

# Arthur Bottacin Cambler



Mestrando em Botânica pelo Instituto de Biociências da USP

Título do projeto: Adaptação do método de fracionamento de parede celular de cana-de-açúcar para análise de um grande número de amostras

Orientador: Marcos Silveira Buckeridge

---

## Meus Exercícios

Resolução dos exercícios: [Exercícios](#)

---

## Trabalho Final

### Plano A: Preenchendo a ficha de "Dungeons and Dragons"

#### Contextualização:

Para jogar "Dungeons and Dragons", um RPG de mesa, é importante utilizar uma [ficha de personagem](#) para organizar as informações do seu personagem. Entretanto, preencher uma ficha de personagem não é algo trivial, sendo necessária a escolha da raça do seu personagem, de sua classe, o sorteio e distribuição dos seus atributos, a descrição de suas características e a escolha dos equipamentos e magias ([Regras para criar personagens](#)). Todas estas escolhas e sorteios vão afetar diferentes aspectos do seu personagem, o que pode tornar o preenchimento da ficha uma tarefa bastante confusa e demorada. Por esse motivo, a minha função pretende auxiliar a montagem da ficha de personagem de "Dungeons and Dragons", permitindo que o jogador escolha os diferentes parâmetros para o seu personagem de nível 1 e retornando os valores calculados e dados necessários para preencher a ficha. Em outras palavras, minha função recebe os argumentos definidos pelo usuário e utiliza-os para calcular, sortear ou escolher diferentes parâmetros (numéricos ou textuais) retornando para o usuário uma lista com diferentes tipos de informação

#### Planejamento da função

**Entrada:** fichaDnD (atrib, ord, alinh, rac, clas, antec)

atrib = Escolha do modo como os atributos serão escolhidos (vetor composto por caracteres com dois possíveis argumentos)

`ord` = Ordem dos atributos, do que terá o maior valor para o menor em sequência (vetor composto por 6 caracteres em uma ordem específica determinada pelo usuário)

`alinh` = Escolha o seu alinhamento (vetor composto por caracteres escolhendo um frente a 9 possibilidades)

`rac` = Escolha da raça do personagem (vetor composto por caracteres escolhendo um frente a 9 possibilidades)

`clas` = Escolha da classe do personagem (vetor composto por caracteres escolhendo um frente a 12 possibilidades)

`antec` = Escolha dos antecedentes do personagem (vetor composto por caracteres escolhendo um frente a 13 possibilidades)

### Verificando os argumentos da função:

Para `atrib`, `alinh`, `rac`, `clas` e `antec`: é um vetor de caracteres com um único elemento? Se não, retornar: "atrib/alinh/rac/clas/antec deve conter apenas caracteres e um único elemento"

Para `atrib`, `alinh`, `rac`, `clas` e `antec`: contém ao menos um dos argumentos possíveis? Se não, retornar: "atrib/alinh/rac/clas/antec tem X possíveis argumentos, (lista dos argumentos), escolha um deles"

`ord` é um vetor de caracteres com 6 elementos? Se não, retornar: "ord deve conter apenas caracteres e 6 elementos, "For", "Des", "Con", "Int", "Sab", "Car", do maior para o menor"

`ord` contém os elementos corretos? Se não, retornar: "ord deve conter 6 elementos denominados "For", "Des", "Con", "Int", "Sab", "Car""

### Pseudo-código:

1- Criar os objetos que vão compor a saída:

Data frame *atributos*

Data frame *dados.gerais*

Data frame *teste.resist*

Data frame *habilidades.pro*

Data frame *habilidades.tot*

Vetor *magias*

Data frame *especiais*

Data frame *caracteristicas*

Data frame *equipamentos*

Data frame *proficiencias*

Data frame *info*

2- A partir do argumento `atrib` fazer:

- Sorteio dos valores (uso de um **for** para colocar cada valor sorteado em uma coluna)

OU

- Usar valores fixos pré-determinados (livro do jogador)

3- Com os valores definidos, ordená-los conforme o argumento `ord` e guardar no objeto *atributos*.

Alterações no objeto *dados.gerais*

4, 5, 6 e 7- Usar de **ifs** encadeados ou da função **switch** para os argumentos `rac`, `clas`, `alinh` e `antec` permitindo que cada um deles execute uma tarefa diferente.

- `rac` vai gerar alterações nos objetos *atributos*, *dados.gerais* e *info*.

- `clas` vai alterar os objetos *especiais*, *dados.gerais*, *proficiencias*, *equipamentos*, *teste.resist* e *magias*.

- `a_linh` vai alterar o objeto *características*
- `antec` vai alterar o objeto *características*. Para isso é necessário sortear as características usando um **for** e checar a compatibilidade com o argumento `a_linh` usando de **ifs**.

8- De resto serão feitos 4 tipos de procedimento para os diferentes objetos de modo a obter os números e textos necessários para o preenchimento da ficha de personagem:

- Acrescentar valores pré-definidos (livro do jogador) de acordo com os argumentos escolhido
- Seleção e cópia de números entre data frames (por indexação)
- Uso de valores contidos em data frames para cálculos diversos em outros data frames (por indexação)
- Operações matemáticas básicas

9- Retornar ao usuário texto informando sobre alguns cálculos que não serão feitos e como ele deve fazê-lo (função **message**)

### Saída:

Lista juntando todos os objetos criados:

Data frame *atributos* (2 linhas e 6 colunas)

Data frame *dados.gerais* (2 linhas e 6 colunas)

Data frame *teste.resist* (1 linhas e 6 colunas)

Vetor *habilidades.pro*

Data frames *habilidades.tot* (1 linha e 18 colunas)

Vetor *magias*

Data frame *especiais* (1 linha, número variável de colunas)

Data frame *características* (1 linha e 5 colunas)

Data frame *equipamentos* (4 linhas e 4 colunas)

Data frame *proficiencias* (3 linhas e 4 colunas)

Data frame *info* (5 linhas e 2 colunas)

[Versão mais detalhada da função](#)

---

## Plano B: Escolha de disciplinas da pós graduação

### Contextualização:

A busca pelas disciplinas para cursar durante o semestre no sistema [Janus](#) demanda a atenção para diferentes aspectos. Além do interesse pelo curso é necessário verificar diversos fatores como o número de créditos oferecidos, os dias da semana em que as disciplinas vão ocorrer, o período do dia em que as aulas serão ministradas e em qual instituto ou faculdade ocorrerão as aulas. Apesar de importante para organizar seu semestre a obtenção destes dados pode ser bastante tediosa e demorada, sendo necessário acessar cada disciplina de interesse individualmente para poder acessar estas informações. Minha função pretende evitar essa busca manual pelas informações das disciplinas, permitindo que o aluno determine parâmetros que retornem somente as disciplinas oferecidas para aquele semestre que condigam com suas necessidades. Em outras palavras, os argumentos da função permitirão escolher os parâmetros de busca e a função retornará um data frame contendo os resultados (disciplinas) compatíveis e as informações correspondentes a esses resultados.

## Planejamento da função

**Entrada:** disciplinas.ofeRecidas (dia, inic, fim, cred, loc, per)

dia = dia da semana a ser escolhido (vetor composto por caracteres com 7 possibilidades)

inic = mês de início da disciplina (vetor composto por números inteiros onde  $0 < inic < 13$ )

fim = mês de término da disciplina (vetor composto por números inteiros onde  $0 < fim < 13$ )

cred = número de créditos oferecidos (vetor composto por números inteiros onde  $0 < cred < 11$ )

loc = Instituto ou faculdade em que serão procuradas as disciplinas (vetor composto por caracteres)

per = Período do dia em que a disciplina será ministrada (vetor composto por caracteres com 5 possibilidades)

### Verificando os argumentos da função:

Para dia, loc e per: é um vetor de caracteres com um único argumento? Se não, retornar:

"dia/loc/per deve conter apenas caracteres e um único elemento"

Para dia, loc e per: contém ao menos um dos argumentos permitidos? Se não, retornar:

"dia/loc/per tem X possíveis argumentos, (lista dos argumentos), escolha um deles"

Para inic e fim: é um vetor de números inteiros com um único argumento maior do que zero e

menor do que 13? Se não, retornar: "inic/fim deve conter apenas um número inteiro maior do que 0 e menor do que 13"

inic é menor do que fim? Se não, retornar: "inic precisa ser menor do que fim"

cred é um vetor de números inteiros com um único argumento maior do que zero e menor do que

11? Se não, retornar: "cred deve conter apenas um número inteiro maior do que 0 e menor do que 11"

### Pseudo-código:

1- Criar o objeto que vai compor a saída: Data frame *disciplinas*

2- Ler as informações da página "Disciplinas oferecidas" do sistema Janus a partir do URL

3- De acordo com o argumento loc, acessar a página do local escolhido pelo URL

- Acessar e guardar os dados referentes a cada disciplina do local escolhido no objeto disciplinas

4- De acordo com os argumentos de dia, inic, fim, cred e per, buscar em *disciplinas* apenas as disciplinas compatíveis com as exigências dos argumentos

- Caso existam disciplinas que correspondam a todos os argumentos pedidos retornar o data frame *disciplinas* com a lista das disciplinas selecionadas e as informações sobre as mesmas

- Caso não existam disciplinas que correspondam a todos os argumentos pedidos retornar

"Infelizmente não foram encontradas disciplinas com as características desejadas"

### Saída:

Data frame *disciplinas* com as informações das disciplinas filtradas pela função

### [Versão mais detalhada da função](#)

Olá Arthur, Você postou duas propostas detalhadas e com os respectivos pseudocódigos. Ambas são de uso para um amplo público, cumprem uma tarefa aplicável recorrentemente além de fornecer ao usuário varias opções de variações na tarefa. No caso da proposta A, você propõe e descreve bem

sua função que resultaria em um objeto de saída que considero ira cumprir o seu objetivo de facilitar preencher a ficha de "Dungeons and Dragons". De forma similar a porposta B, de escolha de disciplinas da pós graduação, explica o processo para cumprir o seu objetivo. Considero que a proposta A é criativa e também um pouco mais desafiante do que a proposta B. Além disso, supeito que o tema de RPG será muito empolgante para você e isso pode te estimular a fazer um ótima função. Por isso sugiro continuar desenvolvendo a função da proposta A. Bom trabalho!

. — [SolimaryGarcia](#) 2018/05/10 20:40

Olá Solimary,

Está certo, farei a proposta A. E sem dúvida, a função estar relacionada a RPG traz uma motivação extra para fazê-la! 😊  
Obrigado pelo retorno!

. — [Arthur](#) 2018/05/11 11:56

---

## Função finalizada

Link para a função finalizada: [fichaDnD](#)

Link para o help da função: [Help](#)

Arquivo com a função: [fichaDnD](#)

From:

<http://ecor.ib.usp.br/> - **ecoR**

Permanent link:

[http://ecor.ib.usp.br/doku.php?id=05\\_curso\\_antigo:r2018:alunos:trabalho\\_final:arthur.cambler:start](http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2018:alunos:trabalho_final:arthur.cambler:start) 

Last update: **2020/08/12 06:04**