**Gestione Apiario**

Sommario

[1 Introduzione 3](#_Toc50629725)

[1.1 Informazioni sul progetto 3](#_Toc50629726)

[1.2 Abstract 3](#_Toc50629727)

[1.3 Scopo 3](#_Toc50629728)

[2 Analisi 4](#_Toc50629729)

[2.1 Analisi del dominio 4](#_Toc50629730)

[2.2 Analisi e specifica dei requisiti 4](#_Toc50629731)

[2.3 Use case 7](#_Toc50629732)

[2.4 Pianificazione 7](#_Toc50629733)

[2.5 Analisi dei mezzi 8](#_Toc50629734)

[2.5.1 Software 8](#_Toc50629735)

[2.5.2 Hardware 8](#_Toc50629736)

[3 Progettazione 8](#_Toc50629737)

[3.1 Design dell’architettura del sistema 8](#_Toc50629738)

[3.2 Design dei dati e database 9](#_Toc50629739)

[3.3 Design delle interfacce 9](#_Toc50629740)

[3.4 Design procedurale 9](#_Toc50629741)

[4 Implementazione 10](#_Toc50629742)

[5 Test 10](#_Toc50629743)

[5.1 Protocollo di test 10](#_Toc50629744)

[5.2 Risultati test 10](#_Toc50629745)

[5.3 Mancanze/limitazioni conosciute 11](#_Toc50629746)

[6 Consuntivo 11](#_Toc50629747)

[7 Conclusioni 11](#_Toc50629748)

[7.1 Sviluppi futuri 11](#_Toc50629749)

[7.2 Considerazioni personali 11](#_Toc50629750)

[8 Bibliografia 11](#_Toc50629751)

[8.1 Bibliografia per articoli di riviste: 11](#_Toc50629752)

[8.2 Bibliografia per libri 11](#_Toc50629753)

[8.3 Sitografia 11](#_Toc50629754)

[9 Allegati 11](#_Toc50629755)

# Introduzione

## Informazioni sul progetto

Titolo Progetto: Gestione Apiario

Docente responsabile: Luca Muggiasca, Geo Petrini

Allievo responsabili: Alessandro Aloise

CPT Trevano sezione Informatica

Classe: I3AC

Materia: Modulo 306

Inizio progetto 3 settembre 2020

Fine progetto 20 dicembre 2020

## Abstract

Al giorno d’oggi è impensabile che un apicoltore si debba ancora marcare lo stato delle sue arnie o i suoi trattamenti effettuati tutto su un foglio di carta che poi deve immettere in una cartella per tenere memoria di tutto questo. Questo progetto permette di avere una gestione delle proprie arnie in modo moderno. Grazie al comodo calendario e alla possibilità di vedere il meteo della propria arnia in qualsiasi momento. Con questo semplice progetto non è più necessario utilizzare carta e penna per la gestione delle proprie arnie, grazie a Gestione Apiario le tue api sono a portata di mano.

*“Nowadays it is unthinkable that a beekeeper still has to mark the state of his hives or his treatments carried out all on a sheet of paper which he then has to put in a folder to keep track of all this. This project allows you to manage your hives in a modern way. Thanks to the convenient calendar and the ability to see the weather of your hive at any time. With this simple project it is no longer necessary to use pen and paper to manage your hives, thanks to Gestione Apiario your bees are at your fingertips.”*

## Scopo

Lo scopo di questo progetto didattico è quello di creare un sistema di gestione delle arnie dislocate in vari punti geografici. La gestione viene tramite applicativo Java. Si ha la possibilità di avere vari utenti e ognuno gestisce le proprie arnie, l’applicativo da anche la possibilità di gestire il meteo e varie altre funzioni.

# Analisi

### Analisi e specifica dei requisiti

Questa è la lista di requisiti discussi in classe con il professore:

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Possibilità di inserire il numero di arnie |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Registrare delle annotazioni in stile diario per ogni arnia |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Trattamenti sanitari per ogni arnia |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Data o giorni |
| **Sotto requisiti** | |
| **001** | Deve essere indicato tramite data di inizio e di finite |
| **002** | Può essere indicato tramite data di inizio e quantità di giorni che deve durare e il sistema fa il calcolo fino a quando dura |
| **003** | Notifica di fine attività |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Tutte le attività registrate devono poter essere visibili in stile calendario. |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Formattazione stile calendario di Google |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | Data di nascita di ogni ape regina |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Meteo visibile nel calendario |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Da definite |

|  |  |
| --- | --- |
| **ID: REQ-007** | |
| **Nome** | Formazazione data 01.12.2020 (Data usata in tutto l’applicazione ) |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-008** | |
| **Nome** | Esportazio del calendario con un riassunto |
| **Priorità** | 4 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-009** | |
| **Nome** | Esportazioni del diario con un riassunto |
| **Priorità** | 4 |
| **Versione** | 1.0 |
| **Note** | bonus |

|  |  |
| --- | --- |
| **ID: REQ-010** | |
| **Nome** | Esportazioni del Calendario con un riassunto |
| **Priorità** | 4 |
| **Versione** | 1.0 |
| **Note** | Bonus |
|  |  |

|  |  |
| --- | --- |
| **ID: REQ-011** | |
| **Nome** | Multi utente |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** |  |

## 

|  |  |
| --- | --- |
| **ID: REQ-012** | |
| **Nome** | Online e offline |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** | (Bonus) possibilità di gestire la sincronizzazione online offline |

## 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:

|  |
| --- |
| gantt  Figura 1: Esempio di diagramma di Gantt. |

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

## Analisi dei mezzi

### Software

Per realizzare questo progetto sono stati utilizzati i seguenti software:

* ProjectLibre - 1.9.1: Per la creazione del Gantt.
* Java – 14.0.2 Per la creazione dell'applicativo principale
* SQLite studio 3.2.1: Per creare e contenere il database
* Netbeans 12.0: Per tutto ciò che riguarda la scrittura del codice Java
* LGoodDatePicker-11.1.0 Libreria per la creazione del calendario
* Sqlite-jdbc-3.7.2: Libreria per implementare sqlite su Java

L'applicativo è multipiattaforma (Windows, MacOS, Linux). Il progetto non necessa forze computazionali esagerate, necessita però di ava installato sul pc. Senza Java l'applicativo non funzionerà.

### Hardware

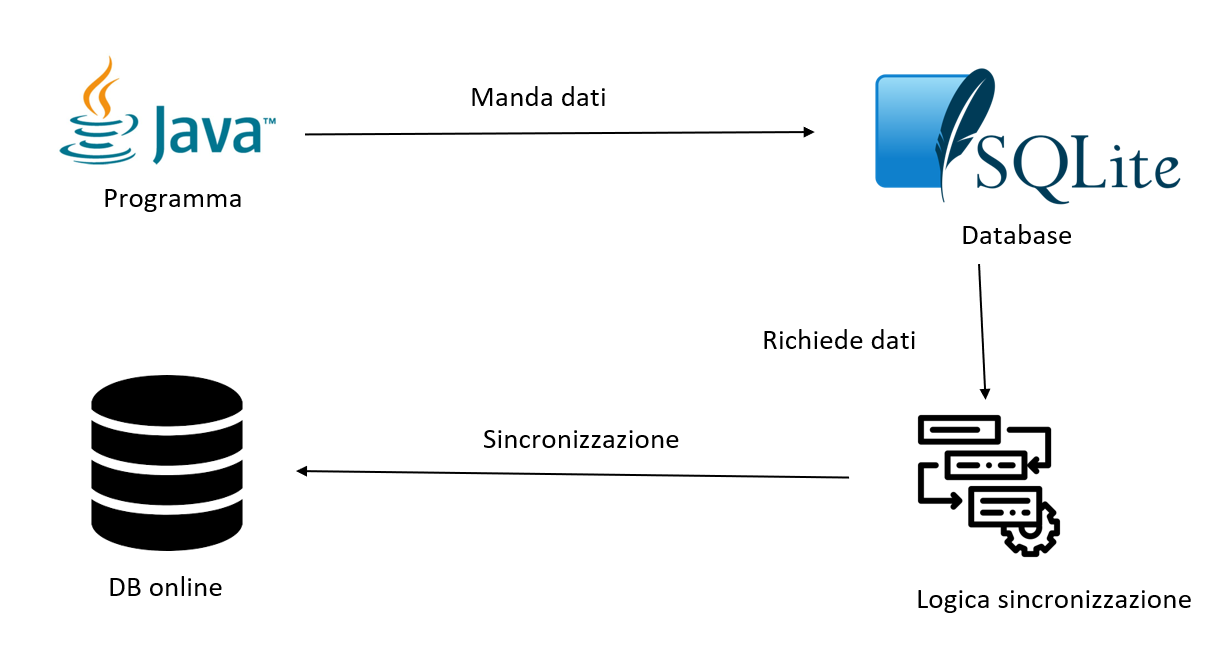
Per questo progetto sono stati usati i seguenti software:

* Pc fisso di scuola
* Del XPS 15 9500 (2020)

# 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



L'immagine mostra lo schema della struttura del progetto. Il progetto è composto da un programma che riceve i dati dal utente. Ad ogni modifica inserisce un dato all'interno del database (all'inizio lo crea se non esiste). La logica per sincronizzare i due database è che ogni volta che mi collego tramite applicativo in locale il programma scarica la versione più recente del database da database online, invece ogni volta che faccio il logout il database appena modificato viene caricato online. In caso che non ci sia una connessione a internet il programma usa l’ultima versione che trova del database locale.

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
* …