DattiloKing

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

2.2.1 Spiegazione elementi tabella dei requisiti: 5

2.3 Use case 5

2.4 Pianificazione 5

2.5 Analisi dei mezzi 5

2.5.1 Software 6

2.5.2 Hardware 6

3 Progettazione 6

3.1 Design dell’architettura del sistema 6

3.2 Design dei dati e database 6

3.3 Design delle interfacce 6

3.4 Design procedurale 6

4 Implementazione 7

5 Test 7

5.1 Protocollo di test 7

5.2 Risultati test 8

5.3 Mancanze/limitazioni conosciute 8

6 Consuntivo 8

7 Conclusioni 8

7.1 Sviluppi futuri 8

7.2 Considerazioni personali 8

8 Glossario 8

9 Bibliografia 9

9.1 Bibliografia per articoli di riviste: 9

9.2 Bibliografia per libri 9

9.3 Sitografia 9

10 Allegati 9

# Introduzione

## Informazioni sul progetto

* Allievi:
  + Amos Haefliger
  + Leonardo Sciara
  + Nemanja Zecevic
  + Robin Sartore
* Docente: Michel Palucci
* Scuola: Scuola Arti e Mestieri di Trevano
* Sezione: Informatica
* Materia: M306
* Data di inizio: 29.01.2025
* Data di consegna: 28.05.2025

## 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 nostro sito web “DattiloKing” è quello di migliorare la propria scrittura da tastiera in modo didattico e simpatico. Si avrà la possibilità di giocare in single player per migliorare la propria abilità in modo autonomo e di selezionare la modalità multiplayer per poter sfidare i propri amici nello scrivere la frase nel modo più veloce, avendo poi la possibilità di vedere il proprio score e quello dei nostri amici.

Questo sito è adatto a chiunque si interfacci con un Pc, indipendentemente dall’età.

Questo progetto permette al nostro team di migliorare le proprie skill con i linguaggi che utilizzeremo per costruire il sito web nonché HTML, BOOTSTRAP, JS, PHP, MYSQL.

Grazie al progetto miglioreremo anche le nostre abilità di lavorare in gruppo, di sperimentare e approfondire la metodologia di progetto Agile.

Con questo progetto puntiamo a rendere divertente e stimolante l’apprendimento e il miglioramento a scrivere con la tastiera. Grazie alla possibilità di sfidare i propri amici puntiamo a una voglia di superarsi e di superare i nostri amici che migliorerà notevolmente la nostra velocità nella scrittura da tastiera.

# Analisi

## Analisi del dominio

Il nostro sito web sarà utilizzabile sui più famosi sistemi operativi quali Windows, MacOs e tutte le distro di Linux da un qualsiasi browser. Può venir utilizzato da chiunque, e da qualsiasi fascia di età (a patto che sappia utilizzare la tastiera). Chiunque abbia il bisogno di migliorare le proprie skill dattili può utilizzare DattiloKing per migliorare e per sfidare i propri amici migliorando assieme. Sul web ci sono numerosi siti web che permettono di migliorare la propria dattilografia quali “dattilografia-online” o “agilefingers” ma con DattiloKing abbiamo implementato un’interfaccia innovativa rendendo unico il nostro sito web.

Naturalmente non sono necessarie skill, conoscenze o competenze particolari per utilizzare il sito web, un completo principiante che utilizza per la prima volta il suo dispositivo desktop o laptop che sia può utilizzare DattiloKing per migliorare le proprie skill e imparare a digitare da tastiera in modo più rapido.

## 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 DB |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessitano i permessi di root |
| **Sotto requisiti** | |
| **001** | Si necessita del diagramma ER |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Creazione interfaccia Login |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Creazione interfacce di gioco |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Collegamento pagine signIn/signUp con DB |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Collegamento con DB necessario per poter sapere con che account si effettua il login o in caso ci si registri di salvare le informazioni dell’utente nel DB |
| **Sotto requisiti** | |
| **001** | Si necessita del DB |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | Collegamento pagine di gioco con DB |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Si necessita del DB |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Score utente singleplayer |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Si necessita dell’interfaccia di gioco |
| **002** | Si necessità del DB |

|  |  |
| --- | --- |
| **ID: REQ-007** | |
| **Nome** | Score utenti multiplayer |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Si necessita dell’interfaccia di gioco |
| **002** | Si necessità del DB |

|  |  |
| --- | --- |
| **ID: REQ-008** | |
| **Nome** | Percentuale di errori dell’utente sul singolo try |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Si necessita dell’interfaccia di gioco |
| **002** | Si necessità del DB |

|  |  |
| --- | --- |
| **ID: REQ-009** | |
| **Nome** | Percentuale di errori dell’utente su tutte le try |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Si necessita dell’interfaccia di gioco |
| **002** | Si necessità del DB |

|  |  |
| --- | --- |
| **ID: REQ-010** | |
| **Nome** | Velocità di scrittura (parole al minuto) |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Si necessita dell’interfaccia di gioco |
| **002** | Si necessità del DB |

|  |  |
| --- | --- |
| **ID: REQ-011** | |
| **Nome** | Tempo di scrittura giornaliero |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Si necessita dell’interfaccia di gioco |
| **002** | Si necessità del DB |

|  |  |
| --- | --- |
| **ID: REQ-012** | |
| **Nome** | Tempo di scrittura dalla creazione dell’utente |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Si necessita dell’interfaccia di gioco |
| **002** | Si necessità del DB |

### 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



Questo è lo Use Case del progetto DattiloKing. L’utente può gestire diversi parametri e registrarsi per poter visualizzare le proprie statistiche in modo accurato e accedere al multiplayer.

## 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: Esempio di diagramma di Gantt. |

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

## Analisi dei mezzi

### Software

Il software include:

* GitHub Desktop: strumento per la gestione di repository Git
* PhpStorm 2024.1.1: IDE per lo sviluppo
* MySQL Workbench 8.0 CE: Strumento per la gestione di database MySQL
* Google Chorme: Browser utilizzato per la navigazione
* FireFox: Browser utilizzato per la navigazione
* Microsoft Edge: Browser utilizzato per la navigazione
* Word: Software per l’elaborazione della documentazione
* Project: Software per l’elaborazione dei Gantt
* Visio: Software per l’elaborazione degli allegati
* Blocco note: Editor per appunti
* Strumento di cattura: Software per la cattura delle schermate

### Hardware

L’hardware include:

* PC di scuola per la realizzazione:
  + CPU: 13th Gen Intel(R) Core(TM) i7-13700
  + GPU: Intel(R) UHD Graphics 770
  + RAM: 32GB
  + SSD: 512GB
* PC di casa sempre per la realizzazione, per poter continuare in caso di ore saltate a lezione

# 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

Il nostro sistema delle architetture non è molto grande dato che abbiamo solo un DB e il WebServer con l’applicativo, però abbiamo fatto uno schema per spiegarlo.



L’utente una volta digitato l’url del nostro applicativo riceverà le view dal WebServer che si situa nel server Infomaniak. Poi l’utente può registrarsi o accedere grazie al database con tutti gli utenti registratati.

## Design dei dati e database

## Ho creato il database di nome dattiloking con cinque tabelle al suo interno, una tabella “stanza” la quale ha i campi codice, il quale rappresenta un codice univoco il quale serve ad accedere alla stanza del multiplayer, “vincitore” il quale ha salvato il nickname del vincitore della stanza attuale e “numeroTurni” il quale indica il numero di turni che la stanza ha avuto.

La tabella “frase” contiene le frasi che devono apparire quando ci si esercita e quando si gioca alla modalità multiplayer, la lingua della frase e la lunghezza.

Ci sono inoltre due tabelle: “turno\_storico” e “turno\_odierno” le quali contengono i dati di tutti i turni dei giocatori, per quanto riguarda la giornata stessa e lo storico totale. Infine è presente una tabella “utente” la quale salva lo username il quale deve essere univoco e la password degli utenti una volta che si registrano.

## 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 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.

# Glossario

Inserite una semplice tabella con due colonne che spieghi i termini specifici del progetto (lista dei termini in ordine alfabetico A-Z)

Esempio:

|  |  |
| --- | --- |
| **Termine** | **Descrizione** |
| AJAX | **Asynchronous JavaScript And XML**: una tecnica che permette di eseguire richieste ed ottenere dati da una pagina web in modo asincrono. |
| CSS | **Cascading Style Sheets**: linguaggio che permette di definire il layout e la grafica di una pagina web. |

# 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
* …