

**Desafio semanal Brasil System**

Exemplo de fluxo de execução de uma tarefa:

i. Ler a descrição da tarefa. Crie uma branch do repositório com o nome de develop ii.Fazer o pull da master: git pull

iii.Criar um novo branch à partir do develop usando o padrão: git branch -b tarefa-001

v.Resolveu a tarefa:

● git add . (ou git add <nome\_do\_arquivo\_alterado>)

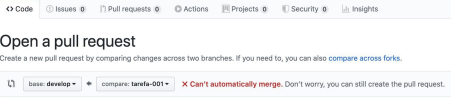
obs.: Commitar apenas o que foi alterado.

● git commit -m "Concluo parte da alteração na tela de cadastro". obs.: Toda mensagem commit deve ser afirmativa, conclui, criei, adicionei..

**obs2: Faça quantos commits forem necessários.**

● git push origin tarefa-001

● Faça o pull request na interface do Github da branch que você está, no caso tarefa-001 para develop **obs: Nunca para master** . Como fazer? Você deve navegar até o seu repositório e pressionar o botão “ New pull request ” no lado esquerdo da página.

● O GitHub vai lhe alertar de que é possível mesclar as duas branches porque não há código concorrente. Você deve adicionar um título, um comentário e, em seguida, pressionar o botão “Create pull request”. **(Neste ponto, o scrum master decidirá se aceitam ou não o seu pull request. Eles podem solicitar que você edite ou revise seu código antes de aceitar o pull request, mas não vai ser nosso caso).**

● Se à tarefa foi testado e foi dado “ok” da demanda aí então você enviará para o master.

● Quando for pegar outra demanda sempre lembra de fazer à atualização: git pull origin develop e criar a branch à partir da mesma

**Tarefa-001:** Crie um algoritmo que receba o nome como entrada e nós dê como saída o nome digitado e o comprimento do mesmo

ex: Seu nome: **Tiago**

Comprimento do nome: **5 caracteres**

**tarefa-002:** Escreva um algoritmo que aceite uma string e retorne os caracteres na ordem inversa.

ex: Seu nome: **Tiago**

Inverso do nome: **ogaiT**

**tarefa-003:** Receba uma string e imprima metade da mesma.

ex: Seu nome: **Tiago**

Inverso do nome: **Tia**

**tarefa-004:** Faça o somatório da sua data de nascimento.

ex: 07/07/1987 = 7+7+1+9+8+7 = **39**

**tarefa-005:** O usuário informará 10 números que será guardado em um Array e A. Como resultado quero o valor da seguinte fórmula: (1º número + último número)/2.

B. Quais numeros digitados foram maiores do 10 e suas respectivas posições.

**tarefa-006:** Crie uma Hash de quatro valores [a,b,c,d]

1. Me diga o valor da chave b

2. Adicione um novo valor chamado [e] e receba o valor como entrada 3. Me diga quais chave tem valor maior do 3.5

ex: Valor da chave b: **2**

Valores maior de que 5: **d**

Valores maior de que 5: **e**

**tarefa-007:** Use o método each para iterar sobre o Array [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] e imprima cada valor. Mostre apenas os valores maiores que 5.

**tarefa-008:** Quando o número 0 (zero) for digitado, o script deve informar quantos números entre 100 e 200 foram digitados pelo usuário, caso o 0 nao seja informado o script deve continuar.

**tarefa-009:** Leia 20 números inteiros e armazene-os num Array. Armazene os números pares no Array PAR e os números ÍMPARES no Array ímpar. Imprima os três Arrays.

**tarefa-010:** Leia 10 números inteiros e armazene-os num array. Depois, o script deve ordenar esses números em ordem decrescente e imprimi-los.

**tarefa-011:** Algoritmo que leia dois Arrays com 10 elementos cada. Gere um terceiro Arrays de 20 elementos, cujos valores deverão ser compostos pelos elementos intercalados.

ex:Digite 10 numeros

Numero vai pro primeiro Array: 10

Numero vai pro segundo Array: 8

Numero vai pro primeiro Array: 9

Numero vai pro segundo Array: 7

Numero vai pro primeiro Array: 6

Numero vai pro segundo Array: 5

Numero vai pro primeiro Array: 4

Numero vai pro segundo Array: 3

Numero vai pro primeiro Array: 2

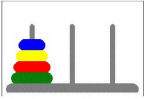
Numero vai pro segundo Array: 1

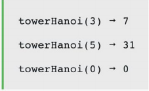
total: [10, 8, 9, 7, 6, 5, 4, 3, 2, 1]

**tarefa-012:**

**Situação Problema**

Existem três torres. O objetivo do jogo é mover todos os discos para a torre nº 3, mas você não pode colocar um disco maior em cima de outro disco menor em nenhuma situação.



Crie uma função que use um número de discos como argumento e retorne a quantidade mínima de etapas necessárias para concluir o jogo. Exemplos: 

**Linguagem de programação**: **Ruby**

**Prazo**: 02/12/2020 - 23h59min59seg - (quarta-feira)

**sistema.brasilsystem@gmail.com**