Stampe gadget



# INDICE

[Titolo del progetto 1](#_heading=h.44sinio)

[INDICE 2](#_heading=h.gjdgxs)

[Analisi 3](#_heading=h.30j0zll)

[Descrizione del sistema 3](#_heading=h.1fob9te)

[1.2](#_heading=h.3znysh7) Requisiti Funzionali 4

[2.](#_heading=h.2et92p0) Progettazione 5

[2.1](#_heading=h.tyjcwt) Progettazione dei tipi di dato, delle strutture dati 5

[2.2](#_heading=h.3dy6vkm) Progettazione della soluzione prodotta 7

[<Indicare con un Albero di Decomposizione come è stata progettata la soluzione prodotta. Aggiungere lo Pseudocodice delle **funzionalità principali**: Main, Inserimento dati, Ricerca, ecc. Non inserire i controlli di data, delle stringhe, ecc.> 7](#_heading=h.1t3h5sf)

[2.3](#_heading=h.4d34og8) Progettazione delle librerie / funzioni (solo se in casi eccezionali in cui non si riesce ad utilizzare DOXYGEN) 7

[4.](#_heading=h.2s8eyo1) Testing 8

[4.1](#_heading=h.17dp8vu) Definizione del Piano di Test 8

[4.2](#_heading=h.3rdcrjn) Esiti del Piano di Test ed eventuali commenti 8

[5. Presentazione della soluzione 9](#_heading=h.26in1rg)

[5.1 Funzionalità Fondamentali 9](#_heading=h.lnxbz9)

[5.2 Funzionalità aggiuntive/opzionali 9](#_heading=h.35nkun2)

# Analisi

## Descrizione del sistema

Si vuole realizzare un programma che simuli una parte delle funzionalità di un software per una smart tv per suggerire programmi televisivi. Il software lavora principalmente raccogliendo informazioni sui programmi televisivi di interesse dell’utente e fornendo dei suggerimenti su programmi che potrebbero essere di interesse per l’utente.

Nella progettazione del sistema si è ipotizzata la presenza di un unico utente. L’utente ha un proprio profilo costituito dai seguenti dati:

* **Codice utente**
* **Nome**
* **Cognome**
* **Tipo di programma d’interesse**
* **Generi d’interesse**

Il **codice utente** viene acquisito come una stringa di 6 cifre. Il codice permette di identificare un determinato programma perciò è univoco.

Il **nome** e il **cognome** sono entrambe delle stringhe 20 di caratteri. Se il nome è composto da più parole esse sono inserite separandole con un “\_”.

I **tipo di programma d’interesse** sono le tipologie di programmi che l’utente preferisce: serie, documentari, film, intrattenimento, bambini e informazione. L’utente può specificare al massimo 3 tipi di programma d’interesse.

I **generi di interesse** riguardano il contenuto dei programmi che l’utente preferisce. I generi sono: fantascienza, cucina, natura, sport, storico, horror, attualità, comico, drammatico, fantasy e romantici. L’utente può specificare al massimo 5 argomenti d’interesse.

Il software verterà sui programmi. Per ciascun programma sono riportati i seguenti dati:

* **Codice**
* **Titolo**
* **Tipologia**
* **Genere**
* **Descrizione**
* **Valutazione**

Il **codice** viene acquisito come una stringa di 6 cifre. Il codice permette di identificare un determinato programma perciò è univoco.

Il **titolo** è una stringa di 30 caratteri (lettere, numeri, caratteri speciali).

La **tipologia** è una stringa di 16 caratteri che indica il tipo di programma: serie, documentari, film, intrattenimento, bambini e informazione. Il programma può avere una sola tipologia.

Il **genere** indica la tematica del programma: fantascienza, cucina, natura, sport, storico, horror, attualità, comico, drammatico, fantasy e romantici. Esso è rappresentato da una stringa di 13 caratteri. Un programma può essere di un solo genere.

La **descrizione** è una stringa di 50 caratteri che contiene una breve testo che riporta il contenuto del programma.

La **valutazione** è una struttura dati che contiene il numero di valutazioni fatte per quel programma e la media delle valutazioni. È un valore compreso tra 0 a 5 e rappresenta l’indice di gradimento del programma per gli utenti. Lo 0 indica che non sono state espresse valutazioni.

Il sistema si occuperà di gestire i programmi che sono stati visti da un utente. La **lista programmi visti** è un elenco che riporta il codice utente e il codice di tutti i programmi che l’utente ha visto, cioè quelli che l’utente ha riprodotto.

Il sistema si occuperà nello specifico:

1. **Aggiunta-Modifica-Cancellazione dei programmi TV**.

I dati relativi ai programmi tv vengono salvati per un utilizzo successivo. Il sistema permette di inserire nuovi programmi tv di cui vengono memorizzati i dati sopra riportati all’infuori della valutazione che ha come valore iniziale 0. Al contempo è possibile modificare alcuni dati relativi al programma tv; in particolare genere, titolo, descrizione e tipologia. Infine, il sistema permette di eliminare un programma tv.

1. **Aggiunta e Modifica del profilo dell’utente**

Il sistema permette all’utente di creare un profilo personale e successivamente di modificarne i dati inseriti. In particolare può modificare il proprio nome, cognome, tipo di programma d’interesse e genere d’interesse.

1. **Ricerca del programma TV**

Il sistema permette all’utente di ricercare un programma tv in base ad uno di questi parametri: titolo, genere, codice e tipologia.

1. **Valutazione dei programmi proposti**

Il sistema permette all’utente di ricercare uno dei programmi che ha visto e di valutarlo. La valutazione è un valore compreso tra 1 e 5.

Se la lista dei programmi visti è vuota all’utente viene comunicato che non può effettuare valutazioni.

1. **Suggerimento**

Il sistema propone all’utente una serie di programmi in base al genere d’interesse dell’utente, alla tipologia e alla valutazione che riporta. Sono esclusi dai suggerimenti quei programmi che rientrano nella lista dei programmi visti.

1. **Funzionalità di ordinamento**

Il sistema permette di ordinare i vari programmi proposti nelle varie sezioni in base all’ordine alfabetico crescente del titolo.

1. **Gestione Funzionalità tramite Menu Utente**

L’utente gestirà tutte le funzionalità del sistema attraverso l’utilizzo di un menu.

Accanto alle funzioni di base previste dal sistema, è stata implementata anche la seguente funzione aggiuntiva:

1. **Riproduzione del programma**

All’utente sarà permesso di ricercare il programma di interesse e di riprodurlo. Alla scelta di questa funzionalità il sistema riporta che “il programma x è stato riprodotto”. Di seguito sarà memorizzato che l’utente y ha riprodotto il programma x.

## 1.2 Requisiti Funzionali

| Codice | Nome | Descrizione |
| --- | --- | --- |
| R01 | Inserisci programma | Il sistema deve consentire all’utente di inserire i dati relativi ad un nuovo programma che poi saranno memorizzati su un file |
| R02 | Modifica programma | Il sistema deve consentire all’utente di modificare i dati relativi ad un programma ricercato. Il file viene aggiornato con le modifiche. |
| R03 | Elimina programma | Il sistema accedendo al file dei programmi potrà eliminare i dati relativi ad un programma. |
| R04 | Inserisci utente | Il sistema deve consentire di inserire i dati relativi ad un nuovo utente che poi saranno memorizzati su un file |
| R05 | Modifica utente | Il sistema deve consentire di modificare i dati relativi ad un utente. Il file viene aggiornato con le modifiche. |
| R06 | Ricerca programma | Il sistema permette all’utente di ricercare un programma in base al titolo, al genere, al codice o alla tipologia immessa in input |
| R07 | Valutazione programma | Il sistema consente all’utente di valutare uno tra i programmi che sono presenti nella lista dei programmi visti. Il voto va da 1 a 5. |
| R08 | Suggerimento |  |
| R09 | Ordinamento | Il sistema permette di ordinare i dati in base al titolo in ordine crescente. |
| R10 | Visualizzazione Menu | Il sistema deve mostrare all’utente un menu iniziale dove vengono visualizzate le opzioni disponibili. |
| R11 | Riprodurre programma | Il sistema consente all’utente di riprodurre un programma a sua scelta. Memorizza, inoltre, che il programma è stato riprodotto |

# 2. Progettazione

## 2.1 Progettazione dei tipi di dato, delle strutture dati

<Indicare i tipi di dato e le strutture dati utilizzate nell’ambito del progetto. Indicare anche eventuali file utilizzati nel programma e il relativo scopo>

*Esempio*

| Nome | Tipologia | Descrizione | Tipi / Campi / Valori |
| --- | --- | --- | --- |
| User | struct | Tipo di dato definito per descrivere le caratteristiche dell’utente | Nome: char[20]  Cognome: char[20]  Etc. etc. |
| Preferences | enum | Tipo di Dato per memorizzare le scelte dell’utente | Like / dislike |
| … | … | … | … |
|  |  |  |  |

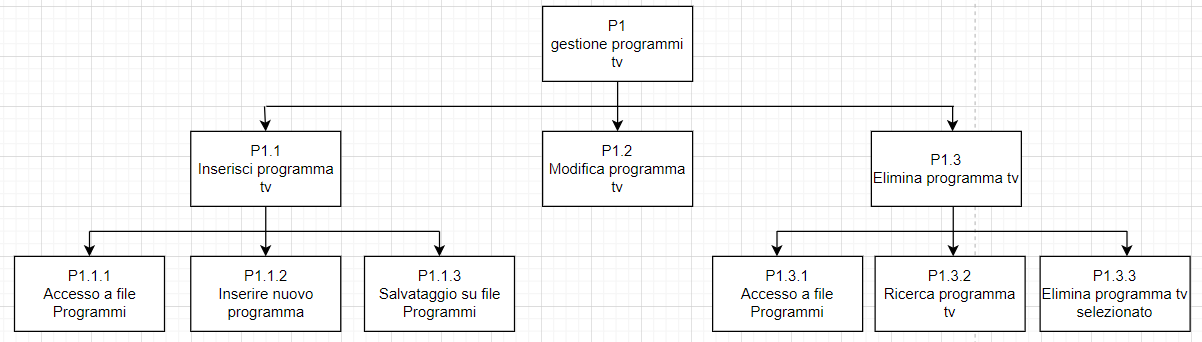
| Nome | Tipologia | Descrizione | Tipi / Campi / Valori |
| --- | --- | --- | --- |
| Max\_Items | costante | Costante utilizzata per memorizzare il massimo numero di oggetti registrabili | 1000 |
| A | Variabile globale | Variabile globale utilizzata per…. | int |
| … | … | … | … |
| Dat.dat | File | File utilizzato per… | …. |

***Non serve indicare le variabili locali!***

## 2.2 Progettazione della soluzione prodotta

## <Indicare con un Albero di Decomposizione come è stata progettata la soluzione prodotta. Aggiungere lo Pseudocodice delle **funzionalità principali**: Main, Inserimento dati, Ricerca, ecc. Non inserire i controlli di data, delle stringhe, ecc.>

## 



## 

## 2.3 Progettazione delle librerie / funzioni (solo se in casi eccezionali in cui non si riesce ad utilizzare DOXYGEN)

<Indicare quali sono le librerie progettate. Per ciascun file .h indicare le procedure e le funzioni incluse nell’header file. Per ogni funzione indicare scopo, tipi di ingresso e di uscita.

# 4. Testing

## 4.1 Definizione del Piano di Test

<Per ciascun caso d’uso definito in sezione 1.2 definire i casi di test e validarne l’esito. Quando possibile, progettare i casi di test come asserzioni CUnit>

| Codice Requisito | Codice Test | Nome | Descrizione Test | Eventuale Input | Risultato Atteso | Risultato Ottenuto |
| --- | --- | --- | --- | --- | --- | --- |
| R01 | 1.1 | Menu Iniziale | Scelta n.1 | 1 | Caricamento del menu X | Programma chiuso |
| R02 | 1.2 | Menu iniziale | Scelta errata | 100 | Messaggio di Errore | <indicare risultato> |
| R02 | 2.1 | Caricamento Dati da File | File non esistente | xxx | Visualizzazione Messaggio di Errore e Creazione Nuovo File | <indicare risultato> |
| R02 | 2.2 | Caricamento Dati da File | File Esistente | xxx | Caricamento dei File effettuato correttamente | <indicare risultato> |

## 4.2 Esiti del Piano di Test ed eventuali commenti

<Commentare gli esiti del piano di test, individuare eventuali criticità (test che l’attuale implementazione non riesce a superare) e pianificare eventuali azioni migliorative sul codice>

# 5. Presentazione della soluzione

## 5.1 Funzionalità Fondamentali

<Per ogni Requisito funzionale commentare il funzionamento della soluzione prodotta utilizzando screenshot e commenti per presentare i risultati a video, No programmaizone difensiva>

## 5.2 Funzionalità aggiuntive/opzionali

<Tutte le eventuali funzionalità aggiuntive che il gruppo degli autori ritiene di voler mettere in evidenza può essere commentata in questa sezione>