 

**UNIVERSITÀ DEGLI STUDI DELLA BASILICATA**

DIPARTIMENTO DI INGEGNERIA

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA E DELLE TECNOLOGIE DELL'INFORMAZIONE

**RELAZIONE PROGETTO FINALE TAV**

**ASFJ: A Simple Firewall in Java**

Docente:

Ch.mo Prof. Gianvito Summa

Studente:

Michael Pio Stolfi 68787

**ANNO ACCADEMICO 2024-2025**

Sommario

[Sommario 2](#_Toc201860024)

[Introduzione 3](#_Toc201860025)

[Conclusioni 4](#_Toc201860026)

[Bibliografia 5](#_Toc201860027)

# Introduzione

La sicurezza delle applicazioni è un aspetto fondamentale nello sviluppo software, in particolare per le applicazioni che gestiscono dati sensibili e informazioni critiche. Questo report si propone di analizzare l'impatto dei design pattern sulla sicurezza di un’applicazione web, confrontando due versioni della stessa: una sviluppata senza design pattern e un’altra che implementa pattern mirati a mitigare vulnerabilità comuni.

L’analisi è stata condotta attraverso test di sicurezza focalizzati sulle vulnerabilità più critiche identificate dall’OWASP Top 10, tra cui “SQL Injection” e “Broken Access Control”. Per condurre i penetration test è stato utilizzato “Burp Suite”, una piattaforma avanzata che consente di intercettare e modificare le richieste HTTP per individuare e sfruttare eventuali punti deboli dell’applicazione.

Il progetto oggetto di studio è un sistema di gestione dell’autenticazione del personale aziendale, che prevede la gestione di tesserini digitali e il loro aggiornamento con diversi livelli di accesso.

Lo scopo di questo progetto è di dimostrare come l'adozione di design pattern possa contribuire a migliorare la sicurezza by design, limitando l’esposizione dell’applicazione a vulnerabilità note e migliorando la robustezza del sistema contro attacchi informatici. Attraverso test automatici e manuali, sono stati raccolti dati concreti sulle differenze di sicurezza tra le due implementazioni, permettendo di valutare l’efficacia delle tecniche di protezione adottate.

Nei capitoli successivi verranno presentati i dettagli del progetto, i metodi di testing impiegati e i risultati ottenuti, evidenziando i vantaggi dell’approccio basato sui design pattern nella prevenzione delle vulnerabilità più critiche.

# Test

Ciao

# Conclusioni

L'analisi condotta ha evidenziato come l'uso dei design pattern possa migliorare significativamente la sicurezza di un'applicazione web, mitigando vulnerabilità come Broken Access Control, SQL Injection, Identification and Authentication Failures e Security Logging and Monitoring Failures . Il confronto tra il progetto senza design pattern e quello con un'implementazione sicura ha dimostrato che l'adozione di pattern strutturati permette di prevenire attacchi e ridurre le superfici di rischio.

Tuttavia, è importante sottolineare che i pattern da soli non garantiscono la sicurezza. Devono essere integrati all'interno di un approccio più ampio di security by design e defense in depth, che preveda una gestione attenta delle credenziali, logging sicuro, monitoraggio continuo e aggiornamenti costanti. Solo attraverso un'architettura ben progettata e una strategia di sicurezza completa è possibile garantire un sistema robusto e resiliente agli attacchi.

In sintesi, l'utilizzo dei design pattern rappresenta una best practice fondamentale nello sviluppo software sicuro, ma deve essere affiancato da una consapevolezza continua sulle minacce emergenti e da una strategia di protezione che combini più livelli di difesa.

# Bibliografia

1. OWASP. “OWASP Top Ten: 2021”. <https://owasp.org/Top10/>
2. AGID. “Linee guida per lo sviluppo del software sicuro”. <https://www.agid.gov.it/it/sicurezza/cert-pa/linee-guida-sviluppo-del-software-sicuro>
3. Microsoft. “Microsoft Security Development Lifecycle (SDL)”. <https://learn.microsoft.com/it-it/compliance/assurance/assurance-microsoft-security-development-lifecycle>
4. Port Swigger. “Authentication vulnerabilities”. <https://portswigger.net/web-security/authentication>
5. Port Swigger. “SQL injection”. <https://portswigger.net/web-security/sql-injection#what-is-sql-injection-sqli>