

**Progetto Centri Vaccinali**  
Laboratorio Interdisciplinare B

# **Centro Operatori**

## **Manuale Tecnico**

(Manuale) Preparato da

**Daniel Satriano**

**Mat. 745232**

## Sommario

<b>INTRODUZIONE .....</b>	<b>3</b>
<b>Librerie esterne utilizzate .....</b>	<b>3</b>
<b>Informazioni per la comunicazione .....</b>	<b>3</b>
<b>Informazioni per l'avvio.....</b>	<b>3</b>
<b>Aumentate la quantità di RAM dedicata al server.....</b>	<b>4</b>
<b>Terminare il server .....</b>	<b>5</b>
<b>CLASSI .....</b>	<b>5</b>

## INTRODUZIONE

Il **server** è il cervello dell'ecosistema che abbiamo creato, permette la comunicazione tra database e client, senza che il client debba comunicare direttamente con il database.

### Librerie esterne utilizzate

L'applicazione fa uso di una libreria:

- **PostgreSQL – 42.5.0:** Questa libreria permette la comunicazione tra database e server e viene usata per effettuare query e operazioni sul database.

### Informazioni per la comunicazione

Il server può essere avviato su una qualsiasi macchina a patto che il client risieda nella stessa rete locale. Il server non è progettato per la rete WAN, ma per quella LAN.

Le porte RMI del server non possono essere cambiate e sono:

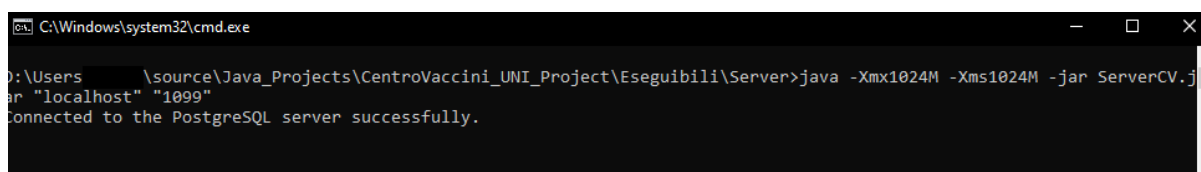
- Portale Operatori: Registry porta 1099
- Portale Cittadini: Registry porta 7272

Il server utilizza due Registry diversi per la gestione delle chiamate in entrata dai due progetti, in questo modo i due tipi di servizi rimangono separati e indipendenti.

### Informazioni per l'avvio

Il server deve essere necessariamente avviato prima dell'avvio dei client, altrimenti quest'ultimi non riusciranno a connettersi e ad effettuare operazioni.

È inoltre necessario avviare il server utilizzando l'apposito file denominato "startServer.bat" il quale vi permetterà di avere una console visiva di quello che sta facendo il server come da immagine [Figura 1].

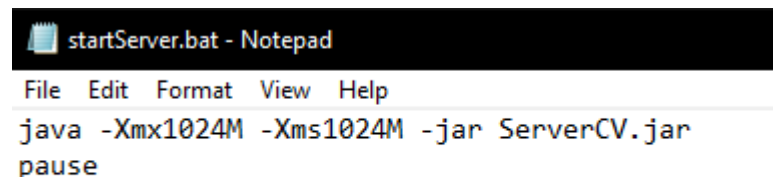


```
C:\Windows\system32\cmd.exe
D:\Users\... \source\Java_Projects\CentroVaccini_UNI_Project\Eseguibili\Server>java -Xmx1024M -Xms1024M -jar ServerCV.j
ar "localhost" "1099"
connected to the PostgreSQL server successfully.
```

FIGURA 1

## Aumentate la quantità di RAM dedicata al server

Tramite il file sopracitato “startServer.bat” è possibile, inoltre, definire la quantità massima e minima di RAM che il server può occupare. Attualmente entrambi i parametri sono impostati a 1GB di RAM, ovvero 1024M, per modificare questi valori è necessario cliccare con il tasto destro del mouse su “startServer.bat” e successivamente premere su “edit” o “modifica”. A questo punto vi si aprirà una console che vi permetterà di modificare i parametri per l’esecuzione del programma [Figura 2].



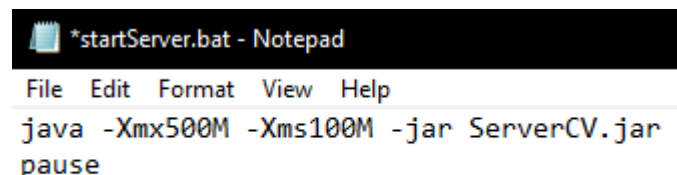
```
startServer.bat - Notepad
File Edit Format View Help
java -Xmx1024M -Xms1024M -jar ServerCV.jar
pause
```

FIGURA 2

Per modificare la quantità massima di RAM è necessario modificare il parametro -Xmx, impostando poi la quantità che vogliamo.

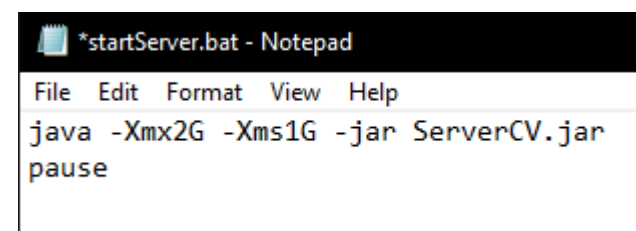
Per modificare la quantità minima di RAM è necessario modificare il parametro -Xms, impostando la quantità desiderata.

A seguito adesso verranno mostrati alcuni esempi [Figura 3] [Figura 4]:



```
*startServer.bat - Notepad
File Edit Format View Help
java -Xmx500M -Xms100M -jar ServerCV.jar
pause
```

FIGURA 3

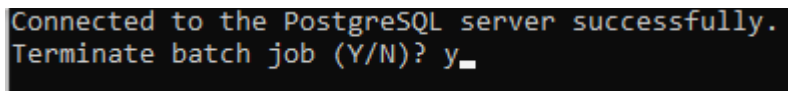


```
*startServer.bat - Notepad
File Edit Format View Help
java -Xmx2G -Xms1G -jar ServerCV.jar
pause
```

FIGURA 4

## Terminare il server

Per terminare il processo del server è necessario andare sulla console del server e premere la combinazione di tasti CTRL+C (su Windows). A questo punto vi verrà mostrato un messaggio [Figura 5] che vi chiede se volete terminare il processo, scrivete y o Y e cliccate invio. A questo punto la finestra si chiuderà e il server sarà chiuso.



```
Connected to the PostgreSQL server successfully.  
Terminate batch job (Y/N)? y_
```

FIGURA 5

## CLASSI

Il server implementa tutte le classi dei client in quanto sono richieste per il passaggio di oggetti personalizzati e due interfacce (intOperators e ICittadiniMetodi). Tutte queste classi non verranno discusse in quanto sono già state discusse ampiamente nei manuali dei corrispettivi programmi.

È bene però citare la classe **PostgressConnection** che si occupa della connessione iniziale al database.

Presenta quattro proprietà:

```
static final String url = "jdbc:postgresql://localhost:5432/ProgettoLabB";  
static final String user = "postgres";  
static final String password = "admin";  
private Connection conn;
```

Presenta inoltre due metodi: Connect() e SQLquery(), intuitivamente connect si occupa della connessione al database e SQLquery si occupa della gestione delle query in entrata e uscita.