

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO



AMÓS YI CONG LU ZHENG – 7629737

KLEBER HIRO YOKOYAMA – 7629866

**PCS2502 – PROJETO DE FORMATURA II:
SMART SHOWER**

SÃO PAULO

2016

AMÓS YI CONG LU ZHENG – 7629737
KLEBER HIRO YOKOYAMA – 7629866

PCS2502 – PROJETO DE FORMATURA II:
SMART SHOWER

Título

Trabalho de conclusão de curso (TCC),
apresentado à banca examinadora da Escola
Politécnica da USP para obtenção do grau de
habilitação em Engenharia Elétrica – Ênfase
em Computação, sob orientação do Prof. Dr.
André Riyuiti Hirakawa.

SÃO PAULO
2016

RESUMO

O *Smart Shower* pode ser considerado uma parte do conceito *Smart Home*, termo utilizado para designar a automação de processos e serviços em uma residência com base nos conceitos de Internet das Coisas. No presente caso, o foco do projeto é a automação do processo de aquecimento de água do chuveiro de uma maneira inteligente. Há algumas pesquisas e implementações com o mesmo codinome “*Smart Shower*”, porém há um problema comum entre eles: em nenhum deles há uma lógica inteligente para a automação do cálculo da temperatura da água que o usuário desejará para tomar banho. Este trabalho propõe uma arquitetura de sistema para atingir tal objetivo, utilizando adaptações da infraestrutura já existente na maioria das casas com o auxílio de tecnologias emergentes (Internet das coisas, Fog computing e sistema operacional Android). O objetivo de utilizar essa arquitetura é reduzir a complexidade através da abstração e separação de requisitos do projeto e, ao mesmo tempo, diminuir os custos de implementação do projeto. A presente solução propõe um sistema inteligente para aquecimento de água do chuveiro, identificando os requisitos gerais do sistema, do hardware e do software, assim como os possíveis casos de uso, objetivando elaborar uma implementação. Tendo em vista a disponibilidade dos diferentes tipos de tecnologias, o documento especifica possíveis implementações e seleciona a mais adequada, focando na disponibilidade de recursos e a adequação aos requisitos do sistema. A implementação do modelo escolhido é apresentada, assim como os resultados dos testes. Não se visa aqui a apresentação ou definição de métricas específicas, sendo os resultados apresentados do tipo qualitativo.

Palavras-chave: *Smart Shower, Smart Home, Internet das Coisas, Fog computing, Android.*

ABSTRACT

Smart Shower can be considered a part of *Smart Home*, which is a term used for home process and service automation through concepts of Internet of Things (IoT). In our case, the goal of this project is to automate the water heating process in a *smart* way. There exists research and implementations with the same code-name “Smart Shower”, but there is a common problem among them: none of them has a smart logic to automate water temperature prediction for the shower. This project proposes a system architecture to achieve this goal, incorporating emerging technology (Internet of Things, Fog computing and Android apps) into existing infrastructure of most homes. The architecture of the system is designed to reduce complexity through abstraction and project requirements separation, as well as reduce project implementation costs. Our solution proposes a smart system for shower water heating. It identifies system, hardware and software general requirements, as well as possible use cases, with the goal of implementation. Considering the availability of different technologies, we specify possible implementations and choose the most appropriate according to the availability of resources and the system requirements. We then present our implementation together with the test results. We didn't define specific metrics, so the results are presented mostly in a qualitative way.

Keywords: *Smart Shower, Smart Home, Internet of Things, Fog Computing, Android.*

LISTA DE FIGURAS

Figura 1 - Diagrama de classe do aplicativo Android	22
Figura 2 - Processo de configurar a temperatura (1).....	23
Figura 3 - Processo de configurar a temperatura (2).....	24
Figura 4 - Arquitetura do sistema Smart Shower.....	30
Figura 5 - Rede neural artificial direta	31
Figura 6 – Diagrama de um sistema simples de aquecimento elétrico (CHEGG STUDY, n.d.).....	35
Figura 7 – Funcionamento de um aquecedor a gás de passagem (WILSON, n.d.) ..	37
Figura 8 – Um chuveiro elétrico convencional.....	39
Figura 9 – Termômetro digital 1-Wire DS18B20.....	40
Figura 10 – Pinagem do ESP8266 (ESP-01)	41
Figura 11 – Relé de estado sólido Fotek SSR-40 DA	42
Figura 12 – Pinagem do módulo FT232RL	43
Figura 13 – Conexão entre ESP8266 e FT232RL para a substituição de firmware ..	43
Figura 14 – Conexão entre ESP8266 e FT232RL para subir o código ao módulo....	44
Figura 15 - Screenshot do programa nodemcu-flasher	46
Figura 16 – Um script Lua simples para o módulo ESP8266 para piscar um LED....	47
Figura 17 – IDE ESPlorer.....	47
Figura 18 – Diagrama de fluxo do código no módulo ESP8266	48
Figura 19 - Conexão HTTP com a API do Open Weather Map (1)	49
Figura 20 - Conexão HTTP com a API do Open Weather Map (2)	49
Figura 21 - Exemplo de login com uma conta Facebook	50
Figura 22 - Exemplo de retomada de um login.....	50
Figura 23 - Ambiente de programação do Android Studio	52
Figura 24 - Emulador do Android Studio	52
Figura 25 - Layout da tela ScreeLogin	53
Figura 26 - Layout da tela ScreenRegister.....	53
Figura 27 - Layout da tela ScreenMain	54
Figura 28 - Layout da tela WeatherScreen.....	54
Figura 29 - Fórmula Heat index implementada no Android Studio	56
Figura 30 - Fórmula Wind chill implementada no Android Studio.....	56
Figura 31 - Exemplo do algoritmo de média ponderada (1)	58

Figura 32 - Exemplo do algoritmo de média ponderada (2)	58
Figura 33 - tblogin	59
Figura 34 - tbdadostemperatura	59
Figura 35 - Dados coletados dos banhos	60
Figura 36 - Gráfico dos dados coletados dos banhos	61
Figura 37 - Diagrama de conexão do ESP8266	63
Figura 38 - Teste do módulo ESP8266 e periféricos	63
Figura 39 - Registrando uma nova conta	64
Figura 40 - Banco de dados após registro da nova conta	64
Figura 41 - Login com conta “normal”	65
Figura 42 - Registro de uma conta Facebook	65
Figura 43 - Login com conta Facebook	66
Figura 44 - Testes de integração.....	67
Figura 45 - Configurando manualmente a localização	67
Figura 46 - Acessando os dados do módulo via URL	68
Figura 47 - Verificando a localização	68
Figura 48 - Inserindo manualmente uma temperatura	69
Figura 49 - Verificando a temperatura alvo	69
Figura 50 - Estrutura da divisão do trabalho do projeto I.....	78
Figura 51 - Tela de login	79
Figura 52 - Botão Create account	79
Figura 53 - Tela de registro	79
Figura 54 - Configuração manual da localização	80
Figura 55 - Temperatura da água estimada	80
Figura 56 - Campos para mudar a temperatura	80
Figura 57 - Apparent temperature	81

LISTA DE TABELAS

Tabela 1 - Requisitos funcionais do software	18
Tabela 2 - Requisitos não-funcionais do software	21
Tabela 3 - Exemplo de tabela dinâmica	33
Tabela 4 – Descrição da pinagem do ESP8266 (ESP-01)	41
Tabela 5 – Descrição da pinagem do módulo FT232RL	43
Tabela 6 – Custos estimados de hardware do projeto	44

ÍNDICE DE EQUAÇÕES

Equação 1 - Fórmula Heat Index (SERVICE).....	26
Equação 2 - Ajustamento do Heat Index para RH < 13% e 80 < T < 112 (SERVICE)	26
Equação 3 - Ajustamento do Heat Index para RH > 85% e 80 < T < 87 (SERVICE) 27	27
Equação 4 - Fórmula Heat Index para T < 80 (CENTER)	27
Equação 5 - Fórmula Wind Chill (SERVICE).....	27
Equação 6 - Modelo de regressão linear simples.....	33
Equação 7 - Média ponderada entre a temperatura antiga da água e a temperatura nova	57

SUMÁRIO

1.	INTRODUÇÃO.....	11
1.1.	Objetivo	11
1.2.	Motivação.....	11
1.3.	Aspectos conceituais	12
1.3.1.	Internet das coisas (IoT)	12
1.3.2.	Sensação térmica	13
2.	PROJETO.....	14
2.1.	Especificações Do Projeto	15
2.1.1.	Casos de uso do sistema.....	15
2.1.1.1.	Caso de uso 1: Controle automático da temperatura.....	15
2.1.1.2.	Caso de uso 2: Controle manual da temperatura	15
2.1.1.3.	Caso de uso 3: Desligamento do sistema.....	16
2.1.2.	Requisitos funcionais do sistema.....	16
2.1.3.	Requisitos não funcionais do sistema	16
2.2.	Hardware.....	17
2.2.1.	Requisitos funcionais do hardware	17
2.2.2.	Requisitos não funcionais do hardware	17
2.3.	Software	18
2.3.1.	Requisitos funcionais do software.....	18
2.3.2.	Requisitos não-funcionais do software.....	21
2.3.3.	Diagrama de classe do software	22
2.3.4.	Diagrama do processo do software.....	22
2.3.5.	Algoritmo para estimar a temperatura da água.....	25
2.3.5.1.	Temperatura aparente	25
2.3.5.2.	Fórmula Heat Index.....	26
2.3.5.3.	Fórmula Wind Chill.....	27

2.3.5.4.	Fórmula Ambient Air Temperature	28
2.3.5.5.	Temperatura máxima/mínima	28
2.3.5.6.	Outros fatores que podem afetar a preferência de temperatura	28
2.4.	Arquitetura Do Sistema	29
3.	POSSÍVEIS ALGORITMOS INTELIGENTES	31
3.1.	Redes neurais artificiais	31
3.2.	Régressão linear	33
3.3.	Tabela dinâmica	33
4.	POSSÍVEIS ESCOLHAS DE HARDWARE	35
4.1.	Chuveiro elétrico	35
4.2.	Chuveiro a gás	36
4.3.	Segurança	37
5.	PROVA DE CONCEITO	39
5.1.	Componentes de hardware do sistema	39
5.1.1.	Chuveiro elétrico convencional	39
5.1.2.	Sensor de temperatura DS18B20	39
5.1.3.	Módulo ESP8266	40
5.1.4.	Relé de estado sólido	42
5.1.5.	Módulo FT232RL	42
5.1.6.	Roteador	44
5.1.7.	Dispositivo móvel Android	44
5.1.8.	Custos estimados de hardware	44
5.2.	Software - Web Server	45
5.3.	Software – Aplicação Android	49
5.3.1.	Open Weather Map API	49
5.3.2.	Login Facebook para Android	49
5.3.3.	Android Studio	50

	10	
5.3.4.	Algoritmo - Apparent air temperature	56
5.3.5.	Algoritmo – Temperatura da água do banho.....	57
5.4.	Software – Banco de dados My SQL	58
6.	TESTES E RESULTADOS	62
6.1.	Testes preliminares.....	62
6.1.1.	ESP8266.....	62
6.1.2.	Aplicação Android	64
6.1.3.	Integração	66
6.2.	Testes finais	69
7.	CONCLUSÃO	71
7.1.	Trabalhos Futuros	71
	REFERÊNCIAS.....	73
	APÊNDICES.....	78
	Apêndice I: Estrutura De Divisão Do Trabalho.....	78
	Apêndice II: Manual Do Usuário	79

1. INTRODUÇÃO

1.1. Objetivo

O projeto SMART Heater tem como objetivo o desenvolvimento de um sistema de controle automático de aquecimento de água baseado nos conceitos da Internet das Coisas (IoT). Os diferenciais de nosso projeto são:

- Aumentar o bem estar do nosso usuário através do controle automático do aquecimento de água.
- Gerar praticidade para o controle de temperatura.
- Uso de tecnologias emergentes (Internet das Coisas, Desenvolvimento Android).
- Capacidade de analisar fatores que influenciam a real temperatura que o usuário está sentindo (sensação térmica).

1.2. Motivação

A motivação do nosso projeto é baseada totalmente no usuário final. Um sistema de aquecimento de água comum possui características que geram incômodo no dia a dia do consumidor e esse sistema visa acabar com a maioria dos problemas encontrados neles. Pode-se destacar as seguintes motivações:

- Aumento do conforto – Ao se atingir a temperatura ideal da água, o usuário sempre irá tomar banho na temperatura que acha agradável, sem passar calor ou frio durante o processo;
- Usabilidade – É possível ajustar a temperatura da água facilmente e de forma muito mais prática do que é atualmente;
- Economia – O sistema é projetado para estimar especificamente a temperatura ideal para tomar banho no momento, evitando assim aquecimento desnecessário em casos em que a temperatura da água é maior do que a ideal para o usuário;

- Inteligente (*smart*) – O sistema será capaz de prever a temperatura desejada do usuário com base em condições climáticas no momento do banho.

1.3. Aspectos conceituais

1.3.1. Internet das coisas (IoT)

O advento da Internet mudou completamente os meios de comunicação não só entre pessoas, mas também entre objetos. Assim como utilizamos a internet para nos comunicar com os outros, houve grande expansão nos meios de comunicação entre dispositivos. Para se ter uma ideia, hoje a população mundial é de 7 bilhões e existem 25 bilhões de dispositivos conectados. Para não nos confundir com os termos é necessário entender a diferença entre a Internet, a World Wide Web (ou Web) e a *Internet of Things (IoT)*:

- Internet: Camada ou rede composta por switches, roteadores e outros equipamentos, com função de transportar informações de um ponto a outro de forma rápida, confiável e segura.
- Web: Camada de aplicação que opera sobre a Internet, com função de oferecer uma interface que transforme as informações que fluem pela Internet em algo utilizável.
- IoT: Coleção livre de redes distintas e criadas para certa finalidade, a IoT pode ser vista como uma rede de redes, com função de coletar informações, transmitir, analisar e distribuir dados em grande escala. Pode-se dizer que a IoT é uma evolução da Internet.

Desse modo, o conceito de Internet das Coisas viabiliza nosso sistema a coletar dados em tempo real da Internet, analisar esses dados e distribuir para o sistema de aquecimento de água, assim como a comunicação interna entre todos os dispositivos do sistema.

1.3.2. Sensação térmica

Apesar de frequentemente nos basearmos na temperatura atual para nos orientarmos se está frio ou quente, esta prática não é muito confiável. O conceito de sensação térmica é utilizado para entendermos melhor como fatores externos podem influenciar a real temperatura sentida pelo corpo humano. Para isso é necessário entender os seguintes conceitos:

- Temperatura: Medida da energia cinética média das partículas de um sistema. Em termos mais práticos, é a grandeza física que o termômetro registra, medido sempre na sombra para não ter influência da incidência solar.
- Umidade do ar: Quantidade de vapor de água na atmosfera. Quanto mais alta a umidade do ar, maior a dificuldade do processo de transpiração ocorrer e consequentemente maior é a sensação térmica. Em temperaturas mais baixas, a umidade do ar tende a influenciar cada vez menos, devido à diminuição da transpiração humana (o corpo tende a reter calor no frio).
- Velocidade do vento: Deslocamento do ar, que migra de regiões de alta pressão atmosférica para pontos com pressões atmosféricas menores. É um dos principais fatores que influenciam na sensação térmica, pois conforme a velocidade do vento aumenta, aumenta a taxa de evaporação da pele e consequentemente aumenta a retirada de calor.
- Dia/horário: Dependendo do período do dia (por exemplo, antes de deitar ou depois de se levantar), nosso corpo pode ter diferentes percepções da temperatura.

Devido ao conhecimento superior das instituições e empresas especializadas em estimar o tempo e o clima em relação ao conhecimento do grupo, iremos adotar que as aproximações feitas por essas empresas são mais condizentes com a real sensação térmica do que se nós fossemos calcular por conta própria.

2. PROJETO

O advento da internet das coisas permite cada vez mais integração de equipamentos e objetos que agem de forma autônoma para atingir um fim comum. O projeto é uma aplicação dessa tecnologia na área de aquecimento de água do banho, e consiste em desenvolver um sistema automatizado e inteligente em cima da infraestrutura física existente dos chuveiros, que consiga prever a temperatura ideal que o usuário deseja para tomar banho. Isso é feito analisando-se um histórico de temperaturas de água que o usuário ajusta no sistema em relação à temperatura ambiente no momento do banho, obtida da internet. O produto esperado é um sistema que consiga prever e ajustar automaticamente a temperatura da água conforme as preferências do usuário, da melhor forma possível.

O projeto envolverá os seguintes tópicos:

- Aspectos teóricos (IoT, sensação térmica)
- Especificação do projeto (Requisitos do sistema, do hardware e do software)
- Informações sobre o funcionamento de chuveiros
- Implementação usando o módulo ESP8266
- Desenvolvimento na plataforma Android/Web

Nesse projeto, será implementado um algoritmo do cálculo da sensação térmica de acordo com as condições climáticas. Será utilizado um SoC (“System on a Chip”), Wi-fi e uma plataforma Android para implementar o Sistema. Duas possíveis implementações serão estudadas para depois selecionar a mais adequada para implementar a prova de conceito do projeto. Os detalhes da implementação serão descritos posteriormente.

2.1. Especificações Do Projeto

2.1.1. Casos de uso do sistema

2.1.1.1. Caso de uso 1: Controle automático da temperatura

A temperatura da água irá ser automaticamente configurada.

1. O usuário liga o registro da água.
2. O sistema é ativado.
3. O sistema obtém dados atuais da temperatura, vento e umidade da Internet.
4. Sistema estima temperatura desejada pelo usuário com base nos dados que possui.
5. Finalmente, o sistema ajusta a temperatura da água automaticamente, de acordo com a temperatura estimada.

2.1.1.2. Caso de uso 2: Controle manual da temperatura

Caso o usuário não goste da temperatura estimada, ele poderá ajustar manualmente a temperatura.

1. O sistema utiliza a temperatura da água estimada.
2. Usuário insere nova temperatura pela aplicação Android.
3. Sistema guarda o valor da temperatura que o usuário colocou, juntamente com outras informações (como dados do clima, dia e horário).
4. Sistema ajusta o calor da água até obter a temperatura inserida.

2.1.1.3. Caso de uso 3: Desligamento do sistema

Caso o usuário não esteja utilizando o aquecedor, o sistema precisa desativar-se.

1. Usuário termina de utilizar o chuveiro.
2. Sistema identifica que não há mais passagem de água e se desativa.

2.1.2. Requisitos funcionais do sistema

- RF01 - O sistema deve possuir conexão com a Internet.
- RF02 - O sistema deve permitir controle automático da temperatura.
- RF03 - O sistema deve permitir ajuste manual da temperatura.
- RF04 - O Sistema deve permitir visualização da temperatura atual.
- RF05 - O sistema deve ser inteligente e ser capaz de realizar estimativas de temperatura com base em dados armazenados e dados obtidos no momento do banho.
- RF06 - O sistema deve ser capaz de se desligar automaticamente quando não estiver em uso.

2.1.3. Requisitos não funcionais do sistema

- RNF01 - A conexão entre o sistema e o usuário deve ser sem-fio.
- RNF02 - O custo máximo de produção do sistema não pode ser superior a R\$ 200,00.
- RNF03 - O sistema deve ser intuitivo e operado sem que o usuário perceba o sistema computacional que o implementa.
- RNF04 - O sistema, tanto em modo automático ou manual, deve possuir um tempo de resposta inferior a 20 segundos.
- RNF05 - Portabilidade: A aplicação deve funcionar em dispositivo móvel.

- RNF06 - Eficiência: O sistema deve possuir eficiência energética maior ou igual ao chuveiro que foi modificado.
- RNF07 - Facilidade de uso: O usuário deve ser capaz de utilizar o sistema sem ter tido treinamento específico para isso.

2.2. Hardware

Os requisitos de hardware serão classificados em requisitos funcionais e requisitos não-funcionais.

2.2.1. Requisitos funcionais do hardware

- RF01 - O chuveiro inteligente deve suportar conexão com a internet sem fio.
- RF02 - O chuveiro inteligente deve ser capaz de controlar a temperatura da água de acordo com os comandos recebidos de forma sem fio.
- RF03 - O dispositivo Android deve ser capaz de mostrar a temperatura atual da água para o usuário.
- RF04 - O dispositivo Android deve possuir processamento mínimo necessário para que o aplicativo instalado possa executar normalmente algoritmos necessários para a estimativa da temperatura da água do banho.

2.2.2. Requisitos não funcionais do hardware

- RNF01 - O hardware deve ser interoperável com aparelhos móveis.
- RNF02 - O hardware deve permitir leituras da temperatura atual da água com tempo de resposta inferior a um segundo.
- RNF03 - O hardware deve possuir eficiência energética igual ou melhor do que um sistema de aquecimento convencional.

2.3. Software

O software especificado nessa seção é uma parte crucial do projeto. Ele é responsável por conectar todo o sistema: O banco de dados, o web service de login, o web service do clima e o sistema de aquecimento. Adicionalmente, irá proporcionar uma interface e uma configuração intuitiva para o usuário, sem necessidade de treinamentos prévios para utilizar o sistema.

É importante reforçar que a arquitetura utilizará *fog computing*, com o processamento de dados descentralizado nos aparelhos móveis dos usuários. Consequentemente, isso irá reduzir a quantidade de dados necessária para ser processada, analisada e armazenada na nuvem, aumentando assim a eficiência do sistema.

Os requisitos de software serão classificados em requisitos funcionais e requisitos não-funcionais.

2.3.1. Requisitos funcionais do software

A Tabela 1 apresenta todos os requisitos funcionais e suas respectivas prioridades.

Tabela 1 - Requisitos funcionais do software

RF	Nome	Prioridade
RF01	Ligar o sistema de aquecimento	Alta
RF02	Desligar o sistema de aquecimento	Alta
RF03	Registrar um novo usuário	Média
RF04	Logar como um usuário normal	Média
RF05	Logar como um usuário Facebook	Alta
RF06	Obter dados do usuário pelo Facebook	Alta
RF07	Obter manualmente a localização do usuário	Média
RF08	Obter dados do clima	Alta
RF09	Mostrar dados do clima	Baixa

RF10	Calcular temperatura aparente	Alta
RF11	Estimar temperatura da água	Alta
RF12	Setar temperatura escolhida pelo usuário	Alta
RF13	Setar temperatura estimada pelo sistema	Alta
RF14	Solicitação de utilização de dados do Facebook	Alta

RF01 - Ligar o sistema de aquecimento

Prioridade: Alta

O aplicativo terá que estabelecer uma conexão com o sistema de aquecimento e ter acesso para ligá-lo.

RF02 - Desligar o sistema de aquecimento

Prioridade: Alta

O aplicativo terá que estabelecer uma conexão com o sistema de aquecimento e ter acesso para desligá-lo.

RF03 - Registrar um novo usuário

Prioridade: Média

Preferencialmente, o login será feito via Facebook. Entretanto, o sistema deve permitir que o usuário registre uma nova conta independente da conta de Facebook.

RF04 - Logar como um usuário normal

Prioridade: Média

Caso o usuário opte por registrar uma conta independente da conta de Facebook, ele poderá logar utilizando essa conta.

RF05 - Logar como um usuário Facebook

Prioridade: Alta

O principal tipo de acesso será pela conta de Facebook do usuário. O aplicativo terá um botão para o usuário logar via Facebook.

RF06 - Obter dados do usuário pelo Facebook

Prioridade: Alta

O acesso via Facebook será fácil e prático. Adicionalmente, irá proporcionar ao sistema dados relevantes do usuário para proporcionar uma melhor estimativa da temperatura da água.

RF07 - Obter manualmente a localização do usuário

Prioridade: Média

Se o usuário optar por logar utilizando uma conta independente da conta de Facebook, ou caso não tenha informação da localização nos dados da conta de Facebook, ele irá precisar inserir manualmente sua localização.

RF08 - Obter dados do clima

Prioridade: Alta

Quando o usuário entrar no sistema, ele automaticamente obterá os dados do clima baseado na localização do chuveiro.

RF09 - Mostrar dados do clima

Prioridade: Baixa

Como uma função adicional, o sistema deve permitir que o usuário visualize os dados do clima local (temperatura aparente, velocidade do vento, umidade).

RF10 - Calcular temperatura aparente

Prioridade: Alta

O sistema deve calcular automaticamente a temperatura aparente, baseado na temperatura do ar, velocidade do vento e umidade.

RF11 – Estimar temperatura da água

Prioridade: Alta

O algoritmo inteligente deve utilizar a temperatura aparente, juntamente com os dados anteriores do usuário, para estimar a temperatura que o usuário deseja para tomar banho.

RF12 - Setar temperatura escolhida pelo usuário

Prioridade: Alta

O usuário deve poder setar a temperatura manualmente caso ele não goste da temperatura estimada pelo sistema. Esse dado será muito importante para o sistema ajustar a preferência do usuário em futuras estimativas.

O aplicativo enviará a temperatura inserida manualmente para o sistema de aquecimento.

RF13 - Setar temperatura estimada pelo sistema

Prioridade: Alta

Caso o usuário não sete manualmente uma temperatura, o aplicativo reconhecerá que a temperatura estimada está apropriada e manterá a água nessa temperatura.

RF14 – Solicitação de utilização de dados do Facebook

O Sistema deve solicitar permissão de acesso aos dados da conta Facebook do usuário, para evitar ser invasivo.

2.3.2. Requisitos não-funcionais do software

A Tabela 2 representa todos os requisitos não-funcionais do sistema e suas prioridades.

Tabela 2 - Requisitos não-funcionais do software

RNF	Name	Prioridade
RNF01	Interface Android intuitiva	Alta
RNF02	Tempo de resposta inferior a 20 segundos	Alta
RNF03	Conexão sem-fio	Alta
RNF04	Mensagens claras e palavras simples	Alta

RNF01 - Interface Android intuitiva

Prioridade: Alta

O aplicativo Android desenvolvido não exigirá treinamento específico para ser utilizado. Por exemplo: Botões grandes autoexplicativos, reduzindo a necessidade de ter textos explicando o contexto.

RNF02 - Tempo de resposta inferior a 20 segundos

Prioridade: Alta

Se o tempo de resposta for muito alto, o usuário se sentirá frustrado e a aplicação não terá atingido o seu objetivo de oferecer mais conforto e praticidade para o usuário; consequentemente, a resposta deve ser inferior a 20 segundos. Adicionalmente, para garantir eficiência energética e da água, o tempo de resposta não pode ser muito alto.

RNF03 - Conexão sem-fio

Prioridade: Alta

O aparelho móvel do usuário deve estar conectado na mesma rede sem-fio do sistema de aquecimento para garantir a conexão entre eles.

RNF04 - Mensagens claras e palavras simples

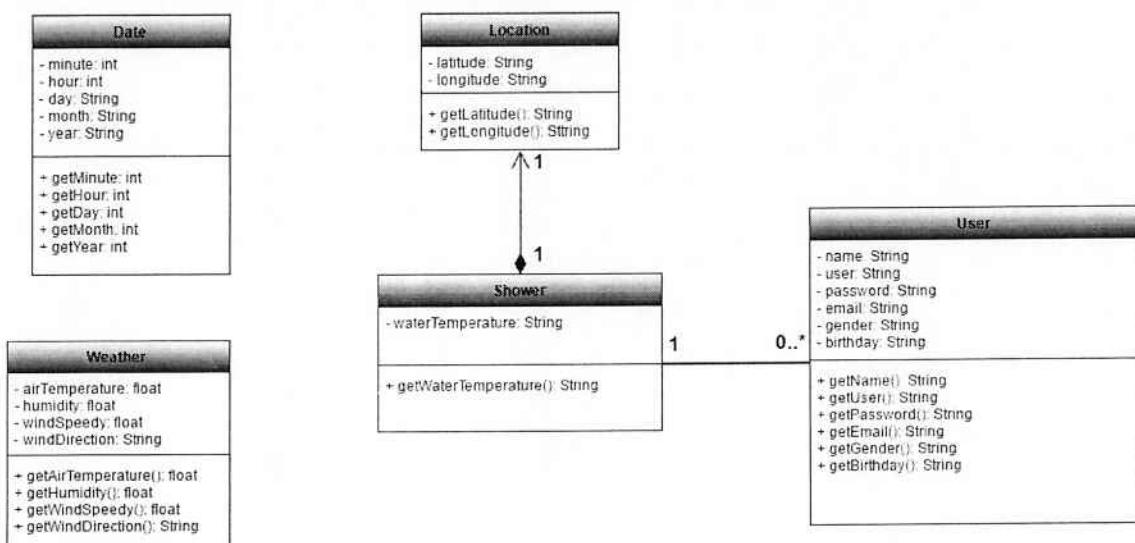
Prioridade: Alta

O sistema deve apresentar mensagens e vocabulário simples, pois um dos objetivos do sistema é simplificar a vida do usuário.

2.3.3. Diagrama de classe do software

A Figura 1 representa o diagrama da estrutura da aplicação Android.

Figura 1 - Diagrama de classe do aplicativo Android



2.3.4. Diagrama do processo do software

A Figura 2 e a Figura 3 mostram em detalhes o fluxo do processo de configurar a temperatura da água.

Figura 2 - Processo de configurar a temperatura (1)

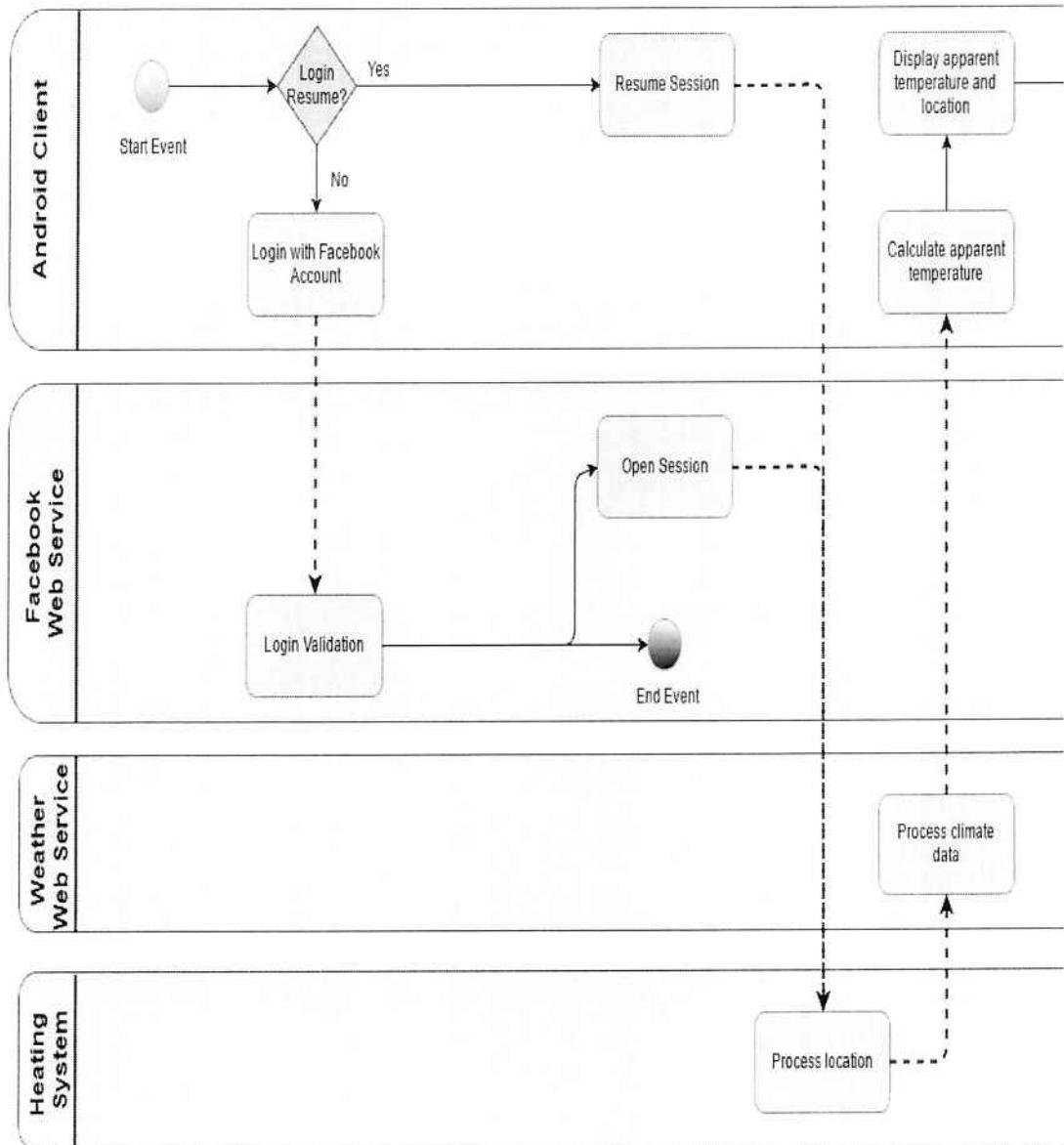
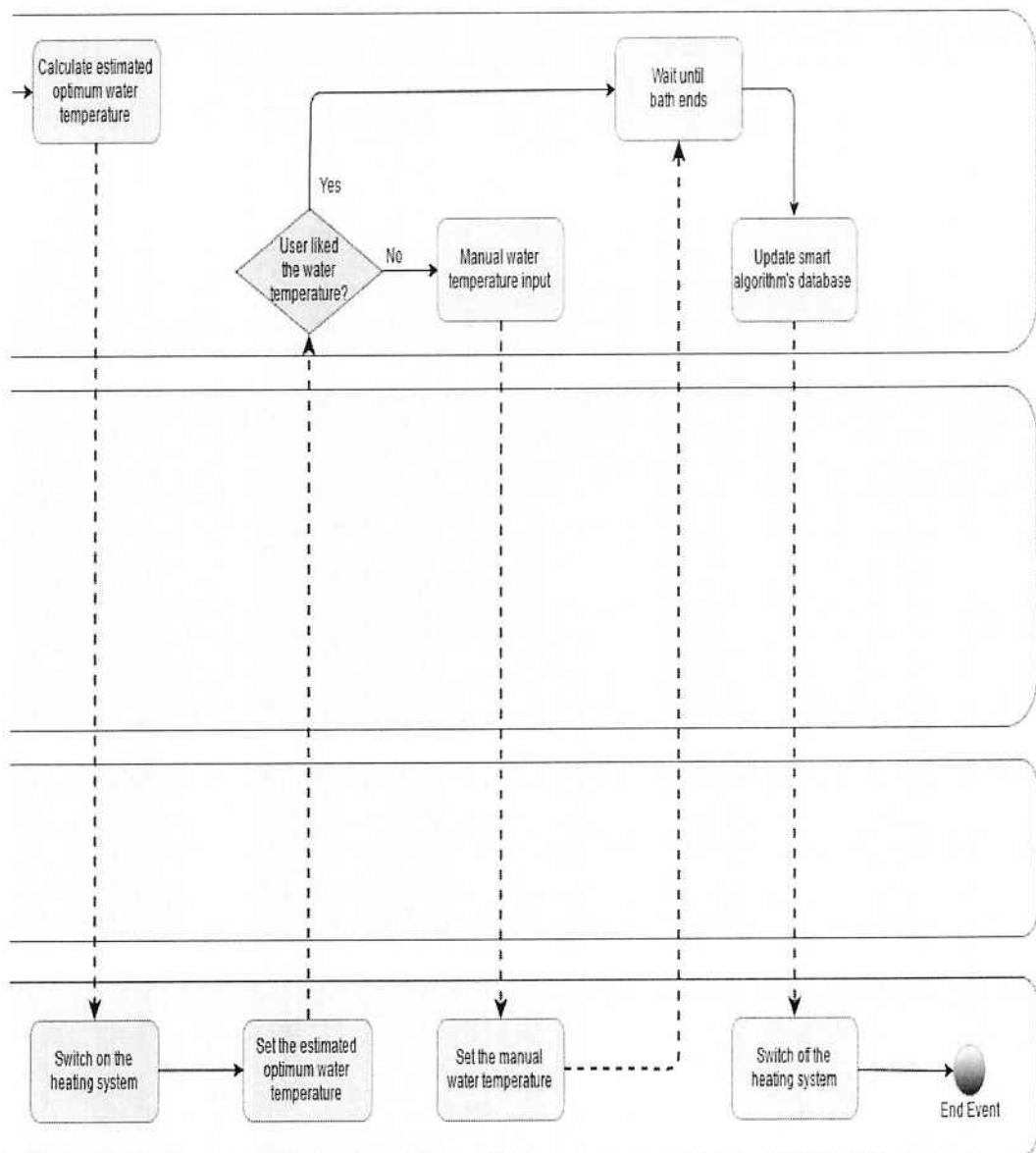


Figura 3 - Processo de configurar a temperatura (2)



O processo de configurar a temperatura da água começa quando o usuário logar com a sua conta Facebook. Se o login está sendo resgatado, a sessão é resumida, caso contrário, o web service do Facebook irá validar o login e abrir a sessão do usuário. O sistema de aquecimento é responsável por fornecer a localização e o web service do clima é responsável por fornecer os dados do clima de acordo com a localização do sistema de aquecimento.

A aplicação Android irá calcular a temperatura aparente e mostrar tanto a temperatura aparente quanto a localização. Adicionalmente, será calculado a temperatura ótima estimada de acordo com as preferências do usuário.

Após os cálculos, o sistema de aquecimento será ligado para que ele ajuste a temperatura da água de acordo com a temperatura enviada. A aplicação Android será

responsável por verificar se o usuário inseriu manualmente uma temperatura e o sistema de aquecimento será responsável por ajustar a temperatura para esse valor.

No final do banho, a aplicação Android irá atualizar o banco de dados do sistema para usos futuros. Quanto mais o usuário utilizar o sistema, mais customizado ele ficará.

O processo acaba com o desligamento do sistema de aquecimento.

2.3.5. Algoritmo para estimar a temperatura da água

Para obter uma boa estimativa da temperatura da água, é extremamente importante saber qual é a temperatura aparente do ar e quais são as principais variáveis que influenciam no cálculo. Nesse projeto, o algoritmo utilizará três tipos diferentes de fórmula para o cálculo: *Heat Index*, *Wind Chill* e *Ambient Air Temperature*. Os próximos tópicos explicam o porquê é interessante utilizá-los e em qual ocasião usar cada um deles.

2.3.5.1. Temperatura aparente

Descrição proposta pelo National Digital Forecast Database (NDFD):

"The perceived temperature in degrees Fahrenheit derived from either a combination of temperature and wind (Wind Chill) or temperature and humidity (Heat Index) for the indicated hour. When the temperature at a particular grid point falls to 50 F or less, wind chill will be used for that point for the Apparent Temperature. When the temperature at a grid point rises above 80 F, the heat index will be used for Apparent Temperature. Between 51 and 80 F, the Apparent Temperature will be the ambient air temperature."

2.3.5.2. Fórmula Heat Index

Alta umidade combinada com temperaturas altas reduz a habilidade do corpo de se resfriar, aumentando assim o risco de exaustão por calor, insolação, e outros tipos de danos decorrentes do calor. A fórmula *Heat Index* é uma estimativa da temperatura que afetaria o corpo em condições de umidade normais (em torno de 20%). Por exemplo, se a atual temperatura é 100°F com 40% de umidade relativa, o *Heat Index* é 110°F, ou seja, a temperatura aparente para o corpo é de 110°F.

O cálculo do *Heat Index* é o resultado de um refinamento obtido por regressões lineares múltiplas feitas por Lans P. Rothfusz e descrito em 1990 pela National Weather Service (NWS) Technical Attachment (SR 90-23). A regressão de Rothfusz pode ser representada pela Equação 1:

Equação 1 - Fórmula Heat Index (SERVICE)

$$\begin{aligned} HI = & -42.379 + 2.04901523 * T + 10.14333127 * RH - .22475541 * T \\ & * RH - .00683783 * T * T - .05481717 * RH * RH + .00122874 * T \\ & * T * RH + .00085282 * T * RH * RH - .00000199 * T * T * RH * RH \end{aligned}$$

Onde:

- T – Temperatura em graus Fahrenheit (°F).
- RH – Umidade relativa em porcentagem.
- HI – *Heat Index* expressado como a temperatura aparente em graus Fahrenheit (°F).

Se o RH for menos que 13% e a temperatura estiver entre 80 e 112 graus Fahrenheit, então o HI deverá ser subtraído pelo ajustamento apresentado pela Equação 2:

Equação 2 - Ajustamento do Heat Index para RH < 13% e 80 < T < 112 (SERVICE)

$$ADJUSTMENT = [(13 - RH)/4] * SQRT{[17 - ABS(T - 95.)]/17}$$

Onde:

- ABS – Valor absoluto da expressão.
- SQRT – Raíz quadrada da expressão.

Por outro lado, se o RH for maior que 85% e a temperatura estiver entre 80 e 87 graus Fahrenheit, então o HI deverá ser somado pelo ajustamento apresentado pela Equação 3:

Equação 3 - Ajustamento do Heat Index para RH > 85% e 80 < T < 87 (SERVICE)

$$\text{ADJUSTMENT} = [(RH - 85)/10] * [(87 - T)/5]$$

A regressão calculada por Rothfusz não é apropriada quando as condições de temperatura e umidade geram um Heat Index abaixo de 80 graus Fahrenheit. Nesses casos, uma fórmula mais apropriada é aplicada para calcular valores mais consistentes com os resultados obtidos por Steadman é apresentada na Equação 4:

Equação 4 - Fórmula Heat Index para T < 80 (CENTER)

$$HI = 0.5 * \{T + 61.0 + [(T - 68.0) * 1.2] + (RH * 0.094)\}$$

Na prática, essa fórmula é computada primeiramente. Se o valor do *Heat Index* for 80 graus Fahrenheit ou maior, então a equação de regressão juntamente com o ajuste necessário são calculados.

A regressão de Rothfusz não é válida para temperaturas extremas e condições de umidade relativas além do intervalo analisado por Steadman.

2.3.5.3. Fórmula Wind Chill

Uma porção do resfriamento humano é causado pelo movimento do ar. O movimento do ar acelera a taxa de transferência de calor do corpo humano para a atmosfera circundante, especialmente quando as temperaturas estão abaixo de 7°C (45°F), modelo representado pela Equação 5.

Equação 5 - Fórmula Wind Chill (SERVICE)

$$\begin{aligned} \text{Wind Chill} = & 35.74 + (0.6215 \times T) - (35.75 \times \text{Wind}_{sfc}^{0.16}) \\ & + (0.4275 \times T \times \text{Wind}_{sfc}^{0.16}) \end{aligned}$$

Onde:

- Wind Chill - Temperatura aparente em graus Fahrenheit (°F).
- Wind_{sfc} - Velocidade do vento em milhas por hora (mph).
- T - Temperatura do ar em graus Fahrenheit (°F)

2.3.5.4. Fórmula Ambient Air Temperature

Entre 51 e 80 graus Fahrenheit, nem a velocidade do vento, nem a umidade relativa do ar são fatores que influenciam na temperatura aparente (NDFD). Nesse caso, uma boa aproximação para a temperatura aparente é a própria temperatura do ar.

2.3.5.5. Temperatura máxima/mínima

A temperatura certa para tomar banho tem sido bastante discutida. Dr. Melissa Piliang (THE WALL STREET JOURNAL), uma dermatologista especialista na Clínica Cleveland, diz que a temperatura ideal para lavar a sujeira vinda do ambiente e bactérias é de 112 graus Fahrenheit (44,44 graus Celsius) ou menos.

Além disso, ela confirma que a epiderme das pessoas mais jovens pode tolerar banhos mais quentes do que a dos mais velhos. Ela diz: "As we age, we have less tolerance for hot water, which may result in dry, itchy, red skin and possibly even eczema".

Consequentemente, por questões de segurança o sistema terá um limitador de temperatura máxima da água de 112 graus Fahrenheit e não terá um de temperatura mínima.

2.3.5.6. Outros fatores que podem afetar a preferência de temperatura

Durante o projeto, consideramos alguns outros fatores que podem afetar na preferência do usuário na temperatura da água do banho. Através de maior integração com outras fontes de dados (como Facebook e calendário) e incluindo esses fatores no algoritmo de previsão de temperatura, é possível aumentar a precisão da previsão da temperatura ótima para o usuário. Entre esses fatores estão:

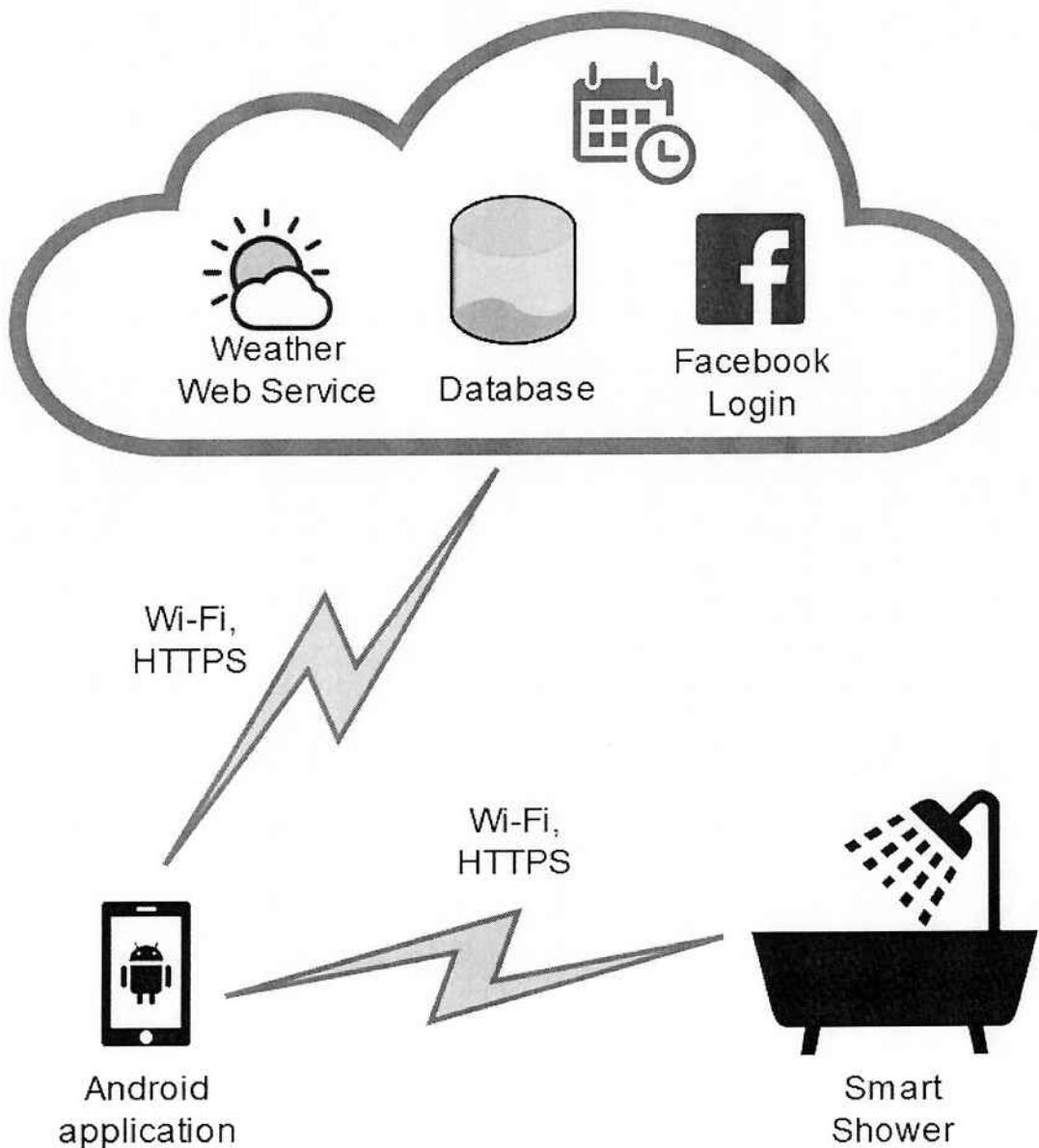
- **Período do dia:** Alguns usuários podem preferir diferentes temperaturas para banhos diurnos e banhos noturnos.
- **Estação do ano:** Alguns usuários podem preferir banhos mais quentes no inverno quando comparado com banhos no verão.
- **Perda de calor para o ambiente:** Se o local onde o chuveiro estiver instalado é mais aberto, a perda de calor pode ser maior e o usuário pode preferir uma temperatura mais alta.
- **Estado de saúde do usuário:** Se o usuário estiver doente (por exemplo, com uma gripe), ele pode preferir um banho mais quente.
- **Atividades do usuário antes do banho:** As atividades realizadas pelo usuário podem ter um grande impacto na temperatura que ele irá preferir. Por exemplo, se o usuário toma um banho logo após retornar da academia, ele provavelmente irá preferir um banho mais frio.

Levando em conta todos esses fatores, a previsão pode ser muito mais precisa, fornecendo a melhor experiência possível para o usuário.

2.4. Arquitetura Do Sistema

A Figura 4 mostra a arquitetura proposta. Os principais componentes do Sistema são o chuveiro inteligente (chuveiro convencional modificado para incluir circuitos integrados que permitem a integração com a IoT), uma aplicação Android responsável pela predição de temperatura e configuração, um banco de dados na nuvem que armazena dados históricos do chuveiro, um *web service* a partir do qual o sistema pode obter dados de condições climáticas, e um login por Facebook para melhorar a usabilidade.

Figura 4 - Arquitetura do sistema Smart Shower



3. POSSÍVEIS ALGORITMOS INTELIGENTES

3.1. Redes neurais artificiais

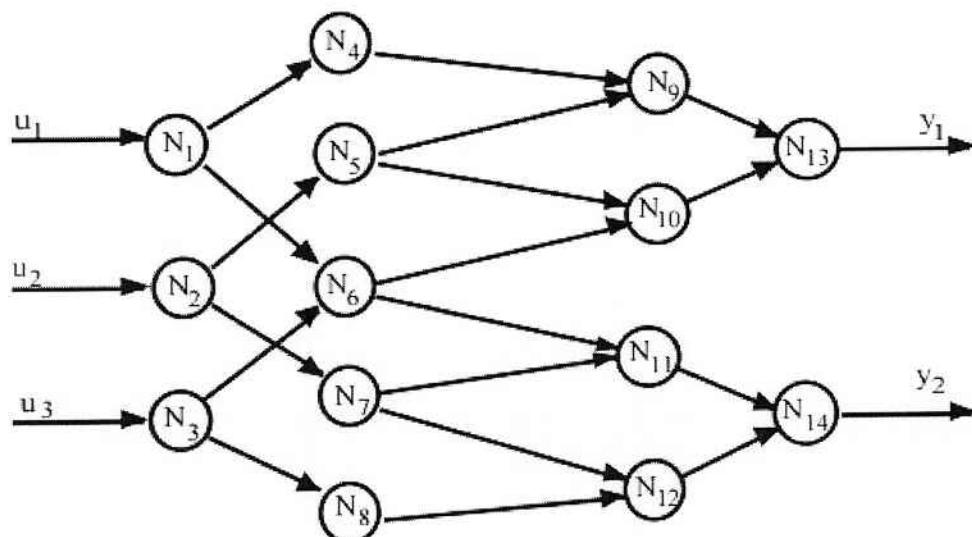
Entre os vários tipos de modelos, as redes neurais artificiais (RNAs) têm se destacado, despertando o interesse em estudar e usá-las em muitos projetos. RNAs são baseadas no cérebro humano, que é conhecido por ter muitos neurônios, ligados por conexões sinápticas.

É comum representar essas redes em camadas. Nesse caso, elas são chamadas de Redes em camadas, e são compostas de 3 tipos de neurônios:

- Neurônios que recebem excitações de fora são chamados neurônios de entrada;
- Neurônios que usam as respostas para mudar o mundo externo são chamados neurônios de saída;
- Outros neurônios que não são nem de entrada nem de saída são chamados de neurônios ocultos.

A Figura 5 mostra uma rede direta com 3 camadas. Este tipo de rede é muito comum, principalmente porque existem métodos de aprendizagem bastante difundidos e fáceis de usar. Essa rede pode estimar, com mais ou menos precisão, dependendo do número de neurônios na rede, qualquer função não-linear.

Figura 5 - Rede neural artificial direta



O comportamento da camada oculta depende do estímulo da camada de entrada, e o comportamento da camada de saída depende do estímulo da camada oculta. A interação entre as camadas depende do peso estabelecido para cada entrada, permitindo que uma rede mais complexa represente mais realisticamente e precisamente funções não-lineares.

Estudos mostram que não há vantagem de usar duas camadas ao invés de apenas uma para a construção de uma rede neural (MAS & FLORES). Há quatro passos para elaborar uma rede neural:

1. Organização da base de dados: A necessidade de se obter dados significativos, que abrangem situações normais e também exceções. Em seguida, alguns dados devem ser selecionados para o treinamento da rede, alguns devem ser selecionados para a validação da rede e outros devem ser selecionados para a verificação do desempenho da rede.
2. Arquitetura da rede: Este passo determina o número de neurônios de cada camada e se haverá ou não camadas ocultas.
3. Treinamento: Primeiro, os controles para inicialização e o tempo para o treinamento da rede devem ser determinados. Para determinar esse tempo, deve ser estabelecida uma taxa de erro médio, e mesmo que essa taxa de erro não seja atingida, o treinamento pode ser interrompido se ele estiver com uma taxa baixa ou a rede apresentar uma boa correlação entre dados reais e dados equiparados.
4. Desempenho da rede: É muito importante avaliar o desempenho da rede através de casos que não foram considerados para realizar o treinamento e a validação da rede.

Mesmo que as redes neurais artificiais possuam boas peculiaridades, algumas desvantagens podem se destacadas. Por exemplo, o atraso no treinamento; a necessidade de uma grande base de dados (alguns autores dizem que o número de ciclos de treinamentos considerados eficientes varia entre 500 e 3000); e a “caixa preta”, que é a falta de informação no porquê tal conclusão foi decidida pela rede.

A principal desvantagem é a necessidade de uma grande base de dados para treinar a rede. Seriam necessários muitos meses para obter tal banco de dados, e a complexidade para implementar uma RNA não geraria bons resultados considerando um período de uso inferior a um ano.

3.2. Regressão linear

A regressão linear é um modelo com um regressor x que possui uma relação com uma resposta y , que é uma reta. A regressão linear mais simples pode ser representada pela Equação 6 (DOUGLAS C. MONTGOMERY):

Equação 6 - Modelo de regressão linear simples

$$y = \beta_0 + \beta_1 \cdot x + \epsilon$$

Onde:

- β_0 é uma constante desconhecida que representa y quando x é igual a 0.
- β_1 é uma constante desconhecida que representa a inclinação da função.
- ϵ é um componente aleatório de erro do modelo.

O modelo usa um conjunto de dados de entrada (x, y) para estimar os coeficientes de regressão β_0 , β_1 e aproxima a função por uma reta. Consequentemente, quando introduzimos um valor x aleatório, a função calcula a respectiva saída y .

A principal desvantagem desse modelo é que ele só aproxima funções contínuas. Testes de usuários conduzidos pela nossa equipe concluíram que uma função contínua não seria uma boa aproximação da função da temperatura da água do banho, pois o usuário pode apresentar variações significativas da temperatura da água do banho em uma determinada faixa de temperatura aparente e não apresentar variações significativas em outras faixas.

3.3. Tabela dinâmica

Um modelo simples é utilizar os dados (temperatura aparente do ar, temperatura da água do banho) e construir uma tabela. Um exemplo é representado na Tabela 3:

Tabela 3 - Exemplo de tabela dinâmica

Temperatura aparente (°C)	Temperatura da água (°C)
...	...
10-11	39
11-12	38.5

12-13	38.5
13-14	38.5
14-15	38.5
15-16	38.5
16-17	37
17-18	37
18-19	37
19-20	37
20-21	37
21-22	37
22-23	37
23-24	37
24-25	35
25-26	35
26-27	35
27-28	35
...	...

Ao introduzirmos uma temperatura aparente, o modelo procuraria a linha da tabela que a contém e encontraria a respectiva temperatura da água. Se o usuário não gostar da temperatura, ele introduzirá uma nova de acordo com a sua preferência. Nesse caso, o modelo teria que calcular uma média ponderada entre a temperatura anterior da água do banho e a temperatura de entrada do usuário, e salvar a nova temperatura calculada para ser utilizada nos próximos banhos.

A principal desvantagem desse modelo é a quantidade de dados a serem salvos, porque seria necessário ter uma linha para cada temperatura aparente do ar. Com base no pressuposto de que os seres humanos não têm uma percepção sensorial tão aguçada a ponto de perceber uma variação de uma casa decimal, é possível implementar uma variação de apenas uma unidade na temperatura e ainda obter bons resultados.

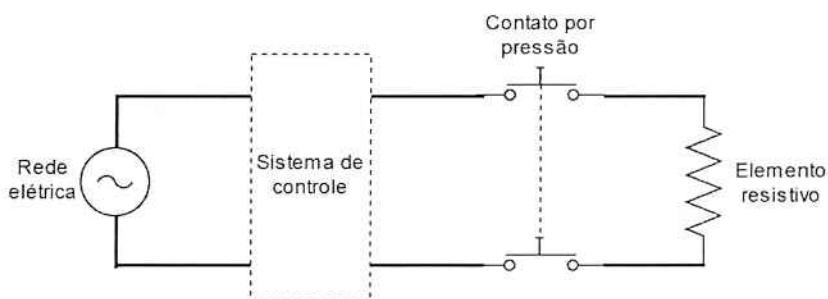
4. POSSÍVEIS ESCOLHAS DE HARDWARE

Há dois tipos principais de chuveiros no Mercado brasileiro: o chuveiro elétrico e o chuveiro a gás. (SISTEMAS PREDIAIS DE ÁGUA QUENTE, s.d.). Implementação em ambos os tipos de chuveiros será discutida abaixo. O chuveiro elétrico foi escolhido neste projeto para a prova de conceito.

4.1. Chuveiro elétrico

O chuveiro elétrico funciona a partir de um elemento resistivo que, quando energizado, transfere o calor produzido à água que passa por ele. A Figura 6 ilustra de forma simplificada o diagrama elétrico desse tipo de chuveiro. No mercado brasileiro, a maioria dos chuveiros elétricos possuem a potência nominal entre 4000W e 8000W e uma vazão de cerca de 4.5L/min (INSTITUTO NACIONAL DE METROLOGIA, NORMALIZAÇÃO E QUALIDADE INDUSTRIAL, s.d.). Alguns chuveiros incluem um termostato eletrônico, no entanto no Brasil a maioria possui apenas uma chave seletora de três posições (desligado, morno e quente) ou um potenciômetro manual para o ajuste de temperatura.

Figura 6 – Diagrama de um sistema simples de aquecimento elétrico (CHEGG STUDY, n.d.)



Uma maneira simples de implementar o sistema nesse tipo de chuveiro é através da instalação de um TRIAC no circuito para o controle da corrente elétrica e consequentemente para o ajuste da temperatura da água. Nesse caso, haveria um microcontrolador conectado à internet capaz de atuar no TRIAC para alcançar a temperatura desejada. Se o chuveiro já dispõe originalmente de um termostato digital, o número de componentes necessários é ainda menor. A aplicação Android dentro do dispositivo móvel seria usada para a comunicação com o microcontrolador através da

rede local, obtenção de informações a partir da internet, armazenamento de dados no banco de dados e predição de temperatura ideal para o usuário.

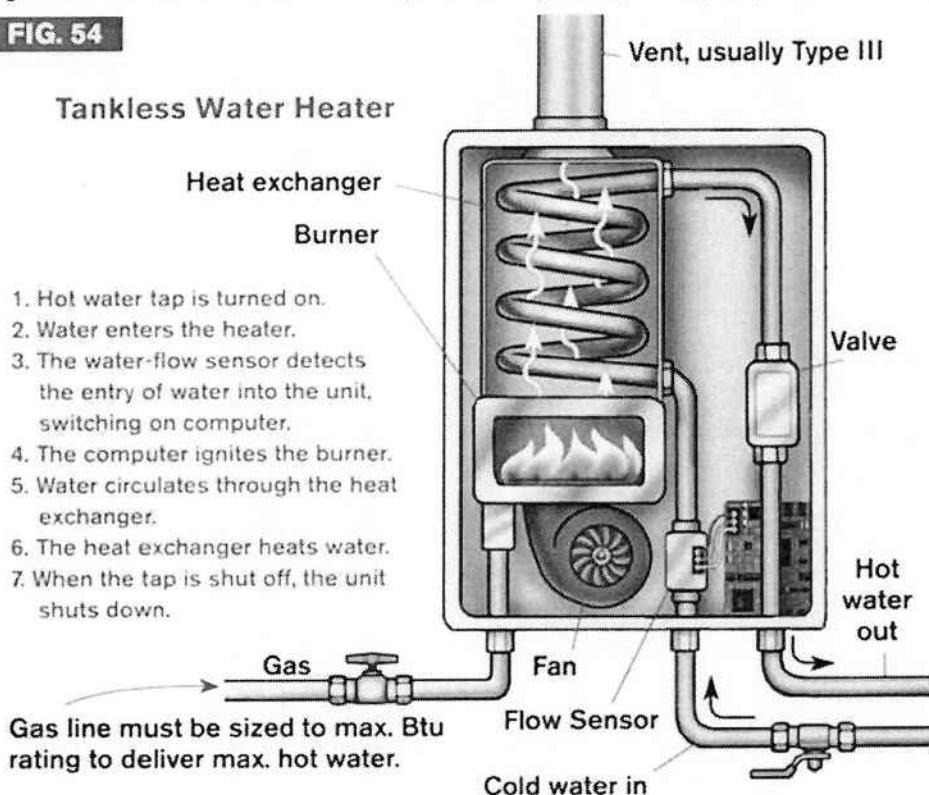
4.2. Chuveiro a gás

Chuveiros a gás possuem um aquecedor que aquece a água através do calor produzido pela combustão de gás. Os dois tipos mais comuns são: aquecedores com reservatório que armazenam a água aquecida para o momento do uso, e aquecedores de passagem que aquecem a água assim que passa, sem armazená-la. Atualmente o tipo mais comum no Brasil é o de passagem, já que esses aquecedores são mais eficientes e a água quente nunca acaba.

A Figura 7 ilustra um aquecedor de passagem convencional. Ele é composto basicamente por um queimador, um trocador de calor, válvulas de água e gás para ajuste manual (e, raramente, automático), e um circuito de controle eletrônico. Quando a vazão de água é detectada no sensor de fluxo, a ignição é acionada. O trocador de calor transfere o calor produzido pela chama para a água que passa pela serpentina, produzindo água quente na saída. Em alguns modelos existe um exaustor que é acionado no uso do aparelho para a exaustão forçada de gases. A Figura 7 mostra de forma geral como um aquecedor a gás de passagem funciona.

Figura 7 – Funcionamento de um aquecedor a gás de passagem (WILSON, n.d.)

FIG. 54



No caso do aquecedor a gás de passagem, uma válvula eletrônica pode ser usada para regular o fluxo de gás. Se presentes, válvulas manuais de gás podem ser acopladas a motores de passo para obter o mesmo efeito. Dessa forma, o sistema seria implementado utilizando-se um microcontrolador conectado à válvula ou motor, e também conectado à internet. De forma semelhante ao chuveiro elétrico, o microcontrolador e o dispositivo Android estariam na mesma rede, e então a aplicação Android seria responsável por basicamente toda a funcionalidade do sistema (exceto o controle do fluxo de gás, que seria tarefa do microcontrolador).

4.3. Segurança

Uma vez que o sistema envolve interação humana direta, é de suprema importância levar os riscos ao usuário e mitigá-los desde o início. Para o projeto, foi determinado que o sistema deve ter pelo menos a mesma segurança dos chuveiros convencionais.

No caso de chuveiros elétricos, a única modificação a ser feita seria o uso de um TRIAC para o controle de corrente elétrica. Dessa forma, o sistema irá apenas reduzir a potência do chuveiro e nunca aumentá-la acima daquela do chuveiro convencional. Portanto, a temperatura da água ficará entre a temperatura ambiente e a temperatura da potência nominal do chuveiro tradicional. Além de não aumentar o risco de queimaduras, o Sistema possui a vantagem de ser capaz de limitar a temperatura da água a um máximo de 44°C (a máxima temperatura adequada para banho, de acordo com a United Kingdom Homecare Association (UNITED KINGDOM HOMECARE ASSOCIATION) e a Health and Safety Executive (HEALTH AND SAFETY EXECUTIVE)), dessa forma garantindo a segurança do usuário.

Já no caso de chuveiros a gás, todos eles dispõem de mecanismos de segurança como detecção de chama, proteção contra sobreaquecimento, detecção de fluxo de água e verificação do estado do ventilador (quando existente). O presente projeto irá manter todos esses mecanismos em perfeito funcionamento, e irá apenas adicionar um sensor de temperatura à superfície do trocador de calor e uma válvula de controle na tubulação de entrada de gás (ou um motor de passo para girar a válvula manual dentro dos seus limites). Além do mais, o sistema irá implementar um limitador de temperatura mais restritivo do que a proteção de superaquecimento existente, garantindo ainda mais a segurança.

O Sistema irá comunicar-se tanto com componentes internos quanto com serviços web utilizando protocolos seguros como o HTTPS, e a comunicação local sem-fio será criptografada com os protocolos mais recentes, como o PSK WPA2. Ao utilizar os protocolos mais atualizados, pode-se assegurar que a segurança e a privacidade não serão comprometidas. Além do mais, as informações do usuário coletadas serão armazenadas em banco de dados protegido e serão utilizadas apenas para o usuário de quem ela foi extraída, não sendo compartilhada com ninguém mais.

5. PROVA DE CONCEITO

Para a prova de conceito, foi escolhida a implementação do sistema no chuveiro elétrico convencional pela sua maior simplicidade e menor custo de implementação.

5.1. Componentes de hardware do sistema

Os componentes de hardware utilizados no sistema são descritos a seguir.

5.1.1. Chuveiro elétrico convencional

Trata-se de um chuveiro elétrico convencional simples, composta basicamente por um resistor e um detector de pressão de água. Para a prova de conceito foi escolhida um chuveiro elétrico simples de 220V.

Figura 8 – Um chuveiro elétrico convencional



5.1.2. Sensor de temperatura DS18B20

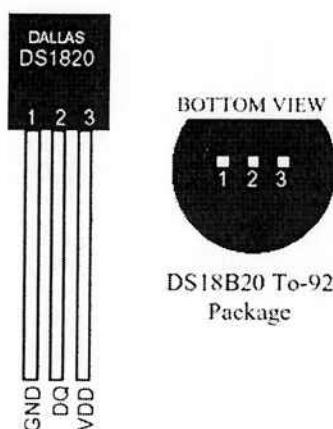
Foi escolhido o sensor DS18B20, um sensor digital de temperatura com as seguintes características: (MAXIMINTEGRATED®):

- Interface 1-wire, requer apenas um pino para a comunicação de dados

- Não requer componentes externos adicionais
- Tensão de alimentação entre 3.0V e 5.5V
- Faixa de temperatura: -55°C a +125°C
- +/- 0.5°C de precisão para temperaturas entre -10°C e +85°C
- Resolução ajustável de 9 a 12 Bits
- Tempo de conversão para 12 bits: 750ms (máx.)

Mais informações a respeito do sensor podem ser encontradas em suas especificações oficiais. (MAXIMINTEGRATED®).

Figura 9 – Termômetro digital 1-Wire DS18B20



Será utilizada a biblioteca DS18B20 disponível para o firmware NodeMCU (NODEMCU LIBRARY FOR ESP8266, n.d.).

5.1.3. Módulo ESP8266

O módulo ESP8266 (ESP8266EX OVERVIEW, n.d.) foi projetado inicialmente para ser um módulo Wi-Fi que, quando utilizado juntamente com outros microcontroladores como o Arduino, proverá a estes a conectividade Wi-Fi. No entanto, é possível usá-lo como um microcontrolador propriamente dito, com conectividade Wi-Fi, bastando para isso a substituição do seu firmware nativo por um terceirizado. Para este projeto o firmware NodeMCU será usado. O firmware permite que o módulo seja programado através de *scripts* em linguagem Lua.

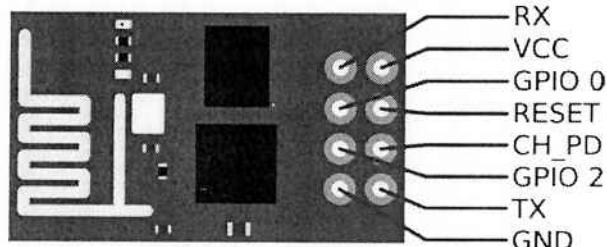
Algumas das características do módulo estão descritas a seguir: (EXPRESSIF):

- Wi-Fi 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP

- Pilha do protocolo TCP/IP integrada
- PLLs, reguladores, DCXO e unidades de gerenciamento de energia integrados
- Potência de saída de +19.5dBm no modo 802.11b
- Corrente de fuga em modo *standby* <10uA
- CPU de 32-bits integrado de baixo consumo, o qual pode ser usado como um processador de aplicações
- SDIO 1.1/2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- Consumo em modo *standby* < 1.0mW (DTIM3)

O módulo ESP8266 está disponível em várias versões. Na Figura 10 está a definição da pinagem de uma das versões (ESP-01):

Figura 10 – Pinagem do ESP8266 (ESP-01)



Na Tabela 4 está a descrição de cada pino:

Tabela 4 – Descrição da pinagem do ESP8266 (ESP-01)

Pino	Descrição
RX	Receive data
VCC	Power supply (3.3V)
GPIO 0	General Purpose IO 0
RESET	Reset (ativo baixo)
CH_PD	Chip power-down (ativo alto)
GPIO 2	General Purpose IO 2
TX	Transmit data
GND	Ground

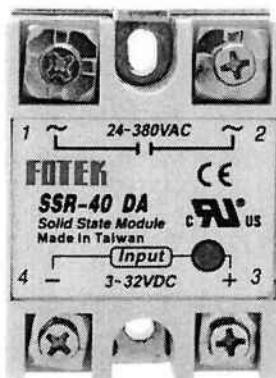
Os pinos TX e RX serão utilizados para programar o módulo e para a comunicação serial. GPIO 0, GPIO 2 serão utilizados respectivamente para obter dados do sensor de temperatura e atuar no relé.

O módulo será configurado para conectar a uma rede Wi-Fi local. Portanto, ele será capaz de receber comandos da aplicação Android e retornar informações para a aplicação.

5.1.4. Relé de estado sólido

Para o projeto será utilizado o relé de estado sólido Fotek SSR-40DA. O relé possui um ótimo custo benefício e suporta uma corrente AC continuada de 40A para uma tensão de entrada de 24-380VAC, valores muito acima dos requisitos do projeto. Além disso, possui acoplador ótico embutido que requer apenas 3V e alguns mA para atuar. Mais informações a seu respeito podem ser encontradas nas referências desse documento. (FOTEK). A Figura 11 representa o relé de estado sólido.

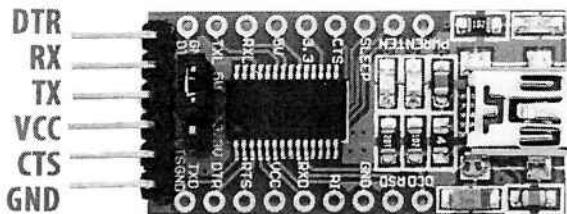
Figura 11 – Relé de estado sólido Fotek SSR-40 DA



5.1.5. Módulo FT232RL

O módulo FT232RL é uma interface USB-serial genérica. Ele não é parte do Sistema, mas será utilizado para programar o módulo ESP8266 neste projeto. Na Figura 12 está a definição da pinagem do módulo:

Figura 12 – Pinagem do módulo FT232RL



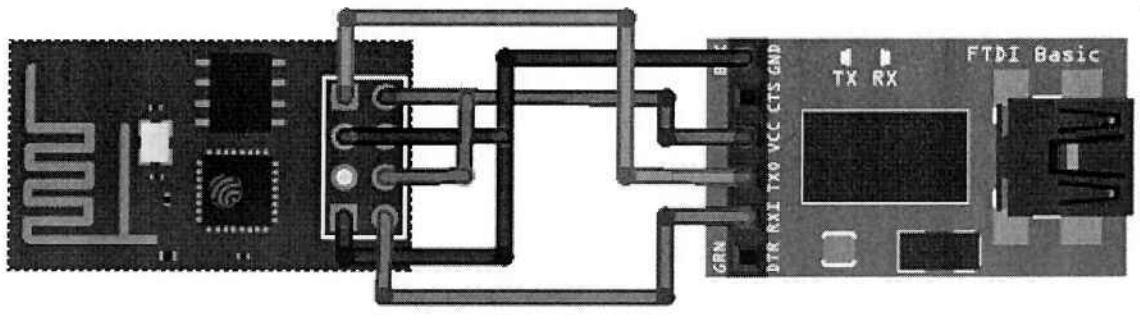
A Tabela 5 contém a descrição da pinagem:

Tabela 5 – Descrição da pinagem do módulo FT232RL

Pino	Descrição
DTR	Data Terminal Ready
RX	Receive data
TX	Transmit data
VCC	Power supply output
CTS	Clear to send
GND	Ground

Para a substituição do firmware para o NodeMCU a Figura 13 mostra a conexão necessária:

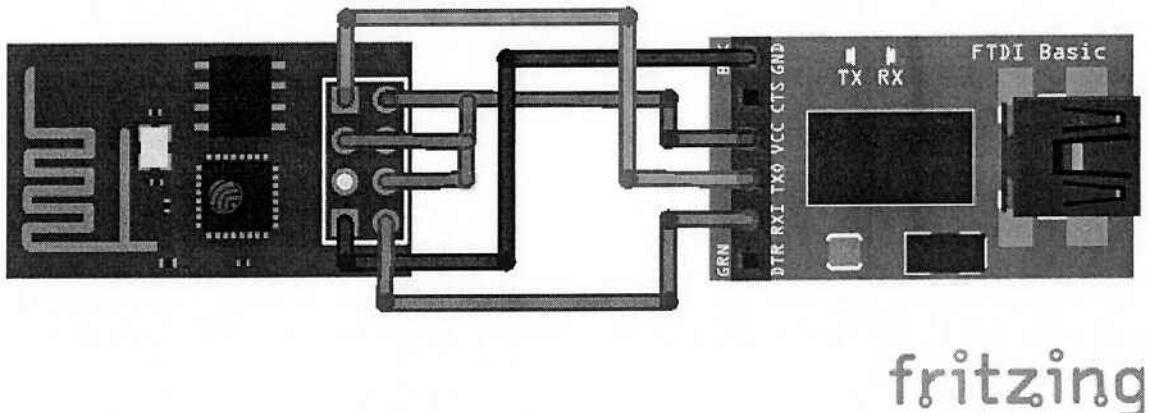
Figura 13 – Conexão entre ESP8266 e FT232RL para a substituição de firmware



fritzing

Para subir o código em linguagem Lua depois que o novo firmware estiver no lugar, a conexão mostrada na Figura 14 é utilizada:

Figura 14 – Conexão entre ESP8266 e FT232RL para subir o código ao módulo



5.1.6. Roteador

Será utilizado um roteador comercial com Wi-Fi 802.11n integrado para a infraestrutura de rede sem fio. Aqui, a criptografia WPA2-AES será utilizada.

5.1.7. Dispositivo móvel Android

Para este projeto, qualquer dispositivo Android recente poderá ser usado, uma vez que todos eles terão conectividade sem-fio Wi-Fi.

5.1.8. Custos estimados de hardware

Após a especificação dos componentes do sistema, os custos estimados do projeto foram calculados. Uma vez que o chuveiro elétrico, o roteador e o *smartphone* Android são dispositivos já existentes, eles não foram incluídos no cálculo de custos. Os custos estão a seguir (convertidos em reais):

Tabela 6 – Custos estimados de hardware do projeto

Hardware	Custo
Relé de estado sólido	R\$ 15,00

Sensor de temperatura	R\$ 5,00
Módulo ESP8266	R\$ 10,00
Módulo FT232RL	R\$ 10,00
Roteador sem-fio	R\$ 0,00
Smartphone Android	R\$ 0,00
Chuveiro elétrico	R\$ 0,00
TOTAL	R\$ 40,00

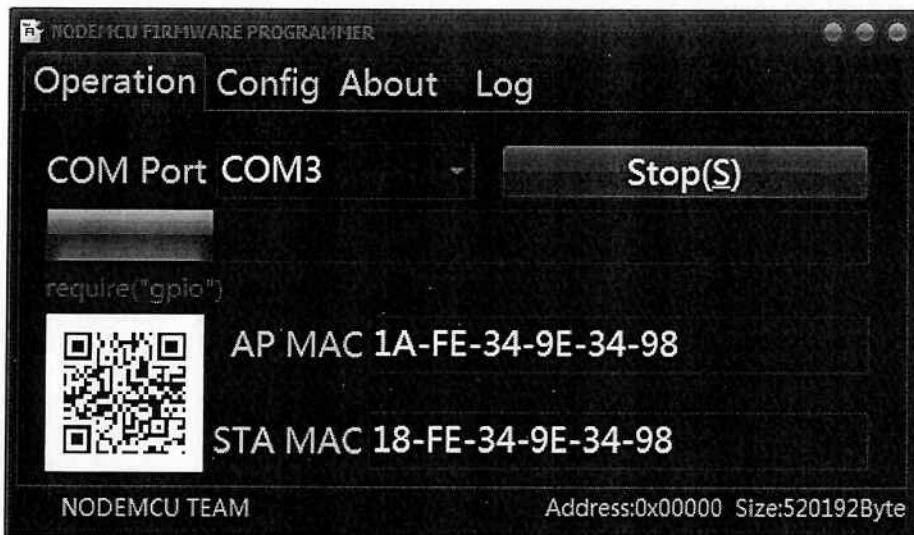
5.2. Software - Web Server

A partir das especificações do projeto (seção 2.1) pode-se definir alguns aspectos da implementação prática. A seguir serão definidos os componentes de software necessários para a construção do sistema.

O módulo ESP8266 implementará um web server. Para isso, será utilizado o NodeMCU, plataforma *open-source* projetada para o módulo que permite a programação deste em linguagem Lua. Através dessa linguagem, a programação do módulo será tão simples quanto a do Arduino. Mais informações sobre o NodeMCU podem ser encontradas nas referências deste projeto (NODEMCU - AN OPEN-SOURCE FIRMWARE BASED ON ESP8266 WIFI-SOC, s.d.).

Para colocar o firmware NodeMCU no módulo ESP8266 será usado o programa *nodemcu-flasher*, que pode ser encontrado nos repositórios oficiais de seus desenvolvedores (NODEMCU, s.d.). A Figura 15 mostra o programa em funcionamento.

Figura 15 - Screenshot do programa nodemcu-flasher



Depois de substituir o firmware, será possível programar o módulo usando a linguagem Lua. Trata-se de uma linguagem desenvolvida por uma equipe na Universidade PUC-Rio. Assim como outras linguagens de *script*, é interpretada e dinamicamente tipada. O interpretador usado no NodeMCU é rápido, robusto e tão compacto (apenas 276KB quando comprimido) que cabe dentro da memória flash de um microcontrolador. Mais informações a respeito da linguagem Lua podem ser encontradas em www.lua.org.

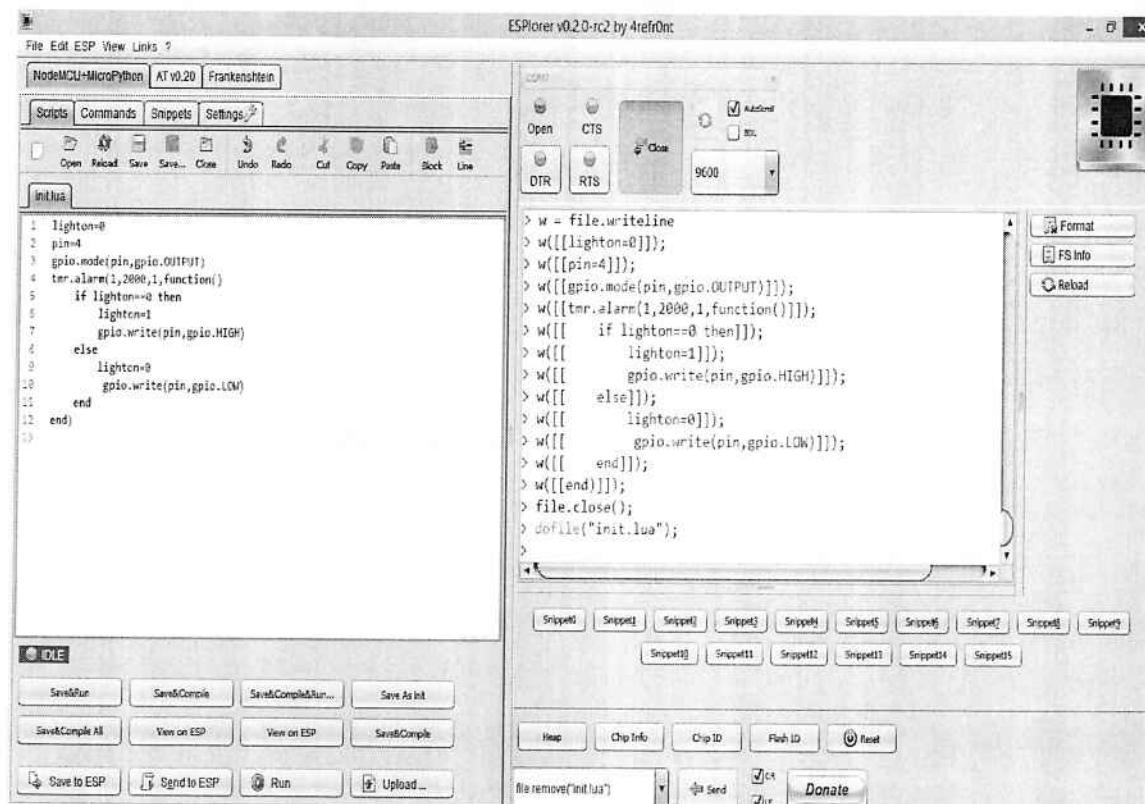
Através da linguagem Lua a programação do módulo torna-se muito mais simples. Com poucas linhas de código é possível estabelecer uma conexão TCP, criar um web server e aceitar conexões externas. Além disso, pode-se programar a entrada e a saída de forma parecida à do Arduino, o que é muito favorável para aqueles com experiência na programação do Arduino. A Figura 16 representa um script simples em linguange Lua.

Figura 16 – Um script Lua simples para o módulo ESP8266 para piscar um LED

```
initlua
1 lighton=0
2 pin=4
3 gpio.mode(pin,gpio.OUTPUT)
4 tmr.alarm(1,2000,1,function()
5     if lighton==0 then
6         lighton=1
7         gpio.write(pin,gpio.HIGH)
8     else
9         lighton=0
10        gpio.write(pin,gpio.LOW)
11    end
12 end)
```

Para subir o código ao ESP8266 será usada uma IDE chamada ESPlorer (ESPLORER - INTEGRATED DEVELOPMENT ENVIRONMENT (IDE) FOR ESP8266 DEVELOPERS, s.d.) (SANTOS, n.d.). Essa IDE integra um editor de texto com suporte às linguagens Lua e MicroPython, além de funções para fazer *upload* de código com apenas um clique.

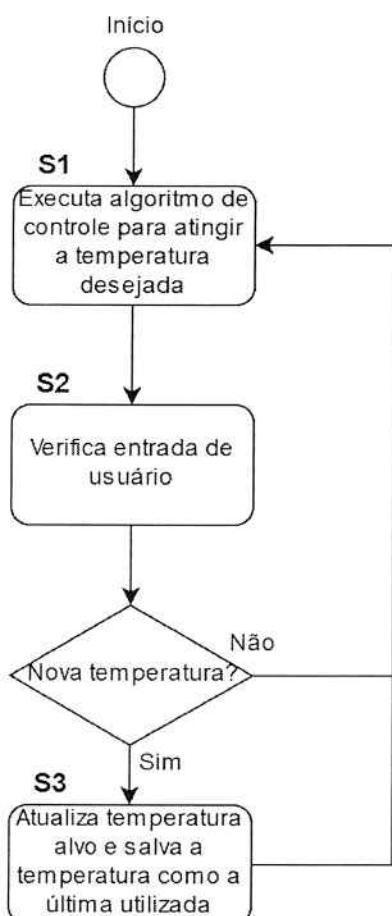
Figura 17 – IDE ESPlorer



O diagrama da Figura 18 mostra a sequência lógica do código a ser implementado no ESP8266. Quando o usuário abre o registro de água, o módulo será ativado e utilizará a última temperatura configurada pelo usuário, ou 37°C caso seja o primeiro uso do sistema. O algoritmo de controle é então executado (estado S1). Esse algoritmo lê a temperatura atual da água através do sensor DS18B20 e ajusta a corrente elétrica até atingir a temperatura configurada.

No próximo estado, o sistema aguarda por um comando do usuário através do *smartphone* (estado S2). Caso o usuário configure uma nova temperatura, esta é atualizada como a nova temperatura alvo e armazenada no módulo (estado S3). Se o usuário não configurar uma nova temperatura o modulo volta ao estado S1 para continuar a controlar a temperatura da água. Esse ciclo continua até o registro da água ser fechado, então o módulo é desativado.

Figura 18 – Diagrama de fluxo do código no módulo ESP8266



5.3. Software – Aplicação Android

5.3.1. Open Weather Map API

Open Weather Map é um serviço online que fornece dados meteorológicos como temperatura, umidade, pressão, velocidade do vento e direção do vento de acordo com a localização de seu interesse.

Esta API é livre para uso e possui muitas vantagens (MAP):

- API gratuita com acesso aos dados atuais do tempo para qualquer local, inclui 200,000 cidades
- Dados frequentemente atualizados com base em modelos globais e dados de mais de 40,000 estações meteorológicas
- Dados disponíveis em formatos JSON, XML, e HTML
- Disponível gratuitamente e possui outros tipos de contas pagas

Além disso, o Open Weather Map utiliza os códigos dos países seguindo a norma ISO 3166 (ISO 3166), o padrão internacional para códigos de países e suas subdivisões. Todos esses recursos descritos facilitam o acesso dos clientes aos dados meteorológicos.

Para chamar essa API pelo nome da cidade, uma conexão HTTP foi implementada, conforme mostra Figura 19 e Figura 20:

Figura 19 - Conexão HTTP com a API do Open Weather Map (1)

```
private static String BASE_URL = "http://api.openweathermap.org/data/2.5/weather?q=";
```

Figura 20 - Conexão HTTP com a API do Open Weather Map (2)

```
con = (HttpURLConnection) ( new URL(BASE_URL + location + "appid=73520525e803584275-931e2fe6c9124")).openConnection();
```

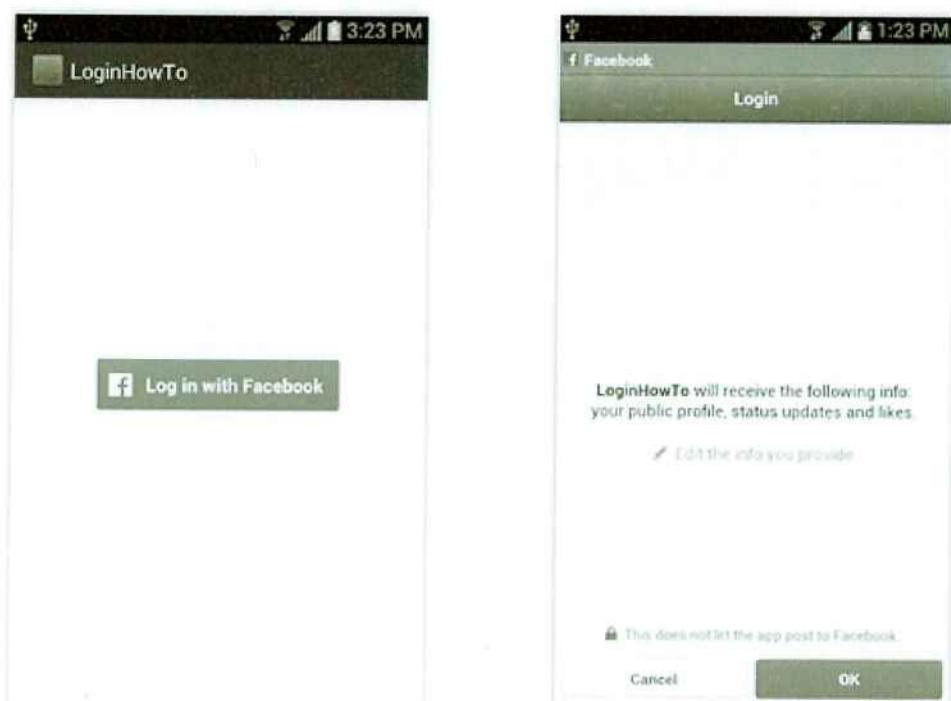
5.3.2. Login Facebook para Android

O SDK do Facebook para Android permite que as pessoas façam o login no seu aplicativo com sua conta do Facebook. É possível pedir-lhes permissão para

utilizar suas informações do Facebook para que sua aplicação possa ser executada com o máximo desempenho.

É rápido e conveniente para o usuário. A Figura 21 **Figura 21 Erro! Fonte de referência não encontrada.** representa um exemplo do login.

Figura 21 - Exemplo de login com uma conta Facebook



Também é possível verificar se o usuário já fez o login anteriormente e retomar seu perfil atual no aplicativo. A Figura 22 representa um exemplo de retomar um login.

Figura 22 - Exemplo de retomada de um login

```
if (Profile.getCurrentProfile() != null) {  
    launchMainScreen(TypeOfLogin.FACEBOOK_LOGIN_RESUME);  
}
```

5.3.3. Android Studio

Um aplicativo foi desenvolvido na plataforma Android, que se comunica com o módulo ESP8266 via Wi-Fi através do roteador. Ele pode obter a temperatura atual

da água do banho, e também possibilita o usuário ajustar manualmente a temperatura à sua necessidade.

O IDE do Android Studio foi utilizado para o ambiente de desenvolvimento. Essa IDE foi construída usando a infra-estrutura IDE IntelliJ IDEA da JetBrains, que possui os seguintes recursos (STUDIO):

- Um sistema de compilação flexível baseado no Gradle
- Um emulador rápido com muitos recursos
- Um ambiente unificado onde se pode desenvolver para todos os dispositivos Android
- Instant Run para enviar alterações a aplicativos em execução sem compilar um novo APK
- Modelos de códigos e integração com GitHub para ajudar a criar recursos comuns de aplicativos e importar exemplos de código
- Ferramentas e estruturas de teste abrangentes
- Ferramentas de verificação de código suspeito para detectar problemas de desempenho, usabilidade e compatibilidade de versões, entre outros
- Compatibilidade com C++ e NDK
- Compatibilidade integrada com o Google Cloud Platform, facilitando a integração do Google Cloud Messaging e do App Engine

O projeto Android é composto por dois pilares principais: Arquivos .xml, responsáveis pelo layout da tela; e arquivos .java, que permitem ao desenvolvedor implementar a lógica e possíveis conexões com outras plataformas, serviços e dispositivos.

A Figura 23 representa o ambiente de programação com um fragmento de código. O download da IDE e tutoriais podem ser encontrados nas referências bibliográficas no final deste documento (STUDIO).

Figura 23 - Ambiente de programação do Android Studio

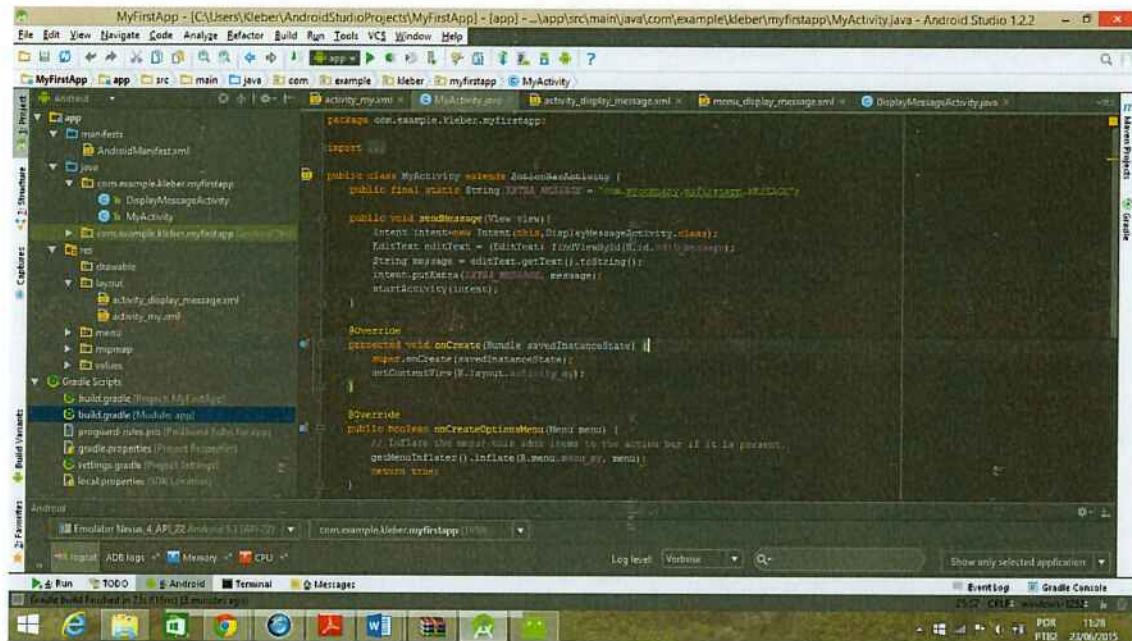


Figura 24 representa um programa de um tutorial executado em um emulador de um dispositivo Nexus 4 fornecido pelo Android Studio.

Figura 24 - Emulador do Android Studio



Arquivos .xml representam todos os layouts das telas da aplicação Android.

Figura 25 - Layout da tela ScreeLogin



Figura 26 - Layout da tela ScreenRegister

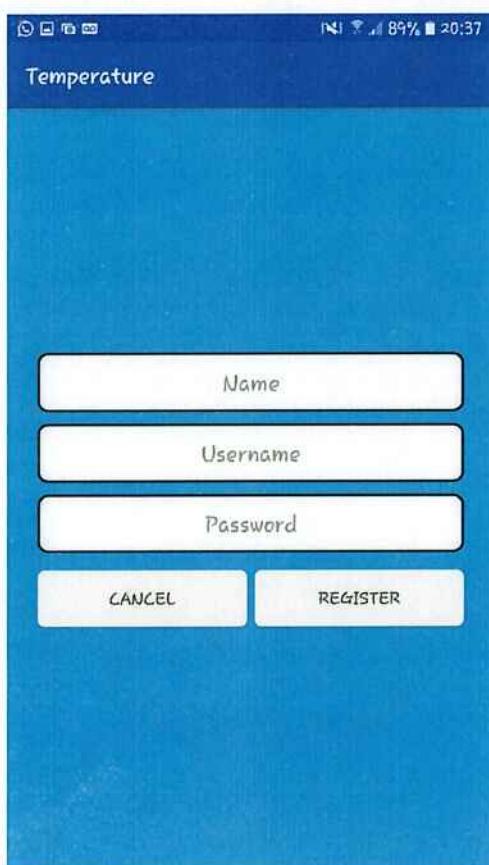


Figura 27 - Layout da tela ScreenMain

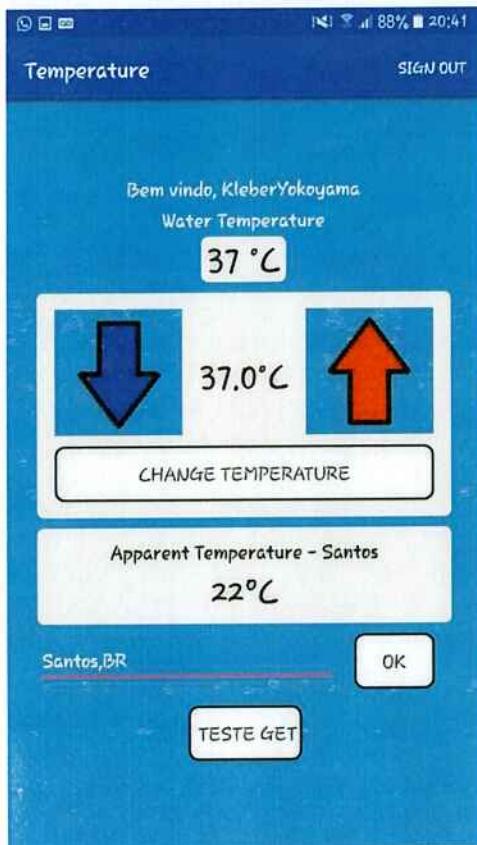


Figura 28 - Layout da tela WeatherScreen



A Figura 25 mostra a primeira tela que o usuário irá ver e é autoexplicativa. Ele pode fazer login com uma conta criada ou pode fazer login com a sua conta Facebook. O login pelo Facebook permite um melhor uso da aplicação, uma vez que os dados obtidos podem ser utilizados para estimar as preferências do usuário.

A Figura 26 mostra a tela quando o usuário não quiser utilizar o login pelo Facebook e optar por criar uma conta normal. É necessário introduzir *Nome*, *Nome do usuário* e *Senha* para criar uma conta.

A Figura 27 mostra a tela principal da aplicação. O usuário pode ver a temperatura aparente do ar e a temperatura estimada da água do banho. Além disso, se ele não gostar da temperatura estimada da água, ele pode inserir uma nova manualmente.

A Figura 28 mostra a tela quando o usuário desejar ver mais informações sobre o tempo, como: Pressão, umidade, velocidade do vento e direção do vento.

Arquivos .java:

- ScreenLogin.java: *Activity* responsável por logar os usuários na aplicação.
 - O fluxo de dados entre o Android e o banco de dados é realizado pelo Connection.java
- ScreenCadastro.java: *Activity* responsável por registrar novos usuários no banco de dados.
 - O fluxo de dados entre o Android e o banco de dados é realizado pelo Connection.java
- ScreenMain.java: *Activity* responsável por realizar a lógica inteligente e conectar a aplicação com o módulo ESP8266 via Wi-Fi.
 - O fluxo de dados entre o Android e o ESP8266 é realizado pelo ConnectionGet.java
- ScreenWeather.java: *Activity* responsável por mostrar as informações do tempo para o usuário.
 - O fluxo de dados entre o Android e a API do Open Weather Map é realizado pelo WeatherHttpClient.java
 - As informações do tempo são armazenadas no objeto do tipo Weather
- Connection.java: Classe responsável por conectar a aplicação Android ao banco de dados.

- Manda dados da aplicação Android para o banco de dados através do objeto do tipo DataOutputStream
- Recebe dados do banco de dados através do objeto do tipo InputStream
- WeatherHttpClient.java: Classe responsável por conectar a aplicação Android à API do Open Weather Map.
- Weather.java: Classe responsável por armazenar a informação do tempo.

O aplicativo Android é responsável por solicitar dados dos *web services* (Facebook e Open Weather Map) e do banco de dados (descrito na seção 5.4). Além disso, o aplicativo é responsável pelo envio de informações para o chuveiro e para o banco de dados atualizar a preferência do usuário.

Após resgatar os dados, a aplicação transforma os dados em informação para o sistema de aquecimento inteligente, utilizando algoritmos explicados nas subseções 5.3.4 e 5.3.5. A aplicação Android realiza a maior parte do processamento de dados do sistema.

5.3.4. Algoritmo - *Apparent air temperature*

Conforme explicado em 2.3.5, esse aplicativo utiliza três diferentes tipos de algoritmos para estimar a temperatura aparente do ar:

- Fórmula *Heat index* (2.3.5.2) mostrada na Figura 29.

Figura 29 - Fórmula *Heat index* implementada no Android Studio

```
private float heatIndex ( float temperature, float humidity ) {
    float adjustment = 0;
    float heatInd = (float) ( (-42.379) + (2.04901523*temperature) + (10.1433127*humidity) -
        (0.22475541*temperature*humidity) - (0.00683783*temperature*temperature) -
        (0.05481717*humidity*humidity) + (0.00122874*temperature*temperature*humidity) +
        (0.00085282*temperature*humidity*humidity) - (0.00000199*temperature*temperature*humidity*humidity) );
    if ( temperature < 112 && humidity < 13 ) {
        adjustment = (float) ( ((13-humidity)/4) * (Math.sqrt((17-Math.abs(temperature-95))/17)) );
    }
    else if ( temperature < 87 && humidity > 85 ) {
        adjustment = ( (humidity-85)/10 ) * ((87-temperature)/5);
    }
    return heatInd - adjustment;
}
```

- Fórmula *Wind chill* (2.3.5.3) mostrada na Figura 30.

Figura 30 - Fórmula *Wind chill* implementada no Android Studio

```
private float windChill ( float temperature, float airSpeed ) {
    return (float) ( 35.74 + (0.6215*temperature) - (35.75*Math.pow(airSpeed, 0.16)) + (0.4275*Temperature*Math.pow(airSpeed, 0.16)) );
}
```

- Fórmula *Ambient air temperature* (2.3.5.4). Temperatura aparente é igual à temperatura ambiente do ar.

5.3.5. Algoritmo – Temperatura da água do banho

Esse aplicativo usa a tabela dinâmica explicada na seção **Erro! Fonte de referência não encontrada..**. Esse modelo demonstra bons resultados, pois leva em conta apenas os dados anteriores que possuem a mesma temperatura aparente do ar. Por exemplo, se a temperatura aparente do ar for de 23°C, a aplicação estimará a temperatura da água do banho apenas com base nos dados que possuem a temperatura aparente do ar igual a 23°C. A temperatura padrão quando o usuário não possui um banco de dados é de 37°C.

Caso o usuário não goste da temperatura estimada da água, ele poderá manualmente inserir uma temperatura de sua preferência. Quando isso ocorrer, a aplicação irá calcular a média ponderada da temperatura antiga da água e a temperatura nova, conforme mostrado na Equação 7.

Equação 7 - Média ponderada entre a temperatura antiga da água e a temperatura nova

$$\text{Water temperature} = \frac{(1 * \text{Old Temperature} + 2 * \text{New Temperature})}{3}$$

Figura 31 e a Figura 32 mostram o que ocorre no banco de dados quando um usuário manualmente insere uma temperatura da água.

Figura 31 - Exemplo do algoritmo de média ponderada (1)

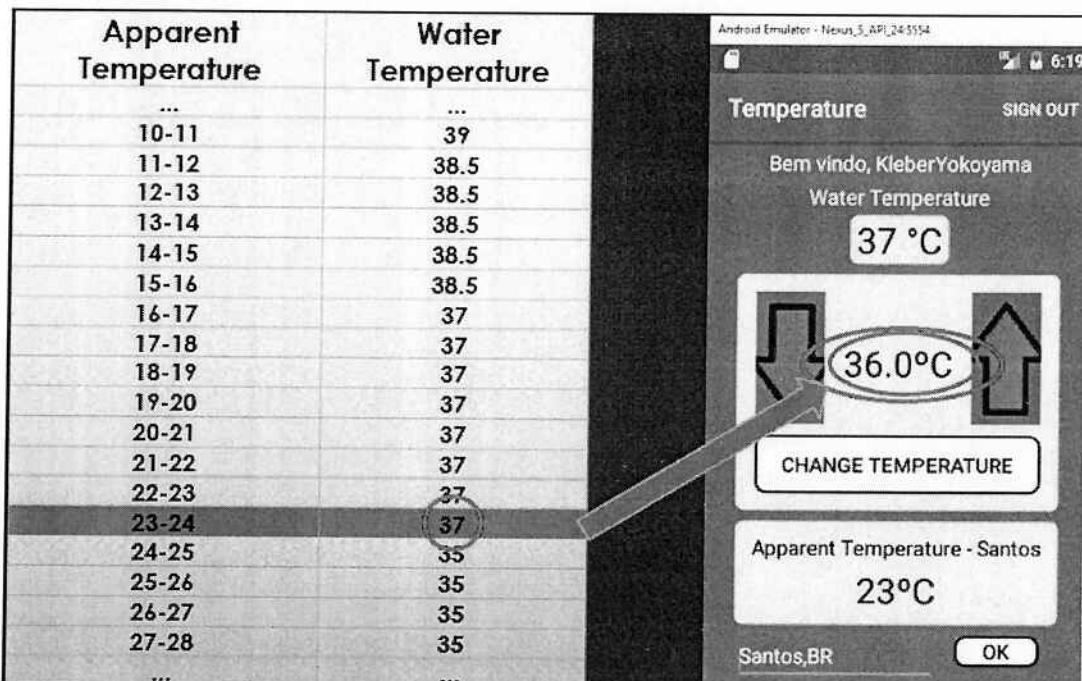
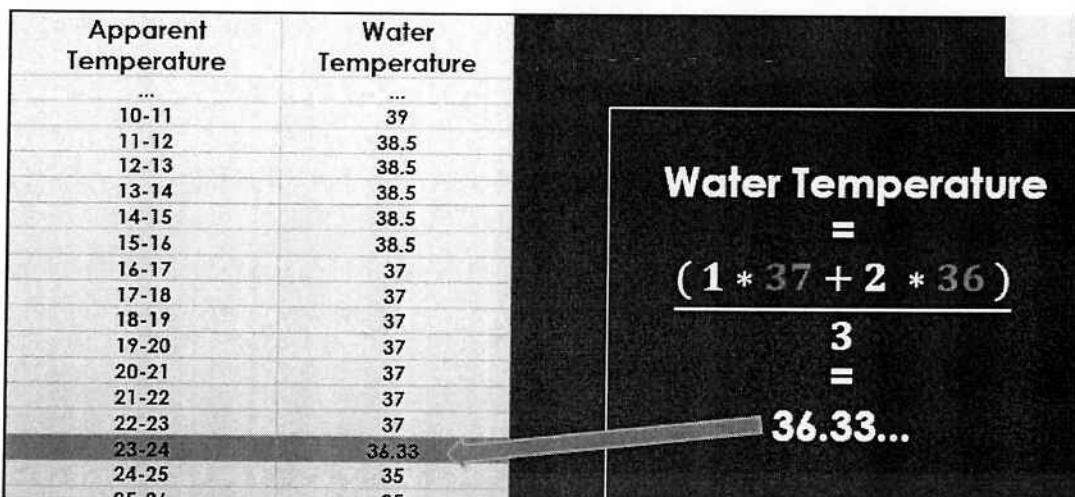


Figura 32 - Exemplo do algoritmo de média ponderada (2)



5.4. Software – Banco de dados MySQL

O aplicativo Android necessita de acesso a um banco de dados, a fim de armazenar dados dos usuários. Com o intuito de diminuir os custos, nós utilizamos o nosso computador pessoal como um servidor. O software WampServer instala automaticamente um servidor Apache, configura um banco de dados MySQL e instala aplicativos de suporte para PHP com fácil configuração e manutenção.

Utilizamos o WampServer para gerenciar os serviços do Apache e do MySQL. O WampServer possui vários recursos fáceis de utilizar (WAMP SERVER, s.d.):

- Alternar entre on-line/off-line (permitir acesso a todos ou apenas localhost)
- Instalar e trocar versões do Apache, MySQL e PHP
- Gerenciar as configurações do servidor
- Acessar os logs
- Acessar os arquivos de configurações
- Criar *aliases*
- Alterar o idioma do menu do WampServer

O banco de dados do projeto possui duas tabelas, mostradas nas Figura 33 e Figura 34:

Figura 33 - tblogin

	name	user	password	email	gender	birthday	country	hometown	location
<input type="checkbox"/>	Editar Copiar Remover Kleber	kleber123	kleber123		NULL	NULL	NULL	NULL	NULL
<input type="checkbox"/>	Editar Copiar Remover maria	maria123@maria.com	maria123		NULL	NULL	NULL	NULL	NULL
<input type="checkbox"/>	Editar Copiar Remover joao	joao123@joao.com	joao123		NULL	NULL	NULL	NULL	NULL
<input type="checkbox"/>	Editar Copiar Remover Test	test	123		NULL	NULL	NULL	NULL	NULL
<input type="checkbox"/>	Editar Copiar Remover jose	jose123	jose123		NULL	NULL	NULL	NULL	NULL
<input type="checkbox"/>	Editar Copiar Remover KleberYokoyama	1518363338177689	1518363338177689	kleberthy@gmail.com	male	09/19/1991	pt_BR	SaoPaulo	Brazil

Tblogin é responsável por armazenar todos os usuários e seus respectivos dados. A chave primária dessa tabela é definida pela coluna “user”, portanto, essa aplicação não permite registrar duas contas com o mesmo “user”.

Figura 34 - tbdadostemperatura

	user	apparentTemperature	waterTemperature	period
<input type="checkbox"/>	Editar Copiar Remover Kleber123	23	39.1605	noite
<input type="checkbox"/>	Editar Copiar Remover Kleber123	28	40	noite
<input type="checkbox"/>	Editar Copiar Remover Kleber	23	41.5	noite
<input type="checkbox"/>	Editar Copiar Remover kleber123	24	40.1111	noite
<input type="checkbox"/>	Editar Copiar Remover kleber123	0	37	noite
<input type="checkbox"/>	Editar Copiar Remover kleber123	20	38.4444	noite
<input type="checkbox"/>	Editar Copiar Remover 1518363338177689	20	41.2442	noite
<input type="checkbox"/>	Editar Copiar Remover 1518363338177689	20	37	dia
<input type="checkbox"/>	Editar Copiar Remover 1518363338177689	22	37.3333	noite
<input type="checkbox"/>	Editar Copiar Remover 1518363338177689	0	37.5	noite
<input type="checkbox"/>	Editar Copiar Remover 1518363338177689	21	37	noite

Tbdadostemperatura é responsável por armazenar os dados de todos os usuários. A chave composta é definida por 3 colunas: “user”, “apparentTemperature”

e “period”, permitindo que a aplicação distinga entre um banho noturno e um banho matinal.

Para obter dados reais, os desenvolvedores desse projeto coletaram dados de seus banhos para popular o banco de dados. É importante enfatizar a imprecisão do termômetro e a imprecisão da coleta de dados (por exemplo, perda de calor da água para o ambiente). A Figura 35 mostra uma tabela com os dados obtidos experimentalmente.

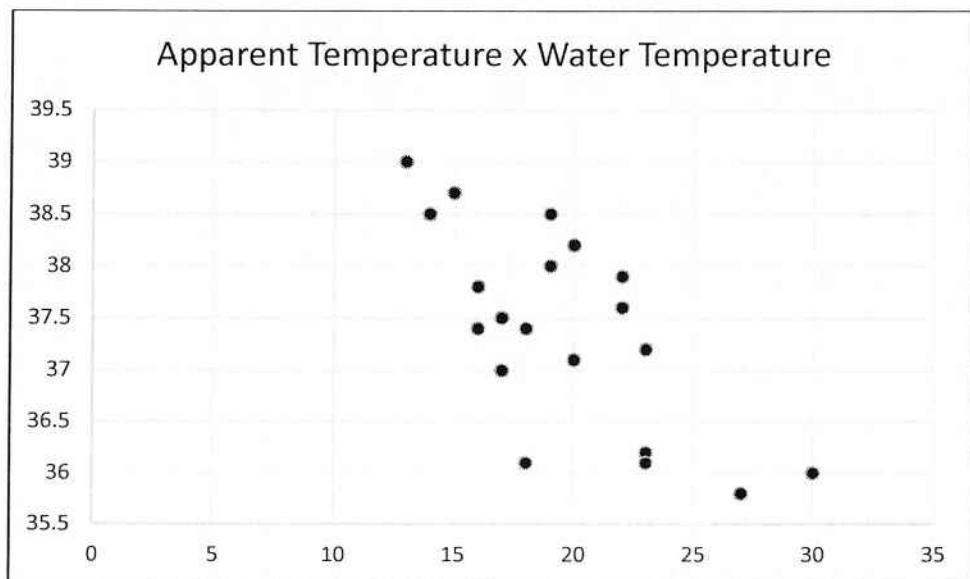
Figura 35 - Dados coletados dos banhos

Temperatura Aparente	Temperatura da água	Horário	Período
13	39	7:40	dia
14	38.5	22:00	noite
15	38.7	20:00	noite
16	37.8	8:23	dia
16	37.4	7:50	dia
17	37	15:50	dia
17	37	10:00	dia
17	37.5	19:12	noite
18	37.4	8:20	dia
18	37.4	8:10	dia
18	36.1	0:00	noite
19	38	7:00	dia
19	38.5	23:00	noite
20	38.2	9:30	dia
20	38.2	13:00	dia
20	37.1	23:20	noite
22	37.9	10:00	dia
22	37.6	23:00	noite
23	37.2	10:00	dia
23	36.2	10:00	dia
23	36.1	7:30	dia

27	35.8	14:20	dia
30	36	14:00	dia

Existe uma relação inversamente proporcional entre a temperatura aparente do ar e a temperatura da água do banho. Podemos ver essa tendência claramente definida no gráfico abaixo pela Figura 36.

Figura 36 - Gráfico dos dados coletados dos banhos



6. TESTES E RESULTADOS

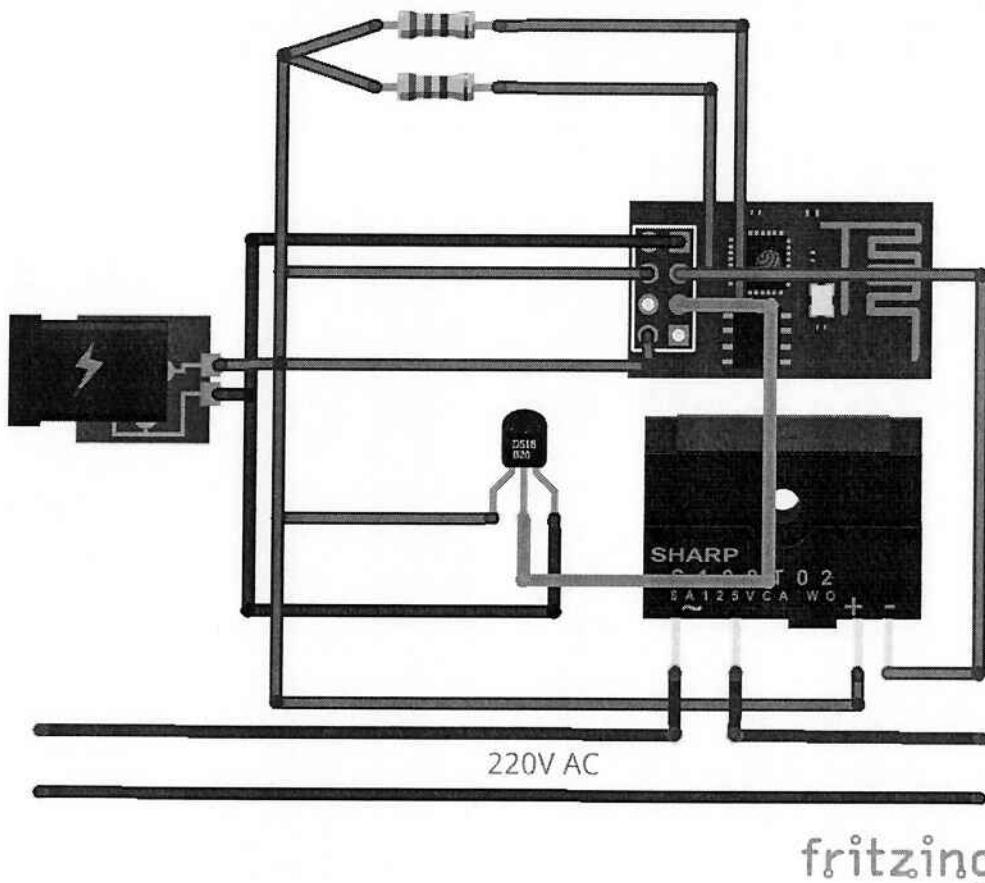
6.1. Testes preliminares

6.1.1. ESP8266

Primeiramente, o hardware foi testado ligando o módulo ao computador e instalando os drivers necessários. Foi necessário testar diferentes taxas de comunicação até encontrar uma que o módulo utiliza para se comunicar. Depois, o firmware NodeMCU foi colocado no módulo, e verificamos que estava funcionando corretamente enviando comandos simples em linguagem Lua.

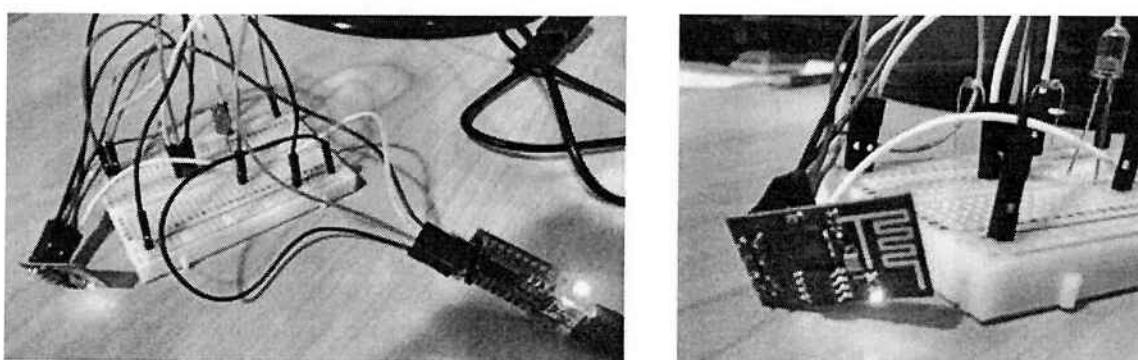
O módulo ESP8266 foi programado usando o módulo FTDI como descrito anteriormente. O sensor DS18B20 e o relé foram conectados aos pinos GPIO0 e GPIO2, respectivamente. Resistores *pull-up* foram utilizados em ambos pinos, uma vez que o módulo necessita deles para ser inicializado. A Figura 37 representa o diagrama de conexão utilizado:

Figura 37 - Diagrama de conexão do ESP8266

*fritzing*

Além do software do módulo, o sensor de temperatura foi testado para a familiarização com o protocolo 1-Wire e os comandos aceitos pelo sensor. Então, um algoritmo de controle simples foi implementado para ler o valor a partir do sensor e ligar ou desligar o relé para atingir a temperatura alvo, bem com um servidor web com uma simples API RESTful, que pode receber comandos e enviar informações de volta. Em seguida, o módulo foi testado pelo navegador para confirmar que tudo estava funcionando conforme o esperado. A Figura 38 representa os testes com o módulo e seus periféricos.

Figura 38 - Teste do módulo ESP8266 e periféricos



6.1.2. Aplicação Android

O aplicativo Android foi a segunda parte do teste. Foi necessário testar a aplicação utilizando o debugger do Android Studio até que todas as *activities* funcionassem conforme o esperado. O primeiro teste de comunicação Android foi o registro de um novo usuário no banco de dados, representado nas Figura 39 e Figura 40.

Figura 39 - Registrando uma nova conta

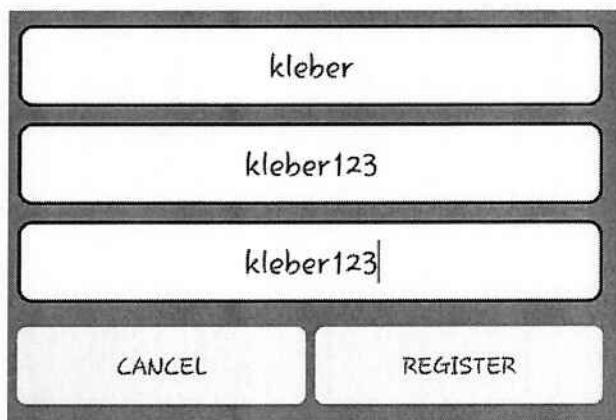
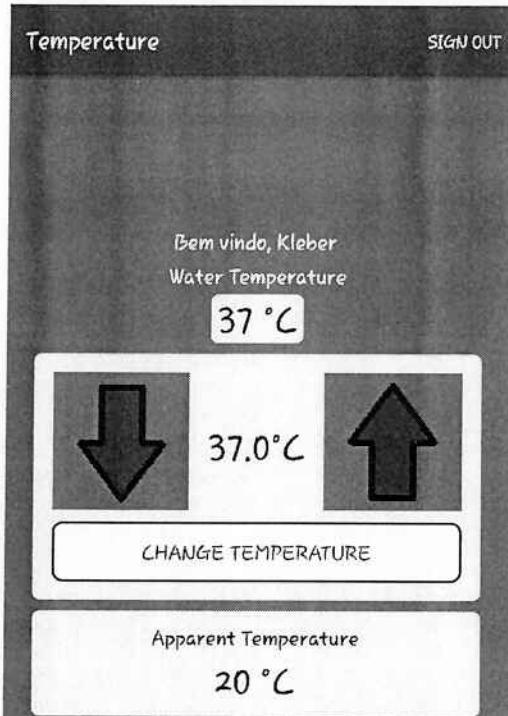


Figura 40 - Banco de dados após registro da nova conta

	name	user	password	email	gender	birthday	country	hometown	location
<input type="button" value="Editar"/> <input type="button" value="Copiar"/> <input type="button" value="Remover"/> Kleber	Kleber	Kleber123	Kleber123		NULL	NULL	NULL	NULL	NULL

Após o usuário ser criado, testamos o processo de login, como pode ser visto na Figura 41.

Figura 41 - Login com conta "normal"

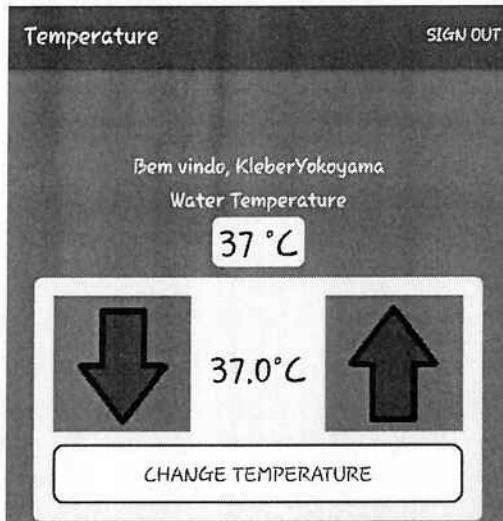


Embora o aplicativo tenha esse login "normal", é extremamente recomendável que o usuário faça o login com a sua conta Facebook para maximizar a eficiência do aplicativo. Na primeira vez que o usuário clicar no botão "Log in with Facebook", a aplicação automaticamente registrará o novo usuário no banco de dados. Nas Figura 42 e Figura 43, está o teste do registro do usuário Facebook e o seu login.

Figura 42 - Registro de uma conta Facebook

	name	user	password	email	gender	birthday	country	hometown	location
<input type="checkbox"/>	Editar Remover Kleber	Kleber123	Kleber123		NULL	NULL	NULL	NULL	NULL
<input type="checkbox"/>	Editar Remover maria	maria123@maria.com	maria123		NULL	NULL	NULL	NULL	NULL
<input type="checkbox"/>	Editar Remover joao	joao123@joao.com	joao123		NULL	NULL	NULL	NULL	NULL
<input type="checkbox"/>	Editar Remover Test	test	123		NULL	NULL	NULL	NULL	NULL
<input type="checkbox"/>	Editar Remover jose	jose123	jose123		NULL	NULL	NULL	NULL	NULL
<input type="checkbox"/>	Editar Remover KleberYokoyama	1518363338177689	1518363338177689	Heberthy@gmail.com	male	09/19/1991	pt_BR	SaoPaulo.Brazil	SaoPaulo.Brazil

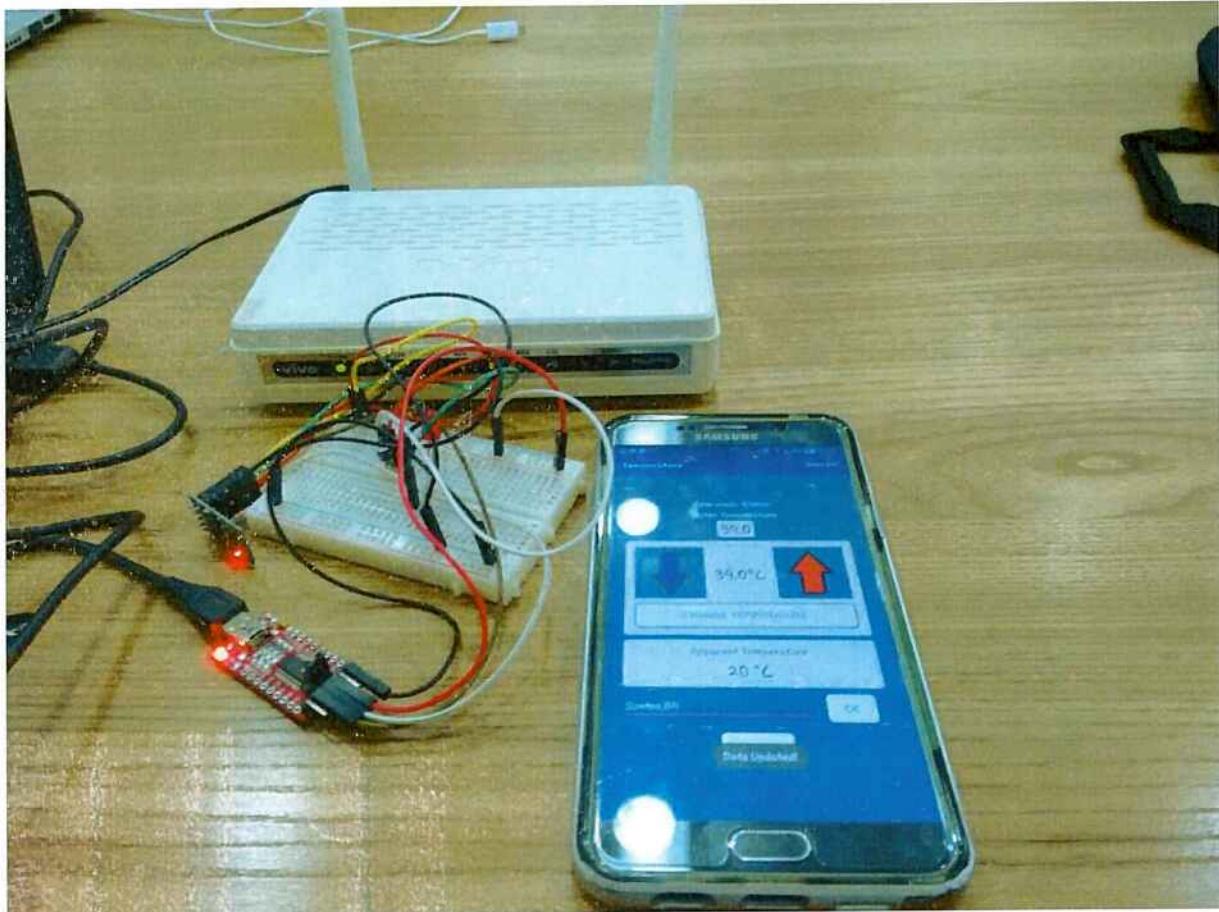
Figura 43 - Login com conta Facebook



6.1.3. Integração

A aplicação Android foi executada em um celular e a comunicação entre o módulo e o celular foi testada em uma rede Wi-Fi local. No início, houve alguns problemas, em que o aplicativo estava falhando por algum motivo desconhecido, mas depois de muitas correções e depurações foi confirmado que a comunicação funcionava conforme esperado, ou seja, o dispositivo Android podia mandar comandos para setar a temperatura e recuperar as informações necessárias do módulo. A Figura 44 representa os testes de integração.

Figura 44 - Testes de integração



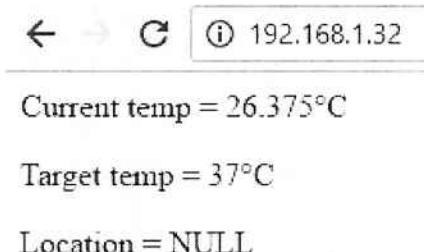
A primeira comunicação entre a aplicação Android e o módulo ocorre quando o sistema de aquecimento não possui uma localização. Se o usuário tiver efetuado o login com a conta Facebook, a aplicação automaticamente recuperará o local do usuário do Facebook e mandará para o sistema de aquecimento, caso contrário, será solicitado que o usuário insira sua localização manualmente, mostrado na Figura 45.

Figura 45 - Configurando manualmente a localização



Para uma melhor verificação dos testes de comunicação, implementou-se uma rotina que envia os atributos do módulo quando uma página web acessa seu IP via URL, como pode ser visto na Figura 46.

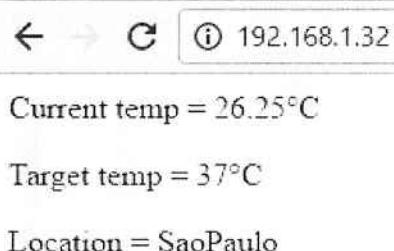
Figura 46 - Acessando os dados do módulo via URL



A screenshot of a web browser window. At the top, there is a header with left and right arrows, a refresh icon, and a text input field containing the IP address "192.168.1.32". Below the header, the page content displays three lines of text: "Current temp = 26.375°C", "Target temp = 37°C", and "Location = NULL".

E pode-se verificar que a localização foi mudada para “SaoPaulo”, conforme a Figura 47.

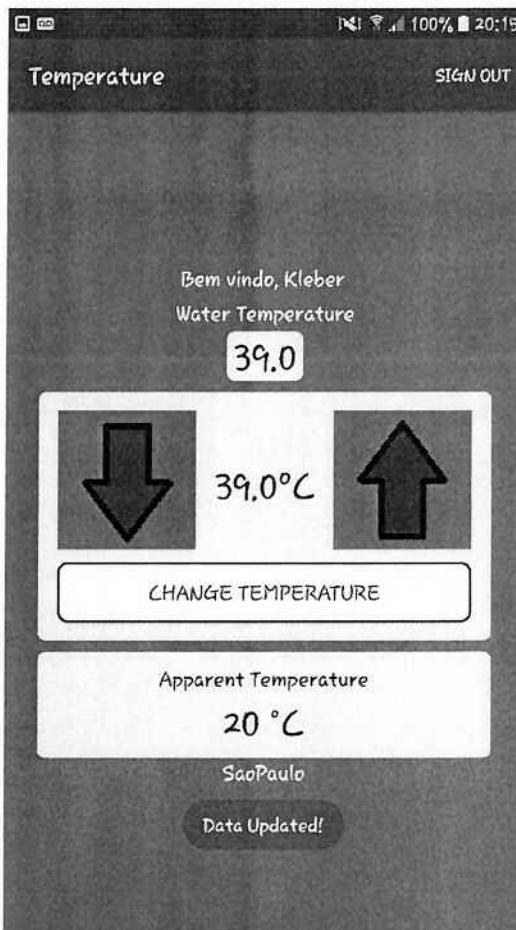
Figura 47 - Verificando a localização



A screenshot of a web browser window. At the top, there is a header with left and right arrows, a refresh icon, and a text input field containing the IP address "192.168.1.32". Below the header, the page content displays three lines of text: "Current temp = 26.25°C", "Target temp = 37°C", and "Location = SaoPaulo".

A Figura 48 representa o caso de se ajustar manualmente a temperatura da água do banho. O usuário deve pressionar a seta para cima e a seta para baixo para selecionar a temperatura desejada e pressionar o botão “Change Temperature”.

Figura 48 - Inserindo manualmente uma temperatura



Pode-se verificar que a temperatura alvo da água do banho foi mudada para “39°C”, conforme Figura 49.

Figura 49 - Verificando a temperatura alvo

← → C 192.168.1.32
Current temp = 26.5°C
Target temp = 39°C
Location = NULL

6.2. Testes finais

Para o teste final, precisou-se do sistema completamente construído e pronto para o teste. O teste possui os seguintes componentes: Um chuveiro elétrico controlado pelo circuito ESP8266, um tanque de água para fornecer água para o

chuveiro e coletar a água de volta, tubulações para a conexão do fluxo de água, uma bomba para circular a água coletada de volta para o chuveiro, um roteador para a infraestrutura de rede, um banco de dados on-line para armazenar os dados dos usuários, e um dispositivo Android com o aplicativo projetado para controlar o chuveiro e realizar todas as previsões de temperatura. Após todos os testes preliminares, obteve-se a implementação corretamente conectada e configurada, pronta para os testes finais.

No início, foi aberto o aplicativo Android. As instruções iniciais foram seguidas para logar e setar corretamente os dados no sistema (como nome, localização, etc.). Então, chegou-se à interface principal do aplicativo no qual era possível controlar a temperatura da água. Depois, o chuveiro foi ligado e verificou-se que a temperatura atingiu o valor configurado pelo celular. Em seguida, a água foi desligada e, em seguida, ligada novamente, para verificar se o algoritmo de previsão está funcionando como deveria. Uma vez que tudo ocorreu como esperado e não houve nenhum problema de conectividade ou erro de programa, os testes foram concluídos com sucesso. Durante o projeto foi necessário procurar por erros, identificá-los e corrigi-los, repetindo esse processo até que não haja mais erros ou anomalias.

7. CONCLUSÃO

O conceito de Internet das Coisas e Fog Computing possuem aplicações muito amplas. Este projeto demonstrou isso aplicando esses conceitos, desenvolvendo um sistema para um chuveiro inteligente. Ao integrar um microcontrolador em um chuveiro convencional, foi criado um chuveiro que é conectado ao mundo e consciente de seu contexto, sendo capaz de se comunicar com dispositivos em seu entorno. Ao adicionar um smartphone, web services e um banco de dados, obteve-se uma plataforma de *Fog Computing* no qual objetos são integrados com a nuvem. Assim, o sistema ganha muitos recursos que são indisponíveis em sistemas de chuveiro convencional, como por exemplo, ser capaz de configurar uma temperatura com precisão com um simples toque, realizar previsões de temperatura através da aprendizagem de máquina e coletar informações na nuvem.

Ainda há desafios para tornar esse sistema perfeito para qualquer tipo de tecnologia. Por exemplo, uma válvula de controle para aquecedores a gás pode ter um preço muito elevado para um consumidor médio. Além disso, a inteligência e as informações poderiam ser aprimoradas para tornar a previsão ainda mais precisa. No entanto, em um futuro próximo, dispositivos e sistemas devem se tornar comuns em todos os campos, incluindo nos chuveiros.

7.1. Trabalhos Futuros

Identificamos neste projeto muitas oportunidades de melhoria. Elas podem ser resumidas da seguinte forma:

Melhor algoritmo de controle para o módulo: O módulo poderia ter seu algoritmo de controle melhorado, de modo que a temperatura desejada possa ser alcançada de forma mais rápida. O algoritmo poderia ser customizado para diferentes tipos de chuveiro (elétrico ou a gás) para maximizar o desempenho e a eficiência.

Mais entradas: O sistema poderia integrar mais entradas do usuário e da internet. Como discutido anteriormente, fatores adicionais podem desempenhar um papel importante na preferência do usuário. Por exemplo, podemos incluir a agenda

do usuário, idade, sexo, etnia, condição de saúde, etc. no algoritmo de previsão, bem como a época do ano e hora do dia, para obter previsões mais precisas e uma melhor experiência para o usuário.

Algoritmos de previsão mais inteligentes: Pode-se usar algoritmos mais poderosos para lidar com a quantidade adicional de dados, tais como redes neurais e outros tipos de aprendizagem de máquina, juntamente com o *analytics*, para aumentar a precisão da predição. Esses algoritmos devem ser ajustados com cuidado para obter resultados significativos.

Mais integração: O sistema poderia ser integrado com outros objetos da casa (refrigerador, ar condicionado, iluminação, etc.) para obter mais informações a respeito do usuário, bem como providenciar as informações que ele possui para outros sistemas, melhorando a experiência do usuário como um todo. Por exemplo, se o usuário sempre toma banho antes de dormir, as lâmpadas podem utilizar essa informação sabendo quando elas devem se apagar.

Interface impermeável: Uma interface impermeável poderia ser construída e colocada dentro do box do chuveiro, de modo que o usuário possa mudar a temperatura enquanto toma banho sem a necessidade de ter um *smartphone* à prova da água ou sair do box do chuveiro.

REFERÊNCIAS

CENTER, W. P. (s.d.). *National Weather Service*. Acesso em 30 de Outubro de 2016, disponível em http://www.wpc.ncep.noaa.gov/html/heatindex_equation.shtml

CHEGG STUDY. (s.d.). *Electric Resistance Heating*. (Chegg Study) Acesso em 31 de Outubro de 2016, disponível em <https://www.chegg.com/homework-help/definitions/electric-resistance-heating-4>

CISCO. (s.d.). *Internet Of Things*. Acesso em 30 de Outubro de 2016, disponível em http://www.cisco.com/web/BR/assets/executives/pdf/internet_of_things_iot_ibsg_0411final.pdf

DOUGLAS C. MONTGOMERY, E. A. (s.d.). *Introduction to Linear Regression Analysis*. Acesso em 23 de Novembro de 2016

ESP8266EX OVERVIEW. (s.d.). Acesso em 24 de Novembro de 2016, disponível em <http://espressif.com/en/products/esp8266/>

ESPLORER - INTEGRATED DEVELOPMENT ENVIRONMENT (IDE) FOR ESP8266 DEVELOPERS. (s.d.). Acesso em 23 de Novembro de 2016, disponível em <http://esp8266.ru/esplorer/>

EXPRESSIF. (s.d.). *ESP8266EX Datasheet*. Acesso em 31 de Outubro de 2016, disponível em https://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

FOTEK. (s.d.). *SSR-40DA Solid State Relay datasheet*. Acesso em 31 de Outubro de 2016, disponível em <https://cdn.sparkfun.com/datasheets/Components/General/SSR40DA.pdf>

HEALTH AND SAFETY EXECUTIVE. (s.d.). *Managing the risks from hot water and surfaces in health and social care*. Acesso em 31 de Outubro de 2016, disponível em <http://www.hse.gov.uk/pubns/hsis6.pdf>

INSTITUTO NACIONAL DE METROLOGIA, NORMALIZAÇÃO E QUALIDADE INDUSTRIAL. (s.d.). *TABELA DE CONSUMO DE ENERGIA ELÉTRICA - CHUVEIROS ELÉTRICOS - EDIÇÃO 03/2016*. Acesso em 31 de Outubro de 2016, disponível em <http://www.inmetro.gov.br/consumidor/pbe/chuveiro.pdf>

ISO 3166. (s.d.). Acesso em 30 de Outubro de 2016, disponível em http://www.iso.org/iso/country_codes

JORGE, M. B. (s.d.). *Introduction to artificial neural networks*. Federal university of Santa Catarina. Acesso em 10 de Outubro de 2016

KIMES, D. S., NELSON, R. F., MANRY, M. T., & FUNG, A. K. (1998). *Attributes of neural networks for extracting continuous vegetation variables from optical and radar measurements*. *International journal of remote sensing*, v.19, n.14, p.2639-2663. Acesso em 10 de Outubro de 2016

MAP, O. W. (s.d.). Acesso em 10 de Outubro de 2016, disponível em <http://openweathermap.org/api>

MAS, J. F., & FLORES, J. J. (s.d.). *The application of artificial neural networks to the analysis of remotely sensed data*. *International Journal of Remote Sensing*, v.29, n.3, p.617-663. Acesso em 30 de Outubro de 2016

MAXIMINTEGRATED®. (s.d.). *Programmable Resolution 1-Wire Digital Thermometer*. Acesso em 31 de Outubro de 2016, disponível em <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>

MELO, A. P. (s.d.). *Developing a method to estimate the energy consumption of commercial building though the artificial neural networks*. (Doctoral thesis). Federal university of Santa Catarina. Acesso em 10 de Outubro de 2016

NDFD, N. D. (s.d.). *Apparent temperature*. Acesso em 30 de Outubro de 2016, disponível em <http://www.nws.noaa.gov/ndfd/definitions>

NODEMCU - AN OPEN-SOURCE FIRMWARE BASED ON ESP8266 WIFI-SOC. (s.d.). Acesso em 23 de Novembro de 2016, disponível em http://nodemcu.com/index_en.html

NODEMCU. (s.d.). *GitHub - nodemcu/nodemcu-flasher: a firmware flash tool for the nodemcu*. Acesso em 23 de Novembro de 2016, disponível em <https://github.com/nodemcu/nodemcu-flasher>

NODEMCU LIBRARY FOR ESP8266. (s.d.). Acesso em 24 de Novembro de 2016, disponível em https://github.com/nodemcu/nodemcu-firmware/tree/master/lua_modules/ds18b20

SANTOS, R. (s.d.). *Getting Started with the ESPloerer IDE*. Acesso em 24 de Novembro de 2016, disponível em <http://esp8266.ru/download/esp8266-doc/Getting%20Started%20with%20the%20ESPloerer%20IDE%20-%20Rui%20Santos.pdf>

SERVICE, N. W. (s.d.). *Weather Prediction Center*. Acesso em 30 de Outubro de 2016, disponível em <http://www.weather.gov/media/epz/wxcalc/windChill.pdf>

SISTEMAS PREDIAIS DE ÁGUA QUENTE. (s.d.). Acesso em 24 de Novembro de 2016, disponível em <http://www.suzuki.arq.br/unidadeweb/aula%2011/aula11.htm>

STUDIO, A. (s.d.). *Android Developer*. Acesso em 30 de Outubro de 2016, disponível em <https://developer.android.com/index.html>

THE WALL STREET JOURNAL. (s.d.). *Burning Question: What Is the Best Water Temperature for Your Bath or Shower*. Acesso em 10 de Outubro de 2016

UNITED KINGDOM HOMECARE ASSOCIATION. (s.d.). *Controlling Scalding Risks from Bathing and Showering*. Acesso em 31 de Outubro de 2016, disponível em <http://www.ukhca.co.uk/pdfs/UKHCA%20Guidance%20-%20Bathing%20and%20Showering%20-%20August%202016%20final.pdf>

WAMPSERVER. (s.d.). *wampserver.com*. Acesso em 24 de Novembro de 2016, disponível em <http://www.wampserver.com/en/>

WILSON, A. (s.d.). *Green Building Advisor*. Acesso em 31 de Outubro de 2016, disponível em Knoxville plumbing:

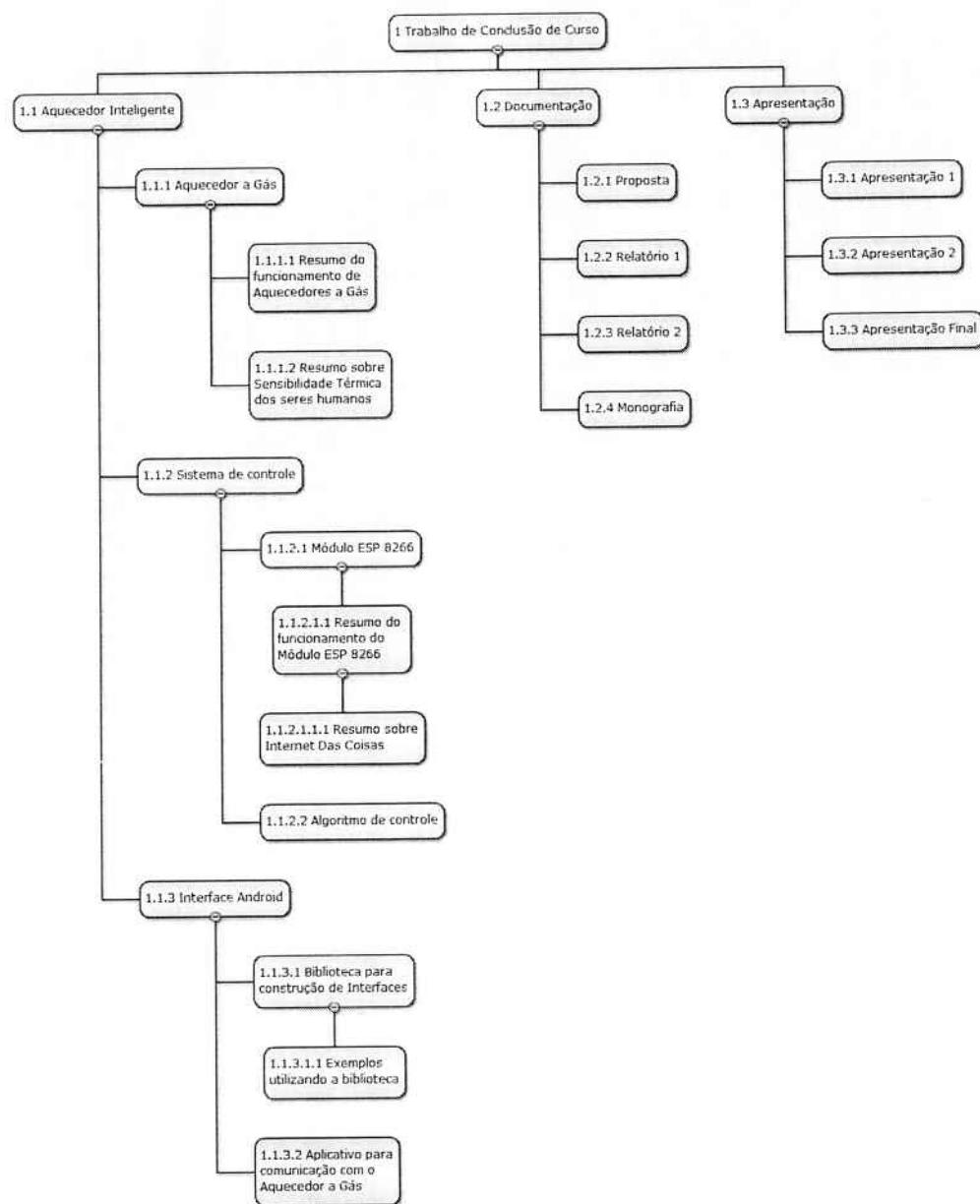
<http://www.greenbuildingadvisor.com/blogs/dept/energy-solutions/storage-vs-tankless-water-heaters>

APÊNDICES

Apêndice I: Estrutura De Divisão Do Trabalho

A Figura 50 representa a estrutura da divisão de tarefas do projeto I:

Figura 50 - Estrutura da divisão do trabalho do projeto I

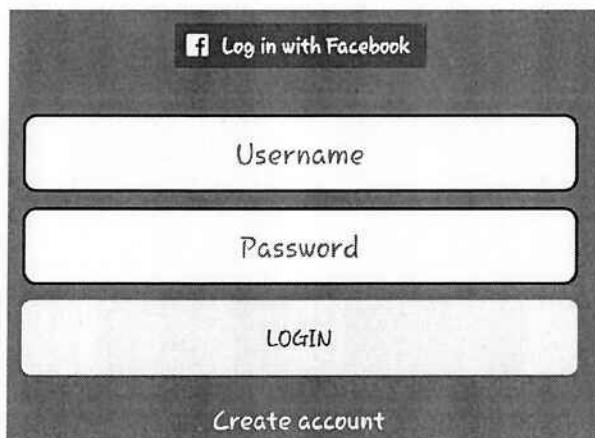


Apêndice II: Manual Do Usuário

Passo 1: Instale o aplicativo Android no seu smartphone Android.

Passo 2: Abra o aplicativo. A primeira tela é a tela de login, conforme Figura 51.

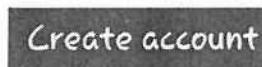
Figura 51 - Tela de login



Passo 3: Faça o login com a sua conta Facebook para poder usufruir ao máximo do aplicativo. Pule para o Passo 7. Se você não possui uma conta Facebook, por favor siga os Passos 4, 5 e 6.

Passo 4: Clique no botão “Create account”, conforme Figura 52.

Figura 52 - Botão Create account



Passo 5: Preencha todos os 3 campos necessários (*Name*, *Username* e *Password*) e clique no botão “Register”, conforme Figura 53.

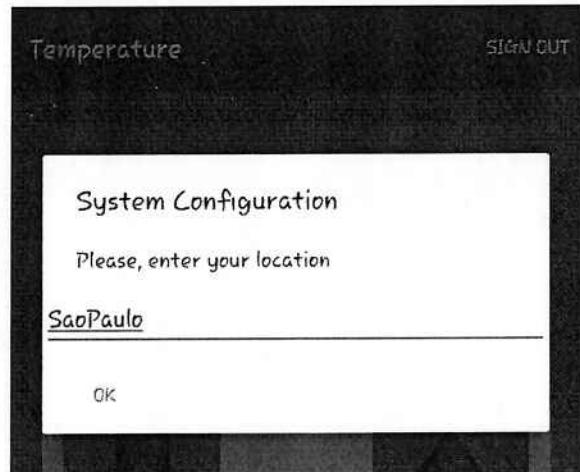
Figura 53 - Tela de registro

kleber	
kleber123	
kleber123	
CANCEL	REGISTER

Passo 6: Faça o login com a conta que você registrou. Caso seja a primeira utilização do sistema siga os próximos passos, caso contrário, pule para o Passo 8.

Passo 7: Antes de utilizar o sistema, você precisa configurar a localização do chuveiro. Caso você tenha logado utilizando sua conta Facebook, a aplicação automaticamente obterá a localização da sua cidade, caso contrário, insira manualmente sua localização, conforme Figura 54.

Figura 54 - Configuração manual da localização



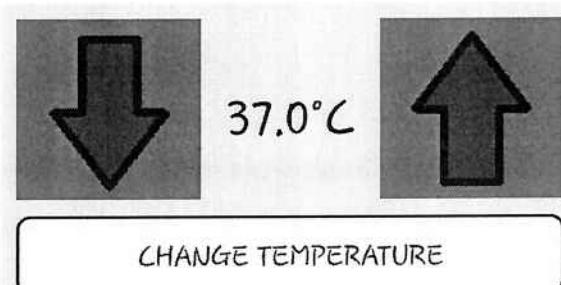
Passo 8: Hora do banho! É fácil e autoexplicativo. No momento que você ligar o chuveiro, o algoritmo inteligente irá estimar a temperatura da água para o seu banho, conforme Figura 55.

Figura 55 - Temperatura da água estimada



Passo 9: Caso você não goste da temperatura estimada, você pode aumentá-la ou diminuí-la. Para enviar a temperatura selecionada pressione o botão "Change Temperature". Quanto mais você escolher a sua preferência da temperatura da água do banho, mais inteligente o aplicativo ficará. A Figura 56 mostra os campos para mudar a temperatura da água.

Figura 56 - Campos para mudar a temperatura



Passo 10: Você pode ver informações extras sobre o tempo apertando no texto "Apparent Temperature", conforme Figura 57.

Figura 57 - Apparent temperature

