

# Programmazione avanzata

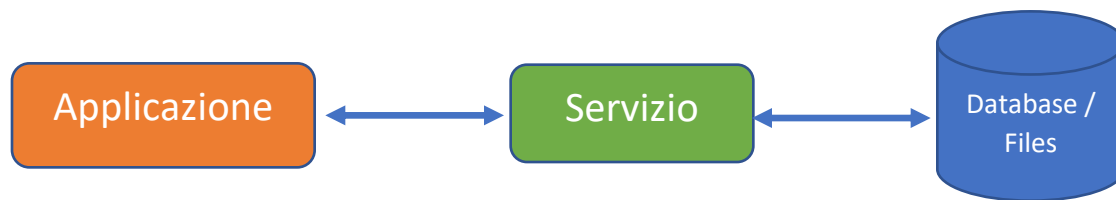
## Lezione 10

Specifiche progetto per esame finale, anno accademico 2023/2024.

### Obiettivo

Realizzare un'applicazione Java distribuita che sia composta da un'interfaccia grafica costruita con JavaFX e da un servizio attraverso il quale vengono gestiti i dati dell'applicazione. I dati dell'applicazione potranno essere memorizzati in un database o in un insieme di files locali.

### Architettura



I dati dell'applicazione dovranno essere memorizzati all'interno di un database MySQL che verrà interrogato solamente dal servizio oppure all'interno di un insieme di file locali al servizio. La scelta del tipo di tecnologia per la memorizzazione dei dati (DB o files) è libera.

Nel caso dell'utilizzo di files, i dati dovranno essere letti e caricati in strutture apposite durante l'inizializzazione del servizio. Durante il ciclo di vita del servizio, tali strutture dati potranno essere utilizzate per rispondere alle richieste dell'applicazione. All'occorrenza quando richiesto dall'utente attraverso l'invocazione di un apposito metodo (e.g., "/salva"), le strutture dati dovranno essere salvate nuovamente su file, sovrascrivendo i vecchi.

L'applicazione, che espone un'interfaccia utente, dovrà interagire solamente con il servizio e non potrà accedere al database oppure leggere i files locali del servizio.

### Caso d'uso

Il caso d'uso può essere sviluppato e scelto dallo studente. Per ispirazione sui dati che l'applicazione può usare si possono cercare dei servizi web esistenti che forniscono dei dati dalla seguente pagina:

<https://free-apis.github.io/>

In alternativa possono essere utilizzati dei dati fasulli generati attraverso il seguente servizio (solo nel caso con memorizzazione nel DB):

<https://generatedata.com/generator>

## Scambio di dati

Lo scambio dei dati tra il servizio e l'applicazione dovrà avvenire attraverso richieste http con le quali vengono scambiati dei dati codificati. Il tipo di azione richiesta come visto potrà essere codificata attraverso l'URL esposto dal servizio e il metodo usato per la richiesta.

I dati scambiati dovranno in ogni modo essere codificati in formato JSON o XML.

## Requisiti

L'applicazione e il servizio dovranno rispettare rigorosamente<sup>1</sup> i seguenti requisiti progettuali:

1. Nel caso di uso di database, il database e le tabelle dovranno essere creati dal servizio su un'istanza di MySQL locale (parametri di connessione – IP: 127.0.0.1, Username: root, Password: root). Il nome del database creato dovrà essere il numero di matricola dello studente. I dati iniziali dovranno essere caricati dal servizio durante la fase di inizializzazione, partendo da un file locale (file con dati random fasulli) oppure a partire dai dati scaricati da un servizio web pubblico. Questa fase di inizializzazione dovrà essere eseguita una sola volta automaticamente all'avvio del servizio, ad esempio andando a verificare l'esistenza del database. Nel caso di avvii successivi, la fase di inizializzazione non dovrà essere eseguita.
2. Nel caso di utilizzo di files, questi dovranno essere creati dal servizio durante la fase di inizializzazione obbligatoriamente a partire da dati scaricati da un servizio web pubblico. La fase di inizializzazione del sistema dovrà essere eseguita automaticamente solo al primo avvio, ad esempio andando a verificare che nessuno dei file locali di memorizzazione sia presente. Durante le esecuzioni successive, il servizio dovrà limitarsi a caricare nelle strutture dati le informazioni dai files. Al fine di salvare eventuali cambiamenti ai dati, il servizio dovrà implementare un metodo apposito per salvare nei file locali il contenuto corrente delle strutture dati, sovrascrivendo i file precedenti. Al fine di permettere all'utente di invocare tale metodo, l'applicazione dovrà contenere un pulsante "Salva" nella schermata principale.
3. L'applicazione dovrà contenere almeno una schermata che mostra una lista di elementi e un menù a scomparsa per la cancellazione degli elementi.
4. Il sistema deve almeno un Unit test.

## Consegna

La consegna dovrà essere fatta creando due repository su gitlab, uno per l'applicazione e uno per il servizio. I repository dovranno essere privati e condivisi solo con il docente (nome utente 'carlo.vallati'). Per la procedura dettagliata vedere la fine del documento.

La creazione dei repository e il caricamento del codice dovrà essere accompagnata da una mail inviata al docente con oggetto 'Consegna progetto programmazione avanzata appello XX/YY/YYYY' con il link al repository gitlab nel corpo della mail.

Per la data di consegna fa fede l'invio della mail e non la condivisione del repository gitlab, eventuali problemi di condivisione potranno essere risolti anche dopo la consegna (no panic!).

---

<sup>1</sup> Il non rispetto di questi requisiti minimi, rendono la consegna insufficiente

La consegna dovrà essere accompagnata da un documento o una presentazione che presenta l'applicazione, la struttura del software realizzato e le scelte progettuali fatte. Il documento verrà poi utilizzato in sede d'esame per presentare l'applicazione nella prima parte della discussione. Il file del documento dovrà essere caricato sul repository gitlab all'interno di una cartella 'doc' del codice del server.

Una consegna che non rispetterà le linee guida sopra non verrà considerata per la correzione.

La consegna del progetto dovrà avvenire almeno 5 giorni prima dell'appello di esame, esempio se l'appello d'esame è fissato per il 10/01/2024 l'ultimo giorno utile per la consegna è il 04/01/2024 (incluso). Consegne arrivate oltre la data ultima verranno considerate per l'appello successivo.

## Discussione orale

La discussione del progetto viene svolta il giorno dell'esame orale. La discussione del progetto e lo svolgimento della prova orale con il Prof. Vecchio si dovranno svolgere obbligatoriamente allo stesso appello, fatta eccezione nel caso in cui la prova orale venga sostituita con i compitini.

La discussione si svolgerà con le seguenti modalità:

- Presentazione dell'applicazione con introduzione al caso d'uso e alle scelte progettuali principali [5 minuti massimo]
- Breve dimostrazione del funzionamento dell'applicazione
- Domande sul codice e discussione delle scelte implementative

La presentazione dell'applicazione con introduzione al caso d'uso dovrà essere effettuata rigorosamente nei 5 minuti a disposizione, si consiglia quindi di effettuare una presentazione generale senza troppi dettagli, i quali, se necessario, verranno chiesti dal docente durante la fase di discussione.

## Raccomandazioni

L'implementazione del progetto è individuale, non sono ammessi progetti in gruppo. Nella parte orale lo studente dovrà dimostrare completa padronanza del codice e delle scelte implementative effettuate. Mancanza di **completa padronanza** del codice consegnato risulterà in un rifiuto della consegna e richiesta di nuova consegna con diverso caso d'uso e relativa implementazione. Al fine di evitare situazioni in cui "non mi ricordo cosa fa questo pezzo di codice perché l'ho scritto diversi mesi fa" si consiglia di consegnare e discutere il progetto all'appello successivo al completamento della sua implementazione.

Il codice consegnato sarà analizzato con un sistema per la verifica delle similarità del software, per verificare la similarità con altre consegne o codici open-source liberamente disponibili. Nel caso in cui la similarità con altri codici sia troppo alta la consegna verrà rifiutata e si dovrà procedere con un'altra implementazione considerando un caso d'uso diverso. È sconsigliato svolgere il progetto in coppia o di lavorare sullo stesso caso d'uso, per evitare situazioni in cui "si questo pezzo di codice è uguale perché l'abbiamo fatto insieme".

## Valutazione

La valutazione del progetto avviene con un punteggio da 0 a 33 che va a comporre la valutazione finale dell'esame:

La valutazione del progettino avverrà secondo le seguenti linee guida:

- 11 punti, presentazione progetto e discussione orale, di cui:
  - 3 punti, sulla qualità e chiarezza della presentazione
  - 8 punti, discussione sul codice e scelte progettuali effettuate
- 12 punti, architettura e struttura del codice
- 10 punti, numero delle funzionalità implementate e loro correttezza
  - 7 punti, implementazione corretta del numero di funzionalità di base richieste dalle specifiche
  - 3 punti, implementazione di funzionalità aggiuntive con costrutti non visti a lezione

La sostituzione di funzionalità richieste da specifiche con altre deve essere concordata con il docente.

### Procedura di condivisione su Gitlab

Per ogni progetto da condividere (uno per il servizio e uno per l'applicazione) si deve creare un repository privato su gitlab. La procedura per condividere il progetto è la seguente:

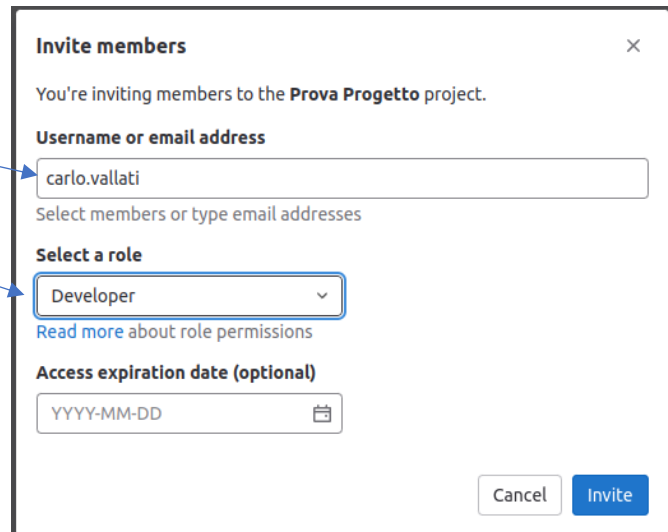
Dal menu a sinistra selezionare "Project Information" -> "Members"

Selezionare poi "Invite members"

The screenshot displays the GitLab web interface. On the right, a sidebar menu lists various project management tools. The 'Members' option is selected and highlighted. The main area shows the 'Project members' page for the 'Prova Progetto' repository. It includes a search bar for filtering members and a table listing the current member, Carlo Vallati, with his role as 'Owner' and the date of his last activity.

Account	Source	Access granted	Max role	Expiration	Created on	Last activity
Carlo Vallati @warner831	Direct member	1 minute ago by Carlo Vallati	Owner	Expiration date	21 Dec, 2022	21 Dec, 2022

Nella nuova finestra selezionare l'account "carlo.vallati" (attenzione ad eventuali altri account) e selezionare come ruolo "Developer"



The image shows a 'Invite members' dialog box with two blue arrows pointing from the text on the left to specific fields. The first arrow points to the 'Username or email address' input field, which contains the text 'carlo.vallati'. The second arrow points to the 'Select a role' dropdown menu, which is currently set to 'Developer'. The dialog box also includes a 'Read more' link, an 'Access expiration date (optional)' field with a calendar icon, and 'Cancel' and 'Invite' buttons at the bottom right.

**Invite members** ×

You're inviting members to the **Prova Progetto** project.

**Username or email address**

carlo.vallati


Select members or type email addresses

**Select a role**

Developer ▾

[Read more](#) about role permissions

**Access expiration date (optional)**

YYYY-MM-DD 

Cancel Invite