

POLITECNICO DI TORINO

Laurea Magistrale in Ingegneria INFORMATICA



Tesi di laurea Magistrale

User Experience della piattaforma CrownLabs

Relatori

Prof. Fulvio CORNO

Prof. Fulvio Giovanni Ottavio RISSO

Candidata

Anna Virginia NARRATONE

Dicembre 2024

Sommario

Con User Experience intendiamo l'insieme di percezioni, emozioni e reazioni che un utente prova nell'interazione con un prodotto, sistema o servizio. È un concetto ampio che include diversi aspetti dell'interazione dell'utente, come l'usabilità, l'accessibilità, l'efficienza, l'estetica e la soddisfazione generale. La UX si focalizza non solo sulla funzionalità del prodotto, ma anche su come l'utente si sente durante il suo utilizzo, cercando di ottimizzare ogni aspetto per creare un'esperienza positiva.

La piattaforma CrownLabs è stata sviluppata durante il periodo del Coronavirus per rispondere alla urgente necessità di offrire laboratori virtuali agli studenti. Nel tempo, ha subito un'evoluzione, ampliando le sue funzionalità fino a diventare una piattaforma utilizzata anche per lo svolgimento di esami e l'erogazione di servizi cloud.

Lo scopo di questa tesi è quello di migliorare e rendere più coerente la User Experience della piattaforma, con l'obiettivo di semplificarne l'uso per tutti gli utenti, indipendentemente dal loro livello di competenza tecnica. Questo significa riprogettare l'interfaccia e i flussi operativi in modo che la piattaforma risulti intuitiva e facile da navigare, eliminando la necessità di conoscenze specialistiche per sfruttarne appieno le funzionalità. L'intento finale è creare un'esperienza più accessibile e user-friendly, che consenta a chiunque di interagire con la piattaforma in modo efficace e senza difficoltà.

La prima fase ha previsto lo studio approfondito della piattaforma tramite una task analysis, analizzando le operazioni svolte dagli utenti per individuare

flussi di lavoro, difficoltà e possibili miglioramenti. Allo stesso tempo, sono state condotte interviste con i docenti per raccogliere feedback diretto sulle loro esperienze d'uso, identificando funzionalità utili e sfide operative. Queste attività hanno permesso di comprendere meglio le esigenze degli utenti e di ottimizzare la piattaforma.

Nella seconda fase, è stato individuato un compito specifico che i docenti devono svolgere sulla piattaforma, con l'obiettivo di migliorare l'esperienza utente attraverso una soluzione mirata. Questo ha portato alla progettazione di un nuovo prototipo, concepito per semplificare il processo, rendendolo più efficiente e soddisfacente. Il prototipo è stato testato con altri utenti, e una parte delle funzionalità sviluppate è stata implementata con successo. Questa implementazione è stata successivamente testata con i docenti, ottenendo riscontri positivi, e sarà inclusa nella prossima release della piattaforma.

Indice

Elenco delle figure	VI
Acronimi	IX
Glossario	X
1 Introduzione	1
1.1 Scopo	1
1.2 Struttura della tesi	2
2 Background	3
2.1 Stato dell'arte	3
2.1.1 Kubernetes	4
2.1.2 Harbor	6
2.2 Struttura interna	6
2.2.1 Logica di backend	7
2.2.2 Qlkube	9
2.2.3 Frontend Dashboard	9

3	Analisi	12
3.1	Identificazione degli utenti	12
3.2	Task Analysis	13
3.2.1	Hierarchical Task Analysis	14
3.3	Studio con gli utenti	15
3.3.1	Struttura dell'intervista	16
3.3.2	Risultati delle interviste	17
4	Design	20
4.1	Paper Prototype	21
4.1.1	Valutazione euristica	26
4.2	Medium fidelity prototype	29
5	Implementazione	34
5.1	Salvataggio di immagini custom per la creazione di template . .	34
5.2	Metadati	38
6	Validazione	40
6.1	Pianificazione dello Usability Testing	41
6.1.1	Task	41
6.1.2	Metriche	43
6.2	Risultati ottenuti	44
7	Conclusioni	47
	Bibliografia	49

Elenco delle figure

2.1	Rappresentazione ad alto livello dei principali elementi architet- turali che compongono CrownLabs	7
2.2	Dashboard iniziale della piattaforma CrownLabs	10
2.3	Scheda “Istanze Attive” iniziale	10
2.4	Modulo per la creazione di un <i>template</i>	11
3.1	HTA per la creazione di un nuovo template all’interno di un corso	14
3.2	HTA per l’aggiunta di nuovi utenti all’interno di un corso	15
4.1	Scheda “Modelli Personalizzati”	22
4.2	Modulo per la creazione di un’istanza da cui salvare un modello	23
4.3	Modulo per la richiesta personalizzata all’amministratore	24
4.4	Modulo per la creazione di un nuovo template all’interno di un corso	25
4.5	Scheda di espansione delle istanze utente	27
4.6	Scheda “Istanze Attive”	27
4.7	Versione iniziale con la pagina “Modelli Personalizzati”	30
4.8	Versione definitiva con il workspace personale	31
4.9	Scheda “Istanze Attive”	32

4.10	Sezione per creare un nuovo template in un corso	33
5.1	Esempio di risorsa di tipo <i>Snapshot</i> su Kubernetes	35
5.2	Modal per la creazione di un nuovo snapshot	36
5.3	Avviso sulla durata del processo di snapshotting	36
5.4	Bottone per gli snapshot	37
5.5	Elementi di feedback sullo stato del sistema per l'utente	38

Acronimi

UX

User Experience

VM

Virtual Machine

CRD

Custom Resource Definitions

CNCF

Cloud Native Computing Foundation

RBAC

Role-based Access Control

HTA

Hierarchical Task Analysis

Glossario

Istanza

Un'istanza di una macchina virtuale (VM) rappresenta una specifica esecuzione di una macchina virtuale configurata con determinate risorse e impostazioni. Un'istanza include una copia operativa del sistema operativo, del software installato e delle configurazioni definite, ed è eseguita all'interno di un ambiente virtuale.

Tenant

Un tenant rappresenta un utente della piattaforma, come uno studente o un docente, definito attraverso informazioni personali e autorizzazioni specifiche per accedere a determinati workspaces. Ogni tenant dispone di uno spazio di lavoro dedicato, denominato namespace, e di risorse assegnate esclusivamente per il proprio utilizzo.

Workspace

Un workspace è una collezione logica di risorse correlate all'interno della piattaforma, come template di macchine virtuali o istanze, che rappresenta un ambito operativo condiviso tra più utenti. I workspaces sono progettati per organizzare le risorse in base a criteri come corsi, progetti o gruppi di lavoro, e definiscono i permessi di accesso per utenti specifici, garantendo che i partecipanti possano collaborare o accedere solo alle risorse pertinenti al loro ruolo. All'interno di ciascun workspace, ai tenant possono essere assegnati uno dei due ruoli disponibili: i *manager*, che hanno la possibilità

di creare nuovi template e gestire l'elenco dei tenant associati al workspace, e gli *user*, che possono esclusivamente avviare istanze basandosi sui template già creati dai manager.

Template

Un template è una risorsa preconfigurata che definisce le caratteristiche di base di un ambiente virtuale, come una macchina virtuale o un container, all'interno della piattaforma. Include informazioni quali le dimensioni delle risorse (CPU, memoria, disco), l'immagine di base del sistema operativo o del software da utilizzare, e una descrizione dell'ambiente stesso. I template servono come punto di partenza per creare istanze operative, garantendo coerenza e facilità di configurazione.

Snapshot

Uno snapshot è una copia puntuale dello stato di un ambiente virtuale in un dato momento. Uno snapshot di una macchina virtuale persistente crea una nuova immagine basata sullo stato attuale del disco, che può essere salvata e utilizzata per generare nuovi template.

Deployment

Un Deployment è una risorsa nativa di Kubernetes utilizzata per gestire il ciclo di vita di applicazioni containerizzate. È uno strumento che permette di descrivere e controllare lo stato desiderato di un'applicazione, garantendo che questa venga distribuita, aggiornata e mantenuta attiva nel cluster Kubernetes in modo affidabile.

Capitolo 1

Introduzione

1.1 Scopo

Questa tesi è stata sviluppata al Politecnico di Torino ed è incentrata sulla User Experience (UX) della piattaforma CrownLabs.

Il progetto di CrownLabs è nato nel 2020 durante la pandemia COVID-19 per rispondere alla necessità di fornire agli studenti dei corsi di networking la possibilità di continuare a svolgere i laboratori anche da remoto.

Nonostante sia nata con un obiettivo molto specifico la piattaforma ha continuato ad espandersi nel tempo, evolvendo in un progetto sempre più grande che coinvolge docenti, dottorandi e studenti e che continua a crescere tutt'ora.

L'obiettivo di questa tesi è progettare la User Experience della piattaforma CrownLabs con un focus specifico sui docenti. Pur costituendo una minoranza rispetto agli studenti, i docenti svolgono le operazioni più complesse, rendendo fondamentale garantire loro un'esperienza d'uso semplice ed efficiente. Il fine ultimo è creare un'interazione che sia intuitiva e naturale, capace di soddisfare le necessità degli utenti attuali e di agevolare l'accesso e l'utilizzo da parte di quelli futuri.

1.2 Struttura della tesi

La tesi inizia con un'analisi del background nel capitolo 2, concentrandosi sullo stato dell'arte relativo alla struttura cloud che costituisce il supporto della piattaforma. Successivamente, viene presentata la struttura interna della piattaforma CrownLabs, con una descrizione dettagliata dei termini tecnici e dei componenti che ne definiscono il funzionamento.

La fase di analisi spiegata nel capitolo 3 si concentra sull'identificazione dei gruppi di utenti e delle loro caratteristiche principali. Da qui si passa alla task analysis, un approfondimento delle attività che gli utenti devono svolgere in relazione al sistema, con l'obiettivo di comprendere le loro esigenze. Questa fase culmina in uno studio diretto con gli utenti, che permette di raccogliere feedback utili e affinare ulteriormente il progetto.

Nel capitolo 4 viene trattata la fase del design che si sviluppa attraverso due fasi principali. La prima prevede la creazione di un prototipo su carta, un approccio essenziale per esplorare idee iniziali e testare concetti di base. La seconda fase si concentra invece su un prototipo a media fedeltà, che consente di introdurre funzionalità più sofisticate e di verificare l'efficacia delle scelte progettuali.

La fase dell'implementazione, spiegata nel capitolo 5, rappresenta la concretizzazione del lavoro, con una spiegazione dei progressi intermedi per l'implementazione delle soluzioni definite nella fase di design.

Segue, nel capitolo 6, la validazione del progetto, in cui si presentano e analizzano i risultati ottenuti, confrontandoli con gli obiettivi prefissati. Questa fase serve a dimostrare l'efficacia delle scelte progettuali e implementative, nonché il valore aggiunto portato dalle soluzioni sviluppate.

La conclusione, nel capitolo 7, offre una riflessione sui principali risultati e al loro impatto complessivo. Viene infine delineata una visione per sviluppi futuri, suggerendo possibili direzioni per migliorare ed espandere il lavoro svolto.

Capitolo 2

Background

CrownLabs è un progetto basato su Kubernetes studiato per offrire un ambiente scalabile e flessibile per la gestione di risorse virtuali e applicazioni in ambito didattico e accademico. CrownLabs permette di creare, configurare e gestire ambienti virtuali, come macchine virtuali e container, per supportare attività di insegnamento, apprendimento e ricerca.

Le sezioni seguenti offrono una panoramica dettagliata dei principali componenti di CrownLabs rilevanti per il contesto di questa tesi. L'obiettivo è fornire una comprensione chiara delle fondamenta tecnologiche e dei meccanismi operativi della piattaforma.

2.1 Stato dell'arte

In CrownLabs, Kubernetes e Harbor vengono utilizzati insieme per garantire una gestione scalabile, sicura ed efficiente sia delle applicazioni containerizzate sia delle macchine virtuali, grazie anche all'integrazione di KubeVirt. KubeVirt è un'estensione di Kubernetes che consente di gestire macchine virtuali (VM) all'interno di un cluster Kubernetes, trattandole come risorse native. Questo approccio combina la flessibilità e l'efficienza dei container con la capacità di eseguire applicazioni legacy o carichi di lavoro che richiedono macchine virtuali.

In questo contesto, Kubernetes si occupa dell'orchestrazione e gestione delle risorse del cluster, permettendo di automatizzare il deployment, il monitoraggio e il ciclo di vita sia dei container che delle VM. KubeVirt estende queste funzionalità, integrando le macchine virtuali come risorse personalizzate, gestibili attraverso gli stessi strumenti utilizzati per i container. Ciò consente di consolidare la gestione dei carichi di lavoro tradizionali e moderni in un'unica piattaforma.

Kubernetes è utilizzato perché è una piattaforma potente per l'orchestrazione dei container. Consente di gestire cluster di container in modo scalabile e automatizzato, distribuendo applicazioni, monitorandone lo stato, garantendo l'alta disponibilità e il bilanciamento del carico, e gestendo il ciclo di vita di ogni componente. Kubernetes è ideale per le applicazioni cloud-native e a microservizi, consentendo agli sviluppatori di astrarre le complessità dell'infrastruttura e concentrarsi sul codice. Inoltre, la sua flessibilità tramite le Custom Resource Definitions (CRDs) e gli operatori lo rende perfetto per implementare logiche personalizzate, come nel caso di CrownLabs.

Harbor si integra in questo ecosistema come registro delle immagini container sicuro e affidabile. Permette di gestire le immagini necessarie per Kubernetes garantendo sicurezza, tracciabilità e conformità. Harbor offre funzionalità di scansione delle vulnerabilità per identificare problemi di sicurezza nelle immagini, controllo degli accessi basato sui ruoli per gestire chi può accedere e manipolare le immagini, e firma delle immagini per garantire l'integrità e la fiducia. Questo è particolarmente utile in ambienti distribuiti e multi-tenant, dove è essenziale proteggere le risorse e fornire un ambiente affidabile per il deployment.

L'uso combinato di Kubernetes, KubeVirt e Harbor permette quindi di gestire in modo unificato e sicuro sia le applicazioni containerizzate che le macchine virtuali, ottimizzando le risorse e semplificando l'orchestrazione.

2.1.1 Kubernetes

Kubernetes è una piattaforma open-source progettata per automatizzare il deployment, la gestione e il ridimensionamento di applicazioni containerizzate

[1]. Sviluppata originariamente da Google e ora mantenuta dalla Cloud Native Computing Foundation (CNCF), Kubernetes rappresenta uno standard de facto per l'orchestrazione di container, consentendo agli sviluppatori di gestire facilmente applicazioni distribuite complesse.

Il cuore di Kubernetes è costituito da un cluster, che include un nodo master, responsabile della gestione del sistema, e una serie di nodi worker, dove i container vengono eseguiti. Attraverso un approccio dichiarativo, Kubernetes permette agli utenti di definire lo stato desiderato per le loro applicazioni, come il numero di istanze, le risorse necessarie e i criteri di rete. La piattaforma si occupa quindi di mantenere automaticamente lo stato specificato, gestendo il bilanciamento del carico, il riavvio dei container falliti e il rollout di nuove versioni.

Grazie a funzionalità come la scalabilità automatica, l'autocorrezione e la gestione delle risorse, Kubernetes è uno strumento fondamentale per sviluppare e mantenere applicazioni cloud-native, ottimizzando l'uso delle risorse hardware e garantendo alta disponibilità e resilienza.

CrownLabs adotta un approccio che integra direttamente i componenti applicativi all'interno dell'infrastruttura di Kubernetes. Il modello dei dati è strutturato utilizzando risorse standard di Kubernetes, che sono sia native sia estese tramite *Custom Resource Definitions* (CRDs) [2].

Le CRD sono un meccanismo di estensione nativo di Kubernetes che consente agli sviluppatori di definire nuove tipologie di risorse (Custom Resources) personalizzate, oltre a quelle predefinite offerte dal sistema (come Pod, Service o Deployment). Una CRD permette di adattare Kubernetes a esigenze specifiche di un'applicazione o di un'infrastruttura, creando risorse su misura che possono essere gestite e controllate come le risorse standard.

Una volta definita una CRD, Kubernetes riconosce la nuova risorsa come parte del proprio modello API. Gli utenti possono quindi creare, aggiornare, eliminare o osservare lo stato di queste risorse personalizzate tramite le stesse modalità utilizzate per le risorse native di Kubernetes.

Ad esempio, in CrownLabs, le CRD vengono utilizzate per definire risorse specifiche come spiegato nella sezione 2.2.1.

Le CRD sono supportate da operatori Kubernetes, che implementano la logica per gestire le risorse personalizzate. Ad esempio, un operatore osserva le modifiche a una risorsa personalizzata e applica la logica necessaria per creare, aggiornare o rimuovere altre risorse Kubernetes per allineare lo stato del sistema a quello desiderato. Questo approccio rende Kubernetes una piattaforma altamente estensibile e ideale per orchestrare sistemi complessi come CrownLabs.

2.1.2 Harbor

Harbor è una piattaforma open source progettata per gestire in modo efficiente i registri di container, fornendo una soluzione sicura e scalabile per l'archiviazione, la gestione e la distribuzione delle immagini dei container. Originariamente sviluppata da VMware e ora anch'esso parte della Cloud Native Computing Foundation (CNCF), Harbor è compatibile con Docker e Kubernetes, rendendola una scelta ideale per ambienti cloud-native [3].

Harbor non è solo un semplice registro; offre funzionalità avanzate come il supporto per la scansione delle vulnerabilità delle immagini, che identifica problemi di sicurezza nelle immagini dei container, e la firma delle immagini, che garantisce l'integrità e l'origine delle immagini distribuite. Supporta inoltre il controllo degli accessi basato sui ruoli (RBAC), consentendo una gestione granulare delle autorizzazioni per utenti e team.

2.2 Struttura interna

L'architettura interna di CrownLabs si basa su tre componenti principali [4]:

- La **logica di backend** fornisce le diverse funzionalità di CrownLabs ed è implementata tramite operatori Kubernetes personalizzati;
- **qlkube**, un middleware che consente di esporre l'API Server di Kubernetes e di Harbor (come spiegato nella sezione 5.2) come un servizio GraphQL;

- La **Frontend Dashboard**, che interagisce con il backend attraverso qlkube ed espone le diverse risorse personalizzate di CrownLabs tramite un'interfaccia grafica.

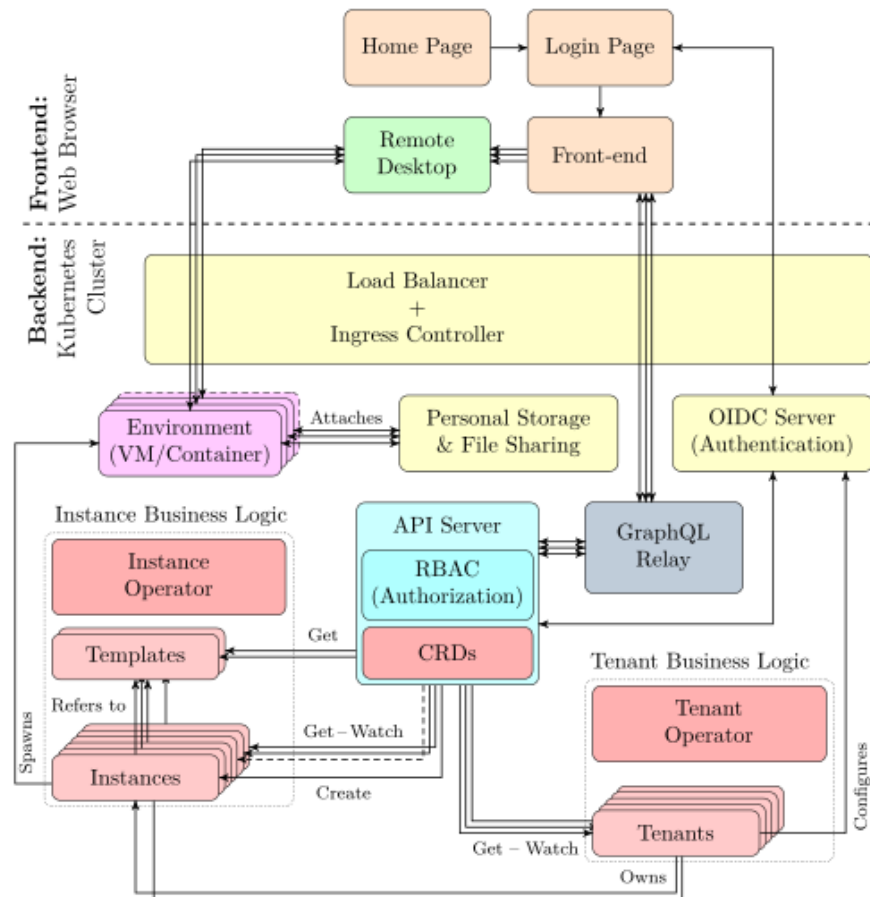


Figura 2.1: Rappresentazione ad alto livello dei principali elementi architetturali che compongono CrownLabs

2.2.1 Logica di backend

I principali componenti della logica di backend sono:

- **Instance Operator**, che implementa la logica per creare nuovi ambienti a partire da *template* predefiniti.

- **Tenant Operator**, che automatizza la gestione degli utenti di CrownLabs (i *tenant*) e dei gruppi (o *workspaces*).
- **Bastion Operator**, che configura un bastion SSH per fornire accesso agli ambienti tramite linea di comando, come alternativa all'interfaccia grafica web.

Instance Operator

L'Instance Operator si basa e gestisce tre diversi tipi di risorse custom di Kubernetes (CRDs):

- **Template** che definisce le caratteristiche dell'ambiente virtuale (dimensione, immagine di base, descrizione).
- **Instance** che rappresenta un'istanza creata da un template e ne avvia la logica di riconciliazione per creare o distruggere risorse associate.
- **InstanceSnapshot** utilizzato per creare snapshot di macchine virtuali persistenti.

Persistenza e Snapshots

La funzionalità di persistenza consente agli utenti di utilizzare Virtual Machine (VMs) che mantengono i dati tra sessioni e che possono quindi essere spente e riavviate senza perdere traccia delle modifiche fatte. Per supportare questa funzione, il sistema monitora e riconcilia costantemente lo stato delle risorse come *DataVolumes* e *Instances*, garantendo consistenza nel cluster.

L'Instance Operator permette anche di salvare lo stato di una VM persistente in un momento specifico, creando un'immagine utilizzabile per creare nuovi *template*. Ogni volta che viene creata una nuova risorsa di tipo *InstanceSnapshot* un controller all'interno dell'Operator, chiamato *Instance Snapshot Controller* inizia il processo di creazione dello snapshot; tuttavia, affinché il processo possa essere avviato, la VM di cui si sta effettuando lo snapshot deve essere spenta, in modo da poter smontare il *DataVolume* associato e utilizzarlo nel processo di snapshotting [5].

2.2.2 Qlkube

Qlkube è un'API GraphQL che consente di interagire con Kubernetes, permettendo l'accesso a tutte le funzionalità offerte dall'API di Kubernetes, ed anche di altri API server come spiegato nella sezione 5.2, attraverso GraphQL. La particolarità di qlkube è che lo schema GraphQL viene generato dinamicamente basandosi sulla specifica OpenAPI/Swagger esposta dal cluster Kubernetes in cui è in esecuzione. Questo assicura che tutte le operazioni e i tipi presenti nell'API REST di Kubernetes siano disponibili nell'API GraphQL di qlkube. Inoltre, lo schema riflette sempre la versione esatta dell'API Kubernetes del cluster e si aggiorna automaticamente ogni volta che il cluster viene aggiornato, mantenendo una perfetta coerenza [6].

Prima di approfondire, è utile comprendere cos'è GraphQL. Si tratta di un linguaggio di query per le API e un runtime che consente ai client di richiedere esattamente i dati di cui hanno bisogno, senza sovraccarichi o limitazioni imposti da risposte standardizzate. A differenza delle API REST tradizionali, che restituiscono dati in blocchi predefiniti, GraphQL permette di specificare nel dettaglio quali campi, tipi e relazioni devono essere inclusi nella risposta. Questo approccio migliora l'efficienza, riduce il consumo di rete e rende più flessibile l'integrazione tra client e server.

Un altro aspetto distintivo di qlkube è l'integrazione con Apollo GraphQL Playground. Questo strumento permette di esplorare e testare immediatamente l'API attraverso un'interfaccia grafica interattiva, senza bisogno di ulteriori configurazioni. Grazie a questa funzionalità, è possibile sperimentare rapidamente e ottenere dati in modo mirato e preciso.

2.2.3 Frontend Dashboard

La Frontend Dashboard è sviluppata utilizzando React, una libreria JavaScript ampiamente utilizzata per la creazione di interfacce utente dinamiche e reattive. Per implementare i componenti dell'interfaccia, viene adottata la libreria Ant Design (antd), che offre un set di componenti predefiniti e personalizzabili, ideali per costruire interfacce moderne e coerenti.

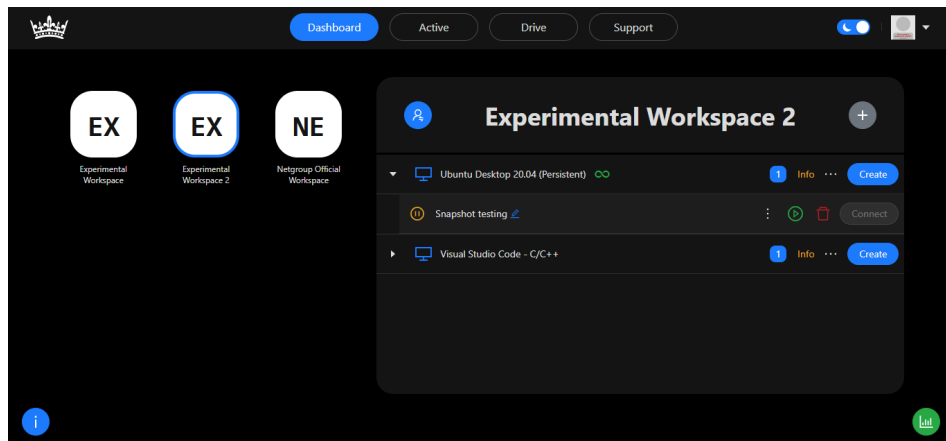


Figura 2.2: Dashboard iniziale della piattaforma CrownLabs

Per gestire esigenze specifiche di stile, come il controllo di padding e margini, si utilizzano le utilità di Tailwind CSS, una libreria che consente di applicare stili in modo rapido tramite classi definite direttamente nel codice HTML/JSEX, garantendo modularità e leggibilità.

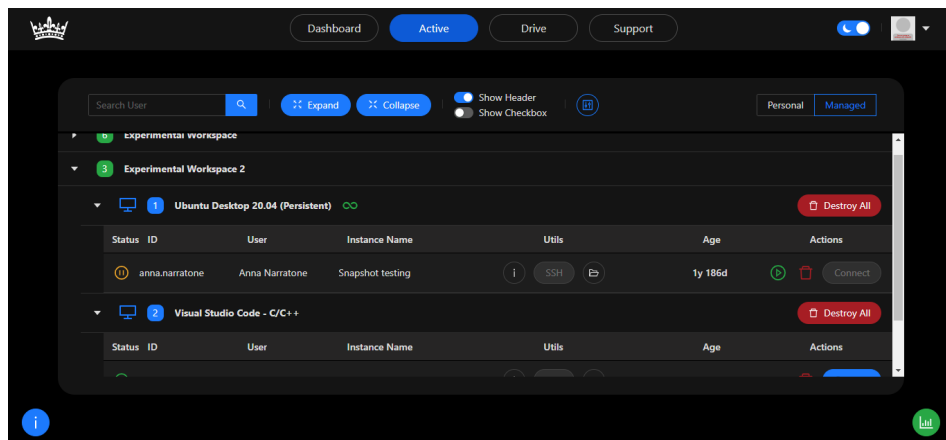


Figura 2.3: Scheda “Istanze Attive” iniziale

TypeScript è stato scelto come linguaggio per migliorare l’esperienza di sviluppo. TypeScript, estensione di JavaScript con un sistema di tipi statico, permette di rilevare errori durante la fase di scrittura del codice, aumentando la robustezza e la manutenibilità dell’applicazione. Questa scelta facilita l’uso di strumenti

come il completamento automatico, la documentazione integrata e la verifica dei tipi, contribuendo a uno sviluppo più affidabile ed efficiente.

Le figure 2.2, 2.3 e 2.4 mostrano alcune schermate dell'interfaccia nello stato in cui si trovava prima dell'inizio delle attività legate a questo lavoro di tesi. Questi esempi forniscono un quadro dello stato iniziale, utile per comprendere il contesto di partenza e confrontarlo con le modifiche e i miglioramenti introdotti successivamente.

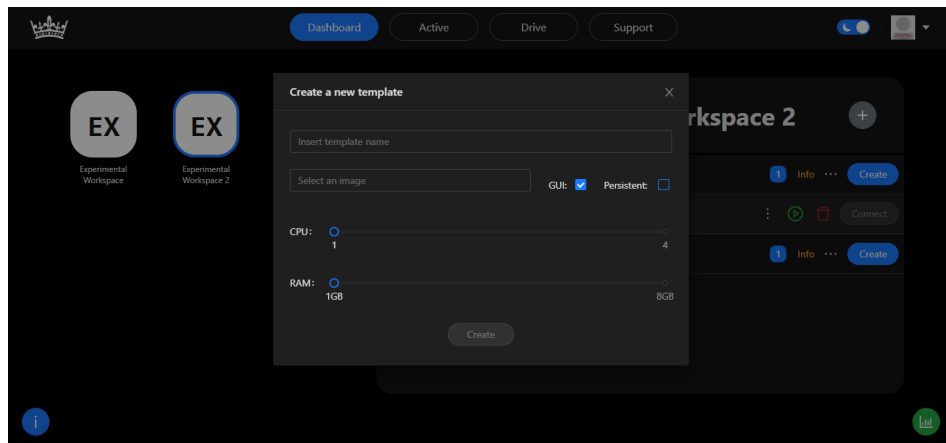


Figura 2.4: Modulo per la creazione di un *template*

Capitolo 3

Analisi

Nel contesto della User Experience la comprensione e l'analisi dei task critici rivestono un ruolo fondamentale per il miglioramento dell'interfaccia e delle funzionalità disponibili agli utenti.

Per questo è necessario individuare e conoscere innanzitutto le tipologie di utenti che utilizzano la piattaforma CrownLabs e poi, in secondo luogo, imparare quali sono i task principali che questi utenti possono svolgere sulla piattaforma.

Per poter, infine, capire come migliorare l'esperienza per gli utenti è fondamentale comprendere come al momento gli utenti svolgano le loro attività e quali sono i bisogni che l'evoluzione della piattaforma può soddisfare.

3.1 Identificazione degli utenti

CrownLabs è utilizzato principalmente da due categorie di utenti:

Gli **studenti** (internamente chiamati *user*) sono coloro che accedono alla piattaforma con privilegi limitati e il cui obiettivo è lanciare e gestire istanze di macchine virtuali o container all'interno dei corsi (internamente chiamati *workspace*) a cui sono iscritti.

I **docenti** (internamente chiamati *manager*), invece, sono coloro che, oltre a gestire le proprie istanze, hanno la responsabilità di amministrare i corsi, supervisionare le istanze degli studenti, e creare nuovi template personalizzati da utilizzare nei laboratori virtuali.

3.2 Task Analysis

Un *goal* rappresenta lo stato desiderato in un contesto applicativo che un sistema di lavoro, costituito da utente e tecnologia, intende raggiungere, e può essere specificato a vari gradi di astrazione.

Il *task* è l'insieme organizzato di attività necessarie o ritenute tali da un agente (umano o macchina) per conseguire un goal, facendo uso di una tecnologia specifica. Questo task viene progressivamente suddiviso in livelli di dettaglio crescente fino a essere descritto in termini di singole azioni.

Un'*azione* è una componente elementare di un task che non comporta risoluzione di problemi e non include alcuna struttura di controllo, configurandosi così come un "task semplice" [7].

Nei paragrafi seguenti sono descritte le operazioni che risultavano già accessibili agli utenti all'inizio della fase di studio e analisi.

Gli studenti hanno a disposizione una serie di task incentrati sull'interazione con i workspace:

- navigare tra i workspace a cui sono iscritti e vedere la lista di template disponibili per ciascuno di essi;
- creare, connettersi ed eliminare le istanze scelte tra i template proposti per il rispettivo workspace;
- accedere alle informazioni dell'account e aggiungere o rimuovere una chiave SSH;
- gestire e accedere ai file nel drive personale;
- contattare il supporto.

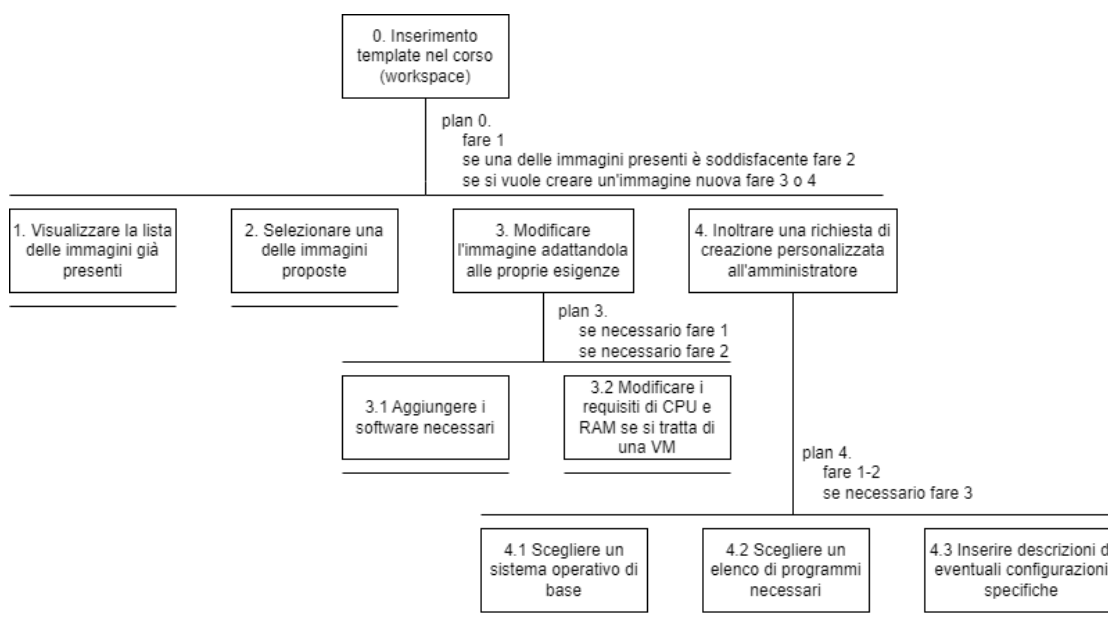


Figura 3.1: HTA per la creazione di un nuovo template all'interno di un corso

I docenti invece, oltre ai task appena elencati, possono eseguire operazioni più complesse nei workspace in cui hanno i privilegi di manager, tra cui:

- modificare il ruolo di un utente di quel workspace;
- aggiungere uno o più utenti;
- creare un nuovo template per un workspace;
- visualizzare, accedere e distruggere le istanze attive degli utenti.

3.2.1 Hierarchical Task Analysis

Uno dei possibili metodi per la decomposizione di questi task in azioni è la Hierarchical Task Analysis (o HTA). Questa procedura permette di dividere i task in sotto-task che possono poi essere analizzati usando la sequenza logica di esecuzione: il *plan*.

Nelle figure 3.1 e 3.2 vediamo un esempio di HTA per due task dei docenti: la creazione di un nuovo template e l'aggiunta di nuovi utenti all'interno di un corso.

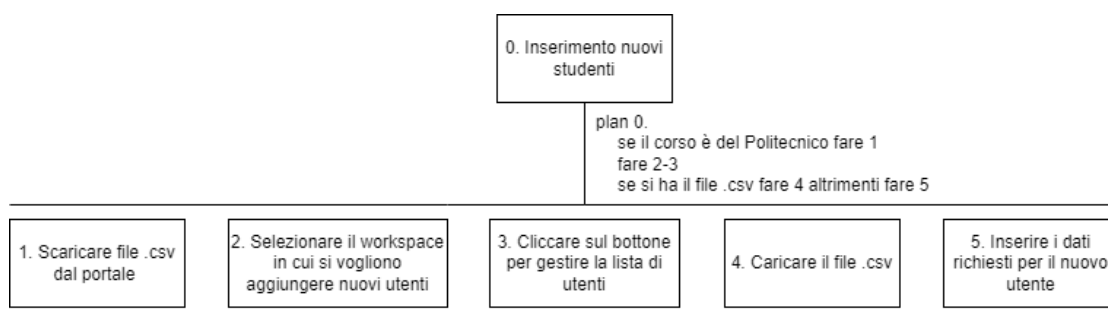


Figura 3.2: HTA per l’aggiunta di nuovi utenti all’interno di un corso

La creazione di nuovi template rappresenta uno dei task più lunghi e complessi per i docenti sulla piattaforma CrownLabs. Questo task consente ai docenti di definire le configurazioni di base per i laboratori, selezionando il tipo di macchina virtuale e i software necessari per lo svolgimento di esercitazioni e laboratori.

Attualmente, la piattaforma consente ai docenti di creare template configurando macchine virtuali o container specifici per i vari workspace e proporre questi template agli studenti per l’uso nei rispettivi workspace, garantendo che le macchine siano configurate con i software necessari per il corso.

La creazione di un template richiede una buona comprensione delle risorse disponibili e delle necessità del corso, nonché la capacità di selezionare le configurazioni corrette in base agli obiettivi didattici. Tuttavia, questo processo non è sempre lineare e presenta diverse difficoltà per i docenti, specialmente quelli meno esperti in ambienti di virtualizzazione.

3.3 Studio con gli utenti

Il *needfinding* è un metodo di ricerca incentrato sull’identificazione dei bisogni degli utenti attraverso l’osservazione e l’ascolto diretto. In questo studio, sono state condotte delle interviste ai docenti per comprendere in profondità le esigenze e i punti deboli legati allo svolgimento dei laboratori e degli esami.

Lo studio è stato condotto selezionando i docenti tra coloro che svolgono i laboratori con il maggior numero di studenti presso i Laib del Politecnico di Torino. I docenti sono stati scelti in base alla tipologia di corsi da loro gestiti, assicurando una diversificazione tra docenti di discipline informatiche e non informatiche, e tra utenti di genere differente per rappresentare una varietà di prospettive.

Sono stati intervistati complessivamente nove docenti, di cui sette uomini e due donne. La maggior parte di loro, ossia sei docenti, insegna in corsi di informatica, mentre i restanti tre appartengono ad ambiti disciplinari diversi: uno è docente di ingegneria biomedica, un altro di ingegneria meccanica e l'ultimo di matematica. Nessuno dei partecipanti aveva avuto precedenti esperienze dirette con la piattaforma CrownLabs. Tuttavia, quattro di loro avevano avuto modo di sentirne parlare attraverso colleghi.

3.3.1 Struttura dell'intervista

Le interviste, della durata di circa 30 minuti, sono state introdotte con una breve presentazione di CrownLabs e dell'obiettivo dello studio. Gli intervistati sono stati informati che le risposte sarebbero state utilizzate per migliorare l'usabilità della piattaforma, e che non vi sarebbero state risposte giuste o sbagliate, ma che l'intento era puramente esplorativo. È stato inoltre chiesto il consenso per registrare le interviste a fini di analisi, garantendo la riservatezza dei dati raccolti.

Le domande miravano ad esplorare vari aspetti dell'organizzazione e dell'esperienza dei laboratori e degli esami, partendo dalla soddisfazione generale del docente rispetto alla gestione attuale e dai possibili cambiamenti desiderati. Si è indagata quindi l'importanza della pratica per l'apprendimento della materia, valutando se le ore di laboratorio risultano essere sufficienti.

Un'altra parte delle domande riguardava il consiglio dato agli studenti sull'installazione dei software necessari sui propri dispositivi, cercando di capire se e perché i docenti preferiscano che gli studenti lavorino su software installato localmente.

È stato chiesto anche in che modo i docenti strutturano le esercitazioni di laboratorio – se sono guidate, libere o sviluppate come progetti a lungo termine – e quali difficoltà ricorrenti si presentano durante le attività, insieme alle soluzioni generalmente adottate per superarle.

Si è approfondito l'uso dei software durante il corso, per capire se le esigenze cambino con l'avanzare delle esercitazioni o se risulti necessaria una versione specifica per alcuni strumenti.

Si è esplorata anche l'organizzazione degli esami e l'utilizzo dei software richiesti, insieme all'esperienza vissuta durante il COVID per comprendere l'impatto del passaggio forzato agli esami online e se vi siano stati aspetti positivi o negativi rilevanti.

3.3.2 Risultati delle interviste

Dalle risposte alle interviste è emerso che, secondo i docenti, i laboratori garantiscono un ambiente uniforme per tutti, evitando problemi legati ai diversi dispositivi degli studenti.

In generale i laboratori sono apprezzati per il loro approccio pratico e coinvolgente che aiuta e consolida l'apprendimento degli studenti.

Tutti i docenti riconoscono l'importanza della pratica nei laboratori, anche se secondo il 45% dei docenti intervistati le ore dedicate non bastano per completare gli esercizi o per un approfondimento completo. Alcuni corsi richiedono esercizi pratici anche a casa, con software installabile sui dispositivi personali degli studenti.

Quando possibile, tutti i docenti intervistati suggeriscono di installare i software necessari sui dispositivi personali degli studenti, al fine di consentire loro di svolgere esercitazioni aggiuntive al di fuori dell'orario di laboratorio. Tuttavia, in due casi tra i docenti coinvolti, gli studenti del corso hanno accesso esclusivamente a versioni trial o limitate dei software, che non permettono loro di eseguire le esercitazioni a casa.

Le modalità di strutturazione delle esercitazioni variano tra i docenti, ma in generale tutte prevedono un testo che illustra gli obiettivi e fornisce una spiegazione introduttiva. Inoltre, due dei docenti offrono anche le soluzioni, affinché gli studenti possano verificare il proprio lavoro e approfondire la comprensione degli argomenti trattati. In molti casi, gli studenti hanno la possibilità di completare le esercitazioni a casa, purché dispongano del software necessario.

Sei docenti su nove intervistati hanno sottolineato difficoltà rilevanti nell'organizzazione e nello svolgimento dei laboratori didattici. Tali problematiche si articolano principalmente in due aspetti: da un lato, i tempi lunghi richiesti per installare il software necessario sui computer dei laboratori universitari, spesso complicati dall'incompatibilità tra i programmi già installati e quelli richiesti per le esercitazioni; dall'altro, la complessità e i problemi legati all'installazione e configurazione dei software sui dispositivi personali degli studenti. Queste difficoltà non solo rallentano l'inizio delle attività pratiche, ma incidono negativamente sull'efficienza generale dell'organizzazione dei laboratori, influenzando in modo negativo l'esperienza formativa sia per i docenti sia per gli studenti.

Per questo motivo cinque tra gli intervistati forniscono guide per l'installazione o assistono gli studenti nella configurazione. In un caso specifico, il docente mette a disposizione il file di configurazione per ripristinare le impostazioni corrette del software in caso di cambi accidentali fatti dagli studenti.

In generale, i software impiegati durante le esercitazioni rimangono costanti per tutta la durata del corso, con alcune eccezioni legate a variazioni nella complessità o nelle esigenze specifiche degli esercizi, che si verificano in tre dei laboratori gestiti dai docenti intervistati. Alcuni docenti, infatti, ricorrono a versioni web-based o a strumenti semplificati per le esercitazioni introduttive, in modo da facilitare l'approccio iniziale degli studenti.

Per quanto riguarda gli esami le modalità variano: la maggior parte dei corsi richiedono supporto informatico, solo tre tra gli esami dei docenti coinvolti sono su carta per favorire il ragionamento. Per gli esami digitali, in tutti i casi, è essenziale poter limitare l'accesso a internet e monitorare l'ambiente, cosa possibile nei laboratori.

La didattica a distanza ha portato difficoltà tecniche e limitazioni. Metà dei docenti ha adottato modalità che sono rimaste anche dopo la pandemia, mentre gli altri quattro sono tornati ai metodi tradizionali per una maggiore interazione e supporto agli studenti. In un paio di casi la digitalizzazione degli esami ha velocizzato le correzioni e migliorato il feedback agli studenti. Circa la metà dei docenti ha però segnalato problemi tecnici come bug o difficoltà di controllo e supporto da remoto, che possono aumentare lo stress per gli studenti.

In sintesi, le interviste hanno evidenziato l'importanza fondamentale dei laboratori nell'apprendimento, ma anche le difficoltà legate alla gestione delle configurazioni personalizzate degli ambienti di laboratorio al Politecnico. La varietà delle richieste provenienti dai numerosi docenti che utilizzano i laboratori, unita ai lunghi tempi necessari, sia fisici che burocratici, per la creazione degli ambienti, sottolinea la necessità di ottimizzare il processo e semplificare le procedure.

Da ciò emerge la necessità di fornire agli studenti gli strumenti per svolgere le esercitazioni anche al di fuori dell'orario di laboratorio, semplificando nel contempo gli step necessari ai docenti per raggiungere tale obiettivo. Un passo fondamentale in questa direzione sarebbe la velocizzazione del processo di creazione dei template, attraverso la generazione di snapshot di macchine virtuali persistenti. Questo permetterebbe di ottenere immagini di base con le configurazioni necessarie, garantendo così ai docenti l'autonomia nella creazione di laboratori personalizzati per i propri studenti in modo rapido ed efficiente.

Capitolo 4

Design

La Knowledge-Based Analysis nello studio della UX è un metodo che si basa sull'applicazione di conoscenze già consolidate per analizzare e migliorare il design di un'interfaccia o di un sistema. In pratica, si utilizzano teorie, principi e dati empirici che derivano da discipline come la psicologia cognitiva, il design centrato sull'utente e l'ergonomia. Questo approccio consente di creare esperienze utente più efficaci ed evitare errori comuni grazie a una solida base scientifica.

Questo approccio non si limita solo a valutare, ma aiuta anche nella progettazione. Si parte dall'analisi delle aspettative e dei modelli mentali degli utenti, ossia come le persone si aspettano che un sistema funzioni sulla base delle loro esperienze precedenti.

Poiché la piattaforma CrownLabs è già in uso da parte degli utenti, lo studio per la riprogettazione dell'interfaccia è stato condotto con un approccio orientato alla continuità e alla familiarità, evitando di introdurre cambiamenti drastici o stravolgimenti nell'esperienza attuale. L'obiettivo principale è stato quello di migliorare l'usabilità complessiva, intervenendo su aspetti critici senza alterare radicalmente ciò che gli utenti già conoscono e utilizzano.

La riprogettazione si è concentrata sulla riorganizzazione delle interfacce e dei flussi di lavoro, semplificando e rendendo più intuitive le operazioni principali.

In particolare, sono stati identificati i punti del sistema che presentavano maggiore complessità o ambiguità, e sono state sviluppate soluzioni volte a ridurre il carico cognitivo e a facilitare l'interazione. L'intento è stato quello di garantire che le funzionalità chiave fossero più facilmente accessibili, migliorando l'efficienza e l'esperienza complessiva degli utenti.

Questo processo ha coinvolto un'analisi approfondita delle modalità con cui gli utenti interagiscono con la piattaforma, identificando le aree che potevano beneficiare di maggiore chiarezza o di una riorganizzazione. Al contempo, si è lavorato per preservare gli elementi familiari dell'interfaccia, consentendo una transizione graduale verso il nuovo design senza che gli utenti si sentissero spaesati o dovessero affrontare un lungo periodo di apprendimento.

4.1 Paper Prototype

Il primo passo nel processo di design è stato la creazione di un prototipo su carta della nuova interfaccia, che è stato successivamente testato con un docente che non aveva mai utilizzato la piattaforma. Questo ha permesso di raccogliere feedback preliminari sull'usabilità e sull'efficacia delle interazioni proposte.

Con il prototipo su carta si è puntato a sviluppare un'interfaccia grafica con un alto livello di affidabilità in termini di profondità funzionale, grazie alla simulazione del comportamento del backend, gestita manualmente da una persona che forniva risposte alle interazioni dell'utente. Allo stesso tempo, il prototipo presenta un basso livello di affidabilità visiva, poiché le schermate sono semplicemente disegnate a mano, senza una cura particolare per l'estetica o i dettagli grafici.

Questo approccio ha consentito di concentrarsi sulle logiche di interazione e sulla chiarezza del flusso operativo, senza disperdere risorse nella definizione dei dettagli visivi, che sono stati affinati nelle fasi successive.

Le principali modifiche apportate nel paper prototype rispetto all'interfaccia corrente di CrownLabs sono state numerose e mirate a migliorare sia la comprensione sia l'usabilità della piattaforma.

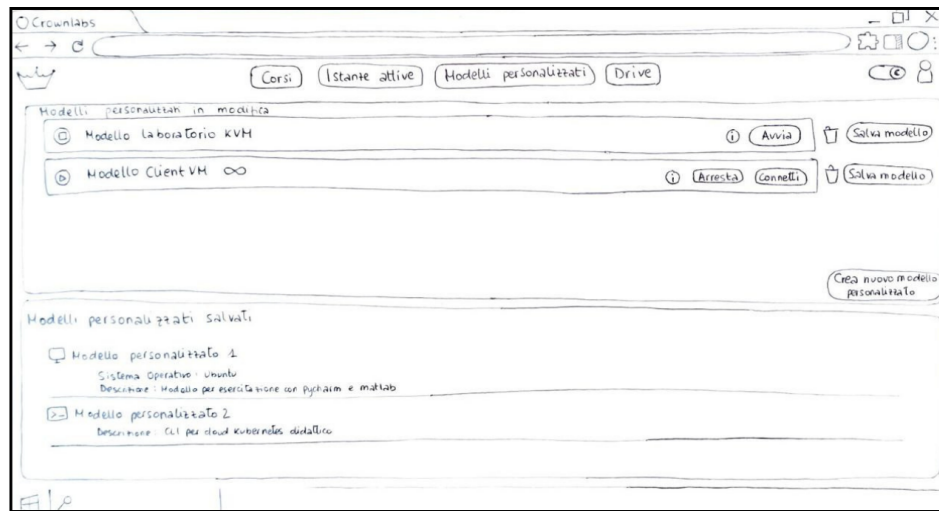


Figura 4.1: Scheda “Modelli Personalizzati”

Tra le innovazioni più significative, la creazione di una scheda dedicata ai modelli personalizzati ha rappresentato un punto cardine nell’evoluzione dell’interfaccia. Questa sezione, nella sua versione preliminare, è stata concepita per raccogliere le immagini create dai docenti a partire da snapshot di immagini persistenti, consentendo non solo di visualizzare queste risorse, ma anche di eseguire operazioni di snapshot direttamente dal frontend, operazione che in precedenza non era possibile. L’obiettivo principale era quello di offrire maggiore autonomia ai docenti nella gestione dei modelli personalizzati, semplificando il flusso di lavoro e affrontando uno dei problemi principali emersi durante lo studio con gli utenti: la difficoltà nel configurare e gestire in autonomia le personalizzazioni necessarie per i laboratori destinati agli studenti.

Questa versione preliminare della scheda è stata una risposta concreta alle esigenze rilevate, ma non ha rappresentato una soluzione definitiva. I test condotti con gli utenti hanno evidenziato ulteriori possibilità di ottimizzazione, sia dal punto di vista dell’usabilità sia nella definizione del flusso operativo. Sulla base dei feedback raccolti, il design iniziale è stato successivamente rivisto e migliorato, portando a una versione più affinata e in grado di rispondere in modo più efficace alle reali necessità dei docenti e degli altri utenti della piattaforma.

Crea nuovo modello personalizzato

Nome modello

Scegli il sistema operativo:

- Ubuntu 20.04
- Kali Linux
- Windows 10
- Mac OS

CPU Cores: 1 — Minimo 2 — 4

RAM: 1GB — Minimo 3GB — 8GB

Se non trovi quello che ti serve:

Figura 4.2: Modulo per la creazione di un'istanza da cui salvare un modello

Alcune delle altre modifiche all'interfaccia, come la distinzione tra template “Pubblicati” e “Privati” e la possibilità di espandere l'elenco delle istanze associate ai template direttamente dalla dashboard, sono state pensate per velocizzare il flusso di lavoro del docente nel seguire i laboratori in corso dei propri studenti.

Altre modifiche sono state apportate per affrontare specifici problemi di usabilità emersi durante lo studio dell'interfaccia corrente. Tra questi, una delle necessità più rilevanti è stata l'introduzione di requisiti minimi per la configurazione corretta di un nuovo template durante la fase di creazione come si vede nella figura 4.2.

Questa esigenza è nata per prevenire errori frequenti, come la creazione di template con risorse insufficienti (ad esempio, memoria RAM o CPU al di sotto delle soglie operative minime), che potevano generare comportamenti anomali o bloccare l'avvio delle macchine virtuali.

Per risolvere questo problema, è stata progettata una soluzione che associa metadati ai modelli di base, specificando i requisiti minimi richiesti per ciascuna configurazione. Durante la creazione di un nuovo template, l'interfaccia verifica automaticamente che le risorse selezionate dall'utente soddisfino tali requisiti, bloccando configurazioni non valide e fornendo un feedback chiaro per guidare la scelta.

Richiedi un nuovo modello personalizzato X

Inserisci le specifiche necessarie per il nuovo modello:

Sistema Operativo:

Interfaccia utente necessaria? Si No

L'ambiente deve essere persistente? Si No

Software necessario:

Note Aggiuntive per l'amministratore:

Invia

Figura 4.3: Modulo per la richiesta personalizzata all'amministratore

Un altro intervento significativo ha riguardato la riformattazione della terminologia utilizzata nell'interfaccia, con l'obiettivo di svincolare il linguaggio dai tecnicismi legati al backend e rendere l'esperienza più intuitiva per gli utenti finali.

Nel dettaglio, i termini utilizzati nell'interfaccia sono stati riorganizzati per riflettere meglio i concetti che gli utenti si aspettano di trovare. Ad esempio, le "immagini" sono state rinominate in "modelli", per enfatizzare il loro utilizzo come base per le configurazioni personalizzate. Il termine "template" è stato

sostituito con “ambiente”, un linguaggio più familiare e accessibile. I workspace, invece, sono stati rinominati “corsi”, per rafforzare l’associazione mentale con un contesto didattico e collaborativo.

Anche le azioni hanno subito modifiche terminologiche: salvare un’immagine facendo uno snapshot è diventato “salvare un modello”, un’espressione che evidenzia il risultato finale piuttosto che il processo tecnico. Inoltre, il flusso per creare una VM persistente da cui salvare un modello è stato semplificato e integrato con la creazione dell’istanza, eliminando passaggi ridondanti e garantendo maggiore fluidità operativa.

Aggiungi nuovo ambiente

Nome ambiente

Scegli il modello di partenza:

MODELLI STANDARD	MODELLI PERSONALIZZATI
<input type="checkbox"/> Ubuntu 20.04	<input type="checkbox"/> Modello personalizzato 1
<input type="checkbox"/> PyCharm	<input checked="" type="checkbox"/> Modello personalizzato 2
<input type="checkbox"/> Matlab 2021b	
<input type="checkbox"/> Visual Studio Code - C/C++	
<input type="checkbox"/> Visual studio Code - Python	
<input type="checkbox"/> Jupyterlab	

Se non trovi quello che ti serve: [Invia una richiesta personalizzata all'amministratore](#)

[Aggiungi](#)

Figura 4.4: Modulo per la creazione di un nuovo template all’interno di un corso

Anche il modal per la creazione di un nuovo template all’interno di un corso è stato oggetto di modifiche significative per semplificarne l’uso e renderlo più accessibile agli utenti. Come illustrato nella figura 4.4, gli utenti possono ora scegliere tra uno dei modelli standard creati dagli amministratori o uno dei propri modelli personalizzati. Questa suddivisione chiara e diretta consente agli

utenti di navigare facilmente tra le opzioni disponibili, riducendo la complessità del processo.

Un'altra modifica importante ha riguardato l'assegnazione delle risorse di RAM e CPU. Questa funzionalità è stata rimossa dal modal per i template standard per ridurre la possibilità di errori e incoraggiare l'utilizzo da parte di utenti con meno esperienza tecnica. Tuttavia, gli utenti più competenti mantengono la flessibilità di creare modelli personalizzati, nei quali possono specificare la quantità di risorse desiderate. Questi valori vengono associati direttamente all'immagine salvata e vengono automaticamente applicati al momento della creazione del template.

Questa revisione ha reso il modal più intuitivo e versatile, favorendo un'esperienza più fluida sia per i principianti sia per gli utenti esperti, che possono contare su una gestione avanzata delle risorse nei modelli personalizzati.

Infine, tra le aggiunte al design dell'interfaccia, vi è stato il modal progettato per consentire ai docenti di richiedere un modello personalizzato direttamente all'amministratore (figura 4.3). Questa funzionalità è stata pensata per semplificare e ottimizzare la gestione delle richieste speciali relative alla creazione di immagini da utilizzare come base per nuovi template.

Con questa soluzione, i docenti non devono più affrontare difficoltà nel contattare gli amministratori per avanzare richieste. Attraverso il modal, possono fornire un breve elenco con le specifiche necessarie, rendendo il processo più chiaro e strutturato. In questo modo, le interazioni successive tra docenti e amministratori risultano più rapide ed efficienti, garantendo una migliore gestione delle risorse e un'esperienza complessivamente più fluida per tutti gli utenti coinvolti.

4.1.1 Valutazione euristica

Il prototipo è stato sottoposto a un test di valutazione con un docente che non aveva precedenti esperienze con la piattaforma. La sessione di test si è svolta in presenza di un facilitatore, il quale ha svolto anche il ruolo di osservatore,

Laboratorio 1 - KVM						
<input checked="" type="checkbox"/>	s280157	AnnaNarratore	Lenient Garfish	SSH	20m	Connetti
<input checked="" type="checkbox"/>	s302928	Mario Rossi	Mint Horse	SSH	1d14h	Connetti

Figura 4.5: Scheda di espansione delle istanze utente

mentre l'utente poteva interagire liberamente con il prototipo. Dopo una breve introduzione che spiegava lo scopo principale dell'applicazione, al docente è stato lasciato pieno controllo per esplorare e utilizzare l'interfaccia senza ulteriori indicazioni o vincoli.

I feedback emersi durante la sessione sono stati analizzati e classificati utilizzando le Euristiche di Nielsen [8], un metodo consolidato per valutare l'usabilità

Crownlabs

Corsi | Istanze attive | Modelli personalizzati | Drive

Cerca utente | Espandi | Riduci | Mostra solo personalizzati

Cloud Computing

ID	Utente	Nome Istanza	Strumenti	Eta'	Azioni
<input checked="" type="checkbox"/> s280157	Anna Narratore	Lenient Garfish	SSH	20m	Connetti
<input checked="" type="checkbox"/> s302928	Mario Rossi	Mint Horse	SSH	1d14h	Connetti
<input checked="" type="checkbox"/> Fulvio Corno	Fulvio Corno	Integral Husky	SSH	6h40m	Connetti

Computer Networks

Figura 4.6: Scheda "Istanze Attive"

Nielsen's Heuristics (+1)	
H1	Visibility of system status
H2	Match between system and the real world
H3	User control and freedom
H4	Consistency and standards
H5	Error prevention
H6	Recognition rather than recall
H7	Flexibility and efficiency of use
H8	Aesthetic and minimalist design
H9	Help users recognize, diagnose, and recover from errors
H10	Help and documentation
NH	Non-heuristic issue

Tabella 4.1: Euristiche di Nielsen

dei sistemi interattivi, riassunte nella tabella 4.1. Ogni problema identificato è stato associato a una delle dieci euristiche, come ad esempio la visibilità dello stato del sistema, la coerenza e standardizzazione, o il controllo da parte dell'utente. Successivamente, a ciascun problema è stata attribuita una valutazione di gravità, secondo la scala visibile nella tabella 4.2. La sintesi dei risultati è riportata nella tabella 4.3, che evidenzia le problematiche riscontrate e il loro livello di priorità per eventuali interventi di miglioramento.

Rating di gravità di Nielsen

0	Non sono d'accordo che questo rappresenti un problema di usabilità
1	Problema puramente estetico: non è necessario risolverlo, a meno che non ci sia tempo extra disponibile nel progetto
2	Problema di usabilità minore: risolverlo dovrebbe avere una bassa priorità
3	Problema di usabilità maggiore: è importante risolverlo, quindi dovrebbe essere considerato una priorità alta
4	Catastrofe di usabilità: è imperativo risolverlo prima che il prodotto possa essere rilasciato

Tabella 4.2: Euristiche di Nielsen

I feedback ricevuti hanno portato ad adattamenti per rendere il processo più fluido e allineato alle esigenze degli utenti. Questi affinamenti successivi hanno

Problema	Euristica	Descrizione	Rating
1	H8	Nella scheda di espansione delle istanze utente (fig. 4.5), visualizzata dalla dashboard, sono presenti troppe informazioni e un numero eccessivo di azioni eseguibili	2
2	H6	Nella scheda delle istanze attive (fig. 4.6), il pulsante “Mostra solo personali” è posizionato in un punto poco visibile, rendendone difficile l’individuazione	2
3	H7	Nella scheda dei modelli personali non è possibile eseguire azioni sui modelli personalizzati salvati	3
4	H5	Nella scheda “Aggiungi nuovo ambiente”, i pulsanti sono troppo ravvicinati, bisogna prestare maggiore attenzione per selezionare il pulsante corretto	3

Tabella 4.3: Risultati della valutazione euristica

permesso di trasformare le idee preliminari in soluzioni più mature e pienamente integrate nella piattaforma.

4.2 Medium fidelity prototype

Il secondo passo nella fase di design è stato lo sviluppo di un prototipo dell’interfaccia utilizzando Balsamiq, un tool grafico pensato per sviluppatori, designer e progettisti. Questo strumento consente di creare rapidamente wireframe, ossia rappresentazioni schematiche delle interfacce utente e delle schermate, ideali per siti web e applicazioni.

Il wireframe, conosciuto anche come mockup, permette di concentrarsi maggiormente sulla struttura dell’interfaccia rispetto al paper prototype. Pur mantenendo un aspetto che comunica all’utente lo stato preliminare del design, questo approccio sacrifica in parte l’affidabilità in profondità che caratterizzava il paper prototype, poiché non consente una simulazione altrettanto realistica delle logiche di funzionamento del sistema. Tuttavia, grazie alla maggiore

chiarezza visiva e organizzativa del wireframe, è stato possibile affinare ulteriormente le proposte di design, preparando il terreno per iterazioni future più dettagliate e tecnicamente solide.

Il mockup dell'applicazione è stato sviluppato partendo dai risultati delle valutazioni condotte sul paper prototype. Questa base iniziale è stata poi sottoposta a test con un utente già familiare con la piattaforma, così da permettere di raccogliere feedback mirati per affinare ulteriormente il design attraverso iterazioni successive.

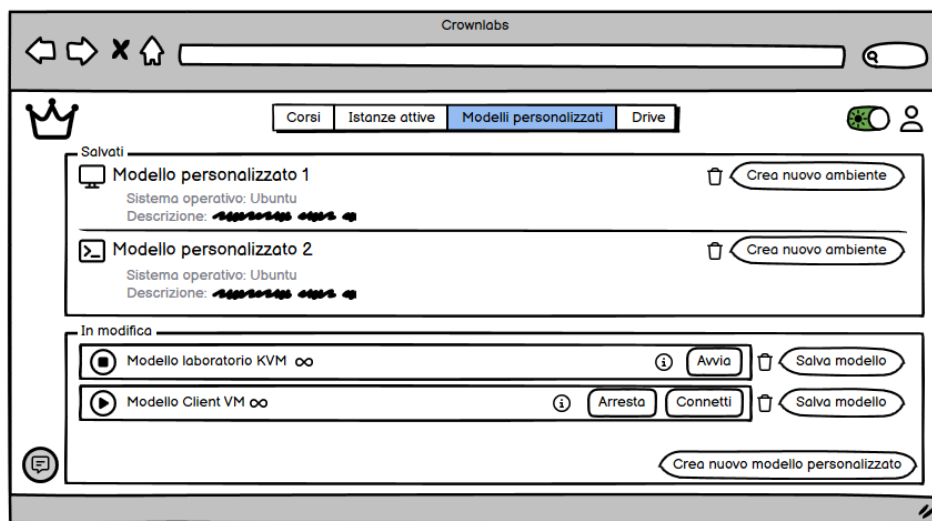


Figura 4.7: Versione iniziale con la pagina “Modelli Personalizzati”

La versione iniziale del mockup, realizzato con Balsamiq, prevedeva una pagina dedicata alla gestione dei “Modelli personalizzati” (figura 4.7), in linea con quanto progettato nel paper prototype e alle valutazioni ricevute. Tuttavia, i test condotti su questa versione hanno evidenziato alcune criticità. Per gli utenti già esperti, la struttura proposta risultava meno intuitiva e non facilmente riconducibile al loro flusso di lavoro abituale.

La valutazione di questo prototipo è stata fatta selezionando un utente che aveva già dimestichezza con l'uso della piattaforma, in modo a poter valutare l'impatto delle modifiche fatte nel paper prototype su coloro che già usavano la

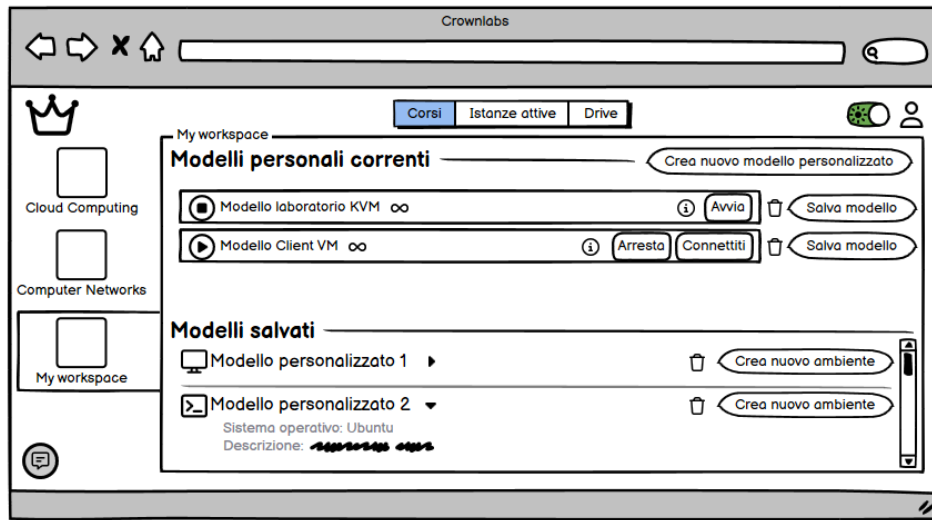


Figura 4.8: Versione definitiva con il workspace personale

piattaforma. Il risultato è riassunto nella tabella 4.4 ed è stato valutato con lo stesso metodo del paper prototype (tabelle 4.1 e 4.2).

Problema	Euristica	Descrizione	Rating
1	H4	Non è chiaro se i modelli in modifica sono personali o condivisi con altre persone	3
2	NH	È meno comodo avere le istanze in modifica in fondo alla pagina	1

Tabella 4.4: Risultati della valutazione euristica del mockup

In risposta ai feedback raccolti, il design è stato rivisto introducendo un workspace personale (figura 4.8), che ha sostituito la precedente pagina dei modelli personalizzati. Questa nuova soluzione ha reso l'interfaccia per la creazione e gestione dei modelli personali, o *snapshot*, più simile a uno degli altri corsi, ma con la peculiarità di essere un ambiente privato e dedicato. All'interno di questo spazio, l'utente ha la possibilità di gestire le proprie istanze e template in autonomia, mantenendoli separati e non condivisi con altri utenti.

Questa modifica ha consentito agli utenti di associare più facilmente l'interfaccia

al concetto di spazio personale e privato. Tale adattamento ha reso il sistema non solo più intuitivo, ma anche più in linea con le aspettative e le esigenze degli utenti con maggiore esperienza sulla piattaforma, che hanno trovato nel workspace un ambiente più familiare e funzionale rispetto alla soluzione iniziale.

Oltre alle modifiche strutturali, l'interfaccia è stata sottoposta a diversi miglioramenti estetici e funzionali, ideati per rispondere ai risultati emersi durante la valutazione del paper prototype. Tra gli interventi più rilevanti ci sono lo spostamento strategico dei pulsanti e la riorganizzazione delle azioni principali, entrambi progettati per ottimizzare l'usabilità e garantire una maggiore chiarezza del layout. Questi cambiamenti mirano a semplificare l'interazione, rendendo le operazioni più intuitive e immediate per l'utente, oltre a offrire un aspetto visivo più coerente e accattivante.

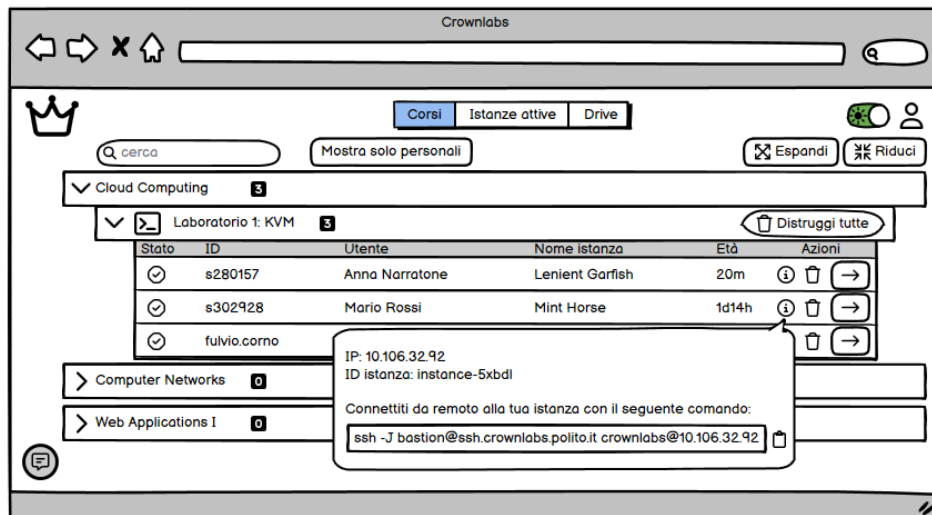


Figura 4.9: Scheda “Istanze Attive”

I bottoni per ottenere informazioni sull'istanza, come l'indirizzo IP, e per connettersi tramite SSH sono stati condensati in un unico elemento interattivo e riposizionati all'interno della colonna delle azioni nella tabella delle istanze attive (figura 4.9). Questa modifica ha mirato a migliorare la coerenza visiva e a rendere più intuitivo l'accesso alle principali funzionalità. Grazie alla riorganizzazione, le azioni correlate sono state raggruppate in modo logico, riducendo il disordine visivo e facilitando la navigazione per l'utente. Il nuovo layout mette in risalto

le opzioni più utilizzate, migliorando l'esperienza complessiva, in particolare per i nuovi utenti della piattaforma.

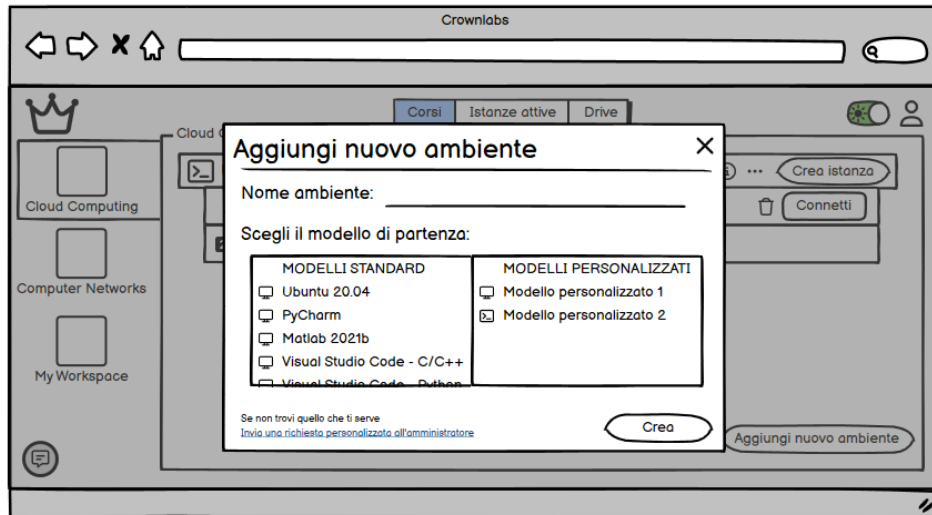


Figura 4.10: Sezione per creare un nuovo template in un corso

Rispetto al prototipo su carta, sono state aggiunte ulteriori azioni eseguibili sui modelli personali salvati. Adesso, come si vede dalla figura 4.8, è possibile non solo visualizzare i modelli, ma anche eliminarli o utilizzarli direttamente per creare un nuovo template. Queste funzionalità aggiuntive aumentano la flessibilità e l'efficienza nella gestione dei modelli, rispondendo meglio alle esigenze degli utenti e semplificando il flusso di lavoro.

Capitolo 5

Implementazione

Nel corso della fase di implementazione, è stato effettuato un aggiornamento significativo della struttura del frontend e di parte del backend, con l'obiettivo di migliorare l'efficienza, la scalabilità e l'esperienza utente della piattaforma. Questo aggiornamento ha comportato una revisione delle dipendenze del progetto, con l'introduzione di versioni più recenti delle librerie e dei framework utilizzati. L'update non solo ha garantito maggiore compatibilità con le tecnologie più attuali, ma ha anche consentito di sfruttare nuove funzionalità e ottimizzazioni, semplificando al contempo la manutenzione del codice.

5.1 Salvataggio di immagini custom per la creazione di template

Durante la fase di implementazione, l'attenzione si è focalizzata sullo sviluppo e sull'integrazione di una funzionalità fondamentale: la creazione di immagini personalizzate (snapshot) per le virtual machine. Questa funzionalità era già prevista nella struttura interna di CrownLabs grazie a un controller dedicato, che monitora costantemente lo stato e gestisce la creazione degli snapshot per le VM persistenti. Tuttavia, l'accesso a questa capacità era limitato agli amministratori di cluster, i quali, avendo accesso diretto al backend, potevano creare risorse specifiche come quella illustrata in figura 5.1.

Questa limitazione rendeva la funzionalità inaccessibile a utenti non amministratori, come i docenti, che necessitavano di strumenti più intuitivi e facili da utilizzare. Mancava, infatti, un'interfaccia frontend user-friendly che consentisse loro di eseguire queste operazioni in autonomia, riducendo la dipendenza da interventi tecnici e semplificando il flusso di lavoro.

Per superare questa limitazione e rendere la funzionalità degli snapshot accessibile a una platea più ampia, sono state progettate e sviluppate diverse caratteristiche interconnesse. Il primo passo è stato la realizzazione di un modal dedicato alla creazione degli snapshot (figura 5.2), progettato per offrire un'interfaccia semplice e intuitiva. Questo modal guida l'utente attraverso il processo, consentendogli di configurare i parametri essenziali, come il nome della nuova immagine e una descrizione associata. La complessità della risorsa sottostante viene opportunamente nascosta, poiché i restanti campi richiesti vengono compilati in automatico dal sistema.

Grazie a questo design user-friendly, il processo risulta più accessibile e meno complesso, migliorando significativamente l'esperienza utente.

Quando l'utente clicca sul pulsante *Salva*, viene creata una nuova risorsa di Kubernetes di tipo *Snapshot*, che prende lo stesso nome definito dall'utente per l'immagine da salvare. A partire da questo momento, entra in azione il controller di Kubernetes che attiva una serie di *job* specifici che eseguono la creazione dell'immagine ISO e ne gestiscono il salvataggio all'interno della repository dedicata.

```
apiVersion: crownlabs.polito.it/v1alpha2
kind: InstanceSnapshot
metadata:
  name: test-snapshot → Nome della risorsa Snapshot
  namespace: tenant-anna-narratone → Namespace della risorsa Snapshot
spec:
  imageName: vlc-ubuntu → Nome dell'immagine che verrà salvata
  instanceRef:
    name: instance-fjr5h → Nome dell'istanza da cui partirà il salvataggio dell'immagine
    namespace: tenant-anna-narratone → Namespace in cui si trova l'istanza
```

Figura 5.1: Esempio di risorsa di tipo *Snapshot* su Kubernetes

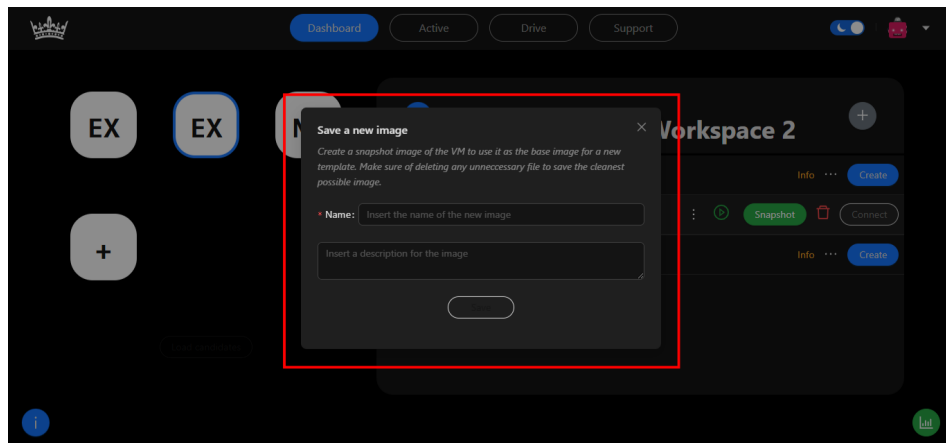


Figura 5.2: Modal per la creazione di un nuovo snapshot

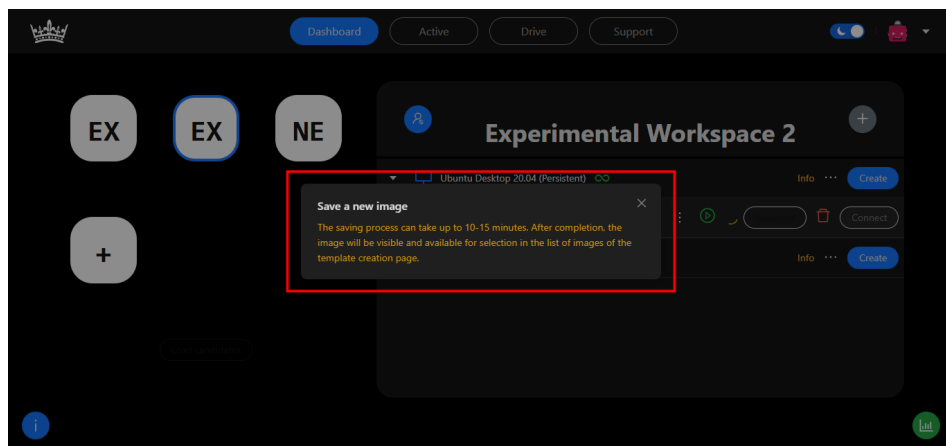


Figura 5.3: Avviso sulla durata del processo di snapshotting

Dopo l'avvio del processo di salvataggio, viene mostrata all'utente una finestra informativa che lo avvisa della durata media stimata per il completamento dell'operazione (figura 5.3). Questa finestra ha lo scopo di rassicurare l'utente, evitando che interpreti eventuali tempi di attesa come un malfunzionamento o un blocco del sistema.

Per rendere il processo di creazione di uno snapshot facilmente avviabile, è stato aggiunto un pulsante apposito al frontend (figura 5.4). Questo pulsante consente di avviare il workflow con un solo clic, integrandosi perfettamente nel flusso di lavoro degli utenti, in particolare dei docenti. La posizione del

pulsante è stata accuratamente scelta per garantire visibilità e praticità, senza interferire con altre funzioni dell'interfaccia.

Per far sì che l'utente possa essere sempre informato sullo stato del processo di snapshotting è stato introdotto un componente di feedback (figura 5.5 punto ②). Questo componente fornisce aggiornamenti in tempo reale, comunicando chiaramente quando il processo è in esecuzione, oppure completato con successo o fallito. La trasparenza offerta da questo sistema migliora la fiducia nell'affidabilità della piattaforma e consente agli utenti di monitorare facilmente il progresso delle loro azioni.

Infine, per garantire la sicurezza e prevenire errori accidentali, sono state implementate logiche di disabilitazione dei pulsanti associati a operazioni potenzialmente rischiose. Ad esempio (figura 5.5 punto ①), i pulsanti per avviare una macchina virtuale o creare un nuovo snapshot vengono automaticamente disabilitati se uno snapshot è già in corso o se la macchina virtuale è attiva. Questa funzionalità, basata sullo stato attuale delle istanze e degli snapshot, evita conflitti operativi e protegge l'integrità del sistema.

Questa integrazione automatizzata tra il frontend e i processi backend garantisce una gestione degli snapshot fluida ed efficiente, mantenendo al contempo un alto livello di astrazione per l'utente finale, che può completare l'operazione senza doversi preoccupare della complessità sottostante.

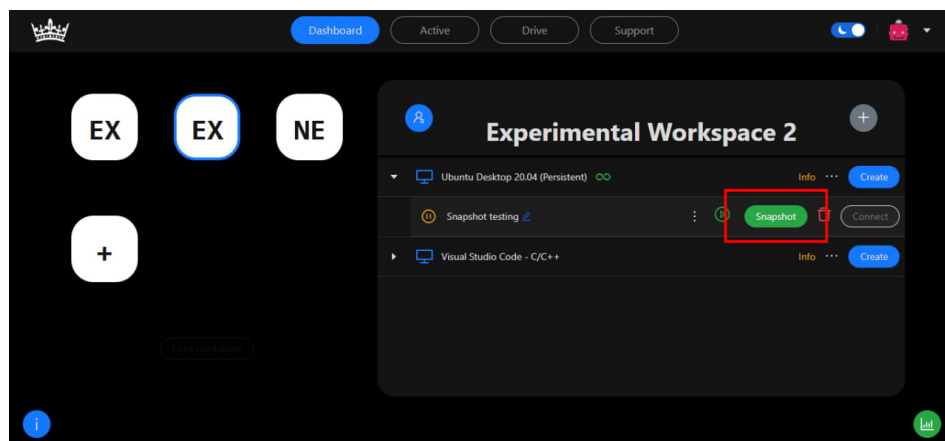


Figura 5.4: Bottone per gli snapshot

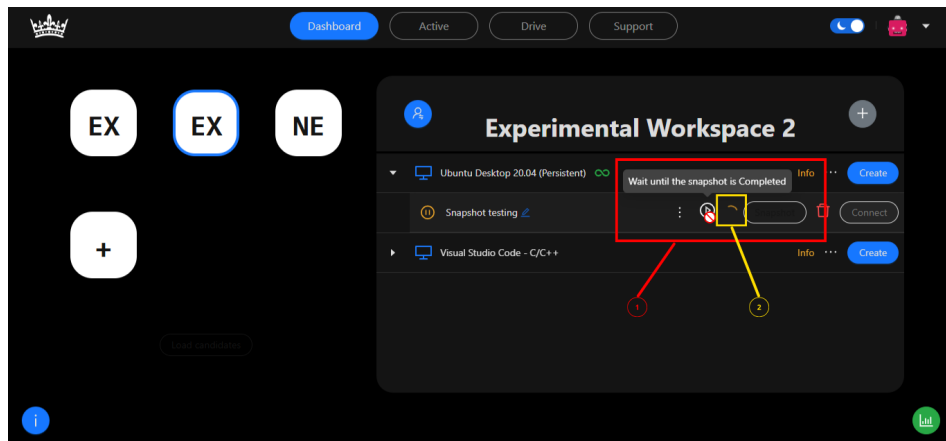


Figura 5.5: Elementi di feedback sullo stato del sistema per l’utente. Nel riquadro ① vediamo come siano stati disabilitati i bottoni “pericolosi”, mentre nel riquadro ② vediamo il componente che fornisce all’utente un feedback sullo stato corrente del sistema.

5.2 Metadati

Durante la fase di task analysis e di studio del funzionamento della piattaforma è emerso un problema critico legato alla creazione di virtual machine (VM), che influiva negativamente sull’esperienza utente e sulla funzionalità del sistema. Analizzando il comportamento della piattaforma e prendendo confidenza con le sue funzionalità, si è scoperto che era possibile generare un errore nella creazione di una VM senza ricevere un feedback chiaro sulle cause. In particolare, l’errore si verificava quando venivano assegnate risorse insufficienti di RAM e CPU alla macchina virtuale. Questo portava il sistema in un loop continuo: la VM tentava di avviarsi ma non disponeva delle risorse minime necessarie per eseguire l’immagine di partenza. Di conseguenza, il processo falliva senza fornire all’utente alcuna indicazione sul problema.

Per risolvere questa criticità è stata sviluppata una soluzione basata sull’associazione di metadati alle immagini di partenza delle VM. Questi metadati contengono informazioni sui requisiti minimi delle risorse necessarie per eseguire correttamente un’immagine, come la quantità minima di RAM e CPU. Con questa modifica, il sistema è in grado di prevenire errori bloccando la configurazione di VM con parametri al di sotto delle specifiche richieste. Gli

utenti, quindi, non potranno più selezionare valori insufficienti, evitando così il verificarsi del problema.

La soluzione è stata implementata utilizzando una feature della registry Harbor, che permette di associare dei metadati alle immagini salvate. Il principale contributo di questa tesi ha riguardato l'integrazione delle API di Harbor con il backend della piattaforma, in particolare mediante l'uso di GraphQL.

È stato sviluppato un set di query API in GraphQL che si affiancano a quelle già presenti per Kubernetes, consentendo al frontend di ottenere dal backend le informazioni relative ai requisiti minimi delle immagini archiviate su Harbor.

Questa integrazione ha richiesto un lavoro significativo di progettazione e sviluppo per garantire che il flusso di dati fosse efficiente e scalabile. Una volta implementate, le query GraphQL permettono al frontend di accedere dinamicamente ai metadati delle immagini, visualizzando le specifiche minime che l'utente in fase di configurazione può assegnare alle macchine virtuali. In questo modo, il sistema guida l'utente verso scelte corrette e consapevoli, eliminando alla radice il rischio di errori legati a una configurazione errata.

Attualmente, manca ancora un passaggio per rendere completamente operativa questa funzionalità. Deve essere ancora implementato il salvataggio dei requisiti minimi direttamente nella repository Harbor durante la fase di salvataggio delle nuove immagini, rendendo disponibili i metadati per tutte le immagini archiviate.

Una volta completata questa modifica, la funzionalità potrà essere pienamente sfruttata, utilizzando le query GraphQL già sviluppate per abilitare il controllo automatico dei requisiti minimi dal frontend.

Questa soluzione rappresenta un passo importante verso una piattaforma più robusta e user-friendly. Non solo elimina un problema tecnico significativo, ma migliora anche la comunicazione con l'utente, fornendo feedback chiari e prevenendo errori evitabili. L'integrazione tra le API di Harbor e Kubernetes, attraverso GraphQL, pone inoltre le basi per future estensioni e ottimizzazioni del sistema.

Capitolo 6

Validazione

La fase di implementazione ha avuto come focus principale la creazione dello snapshot, limitandosi però all'inizio dell'implementazione della feature per il salvataggio dei metadati associati alle immagini. Questo lavoro preliminare ha rappresentato un passo significativo per migliorare la gestione delle risorse necessarie alla creazione di macchine virtuali, ma ha lasciato spazio a sviluppi successivi per completare la funzionalità in modo integrato.

Uno dei risultati più importanti di questa fase è stata la drastica riduzione del tempo necessario ai docenti per ottenere degli ambienti di laboratorio personalizzati per i propri studenti. Prima, il processo poteva richiedere fino a un mese nel caso dei laboratori didattici del politecnico o una settimana per richieste personalizzate inviate agli amministratori di CrownLabs. Con l'introduzione delle funzionalità implementate, questo lasso di tempo è stato ridotto a poche ore, considerando anche il tempo impiegato per modificare la macchina virtuale prima di avviare il salvataggio dell'immagine.

Questa velocizzazione è stata resa possibile grazie a un flusso di lavoro più snello, che elimina passaggi superflui e consente agli utenti, come i docenti, di gestire direttamente le immagini personalizzate. Questo cambiamento non solo ha migliorato l'efficienza operativa, ma ha anche aumentato l'autonomia degli utenti, riducendo la necessità di intervento da parte degli amministratori e ottimizzando il tempo complessivo dedicato al processo.

6.1 Pianificazione dello Usability Testing

Per valutare l'efficacia della nuova funzionalità implementata, sono stati coinvolti tre docenti con diversi livelli di competenze informatiche, tutti alla loro prima esperienza con la piattaforma CrownLabs. Questo approccio è stato scelto per raccogliere una gamma di feedback diversificata, rappresentativa di utenti con differenti gradi di familiarità con strumenti tecnologici.

Il materiale richiesto per eseguire la valutazione comprendeva:

- un computer con accesso a internet e la versione aggiornata dell'applicazione;
- uno screen recorder, come OBS, per registrare l'interazione con la piattaforma;
- un cronometro per misurare i tempi di completamento delle attività;
- carta e penna per annotare osservazioni e feedback qualitativi durante la sessione.

I requisiti di base dell'applicazione per il corretto svolgimento della valutazione erano:

- un account CrownLabs con accesso in modalità manager ad almeno un corso;
- almeno un template all'interno del corso, con un'istanza persistente già avviata.

6.1.1 Task

Scrivere i task per l'Usability Testing richiede precisione e attenzione per garantire che le attività proposte siano rappresentative degli scenari reali di utilizzo e consentano di raccogliere dati significativi sull'esperienza utente.

I task devono rappresentare le attività principali e più frequenti che gli utenti svolgono all'interno del sistema. La loro formulazione deve essere realistica e basata su scenari d'uso comuni o situazioni critiche per il successo del prodotto,

così da simulare il contesto reale in cui l'interfaccia verrà utilizzata. Questo approccio consente di misurare in modo accurato quanto il sistema sia in grado di soddisfare le esigenze degli utenti e di supportarli efficacemente.

Nel caso specifico della valutazione della funzionalità implementata per la creazione di immagini personalizzate attraverso lo snapshot, i task selezionati si sono concentrati proprio su questo aspetto. Ciò ha permesso di analizzare come gli utenti interagissero con la nuova feature e di verificare se il design fosse adeguato per agevolare la creazione di snapshot, una funzione cruciale nel contesto di CrownLabs.

Per incentivare comportamenti naturali e autentici, i task sono stati presentati sotto forma di scenari, che descrivono una situazione o un obiettivo specifico che l'utente deve raggiungere. Questa metodologia consente di contestualizzare le azioni richieste e aiuta i partecipanti a immedesimarsi in un caso d'uso reale, rendendo il test più significativo. Infatti, nella tabella 6.1, è possibile osservare come i task siano stati formulati per guidare l'utente attraverso obiettivi concreti, senza suggerire direttamente le azioni da compiere. Questo approccio ha l'obiettivo di misurare con maggiore precisione l'efficacia dell'interfaccia nel supportare gli utenti nel raggiungimento dei loro obiettivi.

Un altro aspetto cruciale è che i task devono essere progettati per avere un inizio e una fine chiari. Questo permette di misurare facilmente metriche come il tempo impiegato, il numero di errori commessi e la percentuale di successo. Inoltre, i task non devono suggerire direttamente le azioni da compiere, altrimenti si rischia di alterare i risultati del test, dando all'utente una guida che non avrebbe in un contesto reale.

La metodologia Think Aloud specificata nella tabella 6.1 è una tecnica di ricerca usata nel testing dell'usabilità e nello studio del comportamento degli utenti, in cui i partecipanti verbalizzano i loro pensieri, ragionamenti, emozioni e decisioni mentre completano un compito specifico. L'obiettivo è comprendere non solo cosa fa un utente, ma anche perché lo fa, esplorando il processo mentale che guida le sue azioni.

#	Testo del Task	Criteri di successo	Metodologia
T1	Avvia la creazione di uno snapshot	BOOLEANO - Il partecipante clicca sul pulsante salva dopo aver inserito nome e descrizione)	
T2	Cerca la tua immagine dopo averla creata	BOOLEANO - Il partecipante apre la sezione "Crea Template" e cerca la sua immagine	Think Aloud

Tabella 6.1: Tabella dei *Task* da far completare agli utenti durante la valutazione

Agli utenti è stato chiesto di esprimere a voce alta tutto ciò che passava loro per la mente, inclusi dubbi, aspettative, confusione, difficoltà e soddisfazione. Questo ha permesso di comprendere dove incontravano difficoltà o incomprensioni nell'interfaccia, oltre a rivelare come interpretavano le informazioni e navigavano nel sistema.

6.1.2 Metriche

Le metriche nella valutazione della UX sono strumenti fondamentali per analizzare l'efficacia, l'efficienza e la soddisfazione degli utenti durante l'interazione con un prodotto o un sistema. Tra le principali metriche utilizzate, l'efficacia misura la capacità dell'utente di completare con successo un'attività, valutando ad esempio il numero di errori commessi o la percentuale di completamento dei compiti assegnati. L'efficienza, invece, si concentra sul tempo e sulle risorse necessarie per portare a termine un'operazione, considerando quanto rapidamente e facilmente l'utente riesca a raggiungere il suo obiettivo.

Un altro aspetto chiave è rappresentato dalla soddisfazione, che indaga il livello di appagamento o frustrazione percepito dall'utente durante l'interazione, spesso misurato tramite questionari o interviste. Si prendono in considerazione anche metriche come il tasso di successo, che rappresenta la percentuale di utenti che completano correttamente un'attività, e il tasso di abbandono, che indica quanti utenti interrompono il compito prima di terminarlo, segnalando potenziali problemi di usabilità.

Gli errori commessi dagli utenti sono un'ulteriore metrica cruciale, in quanto il loro numero, tipo e gravità aiutano a valutare quanto l'interfaccia sia chiara e intuitiva. L'impegno cognitivo, legato allo sforzo mentale richiesto per completare un'azione, e il tempo di apprendimento, che misura la rapidità con cui un utente diventa competente nell'utilizzo di un sistema, completano il quadro delle metriche utili a valutare un'interfaccia.

Grazie all'analisi di queste metriche, è possibile ottenere una visione approfondita dei punti di forza e delle criticità di un prodotto, identificando le aree che necessitano di miglioramenti per offrire un'esperienza utente più fluida e soddisfacente.

Le metriche adottate per questa valutazione hanno compreso il tempo speso per completare il task, misurato dal momento in cui il candidato iniziava l'operazione fino alla sua conclusione, per valutare l'efficienza dell'interfaccia e la chiarezza del flusso operativo. Al termine dell'attività, è stata inoltre condotta una sessione di valutazione post-operativa con i candidati, durante la quale sono stati raccolti feedback qualitativi, impressioni generali e suggerimenti per migliorare ulteriormente l'esperienza utente, approfondendo eventuali criticità o aspetti particolarmente apprezzati.

6.2 Risultati ottenuti

Come primo passo, è stata fornita una breve introduzione generale sulla piattaforma CrownLabs, spiegando il suo scopo principale e il contesto d'uso. Successivamente, è stato illustrato lo scopo specifico della valutazione: verificare se, anche senza una conoscenza approfondita della piattaforma o una particolare familiarità con essa, gli utenti fossero in grado di comprendere intuitivamente il sistema e navigare tra le sue funzionalità.

Nelle tabelle 6.2, 6.3, 6.4 sono riportati i risultati e i tempi registrati per il completamento dei task previsti. In queste tabelle sono incluse anche le osservazioni dell'osservatore che ha condotto le valutazioni, relative ai comportamenti degli utenti o agli errori riscontrati durante il test. Le osservazioni, espresse tra virgolette "...", riportano ciò che è stato detto dagli utenti sia durante lo

svolgimento dei task che nella sessione di valutazione finale, fornendo insight diretti su come gli utenti percepivano l'interfaccia e le funzionalità, nonché eventuali difficoltà incontrate.

Task	Note	Tempo richiesto
1	OK	2m 15s
2	OK	1m 12s

Tabella 6.2: Utente 1

Durante le valutazioni con gli utenti 2 e 3 è stato esposto un problema significativo: la visualizzazione della lista completa delle immagini salvate nella repository nella fase di creazione del template ha generato confusione rischiando di portare a possibili errori di selezione. Questo ha evidenziato la difficoltà nell'individuare rapidamente l'immagine desiderata tra molte opzioni, rendendo il processo meno fluido e aumentando il rischio di scelte errate. Tuttavia, è importante sottolineare che questa modifica era già stata prevista nella nuova interfaccia, ma non ancora implementata. Nella versione successiva del design, era stato pianificato di modificare la selezione delle immagini rendendola più chiara e mirata, come vediamo nell'immagine 4.10, migliorando così l'esperienza dell'utente e riducendo il rischio di errori.

Task	Note	Tempo richiesto
1	OK - Osservazione dell'osservatore: leggera confusione sul significato del disclaimer per la durata dell'operazione alla fine del procedimento di creazione	2m 2s
2	OK - Osservazione dell'utente: "Ha poco senso che io veda anche le immagini create da altri utenti, dovrei vedere solo le mie"	50s

Tabella 6.3: Utente 2

Tutti gli utenti a cui è stato fatto eseguire il test di usabilità hanno completato i task in tempi coerenti con le aspettative, considerando anche la loro scarsa esperienza nell'uso e nella struttura della piattaforma. Nonostante questa mancanza di familiarità, i tempi di completamento sono risultati in linea con

Task	Note	Tempo richiesto
1	OK - Osservazione dell'osservatore: perso un po' di tempo per assegnare un nome allo snapshot che rispettasse i criteri	3m 19s
2	OK - Osservazione dell'utente: "Ci sono troppe immagini di altre persone, preferisco vedere solo le mie"	49s

Tabella 6.4: Utente 3

gli obiettivi fissati, suggerendo che l'interfaccia fosse sufficientemente intuitiva e che gli utenti fossero in grado di orientarsi nel sistema con relativa facilità. Questo risultato ha indicato che le modifiche implementate nel design stavano migliorando l'accessibilità e la comprensione del sistema, anche per utenti meno esperti.

Capitolo 7

Conclusioni

I miglioramenti apportati alla user experience si sono dimostrati efficaci, nonostante il processo di cambiamento dell'interfaccia si sia concentrato prevalentemente sull'integrazione della nuova funzionalità per la creazione di immagini personalizzate da parte dei docenti. Questa aggiunta ha risposto a una necessità concreta, semplificando e velocizzando il processo rispetto alla versione precedente.

Il nuovo design dell'interfaccia includeva già alcune modifiche pianificate che si sono dimostrate particolarmente adatte a rispondere alle osservazioni e alle necessità espresse dagli utenti durante l'utilizzo della nuova funzionalità implementata. Queste modifiche, pensate per anticipare possibili criticità, hanno contribuito a migliorare ulteriormente l'esperienza d'uso, confermando l'importanza di progettare soluzioni in grado di adattarsi ai bisogni reali degli utenti.

Tra i futuri sviluppi già pianificati, l'introduzione della possibilità di associare metadati alle immagini salvate nella repository rappresenta un importante passo avanti nell'evoluzione della piattaforma. Sebbene questa funzionalità non sia ancora stata completamente implementata, si prevede che avrà un impatto significativo sull'esperienza utente.

Grazie ai metadati, sarà possibile fornire informazioni strutturate e rilevanti per ogni immagine, semplificando la selezione e configurazione delle risorse

durante i processi successivi. Questo non solo ridurrà la complessità per gli utenti, ma permetterà anche di offrire un'esperienza più guidata e intuitiva. Una gestione più efficiente dei dati associati alle immagini contribuirà inoltre a limitare errori nella configurazione e a rendere il sistema più accessibile anche a chi ha competenze tecniche limitate.

Essendo CrownLabs una piattaforma in costante sviluppo, questo lavoro di design si propone come una base solida e una guida per orientare le future evoluzioni e miglioramenti. L'obiettivo è garantire che le innovazioni introdotte continuino a rispondere alle esigenze degli utenti, mantenendo un approccio centrato sull'usabilità e sull'efficienza.

Bibliografia

- [1] The Kubernetes Authors e The Linux Foundation. *Documentazione Kubernetes*. Novembre 2024. Distribuito sotto CC BY 4.0. 2024. URL: <https://kubernetes.io/> (cit. a p. 5).
- [2] Marco Iorio, Alex Palesandro e Fulvio Risso. «CrownLabs—A Collaborative Environment to Deliver Remote Computing Laboratories». In: *IEEE Access* 8 (2020), pp. 126428–126442. DOI: 10.1109/ACCESS.2020.3007961 (cit. a p. 5).
- [3] Harbor Authors e The Linux Foundation. *Documentazione di Harbor*. Novembre 2024. Distribuito sotto CC-BY-4.0. 2024. URL: <https://goharbor.io/> (cit. a p. 6).
- [4] Netgroup Polito. *CrownLabs (README.md)*. README file, novembre 2024. 2024. URL: <https://github.com/netgroup-polito/CrownLabs/blob/master/README.md> (cit. a p. 6).
- [5] Netgroup Polito. *CrownLabs Operators (README.md)*. README file, novembre 2024. 2024. URL: <https://github.com/netgroup-polito/CrownLabs/blob/master/operators/README.md> (cit. a p. 8).
- [6] qlkube. *qlkube (README.md)*. README file, novembre 2024. 2024. URL: <https://github.com/qlkube/qlkube/blob/master/README.md> (cit. a p. 9).
- [7] David Benyon. «Task Analysis and System Design: the Discipline of Data.» In: *Interacting with Computers* 4.2 (1992), pp. 246–259. URL: [https://doi.org/10.1016/0953-5438\(92\)90008-4](https://doi.org/10.1016/0953-5438(92)90008-4) (cit. a p. 13).

BIBLIOGRAFIA

- [8] Jakob Nielsen. *Ten Usability Heuristics for User Interface Design*. Novembre 2024. 1994. URL: <https://www.nngroup.com/articles/ten-usability-heuristics> (cit. a p. 27).