

UNIVERSIDADE FEDERAL DO AMAZONAS – UFAM
INSTITUTO DE CIÊNCIAS EXATAS E TECNOLOGIA – ICET
ENGENHARIA DE SOFTWARE

ARQUITETURA DE SOFTWARE

COMPONENTES:

ALEXSANDER RENE FERREIRA DE OLIVEIRA

JEAN DA SILVA FERREIRA

MICAELLE DE SOUZA QUEIROZ

SHERMAM TÁCIA DA COSTA LIMA

ITACOATIARA – AM

UNIVERSIDADE FEDERAL DO AMAZONAS – UFAM
INSTITUTO DE CIÊNCIAS EXATAS E TECNOLOGIA – ICET
ENGENHARIA DE SOFTWARE

HISTÓRICO DE VERSÃO

DATA	VERSÃO	DESCRIÇÃO	AUTORES
19/19/2019	0.1	Definição da arquitetura e os conceitos.	SHERMAM LIMA

COMPONENTES:

ALEXSANDER RENE FERREIRA DE OLIVEIRA

JEAN DA SILVA FERREIRA

MICAELLE DE SOUZA QUEIROZ

SHERMAM TÁCIA DA COSTA LIMA

ITACOATIARA – AM

Sumário

1. INTRODUÇÃO	4
2. ARQUITETURA APRESENTADA.....	4
3. REPRESENTAÇÃO DA ARQUITETURA	7
3.1 <i>Node.js</i>	7
.....	8
3.2 <i>Arduino</i>	8
4. ARQUITETURA DOS SERVIÇOS	10
5. REFERÊNCIAS	10

1. INTRODUÇÃO

O objetivo deste documento é fornecer uma visão geral e abrangente arquitetural do sistema System Call. Ele deve mostrar de forma clara e objetiva as decisões arquiteturais que foram tomadas em relação ao projeto, fornecendo as informações necessárias para desenvolvedores e demais envolvidas em termos de estrutura da aplicação e tecnologias utilizadas.

Este documento foi construído sobre a visão da arquitetura utilizada na implementação do sistema System Call, de forma a explicitar as decisões estabelecidas. Nele serão descritos os padrões arquiteturais adotados, frameworks escolhidos para o desenvolvimento do projeto. Com o objetivo de automatizar a chamada em sala de aula por meio do leitor biométrico.

2. ARQUITETURA APRESENTADA

A arquitetura utilizada é MVC sendo um padrão de arquitetura de aplicações que divide a aplicação em três camadas: a visão (view), o modelo (model), e o controlador (controller). Traduzido para o português, a expressão significa: modelo-visão-controlador.

O padrão MVC foi desenvolvido em 1979 por Trygve Reenskaug com a finalidade de ser utilizado como arquitetura para aplicativos desktop. Entretanto, o padrão se popularizou para uso em sistemas web, a partir da adesão de milhares de Frameworks de mercado.

Em termos práticos, e de forma resumida, utilizar do padrão MVC significa:

(Dividir a aplicação em camadas: uma da interface do usuário denominada View, uma para manipulação lógica de dados chamada Model, e uma terceira camada de fluxo da aplicação chamada Control)

Criar a possibilidade de exibir uma mesma lógica de negócios através de várias interfaces.

Isolar a camada de negócios (Model) das demais camadas do sistema, de forma a facilitar a sustentabilidade do código

A implementação do controlador deve permitir que esta camada receba os eventos da interface e os converta em ações no modelo

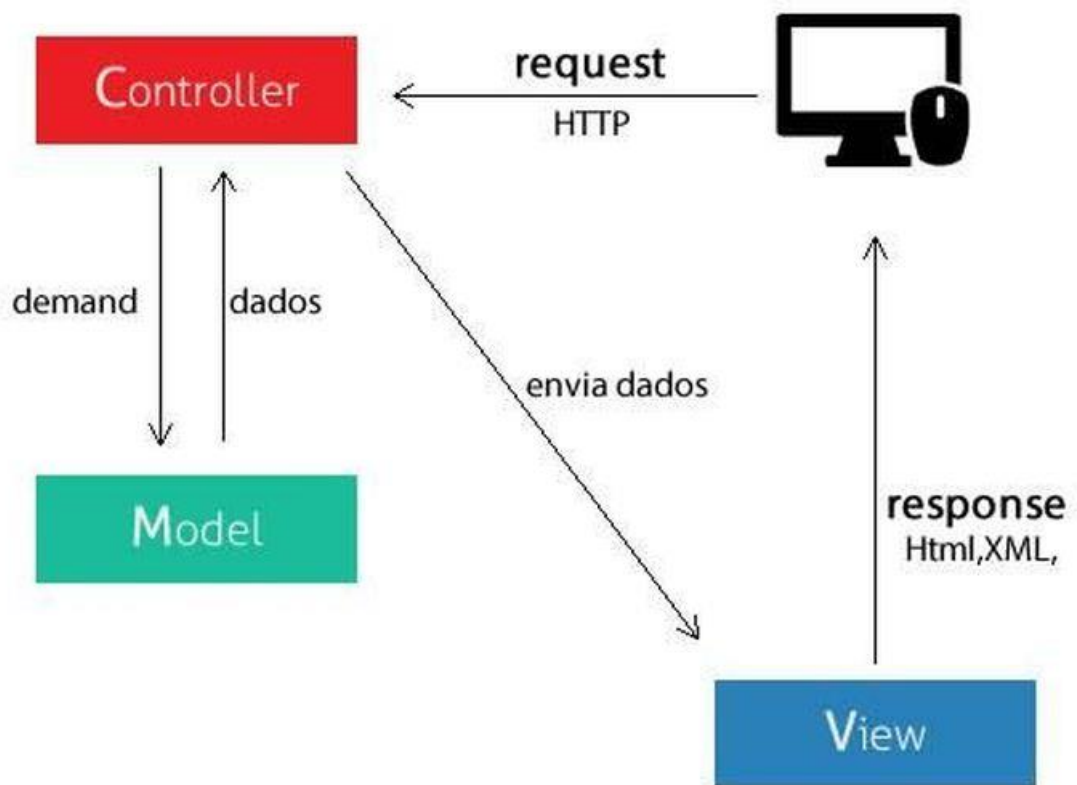


Figura 1 - Arquitetura MVC

A Camada View do MVC

- É a camada que exibe uma representação dos dados.
- É camada de interface com usuário (view).
- Também conhecida como cliente-side.
- Faz a exibição dos dados, utilizando-se de #HTML e/ou XML.
- É responsável por usar as informações modeladas para produzir interfaces de apresentação conforme a necessidade.

A Camada Model do MVC

- É a camada que contém a estrutura de dados de uma parte específica da aplicação
- Usualmente portada em **JSON**
- Responsável pela leitura, manipulação e validação de dados, e também de suas validações.
- Responsável por tratar as regras de negócio.
- Obtém os dados e os traduz em informações relevantes para serem exibidas pela View.
- Notifica a view e o controlador associados quando há uma mudança em seu estado.

A Camada Controller do MVC

- É a camada de controle.
- Exerce o controle de qual modelo deverá ser aplicado e qual view será mostrado ao usuário.
- Podemos dizer que esta camada faz uma gerência das outras duas camadas.
- O controlador manipula e roteia as requisições dos usuários.
- Interpreta as requisições submetidas pelo usuário e traduz em comandos que são enviados para o (Model) e/ou para a View) .
- Valida as requisições dos usuários de acordo com as regras de autenticação e autorização.

Vantagens do Model-View-Controller (MVC)

- Melhor nível de sustentabilidade, pois facilita a manutenção da aplicação
- Melhor performance, graças à separação em camadas
- Fácil transformação da interface, sem que haja necessidade de modificar a camada de negócio
- Melhor desempenho e produtividade, graças à estrutura de pacotes modulares
- A arquitetura modular permite aos desenvolvedores e designers trabalharem em paralelo
- Partes da aplicação podem ser modificadas sem a necessidade de alterar outras

3. REPRESENTAÇÃO DA ARQUITETURA

O projeto irá funcionar da seguinte maneira, para o professor responsável pela disciplina ele deve liberar o acesso ao sensor biométrico, para então o sistema está disponível para fazer essa coleta de dados. O protótipo do sensor é feito em Arduino, com isso o Arduino realiza a comunicação com o sistema web que é projetado na tecnologia do framework node.js.

3.1 Node.js

Apesar de recente, o Node.js já é utilizado por grandes empresas no mercado de tecnologia, como Netflix, Uber e LinkedIn.

O principal motivo de sua adoção é a sua alta capacidade de escala. Além disso, sua arquitetura, flexibilidade e baixo custo, o tornam uma boa escolha para implementação de Microsserviços e componentes da arquitetura Serverless. Inclusive, os principais fornecedores de produtos e serviços Cloud já têm suporte para desenvolvimento de soluções escaláveis utilizando o Node.js.

O Node.js é uma plataforma construída sobre o motor JavaScript do Google Chrome (V8) para facilmente construir aplicações de rede rápidas e escaláveis. Node.js usa um modelo de I/O direcionada a evento não bloqueante que o torna leve e eficiente, ideal para aplicações em tempo real com troca intensa de dados através de dispositivos distribuídos.

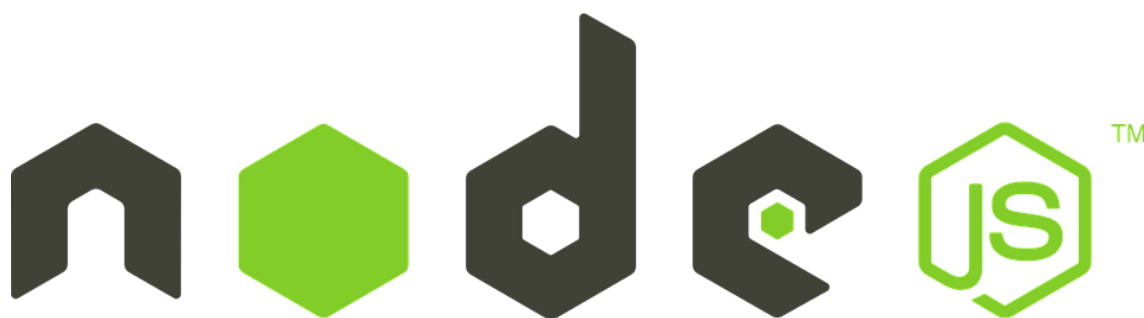


Figura 2 - Framework Node.js

O node.js permite Produtividade da equipe pois:

Maior repositório do mundo: O NPM fornece pacotes de código reusáveis e provavelmente aquela integração que você precisa fazer com outro sistema ou banco de dados já está implementado e disponível gratuitamente para instalar via NPM.

Mesma linguagem no frontend e backend: Javascript é a linguagem padrão para desenvolvimento web client-side. Empresas de desenvolvimento Web contam com esse know-how como um ponto de partida importante para iniciar o uso do Node.js. Além disso, esse fator pode representar ganhos de reutilização de código e criação de equipes multidisciplinares, com melhor aproveitamento de recursos.

Ambiente de inovação: Possibilidade de deploys e iterações mais rápidas, e resolução de problemas On the Fly. Isso também permite a criação de soluções próprias e inovadoras, como fez o Uber criando produtos em Node.js para resolver alguns de seus problemas

3.2 *Arduino*

Arduino é uma plataforma open-source de prototipagem eletrônica com hardware e software flexíveis e fáceis de usar, destinado a artistas, designers, hobbistas e qualquer pessoa interessada em criar objetos ou ambientes interativos.



Figura 3 - Plataforma Arduino

Ou seja, O Arduino é uma plataforma formada por dois componentes: A placa, que é o Hardware que usaremos para construir nossos projetos e a IDE Arduino, que é o Software onde escrevemos o que queremos que a placa faça.

A maior vantagem dessa plataforma de desenvolvimento sobre as demais é a sua facilidade de sua utilização, pois, pessoas que não são da área técnica podem aprender o básico e criar seus próprios projetos em um intervalo de tempo relativamente curto.

Os Arduinos possuem funcionamento semelhante ao de um pequeno computador capaz de interpretar entradas e controlar as saídas afim de criar sistemas automáticos. Para isso, você precisa programá-lo.

Programação nada mais é que falar ao controlador quais decisões devem ser tomadas em cada circunstancia. Para isso, escrevemos um código que segue uma sequência logica de tomada de decisões que leva em conta as vareáveis que serão lidas e/ou controladas.

Para programar essas placas, ou seja, ensiná-las a desempenharem a as funcionalidades que você deseja, basta utilizarmos a sua IDE (ambiente integrado de desenvolvimento), que por sua vez, é um software onde podemos escrever um código em uma linguagem semelhante a C/C++, o qual será traduzido, após a compilação, em um código compreensível pela nossa placa.

4. ARQUITETURA DOS SERVIÇOS

A figura a baixo mostra como é transmitido as informações, o arduino mantém a comunicação com node.js enviando os dados coletados pelo sensor biométrico.

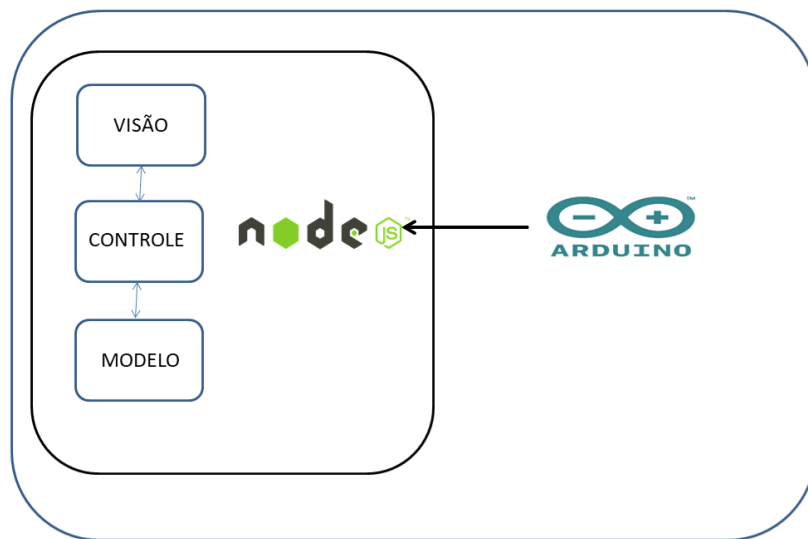


Figura 4 - Arquitetura dos Serviços

5. REFERÊNCIAS

ARTEFATO: DOCUMENTO DE ARQUITETURA DE SOFTWARE. FUNPAR. Disponível em:

http://www.funpar.ufpr.br:8080/rup/process/artifact/ar_sadoc.htm.

Acesso em: 04 de Setembro de 2019

Node.js: estruturando suas aplicações. Disponível em: <https://imasters.com.br/front-end/node-js-estruturando-suas-aplicacoes> Acesso em: 23 de Agosto 2019

NODE.JS: DOCUMENTAÇÃO DO NODE: <https://nodejs.org/en/docs/>. Acesso em 01 de Setembro de 2018.

MEDIUM: What exactly is Node.js? Disponível em: <https://medium.freecodecamp.org/what-exactly-is-node-js-ae36e97449f5>. Acesso em 19 de Setembro de 2018.