

# Laboratorio di informatica A

## Caso di studio

### Specifiche v. 1.0

Questo documento descrive le specifiche di realizzazione del progetto "Gestore Archivio Musicale: TTICSS", da realizzare nell'ambito del corso di studi di Laboratorio di informatica, a.a. 2019/20. Obiettivo del progetto è la gestione di una base dati musicale, al fine di fornire agli appassionati uno strumento per la ricerca di informazioni di qualunque genere riguardanti i brani e gli artisti di loro gradimento.

### Funzionalità del sistema

Le operazioni che il sistema fornirà agli utenti si suddividono in 4 classi:

- 1 Operazioni essenziali per la gestione dei dati
- 2 Operazioni essenziali per la ricerca di informazioni
- 3 Operazioni di registrazione
- 4 Operazioni di interfaccia

### Dati del sistema

I dati che saranno gestiti, per ogni brano musicale saranno:

#### Brano

Nome	Descrizione	Tipo	Vincoli
<b>idBrano</b>	Identificativo univoco del brano	Intero	>0
<b>titolo</b>	Titolo del brano	Stringa	Minimo 1 carattere Massimo 30 caratteri
<b>durata</b>	Durata del brano in secondi	Intero	>0
<b>anno</b>	Anno di uscita del brano	Intero	>= 1000
<b>ascolti</b>	Numero di ascolti	Intero	>=0

#### Artista

Nome	Descrizione	Tipo	Vincoli
<b>idArtista</b>	Identificativo artista	Intero	>0
<b>nome</b>	Nome dell'artista	Stringa	Minimo 1 carattere Massimo 30 caratteri
<b>cognome</b>	Cognome dell'artista	Stringa	Minimo 1 carattere Massimo 30 caratteri
<b>nomeArte</b>	Nome d'arte dell'artista	Stringa	Minimo 0 caratteri Massimo 30 caratteri
<b>linkBio</b>	Url di un sito con informazioni biografiche ed anagrafiche	Stringa	Maggiore di 8 caratteri Può essere anche un valore che indica che l'url non è stato inserito

#### BranoArtista (identifica da chi è cantato questo brano)

Nome	Descrizione	Tipo	Vincoli
<b>idBrano</b>	Identificativo del brano	Intero	>0
<b>idArtista</b>	Identificativo dell'artista	Intero	>0

## Album

Nome	Descrizione	Tipo	Vincoli
<b>idAlbum</b>	Identificativo dell'album	Intero	>0
<b>titolo</b>	titolo dell'album	Stringa	Minimo 1 carattere Massimo 30 caratteri
<b>anno</b>	Anno di uscita dell'album	Intero	>0

## BranoAlbum (da che brani è formato l'album)

Nome	Descrizione	Tipo	Vincoli
<b>idAlbum</b>	Identificativo dell'album	Intero	>0
<b>idBrano</b>	Identificativo del brano	Intero	>0

## Genere (genere della canzone: pop, rock, ...)

Nome	Descrizione	Tipo	Vincoli
<b>idGenere</b>	Identificativo del genere	Intero	>0
<b>nome</b>	Nome del genere musicale	Stringa	Minimo 1 carattere Massimo 30 caratteri

## BranoGenere (brano che appartiene al genere considerato)

Nome	Descrizione	Tipo	Vincoli
<b>idBrano</b>	Identificativo del brano	Intero	>0
<b>idGenere</b>	Identificativo del genere	Intero	>0

## Playlist

Nome	Descrizione	Tipo	Vincoli
<b>idPlaylist</b>	Identificativo della playlist	Intero	>0
<b>idUtente</b>	Identificativo dell'utente	Intero	>0
<b>nome</b>	Nome della playlist	Stringa	Minimo 1 carattere Massimo 30 caratteri
<b>descrizione</b>	Descrizione playlist	Stringa	Minimo 1 carattere Massimo 250 caratteri
<b>pubblica</b>	Tipologia playlist	Booleano	/

## PlaylistBrano (brani da cui è composta una playlist)

Nome	Descrizione	Tipo	Vincoli
<b>idPlaylist</b>	Identificativo della playlist	Intero	>0
<b>idBrano</b>	Identificativo del brano	Intero	>0

## Utente

Nome	Descrizione	Tipo	Vincoli
<b>idUtente</b>	Identificativo dell'utente	Intero	>0
<b>nomeUtente</b>	Nominativo dell'utente	Stringa	Minimo 6 caratteri Massimo 60 caratteri
<b>password</b>	Password dell'utente	Stringa	Minimo 6 caratteri Massimo 60 caratteri
<b>admin</b>	Controllo ruolo utente	Booleano	/

**Database** (Descrizione della sessione attuale. Record che verrà caricato di tutti i dati contenuti nei file così avere un accesso più agile ai dati, contiene anche l'utente che sta utilizzando in questo momento il programma)

Nome	Descrizione	Tipo
<b>utenteCorrente</b>	Identificativo dell'utente corrente	Intero
<b>brano</b>	Tabella brani	Vettore di Brano
<b>artista</b>	Tabella artista	Vettore di Artista
<b>branoArtista</b>	Tabella che indica la relazione tra Brano e Artista	Vettore BranoArtista
<b>album</b>	Tabella Album	Vettore di Album
<b>branoAlbum</b>	Tabella che indica la relazione tra Brano e Album	Vettore BranoAlbum
<b>genere</b>	Tabella Genere	Vettore Genere
<b>branoGenere</b>	Tabella che indica la relazione tra Brano e Genere	Vettore BranoGenere
<b>playlist</b>	Tabella Playlist	Vettore di Playlist
<b>playlistBrano</b>	Tabella che indica la relazione tra Playlist e Brano	Vettore PlaylistBrano
<b>utente</b>	Tabella Utente	Vettore Utente
<b>ultimoEsito</b>	Intero $\geq 0$ , se esito = 0 vuol dire che l'operazione è andata a buon fine; altrimenti esito rappresenta il codice dell'errore che si è verificato	Intero

Inserimento dei dati: Inserimento nella base dati di tutte le informazioni previste per un brano musicale.

L'inserimento di dati che fanno parte di altre sezioni del database faranno parte delle differenti implementazioni dei gruppi (per esempio cercare assegnare un artista ad un brano senza che questo artista sia stato precedentemente inserito nel database).

**InserireBrano** (scrive tutti i campi del nuovo brano nel file dei brani)

**INPUT**

- BaseDati (tipo: Database; il database nel quale verrà inserito il nuovo brano)
- NuovoBrano (tipo: Brano; è il dato strutturato di tipo brano di cui scrivere tutte le informazioni nella tabella dei brani. Inizialmente l'id del brano sarà 0. Nel momento in cui viene inserito, al nuovo brano sarà associato un identificativo univoco >0.)

**OUTPUT**

- BaseDati (tipo: Database; base di dati modificata)

**InserireArtista** (scrive tutti i campi del nuovo Artista nel file degli Artisti)

**INPUT**

- BaseDati (tipo: Database; il database nel quale verrà inserito il nuovo Artista)
- NuovoArtista (tipo: Artista; è il dato strutturato di tipo Artista di cui scrivere tutte le informazioni nella tabella degli Artisti. Inizialmente l'id dell'Artista sarà 0. Nel momento in cui viene inserito, al nuovo brano sarà associato un identificativo univoco >0.)

**OUTPUT**

- BaseDati (tipo: Database; base di dati modificata)

**InserireAlbum** (scrive tutti i campi del nuovo album nel file degli album)

**INPUT**

- BaseDati (tipo: Database; il database nel quale verrà inserito il nuovo album)
- NuovoAlbum (tipo: Album; è il dato strutturato di tipo album di cui scrivere tutte le informazioni nella tabella degli album. Inizialmente l'id dell'album sarà 0. Nel momento in cui viene inserito, al nuovo album sarà associato un identificativo univoco >0.)

**OUTPUT**

- BaseDati (tipo: Database; base di dati modificata)

**InserirePlaylist** (scrive tutti i campi della nuova playlist nella tabella delle playlist)

**INPUT**

- BaseDati (tipo: Database; il database nel quale verrà inserito la nuova playlist)
- NuovaPlaylist (tipo: Playlist; è il dato strutturato di tipo album di cui scrivere tutte le informazioni nella tabella degli album. Inizialmente l'id dell'album sarà 0. Nel momento in cui viene inserito, al nuovo album sarà associato un identificativo univoco >0.)

**OUTPUT**

- BaseDati (tipo: Database; base di dati modificata)

**InserireGenere** (scrive tutti i campi del nuovo genere musicale nella tabella dei generi musicali)

**INPUT**

- BaseDati (tipo: Database; il database nel quale verrà inserito il nuovo genere)
- NuovoGenere (tipo: GenereMusicale; è il dato strutturato di tipo GenereMusicale di cui scrivere tutte le informazioni nella tabella dei generi musicali. Inizialmente l'id del genere musicale sarà 0. Nel momento in cui viene inserito, al nuovo genere musicale sarà associato un identificativo univoco >0.)

**OUTPUT**

- BaseDati (tipo: Database; base di dati modificata)

**InserireUtente** (scrive tutti i campi del nuovo utente nella tabella degli utenti)

**INPUT**

- BaseDati (tipo: Database; il database nel quale verrà inserito il nuovo utente)
- NuovoUtente (tipo: Utente; è il dato strutturato di tipo utente di cui scrivere tutte le informazioni nella tabella degli utenti. Inizialmente l'id dell'utente sarà 0. Nel momento in cui viene inserito, al nuovo utente sarà associato un identificativo univoco >0.)

**OUTPUT**

- BaseDati (tipo: Database; base di dati modificata)

Modifica dei dati: modifica di una o più informazioni di un determinato brano musicale, i nuovi dati verranno letti in input da tastiera in base alle decisioni dell'utente

**ModificareBrano** (modifica le informazioni relative ai brani del file)

**INPUT**

- idBrano (tipo: Intero, identificatore del brano da modificare)
- Campo (tipo: Intero, campo della struttura dati da modificare)
- BaseDati (tipo: Database; il database nel quale verrà modificato il brano)
- BranoModificato (tipo: Brano; il brano modificato dall'utente)

**OUTPUT**

- BaseDati (tipo: Database; base di dati modificata)

**ModificareArtista** (modifica le informazioni relative agli Artisti)

**INPUT**

- BaseDati (tipo: Database; il database nel quale verrà modificato l'Artista)
- idArtista (tipo: Intero, identificatore dell'artista da modificare)
- Campo (tipo: Intero, campo della struttura dati da modificare)
- ArtistaModificato (tipo: Artista; l'artista modificata dall'utente)

**OUTPUT**

- BaseDati (tipo: Database; base di dati modificata)

**ModificareAlbum** (modifica le informazioni relative agli album)

**INPUT**

- BaseDati (tipo: Database; il database nel quale verrà modificato l'album)
- idAlbum (tipo: Intero, identificatore dell'album da modificare)
- Campo (tipo: Intero, campo della struttura dati da modificare)
- AlbumModificato (tipo: Album, l'album modificato dall'utente)

**OUTPUT**

- BaseDati (tipo: Database; base di dati modificata)

**ModificarePlaylist** (modifica le informazioni relative alle playlist)

**INPUT**

- BaseDati (tipo: Database; il database nel quale verrà modificato la playlist)
- idPlaylist (tipo: Intero, identificatore della playlist da modificare)
- Campo (tipo: Intero, campo della struttura dati da modificare)
- ?PlaylistModificata (tipo: Playlist; la Playlist modificata dall'utente)

**OUTPUT**

- BaseDati (tipo: Database; base di dati modificata)

**ModificareUtente** (modifica le informazioni relative agli utenti)

**INPUT**

- BaseDati (tipo: Database; il database nel quale verrà modificato l'utente)
- idUtente (tipo: Intero, identificatore dell'utente da modificare)
- Campo (tipo: Intero, campo della struttura dati da modificare)
- UtenteModificato (tipo: Utente, l'utente modificato dall'utente)

**OUTPUT**

- BaseDati (tipo: Database; base di dati modificata)

Cancellazione dei dati: eliminazione di un brano musicale dalla base dati (tutte le informazioni)

**CancellareBrano** (cancella tutti i dati relativi a ciascun brano musicale)

**INPUT**

- BaseDati (tipo: Database; il database nel quale verrà cancellato il brano)
- IdBrano (campo contenente il valore univoco che identifica ciascun brano, Intero >0.)

**OUTPUT**

- BaseDati (tipo: Database; base di dati modificata)

**CancellareArtista** (cancella tutti i dati relativi a ciascun Artista)

**INPUT**

- BaseDati (tipo: Database; il database nel quale verrà cancellato l'Artista)
- IdArtista (campo contenente il valore univoco che identifica ciascun Artista, Intero >0)

**OUTPUT**

- BaseDati (tipo: Database; base di dati modificata)

**CancellareAlbum** (cancella tutti i dati relativi a ciascun album)

**INPUT**

- BaseDati (tipo: Database; il database nel quale verrà cancellato l'album)
- IdAlbum (campo contenente il valore univoco che identifica ciascun album, Intero >0)

**OUTPUT**

- BaseDati (tipo: Database; base di dati modificata)

**CancellarePlaylist** (cancella tutti i dati relativi a ciascuna playlist)

**INPUT**

- BaseDati (tipo: Database; il database nel quale verrà cancellata la playlist)
- IdPlaylist (campo contenente il valore univoco che identifica ciascuna playlist, Intero >0)

**OUTPUT**

- BaseDati (tipo: Database; base di dati modificata)

**CancellareGenere** (cancella tutti i dati relativi a ciascun genere)

**INPUT**

- BaseDati (tipo: Database; il database nel quale verrà cancellato il genere)
- IdGenere (campo contenente il valore univoco che identifica ciascun genere, Intero >0)

**OUTPUT**

- BaseDati (tipo: Database; base di dati modificata)

**CancellareUtente** (cancella tutti i dati relativi all'utente in input)

**INPUT**

- BaseDati (tipo: Database; il database nel quale verrà cancellato l'utente)
- IdUtente (campo contenente il valore univoco che identifica ciascun utente, Intero >0)

**OUTPUT**

- BaseDati (tipo: Database; base di dati modificata)

Backup dei dati da RAM a disco o viceversa: copia (a scelta dell'utente e a chiusura del programma) della base dati

**CaricareFile** (Carica i file del database in memoria)

**INPUT**

- File (tipo: File; sono i vari dati che rappresentano le tabelle del database)

**OUTPUT**

- BaseDati (tipo: Database; base di dati in cui sono stati caricati i vari file)

**SalvareFile** (copia lo stato attuale del database dalla memoria in file)

**INPUT**

- BaseDati (tipo: Database; base di dati da salvare nei diversi file sul disco)

**OUTPUT**

- file (tipo: File; file risultante dal salvataggio del database)

**BackupFile** (copia lo stato attuale del database)

**INPUT**

- file (tipo: File, i file attuali che rappresentano il database)

**OUTPUT**

- fileBackup (tipo: File, la copia dei file attuali su file diversi)

**RipristinareFile** (copia un precedente backup in memoria)

**INPUT**

- FileDiBackup (tipo: File; stato dell'ultimo backup da riportare in Database)

**OUTPUT**

- baseDati (tipo: Database; base di dati da ripristinare dai file sul disco)

Operazioni di interfaccia (stampe video e menù di scelta operazioni)

Stampa di report: (stampa di tutte o alcune delle informazioni relative a uno o più brani musicali selezionati)

**StampareListaBrani** (Stampa su file tutti i brani presenti nel database)

**INPUT**

- baseDati (tipo: Database; il database da cui selezionare i brani di cui fare il report)

**OUTPUT**

- FileReport (tipo: File, file contenente il report dell'elenco di brani selezionato dall'utente)

Ricerca nel Database (ricerca brani in base ai diversi filtri)

Le operazioni di ricerca sono le operazioni più eseguite su ogni base di dati e pertanto, devono essere efficienti e permettere agli utenti di cercare qualunque informazione sia di loro interesse. Ricerca di brani in base a qualunque dato "anagrafico":

**INPUT**

Dati inseriti dall'utente (dato anagrafico)

**OUTPUT**

- Risultato (tipo: vettore di interi che rappresentano gli id)

Questo è un insieme di funzioni che indica tutte le possibili funzioni di ricerca attuabili per dati anagrafici (una funzione per ogni dato). Ricerca lineare.

Le ricerche possibili sono:



**RicercaBranoTitolo** (Permette di cercare un brano nel sistema dando in input il titolo)

**INPUT**

- titoloBranoCercato (tipo: stringa, indica il titolo del brano ricercato)
- BaseDati (tipo: Database; il database dal quale cercare le informazioni richieste)

**OUTPUT**

- Risultato (tipo: vettore di interi che rappresentano gli id di tutti i brani aventi come titolo titoloBranoCercato)

**RicercaBranoAnno** (Permette di cercare un brano nel sistema dando in input il suo anno di creazione)

**INPUT**

- annoBranoCercato (tipo: intero, indica l'anno di creazione del brano ricercato)
- BaseDati (tipo: Database; il database dal quale cercare le informazioni richieste)

**OUTPUT**

- Risultato (tipo: vettore di interi che rappresentano gli id di tutti i brani aventi come anno di creazione annoBranoCercato)

- Ricerca tra playlist:

**INPUT**

Dati inseriti dall'utente (nome o altri parametri scelti dal gruppo)

**OUTPUT**

- Risultato (tipo: vettore di interi che rappresentano gli id)

Questo un insieme di funzioni che indica le funzioni di ricerca che si possono eseguire per la ricerca delle playlist. Ricerca lineare.

Le ricerche possibili sono:

**RicercaPlaylistUtente** (Permette di cercare una playlist nel sistema dando in input il nome dell'utente creatore della playlist)

**INPUT**

- nomeUtente (tipo: stringa, indica il nome dell'utente creatore della playlist cercata)
- BaseDati (tipo: Database; il database dal quale cercare le informazioni richieste)

**OUTPUT**

- Risultato (tipo: vettore di interi che rappresentano gli id di tutte le playlist aventi come creatore l'utente nomeUtente)

**RicercaPlaylistNome** (Permette di cercare una playlist nel sistema dando in input il suo nome)

**INPUT**

- nomePlaylistCercata (tipo: stringa, indica il nome della playlist cercata)
- BaseDati (tipo: Database; il database dal quale cercare le informazioni richieste)

**OUTPUT**

- Risultato (tipo: vettore di interi che rappresentano gli id di tutte le playlist aventi come nome nomePlaylistCercata)

- Ricerca tra utenti:

**INPUT**

Dati inseriti dall'utente (nome)

**OUTPUT**

Risultato (tipo: vettore di interi che rappresentano gli id)

Ricerca lineare del nome utente nell'array in cui sono elencati gli utenti.

Le ricerche possibili sono:

**RicercaUtenteNome** (Permette di cercare un utente nel sistema dando in input il nome dell'utente)

**INPUT**

- nomeUtente (tipo: stringa, indica il nome dell'utente cercato)
- BaseDati (tipo: Database; il database dal quale cercare le informazioni richieste)

**OUTPUT**

- Risultato (tipo: vettore di interi che rappresentano gli id di tutti gli utenti aventi come nome nomeUtente)

- Ricerca tra artisti:

**INPUT**

Dati inseriti dall'utente (nome, cognome o nome d'arte)

**OUTPUT**

- Risultato (tipo: vettore di interi che rappresentano gli id)

Questo un insieme di funzioni che indica le funzioni di ricerca che si possono eseguire per la ricerca degli artisti. Ricerca lineare.

Le ricerche possibili sono:

**RicercaArtistaNome** (Permette di cercare un artista nel sistema dando in input il suo nome)

**INPUT**

- nomeArtista (tipo: stringa, indica il nome dell'artista cercato)
- BaseDati (tipo: Database; il database dal quale cercare le informazioni richieste)

**OUTPUT**

- Risultato (tipo: vettore di interi che rappresentano gli id di tutti gli artisti aventi come nome nomeArtista)

**RicercaArtistaCognome** (Permette di cercare un artista nel sistema dando in input il suo cognome)

**INPUT**

- cognomeArtista (tipo: stringa, indica il cognome dell'artista cercato)
- BaseDati (tipo: Database; il database dal quale cercare le informazioni richieste)

**OUTPUT**

- Risultato (tipo: vettore di interi che rappresentano gli id di tutti gli artisti aventi come cognome cognomeArtista)

**RicercaArtistaNomeArte** (Permette di cercare un artista nel sistema dando in input il suo nome d'arte)

**INPUT**

- nomeArte (tipo: stringa, indica il nome d'arte dell'artista cercato)
- BaseDati (tipo: Database; il database dal quale cercare le informazioni richieste)

**OUTPUT**

- Risultato (tipo: vettore di interi che rappresentano gli id di tutti gli artisti aventi come nome d'arte nomeArte)

- Ricerca tra album:

**INPUT**

Dati inseriti dall'utente (titolo o anno)

**OUTPUT**

- Risultato (tipo: vettore di interi che rappresentano gli id)

Questo un insieme di funzioni che indica le funzioni di ricerca che si possono eseguire per la ricerca degli album. Ricerca lineare.

Le ricerche possibili sono:

**RicercaAlbumTitolo** (Permette di cercare un album nel sistema dando in input il suo titolo)

**INPUT**

- titoloAlbumCercato (tipo: stringa, indica il titolo dell'album ricercato)
- BaseDati (tipo: Database; il database dal quale cercare le informazioni richieste)

**OUTPUT**

- Risultato (tipo: vettore di interi che rappresentano gli id di tutti gli album aventi come titolo titoloAlbumCercato)

**RicercaAlbumAnno** (Permette di cercare un album nel sistema dando in input il suo anno di creazione)

**INPUT**

- annoAlbumCercato (tipo: intero, indica l'anno di creazione dell'album ricercato)
- BaseDati (tipo: Database; il database dal quale cercare le informazioni richieste)

**OUTPUT**

- Risultato (tipo: vettore di interi che rappresentano gli id di tutti gli album aventi come anno di creazione annoAlbumCercato)

## Operazioni di registrazione

**LoginUtente** (Permette di accedere al sistema attraverso l'uso delle credenziali username e password presenti)

### **INPUT**

- username (tipo: Stringa, l'username dell'utente)
- password (tipo: Stringa, la password dell'utente)
- BaseDati (tipo: Database; il database dal quale cercare le informazioni dell'utente)

### **OUTPUT**

- baseDati (tipo: Database, il database con l'esito del login ed eventualmente l'utente loggato)