Documentazione

[1 Introduzione 2](#__RefHeading___Toc1221_2866232661)

[1.1 Informazioni sul progetto 2](#__RefHeading___Toc1223_2866232661)

[1.2 Abstract 2](#__RefHeading___Toc1225_2866232661)

[1.3 Scopo 2](#__RefHeading___Toc1227_2866232661)

[2 Analisi 3](#__RefHeading___Toc1229_2866232661)

[2.1 Analisi del dominio 3](#__RefHeading___Toc1231_2866232661)

[2.2 Analisi e specifica dei requisiti 3](#__RefHeading___Toc1233_2866232661)

[2.3 Use case 6](#__RefHeading___Toc1235_2866232661)

[2.4 Pianificazione 6](#__RefHeading___Toc1237_2866232661)

[2.5 Analisi dei mezzi 7](#__RefHeading___Toc1239_2866232661)

[2.5.1 Software 7](#__RefHeading___Toc1241_2866232661)

[2.5.2 Hardware 7](#__RefHeading___Toc1243_2866232661)

[3 Progettazione 7](#__RefHeading___Toc1245_2866232661)

[3.1 Design dell’architettura del sistema 7](#__RefHeading___Toc1247_2866232661)

[3.2 Design dei dati e database 8](#__RefHeading___Toc1249_2866232661)

[3.3 Design delle interfacce 8](#__RefHeading___Toc1251_2866232661)

[3.4 Design procedurale 8](#__RefHeading___Toc1253_2866232661)

[4 Implementazione 9](#__RefHeading___Toc1255_2866232661)

[5 Test 9](#__RefHeading___Toc1257_2866232661)

[5.1 Protocollo di test 9](#__RefHeading___Toc1259_2866232661)

[5.2 Risultati test 10](#__RefHeading___Toc1261_2866232661)

[5.3 Mancanze/limitazioni conosciute 10](#__RefHeading___Toc1263_2866232661)

[6 Consuntivo 10](#__RefHeading___Toc1265_2866232661)

[7 Conclusioni 10](#__RefHeading___Toc1267_2866232661)

[7.1 Sviluppi futuri 10](#__RefHeading___Toc1269_2866232661)

[7.2 Considerazioni personali 10](#__RefHeading___Toc1271_2866232661)

[8 Bibliografia 10](#__RefHeading___Toc1273_2866232661)

[8.1 Bibliografia per articoli di riviste: 10](#__RefHeading___Toc1275_2866232661)

[8.2 Bibliografia per libri 10](#__RefHeading___Toc1277_2866232661)

[8.3 Sitografia 10](#__RefHeading___Toc1279_2866232661)

[9 Allegati 11](#__RefHeading___Toc1281_2866232661)

# Introduzione

## Informazioni sul progetto

Allievi coinvolti nel progetto: Michea Colautti

Classe: Informatica 3AC presso la sede Scuola Arti e Mestieri Trevano

Docenti responsabili: Geo Petrini

Data inizio: 01.09.2020  
Data consegna: 23.12.2020

## Abstract

Per questo progetto è stato richiesto di sviluppare un software in grado di generare dei labirinti 2d con svariate forme o opzioni configurabili. Grazie a questo utile e divertente software sarà quindi possibile mettersi alla prova con svariati labirinti di difficoltà disparate, che saranno risolvibile direttamente sul sito. Inoltre sarà possibile ricreare questi labirinti tramite i parametri oppure un seme di generazione. Sarà possibile pure il salvataggio dei labirinti tramite un’immagine e la possibilità di visualizzare la soluzione

## Scopo

L’obiettivo del progetto è lo sviluppo di un’applicazione per la generazione di labirinti 2D.

In questa generazione dovrà essere possibile configurare dei parametri come la forma esterna e del percorso, l’inserimento di percorsi a loop, le posizioni di partenza e fine, le dimensioni del labirinto, l possibilità di esportare la soluzione tramite un’immagine e la generazione di soluzioni multiple. l’utente

L’utente non deve solo avere la possibilità di salvare i parametri oppure un seed di generazione, ma deve essere anche in grado di completare il labirinto online, lasciando dietro di sé la “scia” con il percorso seguito

# Analisi

## Analisi del dominio

Per questo progetto è stato chiesto di sviluppare un software, web nel mio caso, che permetta di generare labirinti 2D. Gli utenti a cui esso sarà indirizzato non dovranno avere particolari conoscenze informatiche, l’interfaccia grafica semplice permetterà infatti a tutti gli utenti il libero.

I labirinti saranno generati in maniera randomica e anche se esistono diversi siti web capaci di generare enigmi simili pochi di questi combinano la grande personalizzazione del labirinto con la possibilità di completarlo comodamente online.

Non sarà necessario creare strati di sicurezza, in quanto il progetto non richiede un database e sarà un'unica pagina internet e il codice java script è comunque in chiaro.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Home Page |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Un pagina html semplice in cui disporre i controlli per la personalizzazione del labirinto |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Campi di controllo per la personalizzazione |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Bottoni e campi di testo per la personalizzazione del labirinto |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Logica per la generazione del labirinto (partendo da un seed) |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Un algoritmo che generi il labirinto partendo da un numero casuale/inserito dall’utente |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Metodo che permetta il salvataggio del labirinto |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | Libreria (Jquery o simili) |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Per permettere all’utente di disegnare il labirinto |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Sistema di salvataggio dei parametri di personalizzazione |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Si può utilizzare il metodo per salvare il labirinto per fare una foto ai parametri selezionati |

## Use case

I casi d’uso rappresentano l’interazione tra i vari attori e le funzionalità del prodotto.

## Pianificazione

Prima di stabilire una pianificazione bisogna avere almeno una vaga idea del modello di sviluppo che si intende adottare. In questa sezione bisognerà inserire il modello concettuale di sviluppo che si seguirà durante il progetto. Gli elementi di riferimento per una buona pianificazione derivano da una scomposizione top-down della problematica del progetto.

La pianificazione può essere rappresentata mediante un diagramma di Gantt:

|  |
| --- |
| Figura 1: Gantt preventivo |

Se si usano altri metodi di pianificazione (es scrum), dovranno apparire in questo capitolo.

## Analisi dei mezzi

Elencare e descrivere i mezzi disponibili per la realizzazione del progetto. Ricordarsi di sempre descrivere nel dettaglio le versioni e il modello di riferimento.

### Software

SDK, librerie, tools utilizzati per la realizzazione del progetto e eventuali dipendenze.

### Hardware

Su quale piattaforma dovrà essere eseguito il prodotto? Che hardware particolare è coinvolto nel progetto? Che particolarità e limitazioni presenta? Che hw sarà disponibile durante lo sviluppo?

# Progettazione

Questo capitolo descrive esaustivamente come deve essere realizzato il prodotto fin nei suoi dettagli. Una buona progettazione permette all’esecutore di evitare fraintendimenti e imprecisioni nell’implementazione del prodotto.

## Design dell’architettura del sistema

Descrive:

* La struttura del programma/sistema lo schema di rete...
* Gli oggetti/moduli/componenti che lo compongono.
* I flussi di informazione in ingresso ed in uscita e le relative elaborazioni. Può utilizzare *diagrammi di flusso dei dati* (DFD).
* Eventuale sitemap

## Design dei dati e database

Descrizione delle strutture di dati utilizzate dal programma in base agli attributi e le relazioni degli oggetti in uso.

Schema E-R, schema logico e descrizione.

Se il diagramma E-R viene modificato, sulla doc dovrà apparire l’ultima versione, mentre le vecchie saranno sui diari.

## Design delle interfacce

Descrizione delle interfacce interne ed esterne del sistema e dell’interfaccia utente. La progettazione delle interfacce è basata sulle informazioni ricavate durante la fase di analisi e realizzata tramite mockups.

## Design procedurale

Descrive i concetti dettagliati dell’architettura/sviluppo utilizzando ad esempio:

* Diagrammi di flusso e Nassi.
* Tabelle.
* Classi e metodi.
* Tabelle di routing
* Diritti di accesso a condivisioni …

Questi documenti permetteranno di rappresentare i dettagli procedurali per la realizzazione del prodotto.

# Implementazione

In questo capitolo dovrà essere mostrato come è stato realizzato il lavoro. Questa parte può differenziarsi dalla progettazione in quanto il risultato ottenuto non per forza può essere come era stato progettato.

Sulla base di queste informazioni il lavoro svolto dovrà essere riproducibile.

In questa parte è richiesto l’inserimento di codice sorgente/print screen di maschere solamente per quei passaggi particolarmente significativi e/o critici.

Inoltre dovranno essere descritte eventuali varianti di soluzione o scelte di prodotti con motivazione delle scelte.

Non deve apparire nessuna forma di guida d’uso di librerie o di componenti utilizzati. Eventualmente questa va allegata.

Per eventuali dettagli si possono inserire riferimenti ai diari.

# Test

## Protocollo di test

Definire in modo accurato tutti i test che devono essere realizzati per garantire l’adempimento delle richieste formulate nei requisiti. I test fungono da garanzia di qualità del prodotto. Ogni test deve essere ripetibile alle stesse condizioni.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-012 | **Nome:** | Import a card with KIC, KID and KIK keys, but not shown with the GUI |
| **Descrizione:** | Import a card with KIC, KID and KIK keys with no obfuscation, but not shown with the GUI | | |
| **Prerequisiti:** | Store on local PC: Profile\_1.2.001.xml (appendix n\_n) and Cards\_1.2.001.txt (appendix n\_n).  PIN (OTA\_VIEW\_PIN\_PUK\_KEY) and ADM (OTA\_VIEW\_ADM\_KEY) user right not set. | | |
| **Procedura:** | 1. Go to “Cards manager” menu,  in main page click “Import Profiles” link, Select the “1.2.001.xml” file, Import the Profile 2. Go to “Cards manager” menu,  in main page click “Import Cards” link, Select the “1.2.001.txt” file, Delete the cards,  Select the “1.2.001.txt” file, Import the cards 3. Research the “41795924770” Card, Click the imsi card link Check the card details 4. Execute the SQL: SELECT imsi, dir, keyset, cntr, rawtohex(kickey), rawtohex(kidkey), rawtohex(kikkey), rawtohex(chv), rawtohex(dap)FROM otacardkey a where imsi='340041795924770' ORDER BY keyset; | | |
| **Risultati attesi:** | Keys visible in the DB (OtaCardKey) but not visible in the GUI (Card details) | | |

## Risultati test

Tabella riassuntiva in cui si inseriscono i test riusciti e non del prodotto finale. Se un test non riesce e viene corretto l’errore, questo dovrà risultare nel documento finale come riuscito (la procedura della correzione apparirà nel diario), altrimenti dovrà essere descritto l’errore con eventuali ipotesi di correzione.

## Mancanze/limitazioni conosciute

Descrizione con motivazione di eventuali elementi mancanti o non completamente implementati, al di fuori dei test case. Non devono essere riportati gli errori e i problemi riscontrati e poi risolti durante il progetto.

# Consuntivo

Consuntivo del tempo di lavoro effettivo e considerazioni riguardo le differenze rispetto alla pianificazione (cap 1.7) (ad esempio Gannt consuntivo).

# Conclusioni

Quali sono le implicazioni della mia soluzione? Che impatto avrà? Cambierà il mondo? È un successo importante? È solo un’aggiunta marginale o è semplicemente servita per scoprire che questo percorso è stato una perdita di tempo? I risultati ottenuti sono generali, facilmente generalizzabili o sono specifici di un caso particolare? ecc

## Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

## Considerazioni personali

Cosa ho imparato in questo progetto? ecc

# Bibliografia

## Bibliografia per articoli di riviste:

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo dell’articolo (tra virgolette),
3. Titolo della rivista (in italico),
4. Anno e numero
5. Pagina iniziale dell’articolo,

## Bibliografia per libri

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo del libro (in italico),
3. ev. Numero di edizione,
4. Nome dell’editore,
5. Anno di pubblicazione,
6. ISBN.

## Sitografia

1. URL del sito (se troppo lungo solo dominio, evt completo nel diario),
2. Eventuale titolo della pagina (in italico),
3. Data di consultazione (GG-MM-AAAA).

**Esempio:**

* http://standards.ieee.org/guides/style/section7.html, *IEEE Standards Style Manual*, 07-06-2008.

# Allegati

Elenco degli allegati, esempio:

* Diari di lavoro
* Codici sorgente/documentazione macchine virtuali
* Istruzioni di installazione del prodotto (con credenziali di accesso) e/o di eventuali prodotti terzi
* Documentazione di prodotti di terzi
* Eventuali guide utente / Manuali di utilizzo
* Mandato e/o Qdc
* Prodotto
* …