Esempio di documentazione

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

Analisi 4

1.4 Analisi del dominio 4

1.5 Analisi e specifica dei requisiti 4

1.6 Use case 6

1.7 Pianificazione 6

1.8 Analisi dei mezzi 6

1.8.1 Software 6

1.8.2 Hardware 6

2 Progettazione 7

2.1 Design dell’architettura del sistema 7

2.2 Design dei dati e database 7

2.3 Design delle interfacce 7

2.4 Design procedurale 7

3 Implementazione 8

4 Test 8

4.1 Protocollo di test 8

4.2 Risultati test 9

4.3 Mancanze/limitazioni conosciute 9

5 Consuntivo 9

6 Conclusioni 9

6.1 Sviluppi futuri 9

6.2 Considerazioni personali 9

7 Bibliografia 9

7.1 Bibliografia per articoli di riviste: 9

7.2 Bibliografia per libri 9

7.3 Sitografia 9

8 Allegati 10

# Introduzione

## Informazioni sul progetto

Titolo: Sito web con pagina di registrazione e verifica dati

Alunno: Mattia Toscanelli

Classe: I3AC

Docenti: Luca Muggiasca, Francesco Mussi, Adriano Barchi,

Scuola: Scuola Arti e Mestrieri Trevano (Informatica)

Data inizio: 07.09.2018

Termine progetto: 09.11.2018

## Abstract

E’ 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 sviluppo di questa applicazione serve per avere un metodo di registrazione in formato digitale. Questo permette a tutte le persone di potersi registrare autonomamente e di avere tutti i dati ordinati in due file distinti csv. Il progetto può venire implementato ovunque ci sia bisogno di una registrazione autonoma e permette agli utenti di non compilare formulari cartecei.

## Analisi

## Analisi del dominio

Bisogna trovare un metodo su come gestire e organizzare i dati delle persone interessate al contesto, l’obbiettivo è quello di creare un form in grado di poter gestire e salvare i dati in modo digitale senza la presenza di fogli cartacei. Esistono già diversi prodotti simili in grado di fare ciò, ma nessuno è adatto ai requisiti del cliente, per assenza di campi oppure per difficoltà di comprensione. L’inserimento dei dati deve essere facile, comprendibile, comodo e veloce per favorire l’utilizzo da parte dell’utente e dal gestore.

## Analisi e specifica dei requisiti

Grazie ad un sito web il committente potrà effetturare una registrazione. La pagina iniziale presentera un messaggio di benvenuto e un pulsante per la registrazione. La registrazione avverrà attraverso una verifica dei vari campi da compilare. Una volta verificati tutti i dati, verrà mostrato a schermo una seconda pagina dove l’utente potrà verificare che i suoi dati siano corretti. A questo punto potrà scegliere se confermare la registrazione o ri-modificare i suoi dati. Se l’utente decide di ri-modificare i suoi dati verrà reindirizzato alla pagina di registrazione, invece se conferma verra trasportato su una pagina finale la quale mostrerà tutta la registrazione effettuata. L’intrefaccia del prodotto deve essere semplice e intuitiva per ogni tipo di utente finale. Non si deve avere un consumo elevato di risorse e infine non è richiesto nessun grado di sicurezza.

|  |  |
| --- | --- |
| **ID: REQ-01** | |
| **Nome** | Form di registrazione |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **rNote** | La pagina dove l’utente inserisce i propri dati da convalidare per la registrazione. |
| **Sotto requisiti** | |
| **001** | Ogni dato deve avere un prorio campo. |
| **002** | I campi obbligatori sono contrassegnati co un \*. |
| **003** | Il campo e-mail deve avere uns struttura testo@testo.testo. |
| **004** | Il campo hobby non deve superare i 500 caratteri |
| **005** | Il campo genere deve essere gestito tramite un imput a scelta singola. |
| **006** | L’estetica è a scelta del programmatore. |
| **007** | La convalidizzazione dei campi deve avvenire prima della registrarione. (Server o Client) |
| **008** | Al momento della registrazione verrà aggiunto un campo con la data e l’ora corrente della registarzione. |
| **009** | La registarzione non pemette la compilazione se i campi obbligatori non sono compilati. |
| **010** | L’utente deve avere la possibilità di azzerare tutti i campi attraverso il tasto “Cancella” , inoltre deve avere la possibilità di poter continuare con il tasto “Avanti”. |

|  |  |
| --- | --- |
| **ID: REQ-02** | |
| **Nome** | Form di controllo |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | La pagina dove vengono mostrati tutti i dati della registrazione per poterli verificare prima di inserirli nel file csv. Dipende dal REQ-01 (Form di registarzione). |
| **Sotto requisiti** | |
| **001** | I dati inseriti nella registrazione vengono visualizzati in questa pagina ma non modificabili. |
| **002** | Ci deve essere il tasto “Correggi” che riporta al form precedente di registrazione dove si può ancora modificare i dati. |
| **003** | Ci deve essere il tasto “Registra” che conferma i dati di registrazione e li scrive nel file csv. |
| **004** | La prima registrazione deve creare il file csv chiamato “Registrazioni\_tutte.csv” e deve avere come separatore fra i campi il ;. |
| **005** | La prima registrazione giornaliera deve creare un file csv chiamato “Registrazione\_aaaa\_mm\_gg.csv”. |
| **006** | L’estetica è a scelta del programmatore. |

|  |  |
| --- | --- |
| **ID: REQ-03** | |
| **Nome** | Form di benvenuto |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | La pagina principale dove l’utente che vuole effettuare la registrazione si collega. |
| **Sotto requisiti** | |
| **001** | Si deve mostare a schermo un form contenente un pulsante “Registarti” che si collegherà alla pagina di registrazione. |
| **002** | L’estetica è a scelta del programmatore. |

|  |  |
| --- | --- |
| **ID: REQ-04** | |
| **Nome** | Form finale di registrazione |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | La pagina finale dove vengono mostrate le registrazioni giornaliere. |
| **Sotto requisiti** | |
| **001** | Si deve mostare a schermo in modo formattato le registrazioni lette dal file csv  “Registraione\_aaaa-mm-gg.csv”. |
| **002** | L’estetica è ascelta del programmatore. |

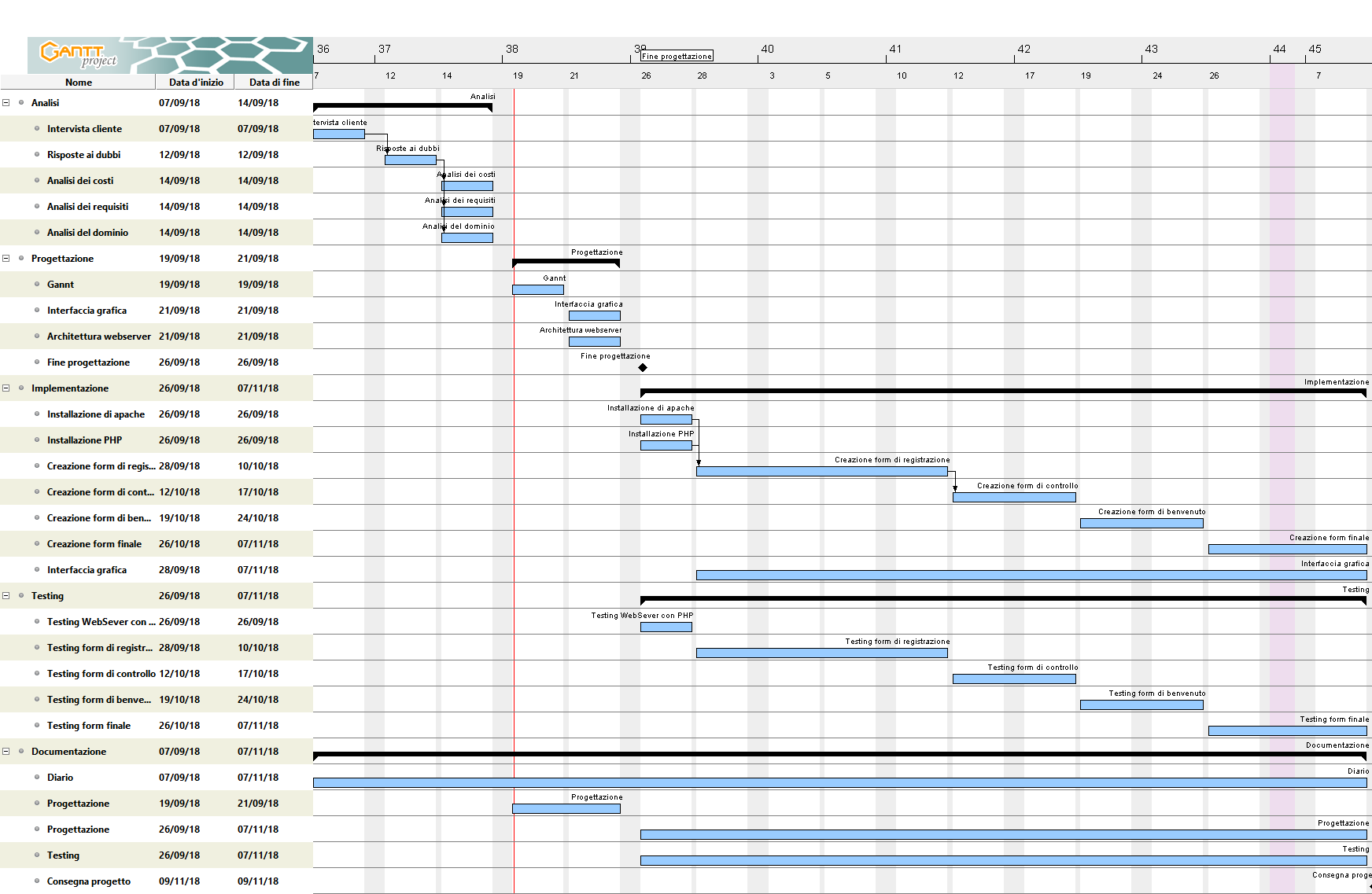
|  |  |
| --- | --- |
| **ID: REQ-05** | |
| **Nome** | Consegna del sito di registrazione |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | - |

## Analsi dei costi

|  |  |  |  |
| --- | --- | --- | --- |
| **Categoria** | **Costo all’ora** | **Ore lavorate** | **Costo totale** |
| Manodopera | 62 CHF |  |  |

## Pianificazione

Ho inserito due pietre miliari, una a fine progettazione, cioè quando il progetto inizierà ad essere concreto; l’altra a fine progetto, quando dovrà essere consegnato sul sito di infomaniak. La fase che mi richiederà piu tempo è l’implementazione, più precisamte l’attività “Creazione form di benvenuto”, in quanto necessita di molti piu controlli e adequatezza rispetto agli altri form.



|  |
| --- |
| Figura 1: Diagramma di Gantt. |

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
* …