# Trabalhos de Hércules - COBOL



Com a chegada de novos profissionais na área de mainframe do banco com pouco ou nenhum conhecimento em COBOL, nasceu a necessidade de criar um projeto prático de codificação que sirva para introduzir alguns conceitos básicos de programação e construção de software. Em homenagem à sexta turma do concurso de 2023 (turma Olimpo) foram feitas 12 tarefas de Hércules com diferentes dificuldades. Esperamos que você aproveite os desafios e aprenda um pouco como programar a incrível linguagem dos mainframes, se tiver dificuldades com os desafios não tenha medo em pedir ajuda para seus colegas de trabalho ou aos responsáveis pelo projeto. É importante ressaltar que esse projeto ainda está em fase de testes e seu feedback é importante para o aprimoramento de mais iniciativas como essa, não se esqueça de responder a pesquisa quando terminar os 12 desafios.

Antes de começar, queremos deixar claro que você pode usar qualquer IDE para realizar os desafios, como por exemplo OpenCobol, VSCode ou se puder o próprio KDZ.

Sumário

[Trabalhos de Hércules - COBOL 1](#_Toc188470878)

[#1 Tarefa: aves do lago Estínfalo 3](#_Toc188470879)

[Conversor de moedas 3](#_Toc188470880)

[#2 Tarefa: Leão de Neméia 5](#_Toc188470881)

[Calculadora de formas geométricas 5](#_Toc188470882)

[#3 Tarefa: Gado de Gerião 7](#_Toc188470883)

[Indicação e limite de crédito do cartão 7](#_Toc188470884)

[#4 Tarefa: Corça de Cerínia 10](#_Toc188470885)

[Aptidão Profissional 10](#_Toc188470886)

[#5 Tarefa: Touro de Creta 12](#_Toc188470887)

[Calculadora de aposentadoria 12](#_Toc188470888)

[#6 Tarefa: Pomares das Hespérides 14](#_Toc188470889)

[Simulador de aplicação financeira 14](#_Toc188470890)

[#7 Tarefa: Cinto de Hipólita 16](#_Toc188470891)

[Ranking e promoção de funcionários 16](#_Toc188470892)

[#8 Tarefa: Éguas de Diomedes 19](#_Toc188470893)

[Calculadora de prêmio de seguro veicular 19](#_Toc188470894)

[#9 Tarefa: Estábulos de Áugias 22](#_Toc188470895)

[Calculadora de índice de massa corporal (IMC) 22](#_Toc188470896)

[#10 Tarefa: Javali de Erimanto 24](#_Toc188470897)

[Análise de perfil do investidor 24](#_Toc188470898)

[#11 Tarefa: Hidra de Lerna 26](#_Toc188470899)

[CRUD de peças 26](#_Toc188470900)

[#12 Tarefa: Guardião dos mortos 27](#_Toc188470901)

# #1 Tarefa: aves do lago Estínfalo

## Conversor de moedas

Nesse desafio vamos criar um conversor de moedas estrangeiras, a ideia dessa tarefa é aprender a usar books para chamar variáveis, como separar módulos e chamá-los em rotinas e alguns conceitos básicos de construção de software Cobol e suas boas práticas.

Iniciamos escolhendo as moedas que serão utilizadas no nosso sistema:

* Real (R$)
* Dólar (US$)
* Euro (€)
* Yuan (¥)

Nosso sistema também vai precisar ter uma data para indicar as cotações mais atuais utilizando as datas do próprio sistema, o ideal é deixar a data no topo.Texto

Descrição gerada automaticamente 

Os principais objetivos do sistema de conversão de moedas são os seguintes:

* Poder selecionar uma das moedas disponíveis para conversão;
* Após selecionar a moeda para conversão o usuário irá entrar com a quantia que será convertida (essa quantia precisa ser validada)

Exemplo:

>Selecione a moeda para conversão  
>1- Real (Brasil)

>2- Dólar (EUA)

>3- Euro (Europa)

>4- Yuan (China)  
\* O usuário precisa digitar um número ente 1 e 4, caso ele digite o número errado uma mensagem de erro será disparada e o pergunta será refeita.

Após selecionar a moeda desejada o usuário irá digitar a quantia desejada para conversão:

>Digite a quantidade da moeda desejada para conversão  
> \* Quantia digitada \*  
\* A quantia não pode ser menor que 1, será necessário criar uma validação para evitar valores negativos.

Após o usuário digitar um valor o sistema irá disponibilizar os valores convertidos em uma lista:

>Moeda selecionada: Real

>Valor para conversão: R$1000,00

>conversão de R$1000,00 em dólar

>conversão de R$1000,00 em euro

>conversão de R$1000,00 em yuan

\* O valor das conversões vai depender do valor das variáveis no book

Se tiver dúvidas para realizar as tarefas você pode conferir o código no Gitlab

ou pedir ajuda para algum colega.

Realize os seguintes testes para validar seu código:

Valores das moedas: dólar = R$6,1; euro = R$6,4; yuan = R$0,84

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Moeda | Valor de conversão | Valor #1 | Valor #2 | Valor #3 |
| Real | R$ 1000,00 | US$163,93 | €156,25 | ¥ 1190,47 |
| Real | R$ 500 | US $81,96 | € 78,12 | ¥ 595,23 |
| Dólar | US$ 1000,00 | R$ 6100,00 | € 953,12 | ¥ 7261,90 |
| Dólar | US$ 500 | R$ 3050,00 | € 476,56 | ¥ 3630,95 |
| Euro | € 1000,00 | R$ 6400,00 | US$ 1049,10 | ¥ 7619,04 |
| Euro | € 500 | R$ 3200,00 | US$ 524,59 | ¥ 3809,52 |
| Yuan | ¥ 1000,00 | R$ 840,00 | US$ 137,70 | € 131,25 |
| Yuan | ¥ 500 | R$ 420,00 | US$ 68,85 | € 65,62 |

O escopo de testes está incompleto, fique à vontade para ampliar o escopo de teste para mais valores.

Após terminar a primeira parte do projeto, tente criar funcionalidades adicionais como:

* Mudar os valores do BOOK-MOEDAS e verificar as saídas e testar novamente
* Adicione uma nova moeda para conversão, o won sul-coreano seria uma ótima opção

# #2 Tarefa: Leão de Neméia

## Calculadora de formas geométricas

Criar calculadoras são uma ótima forma de testar as funcionalidades dos operadores aritméticos de qualquer linguagem, nesse desafio vamos aprender a usar a cláusula COMPUTE para realizar cálculos de formas geométricas e aprimorar ainda mais o desenvolvimento de código COBOL.

Primeiro vamos escolher nossas formas geométricas:

* Esfera
* Cubo
* Paralelepípedo (esse não está no código de exemplo)

Os principais objetivos do sistema de calculadora de formas geométricas são os seguintes:

* Selecionar uma forma geométrica para realizar os cálculos;
* Após selecionar a forma geométrica o usuário irá digitar a medida que a forma precisa para ser calculada (raio, altura, lado e etc.), esse valor precisa ser validado

Exemplo:

>Selecione a forma geométrica  
>1- Área e volume da esfera

>2- Área e volume do cubo  
\* O usuário precisa digitar um número ente 1 e 2, caso ele digite o número errado uma mensagem de erro será disparada e o pergunta será refeita.

Após selecionar a forma desejada o usuário irá digitar o valor da medida para realizar os cálculos de área e volume:

>Digite o raio da esfera em cm  
> \* medida do raio \*  
\* A quantia não pode ser menor que 1, será necessário criar uma validação para evitar valores negativos.

Após o usuário digitar um valor o sistema irá disponibilizar as medidas calculadas:

>Forma geométrica selecionada: Esfera

>Valor do raio da esfera: 5 cm

>Área da esfera: 125,66 cm2

>Volume da esfera: 523,58 cm3

Se tiver dúvidas para realizar as tarefas você pode conferir o código no Gitlab

ou pedir ajuda para algum colega.

Realize os seguintes testes para validar seu código:

|  |  |  |  |
| --- | --- | --- | --- |
| Forma | Medida | Área | Volume |
| Esfera | 5 | 314,15 | 523,58 |
| Esfera | 2,5 | 78,53 | 65,54 |
| Esfera | 100 | 125663,70 | 4188790,20 |
| Cubo | 5 | 150 | 125 |
| Cubo | 2,5 | 37,5 | 15,62 |
| Cubo | 100 | 60000 | 1000000 |

O escopo de testes está incompleto, fique à vontade para ampliar o escopo de teste para mais valores.

Após terminar a primeira parte do projeto, tente criar funcionalidades adicionais e consertar alguns bugs como:

* Adicionar outras formas geométricas;
* Melhorar a formatação das saídas numéricas;
* Refatore o código e adicione potenciação nas fórmulas;
* as entradas de medidas com 3 casa decimais apresentam resultados errados quando são calculadas, identifique e conserte o erro.

# #3 Tarefa: Gado de Gerião

## Indicação e limite de crédito do cartão

Nessa tarefa precisaremos desenvolver um sistema para indicar o melhor cartão para as necessidades do cliente e posteriormente seu limite de crédito. No desafio não há nada novo, mas quanto mais códigos escrevemos, mais aprimoramos nossas habilidades em COBOL.

Para indicar um cartão, as regras do negócio exigem fazer algumas perguntas para o cliente, então vamos começar elaborando essas perguntas:

* 1ª Pergunta: Idade do cliente
  + 1 – Menor que 18 anos
  + 2 – Entre 18 e 29 anos
  + 3 – Entre 30 e 35 anos
  + 4 – Entre 36 e 40 anos
  + 5 – Mais de 40 anos
* 2ª Pergunta: Renda mensal do cliente
  + 1 – Menor que um salário-mínimo (R$ 1.500,00)
  + 2 – Entre R$ 1.500,00 e R$ 4.500,00
  + 3 – Entre R$ 4.600,00 e R$ 7.500,00
  + 4 – Entre R$ 7.600,00 e R$ 20.000,00
  + 5 – Mais de R$ 20.000,00
* 3ª Pergunta: Valor dos bens moveis e imóveis do cliente
  + 1 – Não possuo nenhum bem
  + 2 – Entre R$ 10.000,00 e R$ 50.000,00
  + 3 – Entre R$ 51.000,00 e R$ 100.000,00
  + 4 – Entre R$ 101.000,00 e R$ 1.000.000,00
  + 5 – Mais de R$ 1.000.000,00
* 4ª Pergunta: Regime profissional do cliente
  + 1 – Empregado CLT
  + 2 – Microempreendedor
  + 3 – Empresário de médio porte
  + 4 – Empresário industrial
  + 5 – Produtor rural

O usuário terá que escolher os números de 1 até 5 para responder as perguntas e para evitar erros de digitação será necessário implementar validação nas perguntas.

Cada número das opções apresentam uma pontuação específica:

* Opção 1 – 1 ponto
* Opção 2 – 5 pontos
* Opção 3 – 10 pontos
* Opção 4 – 20 pontos
* Opção 5 – 30 pontos

Após o usuário responder as perguntas, os pontos serão somados e um cartão será indicado baseado na pontuação, segue o exemplo:

* Cartão Pop = 4 pontos
* Cartão Bronze = 20 pontos
* Cartão Prata = 40 pontos
* Cartão Ouro = 80 pontos
* Cartão Black = 110 pontos

Os principais objetivos do sistema de indicação de cartão são os seguintes:

* Poder responder as perguntas do sistema;
* Após responder as perguntas o sistema devera somar corretamente os pontos e indicar um cartão correspondente com a pontuação

Exemplo:

>Qual sua idade  
>1- Menor de 18 anos

>2- Entre 18 e 29 anos

>3- Entre 30 e 35 anos

>4- Entre 36 e 40 anos

>5- Mais de 40 anos  
\* O usuário precisa digitar um número ente 1 e 5, caso ele digite o número errado uma mensagem de erro será disparada e a pergunta será refeita.

Após selecionar a opção desejada o usuário poderá responder as outras perguntas até chegar na pergunta final.

>Idade selecionada: opção 2

>Renda mensal selecionada: opção 3

>Valor dos bens selecionado: opção 3

>Regime profissional selecionado: opção 1

>Pontuação do cliente: 26 pontos

>Melhor cartão para o cliente: Cartão Bronze

Se tiver dúvidas para realizar as tarefas você pode conferir o código no Gitlab

ou pedir ajuda para algum colega.

Realize os seguintes testes para validar seu código:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| idade | Renda Mensal | Valor dos bens | Regime Profissional | Pontuação | Cartão |
| 25 | R$1900,00 | R$50.000,00 | CLT | 21 | Bronze |
| 41 | R$11.000,00 | R$550.000,00 | PJ | 80 | Ouro |
| 17 | R$1.000,00 | R$5.000 | CLT | 4 | Pop |
| 34 | R$5.000,00 | R$30.000,00 | MEI | 40 | Bronze |
| 62 | R$ 8450,00 | R$ 350.000,00 | Produtor Rural | 100 | Black |
| 31 | R$ 4400,00 | R$120.000,00 | PJ médio porte | 45 | Prata |
| 22 | R$3200,00 | R$ 75.000,00 | MEI | 25 | Bronze |

O escopo de testes está incompleto, fique à vontade para ampliar o escopo de teste para mais valores.

Após terminar a primeira parte do projeto, tente criar funcionalidades adicionais e consertar alguns bugs como:

* O EVALUATE das perguntas se repete muitas vezes e isso é uma péssima prática, crie um modulo para chamá-lo sempre que as perguntas forem feitas e evite repetição de código
* Adicione uma nova pergunta (se o cliente é aposentador, por exemplo) e um novo cartão
* Adicione um módulo para mostrar e modificar o limite do cartão do cliente

# #4 Tarefa: Corça de Cerínia

## Aptidão Profissional

Alocar o funcionário na vaga ou área correta é extremamente importante para a produtividade e principalmente para a retenção de talentos, nessa tarefa vamos criar um sistema de alocação com base em um questionário de aptidão profissional bem simples. Iremos aprender um pouco mais sobre o uso de condicionais (IF e ELSE) e controle de fluxo.

Elaboramos as seguintes perguntas para o questionário:

* 1ª Pergunta: Gosta de programar?
  + 1 – Sim
  + 2 – Não
* 2ª Pergunta: Prefere qual área? (se a resposta anterior for sim)
  + 1 – Backend
  + 2 – Frontend
* 3ª Pergunta: Já usou HTML e gostou? (se a resposta anterior for Frontend)
  + 1 – Sim
  + 2 – Não
* 4ª Pergunta: Qual sua faixa etária (se a resposta da 2ª pergunta for Backend)
  + 1 – Menos de 40 anos
  + 2 – Mais de 40 anos

O usuário terá que escolher os números de 1 até 2 para responder as perguntas e para evitar erros de digitação será necessário implementar validação nas perguntas.

Após responder as perguntas o usuário será levado para a respectiva profissão seguindo a lógica do fluxograma (elipses = perguntas e retângulos = respostas): Diagrama

Descrição gerada automaticamente

Se tiver dúvidas para realizar as tarefas você pode conferir o código no Gitlab

ou pedir ajuda para algum colega.

Após terminar a primeira parte do projeto, tente criar funcionalidades adicionais e consertar alguns bugs como:

* Inclua mais áreas de atuação como análise de dados e cyber segurança
* Crie um limite para as vagas e um módulo para alocar o funcionário para outra vaga automaticamente (antes indique que não há mais vagas disponíveis)

# #5 Tarefa: Touro de Creta

## Calculadora de aposentadoria

Você sabia que várias instituições de previdência social, espalhadas pelo mundo, ainda utilizam mainframes para processar a folha de pagamentos de milhões de aposentados? Nesse desafio vamos simular (se baseando pelo regime do INSS) como a aposentadoria de um beneficiário é calculada. Vamos utilizar os vários conceitos dos desafios anteriores como controle de fluxo e loops só que combinados em um programa mais robusto, mantendo as boas práticas de programação.

O sistema de aposentadoria do INSS possui vários critérios de aposentadoria:

* Por idade;
* Tempo de contribuição;
* Invalidez ou problemas de saúde;
* Especial.

No nosso sistema iremos calcular a aposentadoria por pontos que irá se basear na idade e no tempo de contribuição do beneficiário e vamos calcular a aposentadoria por idade também (esse caso serve para o beneficiário com pouco tempo de contribuição).

A aposentadoria por pontos funciona da seguinte forma:

* Média de todos os salários de contribuição desde 1994;
* A pontuação mínima desejada para se aposentar é de 100 pontos (65 anos de idade + 35 anos de contribuição);
* Precisa ter no mínimo 65 anos de idade para se aposentar;
* Atingir o tempo mínimo de contribuição (35 anos)
* O valor da aposentadoria será de 60% da média salarial de contribuição mais 2% a cada ano adicional depois do vigésimo ano de contribuição para os homens e no decimo quinto ano de contribuição para as mulheres;
* Para as mulheres o sistema de pontuação é um pouco diferente, a pontuação mínima é de 92 (63 anos de idade e 29 de contribuição).

Já aposentadoria por idade vai funcionar da seguinte forma:

* Ter a idade mínima de 65 anos para homens e 63 para mulheres;
* Ter contribuído por pelo menos 15 anos (vale para ambos os sexos);
* A aposentadoria será 70% da média salarial de contribuição.

Por exemplo:

> Idade do contribuinte: 70 anos

> Tempo de contribuição: 20 anos

> Média salarial de contribuição: R$ 8000,00

> Sexo do contribuinte: M (masculino)

> Aposentadoria por pontos: Pedido Recusado  
> Aposentadoria por idade: R$ 5600,00 de benefício

Se tiver dúvidas para realizar as tarefas você pode conferir o código no Gitlab

ou pedir ajuda para algum colega.

Realize os seguintes testes para validar seu código:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| idade | Tempo Contribuição | Média Salarial | Sexo | Aposentadoria por pontos | Aposentadoria por idade |
| 70 | 20 | R$ 8000,00 | M | RECUSADO | R$ 5600,00 |
| 65 | 40 | R$ 5000,00 | M | R$ 5000,00 | R$ 3500,00 |
| 55 | 40 | R$ 6000,00 | F | RECUSADO | RECUSADO |
| 60 | 35 | R$ 3000,00 | F | RECUSADO | RECUSADO |
| 62 | 35 | R$ 1800,00 | F | R$ 1800,00 | R$ 1260,00 |

O escopo de testes está incompleto, fique à vontade para ampliar o escopo de teste para mais valores.

Após terminar a primeira parte do projeto, tente criar funcionalidades adicionais e consertar alguns bugs como:

* Insira validação nas entradas dos dados para evitar erros;
* Uma boa prática de manutenção de código é evitar deixar números “soltos” quando for criar um programa.



No exemplo acima, os números “0,6”, “20” e “0,02” representam respectivamente a porcentagem de cálculo de aposentadoria a partir da média salarial de contribuição, o tempo mínimo para receber o valor adicional e o último número representa a porcentagem do valor adicional do valor do benefício. Em uma futura atualização, entender esses números soltos pode ser um pouco difícil, por isso o ideal seria criar variáveis para armazenar esses valores. Como tarefa adicional localize e substitua por variáveis todos os números “soltos” (criar um book seria um ótimo começo);

* Implemente uma funcionalidade que impeça o beneficiário de receber menos que um salário-mínimo e mais que o teto da aposentadoria. Crie variáveis para estabelecer o salário-mínimo e o teto (salário-mínimo: R$ 1500,00 e salário teto: R$ 10000,00)

# #6 Tarefa: Pomares das Hespérides

## Simulador de aplicação financeira

O mercado financeiro possui muitos produtos de investimentos com diferentes taxas de retornos, nesse desafio vamos criar um sistema para simular os retornos que esses diferentes produtos podem oferecer. Basicamente vamos criar uma calculadora de juros compostos com um menu para escolher entre os diferentes produtos, continuaremos aprimorando nossos conhecimentos básicos em COBOL, mas de forma consistente.

Uma calculadora de juros composta precisa das seguintes variáveis para funcionar:

* Valor inicial do investimento
* Taxa do produto escolhido somado a 1
* Tempo do investimento

O valor inicial do investimento e o tempo serão fornecidos pelo usuário, já as taxas serão determinadas pelas regras do negócio e terão os seguintes valores:

* CDB terá 1% de taxa
* LCI ou LCA terá 1,5% de taxa
* Fundos imobiliários terá 2% de taxa
* Fundos hedge 2,5% de taxa

Com as taxas determinadas, podemos continuar com a construção do sistema

Nosso sistema vai começar pedindo para o usuário escolher um produto para investir:

> Selecione seu produto de investimento:

> 1 – Investir em CDB

> 2 – Investir em LCI ou LCA

> 3 – Investir em fundos imobiliários

> 4 – Investir em fundos Hedge

O usuário precisa escolher entre as opções de 1 a 4, caso ele insira um valor errado, será necessário validar esse erro.

Após o usuário escolher uma opção correta, será necessário mostrar qual produto ele escolheu e logo em seguida perguntar o valor que ele gostaria de investir:

> Produto escolhido: \* Produto \*

> Quanto você quer investir R$: \* Quantia válida \*

Em seguida o usuário vai precisar inserir o tempo que ele deseja manter seu dinheiro investido:

> Quantos meses você quer manter o investimento: \* tempo válido \*

Por fim, um loop com o valor e o mês equivalente serão apresentados na tela:

> Resultado \* R$ ????,?? \* no mês ??

Por exemplo:

> Produto escolhido: 1 – CDB

> Quanto você quer investir R$: 1000,00

> Quantos meses você quer manter o investimento: 3

> Resultado R$ 1025,00 no mês 1

> Resultado R$ 1050,62 no mês 2

> Resultado R$ 1076,25 no mês 3

Se tiver dúvidas para realizar as tarefas você pode conferir o código no Gitlab

ou pedir ajuda para algum colega.

Realize os seguintes testes para validar seu código:

|  |  |  |  |
| --- | --- | --- | --- |
| Produto | Valor investido | Tempo (meses) | Resultado (último mês) |
| CDB | R$ 1000,00 | 3 | R$ 1032,20 |
| CDB | R$ 100.000,00 | 12 | R$ 112682,50 |
| CDB | R$ 0,50 | 12 | R$ 0,56 |
| LCI/LCA | R$ 1000,00 | 3 | R$ 1045,68 |
| LCI/LCA | R$ 100.000,00 | 12 | R$ 119561,82 |
| LCI/LCA | R$ 0,50 | 12 | R$ 0,59 |
| FI | R$ 1000,00 | 3 | R$ 1061,21 |
| FI | R$ 100.000,00 | 12 | R$ 126824,18 |
| FI | R$ 0,50 | 12 | R$ 0,63 |

O escopo de testes está incompleto, fique à vontade para ampliar o escopo de teste para mais valores.

Após terminar a primeira parte do projeto, tente criar funcionalidades adicionais e consertar alguns bugs como:

* Crie uma opção de investimento com taxas diferentes
* Faça um loop perguntando se o usuário gostaria de fazer uma nova operação após o fim da primeira
* insira um valor mínimo de investimento e um prazo máximo

# #7 Tarefa: Cinto de Hipólita

## Ranking e promoção de funcionários

Empresas que valorizam seus funcionários tendem a oferecer um plano de carreira com desafios e promoções a altura dos profissionais que elas desejam, para facilitar esse processo uma empresa quer implementar um sistema de promoção automatizado e nós seremos os responsáveis por criar esse sistema.

A empresa possui os seguintes critérios para promover um funcionário:

* Tempo que o funcionário está na empresa;
* Formação acadêmica do funcionário;
* Cursos internos que o funcionário já concluiu;
* Certificações do funcionário.

Cada critério possui níveis onde o funcionário irá se encaixar e cada nível tem uma determinada pontuação:

* Tempo de empresa:

1 – Um ano ou menos (1 ponto)

2 – Entre 2 e 4 anos (5 pontos)

3 – Entre 5 e 9 anos (10 pontos)

4 – Entre 10 e 19 anos (20 pontos)

5 – Mais de 20 anos (30 pontos)

* Formação acadêmica:

1 – Ensino Médio (1 ponto)

2 – Graduação (5 pontos)

3 – Pós-graduação (10 pontos)

4 – Mestrado (20 pontos)

5 – Doutorado (30 pontos)

* Cursos interno realizados:

1 – Entre 2 e 10 cursos (1 ponto)

2 – Entre 11 e 25 cursos (5 pontos)

3 – Entre 26 e 39 cursos (10 pontos)

4 – Entre 40 e 59 cursos (20 pontos)

5 – Mais de 60 cursos (30 pontos)

* Quantidade de certificações do funcionário

1 – 1 Certificação (1 ponto)

2 – 3 Certificações (5 pontos)

3 – 5 Certificações (10 pontos)

4 – 8 Certificações (20 pontos)

5 – Mais de 10 Certificações (30 pontos)

Após o funcionário responder as perguntas, seus pontos serão somados e um cálculo irá determinar o cargo que combina com a devida pontuação:

* Pontuação menor ou igual a 20 pontos: Cargo de Trainee
* Pontuação entre 20 e 40 pontos: Cargo de júnior
* Pontuação entre 40 e 60 pontos: Cargo de pleno
* Pontuação entre 60 e 90 pontos: Cargo de sênior
* Pontuação entre 90 e 120 pontos: Cargo de gerente

Por exemplo:

> Selecione o tempo de empresa: \* Opção 2 \*

> Selecione a formação acadêmica: \* Opção 2 \*

> Cursos internos realizados: \* Opção 3 \*

> Quantidade de certificações: \* Opção 1 \*

> Pontos do funcionário: 21 pontos  
> Cargo que o funcionário pode ocupar: Cargo de júnior

Lembre-se que os dados precisam ser validados, se tiver dúvidas para realizar as tarefas você pode conferir o código no Gitlab ou pedir ajuda para algum colega.

Realize os seguintes testes para validar seu código:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tempo de empresa | Formação acadêmica | Qtd. Cursos internos | Qtd. certificações | pontos | cargo |
| 9 anos | Graduado | 12 | 4 | 25 | JUNIOR |
| 5 anos | Pós-graduado | 25 | 6 | 55 | PLENO |
| 21 anos | Ensino médio | 50 | 2 | 27 | JUNIOR |
| 3 anos | Mestrado | 62 | 10 | 85 | SENIOR |
| 3 meses | Doutorado | 2 | 5 | 42 | PLENO |
| 12 anos | Mestrado | 65 | 12 | 100 | GERENTE |

O escopo de testes está incompleto, fique à vontade para ampliar o escopo de teste para mais valores.

Após terminar a primeira parte do projeto, tente criar funcionalidades adicionais e consertar alguns bugs como:

* As vezes durante a elaboração de uma regra de negócio podem ocorrer falhas no processo, por exemplo durante a seleção de certificações não passou pela cabeça do Product Owner que o funcionário pode ter 9 certificações (e não está no escopo) por isso vamos corrigir essa falha e colocar as certificações entre intervalos, e ficará a seu critério decidir esses intervalos
* O cliente agora quer um quinto critério de avaliação chamado “projetos realizados” que terá a seguinte estrutura de pontuação:

1 – Entre 2 e 4 projetos realizados (1 ponto)

2 – Entre 5 e 7 projetos realizados (5 pontos)

3 – Entre 8 e 12 projetos realizados (10 pontos)

4 – Entre 13 e 16 projetos realizados (20 pontos)

5 – Mais de 17 projetos realizados (30 pontos)

A pontuação também terá um acréscimo de 30 pontos e vai até 150 pontos, o cliente quer que a nova pontuação seja alterada para os cargos e pediu que você distribua a nova pontuação entre os cargos

# #8 Tarefa: Éguas de Diomedes

## Calculadora de prêmio de seguro veicular

O setor de seguros, principalmente o de seguros veicular, é extremamente lucrativo e uma área muito importante para vários bancos, nesse desafio vamos criar uma calculadora simples para simular um modelo de seguros veicular. Vamos aprender um conceito que está em quase todas as linguagens de programação que é o reaproveitamento de código, nesse desafio o código é muito parecido com o do desafio #7 o que torna a tarefa atual mais familiar e fácil de realizar.

O banco que nos contratou para elaborar esse simulador elaborou as seguintes regas de negócio:

* O prêmio será calculado utilizando a média aritmética dos seguintes critérios

1 – Idade (entre 18 e 29 anos = 2 pontos, entre 30 e 59 anos = 1 ponto e mais de 60 anos = 1 ponto)

2 – Gênero (feminino = 1 ponto e masculino = 2 pontos)

3 – Estado civil (casado = 1 ponto, viúvo = 2 pontos e solteiro = 2 pontos)

4 – Histórico de direção (bom (poucas multas) = 1 ponto e mal (muitas multas = 2 pontos))

5 – Tipo do veículo (popular = 1 ponto e luxo = 2 pontos).

* A média da pontuação será fornecida somando os pontos do questionário respondido pelo usuário divido por 5. Por exemplo, se os pontos de um usuário foram iguais a 7, então dividimos por 5 e o resultado será 1,4.
* A pontuação média dará o retorno da porcentagem do prêmio do seguro do cliente da seguinte forma:

1 – Média da pontuação = 1 (prêmio mensal de 1% do valor do veículo)

2 – Média da pontuação entre 1 e 2 (prêmio mensal de 3% do valor do veículo)

3 – Média da pontuação = 2 (prêmio mensal de 5% do valor do veículo).

Com as regras do negócio já elaborado, vamos criar as perguntas para o usuário:

* Selecione a idade do cliente:

1 – Entre 18 e 29 anos

2 – Entre 30 e 59 anos

3 – Mais de 60 anos

* Selecione o gênero do cliente:

1 – Masculino

2 – Feminino

* Selecione o estado civil do cliente:

1 – Casado

2 – Viúvo

3 – Solteiro

* Selecione o histórico de direção do cliente

1 – Bom (poucas multas)

2 – Ruim (muitas multas)

* Selecione o tipo de veículo do cliente

1. Popular
2. Luxo

Após o usuário responder o questionário, a pontuação média será calculada, mas antes de calcular o prêmio será necessário entrar com o valor do veículo do cliente seguindo o seguinte exemplo:

> Selecione a idade do cliente: \* 20 anos \*

> Selecione o gênero do cliente: \* masculino \*

> Selecione o estado civil do cliente: \* casado \*

> Selecione o histórico de direção do cliente: \* bom \*

> Selecione o tipo de veículo do cliente: \* popular \*  
> Digite o valor do veículo do cliente: \* R$ 50.000,00 \*

> Valor mensal do seguro: \* R$ 1.500,00 \*

Lembre-se que os dados precisam ser validados, se tiver dúvidas para realizar as tarefas você pode conferir o código no Gitlab ou pedir ajuda para algum colega.

Realize os seguintes testes para validar seu código:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| idade | gênero | Estado civil | Histórico de direção | Tipo do veículo | Valor do veículo | Valor do seguro |
| 25 | Feminino | Casado(a) | bom | Popular | R$ 70.000,00 | R$ 2.100,00 |
| 32 | Feminino | Viúvo(a) | ruim | luxo | R$120.000,00 | R$ 3.600,00 |
| 61 | Feminino | Solteiro(a) | bom | popular | R$ 40.000,00 | R$ 1.200,00 |
| 18 | Masculino | Solteiro(a) | bom | popular | R$80.000,00 | R$ 2.400,00 |
| 65 | Masculino | Casado(a) | bom | popular | R$ 60.000,00 | R$1.800,00 |
| 22 | Masculino | Viúvo(a) | ruim | luxo | R$130.000,00 | R$6.500,00 |

O escopo de testes está incompleto, fique à vontade para ampliar o escopo de teste para mais valores.

Após terminar a primeira parte do projeto, tente criar funcionalidades adicionais e consertar alguns bugs como:

* O cliente não quer fazer seguro para carros com valores menores que R$ 35.000,00 e maiores que R$150.000,00 faça uma validação para não aceitar os dados inválidos e dispare uma mensagem
* O código possui muita redundância (código repetido) na parte do EVALUATE SELEC-MENU dos critérios do cliente, crie uma função para chamar o EVALUATE e evitar repetição de código desnecessária.

# #9 Tarefa: Estábulos de Áugias

## Calculadora de índice de massa corporal (IMC)

Pensando no segmento de saúde e bem-estar, vamos criar uma calculadora de índice de massa corporal. O índice de massa corporal ou IMC, serve para determinar se uma pessoa está no seu peso ideal levando em conta sua altura e peso atual. Nesse desafio vamos tentar algo diferente dos outros desafios, mas que é muito comum no dia a dia de um programador: manutenção de código legado. Comece o desafio pelo programa incompleto no gitlab e adicione o restante das funcionalidades que o desafio exige.

Agora vamos entender as seguintes regras do negócio antes de começar:

* O cálculo será feito da seguinte forma:
* O usuário precisara inserir sua massa corporal em kg mas não poderá ser menor que 5kg e nem maior que 500kg, do contrário uma mensagem de erro será retornada e as entradas serão refeitas
* A altura também será digitada pelo usuário e será em metros e não serão aceitas entradas com menos de 0,5 metro e nem entradas com mais de 3 metros, a validação precisa ser feita aqui também da mesma forma que na massa corporal.
* Para recomendar dieta e treino vamos seguir a seguinte tabela de IMCs:

|  |  |  |  |
| --- | --- | --- | --- |
| IMC | Categoria | Dieta | Treino |
| Menor que 16 | Baixo peso III | Rica em carboidratos | Musculação |
| Ente 16 e 17 | Baixo peso II | Rica em carboidratos | Musculação |
| Entre 17 e 18,5 | Baixo peso I | Rica em carboidratos | Musculação |
| Entre 18,5 e 25 | Peso Ideal | Balanceada | Musculação |
| Entre 25 e 30 | Sobrepeso | Rica em fibras e proteínas | Aeróbico |
| Entre 30 e 35 | Obesidade I | Rica em fibras e proteínas | Aeróbico |
| Entre 35 e 40 | Obesidade II | Rica em fibras e proteínas | Aeróbico |
| Maior que 40 | Obesidade III | Rica em fibras e proteínas | Aeróbico |

* Após o usuário digitar seu peso e altura, o sistema irá calcular seu IMC e determinar sua categoria de peso, dieta e treino.

Com as regras do negócio já estabelecidas, vamos criar as perguntas para o usuário:

* Digite o peso do aluno em KG:
* Digite a altura do aluno em metros:

As saídas de dados deverão seguir dessa forma:

> Digite o peso do aluno em KG: \* 80 \*

> Digite a altura do aluno em metros: \* 1,8 \*

> IMC do aluno: \* 24,6 \*

> Categoria de peso do aluno: \* Peso ideal \*

> Dieta ideal para o aluno: \* Balanceada \*  
> Tipo de treino para o aluno: \* Musculação \*

Lembre-se que os dados precisam ser validados, se tiver dúvidas para realizar as tarefas você pode conferir o código no Gitlab ou pedir ajuda para algum colega.

Realize os seguintes testes para validar seu código:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Peso (KG) | Altura (M) | IMC | Categoria | Dieta | Treino |
| 80 | 1,8 | 24,6 | Peso Ideal | Balanceada | Musculação |
| 70 | 1,92 | 18,9 | Peso Ideal | Balanceada | Musculação |
| 85 | 1,54 | 35,8 | Obesidade II | Rica em fibras e proteínas | Aeróbico |
| 110 | 1,78 | 34,7 | Obesidade I | Rica em fibras e proteínas | Aeróbico |
| 50 | 1,85 | 14,6 | Baixo peso III | Rica em carboidratos | Musculação |
| 120 | 1,95 | 31,5 | Obesidade I | Rica em fibras e proteínas | Aeróbico |

O escopo de testes está incompleto, fique à vontade para ampliar o escopo de teste para mais valores.

Após terminar a primeira parte do projeto, tente criar funcionalidades adicionais e consertar alguns bugs como:

* Crie uma funcionalidade para retornar quanta massa corporal um usuário deve ganhar ou perder para chegar no peso ideal.

# #10 Tarefa: Javali de Erimanto

## Análise de perfil do investidor

Uma ferramenta muito importante e indispensável para bancos e corretoras que oferecem produtos de investimentos, a análise de perfil do investidor (API) é crucial para atender as demandas de investimentos do cliente. Nesse desafio vamos criar um simulador de API e ver como funciona essa ferramenta.

regras do negócio para desenvolver a ferramenta:

* Serão feitas 5 perguntas para entender qual o melhor perfil do cliente e cada pergunta possui uma pontuação que será somada no final do questionário e a partir da contagem dos pontos o tipo do investidor será selecionado
* Perfis do investidor que podem ser selecionados:

1. – Conservador
2. – Moderado
3. - Arrojado

* Pergunta número 1: Qual seu principal objetivo ao investir?

1 – Acumular capital para aposentadoria (1 ponto)

2 – Obter renda para comprar um bem (2 pontos)

3 – Lucrar o máximo possível no curto prazo (3 pontos)

* Pergunta número 2: Qual sua tolerância ao risco?

1 – Não tolero risco, prefiro ativos seguros (1 ponto)

2 – Permito alguns ativos de risco na carteira (2 pontos)

3 – Prefiro ativos com maiores riscos (3 pontos)

* Pergunta número 3: Prazo previsto para manter seus investimentos?

1 – Mais de 10 anos (1 ponto)

2 – Entre 5 e 10 anos (2 pontos)

3 – Entre 1 e 5 anos (3 pontos)

* Pergunta número 4: Indique seu conhecimento sobre investimentos?

1 – Não conheço nada sobre investimentos (1 ponto)

2 – Tenho um conhecimento superficial (2 pontos)

3 – Conheço bem o mercado financeiro (3 pontos)

* Pergunta número 5: Qual sua preferência na liquidez dos ativos?

1 – Posso esperar até 1 anos para sacar (1 ponto)

2 – Posso esperar até 6 meses para sacar (2 pontos)

3 – Saque imediato ou até 3 dias (3 pontos)

* O tipo do investidor será determinado pelas seguintes pontuações:

1 – Investidor conservador: pontuação entre 5 e 9

2 - Investidor moderado: pontuação entre 9 e 13

3 – Investidor arrojado: pontuação maior que 13

As saídas dos dados deverão seguir a seguinte forma:

> Qual seu principal objetivo ao investir? \* opção 1 \*

> Qual sua tolerância ao risco? \* opção 2\*

> Prazo previsto para manter seus investimentos? \* opção 3 \*

> Indique seu conhecimento sobre investimentos? \* opção 1\*

> Qual sua preferência na liquidez dos ativos? \* opção 3\*

> Perfil do cliente: \* Moderado \*  
> Melhores investimentos para o perfil do cliente:

1 – Fundos de investimentos

2 – Fundos imobiliários

3 – CDB, LCI ou LCA

Lembre-se que os dados precisam ser validados, se tiver dúvidas para realizar as tarefas você pode conferir o código no Gitlab ou pedir ajuda para algum colega.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Pergunta1 | Pergunta 2 | Pergunta 3 | Pergunta 4 | Pergunta 5 | Perfil | Investimentos |
| Opção 1 | Opção 1 | Opção 1 | Opção 1 | Opção 1 | conservador | Pensão, renda fixa e Ourocap |
| Opção 1 | Opção 2 | Opção 1 | Opção 3 | Opção 3 | Moderado | FIF, FIIs e renda fixa |
| Opção 2 | Opção 2 | Opção 3 | Opção 3 | Opção 2 | Moderado | FIF, FIIs e renda fixa |
| Opção 3 | Opção 3 | Opção 1 | Opção 3 | Opção 3 | Moderado | FIF, FIIs e renda fixa |
| Opção 2 | Opção 3 | Opção 3 | Opção 3 | Opção 3 | Arrojado | ETF, ações, cripto e COE |
| Opção 1 | Opção 2 | Opção 3 | Opção 3 | Opção 1 | Moderado | FIF, FIIs e renda fixa |

Realize os seguintes testes para validar seu código:

O escopo de testes está incompleto, fique à vontade para ampliar o escopo de teste para mais valores.

Após terminar a primeira parte do projeto, tente criar funcionalidades adicionais e consertar alguns bugs como:

* Adicione o perfil “Agressivo” e altere as pontuações
* Combine esse programa com o desafio #6 (aplicação financeira) para criar um produto único e mais completo. Não esqueça de fazer as adaptações necessárias para o programa funcionar corretamente.

# #11 Tarefa: Hidra de Lerna

## CRUD de peças

Poder armazenar dados e acessá-los posteriormente é fundamental para qualquer empresa que dependa de uma boa infraestrutura de TI. Backups rotineiros e acesso rápido e eficiente aos dados é a diferença entre uma empresa competitiva no mercado e outra que está à beira da falência. Essa tarefa irá mostrar um código COBOL que realiza um CRUD para armazenar dados de peças de veículos. CRUD é um conceito básico para criar bancos de dados e seu acrônimo significa Create, Read, Update e Delete.

Por se tratar de um desafio complexo, o recomendável seria apenas baixar o desafio no Gitlab sem realizar nenhuma tarefa, apenas entender o código e entender como o programa realiza a criação do arquivo, o acesso e a modificação dos dados. Se preferir pode realizar algumas melhorias no programa ou adicionar novas funcionalidade, a intenção desse desafio e dar um descanso para a última tarefa.

# #12 Tarefa: Guardião dos mortos

## Simulador de empréstimo bancário

Um dos principais serviços de um banco, os sistemas de empréstimos bancários representam boa parte da receita dessas instituições e por isso entender como eles funcionam é importante. Na nossa última tarefa, vamos criar um simulador de empréstimo utilizando os conhecimentos que aprendemos ao longo dessa jornada, vamos combinar partes dos desafios anteriores e adicionar um pouco mais de complexidade (e quantidade de linhas de código) nesse desafio.

regras do negócio do simulador de empréstimo bancário:

* Selecionar o perfil do cliente por meio de um questionário para determinar o limite do empréstimo
* O questionário terá 4 critérios para determinar o limite com 5 perguntas por critério, os critérios são os seguintes:

1. Idade
2. Renda
3. Valor dos bens
4. Regime profissional

As perguntas terão uma pontuação que será calculada para determinar o limite de crédito, fique à vontade para determinar as perguntas dos critérios.

* Após o questionário, o sistema irá calcular o valor limite do empréstimo para o usuário a partir das seguintes pontuações:

1. Pontuação menor ou igual a 20: R$ 5.000,00 de limite
2. Pontuação menor ou igual a 40: R$ 10.000,00 de limite
3. Pontuação menor ou igual a 60: R$ 20.000,00 de limite
4. Pontuação menor ou igual a 90: R$ 50.000,00 de limite
5. Pontuação menor ou igual a 120: R$ 100.000,00 de limite

* Com o limite estabelecido o cliente poderá escolher o valor do empréstimo, o valor precisa ser maior que R$ 100 e menor que o limite estabelecido
* E por fim, o cliente terá a possibilidade de escolher o prazo de pagamento e as taxas, quanto mais tempo o cliente escolher maiores serão os juros:

1. Pague em até 5 meses com juros de 1,2% ao mês
2. Pague em até 10 meses com juros de 1,5% ao mês
3. Pague em até 15 meses com juros de 2% ao mês
4. Pague em até 20 meses com juros de 2,5% ao mês

* Finalizando a escolha do prazo, o sistema mostrará a quantidade de parcelas selecionas, o valor de cada parcela e o valor total do empréstimo somando os juros com a quantia tomada

As saídas de dados deverão seguir dessa forma:

> Qual sua idade: 20 anos

> Valor da renda mensal: R$ 3.500,00

> Valor dos bens moveis e imóveis: R$ 15.000,00

> Indique seu regime profissional: Empregado CLT  
> Limite de crédito determinado pelo sistema: R$ 5.000,00

> Digite o valor de seu empréstimo: R$ 1.000,00  
> Selecione os melhores prazos e taxas: 5 meses

> Parcelas a pagar: 5

> Valor de cada parcela: R$ 212,00

> Valor total do empréstimo: R$ 1.060,00

Lembre-se que os dados precisam ser validados, se tiver dúvidas para realizar as tarefas você pode conferir o código no Gitlab ou pedir ajuda para algum colega.

Realize os seguintes testes para validar seu código:

* Testes das perguntas e limite de crédito

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| idade | Renda mensal | Valor dos bens | Regime profissional | Limite de crédito |
| 25 | R$ 2.500,00 | R$ 6.000,00 | CLT | R$ 5.000,00 |
| 38 | R$ 4.800,00 | R$ 250.000,00 | MEI | R$ 20.000,00 |
| 45 | R$ 7.200,00 | R$ 100.000,00 | CLT | R$ 20.000,00 |
| 62 | R$ 15.700,00 | R$ 825.000,00 | Produtor rural | R$ 100.000,00 |
| 17 | R$ 950,00 | R$ 0,00 | CLT | R$ 5.000,00 |
| 58 | R$ 30.000,00 | R$ 1.250.000,00 | PJ Industrial | R$ 100.000,00 |

* Testes dos valores das parcelas e do total do empréstimo (continuação dos testes anteriores)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Valor do empréstimo | prazo | juros | Valor da parcela | Valor total do empréstimo |
| R$ 1.000,00 | 5 | 1,2% | R$ 212,00 | R$ 1.060,00 |
| R$ 5.000,00 | 10 | 1,5% | R$ 575,00 | R$ 5.750,00 |
| R$ 15.000,00 | 20 | 2,5% | R$ 1.125,00 | R$ 22.500,00 |
| R$ 50.000,00 | 15 | 2% | R$ 4.333,33 | R$ 64.999,95 |
| R$ 4.000,00 | 20 | 2,5% | R$ 300,00 | R$ 6.000,00 |
| R$ 85.000,00 | 15 | 2% | R$ 7366,66 | R$ 110499,90 |

O escopo de testes está incompleto, fique à vontade para ampliar o escopo de teste para mais valores.

Após terminar a primeira parte do projeto, tente criar funcionalidades adicionais e consertar alguns bugs como:

* Além de mostrar o valor de cada parcela e o valor total do empréstimo, exiba também o valor total dos juros do empréstimo e sua porcentagem, por exemplo:   
  1 - Valor do empréstimo: R$ 1.000,00

2 – Prazo: 5 meses com 1,2% de juros ao mês

3 – Valor da parcela: R$ 212,00

4 – Valor total do empréstimo: R$ 1060,00

5 – Valor total dos juros: R$ 60,00

6 – Percentual dos juros: 6%

* Para evitar casos de inadimplência, modifique o limite de empréstimo para R$ 2.000,00 para clientes com pontuação menor que 10.