

# Relazione di Progetto - Lab. Sistemi

## Frontespizio

**Titolo progetto:** Web Arcade & Utility Tools

**Autori:** Ferrarese Tommaso, Zampieron Andrea

**Data di consegna:** 29 aprile 2025

## Indice

1. [Introduzione](#)
  1. [Scopo del progetto](#)
  2. [Requisiti](#)
  3. [Tecnologie utilizzate](#)
2. [Idea iniziale e pianificazione](#)
  1. [Concezione dell'idea](#)
  2. [Analisi dei requisiti](#)
  3. [Suddivisione dei compiti](#)
  4. [Pianificazione temporale](#)
3. [Fasi di sviluppo](#)
  1. [Progettazione dell'architettura](#)
  2. [Implementazione delle funzionalità](#)
    1. [Passaggio dei parametri](#)
    2. [Sessioni](#)
    3. [Funzioni PHP](#)
    4. [Chiamate asincrone](#)
    5. [Gestione dati con database NoSQL](#)
  3. [Interfaccia utente \(UI\)](#)
  4. [Test e debug](#)
4. [Difficoltà e soluzioni](#)
  1. [Difficoltà riscontrate](#)
  2. [Strategie utilizzate](#)
  3. [Lezioni apprese](#)
5. [Conclusioni](#)
  1. [Valutazione del progetto](#)
  2. [Possibili miglioramenti](#)
  3. [Considerazioni finali](#)
- [Bibliografia/Sitografia](#)
  1. [Bibliografia](#)
  2. [Sitografia](#)
- [Appendici](#)
  1. [Codice sorgente](#)
  2. [Screenshot dell'interfaccia](#)
  3. [Diagrammi architetturali](#)

# **1. Introduzione**

## **1.1 Scopo del progetto**

Sviluppare un progetto di sistemi che comprenda gli argomenti svolti durante l'anno.

Il progetto consiste in un sito web che offre giochi arcade e strumenti utili per gli utenti.

L'obiettivo è fornire un'esperienza interattiva e divertente, integrando giochi classici con funzionalità utili.

## **1.2 Requisiti**

- HTML5 e CSS3 per struttura e grafica
- JavaScript per dinamica e minigiochi
- PHP per gestione dati utenti
- Database NoSQL SleekDB
- Misure di cybersecurity per la sicurezza dei dati del sitoweb

## **1.3 Tecnologie utilizzate**

HTML5, CSS3, JavaScript, PHP, SleekDB.

# **2. Idea iniziale e pianificazione**

## **2.1 Concezione dell'idea**

L'idea del progetto è emersa dalla volontà di realizzare un sito web in grado di gestire dati e interazioni utente in modo efficiente e sicuro.

Inizialmente, si è pensato a un progetto che potesse combinare giochi arcade successivamente, invece, si è deciso di integrare anche strumenti utili per l'interazione con gli utenti e per il salvataggio dati nei database.

## **2.2 Analisi dei requisiti**

Analizzando i requisiti, si è deciso di utilizzare HTML5 e CSS3 per la struttura e la grafica del sito, JavaScript per la parte dinamica e per i minigiochi, PHP per la gestione dei dati degli utenti e SleekDB come database NoSQL per la memorizzazione dei dati.

Inoltre, sono state implementate misure di cybersecurity per garantire la sicurezza dei dati degli utenti.

Attraverso queste tecnologie e la idea iniziale, si è cercato di creare un sito web che fosse non solo funzionale, ma anche sicuro e user-friendly.

Si è deciso, quindi, prima di tutto di realizzare una pianificazione del progetto in un file di testo.

Fatto ciò, si è potuto avere una visione d'insieme del progetto e delle sue varie fasi di sviluppo per poi dividersi i compiti tra i membri del gruppo.

## 2.3 Suddivisione dei compiti

La suddivisione dei compiti è stata effettuata in base alle competenze e agli interessi di ciascun membro del gruppo.

Ferrarese Tommaso si è occupato principalmente della parte principale del sito web, più precisamente della sua parte della struttura in html, parte grafica usando css, dinamicità del sito con javascript, php per la gestione dei dati tra le diverse schede, database NoSQL SleekDB per la protezione dei dati degli utenti, cybersecurity usata nel database e qualche minigioco. Zampieron Andrea si è occupato principalmente dei minigiochi, della loro struttura e grafica, la loro dinamicità e esperienza con l'utente.

## 2.4 Pianificazione temporale

Ogni fase del progetto è stata pianificata in modo da garantire un progresso costante e una gestione efficace del tempo.

La pianificazione temporale è stata gestita singolarmente da ogni membro del gruppo, in modo da garantire che ciascuno potesse lavorare secondo le proprie disponibilità e scadenze.

Sottostante si può vedere il cronoprogramma di ogni componente del gruppo:

- **Ferrarese Tommaso**  
cronoprogrammaFerrareseTommaso
- **Zampieron Andrea**  
cronoprogrammaZampieronAndrea

## 3. Fasi di sviluppo

### 3.1 Progettazione dell'architettura

Le tecnologie usate per la realizzazione del progetto sono state accennate nella sezione dei requisiti (1.2 Requisiti).

Per il front-end, ovvero per la parte visibile del sito web, sono stati utilizzati HTML e CSS per la struttura e la grafica ed anche JavaScript per la parte dinamica e per i minigiochi.

Per il back-end, ovvero per la parte non visibile del sito web, sono stati utilizzati PHP per la gestione dei dati degli utenti e SleekDB come database NoSQL per la memorizzazione dei dati.

La parte di front-end è visibile all'utente e di solito comprende tutto ciò che l'utente può vedere e con cui può interagire.

La parte di back-end è invisibile all'utente e comprende tutto ciò che avviene dietro le quinte, come la gestione dei dati e la logica del sito.

In questo progetto il back-end è stato usato principalmente per gestire i dati dei diversi utenti nel database NoSQL.

## **3.2 Implementazione delle funzionalità**

### **3.2.1 Passaggio parametri**

Usando le tecnologie descritte nella sezione 1.3 Tecnologie utilizzate, sono state implementate le funzionalità del sito web.

Usando il back-end si è potuto gestire il passaggio dei parametri tra le varie schede del sito web, la gestione delle sessioni, le funzioni PHP per la gestione dei dati e le chiamate asincrone.

Più specificatamente si sono usati i vari modificatori di PHP come GET e POST per il passaggio dei parametri tra schede.

Un esempio di passaggio dei parametri è visibile nel file di registrazione/login al sito web, dove si usano i metodi GET per mostrare errori quando si inseriscono dati errati e metodi POST per passare i dati di registrazione/login ad una pagina di controllo la quale richiederà al database di controllare se i dati inseriti sono corretti o meno.

### **3.2.2 Sessioni**

Le sessioni sono state implementate affinché si potessero salvare i dati poco importanti degli utenti loggati, come ad esempio il nome utente, e una variabile che assicura l'accesso.

Ogni scheda del sito web ha una variabile di sessione che controlla se la sessione già esiste o se deve essere creata, nella scheda del profilo dell'utente, invece, c'è anche un controllo per il login.

### **3.2.3 Funzioni PHP**

Le funzioni PHP implementate sono state usate per migliorare e generalizzare il codice del sito web.

Le funzioni implementate per la gestione della sessione sono 3:

- `inizializzaSessione()`

Questa funzione viene usata per inizializzare la sessione e per controllare se è già stata creata.

- `distruggiSessione($paginaWeb)`

Questa funzione viene usata per eliminare e distruggere i dati e la sessione.

- `distruggiSessioneConPassaParametriErrore($paginaWeb, $messaggioErrore)` Questa funzione viene usata dalla pagina del profilo per cancellare l'account, eliminare la sessione col passaggio di un messaggio finale.

Le funzioni usate per le chiamate asincrone sono 2:

- `controllaEsistenza($nome, $videogiochi)`

Questa funzione viene usata per controllare se il videogioco esiste o meno tra i giochi disponibili.

- `ottieniPercorso($nome)`

Questa funzione viene usata per ottenere il percorso del videogioco da visualizzare.

La lista dei videogiochi disponibili è stata creata in una funzione PHP ma si poteva salvarla anche in un file JSON.

- `ottieniListaVideogiochi()`

Questa funzione viene usata per ottenere la lista dei videogiochi disponibili.

La lista è stata creata in un file PHP e non in un file JSON per velocizzare il caricamento della pagina.

Le funzioni usate dal database sono salvate nello stesso file php il quale esegue(quando richiesto), le funzioni per recuperare il database creato o per crearlo. Tra le funzioni usate per il database ci sono:

- `creaNuovoUserDatabase($nome, $cognome, $dataNascita, $sesso, $residenza, $username, $mail, $password)`

Questa funzione viene usata per creare un nuovo utente nel database se la mail inserita non esiste già.

- `recuperaAccessoDatabase($username, $mail, $password)`

Questa funzione viene usata per recuperare l'accesso al database se i dati inseriti sono corretti.

Se i dati inseriti sono corretti viene modificata la variabile del nome utente in sessione e attivato l'accesso.

- `ottieniDatiUtente($username, $mail)`

Questa funzione viene usata per ottenere i dati dell'utente loggato. I dati ottenuti sono tutti quelli salvati nel file JSON dell'utente.

- `eliminaUtente($username, $mail)`

Questa funzione viene usata per eliminare l'utente dal database e dal file JSON.

Questa funzione viene usata dalla pagina del profilo dell'utente per eliminare l'account.

- `ottieniDatiUtenti()`

Questa funzione viene usata per ottenere i dati di tutti gli utenti registrati nel database.

I dati ottenuti sono tutti quelli salvati nel file JSON di ogni utente.

- creaTabellaUtenti()

Questa funzione viene usata per creare una tabella con i dati di tutti gli utenti registrati nel database.

- ottieniArrayDati(\$nome, \$cognome, \$dataNascita, \$sesso, \$residenza, \$username, \$mail, \$password)

Questa funzione viene usata per creare un array chiave-valore con i dati passati.

Per garantire la sicurezza dei dati degli utenti, sono state implementate misure di cybersecurity come l'hash della password automatiche dal database. Le funzioni usate per la cybersecurity sono:

- ottieniPepeHash()

Questa funzione viene usata per ottenere il pepe hash per l'hash della password.

Il pepe hash è un codice casuale che viene usato per rendere più sicuro l'hash della password.

- creaHashPassword(\$password)

Questa funzione viene usata per creare l'hash della password.

L'hash della password è un codice casuale che viene usato per rendere più sicura la password.

- verificaHashPassword(\$password, \$hash)

Questa funzione viene usata per verificare se l'hash della password è corretto.

Si era previsto di implementare anche una funzione per controllare se un utente è offline o inattivo. Purtroppo, a causa del tempo limitato a disposizione, non è stato possibile comprendere e realizzare correttamente questa funzionalità.

### **3.2.4 Chiamate asincrone**

Le chiamate asincrone sono state implementate per migliorare l'esperienza utente per trovare velocemente un videogioco dalla homepage, e per l'ordinamento dei diversi videogiochi secondo un criterio preciso.

La seconda idea non si è potuta realizzare a causa del poco tempo a disposizione ma la prima idea è stata implementata.

Infatti questa funzionalità permette di cercare un videogioco dalla homepage senza dover ricaricare la pagina, ma semplicemente scrivendo il nome del videogioco nella barra di ricerca e cliccando sul tasto di ricerca. Il risultato ottenuto sarà un messaggio che serve per il redirect alla pagina col gioco.

### 3.2.5 Gestione dati con database NoSQL

Per il progetto è stato utilizzato un database NoSQL, in particolare SleekDB, che memorizza i dati in formato JSON all'interno del file system.

A differenza dei database relazionali, non esiste uno schema rigido: i dati sono organizzati in collezioni, che possono essere pensate come insiemi di documenti.

I dati sono stati modellati in collezioni (simili a tabelle nei database relazionali), ciascuna contenente documenti JSON.

Ogni documento rappresenta un'entità (ad esempio un utente) ed è composto da coppie chiave-valore.

La scrittura dei dati si è fatta solo quando l'utente si registra al sito web, e si è usata la funzione predefinita di sleekdb insert la quale crea un file JSON con i dati di un array passato come parametro.

La lettura dei dati si è fatta quando l'utente si logga al sito web, accede al profilo per vedere i propri dati o quando si visualizzano i diversi utenti presenti nel database, oppure per eliminare l'account. Le funzioni usate per questo, sono predefinite di sleekdb, e sono:

- `findOneBy()`

Questa funzione legge i diversi file fino a trovarne uno che abbia le stesse caratteristiche dell'array passato come parametro.

- `findBy()`

Questa funzione legge i diversi file fino a trovarne uno o più che abbia/abbiano le stesse caratteristiche dell'array passato come parametro.

- `findAll()`

Questa funzione legge i diversi file e salva i diversi dati in un array.

- `deleteBy()`

Questa funzione legge i diversi file fino a trovarne uno o più che abbia/abbiano le stesse caratteristiche dell'array passato come parametro, una volta trovato lo elimina.

### 3.3 Interfaccia utente (UI)

L'UI è stata fatta e implementata per essere semplice da comprendere da parte dell'utente, tutte le pagine, ad eccezione dei giochi, seguono tutte lo stesso stile e formato, per cercare di fare sentire "a casa" l'utente.

Invece come appena menzionato, i giochi si avvalgono di una UI decisamente accattivante che cerca di spronare l'utente a provarli, cercando comunque di mantenere uno stile pulito e in linea con la semplicità del resto della pagina.

I pulsanti e le caselle di testo sono state realizzate per essere facilmente leggibili e raggiungibili dall'utente. Nel complesso tutto il progetto è stato realizzato per essere di facile uso anche da un'utente non esperto.

La homepage è stata realizzata per essere la pagina principale del sito web, e per essere la prima pagina che l'utente vede quando accede al sito. Il

menù di navigazione permette all'utente di accedere alle diverse pagine del sito web, come la pagina di registrazione, la pagina di login e la pagina del profilo dell'utente.

La barra di ricerca permette all'utente di cercare un videogioco specifico tra quelli disponibili.

### **3.4 Test e debug**

Le pagine sono state prima testate in formato HTML così che la loro struttura e grafica funzionassero come si era pensato poichè usando xampp e il formato PHP, certe volte, può causare il bug visivo delle diverse decorazioni.

Dopo aver testato le pagine in HTML e Css si è passati a testare le varie funzioni javascript usando il comando `console.log(...)` per aiutare a trovare più velocemente pezzi di codice non funzionanti e correggerli. Ovviamente ci sono stati alcuni errori causati dalla distrazione ma il sito web di prova li ha segnalati e si sono prese eventuali contromisure per evitarli e risolverli.

Successivamente si è passati a testare il progetto in php, usando la piattaforma Xampp -> Apache, in localhost.

Vedendo il funzionamento del sito si ha continuato ad aggiungere piccoli dettagli per una esperienza unica per l'utente, attraverso funzioni php e l'utilizzo dei database.

Col penultimo non si sono riscontrati problemi, invece con l'ultimo si hanno avuti problemi. Essi sono nati, dapprima a causa di una mancanza di basi con l'uso di sleekdb, poi con l'uso delle varie funzioni che da accesso la libreria (per esempio `findOneBy`) e infine a causa dell'hashing delle password e la mancanza di corrispondenza tra esse al login.

Per superare questi problemi si hanno fatto dei test unitari e di integrazione.

Infatti si sono copiate e incollate funzioni o parti di esse in nuovi file per testare la loro funzionalità e polimorfismo.

## **4. Difficoltà e soluzioni**

### **4.1 Difficoltà riscontrate**

Durante lo sviluppo del progetto, sono emerse diverse difficoltà sia a livello tecnico che organizzativo.

Alcune problematiche sono state affrontate e risolte grazie allo studio e alla sperimentazione, mentre altre hanno richiesto delle soluzioni temporanee o di compromessi, soprattutto a causa dei limiti di tempo disponibili.

Per lo più i problemi sono sorti quando si è iniziato ad usare la libreria per creare database, sleekdb, e con l'implementazione della cybersicurezza.

Inoltre si sono riscontrate delle difficoltà con i listener in javascript, ma fortunatamente erano degli errori di distrazione.

Il css non ha recato quasi nessun danno poichè sono state usate delle indicazioni più o meno semplici e basiche.

Il php ha, invece, causato difficoltà con gli usi di `$_GET` e `$_POST` poichè i dati venivano quasi sempre condivisi attraverso i database.



## 4.2 Strategie utilizzate

Di fronte alle difficoltà emerse durante lo sviluppo del progetto, è stato necessario adottare diverse strategie per cercare soluzioni efficaci.

Le decisioni sono state prese valutando attentamente le alternative disponibili, considerando sia i vincoli tecnici sia i limiti di tempo.

Si è cercato di mantenere una comunicazione aperta e costante tra i membri del gruppo, in modo da condividere le problematiche e le soluzioni trovate. Inoltre, si è fatto ricorso a risorse esterne, come documentazione e tutorial online, per approfondire le conoscenze necessarie a risolvere le difficoltà riscontrate.

Oltretutto le strategie adottate per risolvere questi inconvenienti sono state principalmente metodi di debug come stampa di dati con console log in javascript, echo in php, ispezioni della pagina per html e css.

Si è cercato in siti online come w3schools.com o html.it per risolvere o imparare e capire meglio il funzionamento delle diverse tecnologie usate.

## 4.3 Lezioni apprese

Durante lo sviluppo del progetto, sono state apprese diverse lezioni importanti che hanno contribuito alla crescita personale e professionale di ciascun membro del gruppo.

La gestione del tempo e delle risorse è stata una delle principali sfide affrontate, e si è imparato a pianificare meglio le attività e a stabilire priorità.

Inoltre, si è compreso l'importanza della comunicazione e della collaborazione all'interno del gruppo, che ha permesso di affrontare le difficoltà in modo più efficace.

Infine, si è acquisita una maggiore familiarità con le tecnologie utilizzate, in particolare con PHP e SleekDB, che hanno richiesto un certo grado di apprendimento e adattamento.

Avendo capito come funzionano i vari linguaggi si è potuto creare un decente sito web con le pagine che servono, le chiamate asincrone e il database sono state due parti fatte più che bene per quanto il nostro studio ci ha permesso.

Oltre a tutto questo si è riusciti ad usare ed implementare delle basi di cybersecurity per difendere i dati degli utenti da attacchi di malintenzionati.

In un prossimo futuro si potrebbe finire di implementare le idee trascurate, magari abbellire il css del sito e cercare di pubblicare una versione alpha come debutto nel web.

# 5. Conclusioni

## 5.1 Valutazione del progetto

Il progetto è stato sviluppato partendo da un'idea principale molto chiara, che mirava a creare uno "spazio" digitale dove l'utente potesse trovare una varietà di giochi pensati per intrattenere e offrire momenti di svago.

L'obiettivo era proprio quello di dare a chi lo utilizza l'opportunità di divertirsi, rilassarsi o magari anche stimolare la propria mente in modo

piacevole. Con il passare del tempo, abbiamo lavorato per rendere questo spazio il più accogliente e funzionale possibile, cercando di offrire una buona esperienza a tutti. Anche se siamo convinti di essere riusciti a realizzare questa visione iniziale e che il progetto stia dando dei risultati positivi, sappiamo bene che ci sono ancora delle aree su cui possiamo e dobbiamo lavorare. Ogni feedback ricevuto ci è stato utile e ci ha dato spunti per poter migliorare il servizio, implementare nuove funzionalità e perfezionare quelle già esistenti. Siamo consapevoli che ogni progetto ha bisogno di un continuo aggiornamento e ottimizzazione, e il nostro impegno è proprio quello di evolverci in base alle esigenze degli utenti, facendo sì che il nostro spazio rimanga sempre stimolante, interessante e, soprattutto, capace di garantire un'esperienza di gioco che possa essere sempre più coinvolgente e appagante. In futuro, lavoreremo per introdurre nuovi giochi, migliorare l'interfaccia e assicurare che tutti possano trovare qualcosa che soddisfi i propri gusti e desideri di intrattenimento.

## **5.2 Possibili miglioramenti**

Sicuramente, la priorità principale al momento sarebbe quella di completare e perfezionare tutti i giochi presenti nel progetto, assicurandoci che ogni gioco funzioni correttamente e offra un'esperienza soddisfacente agli utenti. Una volta raggiunto questo obiettivo, potremmo concentrarci su un ulteriore passo per arricchire l'esperienza di gioco: l'integrazione di un sistema di punteggio.

Questo sistema consentirebbe agli utenti di accumulare punti in base alle loro performance in ogni singolo gioco, creando una dimensione competitiva che potrebbe stimolare ulteriormente il loro coinvolgimento.

Successivamente, potremmo introdurre una classifica generale che raccoglie i punti guadagnati da tutti i giochi, permettendo agli utenti di confrontarsi con altri partecipanti e vedere chi si posiziona in cima alla classifica. Una tale funzionalità non solo aumenterebbe la motivazione degli utenti a giocare di più e migliorarsi, ma creerebbe anche un senso di comunità e competizione sana all'interno della piattaforma.

Naturalmente, l'implementazione di queste caratteristiche richiederà un'attenta pianificazione e sviluppo, ma riteniamo che rappresenterebbero un valore aggiunto per l'intero progetto, rendendo l'esperienza di gioco ancora più dinamica, coinvolgente e interattiva.

## **5.3 Considerazioni finali**

In conclusione, questo progetto ha rappresentato un'importante opportunità di crescita, sia a livello tecnico che creativo.

L'esperienza acquisita durante lo sviluppo ci ha permesso di affrontare sfide complesse, dalla progettazione alla realizzazione, e ci ha dato l'opportunità di esplorare nuove soluzioni, strumenti e tecnologie. Ogni fase del processo ci ha offerto occasioni per migliorare e perfezionare le nostre competenze, contribuendo a una comprensione più profonda di come sviluppare una pagina web user-friendly.

Abbiamo avuto modo di confrontarci con problemi reali legati all'ottimizzazione delle performance, alla gestione della user experience e alla progettazione di un sistema che fosse al contempo funzionale e

coinvolgente per gli utenti. Questi aspetti ci hanno spinto a pensare in modo critico e a trovare soluzioni creative, favorendo lo sviluppo di competenze tecniche, ma anche di capacità di problem-solving e gestione del progetto.

## **Bibliografia/Sitografia**

### **Bibliografia**

- Libro: Sistemi e Reti - Autori: Luigi Lo Russo, Elena Bianchi - Editore: HOEPLI - Volume 2
- Libro: Corso di informatica - Autori: Fiorenzo Formichi, Giorgio Meini, Massimiliano Sartour, Ivan Venuti - Editore: ZANICHELLI - Volume 2

### **Sitografia**

- [W3Schools](#)
- [HTML](#)
- [Google Search Central](#) (sitemap)
- [Support Host](#) (sitemap)
- [SleekDB](#)
- [stewardle](#)

## **Appendici**

Codice sorgente, screenshot dell'interfaccia e diagrammi architetturali sono allegati a parte.