ZIQUIZ

# Indice

1 Indice 2

2 Introduzione 3

2.1 Informazioni sul progetto 3

2.2 Abstract 3

2.3 Scopo 4

3 Analisi 5

3.1 Analisi del dominio 5

3.2 Analisi e specifica dei requisiti 5

3.3 Use case 9

3.4 Pianificazione 10

3.5 Descrizione Gantt 11

3.5.1 Analisi 11

3.5.2 Progettazione 11

3.5.3 Implementazione 11

3.5.4 Test 11

3.5.5 Documentazione 11

3.6 Analisi dei mezzi 12

3.6.1 Software 12

3.6.2 Hardware 12

4 Progettazione 13

4.1 Design dei dati e database 13

4.2 Design dell’architettura del sistema 13

4.3 Design delle interfacce 16

4.4 Design procedurale 26

5 Implementazione 27

6 Test 27

6.1 Protocollo di test 27

6.2 Risultati test 29

6.3 Mancanze/limitazioni conosciute 29

7 Consuntivo 29

8 Conclusioni 31

8.1 Sviluppi futuri 31

8.2 Considerazioni personali 31

9 Bibliografia 32

9.1 Bibliografia per articoli di riviste: 32

9.2 Bibliografia per libri 32

9.3 Sitografia 32

10 Glossario 33

11 Indice delle figure 33

12 Allegati 33

# Introduzione

## Informazioni sul progetto

Allievo: Robin Sartore

Docente: Michel Palucci

Scuola: SAM Trevano

Sezione: Informatica

Modulo:306

Data inizio:11.09.2024

Data fine: 18.12.2024

## 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 consiste nella creazione di un’applicazione per un quiz di nome Ziquiz, con varie modalità di domande, difficoltà, e tipologie e si può usare per creare un quiz personalizzato e nel caso usarlo per studiare si visualizzerà anche un proprio punteggio per le varie categorie e difficoltà e verrà salvato localmente un top score.

Un altro scopo è quello di iniziare a fare progetti dato che l’anno prossimo dovrò fare il LPI e quindi è meglio iniziare a prepararsi prima per arrivare al meglio preparati e con abbastanza esperienza per riuscire a fare un buon progetto.

# Analisi

## Analisi del dominio

Il prodotto potrà essere usato come svago nel tempo libero e sfidare gli amici a chi fa il punteggio più alto e vedere chi ha più conoscenze sulle varie categorie o un allenamento per studiare una qualsiasi materia o formazione con domande personalizzate, chiaramente esistono già vari quiz online su cose diverse ma questo si differenzia dagli altri perché oltre ad avere una modalità studio i quiz sono su varie categorie con varie difficoltà e mi sembra molto innovativo.

Gli utenti potranno migliorare e sviluppare una conoscenza su vari argomenti e migliorarsi ogni volta superando il loro top score.

I quiz che ci sono già nel web sono pieni di animazioni e effetti quindi devo provare ad avere anch’io qualche animazione e non qualcosa di monotono.

## 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.

|  |  |
| --- | --- |
| **Req-01** | |
| **Nome** | Creazione interfaccia Menu |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Comprensibile all’utente |

|  |  |
| --- | --- |
| **Req-02** | |
| **Nome** | Possibilità di scegliere la tipologia di quiz |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Comprensibile all’utente |

|  |  |
| --- | --- |
| **Req-03** | |
| **Nome** | Possibilità di creare un quiz |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Comprensibile all’utente |

|  |  |
| --- | --- |
| **Req-04** | |
| **Nome** | Possibilità di scegliere la difficolta del quiz |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Comprensibile all’utente |

|  |  |
| --- | --- |
| **Req-05** | |
| **Nome** | Possibilita di vedere il Topscore |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Comprensibile all’utente |

|  |  |
| --- | --- |
| **Req-06** | |
| **Nome** | Studiare su quiz creati |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Comprensibile all’utente |

|  |  |
| --- | --- |
| **Req-07** | |
| **Nome** | Musica di sottofondo |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Comprensibile all’utente |

|  |  |
| --- | --- |
| **Req-08** | |
| **Nome** | Visualizzare si ha sbagliato |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Comprensibile all’utente |

|  |  |
| --- | --- |
| **Req-09** | |
| **Nome** | Aggiunta animazioni |
| **Priorità** | 2 |
| **Versione** | 1.0 |

|  |  |
| --- | --- |
| **Req-010** | |
| **Nome** | Scegliere una modalità di studio |
| **Priorità** | 2 |
| **Versione** | 1.0 |

**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

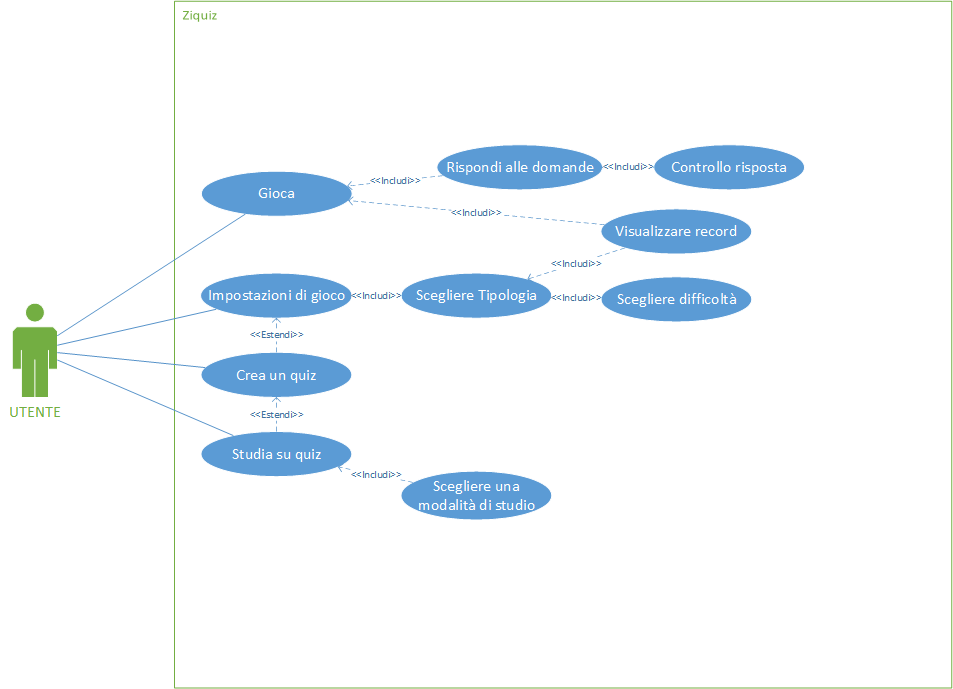


Figura use case

Questo è lo schema delle use case del progetto. In pratica l’utente può scegliere le impostazioni del quiz, la tipologia e la difficolta di cui può vedere il suo record in quella specifica modalità, può anche creare un quiz e studiare sul quiz con varie diverse modalità di studio e può anche giocare rispondendo alle domande e visualizzando gli errori.

## Pianificazione

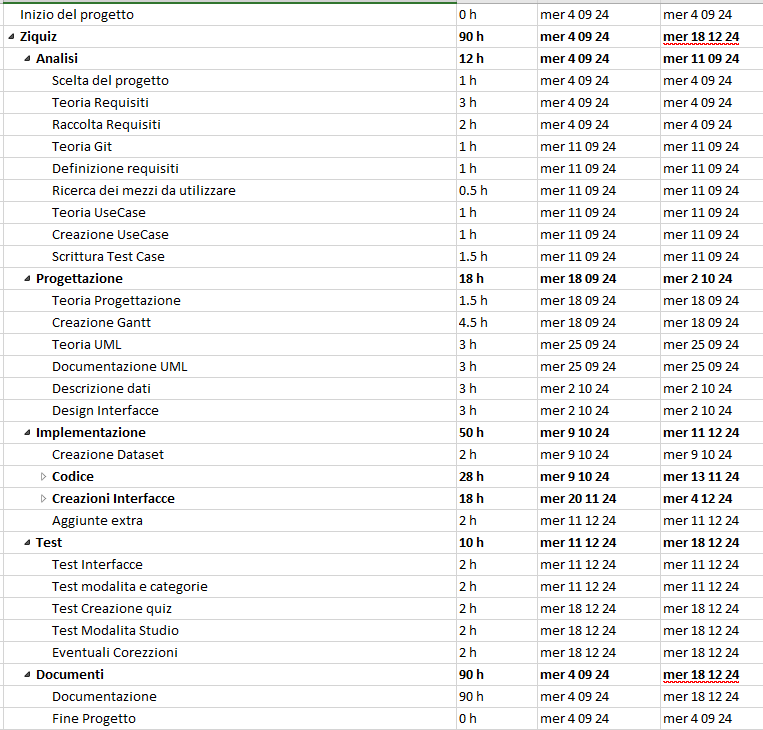


Figura Esempio di diagramma di Gantt

Questo è il Gantt di 90 ore del mio progetto, ho applicato una metologia waterfall.

## Descrizione Gantt

### Analisi

Per l’analisi ho programmato 2 giornate lavorative da 6 ore includendo la teoria fatta a lezione e ho ipotizzato dei tempi per le varie parti.

### Progettazione

Per la progettazione ho calcolato più ore della Analisi, ho messo una lunga tempistica soprattutto per la costruzione del gantt.

### Implementazione

L’implementazione è la parte più importante del progetto per questo ho previsto 50 ore di lavoro dato che devo ancora bene capire come leggere le domande e come crearle.

### Test

Per i test ho previsto 10 ore per i vari test e le eventuali correzioni ho preso più di un giorno perché non riuscirei magari a correggere molti errori in un giorno.

### Documentazione

Per la documentazione ho calcolato la durata intera quindi 90 ore anche se in realtà prenderà il 305 del tempo ma verrà fatta man mano.

## Analisi dei mezzi

Per la realizzazione del progetto è stato utilizzato un PC scolastico con Visual Studio 2022

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

Visual studio 2022

Hosting servito dalla scuola

### Hardware

Computer fornito dalla scuola:

32 gb di RAM

Windows 10 Education 22H2

Processore 13th Gen Intel(R) Core(TM) i7-13700 2.10 GHz

# Progettazione

## Design dei dati e database

L’architettura del sistema si basa su un file json in cui ci saranno le domande e le risposte per le domande e un altro file in cui verrano salvate localmente le domande e le risposte del utente

## Design dell’architettura del sistema

Diagramma di classi

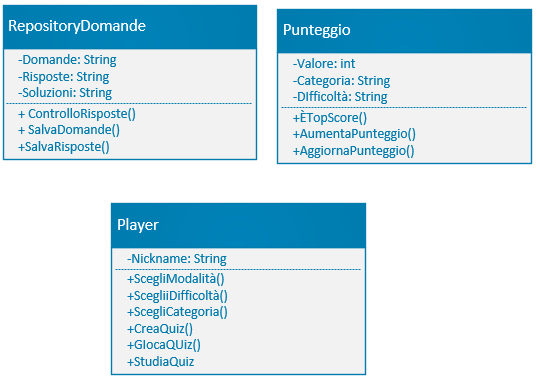


Figura 6 Diagramma di classi

## Design delle interfacce

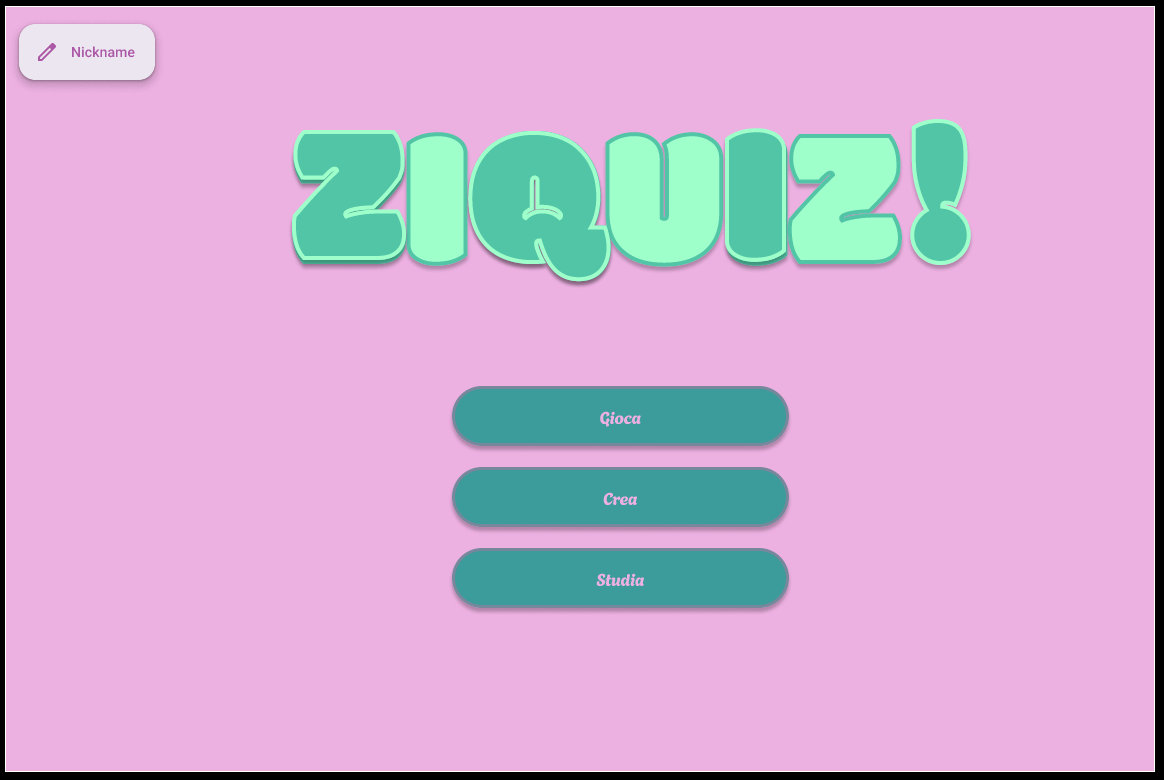
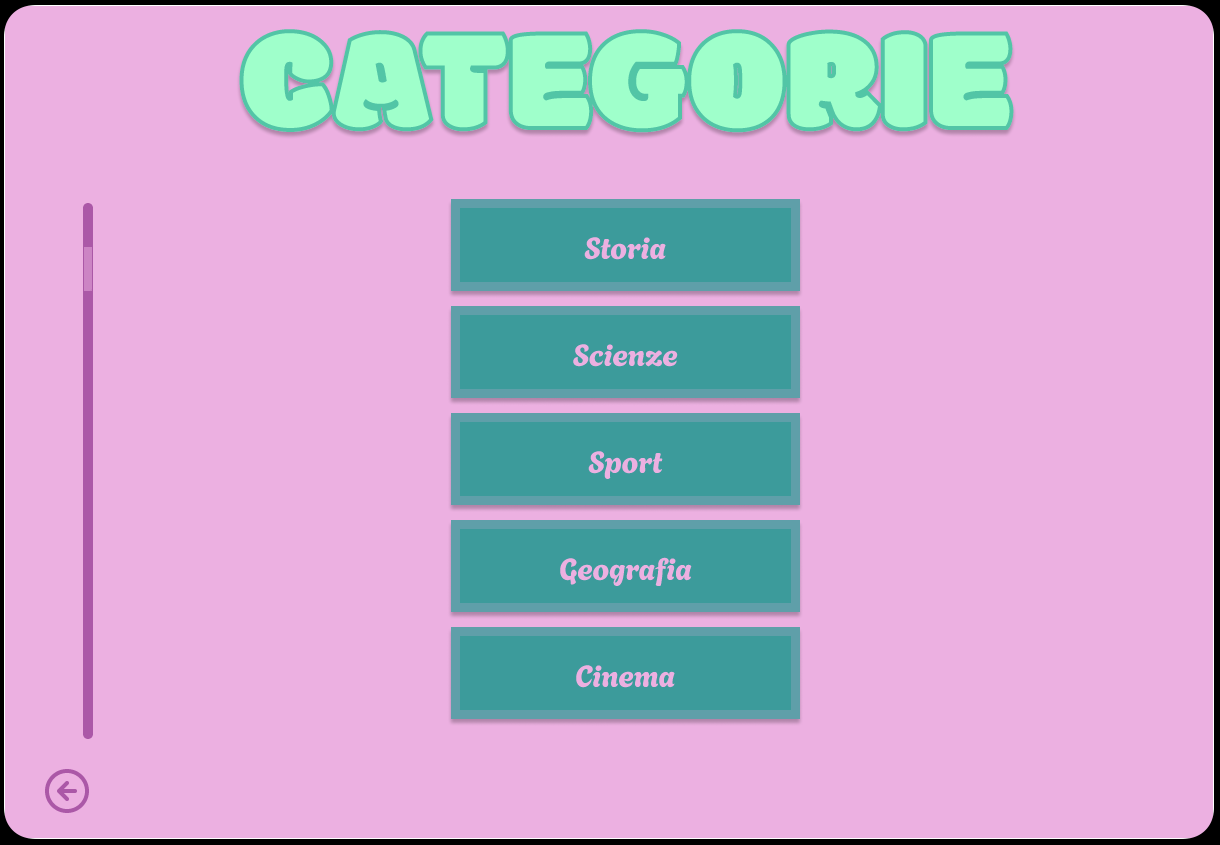


Figura 6 Interfaccia menù

Questa è l’interfaccia del menu principale qui l’utente potrà scegliere cosa fare e inserire il suo nickname, e si può leggere il titolo dell’applicazione.



Figuura 7 Categoria

Qua si potrà selezionare la categoria che più ci piace, si potrà scegliere anche un quiz creato dal utente.



Figura 8 difficoltà

Qui invece si sceglierà la difficoltà e si potrà vedere il topscore in base a ogni difficoltà.



Figura 9 esempio domanda

In Questa interfaccia si può vedere come viene mostrato il quiz con un punteggio in alto a destra



Figura 10 esempio errore

Qua si può vedere come viene mostrato un errore e la risposta correttà.

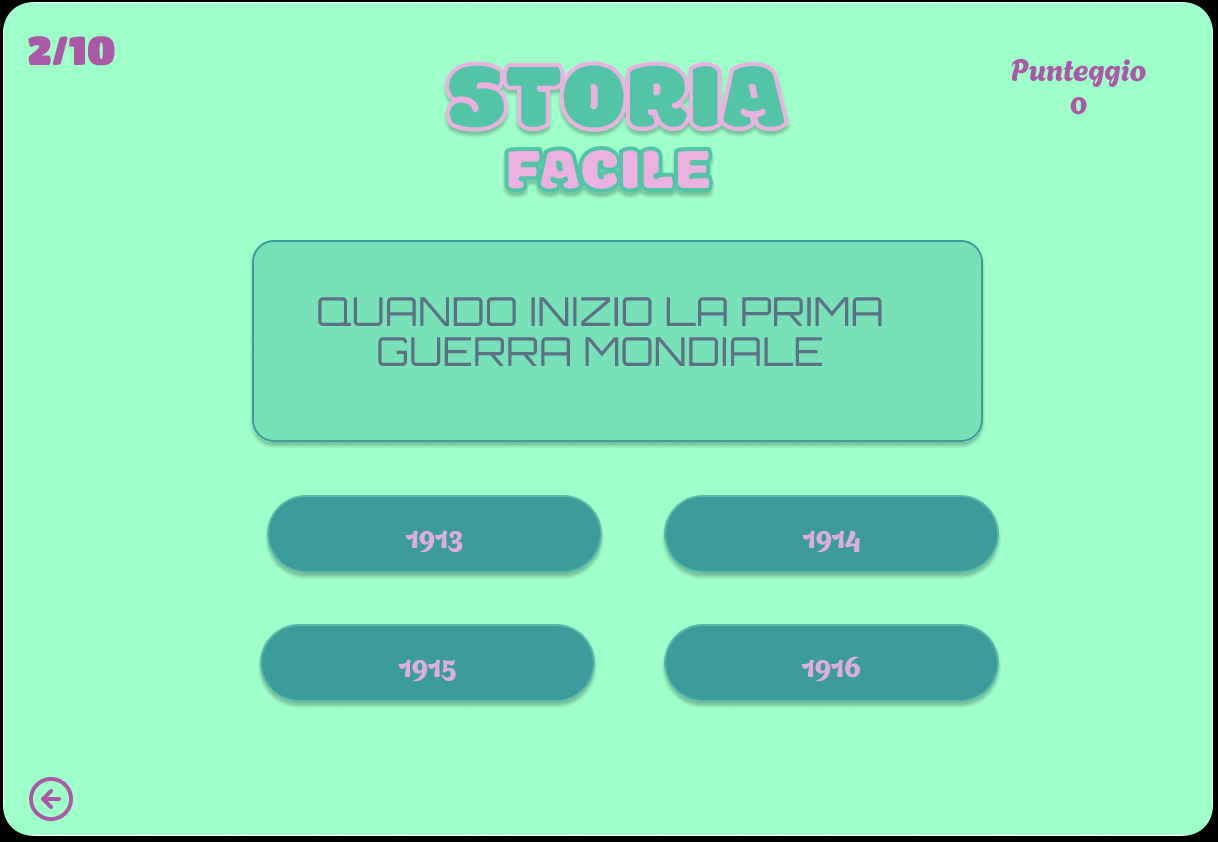
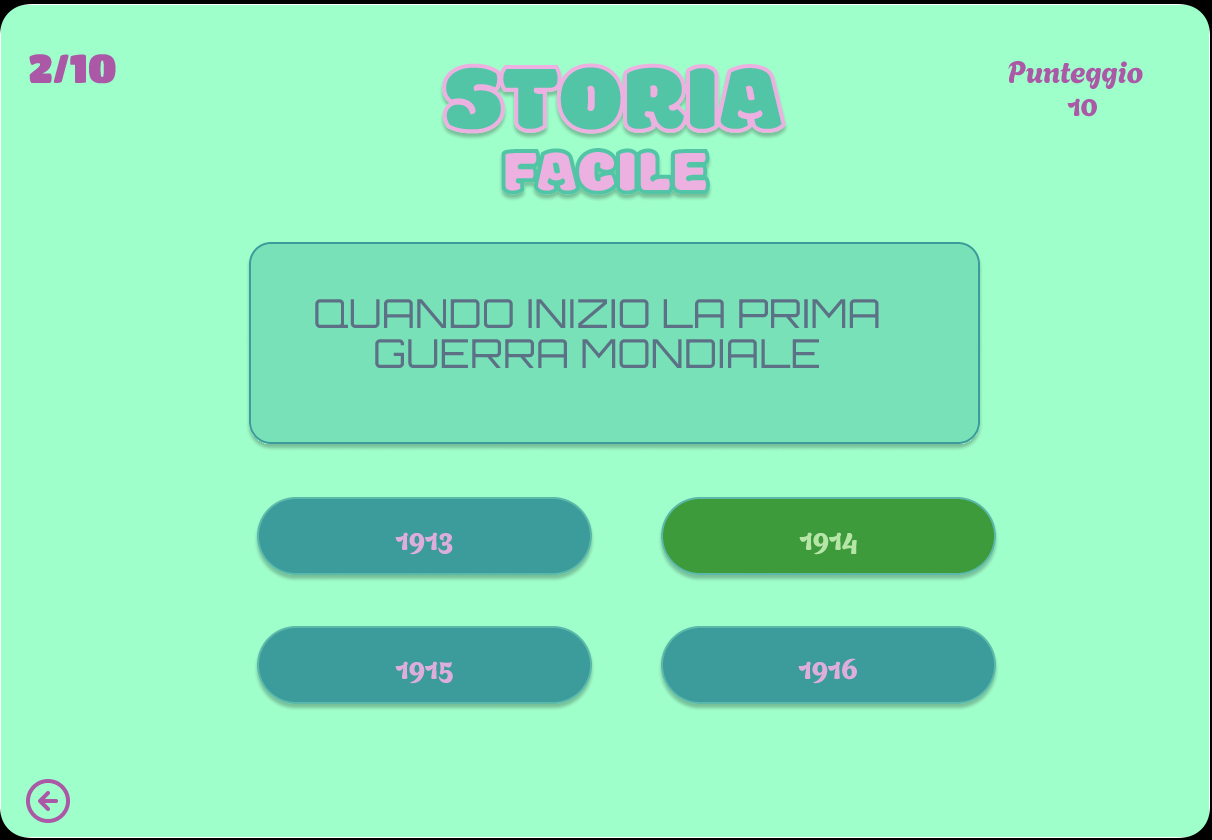


Figura 11 esempio domanda 2

Questa è un'altra domanda e si può vedere in alto a sinistra che è stat modificato il counter delle domande.

Figura 12 Esempio correzione 2

In questa interfaccia si può vedere invece quando viene inserita una risposta corretta , e si vede anche il punteggio aumentare.



Figura 13 Uscita da quiz

Qui viene mostrato il messaggio di quando si vuole uscire dal quiz

Figura 14 Modalità Crea

Qua si vede la schermata principale in cui si vedono i vari dati da inserire per creare un quiz.

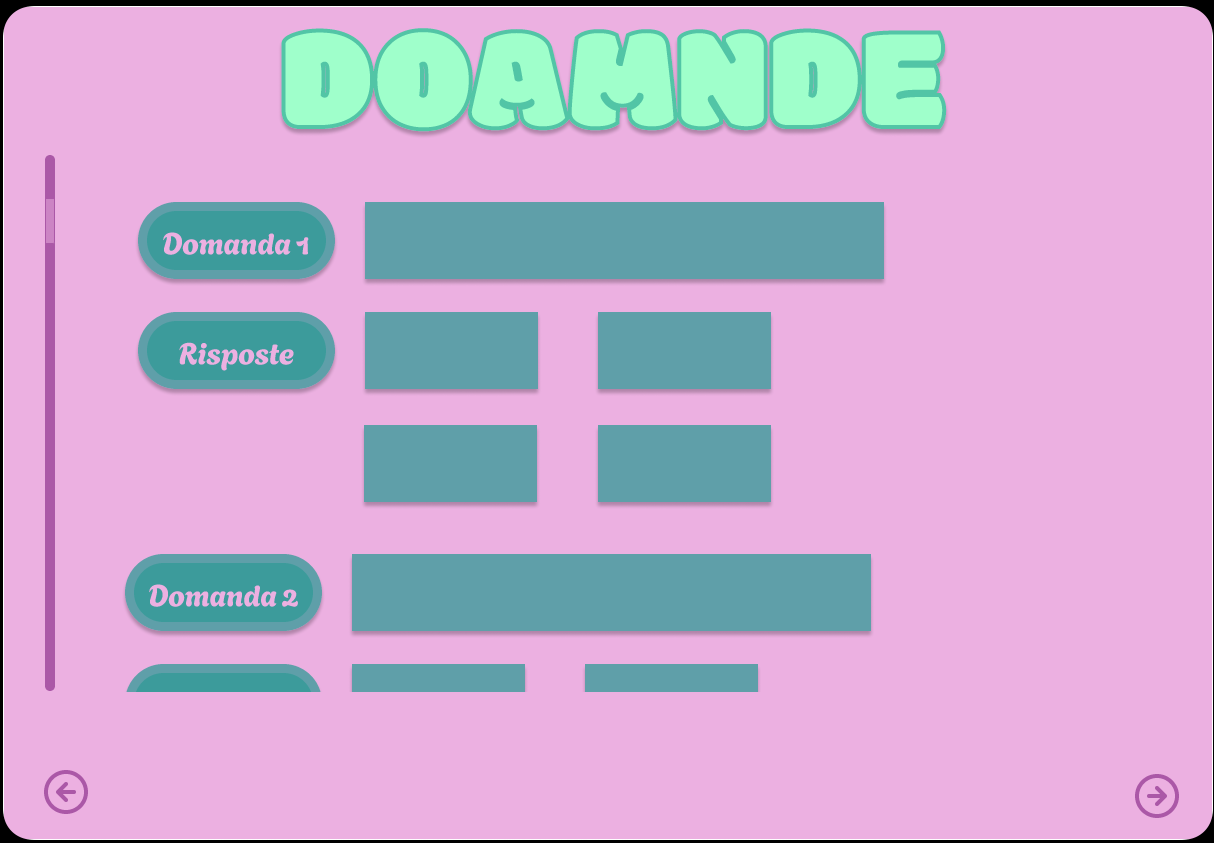


Figura 15 domande modalita crea

Qua invece viene mostrata la schermata in cui si selezioneranno le domande e le risposte da inserire nel nostro quiz personalizzato.

## Design procedurale

Diagramma Swimlanes

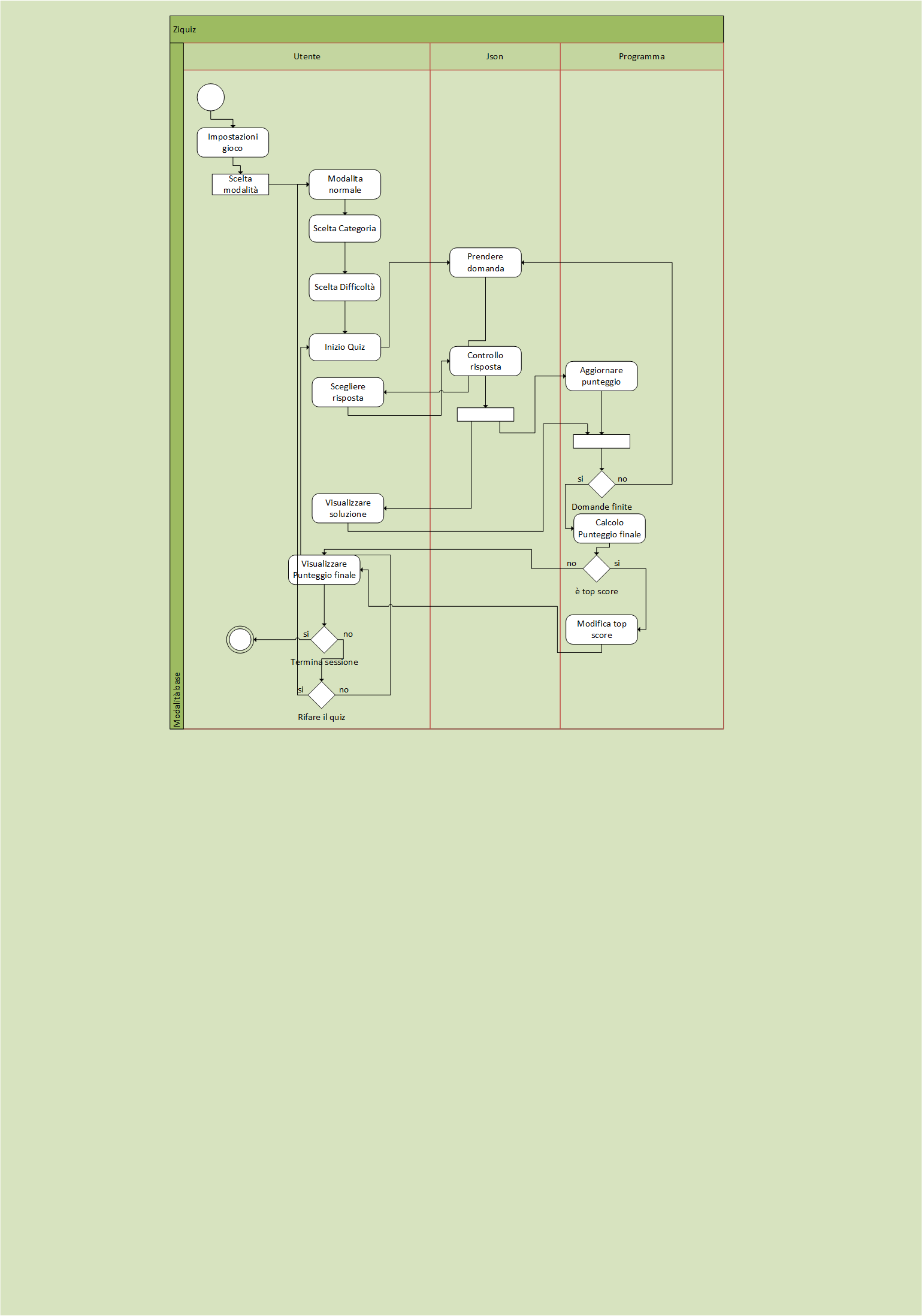


Figura 4 Diagramma Swimlanes base

Diagramma Swimlanes 2

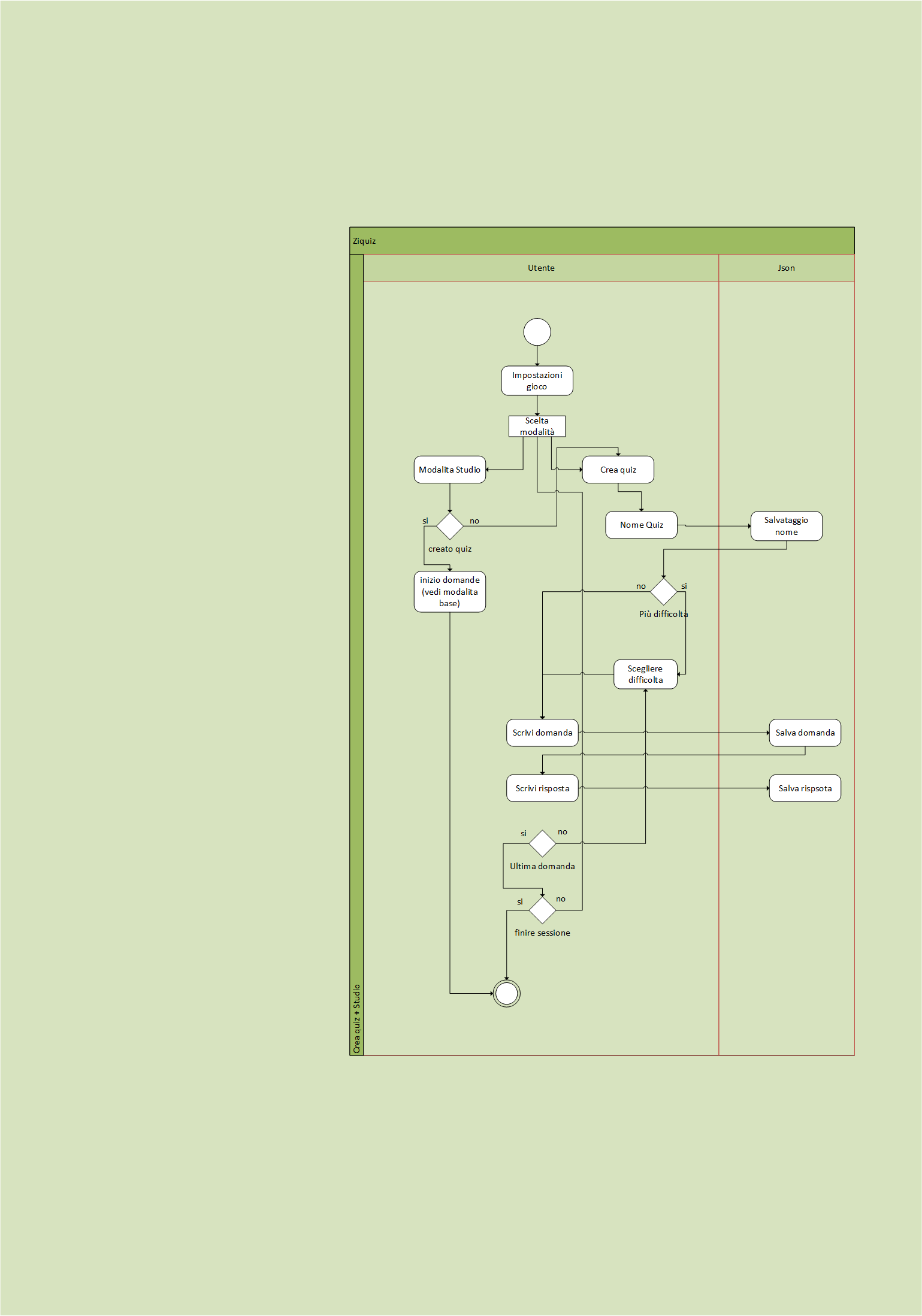


Figura 5 Diagramma Swimlanes (studio + creaquiz)

# 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-001 | **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.

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Risultato ottenuto** | **Stato** |
| TC-001 | Testo o immagine a conferma del test | Passato |
| TC-002 | Testo o immagine a conferma del test | Fallito |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## 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).

Stile orizzontale



CONSUNTIVO

Figura Esempio di diagramma di 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.

# Glossario

|  |  |
| --- | --- |
| **Termine** | **Significato** |
| ASCII | American Standard Code for Information Interchange, codice per la codifica di caratteri. |
|  |  |
|  |  |
|  |  |
|  |  |
| ZFS | Zettabyte File System, filesystem opernsource a 128 bit con funzionalità avanzate. |

# Indice delle figure

[Figura 1 Esempio di diagramma di Gantt 7](#_Toc124775984)

[Figura 2 Esempio di diagramma di Gantt 8](#_Toc124775985)

[Figura 3 Diagramma ER Chen 10](#_Toc124775986)

[Figura 4 Diagramma ER Barker 11](#_Toc124775987)

[Figura 5 Esempio di diagramma di Gantt consuntivo 16](#_Toc124775988)

# 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
* …