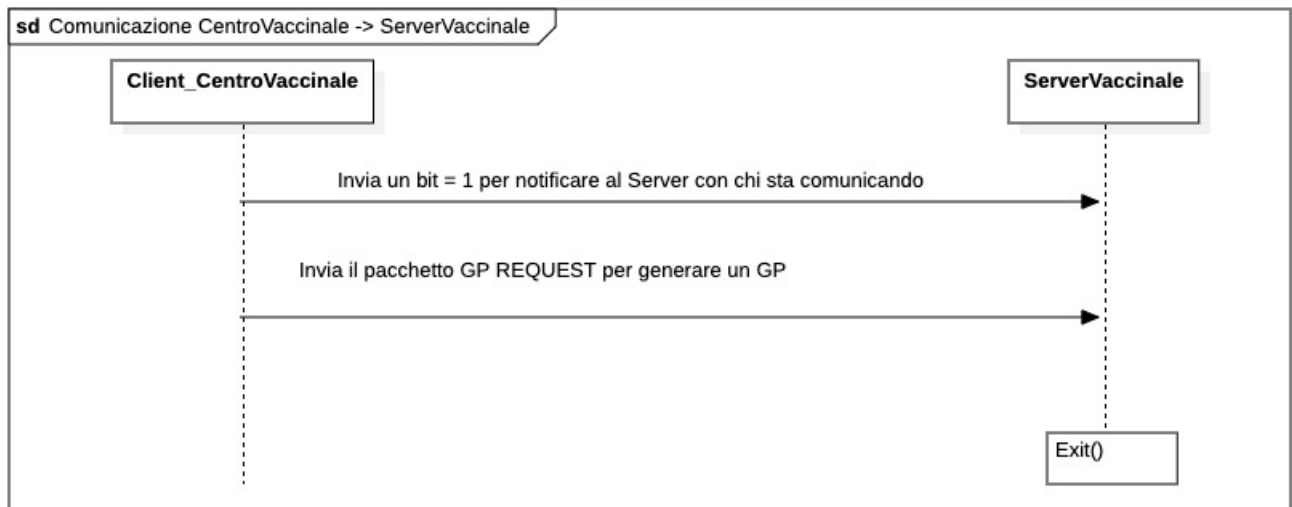
Il protocollo usato per il progetto è TCP socket. Poiché è un protocollo connection oriented, prima di poter trasmettere dati, deve stabilire una connessione.

Di seguito vi è riportato lo schema di funzionamento della trasmissione dati tra i vari client e server.

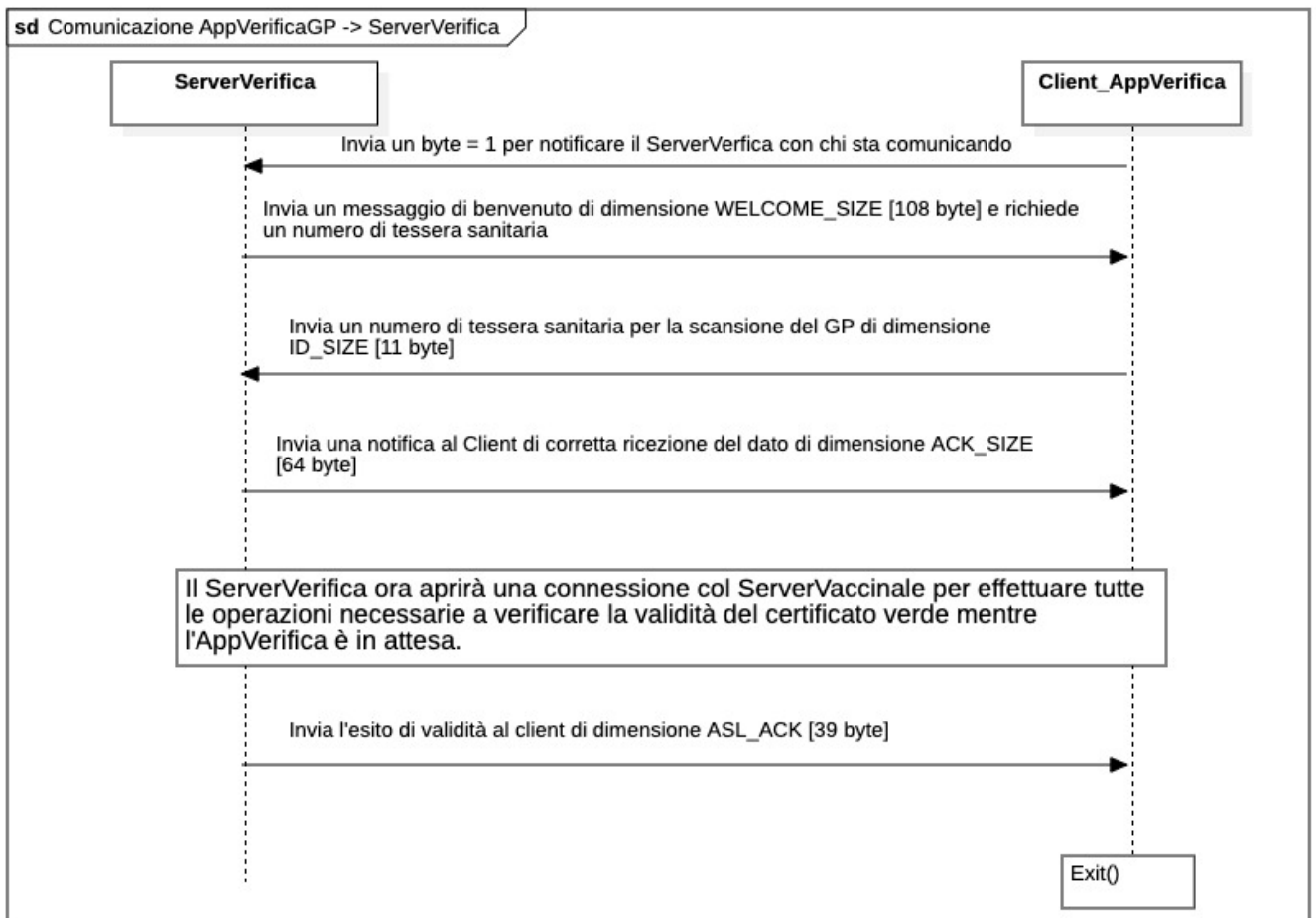
### Comunicazione User → CentroVaccinale



## Comunicazione CentroVaccinale → ServerVaccinale

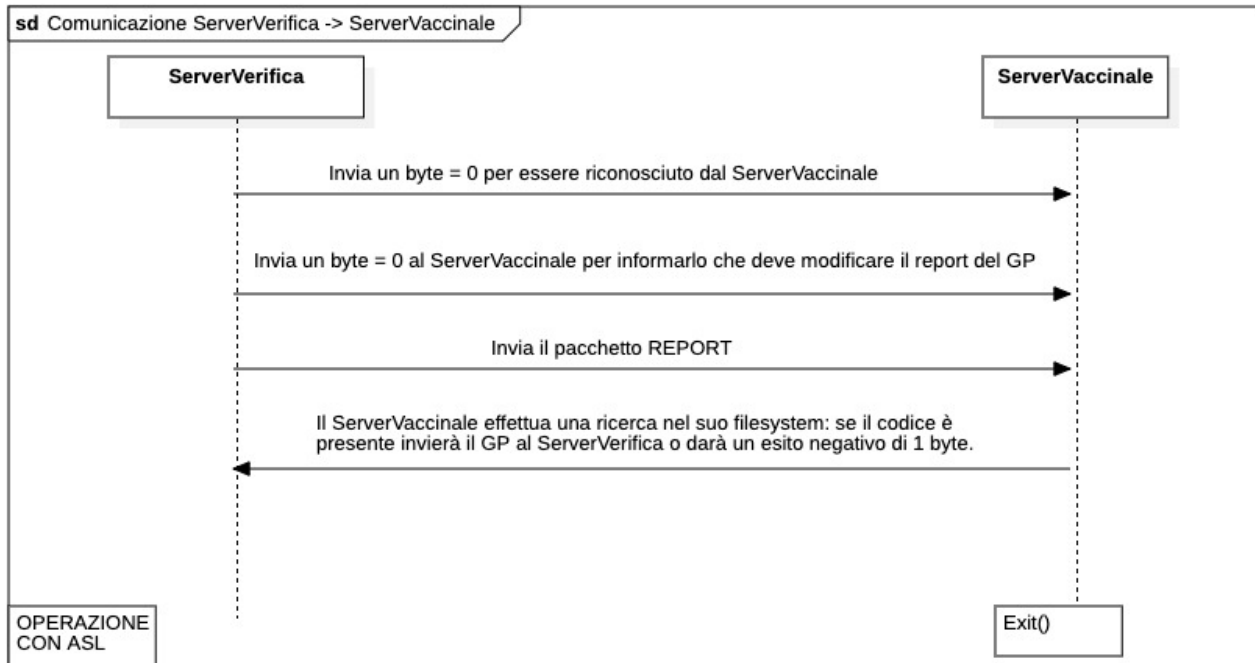


## Comunicazione AppVerifica → ServerVerifica

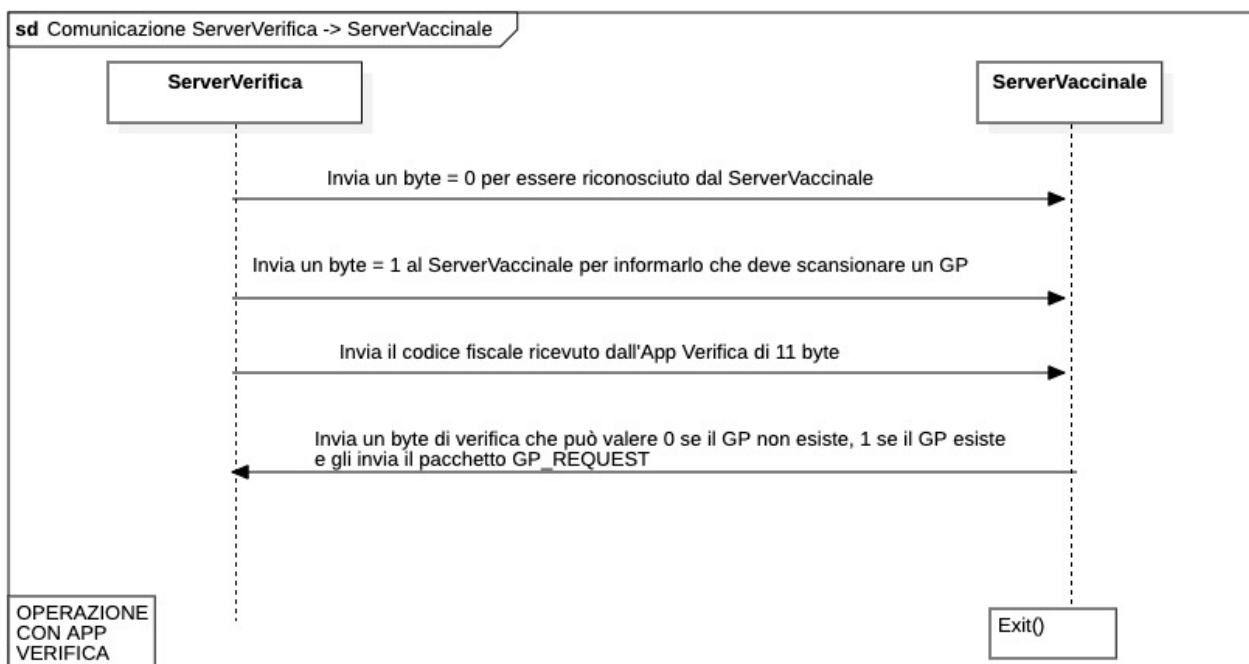


## Comunicazione ServerVerifica → ServerVaccinale

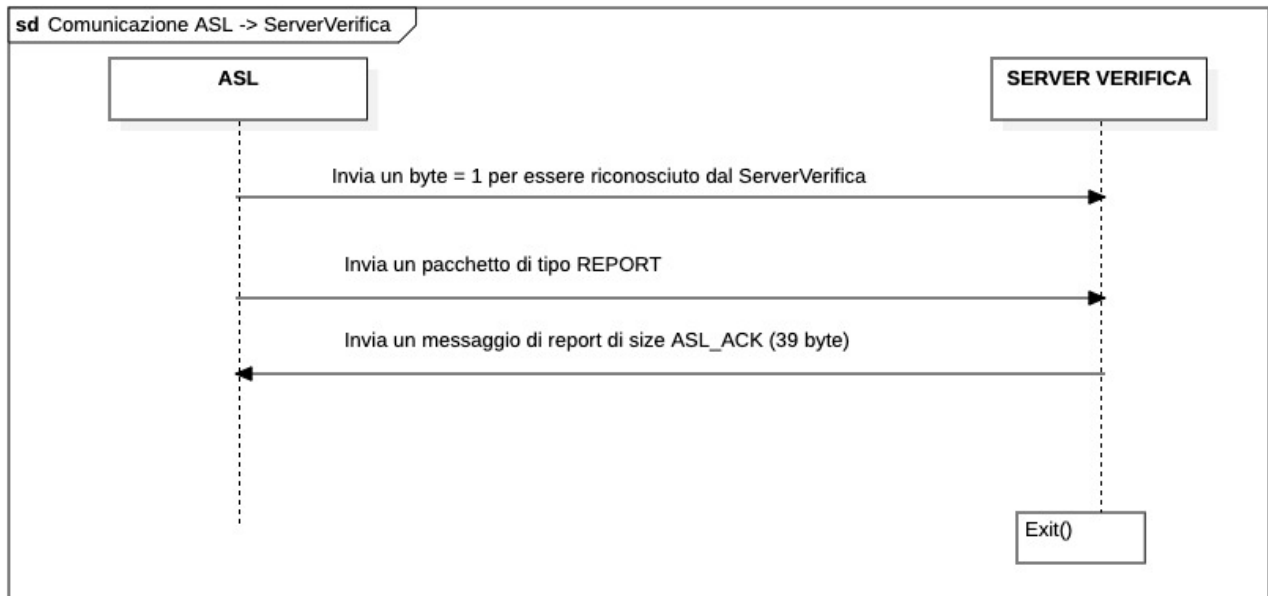
ASL



APP VERIFICA



## Comunicazione ASL → ServerVerifica



# Manuale Utente – Guida alla compilazione ed esecuzione

## *Compilazione*

- Per compilare il *Client\_Utente* , aprire il terminale dirigersi nella cartella “Progetto GP” e digitare il seguente comando:

*gcc -o Utente Utente.c*

- Per compilare il *Centro Vaccinale* , aprire il terminale dirigersi nella cartella “Progetto GP” e digitare il seguente comando:

*gcc -o CentroVaccinale CentroVaccinale.c*

- Per compilare il *ServerVaccinale* aprire il terminale dirigersi nella cartella “Progetto GP” e digitare il seguente comando:

*gcc -o ServerVaccinale ServerVaccinale.c*

- Per compilare il *ServerVerifica*, aprire il terminale dirigersi nella cartella “Progetto GP” e digitare il seguente comando:

*gcc -o ServerVerifica ServerVerifica.c*

- Per compilare l'*AppScansione*, aprire il terminale dirigersi nella cartella “Progetto GP” e digitare il seguente comando:

*gcc -o AppScansione AppScansione.c*

- Per compilare l'*ASL*, aprire il terminale dirigersi nella cartella “Progetto GP” e digitare il seguente comando:

*gcc -o ASL ASL.c*

## *Esecuzione*

- Per eseguire il *Client\_Utente* , digitare il seguente comando:

*./Utente localhost*

- Per eseguire il *Centro Vaccinale*, digitare il seguente comando:

*./CentroVaccinale*

- Per eseguire il *ServerVaccinale*, digitare il seguente comando:

*./ServerVaccinale*

- Per eseguire il *ServerVerifica*, digitare il seguente comando:

*./ServerVerifica*

- Per eseguire l'*AppScansione*, digitare il seguente comando:

*./AppScansione*

- Per eseguire l'*ASL*, digitare il seguente comando:

*./ASL*

## *Ordine di esecuzione*

Per una corretta esecuzione dei 6 codici si consiglia di eseguire in ordine:

ServerVaccinale, CentroVaccinale, ServerVerifica, Utente, AppScansione, ASL.



## Simulazione dell'applicazione

In seguito dimostriamo il funzionamento del progetto, useremo 6 terminali.

Per prima cosa lanciamo il CentroVaccinale, ServerVerifica e ServerVaccinale che resteranno sempre in ascolto per accettare e gestire le connessioni e servizi.

Nella figura in basso lanciamo il programma UTENTE, che si conetterà al CentroVaccinale, compilerà un form e lo invierà a quest'ultimo. Successivamente il CentroVaccinale invierà il pacchetto al ServerVaccinale, che creerà un file univoco che ha come nome il numero di tessera sanitaria dell'utente e al suo interno vi sono data inizio validità e di scadenza del Green Pass. Ricordiamo che la validità del certificato verde è di 3 mesi in questo contesto.

```
MacBook-Pro-di-Dominick:Codice dominickferraro$ ./Utente localhost
***Benvenuto nel centro vaccinale di Bari***
Inserisci nome, cognome e numero di tessera sanitaria per inserirli sulla piattaforma.

Inserisci nome: Mario
Inserisci cognome: Rossi
Inserisci codice tessera sanitaria [Massimo 10 caratteri]: 0123456789
I tuoi dati sono stati correttamente inseriti in piattaforma
```

Nella figura in basso abbiamo la finestra del CentroVaccinale che è in ascolto in attesa di richieste di vaccinazione. Quando ne riceve una, invia il form e stampa il contenuto di questo, inoltre mostrerà la data di inizio e fine validità del GP.

```
In attesa di nuove richieste di vaccinazione

Dati ricevuti
Nome: Mario
Cognome: Rossi
Numero Tessera Sanitaria: 0123456789

La data di inizio validità del green pass e': 09:02:2022
La data di scadenza del green pass e': 09:05:2022
```

Dall'altro lato abbiamo l'APP per scansionare il GP: proviamo a vedere se il GP appena creato è valido:

```
MacBook-Pro-di-Dominick:Codice dominickferraro$ ./APP
***Benvenuto nel server di verifica***
Inserisci il numero di tessera sanitaria per testarne la validità.

Inserisci codice tessera sanitaria [Massimo 10 caratteri]: 0123456789

Il numero di tessera sanitaria è stato correttamente ricevuto

Convalida in corso, attendere...

Green Pass valido, operazione conclusa
```

Come ci si aspettava, la scansione ha avuto buon fine, il certificato verde è valido.

Proviamo adesso a vedere come opera l'ASL: andiamo a invalidare il GP creato, comunicando al ServerVerifica che l'esito del tampone è positivo.

```
MacBook-Pro-di-Dominick:Codice dominickferraro$ ./ASL
*** ASL ***
Immettere un numero di tessera sanitaria ed il referto di un tampone per invalidare o
ripristinare un Green Pass
Inserisci codice tessera sanitaria [Massimo 10 caratteri]: 0123456789
Inserire 0 [Green Pass Non Valido]
Inserire 1 [Green Pass Valido]
[INPUT]: 0

Invio richiesta di sospensione Green Pass...
***Operazione avvenuta con successo***
```

Proviamo adesso a vedere se l'operazione è andata a buon fine, scansionando il GP con l'APP di verifica:

```
MacBook-Pro-di-Dominick:Codice dominickferraro$ ./APP
***Benvenuto nel server di verifica***
Inserisci il numero di tessera sanitaria per testarne la validità.

Inserisci codice tessera sanitaria [Massimo 10 caratteri]: 0123456789

Il numero di tessera sanitaria è stato correttamente ricevuto

Convalida in corso, attendere...

Green Pass non valido, uscita in corso
```

Come vediamo l'operazione è stata completata con successo.

## Glossario

Il glossario ha lo scopo fondamentale di chiarire il gergo tecnico usato e di evidenziare eventuali sinonimie e omonimie. Trattandosi di un contesto **sanitario**, la maggioranza dei termini riguardano tale ambito, le informazioni riportate valgono per lo stato italiano. È possibile che in altri Paesi, tali termini tradotti letteralmente possono essere utilizzati in contesti che differiscono da quelli di nostro interesse.

Termine	Descrizione	Sinonimi	Omonimi
ASL	Acronimo di Azienda Sanitaria Locale. È un ente pubblico della pubblica amministrazione italiana, deputato all'erogazione di servizi sanitari in un determinato territorio, di solito provinciale.	ASP, ASM, ATS.	-
Centro Vaccinale	I centri vaccinali sono presidi sanitari territoriali che somministrano le vaccinazioni previste dal Piano nazionale prevenzione vaccinale.	Hub vaccinale.	-
App per scansionare GP	App che scansiona un Green Pass tramite un codice QR per attestare la validità.	-	-
Green Pass	Il Green Pass è un'attestazione digitale introdotta nel 2021, durante la pandemia di COVID-19.	GP, Certificato Verde.	-
Query	Interrogazione di una base di dati da parte di un utente.	Interrogazione, richiesta	-
QR CODE	QR Code è un codice a barre a risposta rapida. Si tratta di un simbolo che restituisce, ogni qualvolta viene inquadrato dalla fotocamera di uno smartphone, dati e informazioni all'utente.	QR, Quick Response Code	-
Tampone	Materiale assorbente fissato su una bacchetta per praticare la ricerca e coltura di microrganismi patogeni da cavità del corpo.	-	-

# Mapping Hardware Software

Il sistema è pensato per poter essere eseguito su qualsiasi device supportante un OS Unix/Linux e Unix-like.

Uno scenario che si può immaginare è una macchina server sulla quale è presente la parte ServerVaccinale del sistema software contenente il database (filesystem) del sistema e le sue funzioni.

A tale macchina server è possibile collegarsi mediante la rete da un device dell'utente che può avere prestazioni più o meno avanzate.

In ogni caso il segmento utente è necessario che abbia la conoscenza dell'indirizzo (IP-PORTA) del CentroVaccinale per poter effettuare le operazioni di creazione del certificato verde.

Inoltre, per l'elaborazione del database in tempo reale con la continua richiesta di certificati verdi dei centri vaccinali, sarà necessario anche un buon quantitativo di potenza in termini di processore e memoria in merito alle macchine sulle quali è installato il segmento client.

Qualora dovesse essere necessario, il tasso di risposta desiderato dal sistema può essere ottenuto con il potenziamento dell'hardware esistente/presente e dell'ampiezza di banda di comunicazione.

---

## Gestione dei dati persistenti e concorrenza

Il progetto “Reti Green Pass” prevede la memorizzazione di numerosi dati in maniera persistente, l'interfacciamento, la sincronizzazione e la notifica di più attori coinvolti.

Siamo coscienti del fatto che la miglior scelta per la gestione di dati persistenti è un database relazione che offre anche la gestione della concorrenza, gestendo gli accessi ai dati condivisi in mutua esclusione, ma per ragioni progettuali, vincoli e requisiti non funzionali di sistema, si è deciso di adottare un file system, al fine di mantenere traccia di tutti i dati generati, richiesti, da elaborare, da trasmettere e per gestire in maniera più semplice ed efficiente gli accessi concorrenti.

Per implementare la concorrenza utilizzando un filesystem, si è adottata una particolare systemcall (***flock()***), per gestire gli accessi concorrenti in lettura e scrittura ai file dei singoli GP.

Da sottolineare che questo aspetto di gestione dei dati rappresenta il fulcro di tutto il progetto poiché è essenziale che ogni certificato sia univoco, persistente e sicuro.