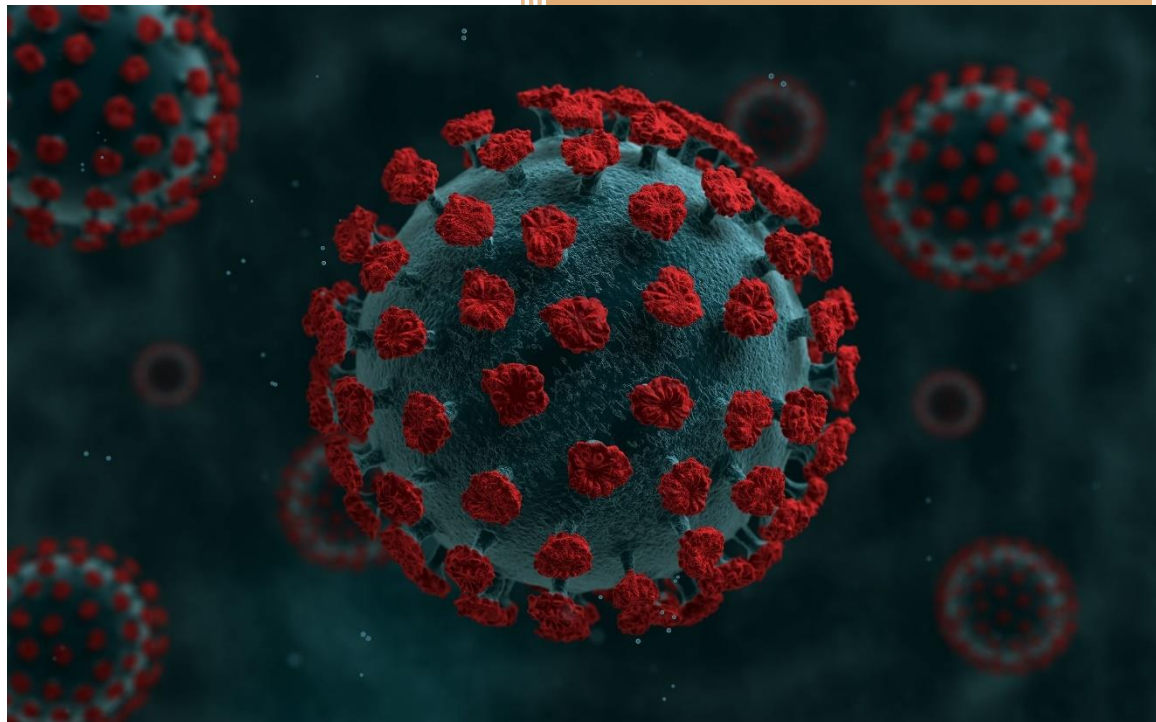


Relatório do Projeto de LAPR1



23 de janeiro de 2022

Turma DK, Equipa 7, Eco-Boys

1211506, Henrique Caldas

1211511, Pedro Campos

1211920, Gonçalo Silva

Carlos Ferreira (CGF)

Luis Lino Ferreira (LLF)

Maria Teresa Costa (MCO)

Claudia Lima (CAL)

Ana Barata (ABT)

ÍNDICE

Conteúdo

1.	INTRODUÇÃO	1
2.	METODOLOGIA DE TRABALHO	2
2.1 SCRUM NO DESENVOLVIMENTO DO PROJETO		2
2.2 PLANEAMENTO E DISTRIBUIÇÃO DE TAREFAS		2
2.3 REFLEXÃO CRÍTICA SOBRE A DINÂMICA DA EQUIPA		2
3.	ANÁLISE DE CADEIAS DE MARKOV E DECOMPOSIÇÃO LU PELO MÉTODO DE CROUT	3
3.1	CADEIAS DE MARKOV	3
3.2	DECOMPOSIÇÃO LU PELO MÉTODO DE CROUT	4
4.	DESENVOLVIMENTO E IMPLEMENTAÇÃO DA APLICAÇÃO	6
5.	CASO DE ESTUDO	10
5.1. ANÁLISE E DISCUSSÃO DOS RESULTADOS		10
6.	CONCLUSÃO	14
.....	ERROR! BOOKMARK NOT DEFINED.	

1. Introdução

Este projeto foi nos atribuído no âmbito da cadeira Laboratórios de Projeto 1 e consiste no desenvolvimento de uma aplicação informática em linguagem Java. Esta aplicação destina-se à instituição Monitorização da Saúde em Portugal (MSP) com o objetivo de estudar e prever a evolução da pandemia Covid-19.

A instituição MSP tem registados dados diários sobre a evolução da pandemia em ficheiros CSV, registando dados acumulados e também dados totais diários para os diferentes estados Covid-19 em que cada habitante se pode encontrar. Mais precisamente regista o número total de não infetados, o número total de infetados, o número total de hospitalizados, o número total de internados em Unidades de Cuidados Intensivos (UCI) e o número total de óbitos que estão registados num determinado dia. Regista também o número acumulado de infetados, o número acumulado de hospitalizados, o número acumulado de internados em UCI e o número acumulado de mortos até um determinado dia em resultado da infeção Covid19.

Com isto, a MSP pretende compreender fenómenos temporais/sazonais, fazer uma análise comparativa entre períodos temporais e prever a evolução da pandemia.

2. Metodologia de Trabalho

De forma a executarmos melhor o nosso trabalho, pusemos em prática uma metodologia de trabalho que potenciase o nosso desempenho a realizar cada tarefa.

2.1 Scrum no desenvolvimento do Projeto

Para uma melhor organização na sua realização atribuímos um scrum master para cada sprint, de forma a termos alguém responsável para guiar e juntar coerentemente toda a informação de cada um dos membros do grupo.

Este projeto consistia em 2 sprints, o primeiro sprint ligado a primeira semana, e as alíneas 2.1 e 2.2 do enunciado do projeto, o scrum master deste sprint foi o Pedro Campos. No segundo sprint, que consistia na alínea 2.3 do enunciado teve como scrum master José Batalha, porém dada a desistência deste do projeto, o scrum master passou a ser Henrique Caldas.

2.2 Planeamento e distribuição de tarefas

Nesto projeto dividimos as variadas tarefas necessárias ao seu desenvolvimento não uniformemente por todos os membros do grupo, dado que dois dos alunos ficaram encarregues da programação (Henrique Caldas, Pedro Campos), enquanto que o outro membro (Gonçalo Silva), embora tenha dito que ficava encarregue do relatório porque não conseguia ajudar os restantes membros do grupo na parte de programação do projeto, mas mesmo assim só concluiu a capa, o índice, o ponto 3 do relatório e mesmo o ponto 2 o qual foi parcialmente feito por este não estava acabado. Utilizamos diversas ferramentas de comunicação e trabalho de equipa. As nossas ferramentas mais utilizadas foram o Trello, com o objetivo de estabelecer e distribuir tarefas, o Whatsapp usado no agendamento de reuniões e o Microsoft Teams em conjunto com o Discord na execução das mesmas. Por último, utilizamos o Bitbucket para trabalhar conjuntamente na realização da aplicação.

2.3 Reflexão crítica sobre a dinâmica da equipa

Refletindo na nossa experiência, percebemos que a dinâmica de trabalho entre os membros do grupo foi pouco fluída e que o membro encarregue do relatório pouco comunicava com os restantes membros, apesar de uma boa comunicação entre os programadores do grupo.

3. Análise de cadeias de Markov e decomposição LU pelo método de Crout

$$A = \begin{pmatrix} 0.9995 & 0.03 & 0.002 & 0.001 & 0 \\ 0.0005 & 0.96 & 0.004 & 0.015 & 0 \\ 0 & 0.007 & 0.98 & 0.02 & 0 \\ 0 & 0.003 & 0.01 & 0.95 & 0 \\ 0 & 0 & 0.004 & 0.014 & 1 \end{pmatrix}$$

Figura 1 - matriz [A]

3.1 Cadeias de Markov

Uma grande quantidade de processos estudados na atualidade, são resultados que são medidos ao longo do tempo. Dentro destes um grande número tem resultados aleatórios, ou seja, são resultados imprevisíveis. Estes processos são chamados processos estocásticos e são estudados usando a teoria das probabilidades.

Um dos modelos que melhor explica estes processos é chamado de Cadeias de Markov, que são processos aleatórios cujo resultado no estágio m depende somente do que aconteceu no estágio $n - 1$ e não dos resultados anteriores a este.

$$P(X_{n+1}=x \mid X_0, X_1, X_2, X_3, \dots, X_n) = P(X_{n+1}=x \mid X_n)$$

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1k} \\ p_{21} & p_{22} & \cdots & p_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ p_{k1} & p_{k2} & \cdots & p_{kk} \end{bmatrix}$$

Figura 2 - Matriz de Transição

3.2 Decomposição LU pelo método de Crout

Esta decomposição é um algoritmo eficiente para decompor a matriz [A] nas matrizes [L] e [U]. Seja [U] uma matriz 5x5, triangular superior com 1's na diagonal principal:

$$U = \begin{pmatrix} 1 & 0.03 & 0.002 & 0.001 & 0 \\ 0 & 1 & 0.004 & 0.015 & 0 \\ 0 & 0 & 1 & 0.02 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Figura 3 - matriz [U]

E [L] uma matriz triangular inferior:

$$L = \begin{pmatrix} 0.9995 & 0 & 0 & 0 & 0 \\ 0.0005 & 0.96 & 0 & 0 & 0 \\ 0 & 0.007 & 0.98 & 0 & 0 \\ 0 & 0.003 & 0.01 & 0.95 & 0 \\ 0 & 0 & 0.004 & 0.014 & 1 \end{pmatrix}$$

Figura 4 - matriz [L]

Podemos então escrever $LU = A$,

$$\begin{pmatrix} 0.9995 & 0 & 0 & 0 & 0 \\ 0.0005 & 0.96 & 0 & 0 & 0 \\ 0 & 0.007 & 0.98 & 0 & 0 \\ 0 & 0.003 & 0.01 & 0.95 & 0 \\ 0 & 0 & 0.004 & 0.014 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0.03 & 0.002 & 0.001 & 0 \\ 0 & 1 & 0.004 & 0.015 & 0 \\ 0 & 0 & 1 & 0.02 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0.9995 & 0.03 & 0.002 & 0.001 & 0 \\ 0.0005 & 0.96 & 0.004 & 0.015 & 0 \\ 0 & 0.007 & 0.98 & 0.02 & 0 \\ 0 & 0.003 & 0.01 & 0.95 & 0 \\ 0 & 0 & 0.004 & 0.014 & 1 \end{pmatrix}$$

O método de Crout derivado através da multiplicação de matrizes L e U e depois equacionando os resultados a matriz A. Recorrendo às regras da multiplicação de matrizes 5x5 temos os seguintes passos:

Multiplicar as linhas da matriz [L] pela 1ª coluna da matriz [U]. Generalizando o resultado temos:

$$l_{i1} = a_{i1} , \text{ para } i = 1, 2, \dots, n$$

De seguida, a 1ª linha da matriz [L] pode ser multiplicada pelas colunas da matriz da matriz [U].

As relações podem ser generalizadas por:

$$u_{1j} = \frac{a_{1j}}{l_{11}} , \text{ para } j = 2, 3, \dots, n$$

Da 2ª à 5ª linha da matriz [L], vamos multiplicá-las pela 2ª coluna de [U]:

$$l_{21}u_{12} + l_{22} = a_{22} , l_{31}u_{12} + l_{32} = a_{32} , l_{41}u_{12} + l_{42} = a_{42}$$

A seguir, os coeficientes da 2ª linha de [U] podem ser calculados multiplicando da 2ª linha da matriz [L] pela 3ª, 4ª e 5ª coluna da matriz [U].

$$l_{21}u_{13} + l_{22}u_{23} = a_{23} , l_{21}u_{14} + l_{22}u_{24} = a_{24}$$

4. Desenvolvimento e Implementação da Aplicação

O projeto consiste em 3 partes principais distintas, estas são, a análise comparativa, a comparação de dados e a previsão da evolução da pandemia.

4.1 Comparação de dados

A comparação de dados consiste em mostrar o que aconteceu durante um período de tempo de duas formas, mostrando o número de novos casos, hospitalizados, internados e óbitos nesse período de tempo e mostrar a diferença, ora diária, ora semanal, ora mensal, dependendo das intenções do utilizador. Depois de selecionar duas datas, estas que devem estar presentes no ficheiro de entrada. Depois de o programa verificar que existe uma data, ou no modo interativo verificar se a data esta dentro do ficheiro através do modulo "ValidarDatasInseridas", que transforma cada data do array de datas, que esta em Date[] para String, e compara cada data até encontrar uma data que seja igual a data introduzida.

Depois desta verificação será necessário selecionar 1 de 3 modos, o modo diário, semanal, ou mensal, isto é feito por introduzir zero para o modo diário, 1 para o semanal e 2 para o mensal. Passada esta seleção existe mais uma etapa para verificação dos dados, caso seja selecionado o modo semanal ou mensal, esta é a verificar a quantidade de dias presentes no espaço de tempo introduzido.

4.1.1 Modo Semanal

Isto no modo semanal é feito através do modulo "ProcurarPrimeiraSegundaFeira", que através de uma mensagem "mensagemSegundaFeiraReferencia"(String) que diz "segunda-feira" no idioma selecionado no computador, transforma as datas no array "listaDeDatas" ([Date[]) em String de modo a que cada data se transforme no respetivo dia da semana, até ser encontrada uma data que depois destas transformações seja igual à "mensagemSegundaFeiraReferencia", ou seja, até ser encontrada a primeira segunda-feira do período temporal. Se não for encontrada uma segunda-feira o programa escreve que o período de tempo selecionado para análise semanal é muito curto, ou se for encontrada uma segunda-feira e não houver dias suficientes para completar uma semana depois desta primeira segunda-feira o utilizador é avisado.

4.1.2 Modo mensal

No modo mensal há mais verificações a fazer, a primeira é encontrar o primeiro dia do mês, isto é feito através do modulo "ProcurarPrimeiroDiaMesParaAnalisar" que transforma as datas no array lista de datas em Strings e divide as datas no formato "aaaa-MM-dd" e divide-as em Strings, estas Strings são depois separadas usando o método .split("-") (divide a String quando encontra o sinal -) e colocadas num array de String, até ser encontrado o primeiro dia "01" e retorna o index do array onde este se encontra.

De seguida é verificado último dia do mês, para a verificação do último dia do mês é utilizado o modulo "ProcurarUltimoDiaMesParaAnalisar" que através de métodos da

biblioteca `time.localtime` é encontrada a data do último dia do mês, que depois é comparada com a data do `"indexFinal"`, se esta data não for igual à data no `"indexFinal"` vai sendo subtraído 1 ao `"indexFinal"` até a data neste índice ser igual a data do último dia do mês. No fim deste módulo é retornado o `"indexFinal"` que irá ter o valor do índice do último dia do mês.

Existe também um módulo para verificar a quantidade de meses, este é o módulo `"CalcularQuantidadeDeMeses"` que utiliza a mesma técnica usada em `"ProcurarPrimeiroDiaMesParaAnalizar"` para transformar as datas em Strings e depois dividir estas Strings através do método `.split("-")`, para analisar quantas vezes muda o mês, ao compará-lo com o valor do mês que está a ser analisado de momento, e sempre que há uma mudança de mês é adicionado +1 a uma variável `"qtMeses"` e o mês a analisar atualiza.

Para a segunda parte do modo mensal, a comparação entre os meses no período de tempo selecionado, são utilizados os módulos `"ProcurarPrimeiroDiaDoMes"` e `"ProcurarUltimoDiaDoMes"` que têm funções parecidas com a dos módulos referidos acima, porém ao introduzir um período de tempo como por exemplo 15-01-2020 até 15-02-2020, o módulo irá andar para trás na lista de arrays de datas até encontrar o primeiro dia do mês ou para encontrar o último dia, neste caso de fevereiro, o `"indexFinal"` irá sendo incrementado até chegar ao último dia do mês, para no final aparecer a mensagem com a diferença entre os meses.

4.1.3 Mostrar os Resultados

Passadas as verificações dá-se entrada na parte de mostrar os resultados, que se passa nos módulos `"MostrarValoresDoDia"` e `"MostrarNovosCasosDiarios"`, para o modo diário, no `"ModoSemanalMostrarDados"` (este módulo inclui também uma verificação, a qual garante que existe pelo menos uma semana no período de tempo inserido para dar início ao modo semanal) para o modo semanal, isto depois de os dados passarem pelo módulo `"SomarDadosDaSemana"`, onde os dados a imprimir são calculados e o modo mensal `"ModoMensalMostrarDados"` depois de os dados passarem pelo módulo `"SomarDadosMes"`, que trata os dados antes de os imprimir.

4.2 Análise Comparativa

Nesta parte do projeto são inseridos 2 períodos de tempo, ou seja, 4 datas, o índice destas 4 datas é enviado para o módulo `"CompararPeriodosDeTempo"`, que vai escrever no ecrã os dados a comparar e a sua diferença.

Para esta parte são necessários mais 2 módulos, estes são os módulos `"MostrarMedias"` e `"CalcularDesvioPadrao"`. O primeiro calcula a média dos dados dos 2 períodos de tempo e da diferença destes 2. No final do módulo é retornado um array com as médias e a quantidade de dias inseridos no período de tempo mais baixo, que vai servir para calcular o desvio padrão no módulo `"CalcularDesvioPadrao"` e `"CalcularVariacao."`. No final os dados são escritos no main.

4.3 Previsão da evolução da pandemia

Para esta alínea começa-se por pedir uma data para a previsão, logo depois, e necessário inserir um ficheiro com a matriz de probabilidades, que é lido através do modulo "LerFicheiroMatriz". Depois é verificado se a data inserida esta no ficheiro, e se não estiver qual é a data mais próxima com "ProcurarDataMaisProxima", que encontra o índice anterior a data pedida se esta estiver no ficheiro, ou a data do índice final se a data não estiver no ficheiro. Através do índice da data mais próxima, que é retornado pelo modulo anterior. São atribuídos valores às variáveis de dados, de não infetados, infetados, hospitalizados, internados na UCI e óbitos, estes valores são colocados numa matriz, "matrizValores". Depois a matriz de probabilidades é elevada pelo número de dias de diferença entre a data selecionada pelo utilizador e a data mais próxima dessa data dentro do ficheiro, isto acontece no modulo "ElevarMatriz". Passado isto faz-se a multiplicação de matrizes, no modulo "MultiplicarMatrizes", e atribuem-se os valores nessa matriz as respetivas variáveis, que vão ser impressas depois.

4.4 Modo não interativo

Para o modo não interativo o programa verifica se existem argumentos que possa usar, se estes existirem, então o modo interativo será iniciado.

No início deste modo, o programa atribui os valores dos argumentos as suas respectivas variáveis, através de um "switch", desta maneira também é possível iniciar o programa sem ter todos os argumentos, isto é, iniciar o programa só com argumentos suficientes para fazer a comparação de dados, ou só a análise comparativa, isto acontece porque as variáveis cujos valores dos argumentos são atribuídos são inicializados com valores anormais, ou seja, valores que não podem ser argumentos, assim durante a execução do programa é possível verificar o valor dos argumentos e usar apenas as funcionalidades, para cujos os argumentos não têm valores anormais, o que significa, apenas os valores que foram introduzidos como argumentos pelo utilizador.

4.5 Modo interativo

No modo não interativo é verificado a quantidade de argumentos existentes no programa, se esta quantidade for zero, o modo interativo começa.

Este modo usa um "switch" dentro de um "while" e os inputs do utilizador para fazer as ações desejadas, como por exemplo fazer o que é mencionado no 4.1, 4.2, 4.3, mudar o ficheiro de entrada se assim for desejado ou gravar os resultados imprimidos.

Se alguma vez o utilizador quiser sair de alguma ação que esta a fazer ou, até mesmo do programa basta introduzir o comando "sair" e o programa dependendo daquilo que o utilizador está a fazer de momento, interrompe essa ação e leva-o para o menu principal, ou acaba o programa, se o utilizador já estiver no menu principal. Isto é possível devido a alguns "ifs" e "switchs" implementados no projeto que garantem que o projeto não termina por ter erros, ou que o utilizador não fica preso numa verificação.

Existem também alguns ciclos "while" que verificam a existência e pedem novamente caso não seja valido, ora as datas , através do modulo "ValidarDatasInseridas", ora os ficheiros, através do metodo ".exists()".

5. Caso de Estudo

Neste ponto dá-se a previsão dos casos da pandemia na primeira semana de janeiro de 2022 utilizando duas matrizes de probabilidades diferentes, A e B, e tem como objetivo ver qual a mais correta, através das técnicas, referidas no enunciado do trabalho, cadeias de Markov.

5.1. Análise e discussão dos resultados

Ao usar a matriz A para prever os primeiros 7 dias de Janeiro, obtemos os seguintes resultados:

Dia 01-01-2022:

Não Infetados previstos = 9969466 , erro = 17907

Infetados previstos = 176554 , erro = 19669

Hospitalizados previstos = 2257 , erro = 1234

Internados UCI previstos = 689, erro = 547

Óbitos previstos = 24, erro = 3

Dia 02-01-2022

Não Infetados previstos = 9969783, erro = 25401

Infetados previstos = 174496 , erro = 28826

Hospitalizados previstos = 3461, erro = 2380

Internados UCI previstos = 1202, erro = 121

Óbitos previstos = 42, erro = 28

Dia 03-01-2022

Não Infetados previstos = 9970041, erro = 30201

Infetados previstos = 172533, erro = 35326

Hospitalizados previstos = 4638, erro = 3471

Internados UCI previstos = 1700 , erro = 1553

Óbitos previstos = 73, erro = 63

Dia 04-01-2022

Não Infetados previstos = 9970243, erro = 36434

Infetados previstos = 170660, erro = 43089

Hospitalizados previstos = 5787, erro = 4584

Internados UCI previstos = 2179, erro = 2032

Óbitos previstos = 115, erro = 100

Dia 05-01-2022

Não Infetados previstos = 9970391, erro = 61989

Infetados previstos = 168875, erro = 70223

Hopitalizados previstos = 6909, erro = 5658

Internados UCI previstos = 2640, erro = 2497

Óbitos previstos = 169, erro = 152

Dia 06-01-2022

Não Infetados previstos = 9970489, erro = 70529

Infetados previstos = 167172, erro = 80268

Hopitalizados previstos = 8006, erro = 6695

Internados UCI previstos = 3083, erro = 2925

Óbitos previstos = 234, erro = 209

Dia 07-01-2022

Não Infetados previstos = 9970538, erro = 77440

Infetados previstos = 165549, erro = 88691

Hopitalizados previstos = 9078, erro = 7725

Internados UCI previstos = 3511, erro = 3350

Óbitos previstos = 309, erro = 292

Ao usar a matriz B para prever os primeiros 7 dias de Janeiro, obtemos os seguintes resultados:

Dia 01-01-2022:

Não Infetados previstos = 9972864, erro = 21305

Infetados previstos = 174768, erro = 21445

Hopitalizados previstos = 1167, erro = 144

Internados UCI previstos = 163, erro = 31

Óbitos previstos = 21, erro = 4

Dia 02-01-2022

Não Infetados previstos = 9976447, erro = 32065

Infetados previstos = 171025, erro = 32297

Hopitalizados previstos = 1304, erro = 223

Internados UCI previstos = 182, erro = 34

Óbitos previstos = 25, erro = 11

Dia 03-01-2022

Não Infetados previstos = 9979846, erro = 40096

Infetados previstos = 167472, erro = 40387

Hopitalizados previstos = 1435, erro = 268

Internados UCI previstos = 201, erro = 54

Óbitos previstos = 30, erro = 20

Dia 04-01-2022

Não Infetados previstos = 9983070, erro = 49261

Infetados previstos = 164099, erro = 49650

Hopitalizados previstos = 1561, erro = 358

Internados UCI previstos = 219, erro = 72

Óbitos previstos = 35, erro = 20

Dia 05-01-2022

Não Infetados previstos = 9986127, erro = 77725

Infetados previstos = 160897, erro = 78201

Hopitalizados previstos = 1682, erro = 431

Internados UCI previstos = 237, erro = 94

Óbitos previstos = 40, erro = 26

Dia 06-01-2022

Não Infetados previstos = 9989027, erro = 89067

Infetados previstos = 157858, erro = 89582

Hopitalizados previstos = 1798, erro = 487

Internados UCI previstos = 255, erro = 97

Óbitos previstos = 46, erro = 29

Dia 07-01-2022

Não Infetados previstos = 9991776, erro = 98678

Infetados previstos = 154973, erro = 99267

Hopitalizados previstos = 1909, erro = 556

Internados UCI previstos = 272, erro = 111

Óbitos previstos = 52, erro = 35

Obtemos então os seguintes erros para a matriz A:

Não infectados = 319901

Infectados = 366092

Hospitalizados = 31675

Internados UCI = 13025

Óbitos = 847

Obtemos então os seguintes erros para a matriz B:

Não infectados = 408197

Infectados = 410829

Hospitalizados = 2467

Internados UCI = 493

Óbitos = 145

Concluimos então que a matriz A de probabilidades é mais precisa para prever o número de não infectados e o número de infectados.

A matriz B é mais precisa na previsão do resto dos valores, hospitalizados, internados UCI e óbitos.

6. Conclusão

Em suma, foi possível concluir o projeto na sua maioria utilizando os conhecimentos de java que foram adquiridos nas aulas de “Aprog”. Porém o grupo não conseguiu concluir a alínea 2.3 do enunciado, ou colocar testes unitários em todos os módulos, uma vez que, e embora tenha sido feita uma gestão cuidadosa do tempo e das tarefas de cada um, só dois dos membros do grupo programaram no projeto (Henrique Caldas, Pedro Campos), o que tornou completar este projeto, um grande desafio para o tempo dado.

