



ISTITUTO NAZIONALE DI FISICA NUCLEARE  
Laboratori Nazionali di Frascati

---

INFN-24-01 LNF  
4/3/2024

**Booking: un software per la prenotazione delle risorse dell'ente**

Giovanni Lorenzo Napoleoni<sup>1</sup>, Claudio Bisegni<sup>1</sup>, Michele Tota<sup>1</sup>, Ramon Orrù<sup>1</sup>,  
Foggetta Gennaro Luca<sup>2</sup>

<sup>1</sup>INFN-LNF, Servizio di Calcolo, I-00044 Frascati (Roma)

<sup>2</sup>INFN-LNF, Servizio LINAC, I-00044 Frascati (Roma)

**Abstract**

Progettazione e sviluppo di un software per la prenotazione delle risorse dell'ente che richiedono un *workflow* approvativo (semplice o complesso) come la prenotazione di *facility* per esperimenti e aule per conferenze. Lo scopo del software consiste nell'informaticizzare tutti i processi di *business*<sup>1</sup> applicati alla prenotazione di una determinata risorsa e nell'eliminazione dei processi cartacei non tracciati, garantendo i principi di autenticazione e autorizzazione in base ai ruoli che l'INFN assegna ai singoli utenti. Attraverso la definizione del *workflow* è possibile configurare, oltre al normale flusso procedurale, timer, sotto *workflow*, invio di email scadenzate e attivare procedure *custom* come l'invio di comandi ad apparecchiature elettroniche. Il *core* del progetto è basato sull'utilizzo della notazione BPMN per descrivere i *workflow* legati alle prenotazioni delle risorse e all'implementazione di un orchestratore per gestirne il ciclo di vita. A differenza di altri strumenti, la notazione BPMN permette di definire un *workflow* senza la scrittura di codice con un linguaggio di programmazione ma con una definizione grafica che facilita la gestione di tutto il processo di prenotazione. E' stata anche sviluppata un'applicazione web che attraverso una GUI guida l'utente e gli amministratori nel processo di creazione e prenotazione delle risorse. Al momento il software è in uso per la prenotazione della *Beam Test Facility* dei LNF e della *Labec Facility* della sezione di Firenze.

---

<sup>1</sup>[https://it.wikipedia.org/wiki/Processo\\_aziendale](https://it.wikipedia.org/wiki/Processo_aziendale)

*Published by  
Laboratori Nazionali di Frascati*

## 1 Introduzione

La crescente complessità delle attività svolte all'interno dell'Istituto Nazionale di Fisica Nucleare (INFN) ha richiesto, nel corso del tempo, una gestione efficiente delle risorse come laboratori, esperimenti e aule per le conferenze. L'accesso a queste risorse è fondamentale per garantire una corretta pianificazione e un utilizzo ottimale degli strumenti e delle strutture a disposizione del personale dell'ente. Nell'attuale contesto organizzativo dell'INFN, la prenotazione di queste risorse avviene ancora attraverso un sistema manuale che comporta diversi ostacoli, come la complessità nel coordinare le richieste, il rischio di sovrapposizioni, la gestione dei processi autorizzativi, l'invio di manuale di email e la difficoltà nel monitorare in quale stato si trova la richiesta. Inoltre, il processo di *business* che guida le prenotazioni non è definito in maniera formale, rendendolo soggetto ad interpretazioni errate o nel mancato rispetto di tutti i passaggi richiesti per portare a compimento la prenotazione della risorsa. Per risolvere questi problemi è stato sviluppato un software basato sul concetto di *workflow* per descrivere i processi di *business* richiesti per la prenotazione di una risorsa. Il software proposto mette a disposizione dell'utente uno strumento per definire in maniera semplice e standardizzata un *workflow* per gestire l'accesso a risorse eterogenee senza dovere scrivere linee di codice in un linguaggio di programmazione. Il software si occuperà di gestire totalmente il ciclo di vita di una richiesta di prenotazione attraverso un orchestratore di workflow, nascondendo la complessità all'utente finale, il quale attraverso una semplice interfaccia *web* potrà gestire in autonomia le proprie prenotazioni.

## 2 Workflow di prenotazione

Una prenotazione di una risorsa consiste nell'effettuare una serie di passi o fasi che devono essere seguiti per avere accesso alla risorsa. Una prenotazione non è altro che un processo *step by step* riconducibile al concetto di *workflow* (esempio Figura 1), ovvero, una sequenza strutturata di azioni o attività che devono essere eseguite in un certo ordine per raggiungere un obiettivo specifico. Analogamente, una prenotazione di una risorsa richiede un insieme di azioni coerenti e ordinate per garantire che la risorsa venga prenotata correttamente. Il software sviluppato permette la modellazione della prenotazione delle risorse attraverso la definizione di un *workflow* e alla sua gestione mediante un orchestratore di *workflow*, fornendo tutti gli strumenti per definire quale utente può eseguire un determinato passaggio e chi può vedere i diversi passaggi.

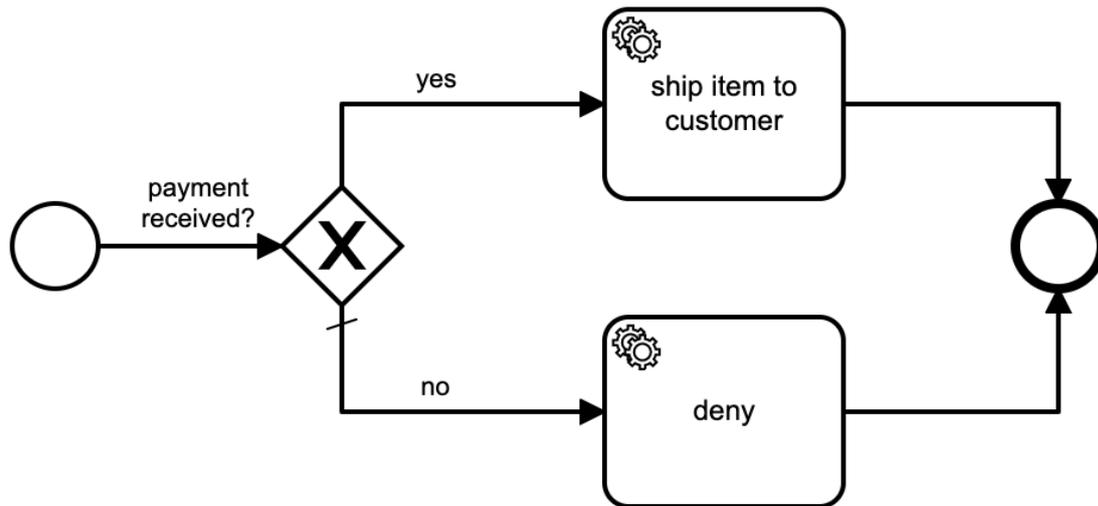


Figura 1: Esempio workflow.

Da un punto di vista matematico un *workflow* è riconducibile ad una *Finite State Machines* (FSM o automi a stati finiti) <sup>2</sup>, ovvero ad un modello matematico e concettuale che rappresenta un sistema che si evolve attraverso una serie di stati discreti. Una FSM è costituita da un insieme di stati, una serie di transizioni tra gli stati e una serie di azioni associate a tali transizioni (esempio in Figura 2).

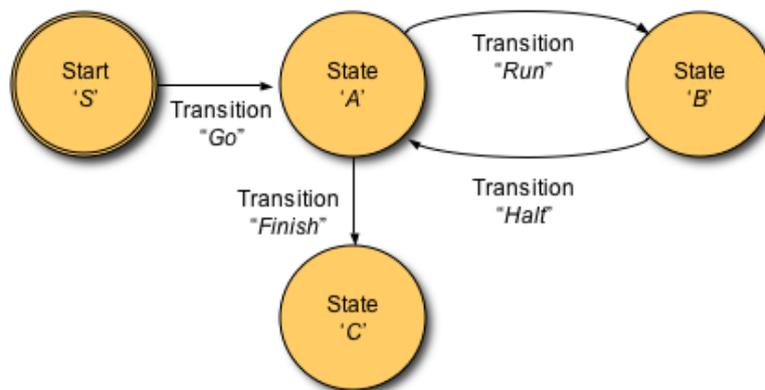


Figura 2: Esempio automa a stati finiti.

### 3 Uno standard per definire i workflow

L'adozione di una modellazione del software basata sul concetto di *workflow* necessita di un opportuno strumento che permetta di comunicare e gestire le procedure in modo standard. Se dal lato teorico le *Finite State Machines* aiutano nel progettare in maniera rigorosa un *workflow*, dal lato software è necessario trovare un medesimo standard di progettazione. Uno standard in questo contesto è fondamentale affinché non si verifichino

<sup>2</sup>[https://en.wikipedia.org/wiki/Finite-state\\_machine](https://en.wikipedia.org/wiki/Finite-state_machine)

determinate problematiche che possano avere un impatto negativo sul software e sull'interazione con i sistemi già esistenti. Tra le potenziali problematiche potrebbe verificarsi un'interoperabilità limitata poiché senza uno standard i sistemi informatici potrebbero avere difficoltà nel comunicare tra loro compromettendo l'interscambio di dati ed informazioni; inoltre l'interpretazione potrebbe non essere chiara e univoca. Per questa serie di motivi sono state valutate diverse tecnologie che meglio potevano fornire uno standard rigoroso, fornendo al tempo stesso un alto livello di astrazione: è necessario ricordare che una delle *feature* di questo progetto consiste nel fornire un software in grado di gestire qualsiasi tipo di prenotazione di una risorsa in maniera semplice e con un sufficiente grado di complessità per gestirne tutti i possibili casi.

Tra i possibili standard che sono stati valutati vi è l'utilizzo di librerie che implementano i modelli e le regole per gestire le *Finite State Machines*. Queste librerie se in prima battuta sono ottime nel gestire *workflow* di piccola-media dimensione, risultano essere problematiche su *workflow* di grande dimensione. Queste problematiche emergono nel dover definire tutti gli stati e le transizioni tra di essi attraverso linee di codice, nel gestire la gestione degli errori, la scalabilità e la limitazione nell'espressione della logica di *business: workflow* complessi, richiedono una logica complessa. Da queste criticità emerge anche la necessità di avere non solo uno strato software che fornisca dei modelli astratti per la definizione di un *workflow* ma anche un ulteriore strato che sia in grado di gestire il ciclo di vita del *workflow* stesso, ossia cosa fare quando arriva una azione che fa cambiare di stato il *workflow*. Un approccio di questo tipo basato sull'utilizzo di librerie che necessitano la definizione del *workflow* attraverso delle linee di codice pone l'attenzione su un ulteriore problema: è lo sviluppatore che deve adattare un *workflow* astratto ad una implementazione fornita dalla libreria *software*. Questa via esclude totalmente le figure coinvolte nella realizzazione di un processo decisionale per la prenotazione di una risorsa: ogni volta lo sviluppatore deve tradurre in codice ciò che il *management* ritiene opportuno configurare nel processo.

Per questa serie di motivi si è scelto di adottare un linguaggio di modellazione grafica utilizzato per descrivere i processi di business in modo chiaro e comprensibile, ossia il *Business Process Model and Notation*<sup>3</sup> (BPMN). Il BPMN è un formato standardizzato che consente di rappresentare visivamente i passaggi, le attività, le decisioni, i flussi di lavoro e le interazioni tra le diverse entità coinvolte in un processo di business. L'obiettivo primario del BPMN è quello di fornire una notazione facilmente comprensibile da tutti gli utenti riducendo la divergenza tra la progettazione di un processo di business e l'implementazione di quest'ultimo.

La figura 3 mostra un esempio di *workflow* definito attraverso la standard BPMN. Il *workflow* rappresenta la prenotazione di una risorsa attraverso la compilazione di un questionario e una approvazione da parte di utenti che hanno un ruolo di *management*.

---

<sup>3</sup><https://www.bpmn.org/>

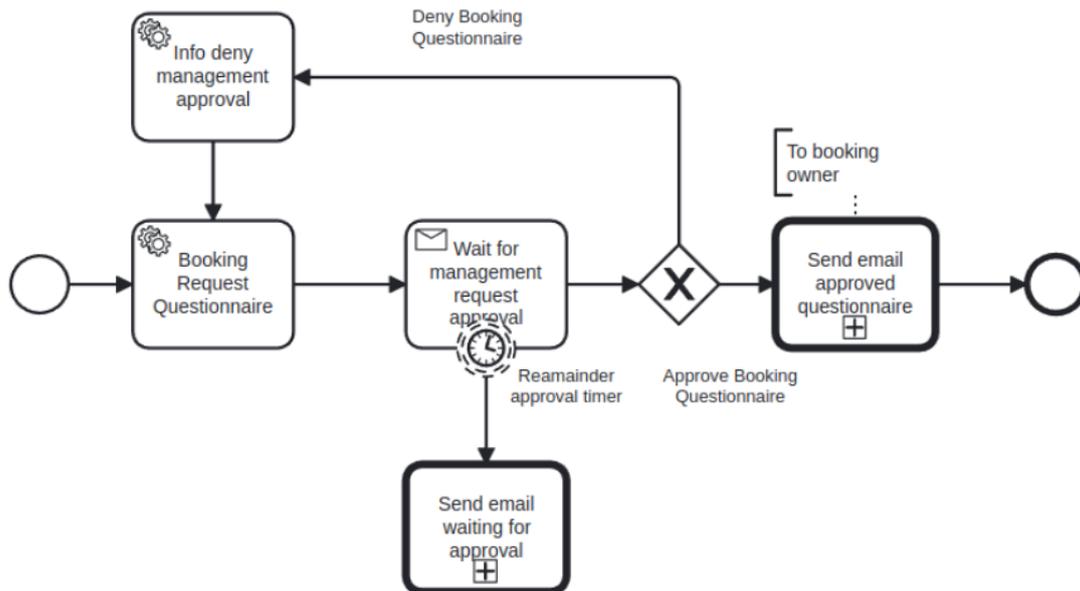


Figura 3: Esempio processo BPMN.

#### 4 Architettura e Implementazione del sistema

Il compito principale del software consiste nel mettere a disposizione dell'utente una piattaforma che semplifichi e tracci le prenotazioni per le risorse appartenenti all'ente. Con la prenotazione di una risorsa si intende quell'insieme di condizioni e di step approvativi tali che permettano di allocare del tempo per la fruizione della risorsa stessa. Tuttavia l'allocazione temporale deve rispettare dei periodi, dettati dalle esigenze dell'ente, nel quale le risorse potrebbero non essere rese disponibili. Il sistema perciò darà la possibilità di creare delle *configurazioni* ovvero permettere di decidere i periodi nei quali le risorse non appariranno tra quelle prenotabili. Per ogni risorsa, in base alle configurazioni disponibili, sarà disponibile un calendario nel quale proporsi per richiedere la riserva di quest'ultima evitando così possibili conflitti di orari con le altre prenotazioni. Inoltre l'avanzamento dello stato della richiesta dovrà essere gestito e mostrato all'utente che sottomette la proposta in modo tale da poter tener traccia dell'andamento, positivo o negativo, di tutti gli step che portano all'approvazione della prenotazione. All'interno del sistema come quello dell'INFN è essenziale che tutta la documentazione riguardante sia i moduli per la richiesta di fruizione delle risorse sia quella riguardante le autorizzazioni per l'accesso sul territorio da parte di utenti esterni e non, venga registrata e validata dalle unità competenti permettendo così una tracciabilità completa sulle operazioni svolte all'interno dell'ente.

Per la definizione dell'architettura è stato utilizzato un approccio basato su microservizi ovvero un criterio architeturale per lo sviluppo di applicazioni software in cui l'applicazione viene suddivisa in un insieme di servizi più piccoli e autonomi, chiamati microservizi. Ogni microservizio rappresenta una funzionalità specifica dell'applicazio-

ne e può essere sviluppato, distribuito e gestito in modo indipendente dagli altri servizi. In un'architettura a microservizi, i vari servizi comunicano tra loro attraverso delle API (Application Programming Interface) <sup>4</sup>. La figura 4 mostra l'architettura del sistema. Per definire l'architettura è stato scelto un modello a tre livelli: *Front-End*, *Back-End* e *Data Layer*. Il *Front-End* si occupa dell'interfaccia utente e dell'interazione con questo, il *Back-End* di contro gestisce tutta la logica di business del software, l'elaborazione dei dati e la comunicazione con i *database*. Infine il *Data Layer* si occupa della memorizzazione e del recupero dei dati dai database.

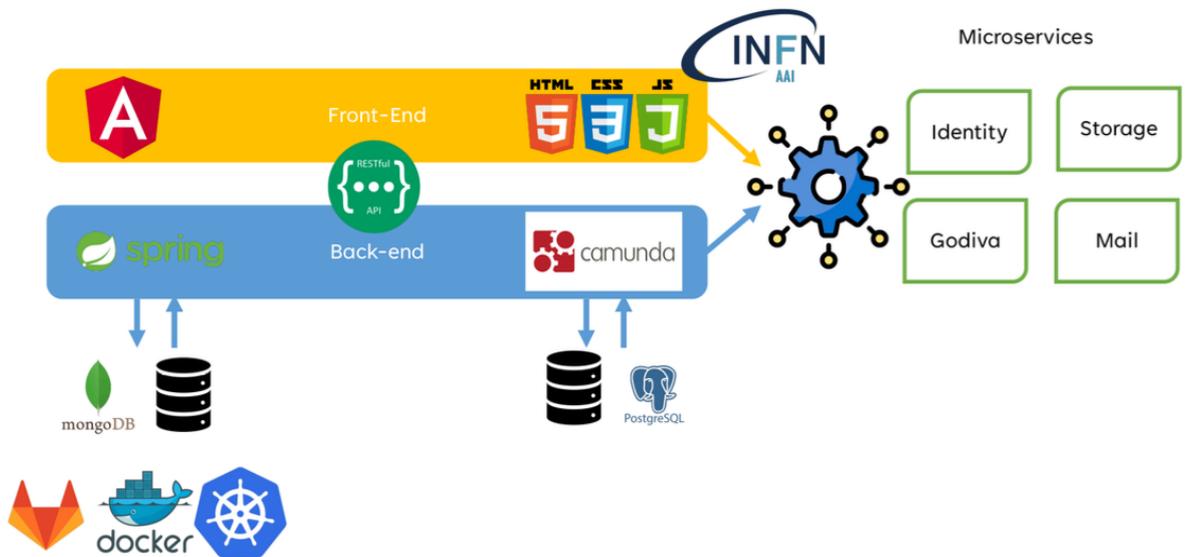


Figura 4: Stack Software.

Per lo sviluppo dell'interfaccia utente (*Front-end*) è stato utilizzato il *framework* di *Angular* mentre il *back-end* è stata sviluppato usando il *framework* di *Spring* basato su *Java*. Il *back-end* mette a disposizione dell'esterno delle API *RESTful* per accedere alle funzioni per la gestione delle prenotazioni. Le API sono autenticate e autorizzate mediante lo *standard OAuth2* <sup>5</sup>. Per il *data layer* è stato utilizzato sia un database *NoSQL mongoDB* e una database relazionale *PostgreSQL*. Il *deploy* dell'applicazione viene gestito attraverso la creazione di build con il sistema di *docker* e attraverso l'*orchestrator container* di *Kubernetes*.

Un altro aspetto fondamentale dell'architettura è l'integrazione con i sistemi già esistenti. Poiché una prenotazione di una risorsa può essere soggetta a *step* autorizzativi e/o approvativi, è importante utilizzare i ruoli e le autorizzazioni che l'INFN, tramite il proprio IAM (*Identity and Access Management*), assegna ai suoi dipendenti per discriminare

<sup>4</sup><https://www.igi-global.com/article/application-programming-interface-api-research/232166>

<sup>5</sup><https://oauth.net/2/>

chi può fare cosa è chi può vedere cosa. Per queste ragioni il sistema deve dialogare con i microservizi che espongono le API per reperire queste informazioni. In base all'utente che sta interagendo con un determinato *workflow*, il sistema sarà in grado di determinare le autorizzazioni associate all'utente. Oltre a questo aspetto, sono stati utilizzati anche i servizi (già esistenti) per l'invio di email e per la conservazione di documenti. L'interazione con questi sistemi avviene sempre attraverso l'utilizzo dello standard *OAuth2* implementato dall'autenticazione del gruppo INFN AAI.

## 4.1 Workflow Orchestrator

Come descritto nella sezione 3 la gestione del ciclo di vita di un *workflow* assume un ruolo fondamentale all'interno del software. Invece di scrivere *ex novo* un componente per la gestione dei workflow si è deciso di esplorare soluzioni *open source* già presenti sul mercato. Avendo il vincolo di trovare una soluzione software basata sullo standard BPMN la scelta è ricaduta sul prodotto di *Camunda*<sup>6</sup>. Camunda è un *process orchestrator* scritto in Java ed è facilmente integrabile all'interno del *framework* di *Spring* usato nel *back-end*; inoltre mette a disposizione dell'utente un *tool* grafico con tutti gli strumenti per disegnare, in maniera agile, un nuovo *workflow*. All'interno della libreria *software* di Camunda è presente il *Camunda Engine* il quale è in grado di interpretare il *workflow* BPMN in maniera automatica, mandando avanti in processo in base all'input che riceve. All'interno del *back-end* è stato costruito un *wrapper* alla libreria di camunda (figura 4): tramite questa astrazione è possibile, nel futuro, cambiare *orchestrator* senza dovere sconvolgere il *core* dell'applicazione.

## 4.2 Implementazione Software

Per lo sviluppo del software sono stati utilizzati due diversi linguaggi di programmazione: *Java* e *TypeScript* rispettivamente utilizzati per la creazione del *Back-end* e del *Front-end*.

### 4.2.1 Entity

All'interno del software è stata definita la classe astratta *Entity* che rappresenta gli oggetti per cui vengono effettuate le richieste (laboratori, aule, ecc...). Per utilizzare i ruoli e le autorizzazioni presenti nel sistema IAM dell'INFN, le entità vengono definite dapprima all'interno dello strumento software di *Godiva*<sup>7</sup>. Su Godiva un'entità è rappresentata come un nodo di una struttura ad albero. La struttura ad albero viene utilizzata per definire in maniera gerarchica i livelli autorizzativi e approvativi delle identità. Su questi speciali nodi vengono definiti dei metadati per indicare che tali nodi saranno utilizzati come entità all'interno di Booking; successivamente, mediante un *import*, verranno importati nel database dell'applicazione.

---

<sup>6</sup><https://camunda.com/>

<sup>7</sup><https://wiki.infn.it/cn/ccr/aai/howto/godiva/home>

#### 4.2.2 Configuration

Ad ogni *entity* deve essere associata una *Configuration*. Una configurazione di un'entità, racchiude le informazioni richieste che servono per configurare l'entità. La necessità di definire queste configurazioni serve per associare ad ogni entità dei campi specifici in maniera tale da rendere la configurazione delle identità indipendenti tra di loro: la prenotazione di una *facility* richiede la configurazione di un indirizzo IP per l'invio di comandi a dei PLC, mentre la prenotazione di un'aula non ha bisogno di questa esigenza. A queste configurazioni è anche collegato il workflow che verrà utilizzato per gestire il processo legato alla prenotazione ossia al file che ha la definizione del *workflow* nello standard BPMN.

#### 4.2.3 Booking Request

Quando verrà inizializzata una nuova richiesta di prenotazione tutte le informazioni su tale richiesta verranno inserite all'interno dell'oggetto *BookingRequest*. In questa struttura saranno presenti tutte le informazioni legate alla richiesta come l'ID dell'entità, i periodi di prenotazione e l'ID dell'istanza del *workflow* (processo BPMN) associata alla prenotazione. Ogni prenotazione avrà la sua istanza e ogni istanza farà riferimento al *workflow* associato all'entità.

### 4.3 Gestione della richiesta

Prima di creare la richiesta verrà controllata che l'entità da associare alla richiesta esista all'interno del database applicativo (vedi sezione 4.2.1), successivamente verranno richiamate le classi per la validazione della richiesta in base al *template* di configurazione. Una volta verificata la disponibilità della risorsa verrà creata la richiesta alla quale saranno associati l'ID dell'istanza del workflow da eseguire e l'ID dell'utente che ha effettuato la prenotazione. In figura 5 viene mostrato il *sequence diagram* che mette in evidenza l'interazione tra i vari componenti del software.

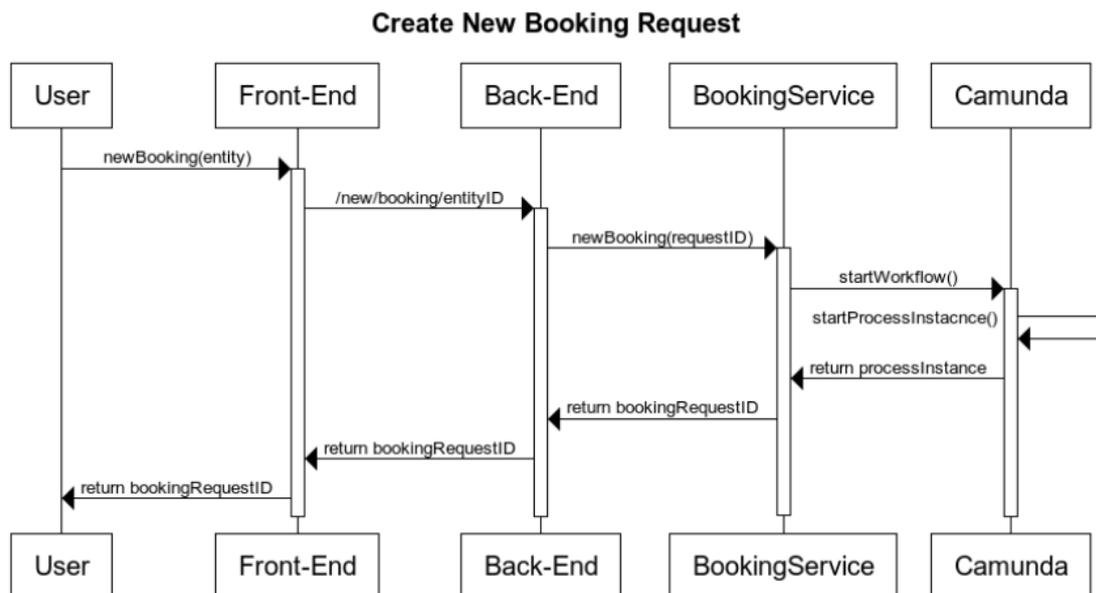


Figura 5: Creazione nuova richiesta di booking.

La figura 6, invece, illustra i passaggi che mostrano all'utente collegato lo stato corrente della richiesta. In base ai ruoli dell'utente verranno restituiti gli stati (o stato) che l'utente può lavorare e/o vedere.

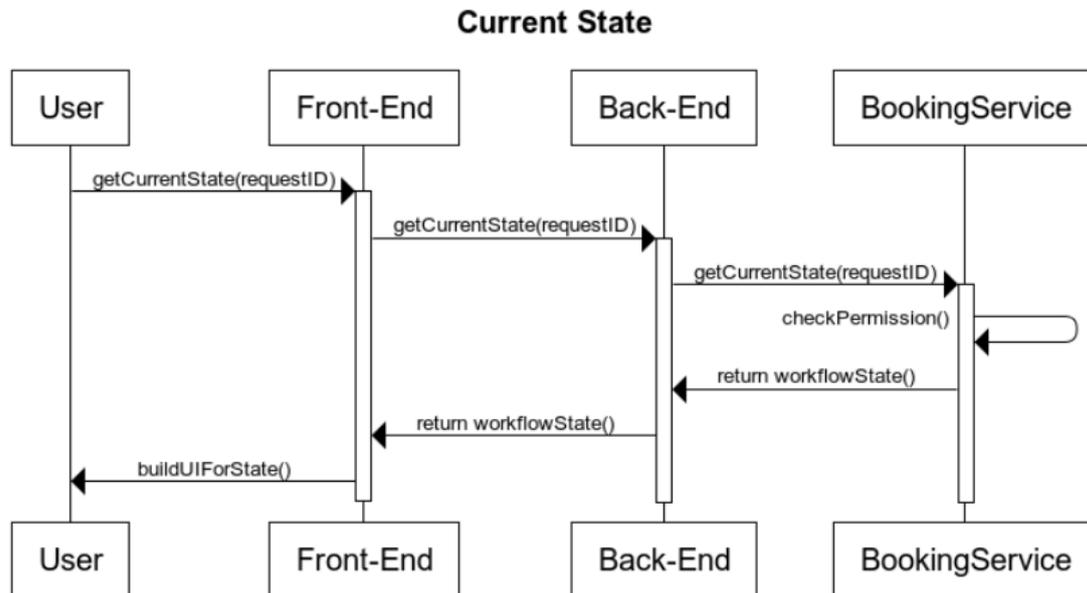


Figura 6: Stato corrente della richiesta in base all'utente.

L'avanzamento della richiesta come anticipato nelle sezioni precedenti è condizionato da determinati step approvativi ossia da stati che rimangono in attesa di essere completati da identità che hanno determinati ruoli. La figura 7 mette in evidenza le operazioni che vengono effettuate quando un utente prova ad avanzare ad uno stato successivo.

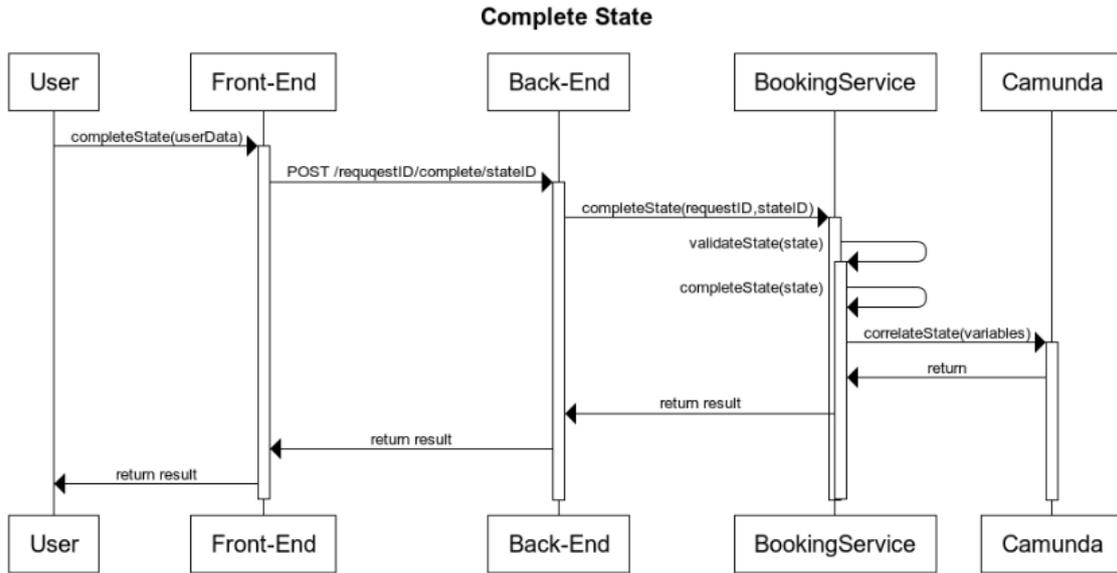


Figura 7: Avanzamento workflow

#### 4.4 Visualizzazione degli stati

Per quanto riguarda l'interfaccia grafica, la visualizzazione degli stati viene gestita in maniera dinamica. Per rendere l'interfaccia grafica il più astratta possibile, il *front-end* elabora la visualizzazione degli stati in base al loro tipo senza dovere scrivere del codice *hard-coded* per ogni singolo stato. Sono stati individuati due tipologie di stati: il primo tipo è di tipo compilativo (figura 8), mentre il secondo di tipo approvativo (figura 9). Nella prima tipologia possono essere inserite delle *form* (costruite in maniera dinamica nel *front-end*) da inserire (tramite codice identificativo) all'interno di uno stato. Al momento della visualizzazione di quello specifico stato, il *front-end* elaborerà la *form* definita in precedenza. Nella seconda tipologia invece, lo stato rappresenta un passaggio decisionale di approvazione o negazione degli stati precedenti o più in generale dello storico degli stati della richiesta. I dati inseriti nella *form* verranno convertiti in variabili dal *back-end* e fatto l'*inject* all'interno dell'istanza del *workflow*. In questo modo le variabili possono essere utilizzate negli altri stati del processo.

2023-02-06 - 2023-02-12 Info

Submitter: Giovanni Lorenzo Napoleoni  
BTF ( Line2 )

Booking Request Questionnaire **Waiting** | Management Request Approval **Next Step** | Team Creation **Next Step** | Waiting Team Members **Next Step** | Personnel Request Approval **Next Step**

**Selected State**

**BTF**

EXPERIMENT Proposal name \*

Involved Institutions/Industries/Countries \*

Experiment motivation related to requested beam time, scientific discipline, research area and purposes: \*

Proposal category (mark the right one): \*

New - If you are submitting this experiment plan for the first time

Needed experimental area (BTFEH1, BTFEH2) \*

Device under test description (please describe possible hazards): \*

Owned setup to be put in experimental hall, brief description (please describe possible hazards related to) \*

Owned setup to be put in control room, brief description (please describe possible hazards) \*

Figura 8: Stato con form

**Selected State**

Approve booking request

Add message to approval state

Il questionario sottomesso è completo. |

X Refuse ✓ Approve

**States History**

Booking Request Questionnaire

Approval message: completedState

**BTF**

experiment\_mouvaouon: 1  
 Proposal\_category: New  
 Needed\_experimental: 1  
 Device\_under\_test: 1  
 Owned\_setup: 1  
 Owned Setup To Be Put In Control Room: 1  
 Btf Needed Setup: 1  
 Time Needed For Experiment: 1  
 Time\_needed\_contiguous\_days: 1  
 Time\_needed\_roll\_in: 1  
 Time\_needed\_roll\_out: 1  
 Measure\_description\_file:  
 Screenshot 2022-10-12 at 11:50:37.png

↓

Figura 9: Stato approvativo

## 5 Caso d'uso: Beam Test Facility

In questa sezione verrà analizzato il *workflow* per la prenotazione della *Beam Test Facility*<sup>8</sup> (BTF), un'infrastruttura di ricerca presente all'interno dei Laboratori Nazionali di Frascati, progettata con una ampia gamma di parametri di fascio selezionabili per condurre esperimenti e test sotto fascio per lo sviluppo di nuove tecniche di rivelazione di particelle cariche e neutre<sup>9</sup>. Il *workflow* della prenotazione della BTF presenta un certo grado di complessità dimostrando la versatilità del software che è stato sviluppato.

La BTF attualmente possiede due linee utilizzabili per condurre gli esperimenti, entrambe prenotabili con richieste separate, a slot di una settimana. La prenotazione di queste risorse richiede la compilazione di un questionario riguardante l'esperimento; il questionario verrà successivamente esaminato dal responsabile BTF e dal PAC per decidere la fattibilità tecnica e l'interesse scientifico. Nel caso in cui la richiesta superasse la prima approvazione l'utente sarà incaricato di indicare gli altri membri del team che parteciperanno all'esperimento. Una volta indicati i componenti del team ognuno di questi dovrà sottoporre un pdf di nullaosta per l'accesso alla BTF che dovrà essere approvato da due segreterie, del personale ed amministrativa, affinché venga ritenuto idoneo per lo svolgimento dell'esperimento e per l'accesso sul territorio. Una volta superata l'approvazione del team ad ogni membro autorizzato verrà creato un ruolo su Godiva che gli garantirà l'accesso alla *facility*. Infine, verrà comunicato alle persone preposte alla sicurezza convenzionale, alla radioprotezione, ai dirigenti responsabili e alle altre figure interessate, la lista dei membri del team autorizzati ad accedere alla campagna di misura presso la BTF. La complessità di questo workflow è mostrata in figura 10. All'interno del processo BPMN sono incluse tutte le possibili ramificazioni del processo: negazioni, invio *reminder*, gestione errori, cambi date e scadenze.

---

<sup>8</sup><https://w3.lnf.infn.it/acceleratori/btf/>

<sup>9</sup>[btf.lnf.infn.it](http://btf.lnf.infn.it)

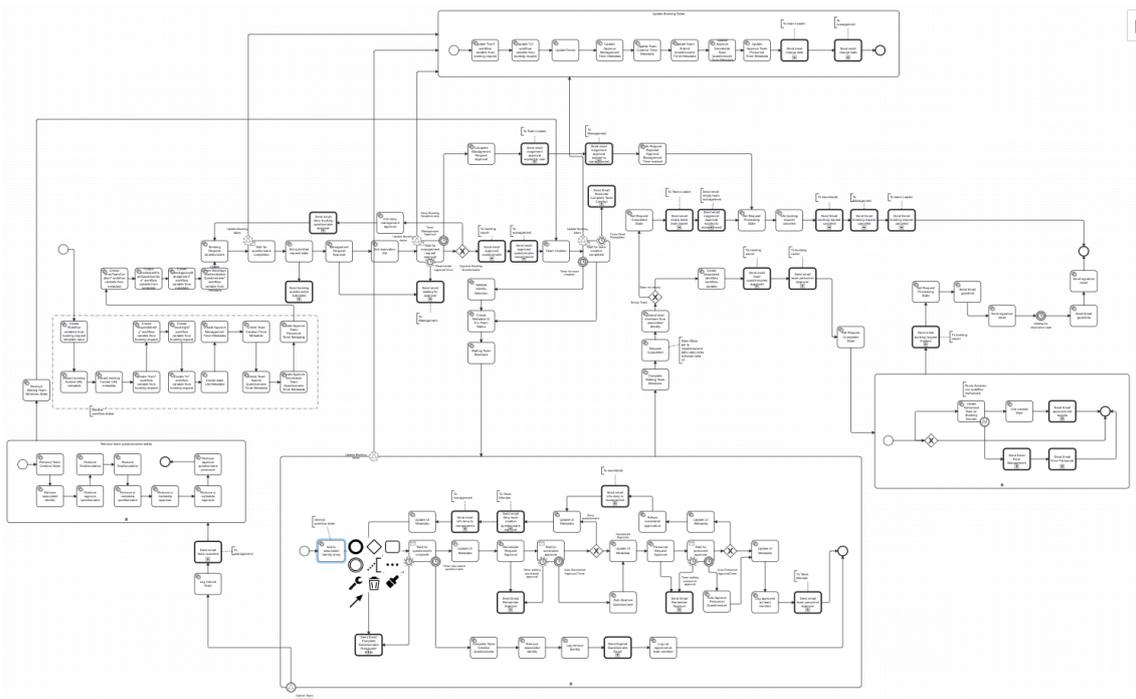


Figura 10: Workflow BPM per la BTF

Prima dell'impiego del software di booking, questo processo veniva gestito totalmente in maniera manuale e cartacea. Al momento il software è all'attivo da più di un anno ed ha permesso un notevole aumento della richiesta di accesso alla BTF coinvolgendo istituzioni esterne ad LNF, nazionali ed internazionali.

## 5.1 Controllo PLC

Per la BTF, oltre al processo di prenotazione, è stato progettato e sviluppato un sistema automatizzato per il controllo dei macchinari all'interno del laboratorio attraverso l'utilizzo di *Programmable Logic Controller*<sup>10</sup>(PLC). I PLC svolgono diverse funzioni, tra cui acquisizione di segnali da sensori e dispositivi di input, esecuzione di calcoli, logica programmata e attuazione di azioni attraverso dispositivi di output come motori, valvole o altri attuatori. Durante la fase di progettazione sono emerse due problematiche fondamentali per lo sviluppo del software. La prima riguardava l'assenza di librerie che permettessero la comunicazione con la specifica marca di PLC presenti all'INFN e la seconda la necessità di effettuare dei test senza interfacciarsi fisicamente con un PLC. Per la comunicazione con i PLC è stato quindi utilizzato un approccio di programmazione ad alto livello attraverso l'impiego del protocollo di comunicazione seriale MODBUS<sup>11</sup>. È stato così possibile creare un servizio al cui interno venissero definiti i metodi per creare ed inviare i comandi ad uno specifico PLC, fornendo la porta e l'indirizzo di que-

<sup>10</sup><https://journals.aijr.org/index.php/ajgr/article/view/185>

<sup>11</sup><https://modbus.org/>

st'ultimo. Questo servizio può essere integrato all'interno degli stati definiti nel workflow BPMN, per esempio per accendere delle macchine, in maniera programmatica, nel giorno di accesso alla BTF.

## **6 Acknowledgements**

Si ringrazia particolarmente il supporto della direzione nella fase di implementazione di questo sviluppo e per l'estrema dedizione dimostrata dal personale delle segreterie amministrativa della Divisione Acceleratori e del personale degli LNF, fondamentali allo sviluppo del protocollo attuato in BTF e nella gestione d'emergenza nella fase di *tuning* operativo del software. Si ringraziano inoltre gli utenti ed il personale BTF che si sono sottoposti all'uso del servizio nella fase di beta *testing*, fondamentale per accordare lo strumento di Booking ad una *user experience* più elevata.