DOCUMENTAZIONE PAROLE CROCIATE

# Indice

Sommario

[1 Indice 2](#_Toc145054952)

[2 Introduzione 3](#_Toc145054953)

[2.1 Informazioni sul progetto 3](#_Toc145054954)

[2.2 Abstract 3](#_Toc145054955)

[2.3 Scopo 4](#_Toc145054956)

[3 Analisi 5](#_Toc145054957)

[3.1 Analisi del dominio 5](#_Toc145054958)

[3.2 Analisi e specifica dei requisiti 5](#_Toc145054959)

[3.3 Use case 9](#_Toc145054960)

[3.4 Pianificazione 9](#_Toc145054961)

[3.5 Analisi dei mezzi 11](#_Toc145054962)

[3.5.1 Software 11](#_Toc145054963)

[3.5.2 Hardware 11](#_Toc145054964)

[4 Progettazione 11](#_Toc145054965)

[4.1 Design dell’architettura del sistema 11](#_Toc145054966)

[4.2 Design dei dati e database 11](#_Toc145054967)

[4.3 Design delle interfacce 14](#_Toc145054968)

[4.4 Design procedurale 14](#_Toc145054969)

[5 Implementazione 15](#_Toc145054970)

[6 Test 15](#_Toc145054971)

[6.1 Protocollo di test 15](#_Toc145054972)

[6.2 Risultati test 17](#_Toc145054973)

[6.3 Mancanze/limitazioni conosciute 17](#_Toc145054974)

[7 Consuntivo 17](#_Toc145054975)

[8 Conclusioni 19](#_Toc145054976)

[8.1 Sviluppi futuri 19](#_Toc145054977)

[8.2 Considerazioni personali 19](#_Toc145054978)

[9 Bibliografia 20](#_Toc145054979)

[9.1 Bibliografia per articoli di riviste: 20](#_Toc145054980)

[9.2 Bibliografia per libri 20](#_Toc145054981)

[9.3 Sitografia 20](#_Toc145054982)

[10 Glossario 21](#_Toc145054983)

[11 Indice delle figure 21](#_Toc145054984)

[12 Allegati 21](#_Toc145054985)

# Introduzione

## Informazioni sul progetto

Il progetto consiste nel generare una “tabella” di lettere al cui interno ci sono delle parole da trovare.

Le parole da cercare sono fornite sotto la tabella sotto forma di elenco.

Se vengono trovate dall’utente, le parole vengono rimosse dall’elenco e marchiate in verde nella tabella.

Persone coinvolte nel progetto:

* Riccardo Deiuri 🡪 principale programmatore e realizzatore del progetto
* Geo Petrini 🡪 Project Manager e aiutante

Scuola Arti & Mestieri Trevano, sezione informatica, Laboratorio progetti

Data di inizio progetto: 01.09.2023

Data di consegna: **01.12.2023**

## Abstract

È una breve e accurata rappresentazione dei contenuti di un documento, senza notazioni critiche o valutazioni. Lo scopo di un abstract efficace dovrebbe essere quello di far conoscere all’utente il contenuto di base di un documento e metterlo nella condizione di decidere se risponde ai suoi interessi e se è opportuno il ricorso al documento originale.

Può contenere alcuni o tutti gli elementi seguenti:

* **Background/Situazione iniziale**
* **Descrizione del problema e motivazione**: Che problema ho cercato di risolvere? Questa sezione dovrebbe includere l'importanza del vostro lavoro, la difficoltà dell'area e l'effetto che potrebbe avere se portato a termine con successo.
* **Approccio/Metodi**: Come ho ottenuto dei progressi? Come ho risolto il problema (tecniche…)? Quale è stata l’entità del mio lavoro? Che fattori importanti controllo, ignoro o misuro?
* **Risultati**: Quale è la risposta? Quali sono i risultati? Quanto è più veloce, più sicuro, più economico o in qualche altro aspetto migliore di altri prodotti/soluzioni?

Esempio di abstract:

*As the size and complexity of today’s most modern computer chips increase, new techniques must be developed to effectively design and create Very Large-Scale Integration chips quickly. For this project, a new type of hardware compiler is created. This hardware compiler will read a C++ program, and physically design a suitable microprocessor intended for running that specific program. With this new and powerful compiler, it is possible to design anything from a small adder, to a microprocessor with millions of transistors. Designing new computer chips, such as the Pentium 4, can require dozens of engineers and months of time. With the help of this compiler, a single person could design such a large-scale microprocessor in just weeks.*

## Scopo

Lo scopo didattico del progetto è quello di riuscire a gestire al meglio la pianificazione e la realizzazione di un progetto. Per poi prepararci al futuro professionale.

Lo scopo operativo è quello invece di creare un applicativo WEB che generi un parole crociate affinché successivamente si possa stampare e giocarci.

# Analisi

## Analisi del dominio

Questo prodotto verrà utilizzato da tutti coloro che vogliono giocare a parole crociate ma non hanno delle riviste, oppure voglia di cercare in qualche sito dove troverebbe tabelle prese da giornali, potendo, invece in questo prodotto, generare una tabella facilmente e velocemente.

Questo progetto è molto universale, infatti l’utente finale potrebbe variare da un bambino di 6 anni a un anziano di 90 anni.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **Req-001** | |
| **Nome** | Creazione interfaccia |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Bisogna generare una tabella di lettere con all’interno delle parole messe in qualsiasi direzione, e elencarle sotto la tabella. |
| **Sotto requisiti** | |
| **Req-001\_1** | Si necessita di un dizionario |

|  |  |
| --- | --- |
| **Req-002** | |
| **Nome** | Modalità di gioco |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Modalità bambini e modalità adulti (parola finale con lettere rimanenti) |

|  |  |
| --- | --- |
| **Req-003** | |
| **Nome** | Difficoltà di gioco |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessita di poter scegliere tra quattro difficoltà in cui varia la grandezza della tabella e il numero di parole. |
| **Sotto requisiti** | |
| **Req-003\_1** | Si necessita di un input per far scegliere l’utente. |

|  |  |
| --- | --- |
| **Req-004** | |
| **Nome** | Regole di gioco |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Le parole non si possono sovrapporre a meno che non siano orientate diversamente l’una dall’altra. |
| **Sotto requisiti** | |
| **Req-004\_1** | Si necessita di non avere né accenti né apostrofi |
| **Req-004\_2** | Si necessita che tutte le lettere della tabella siano in maiuscolo |
| **Req-004\_3** | Si necessita che le parole possano essere in tutte le direzioni possibili: orizzontale, orizzontale invertito, verticale, verticale invertito, obliquo verso destra sotto, obliquo verso destra sopra, obliquo verso sinistra sotto, obliquo verso sinistra sopra. |

|  |  |
| --- | --- |
| **Req-005** | |
| **Nome** | Export |
| **Priorità** | 0.8 |
| **Versione** | 1.0 |
| **Note** | Si necessita inserire la possibilità di esportare la tabella in formato .txt |

|  |  |
| --- | --- |
| **Req-006** | |
| **Nome** | Scelta del font |
| **Priorità** | 0.5 |
| **Versione** | 1.0 |
| **Note** | Si necessita di inserire la possibilità di scegliere il font |

|  |  |
| --- | --- |
| **Req-007** | |
| **Nome** | Inserimento dizionario |
| **Priorità** | 0.5 |
| **Versione** | 1.0 |
| **Note** | Si necessita di inserire la possibilità di aggiungere un dizionario scelto dall’utente. |

**Spiegazione elementi tabella dei requisiti:**

**ID**: identificativo univoco del requisito

**Nome**: breve descrizione del requisito

**Priorità**: indica l’importanza di un requisito nell’insieme del progetto, definita assieme al committente. Ad esempio, poter disporre di report con colonne di colori diversi ha priorità minore rispetto al fatto di avere un database con gli elementi al suo interno. Solitamente si definiscono al massimo di 2-3 livelli di priorità.

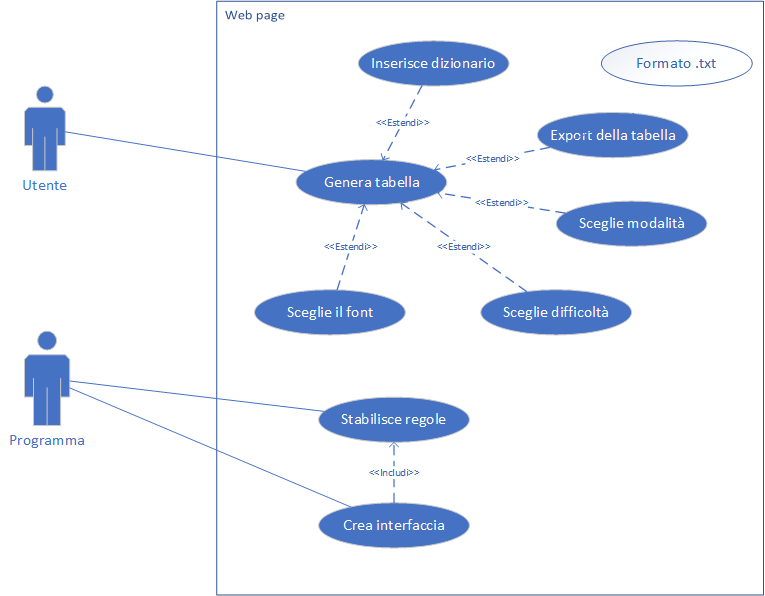
**Versione**: indica la versione del requisito. Ogni modifica del requisito avrà una versione aggiornata.

Sulla documentazione apparirà solamente l’ultima versione, mentre le vecchie dovranno essere inserite nei diari.

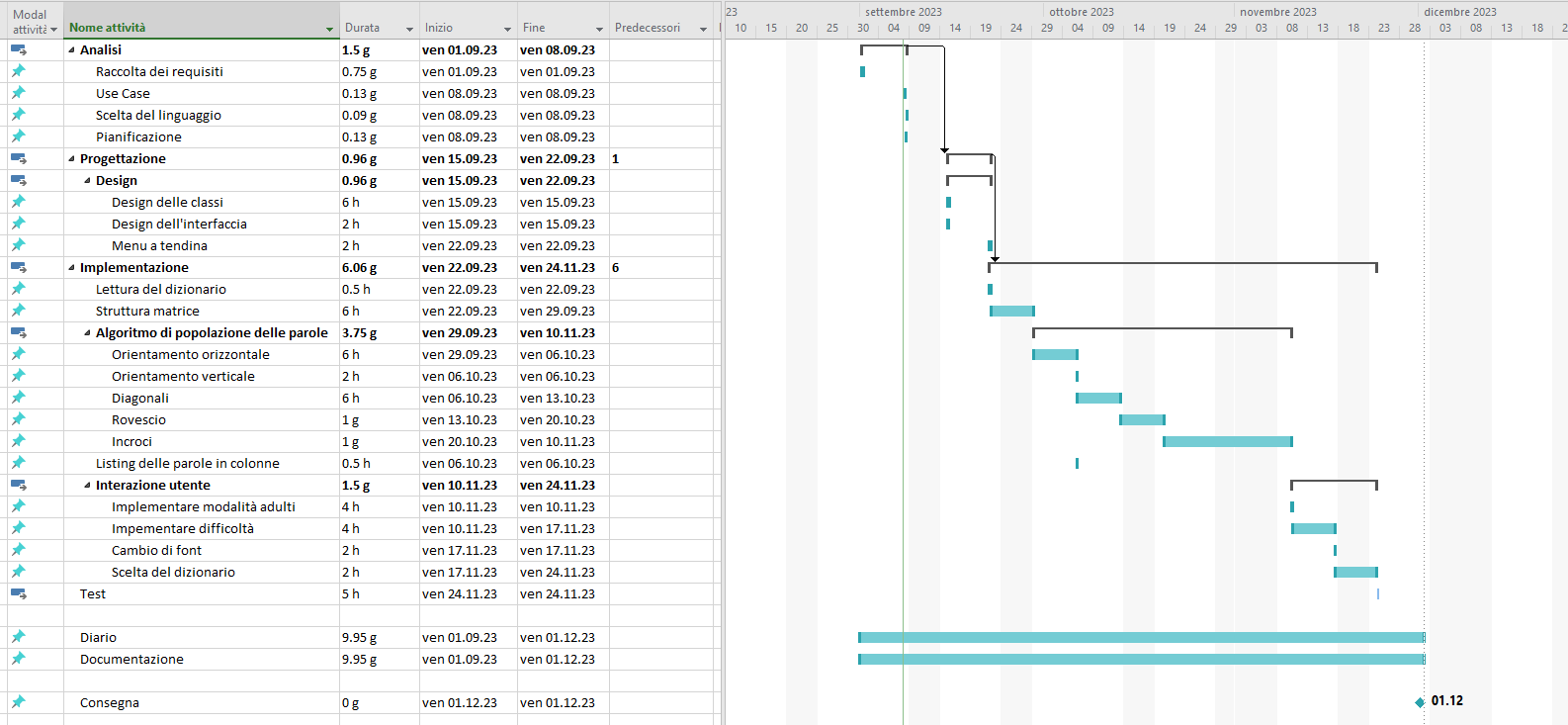
**Note**: eventuali osservazioni importanti o riferimenti ad altri requisiti.

**Sotto requisiti**: elementi che compongono il requisito.

## Use case



## Pianificazione



Stile orizzontale

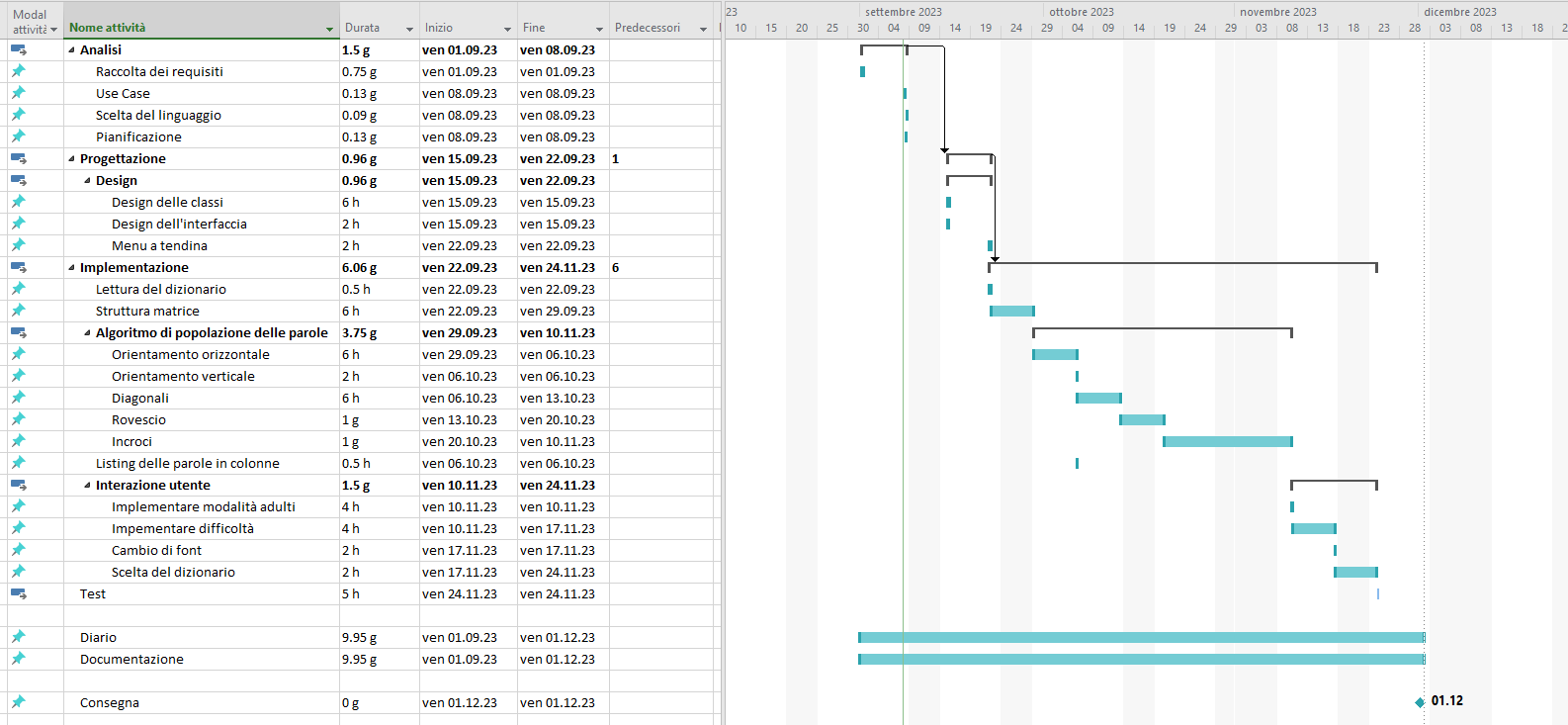


Figura Esempio di diagramma di Gantt

## 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.

Diagramma Chen

Figura Diagramma ER Chen

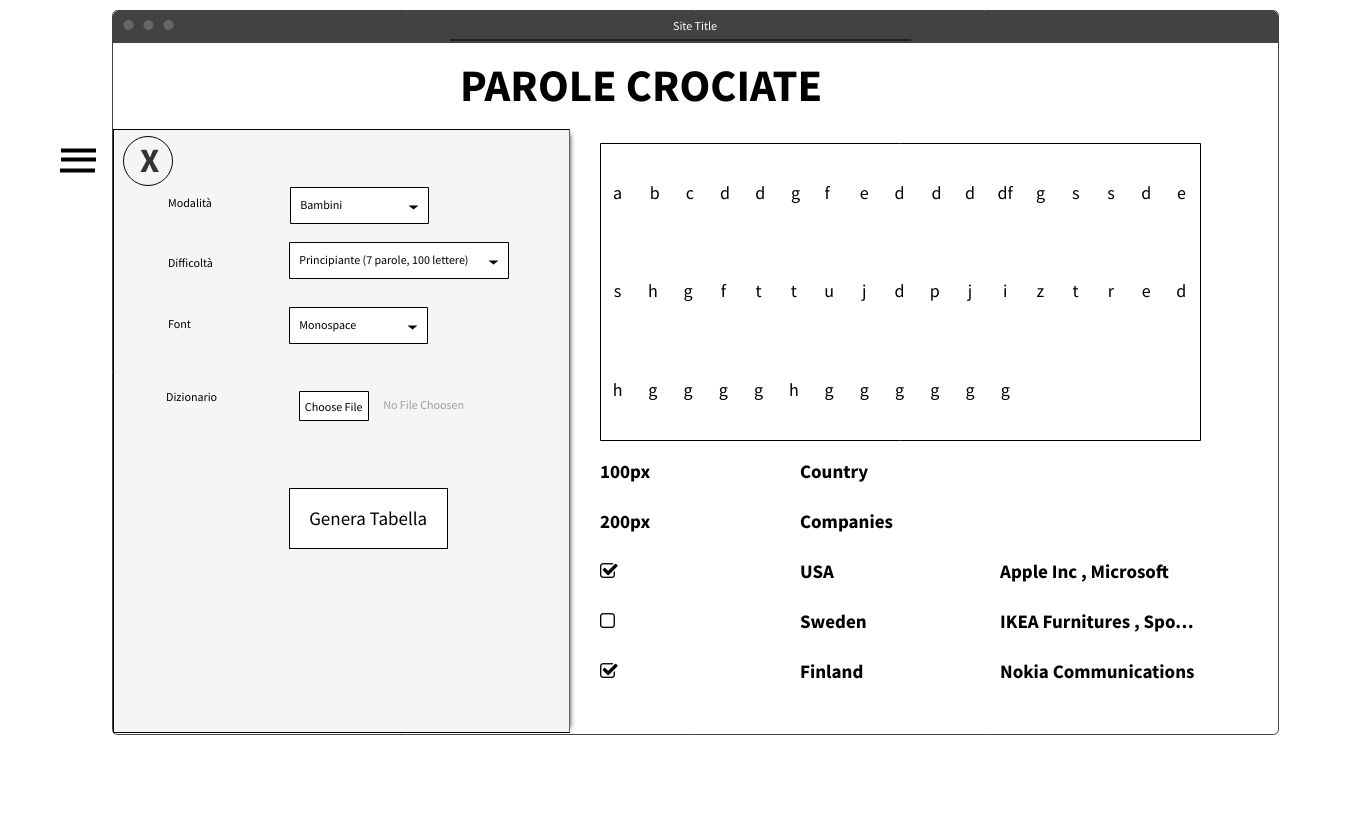
Diagramma Barker/Bachmann



Figura Diagramma ER Barker

## Design delle interfacce

Il design dell’interfaccia è formato da un titolo principale ad esempio “PAROLE CROCIATE”, da un menu a comparsa con le opzioni di creazione della tabella. Mentre la maggior parte della pagina sarà occupata dalla tabella e la lista di parole, tutto ciò centrato al centro.



## Design procedurale

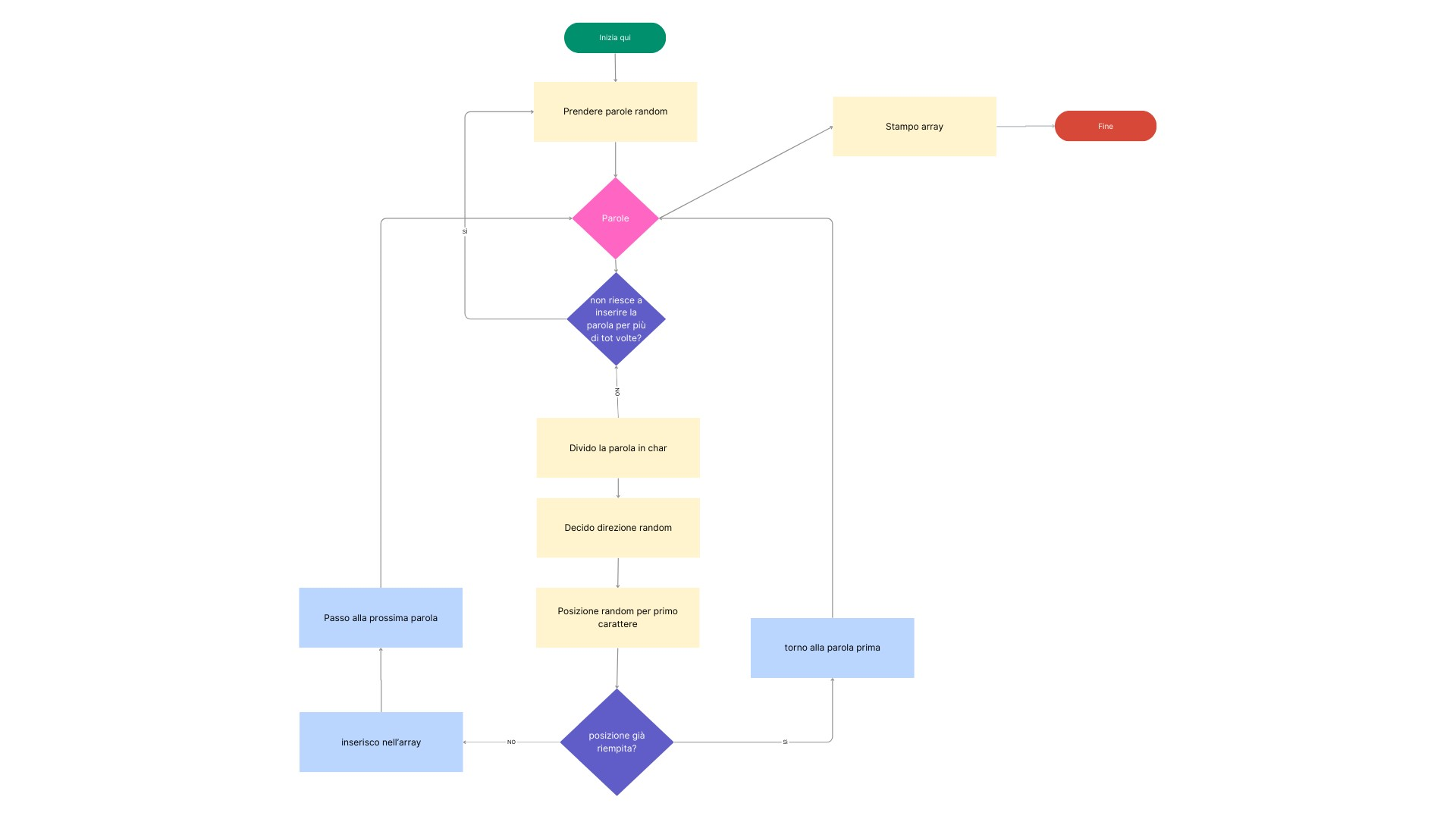


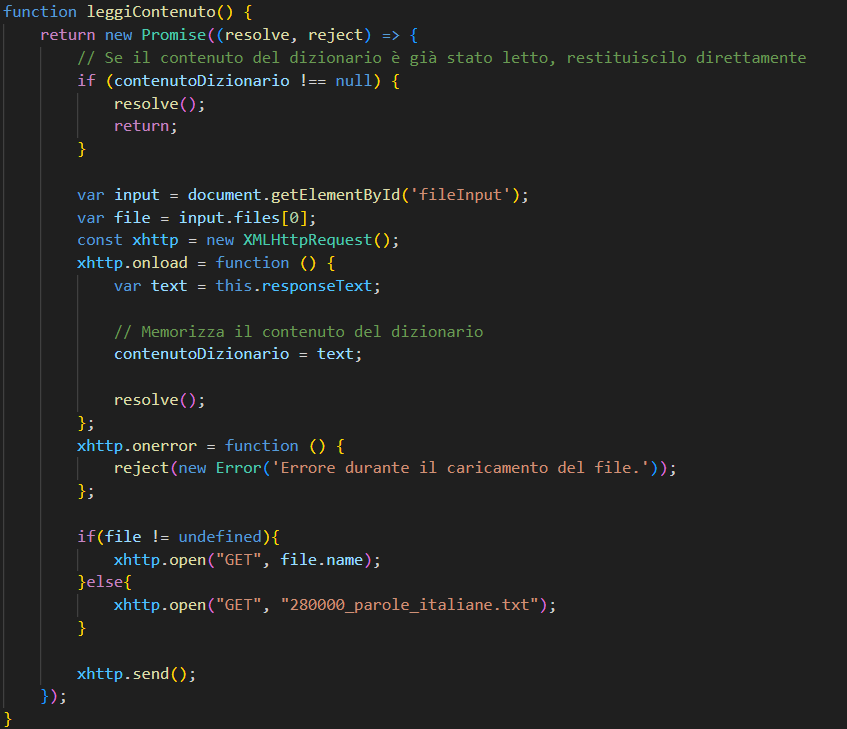
Figura Diagramma di flusso - inserimento parole nella tabella

# Implementazione

## Lettura del file

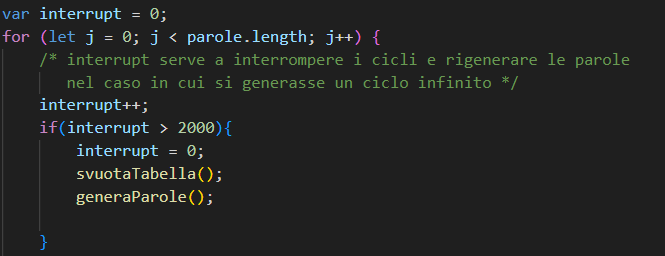
Per la lettura del dizionario ho avuto parecchi problemi non sapendo cos’era e come funzionava il promise.

### LeggiContenuto()

In questo metodo utilizzo un promise, cioè una classe che mi permette di eseguire le operazioni in modo asicrono(la risoluzione del codice non in sequenza), questo mi serve per prendere il file dal file system.

All’interno del promise apro una richiesta XML per poter accedere al file.

Dopodiché prendo il contenuto del file e lo salvo in una variabile globale



In questa porzione di codice ho risolto un grosso problema, che consisteva nel generare ogni tanto un ciclo infinito, dovuto al non riuscire a inserire una parola nella tabella.

Con questo metodo tengo in memoria una variabile dei tentativi che fa una parola per entrare nella tabella, dopo 2000 rigenero le parole e riprovo.

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case**  **Riferimento** | TC-001  REQ-004 | **Nome** | Generazione tabella default |
| **Descrizione** | Si prova a generare la tabella con gli input di default. | | |
| **Prerequisiti** | Tabella completata | | |
| **Procedura** | 1. Cliccare il bottone “Genera tabella” | | |
| **Risultati attesi** | Dovrebbe generarsi la tabella con le parole in base difficoltà, in tutte le direzioni, senza sovrapposizione e con possibili incroci. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case**  **Riferimento** | TC-002  REQ-002 | **Nome** | Modalità adulto |
| **Descrizione** | Si verifica che la modalità adulto funzioni e che quindi crei una parola con le lettere rimanenti. | | |
| **Prerequisiti** | Tabella completata | | |
| **Procedura** | 1. Selezionare la modalità adulti tramite input 2. Cliccare “genera tabella” 3. Trovare tutte le parole e unire le lettere rimanenti | | |
| **Risultati attesi** | Con le lettere rimanenti dovrebbe crearsi una parola. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case**  **Riferimento** | TC-003  REQ-003 | **Nome** | Difficoltà |
| **Descrizione** | Si verifica che le tutte le difficoltà funzionino correttamente, parole e grandezza corrette. | | |
| **Prerequisiti** | Tabella completata | | |
| **Procedura** | 1. Selezionare la difficoltà tramite input 2. Cliccare “genera tabella” 3. Rifare i passaggi per tutte le difficoltà | | |
| **Risultati attesi** | Per ogni difficoltà devono esserci il numero di parole e la grandezza corretta (definita nell’input). | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case**  **Riferimento** | TC-004  REQ-005 | **Nome** | Export tabella |
| **Descrizione** | Si verifica che l’export in formato txt e html funzioni correttamente | | |
| **Prerequisiti** | Tabella completata e export completato | | |
| **Procedura** | 1. Cliccare export txt 2. Aprire file (da verificare) 3. Verificare che contenuto sia corretto 4. Cliccare export html 5. Aprire file (da verificare) 6. Verificare che contenuto sia corretto | | |
| **Risultati attesi** | Nei file ci dovrà essere la tabella riempita con la lista delle parole. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case**  **Riferimento** | TC-005  REQ-00 | **Nome** | Scelta del font |
| **Descrizione** | Si verifica che l’export in formato txt e html funzioni correttamente | | |
| **Prerequisiti** | Tabella completata e export completato | | |
| **Procedura** | 1. Cliccare export txt 2. Aprire file (da verificare) 3. Verificare che contenuto sia corretto 4. Cliccare export html 5. Aprire file (da verificare) 6. Verificare che contenuto sia corretto | | |
| **Risultati attesi** | Nei file ci dovrà essere la tabella riempita con la lista delle parole. | | |

## 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.

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Risultato ottenuto** | **Stato** |
| TC-001 | Testo o immagine a conferma del test | Passato |
| TC-002 | Testo o immagine a conferma del test | Fallito |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## 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 Gantt consuntivo).

Stile orizzontale



CONSUNTIVO

Figura Esempio di diagramma di Gantt 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.

# Glossario

|  |  |
| --- | --- |
| **Termine** | **Significato** |
| ASCII | American Standard Code for Information Interchange, codice per la codifica di caratteri. |
|  |  |
|  |  |
|  |  |
|  |  |
| ZFS | Zettabyte File System, filesystem opernsource a 128 bit con funzionalità avanzate. |

# Indice delle figure

[Figura 1 Esempio di diagramma di Gantt 7](#_Toc124775984)

[Figura 2 Esempio di diagramma di Gantt 8](#_Toc124775985)

[Figura 3 Diagramma ER Chen 10](#_Toc124775986)

[Figura 4 Diagramma ER Barker 11](#_Toc124775987)

[Figura 5 Esempio di diagramma di Gantt consuntivo 16](#_Toc124775988)

# 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
* …