

DSA – Formação Cientista de Dados

3. Big Data Analytics com R e Microsoft Azure Machine Learning.

3.1. Introdução

Breve História da Linguagem R:

- 1993 – Projeto de pesquisa em Auckland, na Nova Zelândia
- 1995 – R liberado como projeto open-source
- 1997 – Formado o grupo R-Core
- 2000 – Liberada a versão 1.0.0 do R
- 2003 – Criação da R Foundation
- 2004 – Primeira conferência internacional de usuários em Vienna
- 2015 – Formado o R Consortium (com participação da IBM e Microsoft)

Por que cientistas de dados usam R?

- Linguagem Open Source
- Versatilidade
 - Extração de Dados
 - Limpeza de Dados
 - Carregamento e Transformação de Dados
 - Análise Estatística
 - Modelagem Preditiva
 - Machine Learning
 - Visualização de Dados

Vantagens e Desvantagens da linguagem R

Vantagens	Desvantagens
<ul style="list-style-type: none">• Grande variedade de pacotes disponíveis• Flexibilidade e Rapidez	<ul style="list-style-type: none">• Não há interface gráfica (tudo é feito por linha de comando)• Limitações no uso de memória principalmente com datasets muito

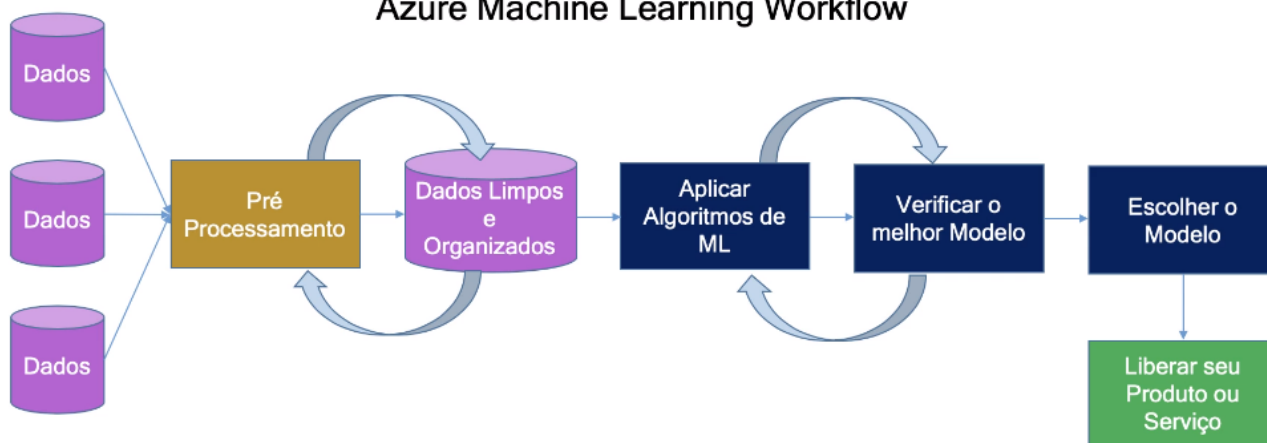
- Análise Estatística

grandes

Azure Machine Learning Workflow



Azure Machine Learning Workflow



Legenda: DADOS > PRÉ-PROCESSAMENTO <> DADOS LIMPOS E ORGANIZADOS > APLICAR ALGORITMOS DE ML <> VERIFICAR O MELHOR MODELO > ESCOLHER O MODELO > LIBERAR SEU PRODUTO OU SERVIÇO

Dados X Informação X Conhecimento X Inteligência

As pessoas trabalham com informações, mas a tecnologia armazena dados.

Inteligência – capacidade de resolver problemas, usando conhecimento, através das informações disponíveis.

Big Data Analytics

Extrair conhecimento a partir dos dados (muitos dados).

Machine Learning

Ensinar algoritmos a usar inteligência, ou seja, usar o conhecimento para resolver problemas.

Isso é feito coletando-se dados, pré processando esses dados e transformando-os em informação, alimentando-se algoritmos (conjunto de operações estatísticas e matemáticas) que aprendem a relação dessa informação, adquirindo conhecimento. Dessa forma, a partir do conhecimento adquirido, é possível resolver novos problemas de negócio a partir de novos dados.

A importância do Big Data Analytics

Tempestade Perfeita

- Crescimento exponencial do volume de dados
- Preço baixo do armazenamento de dados
- Alta capacidade de processamento dos computadores

Os dados são combustível para a Análise Preditiva (essência do que é feito em Data Science).

E podemos ver a Análise Preditiva em ação todos os dias:

- Filtros de Spam dos e-mails
- Programas de concessão de financiamento para casa própria
- Padrões de reconhecimento (voz, imagem)
- Seguro de Vida
- Detecção de Fraudes em Cartões de Crédito
- Controle de Voos
- Engines de Busca na Internet

Tudo isso é alimentado por dados, muitos dados (Big Data).

Quando aplicamos Data Science para análise de Big Data temos o que é chamado de Big Data Analytics e sua importância é cada vez maior no mundo atual.

Data Science e a Evolução dos Sistemas Analíticos



Legenda: Analysis, Structure, Algorithm, Process, Programming, Solving, Knowledge.

Estamos vivendo a era da explosão dos dados

bit byte > kilobyte > megabyte gigabyte terabyte > petabyte exabyte zettabyte yottabyte

Em 2014 a humanidade acumulou em dados o equivalente a 1.8 Zettabyte.

Tipos de Análise Realizados em Ciência de Dados:

- Descritiva: O que aconteceu? (BI Tradicional)
- Diagnóstica: Por que isso aconteceu? (BI Tradicional)
- Preditiva: O que acontecerá?
- Prescritiva: O que deve ser feito?

Data Science e Inteligência Artificial – Há Diferença?

Inteligência Artificial é um conjunto de técnicas para reproduzir um comportamento inteligente. Dentre essas técnicas, se encontra o Machine Learning que, dentre várias, possui a técnica de Deep Learning.

Data Science é uma área multidisciplinar que envolve Estatística, Matemática, Machine Learning, Programação, Banco de Dados, Conhecimento de Negócios, Técnicas de Apresentação e Visualização de Dados.

Aplicamos Data Science para analisar dados, construir modelos preditivos e resolver problemas de negócio. Ou seja, aplicamos Data Science quando estamos trabalhando com Machine Learning e Inteligência Artificial.

Instalando o R, RStudio e RTools no Windows 10

Dicas:

- Não utilize nome de usuário, no seu Sistema Operacional, com espaço ou acento.
- Instale os softwares em inglês (previna possíveis bugs)

Para fazer o download do R (interpretador):

- <https://cran.r-project.org>

- Clique em “Download R for Windows”
- Clique em “base”
- Obs.: no curso foi utilizado a versão “Download R 3.5.2 for Windows”
- Instale num diretório onde não haja espaços no nome:
 - Ex.: C:\R\R-3.5.2
- Selecione todos os itens/componentes
 - Core Files, 32-bit Files, 64-bit Files e Message translations
- Em “Startup options”, selecione “No (accept defaults)”
- Manter “R” no atalho que será colocado no menu iniciar
- Em “Select Additional Tasks”, selecione “Save version number in registry” e “Associate R with .Rdata files”
- Prosiga com a instalação

Para fazer o download do RTools (exclusivo para usuários windows):

- <https://cran.r-project.org>
- Clique em “Download R for Windows”
- Clique em “Rtools”
- Obs.: no curso foi utilizado a versão “Rtools35.exe”
- Instale num diretório onde não haja espaços no nome:
 - Ex.: C:\Rtools
- Em “Select Components”, mantenha as opções recomendadas
- Em “Select Additional Tasks”, selecione “Add rtools to system PATH” e “Save version information to registry”
- Prosiga com a instalação
- Após instalação, caso necessite adicionar o PATH em seguida, crie um arquivo de nome .Renviron, na pasta Documents:
 - Abra o R e digite a linha:
 - `writeLines('PATH="${RTOOLS40_HOME}\\usr\\bin;${D:/Desenvolvedor/CienciaDeDados/Dev/Rtools/rtools40}', con = "~/.Renviron")`
 - Reinicie o R e confirme se deu certo, digitando a linha:
 - `Sys.which("make")`
 - Deverá aparecer o caminho da instalação do Rtools

Para fazer o download do RStudio:

- <https://www.rstudio.com>
- Na parte superior, clique em “Products”, depois em “RStudio”
- Clique na versão “Desktop” (pois há também a “Server”)
- Será mostrado a versão Free e a paga. Clique no botão para fazer o download da versão Free
- Outra vez, procure o botão para opção Free
- Em “Installers for Supported Platforms”, selecione a versão para instalação
- Obs.: no curso foi utilizado a versão “RStudio 1.1.463 Windows Vista/7/8/10”
- Instale num diretório onde não haja espaços no nome:
 - Ex.: C:\RStudio
- prossiga com a instalação

3.2. Fundamentos da Linguagem R

Introdução

Site Oficial: <https://www.r-project.org>

A linguagem R é um ambiente de software gratuito para computação estatística e gráficos.

RStudio

O RStudio é uma IDE para linguagem R.

- Abra um painel para escrever seus scripts em R:
 - New File > R Script
- Altere a organização dos painéis:
 - Tools > Global Options > Pane Layout
 - No primeiro quadrante, selecione Console
 - No segundo, selecione Source (onde se escreve os scripts)
 - Mantenha o 3 e o 4, ou altere como quiser
 - Apply > OK
- Mude o interpretador dos scripts (testar outras versões), sem a necessidade de excluir a versão anterior. Basta:
 - Tools > Global Options > General
 - em R version, selecione a pasta da nova versão e pronto
 - Apply > OK

Definindo Diretório de Trabalho

- Antes de começar, defina um diretório para seus projetos em R. Lembre-se de criar as pastas sem espaços em branco ou acentos.
- No RStudio, aponte para a nova pasta criada (sempre que for realizar um novo projeto):
 - Aba Session > Set Working Directory > Choose Directory
 - Selecione a pasta e clique em Open
 - No console, aparecerá o comando setwd (“caminho da pasta do projeto”)
- Abra um novo script R em New File e salve-o (já no novo diretório)

3.3. Linguagem R – Fatores, Estruturas de Controle e Funções

Variáveis Qualitativas e Quantitativas

Variável é a característica de interesse que é medida em cada elemento da amostra ou população. Como o nome diz, seus valores variam de elemento para elemento. As variáveis podem ter valores numéricos ou não numéricos.

Variáveis			
	Idade	Peso	Sexo
Pessoa 1	41	62	Masculino
Pessoa 2	37	78	Feminino

- Variáveis Qualitativas (Categóricas): representam uma classificação dos indivíduos
 - Nominais: profissão, sexo, religião
 - Ordinais: escolaridade, classe social, fila
- Variáveis Quantitativas (Numéricas):
 - Discretas (limite finito): número de filhos, número de carros, número de acessos
 - Contínuas (qualquer tipo de valor): altura, peso, salário

Uma das principais diferenças entre uma variável categórica e uma variável quantitativa é que uma variável categórica pode pertencer a um número limitado de categorias.

Entender o tipo de variável é importante porque determina o tipo de técnica de análise a ser utilizada na resolução de um problema.

As distinções são menos rígidas do que essas descrições.

Uma variável originalmente quantitativa pode ser coletada de forma qualitativa. Por exemplo, a variável idade, medida em anos completos, é quantitativa (contínua); mas, se for informada apenas a faixa etária (0 a 5 anos, 6 a 10 anos, etc.), é qualitativa (ordinal).

Outro ponto importante é que nem sempre uma variável representada por números é quantitativa. Por exemplo, o número do telefone de uma pessoa, da casa ou de sua identidade; e sexo, às vezes, é registrado na planilha de dados como 1 se masculino e 2 se feminino.

Lembre-se:

Você precisa conhecer os dados que tem em mãos, para poder trabalhar sua análise e selecionar as técnicas adequadas.

Fatores e Fatores Ordenados

Fatores

O termo fator se refere a tipos de dados estatísticos usados para armazenar variáveis categóricas. Fatores são variáveis categóricas que são muito úteis em sumarização de estatísticas, plots e regressões.

Fatores representam uma maneira muito eficiente para armazenar valores de caracteres, porque cada caractere único é armazenado apenas uma vez e os dados são armazenados como um vetor de inteiros.

É como se a linguagem R transformasse uma palavra em um vetor de números inteiros, para que se ganhe em performance na hora de fazer a execução de um processo de análise. É uma maneira de otimizar o armazenamento e o processamento de variáveis categóricas.

Para criar fatores usamos a função `factor()`.

Fatores Ordenados

São fatores pelos quais preserva-se a ordenação natural dos níveis das variáveis. A ordenação segue a ordem alfabética, ao menos que outra ordenação diferente seja definida pelo usuário.

Para criar fatores ordenados usamos a função `factor(..., ord=T)` ou `ordered()`.

Estruturas de Controle

Estruturas de controle permitem que se façam validações e se repitam um bloco de código, um número n de vezes, em que se realize mudanças no comportamento do script, de acordo com determinadas regras.

Condicionais If-Else

- `if(condição){conjunto de tarefas}`
 - `else if(condição){conjunto de tarefas}`
 - `else {outro conjunto de tarefas}`
- `ifelse(condição, tarefa1, tarefa2)`
- `ifelse(condição, tarefa1,`
 - `ifelse(condição, tarefa1, tarefa2))`

Loop For

- `for(i in 1:N){conjunto de tarefas}`

Loop While

- `while(condição satisfeita){conjunto de tarefas}`

Repetições

- `rep(x, y)` - “rep(repita x, y vezes)”
- `repeat {}`

Funções

Funções são objetos, dentro da linguagem R, que evitam repetição de código. Deixam o código mais legível, menos repetitivo e mais elegante.

Tudo o que você atribui com “<-” vira um objeto no R.

Sintaxe Padrão:

```
nome_da_função(parâmetros){código a ser executado}
```

Sintaxe Sem Número Fixo de Parâmetros:

```
nome_da_função(...){código a ser executado}
```

Sintaxe Sem Parâmetros:

```
nome_da_função(){código a ser executado}
```

Funções Anônimas (bom para quando o código for usado uma única vez):

```
teste_func <- sapply(c(1:10), function(x) {x %% 2 == 0})
```

Precisamos ficar atentos ao escopo de uma Função!

Ao criar uma função, tudo que está dentro da função tem um escopo local, ou seja, só existe dentro da função.

Criando Funções:

```
function(argumentos){corpo da função}
```

```
nome_da_funcao <- function(argumentos){corpo da função}
```

Funções Built-in:

São funções internas da linguagem R.

- `c()` - vetor
- `matrix()` - matrizes
- `seq()` - sequencia de números
- `help()` - ajuda da linguagem
- `contributors()` - lista os contribuidores da linguagem R

Família Apply – Uma forma elegante de se fazer loops em R

`apply()` - arrays e matrizes

`tapply()` - os vetores podem ser divididos em diferentes subsets

lapply() - vetores e listas

sapply() - versão amigável da lapply

vapply() - similar a sapply, com valor de retorno modificado

rapply() - similar a lapply()

eapply() - gera uma lista

mapply() - similar a sapply, multivariada

by

Se você estiver trabalhando com os objetos:

list, numeric, character (list/vecor) => sapply ou lapply

matrix, data.frame (agregação por coluna) => by / tapply

Operações por linha ou operações específicas => apply

Expressões Regulares em R

São um conjunto de caracteres, que formam um padrão que permite fazer buscas em strings.

Tabela de Expressões Regulares:

- `\\d` - Dígitos, 0,1,2 ... 9
- `\\D` - Não dígito
- `\\s` - Espaço
- `\\S` - Não Espaço
- `\\w` - Palavra
- `\\W` - Não Palavra
- `\\t` - Tab
- `\\n` - Nova Linha
- `^` - Começo da String
- `$` - Fim da String
- `\\` - Caracteres especiais, e.g. `\\` is `"\"`, `\\+` is `"+"`
- `|` - Alternation match. e.g. `/(e|d)n/` matches `"en"` and `"dn"`
- `.` - Any character, except `\\n` or line terminator
- `[ab]` - a or b
- `[^ab]` - Any character except a and b
- `[0-9]` - All Digit
- `[A-Z]` - All uppercase A to Z letters
- `[a-z]` - All lowercase a to z letters

- [A-z] - All Uppercase and lowercase a to z letters
- i+ - i at least one time
- i* - i zero or more times
- i? - i zero or 1 time
- i{n} - i occurs n times in sequence
- i{n1,n2} - i occurs n1 - n2 times in sequence
- i{n1,n2}? - non greedy match, see above example
- i{n,} - i occurs >= n times
- [:alnum:] - Alphanumeric characters:
- [:alpha:] - and [:digit:] [:alpha:] Alphabetic characters: [:lower:] and [:upper:]
- [:blank:] - Espaços em branco: e.g. space, tab
- [:cntrl:] - Control characters
- [:digit:] - Dígitos: 0 1 2 3 4 5 6 7 8 9
- [:graph:] - Graphical characters: [:alnum:] and [:punct:]
- [:lower:] - Lower-case letters in the current locale
- [:print:] - Printable characters: [:alnum:], [:punct:] and space
- [:punct:] - Punctuation character: ! " # \$ % & ' () * + , - . / : ; < = > ? @ [\] ^ _ ` { | } ~
- [:space:] - Space characters: tab, newline, vertical tab, form feed, carriage return, space
- [:upper:] - Upper-case letters in the current locale
- [:xdigit:] - Hexadecimal digits: 0 1 2 3 4 5 6 7 8 9 A B C D E F a b c d e f

3.4. Linguagem R – Gráficos

Introdução

Os gráficos fazem parte de todo o processo de Data Science, desde a análise exploratória até a apresentação do resultado final. É um equívoco achar que os gráficos servem apenas como produto do resultado final. Utilizá-los durante todo o processo, pode permitir insights, visualizar padrões, etc.

O que é visualização de Dados?

Visualização de dados (DataViz) é a representação de dados em formato gráfico.

Gráficos, Tabelas e Estatísticas tornam a compreensão dos dados muito mais fácil.

O que são gráficos?

O gráfico é uma representação com forma geométrica, construída de maneira exata e precisa, a partir de informações numéricas, obtidas através de pesquisas e organizadas em uma tabela.

Como a Linguagem R Trata as Visualizações de Dados?

Existe um consenso na comunidade de Data Science que a linguagem R fornece algumas das melhores ferramentas para geração de gráficos.

Os gráficos podem ser replicados, modificados e publicados com apenas algumas poucas linhas de código.

Pacote Básico de Plotagem (Base Plotting System)

Ele é formado por 2 pacotes:

- graphics – contém as funções gráficas básicas, incluindo plot, hist e boxplot.
- GrDevices – contém as implementações de dispositivos gráficos como X11, pdf, PostScript, png, etc.

Os plots (gráficos bem simples) são objetos construídos através de funções e com atributos.

Podemos criar gráficos bem mais completos e profissionais usando pacotes como ggplot2, lattice e outros pacotes R.

Usando o Base Plotting System é possível criar gráficos do tipo:

- Colunas
- Barra
- Linha
- Dispersão
- Área
- Bolhas
- Superfície

- Cone
- Pizza

É importante atentar para os tipos de pontos, o peso e tamanho desses pontos, tipos de linhas e cores utilizados nos gráficos, pois impacta em como as pessoas vêm e interpretam as informações dos gráficos.

Gramática dos Gráficos

Os gráficos são construídos em camadas e cada uma delas adiciona um elemento visual.

Essas camadas são:

- os dados – o conjunto de dados a ser analisado
- a estética – a escala em que mapeamos os dados
- a geometria – os elementos visuais usados para representar os dados
- *facets* – visualização do gráfico em porções menores (é quando se utiliza uma única área de visualização para apresentar vários gráficos de maneira simultânea)
- a estatística – representação e análise dos dados
- as coordenadas – a área na qual o gráfico será construído
- os temas – visual geral do gráfico

Quando os elementos se juntam, acontece o que chamamos visualização dos dados.

A gramática dos gráficos descreve como os elementos devem ser combinados para formar uma visualização.

Scatterplot – Gráfico de Dispersão (base plotting system)

Mostra a relação entre duas variáveis, uma independente e outra dependente. Ex.: preço imóvel/nº quartos.

Boxplot – Box-and-whiskey (base plotting system)

Esse gráfico fornece as seguintes informações (por padrão):

- Mediana (linha dentro do retângulo)
- Valores mínimos e máximos da variável (linhas foras do retângulo)
- Quartis (retângulo)

Histogram – Histograma (base plotting system)

Usado para visualizar a distribuição de frequência de uma variável.

O histograma NÃO é um gráfico de colunas; e suas caixinhas são chamadas de bins.

Barplot – Gráfico de Barras (base plotting system)

Um dos gráficos mais utilizados, porque é simples e objetivo, transmitindo a informação de maneira direta.

Pie chart – Gráfico de Pizza (base plotting system)

Amado por usuários comuns, odiado por cientistas de dados. Isso porque existem gráficos melhores para se utilizar, como o gráfico de barras.

Quanto menos fatias, mais simples de se interpretar. Quanto mais fatias, o gráfico acaba se tornando muito poluído e complexo de se interpretar.

Explorando ggplot2

É um sistema completo, alternativo ao sistema básico de gráficos do R. Oferece mais opções de modificação, legendas prontas e formatação mais sólida.

Acesse o endereço abaixo para fazer o download da folha de referência do ggplot2 em português:

<https://rstudio.com/wp-content/uploads/2016/03/ggplot2-cheatsheet-portuguese.pdf>

No link abaixo, encontra-se o guia completo:

<https://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf>

Lattice

O pacote Lattice é um sistema de visualização de dados de alto nível, poderoso e elegante, com ênfase em dados multivalorados (por isso, é uma excelente ferramenta para pesquisadores que necessitam realizar análise multivalorada).

Não é tão elegante como o ggplot2 e nem tão simples como o base plotting system, mas é um intermediário entre os dois, sendo muito útil durante o processo de análise de dados.

Na criação de gráficos, condições e agrupamentos são 2 conceitos importantes, que permitem compreender mais facilmente os dados que se tem em mãos. O conceito por trás do Lattice é agrupar os dados e criar visualizações de forma que fique mais fácil a busca por padrões.

Mapas

as

4. Python Fundamentos para Análise de Dados (gratuito)

5. Big Data Real-Time Analytics com Python e Spark

6. Engenharia de Dados com Hadoop e Spark

7. Machine Learning

8. Business Analytics

9. Visualização de Dados e Design de Dashboards

10. Preparação para Carreira de Cientista de Dados

Capítulo 6 – Treinamento

O que é um Plano de Treinamento?

Um cronograma organizado que descreve os horários e objetivos do treinamento.

É uma forma eficaz de navegar pelo conteúdo de forma organizada.

Por Que eu preciso de um Plano de treinamento?

Determine por que você está estudando e trace seus objetivos.

Crie o seu Plano de Treinamento em 4 Etapas:

Etapa 1: Crie uma planilha de controle para suas atividades atuais.

Etapa 2: Atualize e desenvolva a sua planilha de controle.

Utilize ferramentas de auxílio, como Google Calendar; defina, nas horas disponíveis em sua agenda, a carga horária e o conteúdo a ser estudado.

Etapa 3: Determine os objetivos do seu estudo.

No início de cada semana determine por que você precisa estudar e o que você planeja realizar em cada curso.

Etapa 4: Use sua agenda.

Um plano de treinamento funciona melhor se for seguido de maneira consistente. O mais importante é que ele seja aderido, ou seja, que você utilize-o, atualize-o, etc.

Gerencie o Seu Tempo

Quanto tempo devo me dedicar aos Estudos?

Depende da quantidade de horas/aulas do seu curso na plataforma, seus hábitos de estudo e objetivos profissionais.

Sugere-se 2 horas de estudo para cada 1 hora/aula de curso.

Posso estudar exaustivamente?

Não é produtivo.

Longas sessões, normalmente, se tornam enfadonhas e quando sua mente começar a vagar, esse tempo de estudo será desperdiçado.

Como gerenciar o seu tempo efetivamente?

Equilibre sua vida pessoal, profissional e seu treinamento on-line, para ser bem-sucedido a longo prazo. Negligenciar áreas só funcionará por algum tempo.

Dica 1: Faça um cronograma de estudo Detalhado, usando sua planilha ou app para smartphone.

Dica 2: Priorize suas tarefas e atribuições, para dedicar mais tempo a assuntos que são de maior importância ou novos para você.

Dica 3: Supere as tentações, antecipando distrações, corrigindo seus erros e conhecendo suas fraquezas. Também, recompense-se com atividades divertidas após conclusão dos seus estudos.

Dica 4: Escolha um ambiente de estudo produtivo, onde você possa se concentrar mais e maximizar seu tempo.

Dica 5: Mantenha-se organizado, para não perder tempo ao fazer treinamentos on-line. Ordene seus estudos de acordo com conteúdos, organize seus arquivos utilizando pastas separadas, etc. Também, ao abrir um módulo on-line, certifique-se de ter em mãos tudo o que precisa, para que não haja interrupções.

Faça anotações

Facilita o entendimento do conteúdo, pois estimula-o a ouvir o conteúdo das aulas on-line com mais atenção e a colocar conceitos dentro de seus próprios termos.

Como fazer anotações efetivas?

Tomar notas diretamente do que o professor está ensinando pode ajudá-lo a reduzir a vasta quantidade de material para conceitos-chave que você precisará mais tarde.

Dica 1: Esteja preparado para a aula.

Dica 2: Seja um bom ouvinte, prestando bastante atenção ao que se está sendo dito, anotando conceitos-chave e dúvidas a serem sanadas junto a professores e fóruns de discussão.

Dica 3: Saiba tomar nota, de uma maneira que ela seja útil, posteriormente.

Dica 4: Compare suas notas com a de outros alunos.

Dica 5: Edite e organize suas notas, além de ter cuidados como realização de backups.

Dica 6: Compartilhe suas notas, através da comunidade.

Controle o seu stress

O stress é natural. Existe um tipo positivo de stress, o eustress, que pode servir como motivação para se continuar trabalhando. Mas há o tipo de stress preocupante, que causa problemas e pode afetar seus estudos, o distress.

Reconheça os sintomas do stress

Conheça a diferença entre estresse normal e prejudicial.

Dores de cabeça recorrente, fadiga, insônia, dificuldades em descansar e indigestão, são sintomas de estresse agudo.

Já o estresse crônico, pode se mostrar com o ranger dos dentes, o esquecimento, o consumo excessivo de álcool, confusão e outros sintomas que podem parecer o estado natural das coisas.

O estresse pode enfraquecer o sistema imunológico, causar dores musculares, hipertensão e arritmia cardíaca.

O que fazer nos momentos de stress?

1. Respire fundo.
2. Faça algo reconfortante.
3. Faça pausas regulares.

Como lidar com o stress?

Fique atento aos sinais de stress e estressores.

Divida suas tarefas.

Evite procrastinar.

Não tenha medo de dizer não.

Cuide da sua saúde.

Exercite sua memória

A memória de curto prazo age no momento em que uma informação está sendo adquirida e fica armazenada por um período curto de tempo ou até ser decidido se essa informação será mantida ou descartada.

Já a memória de longo prazo armazena as informações que foram declaradas como definitivas. Para isso, é necessário que uma informação de curto prazo seja transformada em informação de longo prazo.

Como exercitar sua memória?

Divida as informações em partes menores. Sua memória de curto prazo pode armazenar entre 4 e 7 coisas separadas ao mesmo tempo.

Pratique jogos de concentração, para aliviar stress e treinar sua memória ao mesmo tempo.

Estimule a sua memória ao longo do dia, passando um tempo lendo um livro, parando para sentir o cheiro das flores, experimentando novos sabores ou fazendo uma interação social com outros.

Mantenha-se em equilíbrio e controle seus níveis de stress.

Pratique a escuta ativa durante a aula.

Estude suas anotações e revise-as.

Estude diariamente e várias vezes.

Mantenha seu cérebro alimentado