



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DIPARTIMENTO DI INFORMATICA - SCIENZA e INGEGNERIA

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA
INFORMATICA

Analisi, progettazione e distribuzione in cloud di applicativo per l'organizzazione di eventi condivisi

Relatore:
Chiar.mo Prof.
Michele Colajanni

Presentata da:
Giacomo Romanini

Sessione Luglio 2025

Anno Accademico 2025/2026

Abstract

Lo sviluppo di un applicativo multiplatforma diretto all'organizzazione di eventi condivisi, caratterizzato in particolare dalla condivisione multimediale in tempo reale, richiede opportune capacità di scalabilità, atte a garantire una risposta efficace anche con alti volumi di richieste, offrendo prestazioni ottimali. Le tecnologie cloud, con la loro disponibilità pressoché illimitata di risorse e la completa e continua garanzia di manutenzione, offrono l'architettura ideale per il supporto di simili progetti, anche con fondi limitati.

Tuttavia, l'integrazione tra la logica applicativa e i molteplici servizi cloud, assieme alla gestione delle loro interazioni reciproche, comporta sfide specifiche, in particolare legate all'ottimizzazione di tutte le risorse. L'individuazione e la selezione delle soluzioni tecnologiche più adatte per ogni funzionalità, così come l'adozione delle migliori pratiche progettuali, devono procedere parallelamente con lo sviluppo del codice, al fine di sfruttare efficacemente le potenzialità offerte.

In tale prospettiva, questa tesi illustra le scelte progettuali e implementative adottate nello sviluppo di Wyd, applicativo per la condivisione di eventi, evidenziando l'impatto dell'integrazione delle risorse cloud sul risultato finale.

Indice

| | |
|---|----|
| Introduzione | 1 |
| Organizzazione dei capitoli | 3 |
| Capitolo 1 | 5 |
| Individuazione dei requisiti e dei casi d'uso | 6 |
| I requisiti e il vocabolario | 6 |
| Casi d'uso | 9 |
| Requisiti di sicurezza | 15 |
| Bibliografia | 21 |

Introduzione

In un contesto sociale sempre più connesso, la crescente quantità di contatti, la rapidità delle comunicazioni e l'accesso universale alle informazioni rendono la ricerca, l'organizzazione e la partecipazione a eventi estremamente facile, ma al contempo generano un ambiente frenetico e spesso dispersivo.

Risulta infatti difficile seguire tutte le opportunità a cui si potrebbe partecipare, considerando le numerose occasioni che si presentano quotidianamente. Basti pensare, ad esempio, alle riunioni di lavoro, alle serate con amici, agli appuntamenti informali per un caffè, ma anche a eventi più strutturati come fiere, convention aziendali, concerti, partite sportive o mostre di artisti che visitano occasionalmente la città.

Questi eventi possono sovrapporsi, causando dimenticanze o conflitti di pianificazione, con il rischio di delusione o frustrazione. Quando si è invitati a un evento, può capitare di essere già impegnati, o di trovarsi in attesa di una conferma da parte di altri contatti. In questi casi, la gestione degli impegni diventa complessa: spesso si conferma la partecipazione senza considerare possibili sovrapposizioni, o dimenticandosi, per poi dover scegliere e disdire all'ultimo momento.

D'altra parte, anche quando si desidera proporre un evento, la ricerca di un'attività interessante può diventare un compito arduo, con la necessità di consultare numerosi profili social di locali e attività, senza avere inoltre la certezza che gli altri siano disponibili. Tali problemi si acuiscono ulteriormente quando si tratta di organizzare eventi di gruppo, dove bisogna allineare gli impegni di più persone.

In questo contesto, emergono la necessità e l'opportunità di sviluppare uno strumento che semplifichi la proposta e la gestione degli eventi, separando il momento della proposta da quello della conferma di partecipazione. In tal modo, gli utenti possono valutare la disponibilità degli altri prima di impegnarsi definitivamente, facilitando in contemporanea sia l'invito che la partecipazione.

In risposta a tali esigenze si è deciso di creare Wyd, un'applicazione che permette agli utenti di organizzare i propri impegni, siano essi confermati oppure proposti. Essa permette anche di rendere più intuitiva la ricerca di eventi attraverso la creazione di uno spazio virtuale centralizzato dove gli utenti possano pubblicare e consultare tutti gli eventi disponibili, diminuendo l'eventualità di perderne qualcuno. La funzionalità chiave di questo progetto si fonda sull'idea di affiancare alla tradizionale agenda degli impegni confermati un calendario separato, che mostri tutti gli eventi a cui si potrebbe partecipare.

Una volta confermata la partecipazione a un evento, questo verrà spostato automaticamente nell'agenda personale dell'utente. Gli eventi creati potranno essere condivisi con persone o gruppi, permettendo di visualizzare le conferme di partecipazione. Considerando l'importanza della condivisione di contenuti multimediali, questo progetto prevede la possibilità di condividere foto e video con tutti i partecipanti all'evento, attraverso la generazione di link per applicazioni esterne o grazie all'ausilio di gruppi di profili. Al termine dell'evento, l'applicazione carica automaticamente le foto scattate durante l'evento, per allegarle a seguito della conferma dell'utente.



Figura 1: Il logo di Wyd

La realizzazione di un progetto come Wyd implica la risoluzione e la gestione di diverse problematiche tecniche. In primo luogo, la stabilità del programma deve essere garantita da un'infrastruttura affidabile e scalabile. La persistenza deve essere modellata per fornire alte prestazioni sia in lettura che in scrittura indipendentemente dalla quantità delle richieste, rimanendo però aggiornata e coerente. La funzionalità di condivisione degli eventi richiede inoltre l'aggiornamento in tempo reale verso tutti gli utenti coinvolti. Infine, il caricamento e il salvataggio delle foto aggiungono la necessità di gestire richieste di archiviazione di dimensioni significative.

Descrizione dei capitoli

L'elaborato è suddiviso in cinque capitoli.

Nel primo capitolo si affronta la fase di analisi delle funzionalità, durante la quale, partendo dall'idea astratta iniziale, si definiscono i requisiti e le necessità del sistema, per poi creare la struttura generale ad alto livello del progetto.

Nel secondo capitolo si affrontano le principali scelte architettureali e di sviluppo che hanno portato a definire la struttura centrale dell'applicazione.

Il terzo capitolo osserva lo studio effettuato per gestire la memoria, in quanto fattore che più incide sulle prestazioni. Particolare attenzione è stata dedicata, infatti, a determinare le tecnologie e i metodi che meglio corrispondono alle esigenze derivate dal salvataggio e dall'interazione logica degli elementi.

Il quarto capitolo si concentra sulle scelte implementative adottate per l'inserimento delle funzionalità legate alla gestione delle immagini che, oltre a introdurre problematiche impattanti sia sulle dimensioni delle richieste che sull'integrazione con la persistenza, richiedono l'automatizzazione del recupero delle immagini.

Infine, nel quinto capitolo, verranno analizzati e discussi i risultati ottenuti testando il sistema.

Capitolo 1

La realizzazione di qualunque prodotto software inizia da una fase in cui, partendo dall'abstract del progetto, si analizzano i requisiti e le funzionalità da realizzare. L'obiettivo è arrivare a una definizione delle proprietà e del comportamento desiderato nell'applicazione che sia concisa e condivisa col cliente, senza però entrare nel merito delle scelte implementative. Solo a quel punto si può procedere con lo sviluppo del programma vero e proprio.

L'abstract di Wyd è il seguente:

Wyd è un'applicazione che permette ai clienti di organizzare i propri impegni, siano essi confermati oppure proposti. Mette a disposizione due calendari, il primo con gli eventi in cui l'utente è convinto di partecipare, il secondo in cui vengono riuniti gli eventi a cui l'utente è stato invitato ma non ha ancora dato conferma. L'utente ha la possibilità di creare, modificare, confermare o disdire un evento, ma anche condividerlo con altri o allegarci foto.

La condivisione di un evento può avvenire con applicazioni esterne tramite la generazione di un link o grazie all'ausilio di gruppi di profili. Inoltre, al termine di un evento, l'applicazione carica automaticamente le foto scattate durante l'evento, per allegarle a seguito della conferma dell'utente. L'utente può infatti cercare altri profili e creare gruppi con i profili trovati. Tutta l'interazione avviene tramite l'utilizzo di profili, che permettono di suddividere semanticamente gli eventi e le relazioni.

1.1 Individuazione dei requisiti e dei casi d'uso

Lo studio dell'abstract del progetto porta all'individuazione e alla descrizione delle sue caratteristiche essenziali. In particolare, si distinguono i requisiti e i casi d'uso. I requisiti formalizzano le funzionalità che l'applicazione deve fornire, sintetizzando e schematizzando le parti che descrivono del prodotto. I casi d'uso descrivono invece le interazioni previste tra l'utente e il sistema, suddividendo le funzionalità in azioni elementari.

1.1.1 I requisiti e il vocabolario

I requisiti devono risultare chiari e precisi per permettere di procedere alle fasi successive in maniera corretta e trasparente. Ogni requisito deve essere breve e puntuale, limitato a un solo particolare desiderata, focalizzando una necessità specifica.

Si suddividono in funzionali o non funzionali in base alle loro caratteristiche. I requisiti funzionali descrivono le funzionalità che il sistema deve fornire, mentre i requisiti non funzionali illustrano le caratteristiche che il sistema deve soddisfare per essere considerato valido.

Da una prima analisi dell'abstract si evincono immediatamente i principali requisiti funzionali, che riguardano in generale l'esperienza utente nelle sue parti principali, dalla visualizzazione nelle schermate all'inserimento di dati e foto. Si aggiungono quindi le funzionalità dettate da necessità derivate, quali l'esigenza di autenticare l'utente o il bisogno di gestire i profili e i gruppi. Infine si analizzano le specifiche non funzionali, che sono raramente incluse nel testo ma che descrivono le caratteristiche performative e di sicurezza ritenute essenziali per il successo desiderato.

Vengono quindi introdotti i desiderata relativi all'esperienza utente, evidenziando l'intuitività e la reattività dell'applicazione, che richiedono di conseguenza velocità nel recuperare i dati. Inoltre, essendo Wyd pensata per interagire con migliaia di utenti, in simultanea e interconnessi tra loro, le caratteristiche di scalabilità vengono individuate e introdotte fin da subito. La sicurezza del sistema viene trattata in seguito, in quanto necessita di un'apposita analisi approfondita, a esclusione dell'autenticazione che impatta direttamente sull'utente.

| ID | Requisiti | Tipo |
|------|--|----------------|
| R1F | Registrazione di un account tramite l'interfaccia web | Funzionale |
| R2F | Identificazione attraverso mail univoca e password di almeno sei caratteri | Funzionale |
| R3F | Visualizzazione degli eventi confermati | Funzionale |
| R4F | Visualizzazione degli eventi proposti | Funzionale |
| R5F | Creazione di un evento impostando almeno la data d'inizio e quella di fine | Funzionale |
| R6F | La data di fine deve essere successiva alla data d'inizio | Funzionale |
| R7F | Modifica di un evento | Funzionale |
| R8F | La conferma di un evento lo sposta negli eventi confermati | Funzionale |
| R9F | La disdetta di un evento lo sposta negli eventi proposti | Funzionale |
| R10F | Caricamento delle foto di un evento | Funzionale |
| R11F | Condivisione tramite link | Funzionale |
| R12F | Condivisione tramite gruppo o ad altri profili | Funzionale |
| R13F | Ricerca automatica delle foto sul dispositivo mobile | Funzionale |
| R14F | Conferma delle foto | Funzionale |
| R15F | Ricerca di altri profili | Funzionale |
| R16F | Creazione di un gruppo da due o più profili | Funzionale |
| R17F | Visualizzazione dei profili collegati | Funzionale |
| R18F | Creazione di un nuovo profilo | Funzionale |
| R19F | Cambio del profilo attualmente in uso | Funzionale |
| R20F | Aggiornamento in tempo reale delle modifiche agli eventi | Funzionale |
| R1NF | Per interagire l'utente deve essere autenticato | Non Funzionale |
| R2NF | Velocità di richiesta iniziale dei dati | Non Funzionale |
| R3NF | Semplicità e fluidità dell'interfaccia grafica | Non Funzionale |
| R4NF | Velocità in lettura e scrittura dei dati | Non Funzionale |
| R5NF | Velocità nella ricerca dei profili | Non Funzionale |
| R6NF | Scalabilità delle richieste | Non Funzionale |

Tabella 1.1: Tabella dei requisiti di Wyd

Alla tabella dei requisiti si affianca quella del vocabolario, definendo i termini utilizzati nel progetto per allinearli definitivamente alle volontà del cliente. Questo consentirà, quando in seguito verranno citati, di evitare possibili ambiguità derivate dall'uso comune dei termini. Si specifica cosa si intende quindi per utente, profilo e gruppo, ma anche, analizzando i requisiti circoscritti in precedenza, evento confermato e proposto, o ancora email e password.

| Voce | Definizione | Sinonimi |
|-------------------|--|-----------------------------------|
| Account | combinazione di mail e password che identifica un utente | |
| Utente | Persona che utilizza l'applicazione | |
| Profilo | Entità logica che raggruppa eventi e interazioni | |
| Profili collegati | Profili a cui l'utente può avere accesso | |
| Gruppo | Insieme di profili | |
| Evento | Azione(o previsione di azione) con una durata nel tempo | |
| Data e ora evento | Indicazione temporale del momento in cui avverrà l'azione | |
| Evento confermato | Evento a cui il profilo ha dato conferma di partecipazione | |
| Evento proposto | Evento a cui il profilo non ha dato conferma di partecipazione | Evento disdetto, evento condiviso |
| Email | Indirizzo di posta elettronica del cliente utilizzata anche per l'autenticazione | |
| Password | Codice alfanumerico di almeno otto caratteri | |
| Credenziali | Insieme composto da email e password necessari per accedere al sistema | |

Tabella 1.2: Vocabolario di Wyd

1.1.2 Casi d'uso

I casi d'uso descrivono le interazioni tra gli attori e il sistema, suddividendo le funzionalità in azioni elementari. Si definiscono attori tutti gli elementi che compiono una parte attiva nei confronti del programma. Ogni attore può interagire con uno o più casi d'uso, e ogni caso d'uso può essere relazionato con altri, definendo la loro relazione.

I casi d'uso possono essere collegati tra loro tramite rapporti d'inclusione o estensione. Si dice che un caso include un altro se contiene il suo comportamento. Si dice invece che un caso d'uso ne estende un altro se il suo comportamento può essere inserito all'interno del secondo.

Per ogni azione descritta nei requisiti, intuibile dal contesto o necessaria per il soddisfacimento dei requisiti, viene introdotto un nuovo caso d'uso. Ad esempio, la creazione o la condivisione di un evento vengono estratti direttamente dalla descrizione del progetto, così come la ricerca automatica delle immagini. L'eliminazione di un evento, la registrazione di un utente o l'aggiunta di un profilo a un gruppo, per quanto non espressamente elencati tra i requisiti, sono dedotti dal contesto. L'aggiornamento da un server esterno viene introdotto per realizzare la modifica in tempo reale degli eventi.

I casi d'uso così individuati si raggruppano attorno a tre principali. `VisualizzaEvento` permette di visualizzare i dettagli dell'evento, ma anche di modificarli e di eseguire tutte le azioni relative. `GestioneGruppi` racchiude tutte le esigenze che riguardano i contatti e i gruppi, come la visualizzazione dei gruppi, la ricerca dei profili e l'aggiunta di un profilo a un gruppo. Infine `GestioneProfili` permette il controllo dei profili collegati all'utente, visualizzandoli e dando la possibilità di cambiare il profilo corrente.

Si distinguono rispetto ai precedenti casi d'uso e alle loro estensioni altre funzionalità il cui scopo non riguarda nessuno dei tre argomenti, o il cui funzionamento sia scorrelato. Login e Registrazione, ad esempio, vedono il loro utilizzo in maniera trasversale rispetto al resto dell'applicazione, essendo necessari per il funzionamento di tutti ma distanti da un punto di vista logico. `AggiornaEvento` e `RecuperaImmagini` eseguono indipendentemente dall'interazione utente, apportando modifiche automaticamente in base ad attori esterni.

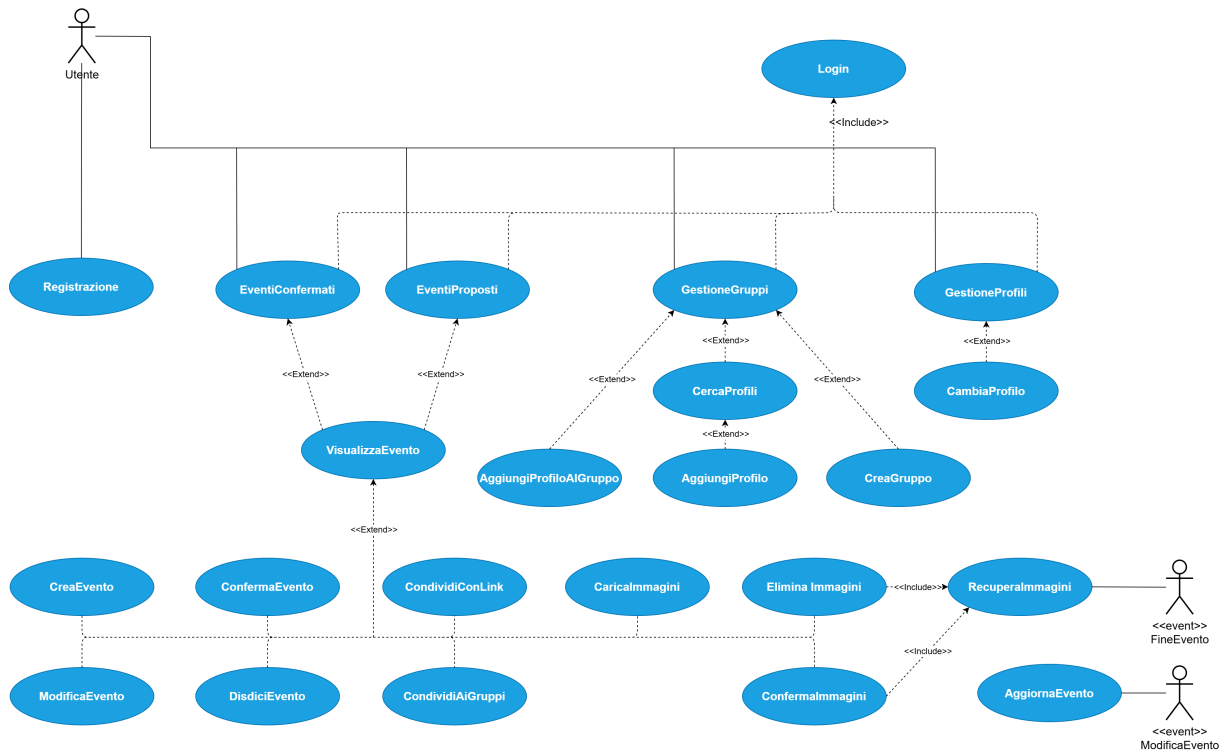


Figura 1.1: Diagramma dei casi d'uso

Per ogni caso d'uso viene poi identificato uno scenario di utilizzo, che chiarifica il contesto, il comportamento e i punti critici dell'utilizzo. Lo scenario ha il solo compito di mostrare il comportamento desiderato, senza scendere quindi in dettagli o complessità progettuali. Dallo scenario risulta non si può quindi dedurre la difficoltà introdotta dall'implementazione del caso d'uso, ma è il tassello da cui si stabiliranno poi la coordinazione e l'interazione delle varie parti del programma.

Si riportano gli scenari di utilizzo per i principali casi d'uso di Wyd, ovvero quelli che più andranno a impattare sulla struttura e sulle esigenze del progetto.

Lo scenario di registrazione vede la responsabilità, oltre che di creare un account, di collegare un profilo all'utente. Questa separazione consente di avere una struttura gerarchica che permette di associare più profili a un unico utente, che può così in seguito crearne o unirne di nuovi.

| | |
|---------------------------------|--|
| Titolo | Registrazione |
| Descrizione | L'utente si registra al servizio |
| Attori | Utente |
| Relazioni | |
| Precondizioni | |
| Post condizioni | L'utente è registrato nel sistema e può interagire con il resto dell'applicazione |
| Scenario principale | 1.L'utente accede alla schermata di registrazione 2. L'utente inserisce email e password 3. Il sistema crea un account con le credenziali inserite, associando un utente e un primo profilo 4. L'utente termina la registrazione, se avvenuta con successo viene reindirizzato alla pagina principale |
| Scenari Alternativi | Il sistema verifica che è già presente un account con la mail inserita, quindi procede con la procedura di login normale. |
| Requisiti non funzionali | Per interagire l'utente deve essere autenticato Velocità in lettura e scrittura dei dati |
| Punti aperti | |

Tabella 1.3: Scenario di registrazione

A seguito della modifica di un evento, che implica il salvataggio dei suoi nuovi dati, viene chiesto l'aggiornamento in tempo reale verso tutti i dispositivi di tutti gli utenti a cui l'evento è stato condiviso. Inoltre, sarà necessario inserire un controllo per evitare che due richieste simultanee causino conflitti.

| | |
|----------------------|--|
| Titolo | ModificaEvento |
| Descrizione | Salva le modifiche a un evento |
| Attori | Utente |
| Relazioni | VisualizzaEvento |
| Precondizioni | L'evento esiste e sono stati modificati dei dati |

| | |
|---------------------------------|--|
| Post condizioni | Le modifiche vengono salvate e propagate a tutti i profili collegati |
| Scenario Principale | <ol style="list-style-type: none"> 1. VisualizzaEvento 2. Il sistema controlla che i dati modificati siano corretti 3. I cambiamenti vengono salvati 4. Tutti i dispositivi collegati ai profili collegati all'evento visualizzano le immagini |
| Scenari Alternativi | 2. Se i dati risultano sbagliati, il sistema notifica l'utente originario indicando l'errore |
| Requisiti non funzionali | Velocità in lettura e scrittura dei dati Scalabilità delle richieste |
| Punti aperti | Le modifiche all'evento devono essere consistenti, soprattutto in caso di richieste simultanee |

Tabella 1.4: Scenario della modifica di un evento

Il salvataggio delle immagini è un'operazione di particolare importanza vista la sua rilevanza nel coinvolgimento degli utenti nell'utilizzo delle funzionalità centrali dell'applicazione, e quindi nel successo del progetto. Oltre a mostrare un'interfaccia intuitiva, il sistema deve essere in grado di gestire queste particolari richieste di caricamento, che generalmente necessitano di più tempo e memoria. Prevedendo che la maggior parte di queste avvenga in seguito alla conclusione dell'evento, la probabilità che più richieste simultanee vertano sullo stesso evento risulta elevata, creando la necessità di una gestione parallela di modifiche concorrenti.

| | |
|----------------------|---|
| Titolo | CaricaImmagini |
| Descrizione | Permette all'utente di selezionare immagini da collegare all'evento, salvandole |
| Attori | Utente |
| Relazioni | VisualizzaEvento |
| Precondizioni | L'evento esiste |

| | |
|---------------------------------|--|
| Post condizioni | Le immagini vengono salvate e propagate a tutti i profili collegati |
| Scenario Principale | <ol style="list-style-type: none"> 1. VisualizzaEvento 2. L'utente seleziona le immagini che vuole caricare 3. Le immagini vengono salvate 4. Tutti i dispositivi relativi ai profili collegati all'evento visualizzano le immagini |
| Scenari Alternativi | <p>Scenario alternativo A:</p> <ol style="list-style-type: none"> 3. Almeno una delle immagini crea problemi di lettura, l'utente viene notificato e può riprovare a caricare le immagini <p>Scenario alternativo B:</p> <ol style="list-style-type: none"> 3. Solo una parte delle immagini vengono salvate, altre comportano errori 4. L'utente viene notificato dell'errore e può riprovare a caricare le immagini 5. Tutti i dispositivi relativi ai profili collegati all'evento visualizzano le immagini <p>Scenario alternativo C:</p> <ol style="list-style-type: none"> 3. Nessuna immagine risulta salvata con successo 4. L'utente viene notificato dell'errore e può riprovare |
| Requisiti non funzionali | <p>Semplicità e fluidità dell'interfaccia grafica</p> <p>Velocità in lettura e scrittura dei dati</p> <p>Scalabilità delle richieste</p> |
| Punti aperti | |

Tabella 1.5: Scenario del caricamento delle immagini

L'azione di recupero delle immagini facilita l'utilizzo dell'applicazione, automatizzando il procedimento di ricerca delle immagini, riducendo l'interazione utente alla sola conferma. Una sua corretta implementazione ne fa apprezzare l'utilità, con una significativa influenza sull'esperienza utente. Richiede però la pianificazione e l'automazione del processo

di cernita di dati, con effetti sull'analisi tecnologica, sui processi in background e sulla gestione della memoria locale.

| | |
|---------------------------------|--|
| Titolo | RecuperaImmagini |
| Descrizione | L'applicazione controlla la galleria e salva in locale le foto scattate durante l'evento |
| Attori | FineEvento |
| Relazioni | EliminaImmagini, ConfermaImmagini |
| Precondizioni | L'evento esiste ed è concluso l'utente ha dato il permesso all'accesso alla galleria |
| Post condizioni | Le immagini sono salvate in locale e l'utente viene notificato |
| Scenario Principale | <ol style="list-style-type: none"> 1. Il sistema attende la fine dell'evento 2. Il sistema controlla la galleria per trovare le immagini scattate nell'arco temporale dell'evento 3. Se ci sono immagini, vengono salvate in locale e l'utente viene notificato |
| Scenari Alternativi | |
| Requisiti non funzionali | Velocità in lettura e scrittura dei dati |
| Punti aperti | L'implementazione dipende dal dispositivo su cui viene eseguita l'applicazione, alcuni dispositivi potrebbero non permetterne l'esecuzione |

Tabella 1.6: Scenario di recupero delle immagini dal dispositivo dell'utente

1.1.3 Requisiti di sicurezza

Ogni sistema è esposto a vulnerabilità che impattano sul corretto funzionamento dell'applicazione e possono comportare disservizi in base alla loro rilevanza nel funzionamento del sistema. La rilevazione dei rischi e la successiva definizione dei requisiti necessari per evitare o minimizzare i danni è alla base della strategia di sicurezza.

La definizione dei requisiti di sicurezza deriva dall'analisi del rischio. L'analisi del rischio individua i possibili vettori di attacco e serve a orientare le risorse dove più necessario, tramite la valutazione dei beni, l'identificazione delle minacce e l'individuazione dei punti deboli delle tecnologie di cui si prevede l'utilizzo.

La valutazione dei beni determina i componenti fondamentali da proteggere, risaltandone il valore e l'esposizione relativa. Questo permette di stabilire le priorità dei componenti sui cui concentrare le attenzioni. In particolare, Wyd non prevede altri sistemi diversi dai comuni sistemi informatici, ma i valori principali da proteggere risiedono nei dati degli utenti.

| Bene | Valore | Esposizione |
|-----------------------------------|--|--|
| Sistema Informativo | Alto. Fondamentale per il funzionamento del servizio | Alta. Perdita finanziaria e di immagine |
| Informazioni dei clienti | Alto. Informazioni personali | Alta. Perdita di immagine dovuta alla divulgazione di dati sensibili |
| Informazioni relativi agli eventi | Medio-alto, necessari per offrire il servizio e contenenti informazioni personali e potenzialmente riservate | Molto Alta. Perdita di immagine possibile con la divulgazione dei dati relativi ai clienti |
| Dati dei gruppi | Medio. Necessario per condividere gli eventi | Alta. Perdita di immagine |

Tabella 1.7: Valutazione dei beni

La tabella delle minacce individua gli attacchi principali previsti che possono avvenire sul sistema. Esamina la loro probabilità, le azioni richieste per controllarli e il costo di realizzazione delle contromisure necessarie. Fornisce quindi una prima analisi sulle necessità implementative.

Tutte le risorse di Wyd sono orientate agli utenti, in particolare al mantenimento e alla distribuzione dei loro dati. Per questo motivo le minacce sono relative alla confidenzialità dei dati o all'interruzione del servizio.

| Minaccia | Probab. | Controllo | Fattibilità |
|---|----------------|---|---|
| Furto credenziali utente | Alta | Controllo sulla sicurezza della password - Log delle operazioni, autenticazione a due fattori | Costo implementativo medio |
| Alterazione o intercettazione delle comunicazioni | Alta | Utilizzo di un canale sicuro - Log delle operazioni, autenticazione integrata nel messaggio | Basso costo di realizzazione con determinati protocolli |
| Accesso non autorizzato al database | Bassa | Accesso da macchine sicure - Log di tutte le operazioni | Basso costo di realizzazione, il server deve essere ben custodito |
| DoS | Bassa | Controllo e limitazione delle richieste | Media complessità di implementazione |
| Saturazione del database | Bassa | <ol style="list-style-type: none"> 1. Limitazione delle richieste in un dato intervallo di tempo. 2. Limitazione della grandezza delle richieste singole 3. Limitazione della grandezza richiesta dallo stesso utente in un dato intervallo di tempo | Media complessità d'implementazione |

Tabella 1.8: Tabella delle minacce

L'analisi tecnologica della sicurezza entra nel merito delle tecnologie di cui si prevede l'utilizzo. Per ognuna esamina i punti deboli e i limiti intrinseci, producendo un quadro delle particolarità su cui porre maggiore attenzione.

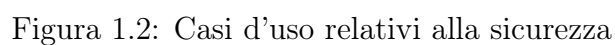
Wyd prevede principalmente la comunicazione tra le applicazioni utenti e un server centrale, per cui la tecnologia da analizzare si concentra sull'architettura ma soprattutto sulle comunicazioni e sull'autenticazione.

| Tecnologia | Vulnerabilità |
|-----------------------------------|--|
| Autenticazione email/password | <ul style="list-style-type: none"> • Utente rivela volontariamente la password • Utente rivela la password con un attacco di ingegneria sociale • Password banali |
| Cifratura comunicazioni | <ul style="list-style-type: none"> • In caso di cifratura simmetrica particolare attenzione va alla lunghezza delle chiavi ed alla loro memorizzazione |
| Architettura Client/Server | <ul style="list-style-type: none"> • DoS • Man in the Middle • Sniffing delle comunicazioni |
| Connessione Server/Persistenza | <ul style="list-style-type: none"> • Limite massimo di connessioni contemporanee • Saturazione del Database |

Tabella 1.9: Analisi tecnologica della sicurezza

A questo punto si prevedono i principali attori malevoli e i relativi casi d'uso, per poi definire i requisiti su cui si baseranno le contromisure necessarie. I casi d'uso sono molto simili alle minacce individuate in precedenza, ma vengono creati in base alla modalità di attacco, più che alla tipologia. A ogni caso d'uso malevolo ne viene corrisposto un altro che ne comporta la mitigazione. Si integrano quindi con i casi d'uso dell'applicazione, evidenziando i punti e la loro applicazione.

Tramite la saturazione del database l'attaccante riesce a inserire quantità importanti di dati, che può comportare un rallentamento dell'applicazione temporaneo o permanente, in base alla configurazione dell'attacco. Questo è particolarmente efficace dal momento in cui si possono inserire delle foto. Per mitigare questo rischio si aggiunge un caso d'uso relativo al controllo delle dimensioni delle richieste.



La confidenzialità e l'integrità dei dati sono minacciati dal furto di credenziali, che permetterebbe a un utente d'identificarsi come qualcun altro; lo sniffing mina la riservatezza delle comunicazioni, che verrebbero così intercettate; infine tramite man in the middle un attore malevolo ha la possibilità di modificare le richieste ingannando entrambi i lati della conversazione. Si introducono i casi d'uso relativi al controllo delle credenziali, che aumenta la difficoltà di un possibile furto; la riservatezza, che permette di nascondere le comunicazioni alle parti non interessate, e l'integrità, che consente l'individuazione di eventuale manipolazione dei messaggi.

Visti i costi e appurate le risorse a disposizione sono stati quindi identificati i seguenti requisiti inerenti alla protezione dei dati e delle funzionalità di Wyd:

1. Implementare un sistema di log per tracciare tutti i messaggi tra i client e i server, inclusi gli accessi, le richieste di prenotazione, di conferma, di sospensione e di invio e ricezione di dati
2. I dati salvati devono essere protetti da un attaccante che abbia accesso al sistema, prendendo misure di sicurezza fisica, eventualmente cifrando i dati
3. I dati inviati tra le parti remote devono essere protetti, utilizzando la cifratura dei dati
4. Tutte le azioni avvenute sul sistema devono essere tracciate tramite un sistema di log.
5. Il sistema deve essere resistente a un alto numero di richieste contemporanee
6. La dimensione delle richieste non deve superare una determinata soglia

La visione e l'analisi dei log verrà gestita con uno strumento esterno, accessibile solo al personale autorizzato.

| ID | Requisiti | Tipo |
|------|---|------------|
| R21F | Implementazione di un sistema di log per tracciare tutti i messaggi tra i client e i server | Funzionale |
| R22F | Le richieste non devono superare una certa dimensione | Funzionale |

| ID | Requisiti | Tipo |
|-----------|--|----------------|
| R7NF | I dati salvati devono essere protetti da un attaccante che abbia accesso al sistema, prendendo misure di sicurezza fisica, eventualmente cifrando i dati | Non Funzionale |
| R8NF | I dati inviati tra le parti remote devono essere protetti, utilizzando la cifratura dei dati | Non Funzionale |
| R9NF | Il sistema deve essere resistente ad un alto numero di richieste contemporanee | Non funzionale |

Tabella 1.10: Requisiti di sicurezza

Bibliografia

Amazon Web Services. Serverless Computing - AWS Lambda Pricing. Amazon Web Services, a. URL <https://aws.amazon.com/lambda/pricing/>.

Amazon Web Services. Amazon DynamoDB Pricing. Amazon Web Services, b. URL <https://aws.amazon.com/dynamodb/pricing/>.

Daniel Barcelona-Pons and Pedro García-López. Benchmarking Parallelism in FaaS Platforms. *Future Generation Computer Systems*, 124:268–84, November 2021. doi: 10.1016/j.future.2021.06.005.

Awel Eshetu Fentaw. Cross Platform Mobile Application Development: A Comparison Study of React Native Vs Flutter. *Jurnal University of Jyväskylä*, 27791:37–38, 2020.

Firebase. Authentication. Firebase Documentation. URL <https://firebase.google.com/docs/auth>.

Flutter. App Architecture. Flutter Documentation. URL <https://docs.flutter.dev/app-architecture>.

Sebastian Gajek, Mark Manulis, Olivier Pereira, Ahmad Reza Sadeghi, and Jörg Schwenk. Universally Composable Security Analysis of TLS. In *Lecture Notes in Computer Science [Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics]*, volume 5324 LNCS, pages 313–27. Springer Verlag. doi: 10.1007/978-3-540-88733-1_22.

Google Cloud. Cloud Firestore Pricing. Google Cloud, a. URL <https://cloud.google.com/firestore/pricing>.

Google Cloud. Cloud Run pricing. Google Cloud, b. URL <https://cloud.google.com/run/pricing/>.

- Simon Hedlund and Ylva Rasmusson Wright. Cross-platform Frameworks Comparison: Android Applications in a Cross-platform Environment, Xamarin Vs Flutter. Bachelor's thesis, Blekinge Institute of Technology, 2021.
- ISO. ISO/IEC/IEEE International Standard - Systems and Software Engineering – Life Cycle Processes – Requirements Engineering. ISO/IEC/IEEE 29148:2018(E), 2018.
- B. Jose and S. Abraham. Exploring the merits of nosql: A study based on mongodb. In *2017 International Conference on Networks and Advances in Computational Technologies, NetACT 2017*, pages 266–271. Institute of Electrical and Electronics Engineers Inc., 2017. doi: 10.1109/NETACT.2017.8076778.
- Benymol Jose and Sajimon Abraham. Performance Analysis of NoSQL and Relational Databases with MongoDB and MySQL. *Materials Today: Proceedings*, 24:2036–43, 2019. doi: 10.1016/j.matpr.2020.03.634.
- Benymol Jose and Sajimon Abraham. Intelligent Processing of Unstructured Textual Data in Document Based NoSQL Databases. *Materials Today: Proceedings*, 80:1777–85, January 2023. doi: 10.1016/j.matpr.2021.05.605.
- Microsoft Azure. Azure Cosmos DB. Azure, a. URL <https://azure.microsoft.com/en-us/products/cosmos-db>.
- Microsoft Azure. Azure Functions pricing. Azure, b. URL <https://azure.microsoft.com/en-us/pricing/details/functions>.
- Microsoft Azure. Azure Key Vault. Azure, c. URL <https://azure.microsoft.com/en-us/products/key-vault>.
- Microsoft Azure. Azure pricing calculator. Azure, d. URL <https://azure.microsoft.com/en-us/pricing/calculator/>.
- Microsoft Azure. Azure Web PubSub. Azure, e. URL <https://azure.microsoft.com/en-us/products/web-pubsub>.
- Microsoft Learn. Application Insights overview. Microsoft Learn, a. URL <https://learn.microsoft.com/en-us/azure/azure-monitor/app/app-insights-overview>.

Microsoft Learn. Choose between Azure messaging services - Azure Service Bus, Azure Event Hubs, and Azure Queue Storage. Microsoft Learn, b. URL <https://learn.microsoft.com/en-us/azure/service-bus-messaging/compare-messaging-services>.

Microsoft Learn. Introduction to Azure Queue Storage. Microsoft Learn, c. URL <https://learn.microsoft.com/en-us/azure/storage/queues/storage-queues-introduction>.

Microsoft Learn. What is Azure Service Bus? Introduction to Azure Service Bus messaging. Microsoft Learn, d. URL <https://learn.microsoft.com/en-us/azure/service-bus-messaging/service-bus-messaging-overview>.

Microsoft Learn. Azure Functions .NET class library development. Microsoft Learn, e. URL <https://learn.microsoft.com/en-us/azure/azure-functions/functions-dotnet-class-library>.

Microsoft Learn. Dependency injection in .NET Azure Functions. Microsoft Learn, f. URL <https://learn.microsoft.com/en-us/azure/azure-functions/functions-dotnet-dependency-injection>.

Microsoft Learn. Durable Functions overview. Microsoft Learn, g. URL <https://learn.microsoft.com/en-us/azure/azure-functions/durable/durable-functions-overview>.

Microsoft Learn. Develop .NET isolated process Azure Functions. Microsoft Learn, h. URL <https://learn.microsoft.com/en-us/azure/azure-functions/dotnet-isolated-process-guide>.

Microsoft Learn. Provision throughput on Azure Cosmos DB for NoSQL resources. Microsoft Learn, i. URL <https://learn.microsoft.com/en-gb/azure/cosmos-db/provision-throughput-autoscale>.

Microsoft Learn. Reliability for Azure Blob Storage. Microsoft Learn, j. URL <https://learn.microsoft.com/en-us/azure/well-architected/service-guides/azure-blob-storage>.

- Microsoft Learn. Reliability in Azure Cosmos DB for NoSQL. Microsoft Learn, k. URL <https://learn.microsoft.com/en-us/azure/reliability/reliability-cosmos-db-nosql>.
- Microsoft Learn. Reliability in Azure Cosmos DB for NoSQL. Microsoft Learn, l. URL <https://learn.microsoft.com/en-us/azure/reliability/reliability-cosmos-db-nosql>.
- Microsoft Learn. Serverless capacity mode in Azure Cosmos DB for NoSQL. Microsoft Learn, m. URL <https://learn.microsoft.com/en-gb/azure/cosmos-db/throughput-serverless>.
- Microsoft Learn. Grant limited access to Azure Storage resources using shared access signatures (SAS). Microsoft Learn, n. URL <https://learn.microsoft.com/en-us/azure/storage/common/storage-sas-overview>.
- NIST. NIST Special Publication 800-30 Revision 1 - Guide for Conducting Risk Assessments. Technical Report September, 2012. NIST Guide for Conducting Risk Assessments.
- Object Management Group Inc (OMG). Unified Modeling Language (UML) Specification Version 2.5.1. Technical report, Object Management Group, Inc. (OMG), 2017. URL <https://www.omg.org/spec/UML/2.5.1>.
- Rabi Prasad Padhy, Manas Ranjan Patra, and Suresh Chandra Satapathy. RDBMS to NoSQL: Reviewing Some Next-Generation Non-Relational Database's. *International Journal of Advanced Engineering Sciences and Technologies*, 11(11):15–30, 2011.
- Moneer Rifai. Serverless showdown: AWS Lambda vs Azure Functions vs Google Cloud Functions. Pluralsight Blog, June 2023. URL <https://www.pluralsight.com/resources/blog/cloud/serverless-showdown-aws-lambda-vs-azure-functions-vs-google-cloud-functions>.
- Mohammad Shahrads, Jonathan Balkind, and David Wentzlaff. Architectural Implications of Function-as-a-Service Computing. In *Proceedings of the Annual International Symposium on Microarchitecture, MICRO*, pages 1063–75. IEEE Computer Society, 2019. doi: 10.1145/3352460.3358296.