

# Waiter Joe - Uma solução tecnológica

Jáder Hartmann

Ciência da Computação – Universidade de Passo Fundo (UPF) – Campus I  
Av. Brasil Leste, 285 - São José, Passo Fundo.

jaderhartt@gmail.com

**Abstract.** *This meta-paper describes a possible solution for some problems faced by bars and parties currently located in the region. The steps to the development of an application with functions to assist in commercial transactions and establish a communication with the client. Also, generate essential information to identify frequency problems on weak days as well explanation of strong days, all, storage in a database which can be accessed through a system developed for the bar.*

**Resumo.** *Este meta-artigo descreve uma possível solução para alguns problemas enfrentados por bares e festas atualmente na região. Os passos para o desenvolvimento de uma aplicação com funções para auxiliar transações comerciais e estabelecer uma comunicação ativa com o cliente. Também, gerar informações essenciais para identificar problemas de frequência em dias fracos e explicações para dias cheios, isso tudo, armazenado em um banco de dados que pode ser acessado através de um sistema desenvolvido para o bar.*

## 1. Introdução

Existe uma distância significativa entre a comunicação do cliente com o negócio local, e vice-versa. Ainda em seus primórdios, a tecnologia pode ser explorada para ser usada de forma efetiva a fim de aproximar esses dois sujeitos. Tem um importante papel no mundo do comércio, não importando o tamanho e muito menos a área de atuação.

Além do benefício da comunicação, a empresa pode melhorar sua eficiência durante a realização de operações de mercado, preservando recursos do estoque, movimentação do caixa e o principal, a obtenção da informação. Administrar o tempo do cliente também é essencial, pode-se reduzir o tempo ocioso em filas com transações automáticas, por exemplo. Usar recursos e inovações tecnológicas tendem a criar uma imagem forte para empresa.

Dois bares podem ter uma movimentação frequente de clientes e eventos ocasionais, um, cuja divulgação é feita a partir de meios físicos, outro, que faz o uso recursos tecnológicos para auxílio. Qual será o lugar mais movimentado? Causar uma boa primeira impressão pode ser o fator determinante. A inovação é um dos mais importante e mais complexo problema que organizações enfrentam hoje. Inovação é a chave do sucesso para as organizações [INSODE 2011].

## 2. A informação

A realidade que assemelha todos homens é a corrida pela obtenção de conhecimento. A leitura de qualquer documento pode trazer o acesso a uma quantidade de dados, cabendo às pessoas, identificar o que é útil ou não. Esses, são interpretados, analisados a fim de identificar uma informação inteligente, e seu uso racional é considerado a base do conhecimento.

Junto com o capital e a matéria prima, é normal considerar a informação como uma das condições básicas para o desenvolvimento. Seu acesso, é o que pode diferenciar um negócio de outro, tanto quanto no relacionamento com o cliente quanto no seu bem estar pessoal e social. Toda informação em excesso, se torna conjunto de dados desorganizados, confusos, principalmente para um usuário que recém está se adaptando a usar um aplicativo. Para evitar isso, todo dado deve ser minerado, extraindo apenas a informação necessária para identificar problemas e soluções.

## 3. Aplicativos

Desde a popularização dos smartphones, a comunicação pela troca de mensagens, informações e dados, passou a ser essencial ao ponto de que, quando otimizada, o usuário possa recebê-la sem consumir nenhum esforço. Nas lojas virtuais das principais plataformas contemporâneas (Apple Store e Google Play) existem cerca de 4 milhões de aplicativos para download [Statista 2018]. A possibilidade de existir idéias e/ou soluções homogêneas é alta, conseqüentemente, a concorrência pela escolha dos usuários também. Levando em conta que o objetivo de uma aplicação é a manutenção dos usuários atuais e a busca por novos, características únicas se fazem necessárias.

### 3.1. Aplicativos sensíveis ao contexto

Aplicativos sensíveis ao contexto tem por intento gerenciar as informações e assumir o papel semelhante a de um *middleware*, ou seja, o usuário não vai precisar solicitar conhecimento algum, cabe ao próprio aplicativo, organizá-lo e entregá-lo. Dados simples, como o reconhecimento de localização e objetos, por exemplo, foram os primeiros capazes de ser identificados por aplicações sensíveis ao contexto.

Segundo Schilit & Theimer, Active Maps é uma companhia de mapeamento e cartografia que produz rotas para ciclismo em ruas ou montanhas, de acordo com o desejo do cliente. Em 1994, foi a primeira a utilizar o conceito de sensibilidade ao contexto, já usando informações de localização, pessoas e objetos.

Em 1996, outros dados começaram a ser identificados da mesma forma, como a percepção de mudanças de temperatura ou qualquer fator ambiental, utilizando a arquitetura *stick-e note*. No lugar de escrever uma informação em um simples pedaço de papel e colocar em algum local físico, a mesma, é escrita digitalmente, anexada à certa localização e, quando o usuário se aproximar, ela é revelada.

Empregando o uso de um sistema eletrônico, a informação exposta não precisa ser necessariamente um pequeno texto, pode ser a hora atual, temperatura, condição climática e, também, em formatos diferentes, como som e imagem. Basicamente, funciona como um bloco auto-adesivo dinâmico, automatizado e virtual.

O termo computação ubíqua, cunhado inicialmente por Mark Weiser em 1986, surgiu a partir dos conceitos de computação pervasiva e móvel (Figura 1). Oferecendo, um valor semântico ainda maior quando usada em conjunto com o conceito de *stick-e note*. O objetivo da computação pervasiva, é fazer com que os objetos se comuniquem

de modo efetivo, com o mínimo de interação humana, realizando tarefas sendo praticamente invisível ao olho nú. Com os dispositivos sempre disponíveis e conectados pela rede.

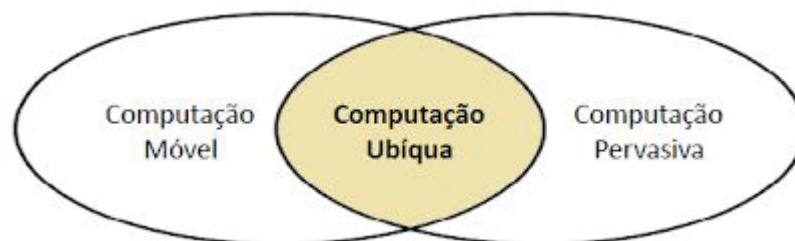


Figura 1. Intersecção de computação pervasiva e móvel

Fonte: TI especialistas, 2015

### 3.2. Computação Ubíqua

Um cenário ubíquo pode ser compreendido como uma analogia a onipresença da informática, promovendo a descentralização, conectividade e diversidade da tecnologia. Entende-se que vários dispositivos cooperam entre si, construindo a inteligência no ambiente refletido pela interação das funções de aplicações distribuídas. Quanto mais presente a tecnologia estiver, menos perceptível deve ser, prevendo as necessidades do usuário e agindo com antecedência.

O projeto UbiReCon [Ferreira 2012], é utilizado na educação e tem um modelo ubíquo que sugere um conteúdo específico analisando apenas o perfil do aluno, em seguida, indica objetos específicos de aprendizagem. Assim como o UbiGroup, que serve como um assistente do professor, identifica os aspectos gerais de um grupo de alunos, recomendando uma série de matérias como auxílio.

Na medicina, o UbHeart, apenas com o uso de sensores, faz um diagnóstico prévio de uma situação de emergência, assim, enviando um alerta para os responsáveis do paciente ou ao hospital que está vinculado, acionando o médico responsável. Outro exemplo, é o conceito de *Smart Homes* (Figura 2), que detém toda a ideia de ubiquidade, tornando um ambiente inteligente com o controle de gastos de energia, luminosidade, temperatura e com os equipamentos programados para satisfazer a vontade de quem está presente.



## Figura 2. Conceito de casas inteligentes e suas funções

Fonte: Computer Concepts USA, 2016

### 4. Solução

O objetivo do trabalho é empregar o uso de aplicativos sensíveis ao contexto a favor do produtor e consumidor em um ambiente comercial. Um bar, tem uma variedade de clientes muito grande, com diferentes faixas etárias, gostos e rendas. Uma quantidade de dados relativamente extensa para ser controlado sem o auxílio de nenhuma tecnologia. Com a chegada de novos recursos, é essencial adotar o uso de novas opções, por exemplo, o *beacon*.

#### 4.1. Beacon

O *beacon* é um dispositivo que manda repetidamente um sinal único para que todos outros dispositivos possam ver (Figura 3). Funciona exatamente como um farol, fica mandando um sinal de luz (sinal de rádio) para que os marinheiros (dispositivos) possam ver. Este, será posicionado estrategicamente para uma distribuição efetiva do seu sinal, identificando se o usuário estiver perto. Se aberto, pode-se notar que sua composição não passa de uma CPU, rádio e baterias de lítio recarregáveis, é encontrado em diferentes formas e tamanhos, incluindo com sensores de temperatura, acelerômetros e até com cabo usb para fornecer energia.



Figura 3. Sinal bluetooth do beacon para o smartphone

Fonte: Realtor Reach, 2017

### 5. Prototipação

Um protótipo é uma amostra ou modelo antecipado de um produto que vai ser construído, serve para ajudar a desenvolver sua estrutura, funções e aparência. Normalmente é utilizado quando uma empresa deseja produzir algo novo e aperfeiçoar seu uso.

Infelizmente, desenhos e ideias podem estar distantes do que você realmente quer. Criando o protótipo, é possível visualizar uma versão sólida que o produto pode vai tomar, determinar aspectos e funções importantes e examinar partes que precisam ser revisadas ou descartadas. Se existe alguma dificuldade durante a produção ou processos que podem criar problemas, vai possibilitar com que o produto seja alterado antes de entrar na fase de implementação.

Para projetos iniciais que necessitam da arrecadação de fundos para desenvolver, é fundamental o uso do protótipo para obter patrocinadores. Apesar de ser apenas um conceito, fica mais fácil vender o produto durante uma apresentação de marketing, com uma versão visual de como vai ficar depois de finalizado. Ideias das pessoas também devem ser levadas em consideração, não importa o quanto bons os analistas são, o cliente pode não gostar de certos aspectos do produto.

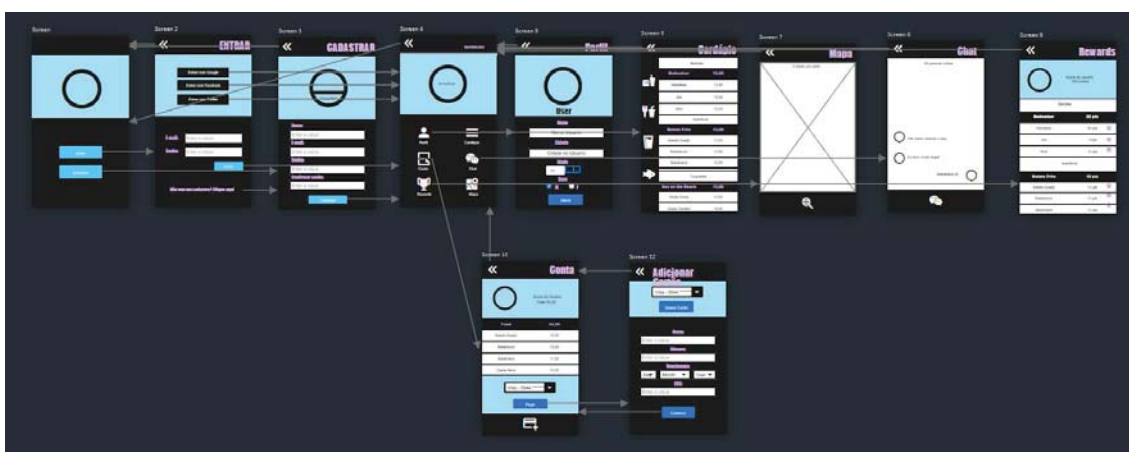


Figura 4. Prototipação do aplicativo WaiterJoe (Mobile)

## 6. Implementação

Com a implementação do aplicativo, serão permitidos ao bar identificar, classificar e bonificar clientes. Identificar quais são os novos, classificar grupos com tipos de bebidas e músicas, e recompensar com produtos e promoções para uma melhor interação de bar e cliente. O desenvolvimento bem sucedido de um novo recurso pode resultar em diferenças significativas em um mercado competitivo.

### 6.1. Tipos de aplicações

O aplicação poderá ser desenvolvida de três formas: híbrida, nativa e *web app*. São os navegadores de internet que hospedam as aplicações web, como um *site* que geralmente fornece uma grande quantidade de informações, o objetivo é condensar-la com o propósito de melhorar sua funcionalidade e exibi-la em site mobile de forma prática. Entretanto, aplicações web não precisam ser baixadas em lojas de aplicativos nem ocupam espaço no seu dispositivo. Tipicamente, são desenvolvidas com JavaScript, CSS, e HTML5 e executam dentro do seu navegador(Chrome, Firefox..).

Progressive web app são híbridos de páginas da internet e aplicações mobile, suprimindo as deficiências de ambas plataformas. Estas aplicações conseguem levantar recursos similares a dos nativos, como mandar mensagens push (notificações), usar funções de touch e vibração, é uma tecnologia recente que está ganhando mercado. Com os aprimoramentos já feitos nesta plataforma, pode-se listar algumas vantagens:

- Fácil construção e manutenção
- Uma aplicação para todas plataformas, desde que consiga rodar o navegador

Até recentemente, estas aplicações só tinham suporte para as plataformas que pudessem usar o Google Chrome, mas o iOS recebeu o funcionamento com a chegada da versão 11.3. Além disso, temos mais algumas desvantagens:

- Precisa de um navegador para rodar, mais lento
- Acervo limitado de funções mobile, menos interação e intuição
- Não consegue fazer uso de todos recursos do dispositivo

Nativo é o tipo mais comum para aplicações mobile. São construídos para plataformas específicas e escritos na linguagem que a plataforma aceita, por exemplo, Objective-C para iOS e java para Android. Também possuem IDE específica e de acordo com o sistema operacional. Ambas empresas, Apple e Google, oferecem suas próprias ferramentas de desenvolvimento, com elementos de interface e SDK(Software Development Kit). Maioria das empresas opta por investir no desenvolvimento de aplicações nativas, por causa de inúmeros benefícios comparados aos outros tipos de aplicações:

- Rápidos e responsivos por ser desenvolvido para uma plataforma específica, melhor performance
- Distribuídos nas lojas de aplicativos (App Store e Play Store)
- Nativos permitem aos desenvolvedores acessar todos recursos dados pela plataforma criando otimizações e oferecendo infinitas possibilidades de interação e intuição
- Não necessita conexão com a internet
- Para o usuário, a navegação flui mais natural utilizando uma interface padrão da plataforma

Também tem algumas desvantagens:

- Maior tempo de implementação

Aplicativos mobile precisam baixar atualizações frequentemente para aperfeiçoar sua acessibilidade, enquanto as aplicações web conseguem atualizar sem envolvimento do usuário. O preço inicial para desenvolver um software nativo pode ser caro, mas com a quantidade de recurso disponível, você vai salvar muito dinheiro pensando a longo prazo (manutenção, atualização).

Aplicações híbridas trabalham em todas plataformas e se comportam semelhante aos nativos. É a combinação essencial dos recursos dos dois tipos citados acima. Apesar de aplicativos nativos terem uma melhor compatibilidade com o sistema operacional do smartphone, serão apresentados os motivos do porque o trabalho será desenvolvido desta forma com um Framework (Ionic). Usuários conseguem instalar em seu dispositivo como os nativos mas é na verdade uma aplicação web, construída a base de HTML, Javascript e CSS. Pode fazer qualquer coisa que o HTML5 pode fazer. Suas vantagens:

- Construído com tecnologia web (HTML, CSS, Javascript) e não necessita de um navegador para funcionar
- Rápido desenvolvimento, mais barato e um aplicativo para todas plataformas (um único código)
- Tem acesso a maioria dos recursos do smartphone (armazenamento, câmera)

Algumas desvantagens:

- Mais lento e menos interativo que os nativos

Com uma aplicação híbrida, é preciso resolver problemas para saciar todas plataformas, o que torna o conserto de um erro, mais difícil. Performance pode ser um

pouco lenta por rodar em *webview* (todo site em um frame), e também é impossível customizar o aplicativo de acordo com a plataforma, ambas usam o mesmo. Adicionalmente, quanto maior for a customização feita, mais cara a solução. Pensando a longo prazo, a aplicação híbrida pode acabar custando mais cara que a nativa.

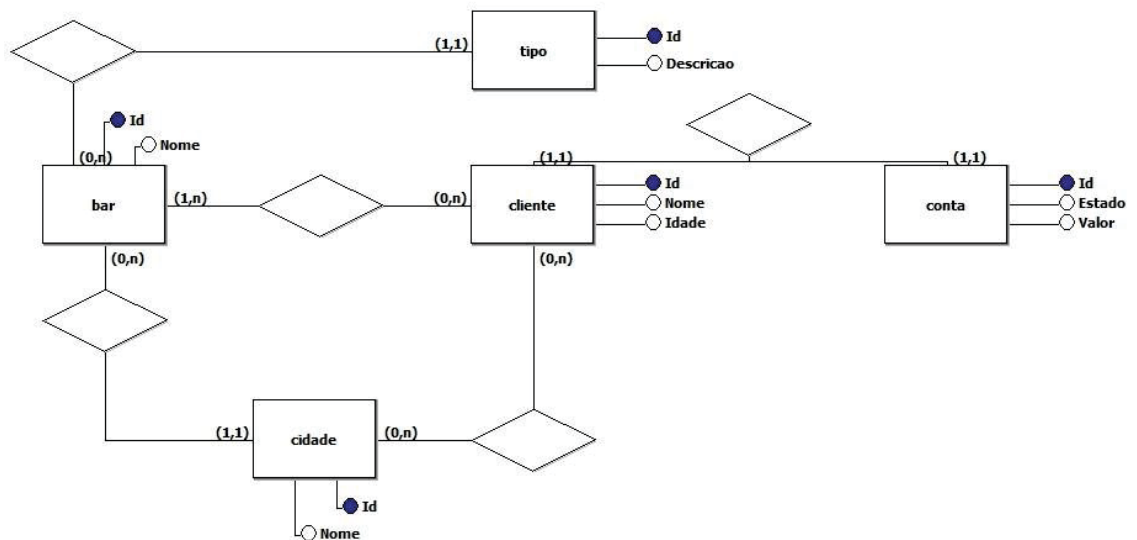
## 6.2. Framework

Escolher um framework para o desenvolvimento de software *mobile*, pode ser considerada uma das etapas mais importantes no processo de criação de um novo produto. Ionic foi criado em 2013 pela Drifty Corporation. É um framework de código aberto e grátis que permite o desenvolvimento de aplicações móveis híbridas. Licenciado pelo MIT, seu uso permanece sem custo por tempo ilimitado pelo fato de ser aperfeiçoado pela comunidade, que oferece todos os recursos para o desenvolvimento de aplicativos altamente interativos. Tem suporte para várias plataformas, com ênfase para Android e iOS, mas também é possível desenvolver para Windows e Blackberry. Utiliza o Angular no lugar do JQuery que alavanca as transições CSS, transforma as animações com o propósito de melhorar a Gpu, otimizando o tempo de processador do dispositivo. A boa notícia é que o conhecimento em desenvolvimento de aplicações web pode ser usado para desenvolvimento de aplicações mobile.

## 7. Banco de dados

### 7.1. Banco de dados relacionais

Banco de dados relacionais envolvem o uso e a administração de vários tipos de transações. Sua estrutura permite conectar informações de diferentes tabelas através de chaves estrangeiras, são usadas para identificar, ou alterar algum dado específico de forma atômica. Estas chaves podem ser referenciadas por outras tabelas, o que é bastante conveniente para aplicações que requerem análise de grandes quantidades de dados. Então, se precisar lidar com grandes bases, com transações, rotinas e análises, deve-se optar pelo banco de dados relacional. Na figura 5 podemos ver como ficaria o modelo de banco do WaiterJoe.



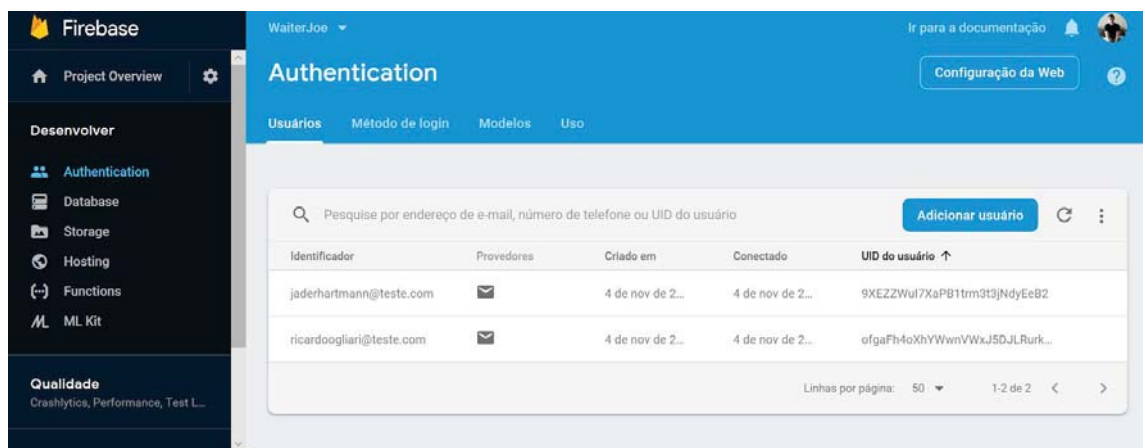
**Figura 5. Modelo de dados relacional da aplicação WaiterJoe**

## 7.2. Banco de dados não relacionais

Um banco de dados não relacional é qualquer um que não segue o modelo tradicional de sistemas de gerenciamentos dados relacionais. Essa categoria é também referida a NoSQL. Estes, cresceram em popularidade por superar as limitações dos bancos relacionais lidando com Big Data. Refere-se aos dados que crescem e movem tão rapidamente e diversificadamente que estruturas convencionais não conseguem lidar. Estes bancos são mais escaláveis e flexíveis porque:

- Modelo de dados: não precisa de um esquema predefinido de banco de dados tornando muito mais fácil atualizar as bases.
- Estrutura de dados: projetado para lidar com dados desestruturados que não entram em tabelas e colunas. Importante para quase todos tipos de dados gerados atualmente (maioria dos dados hoje, são desestruturados).
- Escalável: sistema pode aumentar horizontalmente, não só verticalmente.
- Modelo de desenvolvimento: bancos NoSQL geralmente tem seu código aberto(OpenSource), o que significa que não é necessário pagar nada para utilizá-lo.

Firebase é uma ferramenta popular, que permite ajudar na criação de aplicativos rápidos sem precisar gerenciar toda uma infraestrutura (Figura 5). Um serviço de banco de dados poderoso, não relacional e em nuvem. Construído a partir da infraestrutura do google, aumenta automaticamente de acordo com a necessidade da sua aplicação (escalável).



The screenshot shows the Firebase Authentication console for a project named 'WaiterJoe'. The interface is in Portuguese. On the left, there is a sidebar with navigation options: Project Overview, Desenvolver (with sub-items: Authentication, Database, Storage, Hosting, Functions, ML Kit), and Qualidade (with sub-items: Crashlytics, Performance, Test L...). The main content area is titled 'Authentication' and has tabs for 'Usuários', 'Método de login', 'Modelos', and 'Uso'. Below the tabs, there is a search bar with the placeholder text 'Pesquise por endereço de e-mail, número de telefone ou UID do usuário' and a blue button labeled 'Adicionar usuário'. Below the search bar is a table with the following columns: 'Identificador', 'Provedores', 'Criado em', 'Conectado', and 'UID do usuário'. The table contains two rows of user data:

Identificador	Provedores	Criado em	Conectado	UID do usuário ↑
jaderhartmann@teste.com	✉	4 de nov de 2...	4 de nov de 2...	9XEZZWui7XaPB1trm3t3jNdyEeB2
ricardoogliari@teste.com	✉	4 de nov de 2...	4 de nov de 2...	ofgaFh4oXhYWwmVWxJ5DJLRurk...

At the bottom right of the table, there is a pagination control showing 'Linhas por página: 50' and '1-2 de 2'.

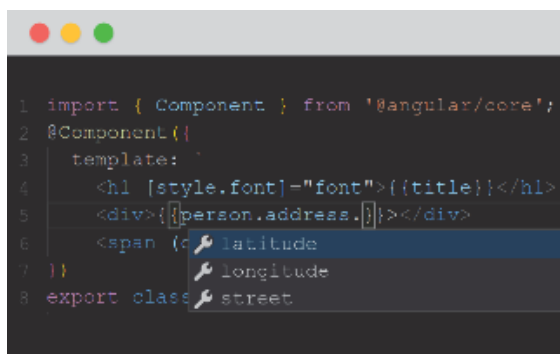
**Figura 6. Exemplo de dados salvos em um banco de dados não relacional da aplicação WaiterJoe**

Não é OpenSource e possui um modelo “Grátis”. Pode ser usado gratuitamente desde que não passe dos limites de acesso grátis. Então, se o plano é criar uma aplicação com muitos usuários, recomenda-se trocar de ferramenta ou checar seus planos pagos. Como o desenvolvimento do trabalho vai ser em cima de tabelas teste, usar um plano sem custo da ferramenta não será um problema.

Angular é uma plataforma OpenSource que facilita a construção de aplicações web e desktop, auxiliando em vários aspectos, como na orientação de objetos, figura 6. Firebase funciona perfeitamente com Angular 2. Juntos, fornecem velocidade e desempenho na sincronização de dados melhorando a experiência do usuário. Para



integrar a os serviços do Firebase com o Angular 2, deve ser usado a biblioteca AngularFire2.



```
1 import { Component } from '@angular/core';
2 @Component({
3   template: `
4     <h1 [style.font]="font">{{title}}</h1>
5     <div>{{person.address.0}}</div>
6     <span class="latitude" >{{person.location.latitude}}</span>
7   `
8 })
9 export class Street
```

Figura 7. Exemplo de uso da orientação de objetos com Angular2

Fonte: SAPO, 2006

## 7. Resultado

Com a implementação do banco é possível salvar dados cruciais para o funcionamento do aplicativo. A partir do momento em que o beacon entra em contato com o bluetooth do celular, um novo dado é salvo. Esse, carrega informações como a identidade do cliente, que primeiramente, permite ao bar analisar fatores de frequência e tempo presencial no estabelecimento. Com uma tabela geral de frequência dos fregueses, pode-se procurar soluções para dias fracos e respostas em dias fortes.

Com futuras implementações, outras funções podem ser adicionadas. Um sistema de pagamento de contas (comanda on-line), para evitar filas em excesso em finais de noite. Ter um controle de gastos, permitindo verificar quanto cada um gastou, possibilitando recompensar clientes fiéis (boa frequência) com produtos sem custo ou promoções.

Para um bar ter acesso ao uso da aplicação, o mesmo deve se cadastrar com os administradores do WaiterJoe, criando um vínculo com o aplicativo. Este, será adicionado ao sistema, possibilitando aos clientes ter acesso às programações diárias e semanais do estabelecimento, como também ao cardápio e comanda (ainda não implementada).

Para fazer uso dos recursos do WaiterJoe, o cliente deve baixá-lo na loja de aplicativos de sua plataforma atual (ainda indisponível para download) e ter seu bluetooth ativado. Para logar-se automaticamente, é necessário uma conta no google plus ou facebook, também é possível cadastro e login por e-mail sem vínculo com outras redes sociais. Acesso à todas funcionalidades atuais do aplicativo e futuras (Figura 8).

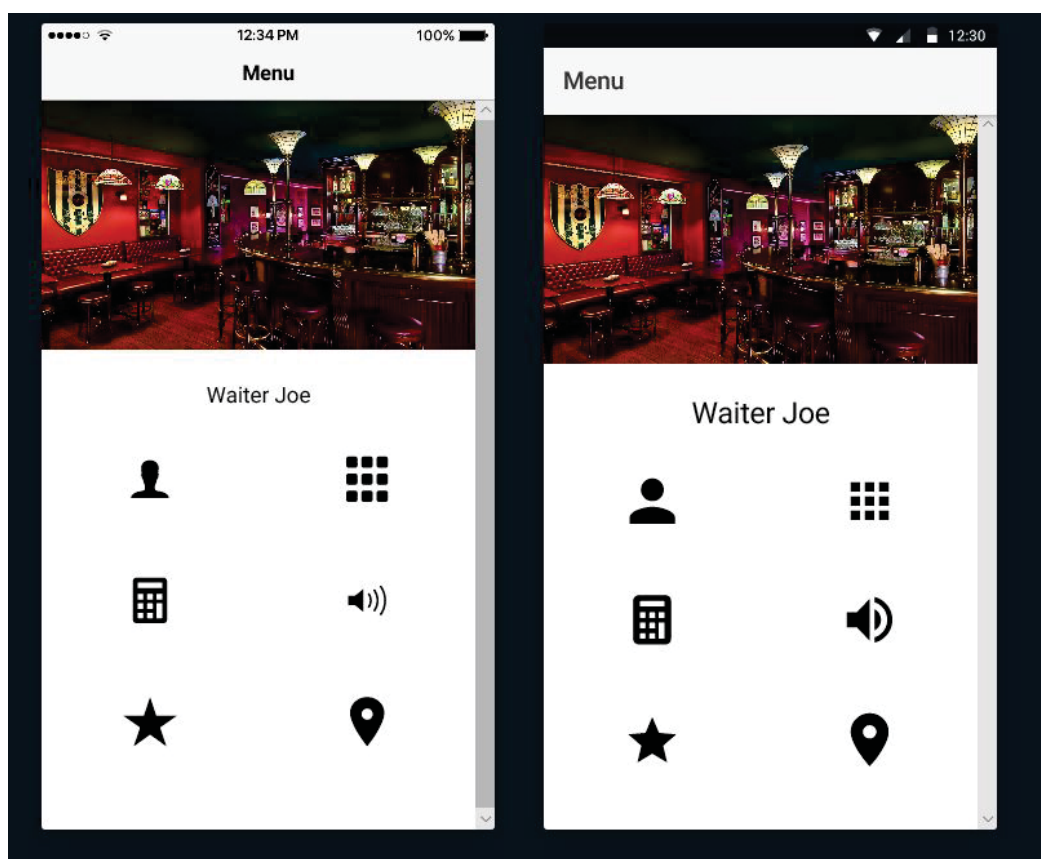


Figura 8. Funções de Perfil, Cardápio, Comanda, Chat, Sistema de Recompensas e Localização

## Referências

- Capurro, R. e Hjørland, B. “O Conceito de Informação”. Disponível em <http://bogliolo.eci.ufmg.br/downloads/CAPURRO.pdf>. Acesso: março/2017.
- Microsoft. Conceito de mineração de dados. Disponível em <https://docs.microsoft.com/pt-br/sql/analysis-services/data-mining/data-mining-concepts?view=sql-server-2017>. Acesso: abril/2017.
- Rocha, L. Play Store passa App Store. Disponível em <https://www.tecmundo.com.br/play-store/72656-play-store-passa-app-store-numero-total-aplicativos-desenvolvedores.htm>. Acesso: abril/2017.
- Aplicações sensíveis ao contexto. Disponível em [http://www2.dbd.puc-rio.br/pergamum/tesesabertas/0511040\\_07\\_cap\\_02.pdf](http://www2.dbd.puc-rio.br/pergamum/tesesabertas/0511040_07_cap_02.pdf). Acesso: março/2018.
- Ryon, B. O excesso de Informação. A Neurose do século XXI. Disponível em <http://www.mettodo.com.br/pdf/O%20Excesso%20de%20Informacao.pdf>. Acesso: junho/2018.
- Rodrigue, C e Blattman, U. Gestão da informação e a importância do uso de fontes de informação para geração de conhecimento. Disponível em <http://www.scielo.br/pdf/pci/v19n3/a02v19n3.pdf>. Acesso: junho/2018.

- Brown, P. The stick-e document: a framework for creating contex-aware applications. Vol. 8(2 e 3), 259-172. Junho 1995. Disponível em <http://cajun.cs.nott.ac.uk/compsci/epo/papers/volume8/issue2/2point1.pdf>. Acesso: julho/2018.
- Gellersen, H. e Handheld and Ubiquitous Computing. 1999. Disponível em <https://books.google.com.br/books?id=WrcSVBYEC6YC&printsec=frontcover&hl=pt-BR#v=onepage&q&f=false>. Acesso: julho/2018.
- The Nature and Importance of Innovation. Disponível em <http://assets.press.princeton.edu/chapters/s9221.pdf>. Acesso: agosto/2018.
- Ogliari, R. e Brito, R. Android - Do Básico ao Avançado. 2014.
- Ionic Platform Documentation. Disponível em <https://ionicframework.com/docs/>. Acesso: setembro/2018.
- What Is A Non Relational Database. Disponível em <https://www.mongodb.com/scale/what-is-a-non-relational-database>. Acesso: outubro/2018.
- Creating a Firebase Application. Disponível em: [https://ionicthemes.com/tutorials/about/building-a-ionic-firebase-app-step-by-step#discussion\\_thread](https://ionicthemes.com/tutorials/about/building-a-ionic-firebase-app-step-by-step#discussion_thread). Acesso: outubro/2018.
- Tohidi, H. e Jabbari, M. The important of Innovation and its Crucial Role in Growth, Survival and Success of Organizations. Disponível em [https://ac.els-cdn.com/S221201731200117X/1-s2.0-S221201731200117X-main.pdf?\\_tid=c0e972be-e877-48a3-8e09-cc813cbf48df&acdnat=1543187951\\_5f1fc30bd7e854af63eb1d70ce3fa0e6](https://ac.els-cdn.com/S221201731200117X/1-s2.0-S221201731200117X-main.pdf?_tid=c0e972be-e877-48a3-8e09-cc813cbf48df&acdnat=1543187951_5f1fc30bd7e854af63eb1d70ce3fa0e6). Acesso: novembro/2018.