

The background features a large, abstract graphic element on the left side. It consists of a teal-colored chevron pattern made of thin, wavy lines, transitioning from dark teal at the top to light teal at the bottom. To the right of this pattern is a large, white, curved shape that resembles a stylized 'G' or a wave. This shape has a thick white outline and a lighter teal fill. The rest of the background is a solid teal color.

# BEM-VINDOS!



BOOTCAMP VNT  
**ANGULAR**  
COM  
**FIREBASE**



# MISSÃO

- *Aprender Angular e Firebase em um dia!*
  - *Fazer uma aplicação básica*
  - *Integrar com o Firebase*
  - *Fazer o deploy.*





# AGENDA - MANHÃ˜

- *Angular*
  - *Setup*
  - *Componentes*
  - *Templates*
  - *Módulos*
  - *Serviços*
  - *Rotas*
  - *Guardas*
  - *Etc.*



VAMOS LÁ!





# MAPEAMENTO DE TERRENO

```
> node -v  
  
> npm -v  
  
> ng --version
```





# EXERCÍCIO 1



```
> sudo npm i -g @angular/cli@1.7.3  
> ng new vnt-shopping-app  
> cd vnt-shopping-app  
> ng serve  
// abrir o projeto em  
// http://localhost:4200/
```

# CAN'T BELIEVE IT WORKED





# COMPONENTES

- *Bloco mais básico da UI Angular;*
- *Principal forma de criar elementos e atributos na página;*
  - *Pode ser usado como uma tag HTML.*
  - *Ciclo de vida: cada vez que é usado, o Angular cria um novo componente, que é destruído assim que deixa de ser visualizado.*
- *Inicialmente criado por aplicação com Angular CLI.*
- *Associado a templates e, opcionalmente, estilos CSS.*



# TEMPLATES

- *Visualização (View) do componente;*
- *Escrito em HTML;*
- *Pode acessar atributos e funções do componente, usando notação especial (Angular expressions);*
- *Pode usar diretivas estruturais do Angular como if, for etc.*
- *Pode referenciar e incluir outros componentes.*



# EXERCÍCIO 2

1. *Instalar o Bootstrap*  
> `npm install bootstrap -s`
2. *O que esse comando fez?*
3. *Incorporar CSS ao build do Angular CLI.*  
> `vi .angular-cli.json`



In case of fire



1. git commit



2. git push



3. leave building



# MÓDULOS

- *Grupos lógicos (funcionais) de componentes e serviços;*
- *Permitem que componentes sejam visíveis à aplicação;*
- *Em aplicações maiores, é uma boa prática isolar cada “parte” do sistema em seu próprio módulo.*
- *Permitem utilizar recursos como lazy loading na aplicação.*



# EXERCÍCIO 3



- *Criar estrutura básica da aplicação*
  - *Cabeçalho*
  - *Menu de navegação*
  - *Itens de exemplo (mock)*
  - *Ícones de ação em cada item!*  
*Para casa: Glyphicons!*
- *HTML puro (por enquanto!)*



# BÔNUS: DIRETIVAS ESTRUTURAIS

- *Controle dinâmico do DOM de acordo com condições e valores definidos no componente (TypeScript).*
- *Similares a condicionais e a estruturas de repetição.*
  - **ngFor** - *repetição (loop)*  
`<li *ngFor="let item of items"> ... </li>`
  - **ngIf** - *condicional*  
`<div *ngIf="isPublic()">Informação pública</div>`



# EXERCÍCIO 4

1. *Criar componentes para o cabeçalho e para a lista:*
  - > ng generate component header
  - > ng g c shopping-list
2. *Criar componente para representar um item da lista:*
  - > ng g c shopping-list/shopping-list-item
3. *Mover os HTMLs para os respectivos componentes!*
  - *O que fazer com os diferentes itens da lista?*



# INJEÇÃO DE DEPENDÊNCIA

- *Problema: relacionamento entre componentes*
  - Aplicação gerenciava ciclo de vida de serviços, componentes etc.
  - Binding de componentes estaticamente no código → pouco reuso;
- *Solução: injeção de dependência*
  - Devs não precisam fazer bindings em tempo de compilação;
  - Aplicação só especifica pontos de injeção - framework lida com eles;
  - *Injector* repassa as dependências aos objetos que precisam delas;
- *Outcome: baixo acoplamento e alta flexibilidade.*
- *TypeScript facilita: **private class members** no construtor!*



# SERVIÇOS

- *Definem funções que podem ser usadas em toda a app;*
- *Geralmente associados a contextos maiores do que um único componente (ou view);*
- *Prática: usar `@Injectable()` - DI;*
- *Principais usos:*
  - *Comunicação entre componentes;*
  - *Realização de tarefas comuns (evitar duplicação de código);*
  - *Permitem maior nível de abstração.*



# EXERCÍCIO 5

1. *Criar um serviço que retorne os itens a exibir na lista*  
    > ng generate service shopping-list
2. *Que arquivos foram afetados por esse comando?*

# E O OSCAR VAI PARA



# PROVIDERS NO ANGULAR



# EXERCÍCIO 5

1. *Criar um serviço que retorne os itens a exibir na lista*  
> ng generate service shopping-list
2. *Que arquivos foram afetados por esse comando?*
3. *Incrementar o serviço com novas operações*
  - add(item)
  - remove(item)
  - cross(item)



# ROTAS

- *Permitem navegação entre diferentes "páginas" (componentes) de uma aplicação Angular;*
- *Geralmente configuradas em um arquivo à parte;*
- *Fazem uso do módulo @angular/router;*
- *Utilizam uma diretiva especial (router-outlet) para injetar o conteúdo no local correto da página;*
- *Para estudar em casa:* ActivatedRoute



# EXERCÍCIO 6

1. *Criar componente para representar uma nova página*  
> ng g c about
2. *Criar rotas para os diferentes componentes*
  - *Cuidado com o auto-import! Verifique o path!*
3. *Preparar o HTML para receber conteúdos diferentes*  
<router-outlet></router-outlet>
4. *Atualizar menu superior com links reais (routerLink)*



# GUARDAS

- *Mecanismo de proteção de rotas;*
- *Permitem validar permissão de acesso a uma determinada rota antes de carregar, abrir ou sair do componente;*
- *São implementadas por meio de interfaces do Angular;*
- *Configuradas individualmente no arquivo de rotas;*
- *Para estudar em casa:* CanActivateChild, CanDeactivate, CanLoad



# EXERCÍCIO 7

1. *Criar um serviço para validar acesso à nova rota:*  
> ng g s auth  
*Implementar a interface CanActivate!*
2. *Configurá-lo como Guard.*  
canActivate: [AuthService]
3. *Vamos deixá-lo "dummy" por enquanto.*  
*Brinque com retornos "true" e "false" para ver o que acontece!*

ALGUÉM  
SE PERDEU?



# ALMOÇO





BOOTCAMP VNT  
**ANGULAR**  
COM  
**FIREBASE**



# AGENDA - TARDE

- *Firebase!*
  - *Introdução*
  - *Integração com Angular*
  - *Banco de dados*
  - *Autenticação*
  - *Hosting*
- *Projeto*
- *Q&A*



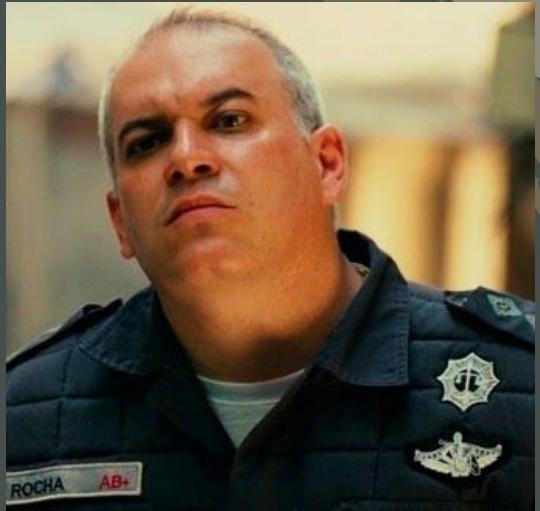
# Firebase





# INTRODUÇÃO AO FIREBASE

- *Fundada em 2011, comprada pelo Google em 2014.*
- ***Gratuito para aplicações pequenas!***
  - "Quem quer rir tem que fazer rir" -- Fábio
- *Plataforma para desenvolvimento web e mobile:*
  - *Dezenas de ferramentas e recursos*
  - *Roda na infraestrutura Google*





### Realtime Database

Armazene e sincronize dados de app em milissegundos



### Crash Reporting

Encontre e priorize bugs, corrija-os mais rapidamente



### Authentication

Autentique usuários de forma simples e segura



### Cloud Storage

Armazene e disponibilize arquivos na escala do Google



### Test Lab para Android

Teste o seu app em dispositivos hospedados pelo Google



### Crashlytics

BETA

Priorize problemas e corrija-os com relatórios de erros avançados e em tempo real



### Cloud Firestore

BETA

Armazene e sincronize dados do app em escala global



### Cloud Functions

BETA

Execute o código de back-end para dispositivos móveis sem precisar gerenciar servidores



### Hosting

Veicle recursos de app da Web com velocidade e segurança



### Monitoramento de desempenho

BETA

Receba insights sobre o desempenho do seu app



### Google Analytics

Receba análise gratuita e ilimitada dos apps



### Previsões

BETA

Defina grupos de usuários dinâmicos com base no comportamento previsto.



### Configuração remota

Modifique seu app sem implantar uma nova versão



### Indexação de apps

Direcione o tráfego das pesquisas para o seu app para dispositivos móveis



### Google AdWords

Direcione as instalações com campanhas de anúncios segmentadas



### Cloud Messaging

Envie mensagens e notificações segmentadas



### Dynamic Links

Direcione a sua expansão usando links diretos com atribuição



### Invites

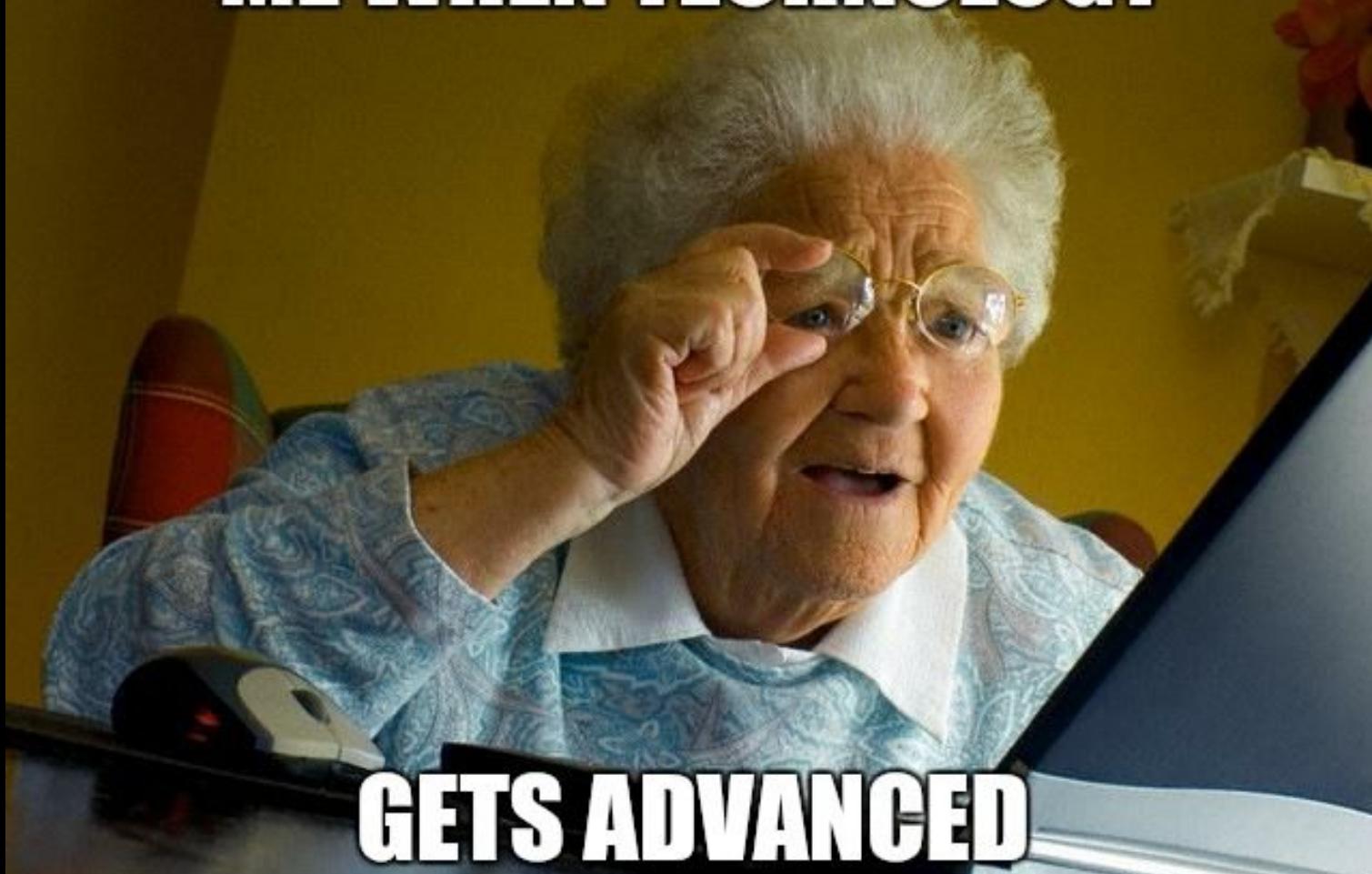
Facilite o compartilhamento do seu app e conteúdo



### AdMob

Maximize a receita com anúncios no app

**ME WHEN TECHNOLOGY**



**GETS ADVANCED**



# SETUP BÁSICO

- *Integração oficialmente suportada pelo Angular!*
- *AngularFire*

```
> npm install firebase angularfire2 --save
```



# EXERCÍCIO 8

1. Acessar  
<https://firebase.google.com/>
2. Fazer login e abrir o console
3. Criar um novo projeto
4. Configurar environment com dados do projeto





# BANCO DE DADOS EM TEMPO REAL

- *Killer features:*
  - *Sincronização em tempo real;*
  - *Off-line;*
  - *Acessível em diferentes dispositivos clientes;*
  - *Escalonável;*
  - *Manupulação de dados por REST.*



# HTTP CLIENT DO ANGULAR

- *Serviço nativo do Angular;*
- *Encapsula chamadas HTTP;*
- *Retorna um Observable para trabalhar o resultado reativamente.*
  - *Bônus:* unsubscribe() é chamado automaticamente pelo Angular após a finalização do request. Menos trabalho para o dev!





# EXERCÍCIO 9

1. *Injetar HTTP Client no service*
2. *Alterar métodos do service para manipular dados do Firebase Database utilizando REST*
  - *Não esqueça de modificar o projeto no Firebase para que não seja necessária autenticação para manipulação de dados!*



# LIGANDO OS PONTOS

- *Integração oficialmente suportada pelo Angular!*
- **AngularFire**  
  > `npm install firebase angularfire2 --save`
- *APIs nativas e simples*





# EXERCÍCIO 10

## 1. *Instalar AngularFire*

```
> npm install firebase angularfire2 --save
```

## 2. *Atualizar service para utilizar os serviços do AngularFire*



# AUTENTICAÇÃO

- Autenticação por senha, telefone e entes federados
- Entre eles Google, Facebook, Twitter e GitHub
- FirebaseUI Auth
- Autenticação anônima





# EXERCÍCIO 11

1. *Alterar o AuthService para usar a autenticação do Firebase*
  - > usar autenticação nos guardas
2. *Usar a autenticação baseada na conta Google*
  - > é necessário autorizar o método de autenticação no projeto do firebase



# HOSTING

- *Permite hospedar a aplicação diretamente nos servidores do Firebase.*
- *Funcionalidades:*
  - *Deploy facilitado;*
  - *Seguro (SSL);*
  - *Firebase CLI;*
  - *Configuração de domínio próprio.*



# EXERCÍCIO 12

1. *Instalar Firebase CLI*

> `npm install -g firebase-tools`

2. *Login conta Firebase*

> `firebase login`

3. *Configurar aplicação*

> `firebase init`

4. *Deploy da aplicação no Firebase*

> `firebase deploy`



PROJETO



# PROJETO FINAL

- ***Terminar a aplicação!***
  - Adicionar quantidade nos itens;
  - Calcular preço total da compra;
  - Dar um visual caprichado com mais CSS.
  - Deixe a imaginação fluir. Novas features são muito bem-vindas!
- ***Enviar por e-mail:***
  - Link do GitHub ou ZIP com o código;
  - URL do deploy no Firebase.
- ***Prazo: 02/05/2018***

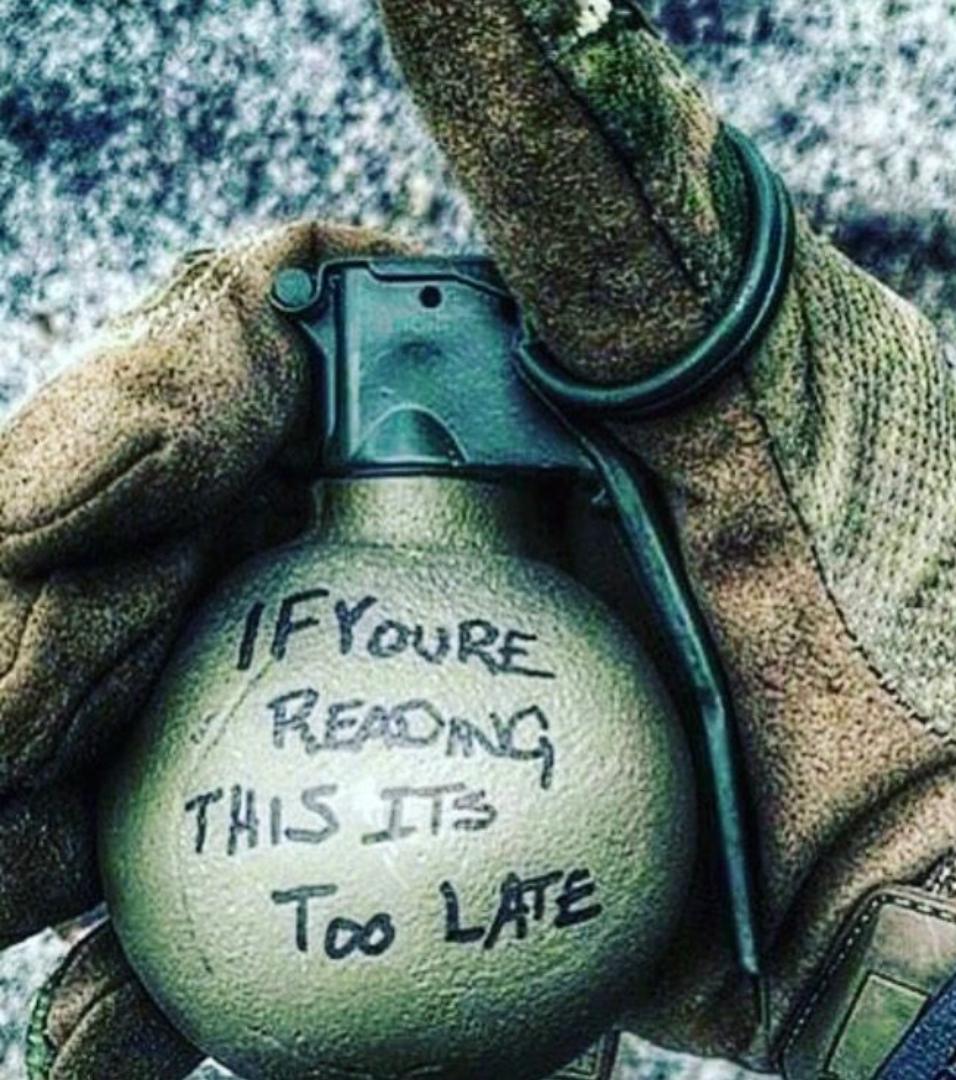


# ONDE APRENDER MAIS

- Tour of Heroes, tutorial (bem básico) no site do Angular;
- YouTube: ótimo material da Loiane Groner;
- Udemy: ótimos cursos bons e baratos
  - Understanding TypeScript
  - Angular 5 - The Complete Guide
  - Reactive JS: Are you ready for the next big paradigm shift?
- Learn Angular 2, feito pela equipe do Ionic;
- Angular 4, ebook gratuito do Assim Hussain.

# MISSÃO CUMPRIDA!





# MUITO OBRIGADO!

**Adriano Gomes**

*adriano.gomes@venturus.org.br*

**Bruno Toffolo**

*bruno.toffolo@venturus.org.br*

**Ramon Ferreira**

*ramon.ferreira@venturus.org.br*



**ventus**