Uma imagem com Tipo de letra, Gráficos, design gráfico, captura de ecrã

Descrição gerada automaticamente

**Relatório Sprint B - ALGAV**

**Turma 3DC-Grupo 14**

1211439 - Pedro Costa

1211134 - Pedro Viana

1211436 - David Pinho

1210825 - Pedro Ferreira

**Professor:**

David Freitas, DCF

**Unidade Curricular:**

Algoritmia Avançada

1. **Domínio**

Para a nossa solução, decidimos trabalhar com o campus e as plantas fornecidas em LAPR5 para o projeto onde estão incluídos:

* 4 Edifícios: [a, b, c, d]
* 12 Pisos:
* a: [a1, a2]
* b: [b1, b2, b3]
* c: [c1, c2, c3, c4]
* d: [d1, d2, d3]
* 6 Passagens: [a2, b2]; [b2, c3]; [b2, d3]; [b3, c4]; [c2, d2]; [c3, d3];
* 4 Elevadores: [a, b, c, d]

Uma imagem com texto, captura de ecrã, Tipo de letra, design

Descrição gerada automaticamenteCom os dados acima descritos decidimos criar bases de conhecimento relativamente aos pontos de acesso, entre eles os pisos de cada edifício, os elevadores e a lista de pisos que serve no edifício respetivo, os corredores/passagens entre pisos e as salas de cada piso. Além disto, criamos uma base de conhecimento para as coordenadas dos respetivos pontos de acesso, as dimensões de cada piso e por último uma para as células das matrizes de cada piso, onde está incluído valores para o número da coluna, linha e o valor da célula.

**Uma imagem com texto, captura de ecrã, Tipo de letra, design

Descrição gerada automaticamentebc\_pontosAcesso.pl:**

**liga** -> Ligação entre edifícios

Parâmetros: 1º Edifício; 2º Edifício

**pisos** -> Pisos de um edifício

Uma imagem com texto, captura de ecrã, Tipo de letra, design

Descrição gerada automaticamenteParâmetros: 1º Edifício; 2º Lista de Pisos

**elevador** -> Elevador de um edifício

Parâmetros: 1º Edifício; 2º Lista de Pisos que serve

**corredor** -> Corredor/Passagem entre dois edifícios

Parâmetros: 1º Edifício; 2º Edifício; 3º Piso; 4º Piso

**sala** -> Sala de um determinado Piso

Parâmetros: 1º Nome da Sala; 2º Piso

**bc\_coordenadas.pl**

**Uma imagem com texto, captura de ecrã, Tipo de letra, design

Descrição gerada automaticamenteUma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente**

**coordenadas corredores** -> Coordenadas na matriz de corredores

Parâmetros: 1º Piso; 2º Piso; 3º Coluna do 1 Piso; 4º Linha do 1 Piso; 5º Coluna do 2 Piso; 6º Linha do 2 Piso

**Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamentecoordenadas salas** -> Coordenadas na matriz de salas

Parâmetros: 1º Nome Sala; 2º Piso; 3º Coluna Sala; 4º Linha Sala

**dimensões** -> Dimensões do Piso

Parâmetros: 1º Piso; 2º Num Colunas; 3º Num Linhas

**Uma imagem com captura de ecrã, Tipo de letra, Gráficos, design gráfico

Descrição gerada automaticamenteUma imagem com Tipo de letra, texto, captura de ecrã, Gráficos

Descrição gerada automaticamenteUma imagem com captura de ecrã, Tipo de letra, texto, Gráficos

Descrição gerada automaticamentebc\_floorMaps.pl:**

**m** -> Célula da matriz de cada Piso

Parâmetros: 1º Piso; 2º Linha; 3º Coluna; 4º Valor da célula (= 0: não tem obstáculo e => 1: tem obstáculo)

**Uma imagem com Retângulo, diagrama, quadrado, Esquema

Descrição gerada automaticamente**Exemplo de matriz:

**Uma imagem com captura de ecrã, texto

Descrição gerada automaticamente**

1. **Solução Movimentação entre Pisos**

**Uma imagem com texto, Tipo de letra, captura de ecrã, file

Descrição gerada automaticamente**Para a solução da movimentação entre pisos, relativamente aos predicados disponibilizados pelos docentes não houve muitas ou nenhumas alterações. Começa-se por chamar o predicado **melhor\_caminho\_pisos/3,** passando como argumento o piso de origem e o de destino que depois irá executar um **findall** no predicado **caminho\_pisos/4** retornando assim uma lista de soluções que irá ser passada como argumento ao predicado **menos\_elevadores/4,** que vai encontrar a solução com menos utilização de elevadores.

As alterações feitas por nós para esta solução foi na criação de um predicado que completasse a lista retornada pelo predicado **melhor\_caminho\_pisos/3,** porque esta lista não contém o elemento de origem nem o de destino. O predicado **caminho\_pisos/3**, primeiro determina o tipo de elemento que o utilizador colocou bem como o respetivo piso desse elemento, isso é realizado com o predicado **determinar\_tipo\_entidade/2** que recebe como argumentos o elemento que pode ser **sala(‘nome da sala’)**, **elev(‘piso’)** e **pass(‘corredor de origem’, ‘corredor de destino’).**

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamenteO predicado depois de conseguir os dois pisos de ambos elementos, executa então o predicado **melhor\_caminho\_pisos/3** passando como argumento esses dois pisos, os dois elementos vão então ser introduzidos na lista retornada pelo predicado anterior através do **append/2.**

1. Solução Movimentação do Robot

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamentePara a solução da movimentação do robot começamos por alterar o predicado do **cria\_grafo/3,** adicionamos um novo argumento que é o Piso, este argumento vai ser necessário porque as células da matriz que contêm os valores, os tais m’s da base de conhecimento e a criação dos **ligacel/4** que vão ser as ligações entre as células com valor 0 da respetiva matriz do piso, vão receber o Piso como argumento para sabermos a qual piso estão associados. Além desta adição também adicionamos formas de criar as ligações nas diagonais no predicado **cria\_grafo\_lin/3**, em que primeiro verificamos se a célula da diagonal e as outras ao seu redor contém o valor 0, se isto se verificar a ligação na diagonal é realizada com um custo de sqrt(2) ao contrário das outras de custo 1.

Uma imagem com texto, Tipo de letra, captura de ecrã

Descrição gerada automaticamentePara fornecer o piso, o número de colunas e linhas para criar o grafo foi criado o predicado **gera\_grafo/1,** este é apenas responsável por receber o piso que vem da parte da integração do ponto 2 e 3 e será explicado mais à frente, vai buscar as dimensões do respetivo piso incluindo número de colunas e linhas através do **dimensões/3** da nossa base de conhecimento.

Uma imagem com Tipo de letra, texto, captura de ecrã, Gráficos

Descrição gerada automaticamenteEm relação aos algoritmos do DFS e Astar muito poucas alterações foram realizadas, para ambos foi adicionado um argumento adicional que é o piso devido a usar o **ligacel/4** que recebe como um dos argumentos o piso.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Devido ao elevado número de diferentes combinações que se podia fazer com um par de pontos de acesso introduzidos pelo utilizador, tivemos de criar predicados que atendessem cada tipo respetivo, isto é, devido a todos os pontos de acesso serem diferentes e levar diferentes argumentos.

Para este efeito foi criado o predicado **processar\_par/3** querecebemos como argumentos um par de elementos, consoante o tipo de par, diferentes predicados do processar\_par/3 vão ser executados, por exemplo podemos ter como par: elev <-> pass, pass <-> pass, sala <-> sala, sala <-> elev, sala <-> pass.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamenteCada predicado vai executar o **gera\_grafo/1** que vai gerar dinamicamente todas as ligações entre as células da matriz do piso **(ligacel/4)** sendo que o piso é fornecido através de um dos elementos, a seguir executa o predicado **encontra\_caminho/3** que vai ser responsável por encontrar o melhor caminho no piso entre os dois elementos respetivos, recebendo como argumentos os mesmos dois elementos.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamenteNo predicado **encontra\_caminho/3,** como o predicado **processar\_par/3** aqui também existe diversos predicados consoante os tipos de pares de elementos, isto é necessário para ir buscar as coordenadas corretas a nossa base de conhecimento. Depois de obtido as coordenadas de ambos, executamos o algoritmo **DFS** ou **Astar** passando como argumentos as coordenadas na forma de **cel (Coluna, Linha),** relembrando que a **ligacel/4** utilizado pelos dois algoritmostem como dois dos 4 argumentos dois **cel**. Depois de obtido o caminho com sucesso e guardado na variável **Cam**, damos print de toda a informação do trajeto, isto incluí o piso, elemento de partida e de chegada, coordenadas de ambos e o respetivo caminho.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

1. Integração do ponto 2 com o ponto 3

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamentePara a integração das duas soluções, criamos um predicado **find\_caminho/3,** que irá receber o input do utilizador neste caso os dois elementos de origem e de destino sendo que o formato já foi explicado no ponto 1 e duas listas, uma lista para o resultado do melhor caminho entre pisos, e outra lista para o resultado do caminho do robot, neste caso uma lista de listas. Primeiro vai executar o predicado **caminho\_pisos/3** onde vai buscar uma lista do melhor trajeto entre pisos incluindo ambos os elementos de input introduzidos no início e no fim. De seguida, executa o predicado **processar\_caminho/2** que recebe como argumento a lista retornada pelo predicado anterior.

No predicado **processar\_caminho/2,** processamos a lista do melhor caminho aos pares de elementos, para cada par é executado o predicado **processar\_par/3** explicado no ponto 3sendo que depois o primeiro elemento dessa lista (1 elemento do par) é retirado e o segundo elemento passa para a cabeça da lista e assim sucessivamente até restar apenas um elemento como podemos na regra geral definido na primeira linha. Além disto, relembrando que a variável Cam era onde estava a ser guardado a solução do caminho do robot, consequentemente com o backtracking guardamos essa lista desse trajeto na cabeça de uma lista, ficando assim ordenado da origem para o destino e retornando todos os caminhos do robot ao predicado **find\_caminho/4**.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Exemplo de uma solução a passar por 3 edifícios, 2 pisos e a utilizar elevadores e passagens:

Ponto de partida: sala k1 (edifício a, piso a1); Chegada: sala c204(edifício c, piso c2)

Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamenteAlgoritmo usado para cada par: DFS