

LocknShop: Documento di architettura



Autori:

Adjetey Isabelle
Andreotti Luca
Carillo Vincenzo
Gandolfi Matteo
Masa Biniam Abraha
Pinto Sabrina
Polzoni David

Data:

27 giugno 2025

Abstract

LocknShop è una piattaforma di e-commerce sviluppata con focus sulla sicurezza avanzata e la facilità d'uso. L'obiettivo è quello di fornire una applicazione web che permetta all'utente finale di effettuare acquisti in maniera sicura e veloce, senza compromettere gli aspetti critici legati alla protezione dei dati e alla gestione sicura dell'identità digitale. LocknShop rappresenta quindi un equilibrio tra esperienza utente fluida e architettura solida.

Il sistema viene diviso in tre livelli principali (frontend, backend, database) con la particolare attenzione al fatto che ogni componente possa comunicare in modo efficace e sicuro. L'utilizzo dei servizi Azure permette di avere una piattaforma coerente e facile da gestire, nonché già pronta per essere mantenibile e monitorata. Le scelte tecniche sono state fatte in base ai bisogni indispensabili per un progetto del genere: semplicità nel deploy, affidabilità nei dati, e strumenti di controllo chiari e accessibili.

Nonostante il progetto sia ancora in fase iniziale, è stato progettato con una logica rivolta all'espansione: in futuro potrà essere arricchito con nuove funzionalità, come per esempio sistemi di pagamento, gestione avanzata degli utenti, pannello dedicato agli amministratori e molto altro.

Il gruppo di lavoro si è impegnato a consegnare un prodotto concreto, ben fatto e soprattutto utile per dimostrare le proprie competenze in ambito tecnico, organizzativo e progettuale.

Roadmap

La seguente tabella Gantt rappresenta una **roadmap indicativa** per lo sviluppo del progetto LocknShop nel periodo compreso tra il **23 giugno** e il **2 luglio 2025**. I primi giorni sono stati dedicati alla documentazione progettuale e alla realizzazione del backend, mentre a partire dal 30 giugno inizierà lo sviluppo concreto delle funzionalità di sicurezza.

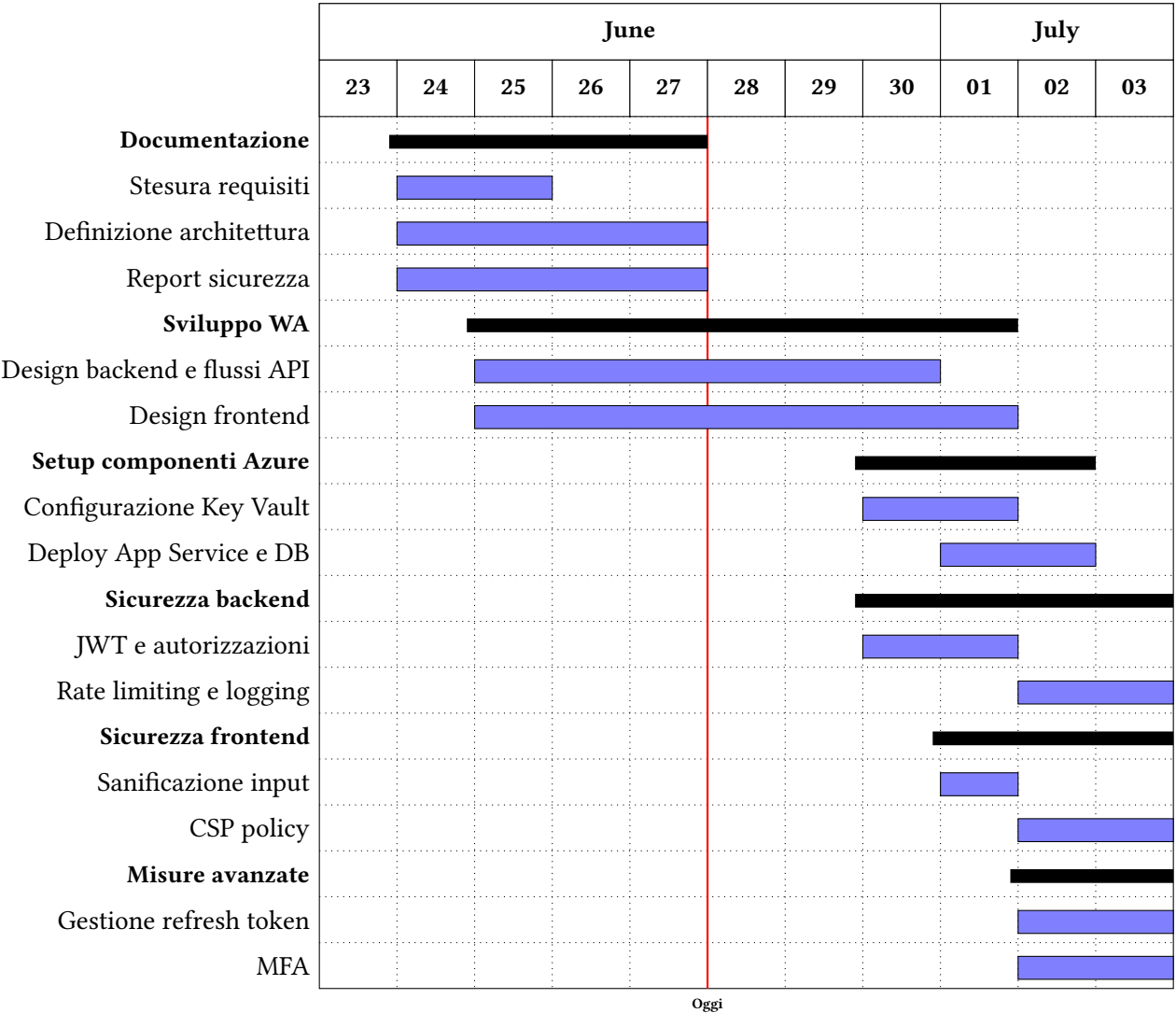


Figura 1: Pianificazione delle attività di sviluppo e sicurezza applicativa

Architettura logica della soluzione

Frontend (statico)

- HTML, CSS, JavaScript
- Responsive design, modalità chiara/scura
- Pagine realizzate: `index.html`, `catalogo.html`, `login.html`, `register.html`
- Comunicazione via *fetch API* con backend C#

Backend

- Linguaggio: C# con ASP.NET Core (Web API)
- API REST per:
 - Registrazione/Login
 - Gestione prodotti (CRUD)
 - Gestione carrelli e checkout
- Autenticazione con JWT
- Filtro di sicurezza (middleware auth on backend)
- Logging degli accessi

Database

- Database relazionale normalizzato
- Costruito secondo best practice (evitando ridondanze, favorendo integrità referenziale)

Modello Relazionale (Schema ER)

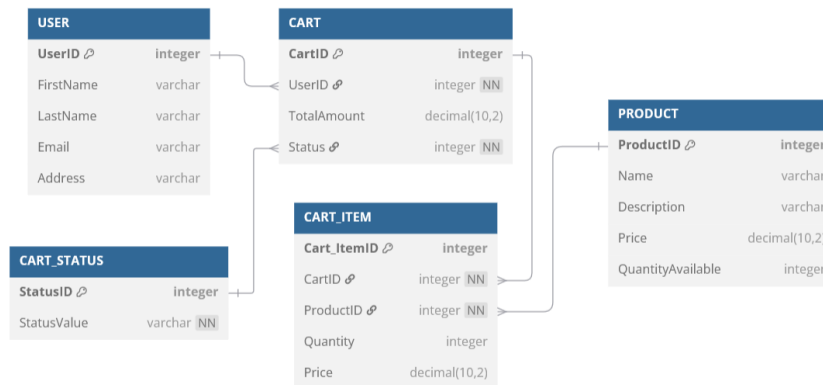


Figura 2: Schema ER del Database

Il database è composto dalle seguenti entità:

- **USER:**
 - UserID: chiave primaria
 - FirstName, LastName, Email, Address: informazioni anagrafiche
- **PRODUCT:**
 - Product ID: chiave primaria
 - Name, Description, Price, QuantityAvailable: gestione catalogo
- **CART:**
 - Cart ID: chiave primaria
 - UserID: utente proprietario del carrello
 - TotalAmount: importo totale
 - Status: stato del carrello (collegato a CART_STATUS)
- **CART_ITEM:**
 - Cart_ItemID: chiave primaria
 - Cart ID, Product ID: *Foreign Key* verso carrello e prodotto
 - Quantity, Price: informazioni su ogni voce del carrello
- **CART_STATUS:**
 - Status ID: chiave primaria
 - StatusValue: es. «Attivo», «Completato», «Annullato»

Servizi Azure scelti

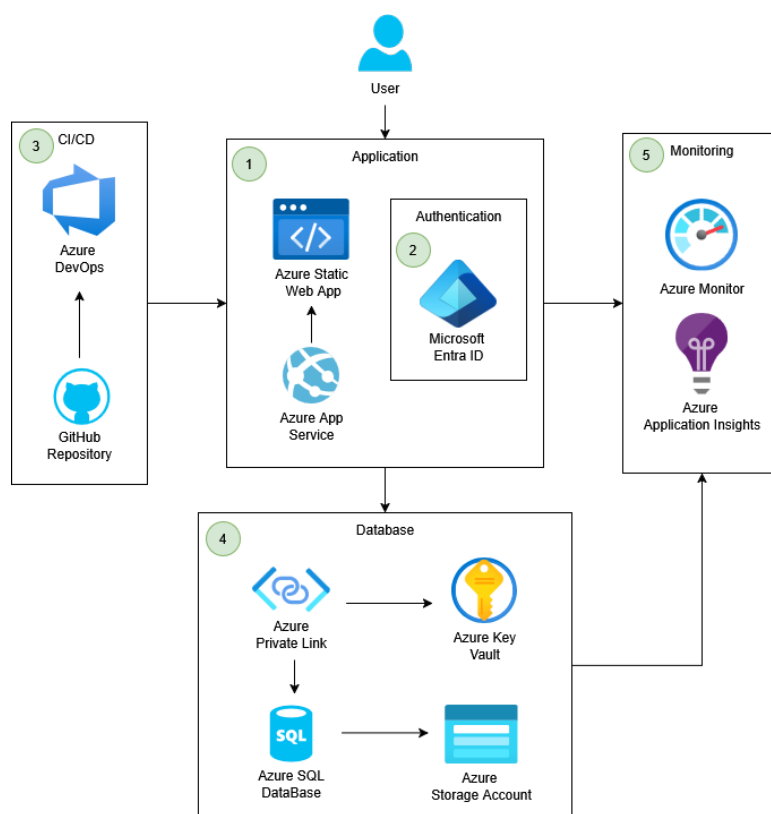


Figura 3: Architettura dei componenti

- **Azure App Service (1)**

Grazie a questo componente il gruppo di lavoro ha accesso a una gestione semplificata del ciclo di vita (deploy, aggiornamenti, scalabilità automatica) del progetto e anche all'integrazione diretta con strumenti di monitoraggio come Application Insights.

Questo servizio ospita il backend scritto in C# con ASP.NET Core. È pensato per pubblicare API in modo scalabile, con supporto nativo a deploy automatici, gestione del traffico e integrazione con strumenti di monitoraggio. Tra i vari piani disponibili, è stato scelto il *piano B1 (Basic Tier)*, dato che fornisce un buon bilanciamento tra costi (**48,44 €/mese**) e risorse (1 core e 1,75 GB di RAM).

- **Azure Static Web App (1)**

Questo servizio permette di pubblicare il frontend statico (HTML, CSS, JS); sfrutta una CDN globale per migliorare la velocità, fornisce HTTPS gratuito e consente il deploy diretto da GitHub. Essendo una soluzione leggera, stabile e **gratuita** è il perfetto candidato per il progetto, garantendo prestazioni elevate e deploy automatici.

- **Microsoft Entra ID (2)**

Questo servizio fornisce una serie di benefici quando utilizzato per la gestione degli accessi. Tra le varie funzionalità, permette agli amministratori del sistema di controllare gli accessi alle risorse, semplifica il processo del controllo multi fattore per l'autenticazione e consente di attivare i *Single Sign-On* (SSO) per gli utenti, evitando di dover inserire la password con frequenza elevata. Per gli scopi di questo progetto, questo servizio ha permesso di concentrare gli sforzi del team di lavoro in altri ambiti, dato che fornisce già un sistema funzionante e integrabile agli altri componenti Azure. Il costo mensile previsto per questo prodotto è di **110,16 €/mese**.

- **Azure Devops (3)**

Questo servizio viene utilizzato per la gestione del codice sorgente e per attivare il processo di CI/CD (*Continuous Integration / Continuous Deployment*). Permette di pubblicare automaticamente modifiche al codice in produzione, in maniera controllata. All'interno del progetto è estremamente utile per migliorare l'efficienza del team e per ridurre errori manuali. La sua versione di base è **gratuita**.

- **Azure Private Link (4)**

Questo servizio permette di accedere ad altri servizi Azure utilizzando un endpoint privato all'interno di una rete virtuale. In questo modo le comunicazioni tra, per esempio, database e backend dell'applicazione non devono più essere esposti alla rete pubblica, limitando così il rischio che possano essere intercettate. All'interno di questo progetto, la scelta di un servizio del genere assicura una maggiore sicurezza durante le comunicazioni interne tra i vari componenti dell'applicazione. Il costo previsto per questo servizio è di **8,23 €/mese**.

- **Azure SQL Database (4)**

Questo servizio gestisce tutti i dati strutturati della piattaforma, come per esempio utenti, prodotti, carrelli e relazioni tra entità. È stato scelto un database relazionale per la sua coerenza, facilità di interrogazione (usando strumenti come SQL) e pieno supporto alla normalizzazione. Il piano scelto è *DTU-based (S0)*, sufficiente per le esigenze iniziali del progetto. Il costo mensile è di **4,69 €/mese**.

- **Azure Key Vault (4)**

Questo servizio serve a proteggere credenziali sensibili, chiavi API o stringhe di connessione, evitando che vengano cablati all'interno del codice stesso dell'applicazione. In questo progetto viene utilizzato per proteggere informazioni sensibili necessarie all'applicazione, in particolare la chiave segreta per firmare i token e la stringa di connessione al database SQL. Inoltre, è compatibile nativamente in ASP.NET. Il costo è **gratuito**.

- **Azure Storage Account (4)**

Questo servizio è opzionale, previsto per l'eventuale gestione di asset statici (immagini prodotto, file, log, backup). Diventa indispensabile nel momento in cui si intende supportare upload di file da parte degli utenti o per caricare contenuti di dimensione elevata. Il costo previsto è di **17,96 €/mese**.

- **Azure Monitor + Application Insights (5)**

Questi due servizi sono componenti fondamentali per la sicurezza e la stabilità della piattaforma. Permettono di tenere traccia delle performance, errori e accessi, offrendo dashboard di analisi e allarmi automatizzati in caso di comportamenti sospetti. Sono strumenti indispensabili per un'applicazione cloud moderna. Il costo mensile stimato è di **1,60 €/mese**.

| Servizio | Utilizzo nel progetto | Costo stimato mensile |
|--------------------------|---|-----------------------|
| Azure App Service | Ospita il backend in C# (.NET Core API) | 48,44 € |
| Azure Static Web App | Ospita il frontend statico (HTML, CSS, JS) | 0 € |
| Microsoft Entra ID | Sistema di sicurezza e gestione degli accessi | 110,16 € |
| Azure DevOps | Deploy continuo (CI/CD) integrato con GitHub | 0 € |
| Azure Private Link | Sicurezza nelle comunicazioni interne all'applicazione | 8,23 € |
| Azure SQL Database | Archiviazione relazionale di utenti, prodotti e carrelli | 4,69 € |
| Azure Key Vault | Gestione sicura di chiavi JWT e stringhe di connessione (no password utenti) | 0 € |
| Azure Storage | Per eventuali immagini, file statici o asset di dimensioni notevoli da caricare in futuro | 17,96 € |
| Azure Monitor + Insights | Logging, tracciamento performance e alert automatici | 1,65 € |

Tabella 1: Tabella riassuntiva dei costi per i componenti Azure