



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
CENTRO TECNOLÓGICO  
DEPARTAMENTO DE INFORMÁTICA

INF09328 – PROGRAMAÇÃO I

CURSO: CIÊNCIA DA COMPUTAÇÃO

PROFESSOR: JORDANA SARMENGI SALAMON

## Especificação do Trabalho Prático

Este trabalho deve ser resolvido em dupla, segundo o paradigma funcional, utilizando a linguagem Python e o ambiente IDLE. O produto a ser entregue até o dia **01/07/19** para o e-mail [jssalamon@inf.ufes.br](mailto:jssalamon@inf.ufes.br) deve ser um código-fonte (.py) e um relatório (.pdf), compactados num arquivo com o nome TProg1-NOME1-SOBRENOME1-NOME2-SOBRENOME2.rar (ou .tar.gz). Use o mesmo padrão de nome no assunto do e mail.

### LEIA E SIGA ATENTAMENTE AS INSTRUÇÕES a seguir:

- Deve-se incluir como comentário no início do arquivo de código uma descrição do seu conteúdo geral, os autores, a disciplina e o que mais for relevante para identificação futura.
- Ao importar funções de algum módulo, utilize a opção de importação específica.
- Cada função deve ter um comentário com sua descrição, dados de entrada e saída. Na descrição, diga se a função é global ou local, paramétrica ou não, e porquê.
- Lembre-se de usar nomes significativos de funções e parâmetros.
- Lembre-se de modularizar quando necessário e de realizar a validação dos dados.
- É importante que seja usado o método de resolução de problemas apresentado na disciplina, que inclui as seguintes etapas:
  - Compreensão do problema
  - Planejamento da solução
    - O plano de testes deve ser incluído no relatório, e não no código
  - Desenvolvimento
    - Os testes propriamente ditos devem estar automatizados no arquivo de código
  - Avaliação da solução obtida
    - Apresente os pontos fortes e fracos.
  - Ao longo desses passos, devem ser coletados dados para o relatório a ser elaborado com uma descrição do processo desenvolvido desde o início até o final desse projeto.
- Este trabalho deve ser realizado utilizando somente estruturas de tuplas e listas. Não serão aceitas estruturas do tipo dicionário.
- Plágios não serão tolerados. Trabalhos iguais receberão nota ZERO.

O Relatório deve ter início, meio e fim, ou seja, introdução, desenvolvimento e conclusão.

- Introdução: Inclua um estudo sobre o problema (compreensão do problema).
- Desenvolvimento: inclua as etapas de planejamento e desenvolvimento (sem código).
- Conclusão: Inclua a avaliação da solução e dos resultados obtidos, e outras considerações que achar necessárias.

### Auxílio ao resgate de vítimas por robôs em ambientes acidentados

É crescente a utilização de robôs no resgate de feridos em acidentes, mas a tecnologia tem potencial para ampliar esse uso. Além de maior precisão, podendo chegar mais rapidamente aos feridos, os robôs podem evitar riscos para os bombeiros e pessoal de resgate. É por isto que engenheiros do mundo todo procuram desenvolver robôs de resgate cada vez mais precisos e com maior autonomia.

Um ambiente genérico de simulação de desastres urbanos é construído com base numa rede de computadores. Agentes heterogêneos inteligentes, tais como bombeiros, polícias, vítimas, voluntários, etc., conduzem ações de busca e salvamento (BS) neste mundo virtual.

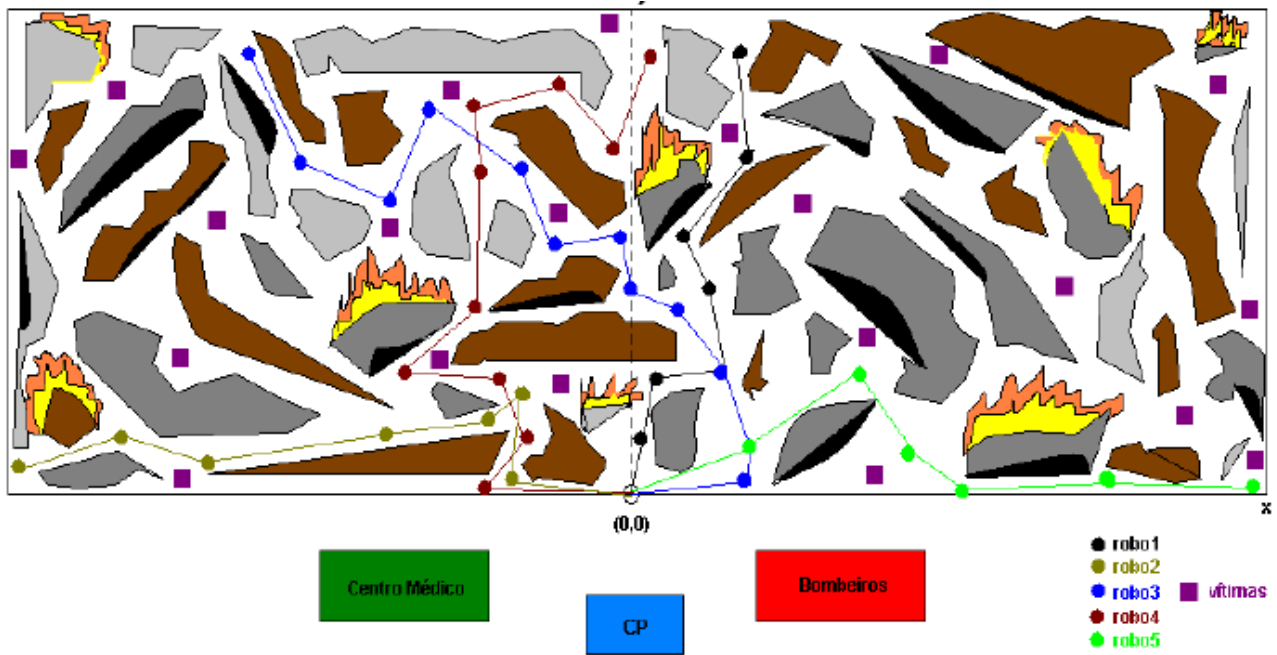
Interfaces com o mundo real, tais como imagens obtidas por helicópteros, sincronizam os dados virtuais com dados reais obtidos através de sensores, como por exemplo, câmaras de vídeo. Interfaces com seres humanos, tais como os assistentes digitais pessoais (ADPs), apoiam o pessoal dos sistemas de proteção civil, as brigadas de salvamento, residentes nas áreas atingidas e voluntários na tomada de decisões que minimizem as consequências do desastre.

Seguindo o contexto acima citado, o problema apresenta um ambiente acidentado (com a presença de escombros) do qual será necessário extrair algumas informações necessárias para que se possa então tomar decisões sobre o resgate das vítimas presentes no ambiente.

O sistema disponível para que se possam retirar as informações necessárias do ambiente é composto por um conjunto de robôs-célula idênticos, os quais têm a função de mapear o ambiente. Fisicamente a implementação desse mapeamento é realizada através do deslocamento relativo de cada robô em relação ao seu ponto de partida.

Cada robô informa periodicamente a uma central de processamento (CP), externa ao ambiente acidentado, o número de vítimas que ele pode identificar no seu ponto de passagem em um determinado instante. O objetivo de cada robô é tentar adentrar o máximo possível ao ambiente para colher a maior quantidade de informações sobre as vítimas. Como o envio das mensagens dos robôs para a central não é simultâneo, a informação resultante do mapeamento é uma lista composta pelos dados de todos os robôs, ordenadas pelo instante de coleta da informação.

A figura a seguir ilustra a situação descrita:



Considere que todos os robôs partem do mesmo ponto inicial: (0,0). As informações enviadas pelos robôs de resgate contêm os seguintes registros: a identificação do robô, o instante em que o robô passou em um determinado ponto da área afetada, o ponto no plano cartesiano referente ao incidente registrado e o número de vítimas presentes na área de visibilidade do robô.

Considerando uma lista com tais informações como os dados de entrada disponíveis, faça um programa em Python que responda às seguintes solicitações:

- Calcular a distância percorrida por um determinado robô ao longo do processo de resgate das vítimas. Considere que a distância total percorrida deve ser calculada como a soma de todas as distâncias entre os pontos de passagem do robô;
- Determine qual dos robôs apresenta o seu último ponto de passagem no terreno de busca que possui a maior distância em relação à origem. Exiba o caminho percorrido pelo robô e o tempo total do percurso;
- Exiba os caminhos percorridos por todos os robôs que entraram no terreno de busca, ordenados crescentemente pela distância total percorrida;
- Forneça a identidade do(s) robô(s) que conseguiu(ram) informar o maior número de vítimas (considerando que não há duplicação de identificação de vítima por um mesmo robô).