

Introdução

1. Introdução geral: 5 min

- a. Iremos criar um aplicativo mobile que poderá ser “exportado” para Android, IOS e Windows Phone e este irá realizar tráfego apenas de dados com um serviço Api Rest no servidor.
- b. Aplicativo móvel:
 - i. Aplicativos mobile são softwares instalados localmente nos sistemas operacionais de hardwares de fácil locomoção como celulares, relógios, carros, tevês e etc..
 - ii. Hoje em dia os sistemas operacionais com maior relevância é o Android baseado Java ou GO, IOS baseado em C++ ou Swift e o Windows Phone baseado em C#.
 - iii. Atualmente os aplicativos são dependentes de conexão com Web Serviços para sincronia de dados entre aparelhos, estes Web Serviços podem ser baseados em qualquer linguagem de programação.
- c. Aplicativos híbridos:
 - i. No começo o desenvolvimento de um aplicativo era necessário que fosse feita em pelo menos três linguagens de programação distintas, foi quando surgiu os aplicativos híbridos no qual se utiliza uma só linguagem de programação interpretada em todos os sistemas operacionais (não somente os de uso móvel).
 - ii. Aplicativos híbridos são somente em HTML5 e JavaScript o que o torna de fácil interpretação.
 - iii. Os aplicativos híbridos são recomendados para o uso em sistemas no qual se tenha um controle simples sobre o hardware, como câmeras, gps, acelerômetro, gestos simples e etc.
 - iv. Os aplicativos híbridos não são recomendados para uso de sistemas de controle mais complexo do hardware ou para jogos. Alguns controles complexos de aparelho como vários toques na tela (reconhecer três dedos de uma vez), precisão de gestos, controle avançado de câmeras, controle de processadores e etc.
- d. Tecnologias:
 - i. Os aplicativos híbridos utilizam o framework Open Source Cordova para intermediar entre o JavaScript e o hardware.
 - ii. Os aplicativos híbridos podem ser escritos puramente com HTML 5 e JavaScript, mas se utiliza alguns frameworks para facilitar e agilizar o desenvolvimento e manutenção como o Phonegap, Xamarin, Ionic e etc.
 - iii. Iremos utilizar o framework Ionic por ser Open Source e por que ele fornece componentes (já integrado com o Cordova) para o framework Angular também Open Source. Além de fornecer o layout pronto para os principais sistemas operacionais de destino no caso Android, IOS e Windows Phone.
 - iv. Iremos utilizar o JavaScript pelo o framework Angular pois seus componentes prontos dão uma agilidade no desenvolvimento além de forçar o uso de boas práticas de desenvolvimento.
 - v. Todos mencionados até então só tecnologias do lado cliente, ou seja, que irá realizar requisições a um servidor de dados, mais conhecido

como front-end. Já do lado servidor conhecido com Back-end, mesmo tendo outras opções tecnologias (algumas até melhor indicada para uso em aplicações mobile) iremos utilizar o .Net Web API por se tratar de algo familiar em nosso desenvolvimento.

1º Dia – Web API+ Ambiente, Projeto e Hello World

2. Projeto Web API .Net: 25min

- a. Resumo API
 - i. Uma tecnologia de uso no Back-end, o API uma exposição de serviços somente via HTTP com REST para servidor e clientes (mobile, browser e aplicação local) que é utilizada para se ter somente valores em um banco de dados, com o formato padrão JSON. O REST (Transferência do Estado Representativo) é um conjunto de arquiteturas aplicadas a componente, ele ignora os detalhes da implementação de componente e a sintaxe de protocolo para focar nas funções do componente.
- b. Criando Web API ****CÓDIGO**
 - i. Abrir solução com estrutura padrão já criada.
 - ii. Mostrar tela padrões de projetos usados.
- c. Configurando filtros no Web API ****CÓDIGO**
 - i. Mostrar classe de configurações de filtros do Controller.
- d. Configurando cultura no Web API ****CÓDIGO**
 - i. Mostrar configurações de captura de dados da requisição no Global.asax.
- e. Configurando CORS no Web API ****CÓDIGO**
 - i. Mostrar configurações de CORS no Global.asax
- f. IOC no Web API ****CÓDIGO**
 - i. Mostrar projeto de IOC padrão.
 - ii. Mostrar configuração de IOC no Global.asax.
- g. Criando CRUD de entidade
 - i. Criar Controller.
 - ii. Mostrar Actions do Controller sobre cada tipo de requisição.
- h. Testando Web API
 - i. Iniciar Solution.
 - ii. Abrir o Postman no Google Chrome, usado para realizar requisições Http das mais variadas formas e tipos.
 - iii. Executar a Action de Update.

3. Instalação do ambiente para o Angular: 30min

- a. Resumo do Node JS
 - i. Uma tecnologia que pode ser usada tanto no Front-end quanto no Back-end, é uma plataforma modular criada sobre o motor JavaScript V8 do Google Chrome para deixar fácil a criação de aplicações rápidas e principalmente escaláveis (habilidade de manipular o aumento do tamanho e uso da aplicação de forma uniforme ou estar preparado para crescer), é ideal para aplicações em tempo real com troca intensa de dados através de vários dispositivos.
- b. Instalação do Node JS ****CÓDIGO**
 - i. Mostrar página download.
 - ii. Mostrar confirmação de instalado e habilitado.
 - iii. Mostrar como inicia um novo projeto Node.
 - iv. Explicar a função do “package.js”.
- c. Resumo dos Node Modules
 - i. O Node conta com o NPM (Gerenciador de Pacotes Node) é um repositório online para publicação de projetos de código aberto para o

- Node e é também uma ferramenta de linha comando que interage e faz gerenciamento de instalação, versões e dependências de pacotes.
- ii. Mostrar instalação de pacote.
 - iii. Ao executar o comando de instalação de um pacote será feito o download do módulo para o Node, por padrão na pasta “node_modules” e adicionado a referência no arquivo de configuração de projeto Node o “package.js”.
 - iv. Mostrar como iniciar projeto.
 - d. Instalando o Visual Studio Code ****CÓDIGO**
 - i. Mostra página download VS Code.
 - ii. Mostrar principais janelas:
 - 1. Mostrar pastas.
 - 2. Mostrar busca.
 - 3. Mostrar controle de versão.
 - 4. Mostrar controle de debug.
 - 5. Mostrar controle de extensões.

4. Projeto Angular: 1h

- a. Resumo do Angular
 - i. Tecnologia Front-end, é um framework JavaScript open-source mantido pelo a Google, está na qual utiliza em vários de seus projetos tanto internos quanto comerciais. O Angular cria e executa aplicações SPA (Aplicações de Página Única, são aplicações que não recarrega ou muda de página, todos os conteúdos são atualizados em trechos da página.), melhorando a experiência do usuário, com o objetivo de facilitar o desenvolvimento e principalmente o teste dos aplicativos. Têm a estrutura baseada em componentes (estimula o reuso de código) para manipulação do HTML5, trazendo serviços comumente designados ao lado servidor diminuindo o peso do mesmo e levando a uma aplicação mais leve. Como segue o padrão MVC encoraja o baixo acoplamento. Com o surgimento do Angular se deu início a separação mais clara do Back-end e Front-end, o que antes era tratado somente pelo o servidor agora é dividido a carga com dispositivo solicitante, aumentando assim desempenho dos servidores e melhorando a usabilidade de uma aplicação para o usuário.
- b. Resumo do TypeScript
 - i. É uma linguagem de desenvolvimento JavaScript em larga escala mantida pela a Microsoft, pois utiliza uma estrutura de uso forte de tipos (muito semelhante ao C#) e depois o código é compilado para JavaScript puro.
- c. Instalando o TypeScript
 - i. Mostrar página do TypeScript.
 - ii. Instalar o pacote via NPM
- d. Resumo @Angular/Cli
 - i. É um pacote facilitador de criação e manutenção de aplicativo em Angular
- e. Instalação do @Angular/Cli
 - i. Mostrar página do @Angular/Cli.
 - ii. Instalar o pacote via NPM.
- f. Criando o projeto Angular ****CÓDIGO**
 - i. Executar criação de projeto angular;
 - ii. Executar aplicação angular;
 - iii. Mostrar no browser.

5. Criando Hello World Angular: 2h

- a. Resumo de Module
 - i. Os módulos do Angular ajudam a organizar a aplicação em blocos de funcionalidades.
- b. Criando Module
 - i. Explicação do módulo raiz;
 - ii. Executar criação de um novo módulo.
- c. Resumo de Service
 - i. Os serviços no Angular são usados para centralizar acesso a dados e aplicação de regra de negócio aplicável em Front-end assim como serviços ou operações em comum em todo o módulo ou aplicação. No MVC seriam a camada Model.
- d. Criando Service
 - i. Executar a criação de uma classe;
 - ii. Executar a criação de um serviço;
 - iii. Configurar um serviço;
 - iv. Adicionar ao módulo.
 - v. Mostrar teste do serviço.
- e. Resumo de Component
 - i. Os componentes são os que preparam e tratam os dados para visualização. No MVC seriam a camada Controller.
- f. Criando Component
 - i. Executar criação de componente;
 - ii. Configurar componente com serviço;
 - iii. Adicionar ao módulo.
 - iv. Mostrar teste do componente.
- g. Resumo de Template
 - i. Os templates é o HTML da aplicação é aonde se têm a saída para o usuário da aplicação. No MVC seriam a camada View.
- h. Criando Template
 - i. Mostrando o template;
 - ii. Mostrando o CSS usado par ao template;
 - iii. Adicionando valores dinâmicos no template.
- i. Resumo de Master/Detail
 - i. No Angular uma das opções de abrir componentes é por seletores, cada componente é vinculado a um Selector, que por sua vez colocado no corpo HTML será substituído pelo o componente referente.
- j. Criando o uso do Master/Detail ****CÓDIGO**
 - i. Executar a criação do segundo componente;
 - ii. Configurar segundo componente;
 - iii. Configurar o primeiro componente para ter o selector do segundo;
 - iv. Adicionar *ngIf para mostrar o segundo componente.
- k. Build aplicação Angular ****CÓDIGO**
 - i. Executar a aplicação

2º Dia – Recursos avançados Angular

6. Melhorando o Hello World Angular: 4:30h

- a. Resumo de Data-Binding
- b. Aplicando o Data-Binding **CÓDIGO
- c. Resumo de Directive
- d. Aplicando Directive **CÓDIGO
- e. Resumo de Pipe
- f. Criando e aplicando Pipe
- g. Resumo de Routing
- h. Aplicando Routing **CÓDIGO
- i. Resumo Routing Guard
- j. Aplicando Routing Guard **CÓDIGO
- k. Publicando aplicação Angular

3º Dia – Ambiente, Projeto, Hello World, Avançado e Build APP Ionic Framework

7. Instalando ambiente Ionic: 30min

- a. Instalando JDK Android **CÓDIGO
- b. Resumo Cordova
- c. Instalando Cordova **CÓDIGO
- d. Resumo Ionic
- e. Instalando Ionic **CÓDIGO

8. Projeto Ionic: 30min

- a. Resumo projetos Ionic
- b. Criando projeto Ionic

9. Criando Hello World Ionic: 1h

- a. Criando Module **CÓDIGO
- b. Criando Component **CÓDIGO
- c. Criando Template **CÓDIGO
- d. Resumo Routing Ionic
- e. Criando Routing Ionic **CÓDIGO
- f. Build aplicação Ionic

10. Melhorando O Hello World Ionic: 1h

- a. Configurações padrões **CÓDIGO
- b. Resumo plug-ins Cordova
- c. Instalando plug-in Geolocation **CÓDIGO
- d. Aplicando plug-in Geolocation

11. Build APP Ionic: 30min

- a. Build aplicação Browser
- b. Build aplicação Android
- c. Build aplicação IOS