

## Sistemas Embarcados

**Aluna:** Larissa Santos Silva

**1- Implemente os seguinte esquema abaixo que deverá controlar um display de 7 segmentos que irá conectado diretamente ao Arduino e fará uma contador hexadecimal configurável através de duas teclas onde você pode usa-lo de forma crescente (0-9) e decrescente (9-0). Descreva o resultado usando o simulador em <https://circuits.io>.**

O esquema de segmentos funciona da seguinte maneira, ao clicar no botão direito irá incrementar mais um, podendo chegar até F que equivale a 15. Já o botão esquerdo ao ser clicado decrementa o número atual, podendo chegar até 0 (zero).

**2- Crie um programa para controlar um elevador que atenda 3 andares (1 botão para cada andar), onde cada andar e correspondente as seguintes posições em graus de um servo motor: andar 0 = 0 graus; andar 1 = 127 graus; e andar 3 = 255 graus. Apresente um esquema da ligação dos componentes necessários. Descreva o resultado usando o simulador em <https://circuits.io>.**

O esquema do elevador de três andares funciona da seguinte maneira, tem três botão sendo que cada um possuem um grau diferente, o primeiro botão apresento o grau 0 que seria o estado inicial do servo, já o segundo botão gira para mostra onde está localizado 127 grau e por último o terceiro botão que apresentará 225 grau, podendo assim clicar nos botões de formas desordenadas, dependendo assim de qual grau deseja visualizar.

**3- Implemente os seguinte esquema abaixo que deverá identificar cores com o Arduino utilizando um LDR e um LED RGB. O programa deve gerar como saída as cores como o nome e o número em RGB. Descreva o resultado usando o simulador em <https://circuits.io>.**

Este esquema identifica de mostra as cores próximas, por exemplo quando uma cor vermelha está perto do LDR, logo a LED ficar vermelha já que foi identificada a cor vermelha.

**4- Defina o que é teste de software e descreva o microciclo do TDD.**

Teste de software é um processo, ou série de processos, projetados para certificar que o código do programa para computador faz o que foi projetado pra fazer, sendo assim previsível e não apresentar comportamento inesperados pelo usuário.

TDD é uma técnica para a construção de software de forma incremental. Simplificando, nenhum código de produção é escrito sem antes escrever um teste de unidade de falha, além disso é uma maneira de resolver os problemas de programação, já q ue ajuda os desenvolvedores de software a tomar boas decisões de design, ou seja,

se escreve um pequeno teste, em seguida se escrever código de produção para que passe no teste, enquanto não quebra nenhum teste existente.

Fazendo você decidir o que deseja antes de construí-lo. Ele fornece um feedback se tudo está funcionando com suas expectativas atuais.

**6- Faça uma pesquisa sobre três frameworks para teste de unidade e apresente as vantagens e desvantagens de cada um, bem como, uma tabela descrevendo os itens (assertivas, análise de memory leak e outros) suportados por cada um deles.**

- **JUnit**

JUnit é um **framework de teste de regressão** usado por desenvolvedores para implementar testes de unidade em Java e acelerar a velocidade de programação e aumentar a qualidade do código. JUnit Framework pode ser facilmente integrado com qualquer um dos seguintes eclipse, formiga e maven.

Funcionalidades do JUnit Test Framework

O framework de teste JUnit fornece os seguintes recursos importantes:

- ✓ Jogos
- ✓ Conjuntos de teste
- ✓ Corredores de teste
- ✓ Classes JUnit

As classes de JUnit são classes importantes, usadas em escrever e testar JUnits. Algumas das classes importantes são -

- ✓ **Assert** - Contém um conjunto de métodos assert.
- ✓ **TestCase** - Contém um caso de teste que define o dispositivo para executar vários testes.
- ✓ **TestResult** - Contém métodos para coletar os resultados da execução de um caso de teste.

- **Capybara**

Capybara é um framework de automação de testes em aplicações web, open-source, e escrito em Ruby. É utilizado para testar as aplicações simulando as ações que os usuários reais executariam na aplicação. Para utilizar o Capybara é necessário ter instalado o Ruby versão igual ou superior a 1.9.3.

A instalação do Capybara se dá através do comando abaixo:

gem install capybara

A simulação das ações dos usuários é realizada a partir da utilização de Drivers, que são frameworks que controlam navegadores ou que simulam a execução dos mesmos em memória, tornando mais rápida a execução dos testes. Algumas ações simuladas seriam, por exemplo: Acessar URLs, Clicar, Preencher campos, Verificar se valores foram apresentados na tela, etc.

Vemos abaixo alguns comandos do Capybara utilizados para simular as ações dos usuários:

| Comando      | Descrição  | Exemplo                              |
|--------------|--|--------------------------------------|
| visit        | Acessar uma URL  | visit "http://www.qualister.com.br/" |
| click_on     | Clica sobre um link, botões do tipo "submit" ou "button"   | click_on "Entrar"                    |
| fill_in      | Adiciona um texto a um elemento ou campo                   | fill_in "login", with: "julio.lima"  |
| has_content? | Verifica se um texto está sendo apresentado em um elemento | page.has_content?("Júlio de Lima")   |
| visible?     | Verifica se um elemento está visível                       | find(".msg_sucesso").visible?        |

Os comandos são uma linguagem de domínio específico do framework Capybara, desta forma, ao substituir o driver, os comandos não mudam.

Por padrão, o driver utilizado pelo Capybara é o RakeTest, driver usado para testar aplicações que possuem interface Rake (Rails, Sinatra e outros Frameworks Ruby), escrito puramente em Ruby. Um driver rápido, mas limitado, pois não há suporte a Javascript e nem a aplicações que não possuem interface Rake.

- **Poltergeist**

Poltergeist é um driver que permite que você rode seus testes em browsers headless. Possui suporte a Javascript e tem suporte a aplicações sem e com interface Rake, uma vantagem sobre os demais drivers.

Para utilizar este framework é necessário substituir o driver RakeTest pelo Poltergeist, e como vimos na seção anterior, não é necessário alterar os comandos de interação entre o usuário e a aplicação que foram adicionados ao script de teste.