Sprint 1:  
  
  
Vocabolario:

|  |  |
| --- | --- |
| Termine | Significato attribuito |
| Microservizio | Componente software progettato per svolgere una specifica funzione del sistema. Ogni microservizio comunica con gli altri tramite messaggi, rendendo il sistema flessibile e scalabile. |
| GUI (Graphical User Interface) | Interfaccia grafica utente accessibile via web, che consente di visualizzare in tempo reale lo stato della stiva e interagire in modo intuitivo con il sistema. |
| WebGUI |  |
| Bounded Context |  |
| IOPort | Punto fisico (porta) attraverso il quale i contenitori dei prodotti entrano o escono dalla nave. È il punto in cui il sonar rileva la presenza di un prodotto. |
| Sonar | Sensore a ultrasuoni che misura la distanza tra sé e un oggetto. Nel nostro sistema serve per rilevare se un contenitore è presente all’IOPort. |
| DDRobot | è un robot che utilizza due motori indipendenti per muovere le ruote o i cingoli. |
| PID (Product Identifier) | Numero intero univoco assegnato a ciascun prodotto registrato, usato per tracciarne l'identità all'interno del sistema. |
| Slot | Spazio fisico nella stiva della nave dove può essere posizionato un contenitore. Esistono 4 slot disponibili; uno è sempre occupato (slot5). |
| Cargorobot | Robot mobile autonomo (a guida differenziale) incaricato di trasportare i contenitori dall’IOPort fino allo slot assegnato e poi tornare alla posizione HOME. |
| Stiva | Area rettangolare della nave in cui i contenitori vengono caricati. Contiene gli slot e l’IOPort. |
| ProductService | Microservizio che gestisce la registrazione dei prodotti. Verifica i dati e assegna un PID univoco |
| CargoService | Microservizio che riceve richieste di carico, controlla i vincoli, assegna gli slot e coordina il caricamento tramite cargorobot. |
| SonarService | Microservizio che rileva la presenza di un contenitore all’IOPort tramite i dati forniti dal sonar. |
| DFREE | Distanza soglia usata dal sonar: se la distanza misurata è maggiore di DFREE per 3 secondi, si ipotizza un malfunzionamento del sensore. |
| MaxLoad | Peso massimo complessivo che la nave può sopportare. Il sistema rifiuta richieste che farebbero superare questo limite. |
| Worker | Persona che colloca fisicamente i contenitori sull’IOPort dopo che sono stati registrati. |
| Sistema logico di riferimento | Rappresentazione concettuale dell’intero sistema, con attori, componenti e interazioni, usata come base per l’architettura e la progettazione tecnica. |
| attore | Entità che svolge un ruolo attivo nel sistema, eseguendo azioni e comunicando con gli altri attori attraverso messaggi |
| Linguaggio QAK | Linguaggio modellistico usato per descrivere e simulare il comportamento dei componenti del sistema come “attori” |

GOAL dello Sprint 1:

1. Enunciazione esplicita dei requisiti del cargoservice
2. Analisi dei requisiti enunciati
3. Definizione dell’architettura logica con modello eseguibile in qak e mockup dei servizi non ancora implementati (sonarservice, webgui)
4. Progetto e realizzazione
5. Enunciazione esplicita dei requisiti:

Per automatizzare le operazioni di carico nella stiva di una nave, il sistema software si articola in microservizi autonomi ma cooperanti, tra cui, cargoservice.

Esso rappresenta il nucleo centrale del processo:

* Riceve le richieste di carico
* Valuta le richieste di carico
* Assegna gli slot ai prodotti
* Riceve le informazioni dal sonar
* Coordina i movimenti del DDrobot

Il team di sviluppo si propone, in questa fase, di analizzare in modo tecnico i requisiti del microservizio da sviluppare cargoservice.

Analisi dei requisiti enunciati:

Esplicitiamo le funzionalità, comportamenti e vincoli che il sistema deve necessariamente e logicamente rispettare.

RF1. **Gestione richiesta di carico**: il sistema deve gestire correttamente le richieste di carico ricevute.

RF1.1 **Controllo esistenza prodotto**:

il sistema chiede al product-service se il prodotto (PID) è già registrato e attende che esso risponda.

Se lo è viene restituito il peso.

RF1.2 **Validazione della richiesta**:

Una volta che si è superato il controllo dell’esistenza del prodotto.

il sistema rifiuta la richiesta se il peso del prodotto supera il MaxLoad o se la stiva è già piena (non ci sono slot disponibili).

RF1.3 **Attesa slot**:

se la richiesta con PID viene approvata, si assegna al prodotto uno slot libero. Il sistema entra in attesa del rilevamento del contenitore all’IOPort e sospende la ricezione di nuove richieste fino al completamento.

RF1.4. Rilevamento contenitore:

il sistema attende che il sonar mandi

RF1.5. Trasporto contenitore:

il sistema comanda il DDRobot affinché trasporti il contenitore dall’IOPort allo slot assegnato.

Al termine, il robot torna alla postazione di default HOME.

Concluso il processo, il sistema riprende la gestione delle richieste.

RF2. Visualizzazione stiva:

il sistema espone tramite una GUI lo stato attuale della stiva, con aggiornamenti dinamici.

RF3. Gestione anomalia sonar:

il sistema gestisce i malfunzionamenti del sonar fermando il DDrobot momentaneamente.

Riprende le attività una volta ripristinata la condizione corretta.

Requisiti non funzionali (vincoli su prestazioni, sicurezza, usabilità):

* AFFIDABILITÀ:
* Il sistema deve essere in grado di rilevare e gestire anomalie hardware (es. malfunzionamento del sonar), garantendo una risposta controllata senza crash e blocchi irreversibili
* REATTIVITÀ / TEMPISTICHE:
* Il sistema deve reagire entro un tempo T che sia ‘ragionevole’:

viene considerata ragionevole un tempo di circa tre secondi.

* DISPONIBILITÀ:
* Il servizio deve essere sempre attivo tranne durante l’elaborazione di una richiesta o un’anomalia. Deve riprendere autonomamente appena possibile.
* USABILITÀ:
* La GUI deve essere intuitiva e aggiornata dinamicamente.
* SCALABILITÀ TECNICA:

Architettura a microservizi indipendenti, scalabili e distribuibili su più nodi computazionali.

* MODULARITÀ / MANUTENIBILITÀ:

Ogni componente deve essere testabile e aggiornabile separatamente.

Vincoli fisici o tecnici (legati all’hardware):

* Il PID deve essere univoco
* Ci sono 5 slot, di cui uno sempre occupato.
* Gli slot una volta liberi sono riutilizzabili.
* I prodotti devono essere inseriti di dimensioni predefinite e registrate
* Il robot mobile è di tipo Differential Drive, ovvero si muove piano, ritorna sempre alla posizione HOME e riceve comandi da cargoservice.
* Il robot può trasportare un massimo peso MaxLoad, quindi ogni richiesta di carico deve essere validata in base al peso totale attuale sommato a quello del nuovo carico.
* L’area della stiva è rettangolare, piatta e dotata di un solo punto di Input/Output (IOPort)
* Il sonar rileva un contenitore se registra una distanza D minore di DFREE/2 per almeno un tempo T. Se D > DFREE/2, registra un malfunzionamento.

1. ANALISI DEL PROBLEMA:

Il gruppo di sviluppo propone la realizzazione di un sistema software distribuito, articolato su più nodi computazionali, con l’obiettivo di automatizzare le operazioni di carico dei contenitori nella stiva della nave.

Il sistema è composto da microservizi indipendenti, che devono cooperare e comunicare in modo coordinato, nonostante siano soggetti a eventi asincroni provenienti dal mondo fisico.

Per raggiungere questo obiettivo, il sistema deve affrontare e risolvere una serie di problematiche chiave, tra cui:

1. **La gestione sequenziale (in modo ordinato e non concorrente) delle richieste:**

mentre il sistema sta eseguendo una richiesta di carico che ha accettato, non può considerarne altre, ovvero, mentre si attende che il prodotto venga posto all’IOPort e mentre il robot lo pone nello slot assegnato.

1. **La sincronizzazione di eventi reali non deterministici:** 
   * La richiesta di carico da parte della compagnia,
   * La presenza fisica del contenitore all’IOPort,
   * Il caricamento del contenitore nello slot da parte del robot.
   * La risposta da parte del productservice (microservizio fornito)

il sistema dipende dall’avvenimento di eventi che non avvengono in ordine predeterminato e richiedono una gestione a stati del cargoservice, con sospensione temporanea dell’elaborazione finché una fase non è completata.

1. **Controllo del robot differenziale DDRobot:**

il robot utilizzato è fornito esternamente, la sua interfaccia è nota ma non modificabile.

Il sistema deve essere in grado di guidare il robot con comandi precisi per raggiungere partendo da un qualsiasi punto della stiva:

* + Il posizionamento di default HOME
  + La IOPort
  + I vari slot

Inoltre, il sistema deve poter inviare dei comandi per fermare e far ripartire il robot in caso di anomalie.

1. **la gestione di anomalie hardware:**

la presenza del contenitore all’IOPort è rilevata da un sonar, che può non rispondere correttamente o rilevare valori errati.

Il sistema deve prevedere l’attesa della risoluzione del guasto per poter continuare.

1. **il monitoraggio dinamico dello stato della stiva:**

il sistema deve mantenere lo stato coerente e aggiornato lo stato dei 4 slot utilizzabili, considerando:

* + assegnazione dinamica degli slot,
  + occupazione di uno slot dopo il caricamento del prodotto.

La WebGUI deve ricevere questi aggiornamenti in tempo reale.

1. Presentazione dei modelli eseguibili: