



PUCPR
GRUPO MARISTA



Sumário

O QUE É UMA ATIVIDADE PRÁTICA?	2
COMO SEREI AVALIADO?.....	2
QUAL É O VALOR (NOTA) DA ATP?	2
DESCRIÇÃO GERAL DA ATP: LOJA EM PYTHON.....	3
ATP – ETAPA 1.....	5
ATP – ETAPA 2.....	5
ATP – ETAPA 3.....	5
ATP – ETAPA 4.....	6
CRITÉRIOS AVALIATIVOS OU RUBRICAS DA ATP	7

O QUE É UMA ATIVIDADE PRÁTICA?

A **Atividade Prática (ATP)** tem como proposta ser voltada para sua atuação no mundo do trabalho, visando à sua formação profissional e resultando, no final da disciplina, em um “produto”, o qual pode ser.

COMO SEREI AVALIADO?

Após entregar o produto de sua ATP, a correção feita pelo professor-tutor levará em conta **critérios avaliativos ou rubricas preestabelecidos**. Esses critérios são elaborados considerando o que é solicitado nas etapas da ATP e sua relação com os RAs e IDs estabelecidos para a disciplina.

Veja um exemplo:

Para correção, o professor-tutor estará pautado em um quadro como o apresentado na sequência. A **coluna 1** corresponde **aos critérios avaliados**, ou seja, os pontos que serão observados no produto entregue (note que **cada uma das linhas representa um critério**). As **colunas 2 a 4** representam os **descritores**, ou seja, a classificação que será aplicada no produto entregue, os quais possuem **níveis de desempenho**, do Autônomo ao Em desenvolvimento.

CRITÉRIOS	DESCRITORES			
	Autônomo (100% a 90%)	Capaz (89% a 70%)	Aprendiz (69% a 40%)	Em desenvolvimento (39% a 0%)
Critério 1. (20%)	Descrição do “produto” que atinge entre 90-100% do que se espera.	Descrição do “produto” que atinge entre 89-70% do que se espera.	Descrição do “produto” que atinge entre 69-40% do que se espera.	Descrição do “produto” que atinge entre 39-0% do que se espera.
Critério 2. (25%)	Descrição do “produto” que atinge entre 90-100% do que se espera.	Descrição do “produto” que atinge entre 89-70% do que se espera.	Descrição do “produto” que atinge entre 69-40% do que se espera.	Descrição do “produto” que atinge entre 39-0% do que se espera.
...

O entendimento desses critérios avaliativos auxilia na transparência do processo de elaboração e correção da atividade; assim, é importante conhecê-los para que saiba quais são as expectativas estabelecidas para o produto final de sua ATP.

Para saber quais são os critérios avaliativos da ATP da disciplina, consulte a última página deste documento.

QUAL É O VALOR (NOTA) DA ATP?

O produto da ATP vale 10 pontos, sendo o peso da atividade na sua média final 3, ou seja, é proporcional a 30% da média final.

Relembre o cálculo da sua média final:

$$\text{MÉDIA FINAL} = (\text{AE} \times 0,2) + (\text{ATP} \times 0,3) + (\text{ARP} \times 0,5)$$

Legenda:

AE: Atividade de Estudo 1 e 2

ATP: Atividade Prática

ARP: Avaliação Regular Presencial

Agora que você compreendeu o que é uma ATP e como desenvolvê-la, faça a leitura da proposta de atividade desta disciplina e inicie o desenvolvimento das etapas.

Bons estudos!

DESCRIÇÃO GERAL DA ATP: Loja em Python

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.

Finalmente, recomendamos que busque quebrar as etapas em pedaços menores, resolvendo-os um de cada vez. Assim, as chances de se confundir durante o desenvolvimento e/ou perder muito tempo em problemas triviais diminuem. Adicionalmente, note que as mensagens de erro em Python costumam ser muito didáticas (ainda que sejam em inglês). Logo, caso venha a encontrar algum erro (geralmente iniciado pelo termo “*traceback*”) enquanto executa o seu código, recomendamos que pesquise no site da documentação oficial do Python (<https://docs.python.org/3/library/exceptions.html>), no Fórum de Dúvidas da disciplina ou, ainda, em fóruns especializados (como <https://stackoverflow.com>).

QUADRO-RESUMO DA ATP		
Semana	Etapas	Atividade
2	1	Construir o sistema da loja desenvolvendo um algoritmo específico
3	2	Fazer uma análise de crédito do usuário
4	Checkpoint	-
5	3	Verificar se o produto que solicitamos ao usuário pode ser realmente comprado por ele
6	4	Modularizar o código (isto é, criar funções) para outros produtos
7 e 8	Finalização e entrega	Entrega do código-fonte (arquivo de extensão .py) que contém o código em Python desenvolvido.

ATP - 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:

1. Mostre ao usuário o seu **nome completo**, junto do nome da sua loja (**não peça ao usuário** o nome dele, apenas o mostre).
 - a. Exemplo: se o seu nome for João Flores da Silva, pode mostrar “Bem-vindo à Loja do João Flores da Silva” ou “Esta é a Loja Desconto da Cidade, bem-vindo! Aqui quem fala é João Flores da Silva”.
2. 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**.
3. Mostre ao usuário o **cargo**, o **salário** e o **ano de nascimento** que digitou.

ATP - 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: $[\text{salário} \times (\text{idade} / 1.000)] + 100$.

ATP - 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**.

ATP - 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.

CRITÉRIOS AVALIATIVOS OU RUBRICAS DA ATP

CRITÉRIOS	DESCRITORES			
	Autônomo (100% a 90%)	Capaz (89% a 70%)	Aprendiz (69% a 40%)	Em desenvolvimento (39% a 0%)
Interpretação das especificações de algoritmos 30%	Implementa adequadamente todos os requisitos expostos em todas as etapas da atividade, traduzindo-os em algoritmos desenvolvidos na linguagem de programação Python.	Implementa adequadamente todos os requisitos expostos em pelo menos três das etapas da atividade, traduzindo-os em algoritmos desenvolvidos na linguagem de programação Python.	Implementa adequadamente todos os requisitos expostos em pelo menos duas das etapas da atividade, traduzindo-os em algoritmos desenvolvidos na linguagem de programação Python.	Implementa adequadamente todos os requisitos expostos em pelo menos uma das etapas da atividade, traduzindo-os em algoritmos desenvolvidos na linguagem de programação Python.
Usabilidade e qualidade do algoritmo 5%	O algoritmo responde adequadamente a todos os casos esperados e inesperados para todas as etapas da atividade (exemplo: aceita somente números decimais e acima de zero para o salário e mostra uma mensagem autoexplicativa de erro, recusando-se a prosseguir caso se informe um texto no lugar do salário ou um ano de nascimento futuro).	O algoritmo responde adequadamente a todos os casos esperados e inesperados em pelo menos três das etapas da atividade (exemplo: aceita somente números decimais e acima de zero para o salário e mostra uma mensagem autoexplicativa de erro, recusando-se a prosseguir caso se informe um texto no lugar do salário ou um ano de nascimento futuro).	O algoritmo responde adequadamente a todos os casos esperados e inesperados em pelo menos duas das etapas da atividade (exemplo: aceita somente números decimais e acima de zero para o salário e mostra uma mensagem autoexplicativa de erro, recusando-se a prosseguir caso se informe um texto no lugar do salário ou um ano de nascimento futuro).	O algoritmo responde adequadamente a todos os casos esperados e inesperados em pelo menos uma das etapas da atividade (exemplo: aceita somente números decimais e acima de zero para o salário e mostra uma mensagem autoexplicativa de erro, recusando-se a prosseguir caso se informe um texto no lugar do salário ou um ano de nascimento futuro).
Lógica e coerência do algoritmo 35%	O algoritmo apresenta coerência integral para a solução proposta, funcionando de forma coesa e com o resultado esperado, independentemente do nome, cargo, salário e demais dados informados pelo usuário.	O algoritmo apresenta coerência integral para a solução proposta, funcionando de forma coesa e com o resultado esperado para a maioria dos usuários, revelando falhas (bugs) a depender do nome, cargo,	O algoritmo apresenta coerência parcial para a solução proposta, funcionando de forma coesa e com o resultado esperado para a minoria dos usuários, revelando falhas (bugs) a depender do nome, cargo, salário e demais dados informados.	O algoritmo apresenta coerência parcial para a solução proposta, funcionando de forma imprecisa e com comportamento divergente do esperado pela especificação para a maioria dos nomes, cargos, salários e demais dados informados pelo usuário.

		salário e demais dados informados.		
<p>Uso de funções, estruturas de seleção e repetição</p> <p>30%</p>	<p>O algoritmo emprega adequada e totalmente, respeitando por completo as especificações da atividade, as funções (com e/ou sem parâmetros de entrada e retornos), estruturas de repetição (<i>for</i> ou <i>while</i> usando critérios de parada adequados) e estruturas de seleção (<i>if</i>, <i>elif</i> e <i>else</i>).</p>	<p>O algoritmo emprega adequada e totalmente, respeitando parcialmente as especificações da atividade, as funções (com e/ou sem parâmetros de entrada e retornos), estruturas de repetição (<i>for</i> ou <i>while</i> usando ou não critérios de parada adequados) e estruturas de seleção (<i>if</i>, sem usar <i>elif</i> e/ou <i>else</i>).</p>	<p>O algoritmo emprega parcial ou totalmente, respeitando parcialmente as especificações da atividade, as funções (com ou sem parâmetros de entrada e sem retornos), estruturas de repetição (<i>for</i> ou <i>while</i> usando ou não critérios de parada adequados) e estruturas de seleção (<i>if</i>, sem usar <i>elif</i> e/ou <i>else</i>).</p>	<p>O algoritmo emprega parcial ou totalmente, sem respeitar as especificações da atividade, as funções (com ou sem parâmetros de entrada e sem retornos), estruturas de repetição (<i>for</i> ou <i>while</i> sem usar critérios de parada adequados) e estruturas de seleção (<i>if</i>, sem usar <i>elif</i> e/ou <i>else</i>).</p>