Gestione Apiario  
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: Aris Previtali

Classe: informatica 3AC Scuola Arti e Mestieri Trevano

Docente responsabile: Geo Petrini

Data inizio:

Data consegna: 2020-12-17

## Abstract //da fare

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 scopo del progetto Gestione Apiario è quello di creare un sito nella quale è possibile creare un account per qualsiasi persona (apicoltore) che voglia gestire le proprie arnia, tra le varie opzioni troviamo la possibilità di aggiungere un’arnia, modificarla o eliminarla. Inoltre è possibile aggiungere trattamenti con descrizione, data d’inizio e data di fine, note sia sulle arnie sia nei giorni del calendario, il quale sarà visibile nella home page del sito, sempre all’interno del calendario poter vedere i dati meteo del luogo di un’arnia della giornata e di quelle precedenti. Il sito è strutturato in modo da poter avere più utenti con accesso a tutte le possibilità che offre il sito.

# Analisi

## Analisi del dominio

Questo progetto serve per gestire un apiario. Potrà e verrà utilizzato da qualunque apicoltore voglia utilizzarlo. Questo progetto è web-based, ovvero verrà creata una pagina web dalla quale si può gestire il proprio apiario. Per questo motivo è infatti necessario, per me, lavorare con un Web Server con Apache, PHP e MariaDB (SQL) installati e andare a configurarli.

Il sito dev’essere appunto raggiungibile da qualsiasi browser.

## Analisi e specifica dei requisiti

Quella che segue è la specifica dei requisiti:

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Creare sito web |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Poter mettere e dover gestire numero di arnie |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Poter mettere e dover gestire luogo di arnie |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
|  |  |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Poter annotare ogni arnia |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | Data trattamenti sanitari e quanti giorni di durata |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Durata di una determinata attività e notifica ad attività finita |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-007** | |
| **Nome** | Attività visibili in calendario, sia passate che future |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-008** | |
| **Nome** | Poter aggiungere anno di nascita dell’ape regina |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** |  | **ID: REQ-007** |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID: REQ-009** | |  |  |
| **Nome** | Multiutente (login) |  | Poter aggiungere anno di nascita dell’ape regina |
| **Priorità** | 1 |  | 2 |
| **Versione** | 1.0 |  | 1.0 |
| **Note** |  |  |  |

|  |  |
| --- | --- |
| **ID: REQ-010** | |
| **Nome** | Informazioni meteo |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-011** | |
| **Nome** | Formato data: YYYY-MM-DD |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** | Per tutta la lunghezza del progetto il formato della data utilizzato sarà YYYY-MM-DD |

## Use case



Figura 1 - Schema use case

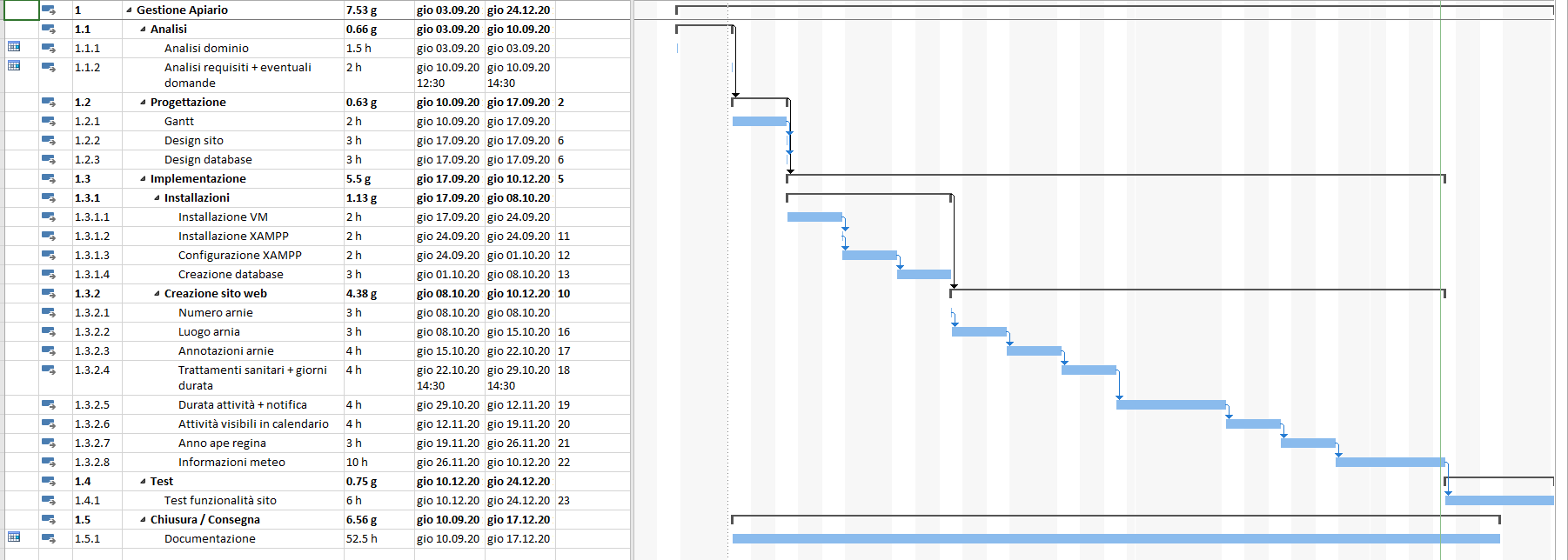
L’utente visitatore del sito potrà accedere al sito loggandosi o, se dovesse essere la prima volta, registrandosi. Essi ovviamente includono il controllo della password, dell’email e del nome utente in caso di login. Per la gestione del calendario da parte dell’utente ci dovrà essere un formato per la data: per l’intero progetto ho scelto che questo formato debba essere YYYY-MM-DD.

L’utente potrà inoltre aggiungere delle note sulle arnie e nel calendario.

Il sistema dovrà notificare all’utente, con appunto delle mail, la fine dei trattamenti aggiunti precedentemente dall’utente.

## Pianificazione

Figura 2 - Diagramma di Gantt



## Analisi:

Ho suddiviso lo svolgimento in quattro grandi e principali attività, quest’attività consiste nel capire cosa mi venisse chiesto di realizzare e porre, se fosse il caso, alcune domande che mi venissero in mente.

Inoltre, in questa fase ho realizzato la tabella dei requisiti.

Oltre a capire in cosa consisteva il progetto ho guardato e mi sono documentato sui migliori strumenti da utilizzare di cui avessi bisogno.

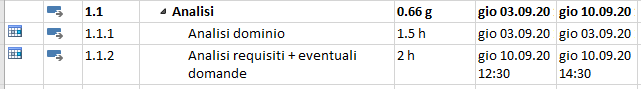


Figura 3 - Analisi, Gantt

## Progettazione

Per la progettazione ho ideato il diagramma di Gantt sopra descritto inoltre ho progettato i due design che mi servivano per la realizzazione del sito, ovvero il design del sito stesso con la struttura principale delle pagine e il design del database il quale serve per salvare le arnie e tutto ciò che concerne un utente.

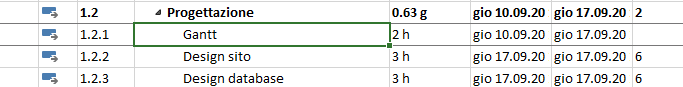


Figura 4 - Progettazione, Gantt

## Implementazione

Nell’implementazione ho aggiunto tutte quelle attività che avevo bisogno di fare per creare il sito web, nell’implementazione c’è, inoltre, le installazioni, macchina virtuale, XAMPP, realizzazione database, e la preparazione della macchina sulla quale ho lavorato.

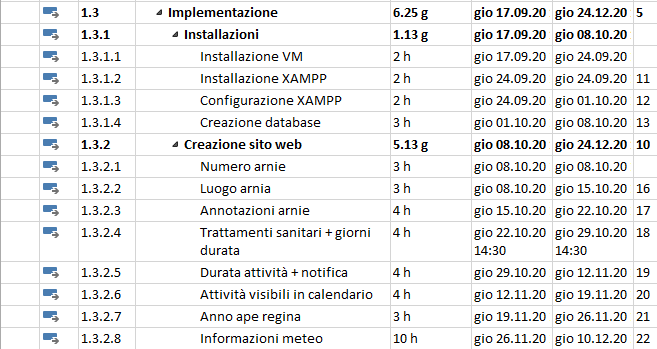


Figura 5 - Implementazione, Gantt

## Test

Nell’ultima principale attività prima della chiusura ho aggiunto i test da fare al sito.



Figura 6 - Test, Gantt

## Documentazione

Per la documentazione ho calcolato di scriverla per tutta la durata del progetto poco alla volta quando avevo dello spazio libero.

## 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

I software che ho utilizzato sono:

* Mozilla FireFox
* PHP
* Apache
* Mysql
* Workbench
* Microsoft VS code
* VMWare Workstation 14 Pro, version: 14.1.2
* Microsoft Project 2016
* Microsoft Word 2016
* Draw.io (<https://draw.io>)
* Mockflow (<https://www.mockflow.com>)

### Hardware

Per l’hardware ho utilizzato una macchina virtuale avente come sistema operativo Windows 10.

# 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
* …