

Incontri parrocchiali


Federico Giacomelli

Matricola 578106


Vista dell'interfaccia grafica

○○○

Incontri parrocchiali



Incontri parrocchiali



N	Nome	Cognome	Sesso	Classe
3	Mario	Rossi	M	Prima
6	Luigi	Bianchi	M	Seconda
10	Anna	Gialli	F	Terza
...

Scegli il giorno

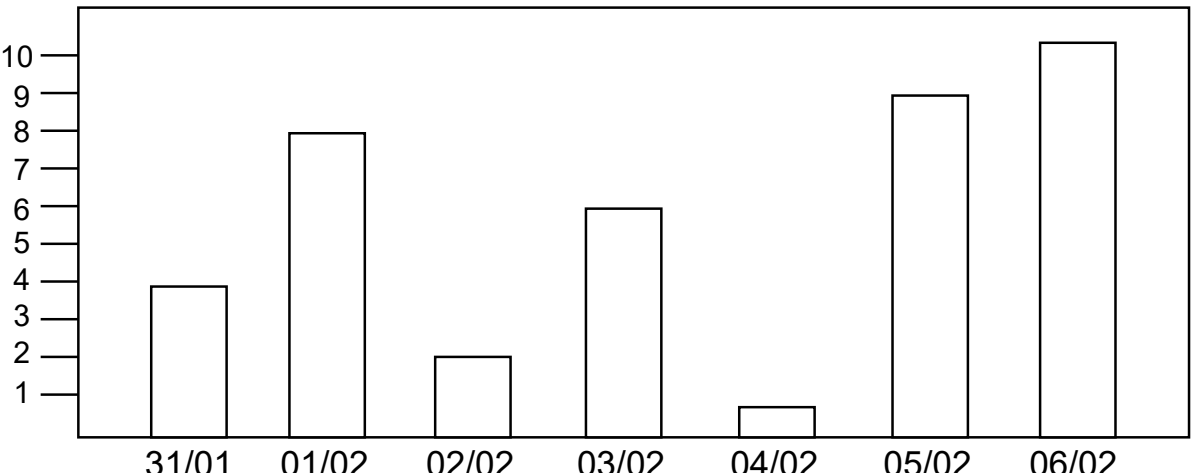
Venerdì 26 Febbraio ▾

Prenota

Rimuovi

Scegli il numero di giorni all'indietro a partire dal giorno scelto

...



31/01	01/02	02/02	03/02	04/02	05/02	06/02
4	8	2	6	1	9	10

Stai visualizzando il numero di presenze giornaliere negli ultimi 7 giorni di **tutti**

Casi d'uso

In caso di **prenotazione**

1. L'Utente seleziona il giorno cui si vuole prenotare.
2. Il Sistema aggiorna la tabella.
3. L'Utente seleziona l'ultima riga della tabella, quella editabile e inserisce i suoi dati.
4. L'Utente inserisce il numero di posto nella riga editabile della tabella
 - 4.1 IF è già presente o supera l'intervallo



- 4.1.1 il Sistema lo notifica impedendolo
- 4.2 IF non è presente ed è compreso nell'intervallo
- 4.2.1 il Sistema permetterà l'operazione
5. L'Utente preme su **Prenota**.
6. Il Sistema aggiorna la tabella, trasformando la riga da editabile a inserita, aggiunge una nuova riga editabile come ultima e aggiorna il grafico di conseguenza.

In caso di **cancellazione** della prenotazione

1. L'Utente seleziona la riga contenente la sua prenotazione.
2. L'Utente preme su **Rimuovi**.
3. Il Sistema elimina la riga corrispondente ed aggiorna la tabella e il grafico.

Indipendentemente dall'inserimento/cancellazione della prenotazione

1. L'Utente seleziona un giorno.
2. Il Sistema aggiorna il grafico mostrando il numero di presenze giornaliere negli ultimi N (scelto) giorni a partire da tale data. Il grafico è un diagramma a barre.
3. L'Utente seleziona una riga della tabella
 - 3.1 IF la riga è non editabile, quindi seleziona una prenotazione
 - 3.1.1 Il Sistema aggiornerà il grafico mostrando le presenze giornaliere del partecipante selezionato negli ultimi N (scelto) giorni.
 - 3.2 IF la riga è editabile
 - 3.2.1 Il Sistema aggiornerà il grafico mostrando il numero di presenze giornaliere negli ultimi N (scelto) giorni a partire da oggi.

File di configurazione locale in XML

All'avvio, il Sistema legge dal file di configurazione locale in XML le seguenti informazioni:

- Giorno della prenotazione.
- Righe della tabella.
- N giorni anteriori a oggi.
- Network Address del client.
- Network Address e porta del server di log.
- User Account, Password, Network Address, TCP/IP Port della connessione con MySQL.

File di log remoto in XML

Ad ogni evento, il Sistema invia una riga di log contenente le seguenti informazioni:

- Nome dell'applicazione.
- Nome dell'evento.
- Network Address del Client.
- Timestamp dell'evento.

Gli eventi classificati come tali sono i seguenti:

- Avvio dell'applicazione ("AVVIO").
- Scelta del giorno dalla prenotazione (tramite DatePicker).
- Modifica di N (tramite TextField).
- Click su **Prenota**
- Click su **Rimuovi**
- Chiusura dell'applicazione ("TERMINE")

Cache locale degli input

Alla chiusura del Sistema,

IF la data è stata cambiata o la riga editabile è stata modificata

Il Sistema salva sul file binario apposito tutte le informazioni.



IF la data non è cambiata o la riga editabile non è stata modificata

Il Sistema non memorizza alcunché perché l'Utente non ha compiuto alcuna operazione di cui è necessario ripristinare coerentemente lo stato.

Gestione archivio

Il Sistema archivia le seguenti informazioni:

- Giorno di prenotazione scelto dall'Utente
- Numero posto dell'Utente
- Nome dell'Utente
- Cognome dell'Utente
- Sesso dell'Utente
- Classe dell'Utente
- Timestamp inserimento prenotazione dell'Utente

Responsabilità delle classi

Classe **IncontriParrocchiali**: contiene tutto il necessario per l'avvio e la gestione degli eventi in tempo reale. Presenta un solo metodo, quello di start, dove, prelevati i parametri locali e i dati in cache, li setta a dovere, crea l'interfaccia grafica, gestisce gli eventi dati dall'interazione con la tabella, la scelta del giorno della prenotazione e la scelta dei giorni all'indietro a partire da oggi cui mostrare i dati nel grafico.

Classe **ConfigurazioneLocale**: contiene i metodi per salvare nuovi parametri in locale, se necessario, ma soprattutto consente di recuperare le varie informazioni dal file XML apposito. Costruisce i parametri facendo uso delle classe Parametri, la quale contiene a monte la struttura del file XML implementando le classi: ParametriPrenotazioni, ParametriNet e ParametriDB. Queste sono mini classi che contengono stringhe o interi intesi come parametri di configurazione che vanno dall'interfaccia grafica, la prima, ai parametri di reti per la gestione del log, la seconda, fino ai parametri della connessione con il database, la terza.

Classe **Cache**: essenziale alla chiusura e all'avvio, qualora l'utente abbia inserito almeno un dato per una nuova prenotazione, la classe si occupa di salvare su file locale BIN tutti i dati, compresi la data di prenotazione scelta, alla chiusura del Sistema. All'avvio, se qualcosa è stato precedentemente scritto, recupera questi dati, altrimenti li cancella.

Classe **Prenotazione**: è di fatto la classe Bean. Contiene esclusivamente i campi per ogni riga della tabella, che siano stringhe o numeri, e i metodi set e get per impostare e recuperare i valore di ciascun campo.

Classe **TabellaPrenotazioni**: gestisce la creazione della TableView con le celle formattate in modo da essere modificabili. Popola la tabella con i dati aggiornati inizialmente alla data corrente o diversa se salvata in cache dal database; la aggiorna aggiungendo l'ultima riga qualora venisse editata o rimuovendo la riga interessata.

Classe **EditingCell**: contiene il necessario per formattare correttamente le celle facendo sì che alla selezione di queste, si crei una TextField all'interno della cella, il cui testo riporta quello precedente scritto nella cella. Consente la modifica di questa e al termine salva il nuovo testo come stringa nella cella.

Classe **GestoreDB**: contiene interamente le operazioni per la connessione al database e per il recupero delle prenotazioni, inserimento e rimozioni di queste; consente inoltre di recuperare i posti occupati per il giorno selezionato

Classe **GraficoABarre**: contiene i metodi in grado di costruire graficamente il grafico a barre in 2 varianti: il caso in cui voglia vedere le presenze giornaliere di tutti negli ultimi n giorni oppure le singole presenze di un partecipante selezionato, sempre negli ultimi n giorni. Questo n è variabile e gli viene passato.

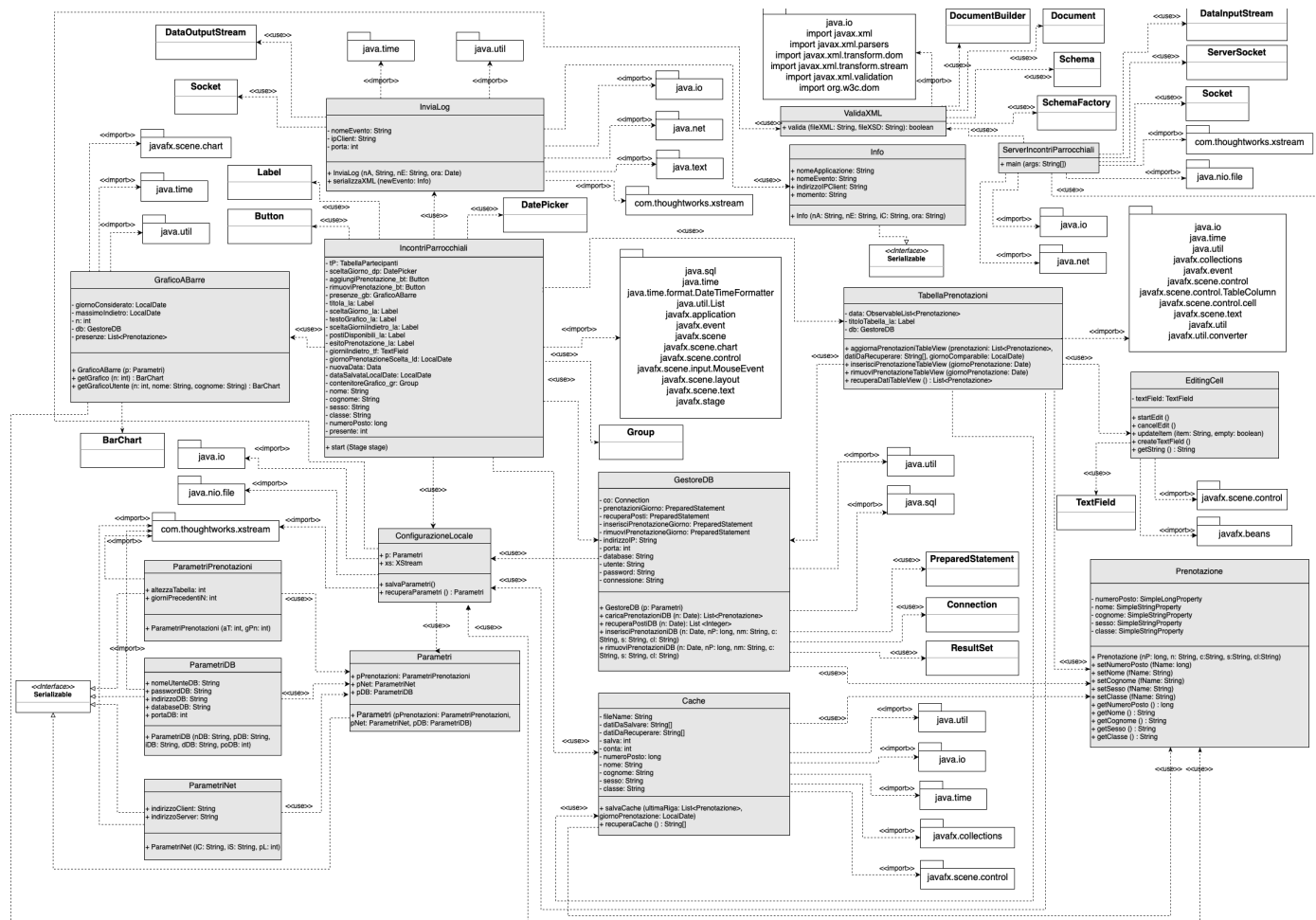
Classe **InviaLog**: la classe più auto-esplicativa di tutte. A seguito di un evento, questa



raccoglie tutte le informazioni riferite all'evento, le codifica in un file XML, sono stringhe, invia il suddetto file alla server, qualora questo fosse disponibile.

Classe **ValidaXML**: prende in ingresso 2 file: uno XML e uno XSD. Chiamata sia da InviaLog che da ConfigurazioneLocale, valida il file XML con il relativo XSD della classe opportuna restituendo come output l'esito della validazione.

Classe **ServerIncontriParrocchiali**: apre la connessione su un server socket su una determinata porta, attende la ricezione degli eventi inviati tramite la classe InviaLog e stampa l'informazione dell'evento. Il tutto avviene all'infinito.



MANUALE UTENTE

Fase preliminare

All'avvio, il Sistema preleva i parametri di configurazione locale, tramite la classe ConfigurazioneLocale dal file cfg.xml, sito nella carte config. Questi sono:

- Parametri sulla gestione delle prenotazioni
- Altezza della tabella: intero.
- Giorni precedenti cui mostrare il numero di presenze giornaliere, separate per classe.

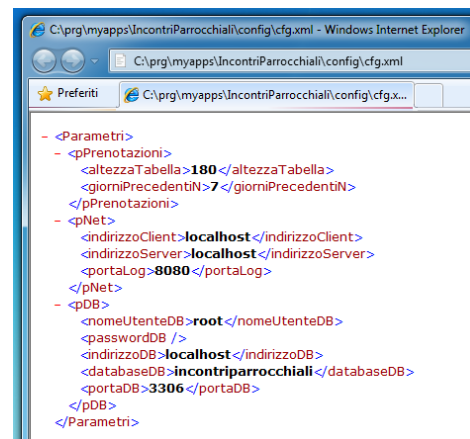


Parametri sulla gestione del log

- Indirizzo IP Client: stringa.
- Indirizzo IP Server: stringa.
- Porta di log: int.

Parametri sulla connessione al database:

- Nome utente: stringa.
- Password: stringa.
- Indirizzo IP: stringa.
- Porta database: stringa.

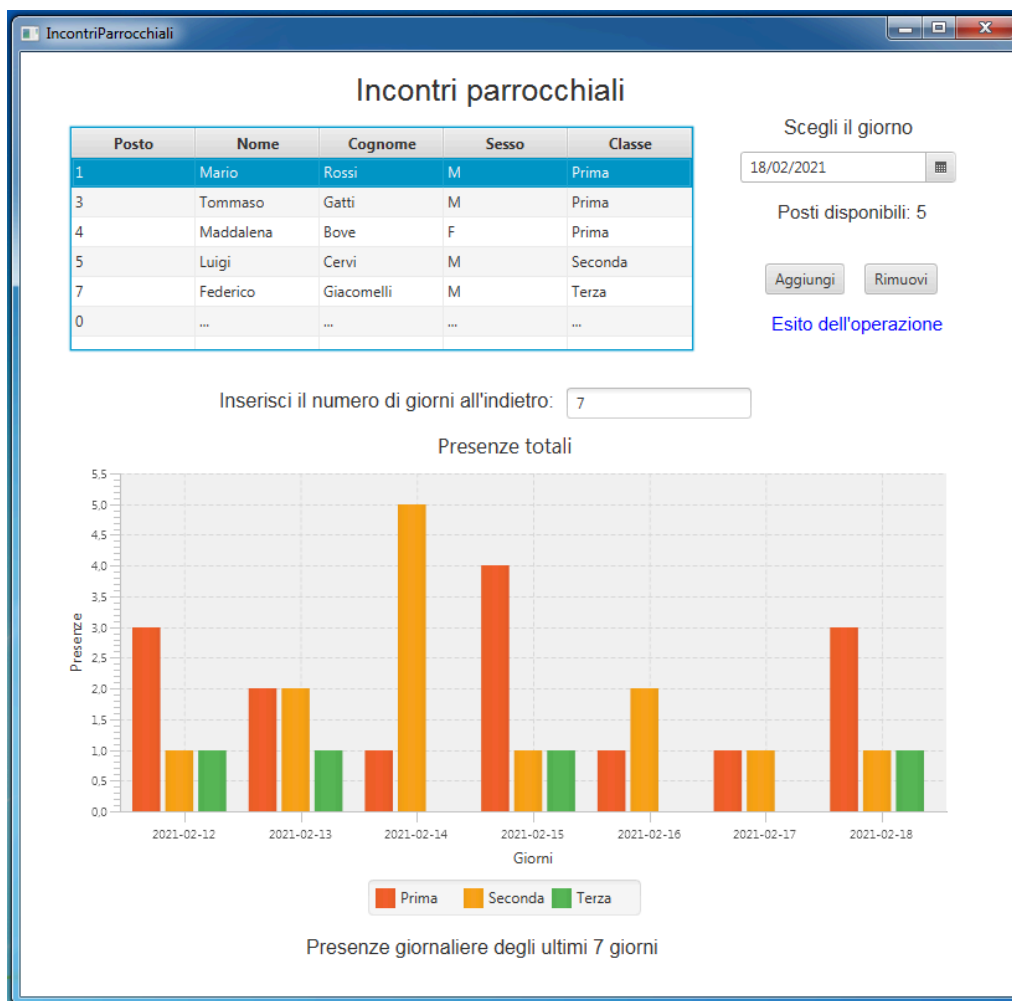


Fase di avvio

All'avvio,

in caso di dati salvati in cache su file binario, questi vengono recuperati e inseriti nell'ultima riga della tabella, dove del resto erano stati scritti prima della chiusura, insieme all'eventuale diversa data di prenotazione.

Recuperati i dati di configurazione locale e di cache, si presenta la seguente interfaccia grafica:



Inserimento di una prenotazione

1. Si sceglie il giorno della prenotazione nel DatePicker in alto a destra.
2. Si seleziona la l'ultima riga della tabella, quella contenente (0,...,...,...).

Posto	Nome	Cognome	Sesso	Classe
1	Mario	Rossi	M	Prima
3	Tommaso	Gatti	M	Prima
4	Maddalena	Bove	F	Prima
5	Luigi	Cervi	M	Seconda
7	Federico	Giacomelli	M	Terza
0

Scegli il giorno

18/02/2021

Posti disponibili: 5

Aggiungi

Rimuovi

Esito dell'operazione

3. A questo punto si scrivono rispettivamente:

- Il numero di posto cui ci si vuole prenotare; è un long e va da 0 a 10.
- Il nome del partecipante; è una stringa.
- Il cognome del partecipante; è una stringa.
- Il sesso del partecipante; è un carattere.
- La classe del partecipante; è una stringa.

4. Il numero del posto non deve corrispondere a uno dei posti già occupati. Qualora anche solo un dato non risulti inserito o il numero del posto è già presente, non verrà inserita la prenotazione.

Posto	Nome	Cognome	Sesso	Classe
1	Mario	Rossi	M	Prima
3	Tommaso	Gatti	M	Prima
4	Maddalena	Bove	F	Prima
5	Luigi	Cervi	M	Seconda
7	Federico	Giacomelli	M	Terza
8	Matteo	Gialli	M	Terza

Scegli il giorno

18/02/2021

Posti disponibili: 5

Aggiungi

Rimuovi

Esito dell'operazione

5. Inserite correttamente tutte e 5 le informazioni, si preme sul pulsante Aggiungi

6. La prenotazione risulterà inserita se verrà inserita una nuova riga vuota nella tabella subito sotto e la scritta **Esito dell'operazione** è ora **Inserimento avvenuto!**

Il numero dei posti disponibili scende.
Il grafico si aggiornerà di conseguenza.

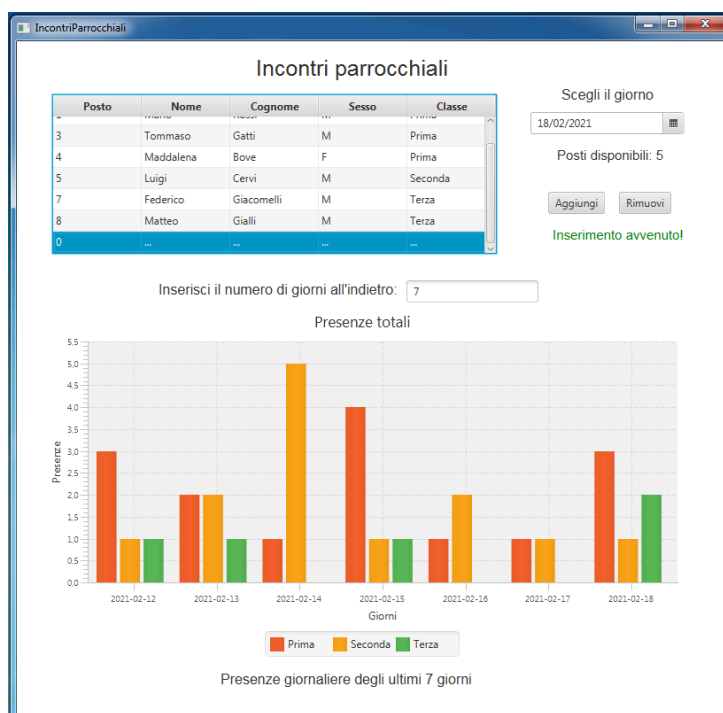
Un istante dopo aver inserito la prenotazione nella tabella, questa viene inserita anche nel database.

Se almeno un dato inserito non è corretto, l'inserimento non va a buon fine mostrando

Aggiungi

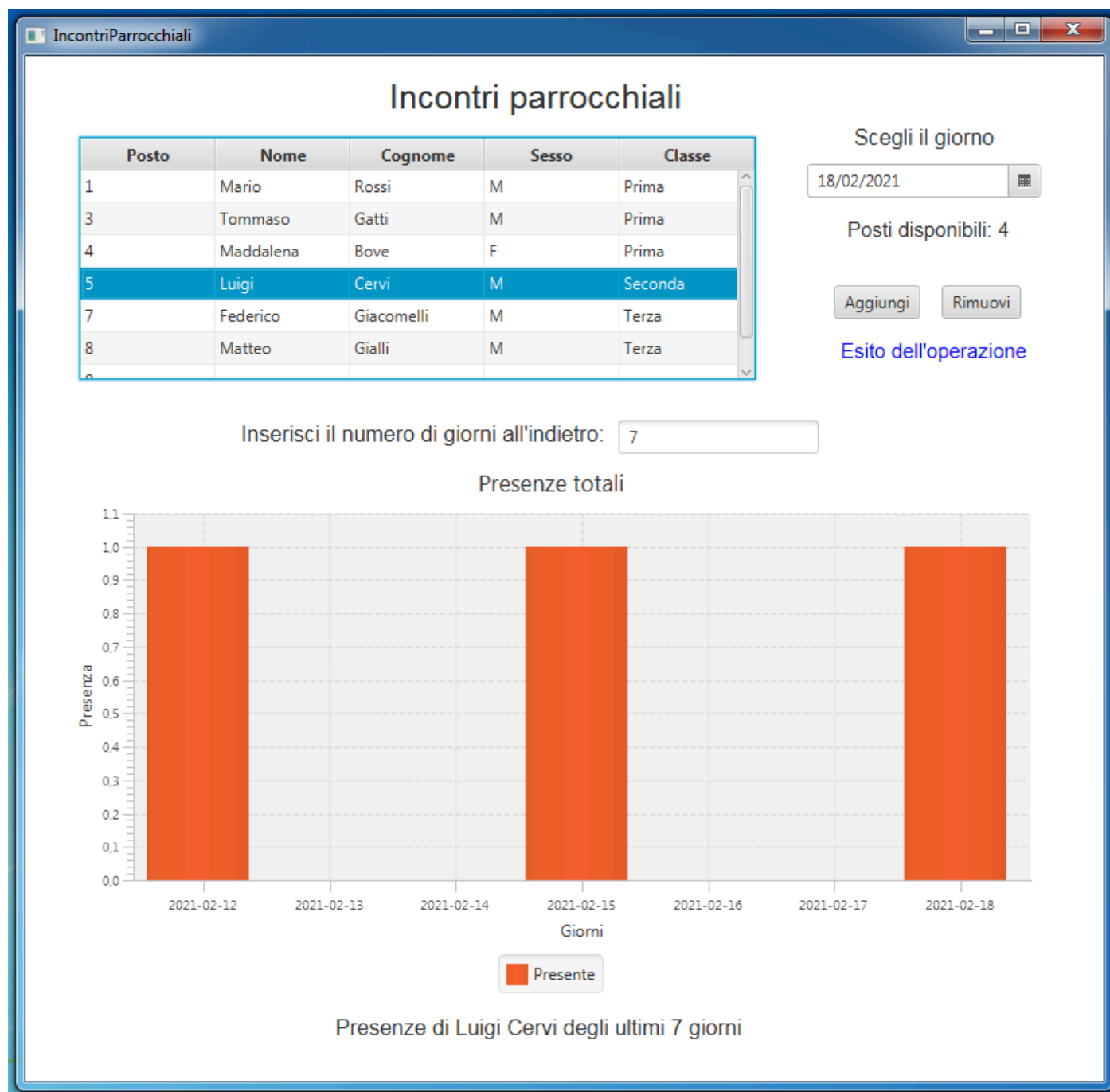
Rimuovi

Inserimento fallito!



Rimozione di una prenotazione

1. Si sceglie la data contenente la prenotazione da rimuovere nel DatePicker in alto a destra.
2. Si sceglie la riga corrispondente alla prenotazione interessata. Il grafico si aggiorna di conseguenza.



3. Se si è selezionata una riga contenente effettivamente una prenotazione, quindi non quella vuota ultima, si preme su **Rimuovi** e la tabella, così come il grafico e il database, si aggiornano di conseguenza. La scritta **Esito dell'operazione** è ora **Rimozione avvenuta!**

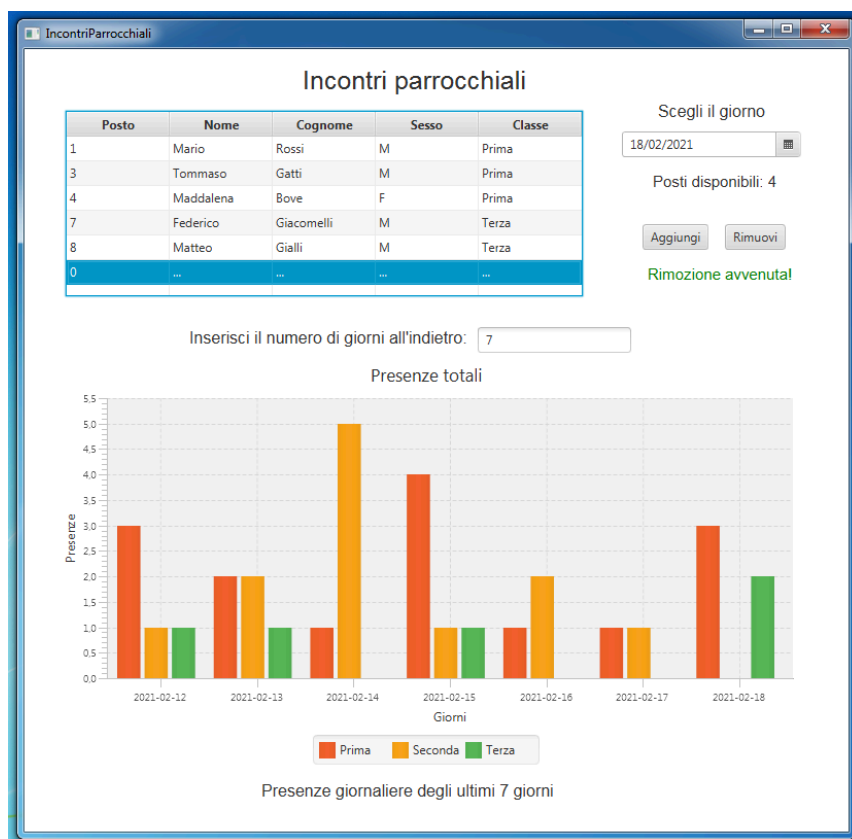
Qualora si sia selezionata una riga vuota e si tenti di rimuoverla, verrà mostrato la seguente scritta. Chiaramente non verrà rimossa, il grafico mostra ancora la vista generale e il database non viene alterato.

Aggiungi

Rimuovi

Rimozione fallita!

Esempio di rimozione di Luigi Cervi.



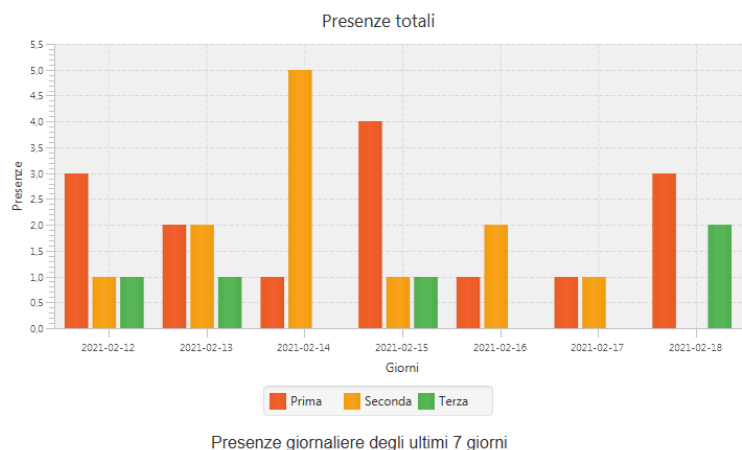
Presenze giornaliere

Viene mostrato, di default, il seguente grafico. Lo si può tornare a vedere, una volta selezionata una prenotazione, selezionando la riga vuota finale.

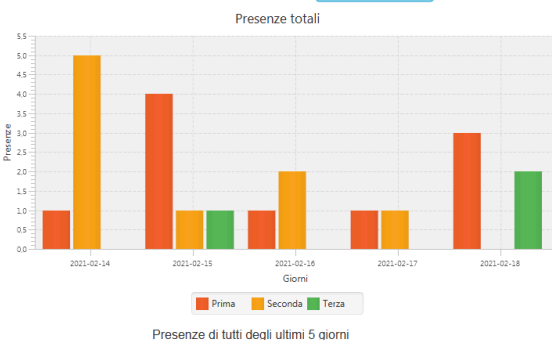
Mostra la **quantità** di partecipanti, suddivisa per classi, negli ultimi 7 giorni precedenti a oggi dove 7 è un parametro di configurazione.

E' possibile cambiare il numero di giorni all'indietro, inserendo un numero nell'apposito TextField e il grafico si aggiorna di conseguenza.

Inserisci il numero di giorni all'indietro: 7

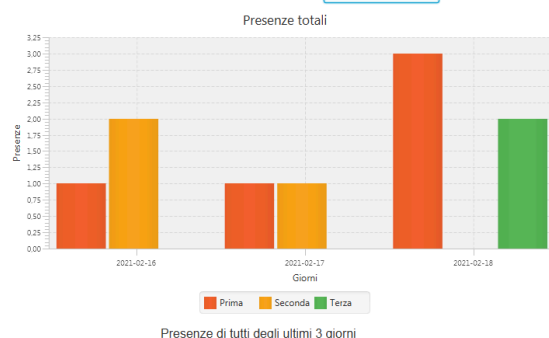


Inserisci il numero di giorni all'indietro: 5



Esempio con n giorni pari a 5.

Inserisci il numero di giorni all'indietro: 3



Esempio con n giorni pari a 3.

Presenze partecipanti

1. Selezionare col mouse la riga contenente il partecipante interessato.
2. Il grafico e la scritta esplicativa si aggiornano di conseguenza.

Posto	Nome	Cognome	Sesso	Classe
1	Mario	Rossi	M	Prima
3	Tommaso	Gatti	M	Prima
4	Maddalena	Bove	F	Prima
7	Federico	Giacomelli	M	Terza
8	Matteo	Gialli	M	Terza
0

Scegli il giorno

18/02/2021

Posti disponibili: 5

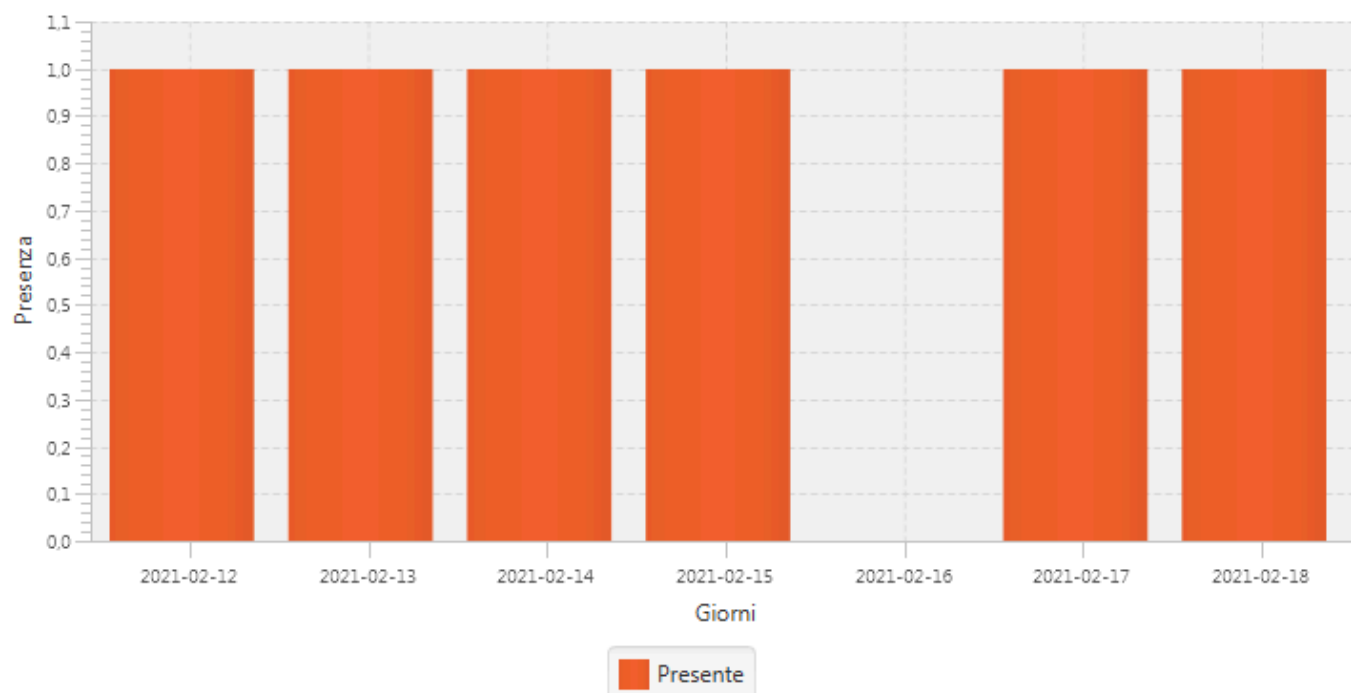
Aggiungi

Rimuovi

Esito dell'operazione

Inserisci il numero di giorni all'indietro: 7

Presenze totali



Presenze di Mario Rossi degli ultimi 7 giorni

Se volessi vedere le presenze di un partecipante negli ultimi n giorni:

1. Seleziono col mouse la riga contenente il partecipante interessato.
2. Modifico n

Inserisci il numero di giorni all'indietro: 7

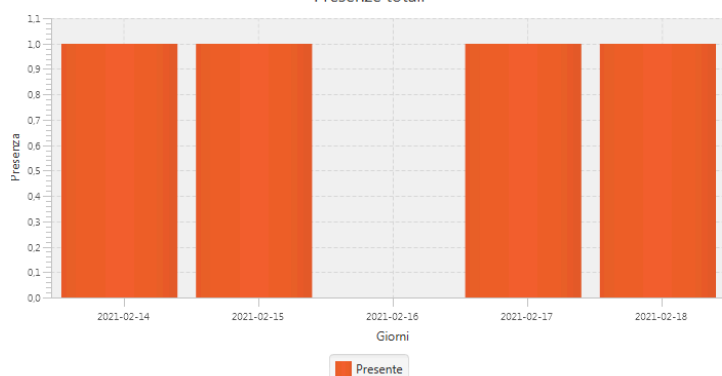
3. Il grafico e la scritta esplicativa si aggiornano di conseguenza

Esempio con n=5.

Pagina 9 di 10

Inserisci il numero di giorni all'indietro: 5

Presenze totali



Presenze di Mario Rossi degli ultimi 5 giorni

GESTIONE LOG

La gestione del log è distribuita sulle seguenti classi:

- **InviaLog**
- **ServerIncontriParrocchiali**

InviaLog

Viene creato un oggetto di tipo **Info**; questo presenta le seguenti variabili:

- *nomeApplicazione*: String il cui testo è “IncontriParrocchiali”
- *nomeEvento*: String il cui testo è dato dal tipo di evento
- *indirizzoIPClient*: String il cui testo è “localhost”, preso dalla configurazione locale.
- *momento*: String che rappresenta il timestamp dell’invio

I tipi di evento possibili sono i seguenti:

- Avvio dell’applicazione: nomeEvento = “AVVIO”
- Aggiunta di una prenotazione: nomeEvento = “Inserita prenotazione”
- Rimozione di una prenotazione: nomeEvento = “Rimozione prenotazione”
- Cambio di n (numero di giorni precedenti a oggi): nomeEvento = “NuovoN”
- Chiusura dell’applicazione: nomeEvento = “CHIUSURA”

Una volta creata la stringa contenente i parametri in XML, viene inviata al server, una volta assicurata l’effettiva connessione.

ServerIncontriParrocchiali

E’ un progetto Java assestante.

Accende il server sulla porta predefinita e riceve i dati in loop. Una volta ricevuti, li stampa esattamente come li riceve, secondo lo schema XML imposto e effettua la validazione del file, mostrando l’esito.

Esempio sull’avvio dell’applicazione

Esempio sul nuovo giorno di prenotazione.

```
ServerIncontriParrocchiali (run) % IncontriParrocchiali (jfxsa-run) %
run:
Attivazione server ...
Server in esecuzione!
<Info>
  <nomeApplicazione>IncontriParrocchiali</nomeApplicazione>
  <nomeEvento>AVVIO</nomeEvento>
  <indirizzoIPClient>localhost</indirizzoIPClient>
  <momento>18-02-2021 10:30:07</momento>
</Info>
Validazione file: myfiles/tmp.xml
Validazione riuscita!
|
```

```
<Info>
  <nomeApplicazione>IncontriParrocchiali</nomeApplicazione>
  <nomeEvento>AggiornoGiornoPrenotazione</nomeEvento>
  <indirizzoIPClient>localhost</indirizzoIPClient>
  <momento>18-02-2021 10:30:35</momento>
</Info>
Validazione file: myfiles/tmp.xml
Validazione riuscita!
|
```

La validazione si attua tramite due file XSD:

- ParametriConfigurazioneLocale.xsd posto in “myfiles” di IncontriParrocchiali
- ParametriInfoLog.xsd posto in “myfiles” di ServerIncontriParrocchiali

Ed è gestita dalla classe ValidaXML.

