

# Desafio SC Clouds

Esse documento consiste na resolução do desafio proposto pela empresa SC Clouds para vaga de desenvolvedor.

## Sumário

### 1. Questões Conceituais

1. [Questão 1](#)
2. [Questão 2](#)
3. [Questão 3](#)

### 2. Questões Práticas

## Questões Conceituais

Segue as respostas das perguntas conceituais propostas no desafio.

### Diferenciar as camadas 2 e 3 do modelo OSI, e indicar os protocolos utilizados para endereçamento nestas camadas.

O modelo Open Systems Interconnection (OSI) é um modelo conceitual que possibilita a comunicação entre diferentes sistemas usando protocolos padronizados. Ele é dividido em sete camadas hierárquicas. As camadas mais próximas ao topo da pilha possuem um grau de abstração maior, podendo consumir serviços providos pelas camadas abaixo delas.

A camada dois do modelo OSI é a Camada de Enlace de Dados/Data Link Layer. Ela trabalha em cima da camada física, gerenciando a transmissão e recepção de quadros, e controlando o fluxo e os erros de dados na comunicação intrarrede (na mesma rede). Uma das atribuições da camada é receber dados como pacotes da Camada de Rede (camada acima), dividi-los em quadros e envia-los bit a bit para a camada física. Essa camada é considerada uma das mais complexas do modelo OSI, visto que abstrai as complexidades de hardware para as camadas superiores.

A Data Link Layer é subdivida em duas outras sub-camadas, a Logical Link Control (LLC) e a Media Access Control (MAC). A LLC lida com multiplexação, fluxo de dados entre aplicações/serviços e fornecimento de mensagens de erros e outros avisos gerais. Já a MAC é responsável gerenciar interações entre dispositivos e endereçar quadros.

Além disso, existem diversos protocolos na Data Link Layer, como: Synchronous Data Link Protocol (SDLC), High-Level Data Link Protocol (HDLC), Point to Point Protocol (PPP), entre outros.

A camada três do modelo OSI é a Camada de Rede/The Network Layer. A sua principal responsabilidade é transportar pacotes da fonte para destino sem que eles sejam modificados ou usados durante o processo. Se os pacotes são muito grandes para serem transportados, então eles são quebrados em pacotes menores. Além disso, dentre as várias rotas disponíveis pela rede, a camada decide qual rota traçar, e os endereços da fonte/destino são adicionados nos pacotes pela camada.

Dentre os protocolos utilizados na Network Layer, destacam-se o Internet Protocol (IP) e o Internet Control Message Protocol (ICMP). O IP é responsável pelo endereçamento IP, determinar a rota que um pacote deve traçar, formatar pacotes em "datagrams" e fragmentar pacotes. O ICMP é responsável pela detecção de erros e pela reportagem deles.

### Qual a diferença entre adotar uma solução proprietária como o sistema operacional Windows quando comparado a adoção de uma solução OpenSource como o sistema operacional Ubuntu? Quais seriam os pontos negativos e positivos de cada abordagem?

Softwares Open Source são softwares cujo código é livremente disponível na internet. Desenvolvedores podem modificá-lo, adicionando novas funcionalidades de acordo com as suas demandas. Esse tipo de software possui diretrizes de contribuição, geralmente disponíveis em suas documentações. Além disso, eles geralmente possuem licenças que estipulam os direitos dos usuários. Alguns exemplos famosos de softwares open source são Ubuntu, Android e Firefox.

Softwares Proprietários são softwares cujo código não é publicamente disponível. Eles são desenvolvidos, testados e mantidos pela organização/indivíduo que possui seus direitos. As organizações proprietárias disponibilizam para os usuários do software licenças, que determinam o escopo de funcionalidades disponíveis. Como exemplo, valem ser ressaltados o Windows, macOS e Microsoft Office.

Alguns prós e contras de cada tipo de software que valem a pena serem elencados:

- Softwares Open Source provêm mais controle e adaptabilidade do código em relação a Softwares Proprietários, visto que eles podem ser examinados constantemente pela comunidade.
- Como os Softwares Proprietários não têm seus códigos disponíveis publicamente, é quase impossível que seus usuários proponham melhoras ou novas funcionalidades. A abordagem Open Source, por outro lado, estimula constantemente a comunidade a contribuir ao projeto, aumentando sua transparência em relação aos seus usuários.
- Softwares proprietários geralmente vêm acompanhados de planos de assinaturas, taxas de licença e de manutenção. Softwares open source não são sempre pagos e quando são, o custo é geralmente baixo.
- Um dos pontos mais fortes dos softwares proprietários é a questão da estabilidade. Eles tendem a ser mais estáveis que softwares de código aberto.
- Softwares proprietários são uma boa opção quando uma empresa deseja escalar um sistema rapidamente, sem se preocupar em relação a compatibilidade com tecnologias existentes.

### O que seria um projeto OpenSource? Como empresas podem adotar tais tecnologias e o que isso acarreta?

Um projeto OpenSource é um projeto que está livre para a comunidade examinar, modificar, aprimorar e documentar seu código. Geralmente, ele possui uma licença que estipula o que pode ser feito com o software, juntamente com diretrizes de contribuição.

A adoção de tecnologias open source por empresas é extremamente válida. Elas geralmente possuem um custo reduzido em relação às tecnologias proprietárias, são flexíveis, customizáveis, etc. É interessante que a empresa crie entre os seus desenvolvedores uma cultura de contribuição ativa a esses projetos. Também devem ser avaliados os riscos associados ao uso dessas tecnologias, relacionados à segurança e à questão legal. Além disso, deve ser levados em consideração a maturidade e estabilidade da tecnologia a ser adotada.

# Questões Práticas

---

## Estrutura de pastas

```
|
├─ .gitignore
├─ README.md # respostas em arquivo MD
├─ answers.pdf # respostas em arquivo PDF
├─ main.py # arquivo do programa a ser executado
├─ fibonacci
|   ├── main.py # arquivo principal FIBONACCI
|   ├── fib.py # algoritmo linear
|   └─ recursive_fib.py # algoritmo recursivo
├─ prime_numbers
|   ├── main.py # arquivo principal PRIME NUMBERS
|   ├── is_prime.py # algoritmo linear
|   └─ recursive_is_prime.py # algoritmo recursivo
|
```

## Setup

Para rodar os algoritmos localmente, certifique-se que a versão do Python seja igual ou superior à 3.10.12 e siga as próximas instruções:

```
# clone o projeto
git clone https://github.com/bernardodemarco/sc-clouds-challenge.git

# rodar o projeto
cd sc-clouds-challenge
python3 main.py
```

Ressalta-se que foi criado um menu intuitivo para navegar entre as resoluções.

## Sobre a documentação dos algoritmos

Foi criada uma breve documentação dos algoritmos usando [docstrings](#), seguindo o seguinte padrão:

```
def docs_pattern():
    """
    Brief description.

    Args:
        <parameter_name> (<parameter_type>): <parameter description>.

    Returns:
        <return_type>: <return_description>.
    """
    pass
```

Além disso, resalta-se que foram adicionadas tipagens do Python, visando melhorar a legibilidade do código.