

# Ingegneria del Software

# **Gestione di tassonomie gerarchiche**

Daniele Costanzo Pellati

A.A. 2024/2025

# Obiettivo

Sviluppare un **framework** per la gestione di **tassonomie gerarchiche**

# Funzionalità richieste

- Parte **gestionale** per la definizione della tassonomia
- Ogni elemento può contenere un numero **indeterminato di attributi**:
  - Stringhe
  - Numeri
  - Valori categoriali
- È possibile definire **elementi sinonimi**
- La tassonomia deve essere:
  - **Navigabile lato client**
  - **Scaricabile in formato JSON**

# Versionamento

- Quando un elemento diventa **obsoleto** o viene **sostituito**, si crea una **nuova versione** della tassonomia
- Deve essere mantenuta la **lista degli elementi precedenti sostituiti**

## Output atteso

- Interfaccia web per la gestione e visualizzazione delle tassonomie
- Possibilità di esportare la tassonomia in formato **JSON completo**
- Navigazione della struttura lato client

# Considerazioni iniziali

- Ogni elemento della tassonomia può essere:
  - **Visualizzato** con i suoi dettagli
  - **Sostituito** da un nuovo elemento (con tracciamento dei predecessori)
  - **Rimosso** (con inserimento nella lista generale degli elementi obsoleti)
  - **Esteso** aggiungendo un **figlio**
- È presente una distinzione tra:
  - **Gestione** degli elementi (modifiche, creazione, eliminazione)
  - **Visualizzazione** della tassonomia (navigazione gerarchica lato client)
- Il sistema supporta:
  - **Attributi tipizzati** per ogni elemento (stringhe, numeri, valori categoriali)

## Funzione: Visualizza dettagli elemento

- L'utente clicca sul pulsante "**Visualizza**" di un elemento
- L'applicazione effettua una richiesta al database per recuperare:
  - I **valori attuali** dell'elemento, con tipo di dato per ciascun attributo
  - I **sinonimi** associati all'elemento
  - La lista dei **valori precedenti**
- I dati vengono mostrati in tabelle separate:
  - Tabella dei valori e tipi
  - Tabella dei sinonimi
  - Tabella dei valori storici precedenti
- Questo permette di avere una panoramica completa dello stato e della storia dell'elemento



## Funzione: Sostituisci elemento

- Quando si sceglie di sostituire un elemento:
  - L'utente viene indirizzato a un **form di creazione** per il nuovo elemento
- La sostituzione avviene solo dopo che:
  - Il **nome** e la **data di modifica** del vecchio elemento vengono salvati tra i **nomi precedenti**
- Gli attributi e i sinonimi **del vecchio elemento vengono rimossi**
- Il nuovo elemento assume il ruolo del precedente nella tassonomia, mantenendo la storia delle modifiche

## Funzione: Aggiungi figlio

- Quando si clicca su "**Aggiungi figlio**" si apre una pagina con un **form di inserimento**
- L'utente deve inserire:
  - Il **nome** (campo obbligatorio, genera errore se vuoto)
  - Gli **attributi** (stringhe, numeri, valori categoriali)
  - I **sinonimi**
- Al salvataggio, il nuovo elemento viene inserito nel database con:
  - L'**ID padre** corrispondente all'elemento da cui si è partiti
- Questo consente di estendere la tassonomia in modo gerarchico, creando rami figli

## Funzione: Rimuovi elemento

- Quando si clicca su "**Rimuovi**" l'applicazione procede con una rimozione controllata:
  - i. Elimina gli **attributi**, i **sinonimi** e i **nomi precedenti** associati all'elemento
  - ii. Copia il **nome** dell'elemento nella tabella dei nomi precedenti
    - In questa copia, l'elemento attuale viene impostato a **NULL**
    - Questo consente di gestirlo come **elemento rimosso**, mantenendone traccia
  - iii. Solo se tutti i passaggi precedenti hanno avuto successo, l'elemento viene **effettivamente rimosso** dal database
- La procedura garantisce consistenza e tracciabilità degli elementi eliminati
- La rimozione rimuove eventuali figli, e per gli eventuali figli viene effettuato lo stesso processo.

# Interfaccia principale: gestione e navigazione

Nella pagina iniziale dell'applicazione sono presenti tre pulsanti principali:

- **Visualizza categorie**

Mostra le **categorie disponibili** per gli attributi di tipo *categoriale*

- **Inserisci categoria**

Permette di **aggiungere una nuova categoria**, con i relativi valori consentiti

- **Nomi precedenti**

Elenca gli elementi della tassonomia che sono stati **rimossi o sostituiti**, per garantire tracciabilità.

# Funzioni principali dell'Index

L'index offre le seguenti operazioni:

- **Aggiungi tassonomia**

Permette di creare una nuova tassonomia vuota (dettagli nella prossima slide)

- **Visualizza tassonomia**

Mostra l'albero gerarchico completo che rappresenta la tassonomia corrente, consentendo la navigazione degli elementi.

- **Modifica informazioni tassonomia**

Permette di modificare il **nome** e la **descrizione** generale della tassonomia.

- **Rimuovi tassonomia**

Funzione complessa, che prevede una gestione particolare (vedi slide successive).

- **Scarica in JSON**

Ricostruisce la tassonomia in una stringa Json che verrà poi scaricata.

## Aggiungi nuova tassonomia

- L'utente accede a un form per creare una nuova tassonomia
- Il form richiede l'inserimento di:
  - **Nome** della tassonomia (campo obbligatorio)
  - **Descrizione** della tassonomia (campo facoltativo)
- Dopo l'invio, viene creata una tassonomia vuota nel database con i dati inseriti
- La nuova tassonomia è pronta per essere popolata con elementi e attributi

# Rimuovi tassonomia

- La rimozione parte dalla **radice** della tassonomia
- Poiché rimuovere un elemento comporta anche la rimozione di tutti i suoi figli, rimuovere la radice equivale a **svuotare l'intera tassonomia**
- Dopo aver svuotato la tassonomia, si eliminano tutti gli **elementi accessori**:
  - Categorie
  - Nomi precedenti
- Solo se tutte le operazioni di rimozione vanno a buon fine, si procede con la cancellazione della tassonomia stessa dal database
- Questo approccio garantisce coerenza e integrità dei dati residui

# Ringraziamenti

Grazie per l'attenzione!

## Contatti

Daniele Costanzo Pellati

Email: [danielecostanzo.pellati@studenti.unipr.it](mailto:danielecostanzo.pellati@studenti.unipr.it)

GitHub: [github.com/danielepellati](https://github.com/danielepellati)