|  |  |  |
| --- | --- | --- |
| **Glossario** | **Todos** |  |
| **Definição problema** | **Cast** | **feito** |
| **Stake/necessidades** | **JT** | **Feito** |
| **Caracs. do software** | **Cast** | **feito** |
| **Reqs. Funcs(UML USE)** | **JT** | **+---** |
| **Reqs. Ñ Funcs** | **Rui** | **+-** |
| **Restricoes Design** | **Rui** | **??** |
|  |  |  |

**NEGRITO= glossário**

**Stakeholders**

Os nossos stakeholders são investigadores do **ISR-UC**, mais concretamente aqueles que têm um papel activo em trabalhos de graduação e investigação no **LRM**, um dos números laboratórios deste instituto. Estes Investigadores são maioritariamente ex-alunos bolseiros, alunos em fase de terminar o curso e os seus coordenadores.

Além destes, o desenvolvimento deste software tem também interesse para o professor da cadeira de **ES** e para os alunos que o desenvolvem.

**Necessidades destes ^**  
 Devido ao uso regular de robôs pelos interessados, surge a necessidade de desenvolver um software que permite registar automaticamente a utilização destas máquinas, baseada nos dados de telemetria enviados pelas mesmas para um servidor remoto, através de **TCP/IP**. Além deste registo, é também requerido pelos investigadores que forneçamos uma maneira de **teleoperar** os robôs até onde são necessários, para experiências e/ou missões.

No caso do professor e dos desenvolvedores do projecto, este software é necessário para a avaliação da componente prática da cadeira de **ES**.

**Requisitos funcionais:**

* Ter uma aplicação que funcione como servidor;
* Receber informação sobre os robôs;
* Gerir a frota de robôs
* Adicionar modelos à frota de robos
* Eliminar um tipo de robô, caso não exista nenhum robô desse tipo
* Guardar dados do robô:
  + Guardar Data de aquisição/construção
  + Guardar capacidade das baterias em **Wh**
  + Guardar parâmetros para poder calcular a potencia elétrica consumida
  + FACULTATIVO. Guardar uma descrição do robo
* Calcular a potência elétrica consumida pelo robô
* Permitir que o **user** defina a carga inicial das baterias (em %)
* Permitir que seja atualizada manualmente a carga existente nas baterias de cada
* Configuração de tipos de tarefas executadas pelos robôs
* Adicionar novos tipos de tarefas
* Guardar uma descrição para cada tipo de tarefa
* Apresentar o estado dos robôs que se encontrem ligados por **TCP IP**
* Estado Inclui:
  + Tarefa que está a executar
  + Tempo que a está a executar (mm:ss)
  + Distancia percorrida desde o inicio da tarefa
  + Velocidades linear e angular
  + Consumo acumulado em **Wh** nessa tarefa
  + Estimativa de carga existente nas baterias (% e **Wh)**
  + Tempo estimado até ser necessário recarregar
* Apresentar um alerta visível quando bateria de um robô for <10%
* Guardar num ficheiro (forma persistente) um histórico da utilização do robô:
  + Tempo total de operação em horas
  + Distancia total percorrida
  + Consumo total de energia
  + Numero de ciclos de carga e descarga
  + Numero de ciclos completos de carga
  + Histograma com numero de vezes que realizou cada tipo tarefa
  + Histograma com tempo total de operação em cada tipo tarefa
  + Histograma com distancia total percorrida em cada tipo tarefa
* Identificação unívoca dos robôs

**Requisitos não funcional**

* Interface Intuitiva
* Utilização de atalhos de teclado
* Confiabilidade na transmissão de informação
* Boa visibilidade no alerta
* Facilidade de análise e partilha do ficheiro exportado
* Uso privilegiado do rato

**Restrições de Design**

* Protocolo utilizado na conexão ser TCP/IP
* Escrito em C++
* Porto usado na conexão ser o porto 10227
* Formato da informação enviada pelos robôs ser ASCII
* Formato das mensagens recebidas
* Tempo em ms
* Consumo em Wh
* Ficheiro exportado ser CSV

**Restrições de Processo**

* Apenas ter acesso aos robôs na fase final do projecto