# LISTA ITERAZIONI

ITERAZIONE 0

**Inizio:** 1/12/22  
**Fine:** 14/12/22

**Descrizione:** Descrizione iniziale del problema e costruzioni dei primi documenti e diagrammi per la rappresentazione ad alto livello di requisiti e struttura architetturale della soluzione.

**Descrizione operazioni:**

* Descrizione del problema, del contesto di lavoro e del risultato che l’algoritmo dovrà essere in grado di produrre dati degli input.
* Descrizione ed elencazione delle user stories
* Definizione dei requisiti
* Costruzione tabella dei requisiti
* Costruzione del diagramma dei casi d’uso
* Definizione delle tecnologie da utilizzare
* Costruzione del diagramma di deploy

ITERAZIONE 1

**Inizio:** 14/12/22  
**Fine:** 3/01/23

**Descrizione:** Suddivisione a medio-alto livello del sistema. I diagrammi creati in questa iterazione ci hanno permesso di spezzare il problema iniziale in sotto problemi e quindi definire una prima strutturazione del codice server e client in base ai sottocomponenti ricavate.

**Descrizione operazioni:**

* Suddivisione del problema in sottocomponenti in funzione dei casi d’uso che ciascun componente deve essere in grado di gestire.
* Costruzione del diagramma a singolo componente
* Costruzione del diagramma con sottocomponenti e i loro collegamenti
* Creazione del diagramma dei package
* Creazione della struttura dei package nel progetto Spring
* Definizione più dettagliata sul funzionamento dell’applicazione e dell’algoritmo (sono stati eseguiti degli aggiornamenti ai diagrammi dei casi d’uso e di deploy).
* Creazione del diagramma state machine per la descrizione degli stati macchina
* Creazione del diagramma activity per la descrizione del flusso di gestione dei lotti e della pianificazione.
* Creazione della struttura iniziale dell’applicazione client + creazione sistema mock di autenticazione.

ITERAZIONE 2

**Inizio**: 4/01/23  
**Fine:** 7/01/23

**Descrizione:**

È stata implementato la componente che gestisce le notifiche. La componente offre due interfacce, *sendOperai* e *sendManager*: queste prendono in input il titolo e il corpo della notifica da inviare e utilizzano il servizio **Firebase** per spedirla ai dispositivi i cui token sono salvati in delle apposite liste. Le componenti che necessiteranno di inviare delle notifiche dovranno utilizzare questa componente tramite i due metodi. Durante gli incontri avvenuti il 4/01 ed il 6/01 è stato modificato il Package Diagram (aggiunti package relativi all’implementazione della componente notifiche). È stato implementato sull’applicativo server il codice della componente notifiche ed è stata implementata la ricezione notifiche sul client flutter.

**Descrizione operazioni:**

* Setup progetto firebase
* Creazioni metodi per invio notifiche da server
* Frontend notifiche
* Testing
* Aggiornamento Diagrammi
* Analisi Statica

ITERAZIONE 3

**Inizio**: 7/01/23  
**Fine:** 10/01/23

**Descrizione:**

In questa iterazione è stata implementata la componente: Gestione Macchine. Sono stati creati i controller/service e models per le API che gestiscono l’aggiunta e rimozione delle macchine dal server, il recupero dello stato di ciascuna macchina e l’aggiornamento dello stato delle macchine. È stata creata la componente mock che simula il comportamento delle macchine tramite l’utilizzo di thread. In questo modo si è potuto testare il funzionamento delle API in tempo reale. Per quanto riguarda il frontend sono state create le pagine per la creazione, cancellazione, visualizzazione delle macchine sul server.

**Descrizione operazioni:**

* API aggiunta e rimozione macchine
* API recupero stato macchina
* API aggiornamento stato macchina
* Creazione componente mock
* Frontend macchine
* Testing
* Aggiornamento Diagrammi
* Analisi Statica

ITERAZIONE 4

**Inizio**: 10/01/23  
**Fine:** 11/01/23

**Descrizione:**

In questa iterazione è stata implementata la componente: Gestione Log. Sono state create le API per la visualizzazione dei log per ciascuna macchina e per l’invio al server dei log macchina. Sul frontend sono state create le pagine per la visualizzazione della lista di log per ciascuna macchina e di tutti i suoi dettagli.

**Descrizione operazioni:**

* API visualizzazione log
* API log al server
* Frontend visualizzazione Log
* Testing
* Aggiornamento Diagrammi
* Analisi Statica

ITERAZIONE 5

**Inizio**: 11/01/23  
**Fine:** 14/01/23

**Descrizione:**   
In questa iterazioni sono state implementate le API per la gestione dei lotti, quindi l’aggiunta, la rimozione e la modifica ed infine la visualizzazione della lista di lotti non ancora schedulati.   
Sul frontend sono stati creati la pagina per l’inserimento dei lotti, la pagina per la visualizzazione/rimozione/modifica dei lotti non ancora schedulati.

**Descrizione operazioni:**

* API aggiunta lotti
* API rimozione lotti
* API visualizzazione
* Fronted pagine lotti
* Testing
* Aggiornamento Diagrammi
* Analisi Statica

ITERAZIONE 6

**Inizio**: 15/01/23  
**Fine:** 18/01/23

**Descrizione:**

In questa iterazione è stata creata la componente: Gestione Pianificazione ed è stato implementato l’algoritmo di schedulazione. Sono state create le API per l’avvio della pianificazione, la visualizzazione dei risultati di una pianificazione e della conferma da parte del manager. È stato fatto lo studio di complessità dell’algoritmo e sono stati testati gli scenari più importanti di funzionamento. Sul frontend è stata aggiunta la pagina per l’avvio della pianificazione e per la visualizzazione dei risultati dopo il completamento della pianificazione. È inoltre stato aggiornato il sistema di notifiche per gestire l’arrivo della notifica di fine pianificazione.

**Descrizione operazioni:**

* API avvio pianificazione
* API visualizzazione pianificazione
* API conferma pianificazione
* Studio Complessità
* Frontend pianificazione
* Frontend Refactoring notifiche
* Testing
* Aggiornamento Diagrammi
* Analisi Statica

ITERAZIONE 7

**Inizio**: 20/01/23  
**Fine:** 18/01/23

**Descrizione:**

In questa fase sono state eseguite le operazioni di refactoring e testing su tutti i componenti sia nel frontend che sul backend. Sono anche stati aggiornati i diagrammi UML per migliorare la chiarezza e l’ordine.   
Inoltre, è anche stato eseguita l’installazione del server “pianificatore” sulla piattaforma AWS.

**Descrizione operazioni:**

* Refactoring
* Unit Test
* Aggiornamento Diagrammi
* Installazione serve
* Testing