Impiccato

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

2 Analisi 4

2.1 Analisi del dominio 4

2.2 Analisi e specifica dei requisiti 4

2.3 Use case 7

2.4 Pianificazione 7

2.5 Analisi dei mezzi 7

2.5.1 Software 7

2.5.2 Hardware 7

3 Progettazione 8

3.1 Design dell’architettura del sistema 8

3.2 Design dei dati e database 8

3.3 Design delle interfacce 8

3.4 Design procedurale 8

4 Implementazione 9

5 Test 9

5.1 Protocollo di test 9

5.2 Risultati test 10

5.3 Mancanze/limitazioni conosciute 10

6 Consuntivo 10

7 Conclusioni 10

7.1 Sviluppi futuri 10

7.2 Considerazioni personali 10

8 Bibliografia 10

8.1 Bibliografia per articoli di riviste: 10

8.2 Bibliografia per libri 10

8.3 Sitografia 11

9 Allegati 11

# Introduzione

## Informazioni sul progetto

In questo capitolo raccogliere le informazioni relative al progetto, ad esempio:

* Allievo/i e docente/i coinvolti nel progetto e rispettivi ruoli,
* scuola, sezione, materia/e,
* date di inizio e termine di consegna,
* …

## 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 del progetto (scopi didattici/scopi operativi). Dovrebbe descrivere il mandato, ma non vanno ricopiate le informazioni del quaderno dei compiti (che va invece allegato).

# Analisi

## Analisi del dominio

Questo capitolo dovrebbe descrivere il contesto in cui il prodotto verrà utilizzato, da questa analisi dovrebbero scaturire le risposte a quesiti quali ad esempio:

* Background/Situazione iniziale
* Quale è e come è organizzato il contesto in cui il prodotto dovrà funzionare?
* Come viene risolto attualmente il problema? Esiste già un prodotto simile?
* Chi sono gli utenti? Che bisogni hanno? Come e dove lavorano?
* Che competenze/conoscenze/cultura posseggono gli utenti in relazione con il problema?
* Esistono convenzioni/standard applicati nel dominio?
* Che conoscenze teoriche bisogna avere/acquisire per poter operare efficacemente nel dominio?
* …

## Analisi e specifica dei requisiti

Il progettista, dopo aver ricevuto il mandato, in collaborazione con il committente redige una lista di requisiti. Durante questi incontri, tramite interviste (da inserire nei diari), il progettista deve cercare di rispondere alle seguenti domande:

* Quali sono i bisogni del committente?
* Quali funzioni deve svolgere il prodotto?
* Come devono essere implementate?
* L’utente, come vorrebbe/dovrebbe interagire con il prodotto?
* Come verrà utilizzato il prodotto?
* Che tipo di interfaccia si immagina?
* Che prestazioni minime deve fornire il prodotto?
* Che grado di sicurezza deve avere il prodotto?
* …

In base alla lista dei requisiti e all’analisi degli stessi, il progettista redige una *specifica dei requisiti* in cui elenca e descrive in modo dettagliato quali sono le funzionalità che il prodotto fornirà. La specifica dovrebbe essere abbastanza dettagliata da poter essere utilizzata come base per lo sviluppo, ma non troppo; ad esempio non dovrebbe contenere dettagli di implementazione, o definizioni dettagliate dell’interfaccia grafica a meno che questi non siano considerati cruciali. Non si deve scordare che i requisiti non rappresentano delle attività bensì delle caratteristiche che il prodotto dovrà possedere.

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Creazione della partita |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | È presente un tasto per creare una partita e uno per unirsi ad una partita |
| **Sotto requisiti** | |
| **001** | Connessione al server |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Controllo nome utente |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Non deve contenere né volgarità né essere vuoto o contenere solo spazi. Deve essere inoltre diverso dai nomi dei giocatori già presenti nella partita |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Unirsi ad una partita |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Nome utente adeguato |
| **002** | Hosting partite funzionante |
| **003** | Generazione token partita |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Chat broadcast |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Chat in cui tutti posso vedere le lettere che gli altri giocatori scrivono |
| **Sotto requisiti** | |
| **001** | Hosting partite funzionante |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | Lista dei giocatori |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Ritornare una lista dei giocatori, il nome di un giocatore deve essere in verde se ha indovinato la parola corrente |
| **Sotto requisiti** | |
| **001** | Hosting partite funzionante |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Parole a disposizione |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessitano almeno 500 parole della lingua italiana |

|  |  |
| --- | --- |
| **ID: REQ-007** | |
| **Nome** | Gestione delle lettere inviate al server |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Vengono gestite lettere fino al carattere U+00FF |
| **Sotto requisiti** | |
| **001** | Invio delle lettere al server funzionante |

|  |  |
| --- | --- |
| **ID: REQ-008** | |
| **Nome** | Gestione errori |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Un giocatore può sbagliare al massimo 10 volte. Una volta raggiuto questo limite il giocatore dovrà aspettare il turno successivo per continuare a giocare. |
| **Sotto requisiti** | |
| **001** | Hosting partite funzionante |
| **002** | Invio lettere funzionante |
| **003** | Dovrà essere possibile la ricerca di allievi |

|  |  |
| --- | --- |
| **ID: REQ-009** | |
| **Nome** | Schermata finale |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Fa vedere i giocatori con il relativo punteggio |
| **Sotto requisiti** | |
| **001** | Ultimo round terminato |
| **002** | Calcolo dei punteggi |

|  |  |
| --- | --- |
| **ID: REQ-010** | |
| **Nome** | Parametrizzazione partita |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si può scegliere il numero e la durata di ogni round (durata minimo 30 secondi) |
| **Sotto requisiti** | |
| **001** | Creazione di una partita |

|  |  |
| --- | --- |
| **ID: REQ-011** | |
| **Nome** | Calcolo punteggio |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il punteggio viene calcolato in base al tempo rimasto alla fine del round |
| **Sotto requisiti** | |
| **001** | Fine di un round |

## 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 : Esempio di diagramma di Gantt. |

Se si usano altri metodi di pianificazione (p.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-0 | **Nome:** |  |
| **Descrizione:** |  | | |
| **Prerequisiti:** |  | | |
| **Procedura:** |  | | |
| **Risultati attesi:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-001 | **Nome:** | Verifica creazione della partita |
| **Descrizione:** | Verifico che dopo aver schiacciato il tasto per creare una partita la partita venga effettivamente creata sul server | | |
| **Prerequisiti:** |  | | |
| **Procedura:** |  | | |
| **Risultati attesi:** | Nella console del server viene stampato “creating <Token>” | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-002  REQ-002 | **Nome:** | Verifica controllo nome utente |
| **Descrizione:** | Vengono controllati i requisiti del nome utente: no volgarità, no nomi vuoti o solo spazi, no nomi di giocatori già presenti nella partita | | |
| **Prerequisiti:** |  | | |
| **Procedura:** |  | | |
| **Risultati attesi:** | L’applicativo stampa un errore che mi dice che il nome utente non può essere utilizzato per un motivo specifico | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-003  REQ-003 | **Nome:** | Verifica connessione ad una partita |
| **Descrizione:** | Verifico che posso unirmi ad una partita | | |
| **Prerequisiti:** |  | | |
| **Procedura:** |  | | |
| **Risultati attesi:** | Viene mostrata la schermata di attesa in cui vedo la lista dei partecipanti correnti. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-004  REQ-004 | **Nome:** | Verifica funzionalità chat |
| **Descrizione:** | Verifico che i client ricevano in boradcast le lettere inviate dai client presenti nella stessa partita | | |
| **Prerequisiti:** |  | | |
| **Procedura:** |  | | |
| **Risultati attesi:** | Nell’area della chat appaiono le lettere scritte da altri giocatori | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-005  REQ-005 | **Nome:** | Verifica stampa lista dei giocatori |
| **Descrizione:** | Durante la partita e mentre si sta aspettando che la partita inizi, la lista dei giocatori con il loro punteggio deve essere visibile e aggiornata | | |
| **Prerequisiti:** |  | | |
| **Procedura:** |  | | |
| **Risultati attesi:** | La lista di giocatori viene visualizzata senza problemi | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-006  REQ-006 | **Nome:** | Verifica parole a disposizione |
| **Descrizione:** | Verifico che ci siano ameno 500 parole a disposizione | | |
| **Prerequisiti:** |  | | |
| **Procedura:** |  | | |
| **Risultati attesi:** | Vengono stampate le parole a che si possono usare | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-007  REQ-007 | **Nome:** | Verifica invio dati al server |
| **Descrizione:** | Verifica del ricevimeto di vari dati (come ad esempio lettere) da client a server | | |
| **Prerequisiti:** |  | | |
| **Procedura:** |  | | |
| **Risultati attesi:** | Nella console del server vengono stampati i dati inviati dal client | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-008  REQ-008 | **Nome:** | Verifica gestione errori |
| **Descrizione:** | Verifico che quando un utente sbaglia una lettera il conteggio degli errori sale di uno. Se gli errori dovessero arrivare al massimo, il giocatore perde. | | |
| **Prerequisiti:** |  | | |
| **Procedura:** |  | | |
| **Risultati attesi:** | Contatore errore sale di uno per ogni errore e a 10 errori la partita viene terminata | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-009  REQ-009 | **Nome:** | Verifica dati schermata finale |
| **Descrizione:** | Quando la partita è finita mostra una schermata con i nomi dei giocatori e il relativo punteggio. | | |
| **Prerequisiti:** |  | | |
| **Procedura:** |  | | |
| **Risultati attesi:** | Viene mostrata una schermata con giocatori e punteggio | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-010  REQ-010 | **Nome:** | Verifica parametri partita |
| **Descrizione:** | Durante la creazione della partita è possibile parametrizzare alcuni aspetti, come la durata del match | | |
| **Prerequisiti:** |  | | |
| **Procedura:** |  | | |
| **Risultati attesi:** | Quando si crea una partita viene mostrata una schermata di parametrizzazione, e durante la partita i cambiamenti vengono applicati | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-011  REQ-011 | **Nome:** | Verifica del calcolo del punteggio |
| **Descrizione:** | Quando il giocatore indovina la parola, viene assegnato un punteggio in base al tempo rimasto alla fine della partita | | |
| **Prerequisiti:** |  | | |
| **Procedura:** |  | | |
| **Risultati attesi:** | Il punteggio del giocatore viene calcolato e mostrato correttamente nella lista dei giocatori | | |

## 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 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.

# 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
* …