**ATP Raciocínio computacional**

Em TI, existem inúmeras linguagens de programação. Apesar das diferentes discussões sobre o uso de cada uma ou, ainda, sobre qual seria a melhor, note que não existe a “melhor linguagem de programação”. Na realidade, há uma linguagem que pode funcionar melhor do que outra em determinado cenário ou sob certo contexto – algumas possuem performance melhor que outras, mas podem necessitar de uma curva de aprendizagem maior; outras funcionam melhor para celulares e sistemas embarcados ou, ainda, para grandes servidores; ainda, algumas servem muito bem em um sistema operacional, mas não em outros, e vice-versa.

Dito isso, o Python é uma linguagem de programação muito versátil e utilizada nas áreas de inteligência artificial (o que inclui ciência de dados), desenvolvimento web e automações. Seu código é mais legível que algumas outras linguagens e é mais rápido de aprender a programar com ele quando não se é iniciante. Como contraponto, certamente existem linguagens que possuem uma performance muito melhor, mas são mais complexas de usar (como o próprio C e C++, por exemplo). Por isso, consideramos o Python um ótimo ponto de partida.

Em conjunto com o conteúdo disponibilizado nesta disciplina, propomos a você o desenvolvimento de uma Atividade Prática (ATP). Trata-se de uma atividade na qual desenvolverá um algoritmo em Python utilizando o conhecimento teórico adquirido ao longo da disciplina, em **quatro** etapas. Assim sendo, com o conteúdo que estudou nas semanas 1 e 2, terá insumos para desenvolver a etapa 1, e assim sucessivamente. Como uma etapa utiliza o que foi desenvolvido na anterior, evite pulá-las e/ou executá-las fora da ordem recomendada.

Você não precisará entregar o seu código referente às diferentes etapas **enquanto** estão em andamento – a divisão em etapas serve para que consiga melhor organizar as suas atividades e evitar contratempos. Na semana 5, nós teremos um checkpoint. Nesse momento, disponibilizaremos um checklist para que consiga se autoavaliar e resolver possíveis erros a tempo. É extremamente recomendável que gerencie o seu tempo para que possa desenvolver o código referente a cada uma das etapas de acordo com as semanas propostas, evitando contratempos e mitigando as possíveis dúvidas com a dedicação requerida.

Após finalizar todas as etapas, durante as últimas semanas da disciplina, você fará a entrega do seu código já finalizado. Essa entrega englobará tudo que desenvolveu durante as etapas em um único arquivo contendo o código. Sendo assim, note que não há necessidade de criar um relatório e/ou outros documentos. Por outro lado, como a sua única entrega será o código, utilize esse tempo para deixá-lo bem-feito e preparado. Empregue, se necessário, o material de apoio disponibilizado em conjunto com o material padrão da disciplina, para obter dicas adicionais de preparação do seu código.

Nesta ATP, você desenvolverá o sistema de uma loja em Python. O conceito de uma loja é interessante de ser trabalhado por se tratar de algo conhecido e com validação fácil (uma loja possui produtos que são cadastrados e vendidos, independentemente do local, e eles sempre terão pelo menos uma descrição e um preço). Assim, você poderá comprovar a sua capacidade de compreender e transformar especificações do mundo real em algoritmos e identificar a sequência lógica de etapas para a resolução de problemas, gerando um algoritmo. Essa capacidade é chave no mercado de TI e reflete que, ao dar um volume suficiente de informações, será capaz de resolver situações-problema com autonomia – especificamente neste ponto, é relevante ressaltar que hoje existe uma miríade de desenvolvedores que trabalham em home office para empresas do mundo todo, sem, muitas vezes, se reunir cara a cara com seus clientes uma única vez.

**Etapa 1**

Nesta primeira etapa, vamos começar a construir o sistema da loja. Até o momento, você aprendeu a trabalhar com variáveis em Python e com a entrada e saída de dados (utilizando as funções print e input).

Assim, para esta etapa, faça um algoritmo que atenda às seguintes instruções:

A - Mostre ao usuário o seu **nome completo**, junto do nome da sua loja (**não peça ao usuário** o nome dele, apenas o mostre).

1 - Exemplo: se o seu nome for João Flores da Silva, pode mostrar “Bem-vindo à Loja do João Flores da Silva” ou “”.

B - Diga ao usuário que fará uma análise de crédito dele. Para tal, peça que digite o **cargo** na empresa em que trabalha atualmente, o **salário** e o **ano** de nascimento.

C - Mostre ao usuário o **cargo**, o **salário** e o **ano de nascimento** que digitou.

**Etapa – 2**

Agora, vamos fazer uma análise de crédito do usuário para saber quanto ele poderá comprar na nossa loja. Assim, continue do ponto em que parou na etapa 1 e, usando o mesmo código, adicione os seguintes passos:

1. Mostre na tela a **idade** aproximada do usuário. Você pode fazer isso ao subtrair o ano em que estamos pelo ano de nascimento que ele digitou.
2. Mostre quanto o cliente **poderá gastar** na sua loja (o limite de gasto), calculado da seguinte forma: [salário x (idade / 1.000)] + 100.

**Etapa – 3**

Agora, vamos verificar se o produto que solicitamos ao usuário pode ser realmente comprado por ele (ou não). Aqui, vamos usar o **seu nome completo** para montar a lógica.

**Atenção!** Vamos usar como lógica o **seu nome completo e não o nome do usuário**. A partir de agora, quando falarmos “quantidade de caracteres do seu primeiro nome”, equivale à quantidade de letras do seu primeiro nome. Por exemplo, se o seu nome completo é **João Flores da Silva**, isso seria igual a quatro. Quando falarmos “quantidade de caracteres do seu nome completo”, contaremos também espaços e hifens. No mesmo exemplo, seria igual a 20.

1. Solicite ao cliente que digite o **nome** de um produto e o **preço** dele.
2. Se o valor do produto for menor ou igual a 60% do limite que o cliente tem para gastar, mostre a mensagem “Liberado!”. Se estiver entre 60% e 90%, mostre a mensagem “Liberado ao parcelar em até 2 vezes”. Se estiver entre 90% e 100%, mostre a mensagem “Liberado ao parcelar em 3 ou mais vezes”. Caso contrário, mostre a mensagem “Bloqueado”.
3. Se o valor do produto estiver entre a **quantidade de caracteres do seu nome completo** e a **idade do cliente**, mostre que ele terá um desconto igual à **quantidade de caracteres do seu primeiro nome**.
4. Mostre também ao cliente o valor do produto **com o desconto**.

**Etapa – 4**

Vamos garantir que o nosso código possa funcionar para vários produtos? Para tanto, vamos modularizá-lo (isto é, criar funções) para eles, em conjunto com estruturas de repetição como for e while.

Execute os seguintes passos:

1 - Coloque o código que você fez nas **etapas 1 e 2** dentro de uma única **função** chamada “obter\_limite”. Essa função deverá **retornar** o limite que o usuário poderá gastar.

2 - Coloque o código que você fez na **etapa 3** dentro de uma única **função** chamada “verificar\_produto”. Essa função terá como **parâmetro** de entrada o limite de gasto do cliente.

3 - Após o cliente informar os dados dele (pela função “obter\_limite”), armazene o limite que ele poderá gastar dentro de uma variável chamada “limite”.

4 - Na sequência, pergunte ao usuário quantos produtos deseja cadastrar.

5 - Por fim, utilize uma estrutura de repetição (**for** ou **while**) por **n** vezes, com **n** equivalendo à quantidade de produtos que ele deseja cadastrar.

6 - Dentro da sua estrutura de repetição, chame sua função “verificar\_produto”, permitindo que o usuário consiga cadastrar todos os produtos e verificar se teria ou não limite sobrando para comprá-los.