

MONIT-RIB II- TECNOLOGIA DA INFORMAÇÃO DE COMUNICAÇÃO PARA MONITORAMENTO DE RIBEIRÕES EM CASOS DE CHEIAS

MONIT-RIB II- INFORMATION COMMUNICATION TECHNOLOGY FOR MONITORING STREAMLET IN CASES OF FLOODS

Fábio Alexandrini,
Doutor e Mestre em Engenharia de Produção e Sistemas, Bacharel em Ciência da Computação
Professor EBTT IFC – Rio do SUL/ fabio.alexandrini@ifc.edu.br

André Alessandro Stein
Mestrado em Computação Aplicada, Bacharel em Engenharia da Computação
Professor EBTT IFC – Rio do SUL / andre.stein@ifc.edu.br

Angelo Victor Kraemer Foletto
Formado em Bacharel em Ciência da Computação – IFC/Campus Rio do Sul /
angelovkfoletto@gmail.com

Ricardo Reis Conde
Acadêmico do curso Bacharel em Ciência da Computação– IFC/Campus Rio do Sul/
linxofre@gmail.com

Resumo:

Eventos climáticos afetam, com maior frequência, as populações próximas de Ribeirões, pois possuem comportamentos diversos aos dos rios. Atualmente, esses rios encontram-se monitorados pela Defesa Civil, que emite alertas à população periodicamente. Com isso, gerou este novo problema de pesquisa, onde é possível expandir e aprimorar técnicas para medição de nível de água por meio de dispositivos vinculados ao conceito de internet das coisas (IoT). Por serem compactos, dinâmicos e consumirem pouca energia, tendem a ganhar emprego em protótipos pequenos e de baixa necessidade de manutenção. O que permite a instalação em ambientes de difícil ou limitado acesso. Foram utilizados um mini-microcomputador, sensor ultrassônico e uma câmera. Após testes e discussões, pode-se verificar que o protótipo experimental é permite agregar aos métodos tradicionais de aquisição destes dados, garantindo precisão, persistência, acesso fácil aos mesmos. Durante os testes, verificou-se que a água translúcida gera problemas de aquisição dos dados. Para trabalhos futuros o projeto, versão 2 pretende expandir os pontos de coleta para diferentes locais no leito em que corre o Ribeirão Fundo Canoas, permitindo melhor prognóstico de aumento de seu nível nestes diferentes pontos.

Palavras-Chave: Monitoramento de Ribeirão, Internet das Coisas, Eventos Climáticos, Tecnologia da informação e comunicação, Governo Eletrônico.

Abstract:

Climatic events affect, with greater frequency, populations close to streamlets, as they have different behaviors from those of rivers. Currently, these rivers are monitored by the Civil Defense,

which periodically issues alerts to the population. With that, it generated this new research problem, where it is possible to expand and improve techniques for measuring the water level through devices linked to the concept of the internet of things (IoT). As they are compact, dynamic and consume little energy, they tend to be used in small and maintenance-free prototypes. Which allows installation in environments with difficult or limited access. A mini-microcomputer, ultrasonic sensor and a camera were used. After tests and discussions, it can be verified that the experimental prototype is able to add to the traditional methods of acquisition of these data, guaranteeing precision, persistence, and easy access to them. During the tests, it was found that translucent water causes data acquisition problems. For future work, the project, version 2, intends to expand the collection points to different locations on the riverbed where Ribeirão Fundo Canoas runs, allowing for a better prognosis of an increase in its level at these different points.

Keywords: Monitoring, Streamlet, Internet of Things, Climatic Events, information and communication technology, e-Government.

1. INTRODUÇÃO

Cada vez mais os desastres naturais estão se tornando um assunto presente na vida das pessoas mesmo não sendo atingidas e, cada vez mais, estão tornando-se frequentes afetando milhões em todo o mundo: inundações, furacões, terremotos, maremotos e incêndios. Segundo a Organização das Nações Unidas (ONU), em 2018, há mais desabrigados por causa dos desastres naturais do que por conflitos e violência. O crescimento populacional, ocupação desordenada, aumento da pobreza e a destruição da natureza são uns dos requisitos que colocam cada vez mais pessoas no caminho do perigo.

No Brasil especialmente em Santa Catarina os desastres naturais mais frequentes são as cheias e os deslizamentos, inúmeras medidas já foram ou estão sendo tomadas para prevenir, algumas destas recorrem à tecnologia da informação que é de suma importância para monitorar a situação decorrente (MARCELINO, 2006). Através de uma rede de sensores espalhados por diversas áreas, os quais monitoram os níveis dos ribeirões, informando e alertando a população sobre eventual mudança de nível que venha afetar a mesma. Entretanto, nas áreas onde há ausência de sensores, a leitura ainda depende da ação humana o que pode tornar as informações imprecisas e com atrasos (STUHLER, et al. 2012).

Para complementar as ações de Governo Eletrônico ou e-Government tornou-se possível aplicar novas tecnologias é possível aumentar ou criar uma rede de sensores que podem auxiliar no monitoramento os níveis de ribeirões informando a situação de forma rápida e em tempo real (LOFFI, et al. 2016). Estes dados serão extraídos por meio de um protótipo funcional, concebido através da

integração/comunicação entre Raspberry, sensor ultrassônico e câmera digital. Tendo a Raspberry exercendo papel de um mini microcomputador, armazenando e enviando os dados coletados pelo sensor e câmera.

Raspberry Pi se trata de um dispositivo possuinte de grande semelhança com desktops e/ou laptops. Porém por ser compacto, articulado e robusto ganha aplicabilidade em ambientes onde a utilização de dispositivos capazes de computar se faz necessário e recursos como disponibilidade de espaço, energia elétrica, controle de temperatura, entre outros são limitados. Existem diversos modelos a serem escolhidos para atender a demanda que lhe é proposta. Porém o Pi 3, Pi 3 B+ e Pi 4 possuem destaque por entregar desempenho superior em relação aos demais.

2. MATERIAL E MÉTODOS

Para ser possível a implementação desse projeto, foram utilizados equipamentos eletrônicos capazes de computar, capturar imagens e mensurar distâncias entre objetos através de ondas ultrassônicas, dentre outros exemplificados na tabela 1.

Figura 1: Tabela de materiais de consumos adquiridos para confecção dos protótipos.

Item	Descrição detalhada
1	Módulo RF LoRa1276 – 915MHz – 100mW – 20dBm – SPI. Transmissor de longo alcance (2 km até 40 km) e de baixo consumo energético. Responsável por garantir a transmissão entre dois (ou mais) dispositivos por via <i>Wireless</i> . Dispensando a necessidade de fios ao longo do percurso a ser enfrentado para garantir tal transmissão.
2	Raspberry PI 3 B+ microcontrolador que no exíguo espaço equivalente a um cartão de crédito, abriga processador, processador gráfico, <i>slot</i> para cartões de memória, interface USB, HDMI e seus respectivos controladores.
3	Módulo ESP32 NodeMcu. Microcontroladores de baixo custo e baixo consumo de energia, também é um sistema-em-um-chip com microcontrolador integrado, Wi-Fi e Bluetooth.
4	Adaptador Wireless Usb Wifi 150mbps Lan Bgn Com Antena.
5	Cartão de memória classe 4 Kingston SDC4/16gb micro SDHC 16gb, com adaptador SD.
6	Mini painel solar fotovoltaico 6 volts 180mA – 84x112mm. Utilizado para conversão de energia solar, de modo que possa ser armazenada em pilhas ou baterias recarregáveis.
7	TFmini Plus Localizador de alcance Lidar Sensor para Arduíno micro ponto único IP65 à prova d'água Anti-poeira UART I2C I / O 1 ~ 1000Hz.
8	Módulo de Câmera Raspberry Pi Câmera De Visão Noturna.
9	Módulo 4G + GPS Shield for Arduino para Raspberry PI 3 B+.

Fonte: Acervo dos autores

Além dos dispositivos e material citado na tabela acima, conta-se com uma máquina virtual hospedada no servidor pertencente ao Instituto Federal Catarinense (IFC) e alocado na Unidade Urbana. Anterior a qualquer processo à ser iniciado para com a coleta de nível da água do ribeirão, é necessário realizar a preparação e configuração do ambiente, o qual denominamos “Protótipo”.

Este Protótipo será constituído pelos itens 2, 5, 7 e 8 da tabela 1 (figura 1) e através do preparo desse ambiente – instalação do sistema operacional, conexão dentre dispositivos e sensores, ajustes iniciais pós-integração, dentre outros – e posterior, a execução de configurações relacionadas à segurança, automação e disponibilidade para possibilitar que evolua para um Protótipo funcional, resiliente, autônomo e independente.

Os materiais 1, 3, 4, 6 e 9 da tabela 1 serão configurados e programados para servir como dispositivos de expansão para o Protótipo. Por meio de comunicação ponto-a-ponto, será interligado o Protótipo e um, ou mais, dispositivo de expansão. Normalmente empregados em pontos de confluência, onde há necessidade do monitoramento de cada um daqueles que deságuam.

A metodologia deste trabalho foi fundamentada nas documentações: Compared with TFmini, TFminiPlus has the following advantages (2019); Debian Documentation (2020); GNU Bash (2017); MySQL Documentation (2019); OpenCV modules (2020); Python 2.7.18 documentation (2020); Raspberry Pi Documentation (2020); Reference manual for OpenVPN 2.4 (2020); e Ubuntu documentation (2020).

2.1 PREPARANDO O AMBIENTE

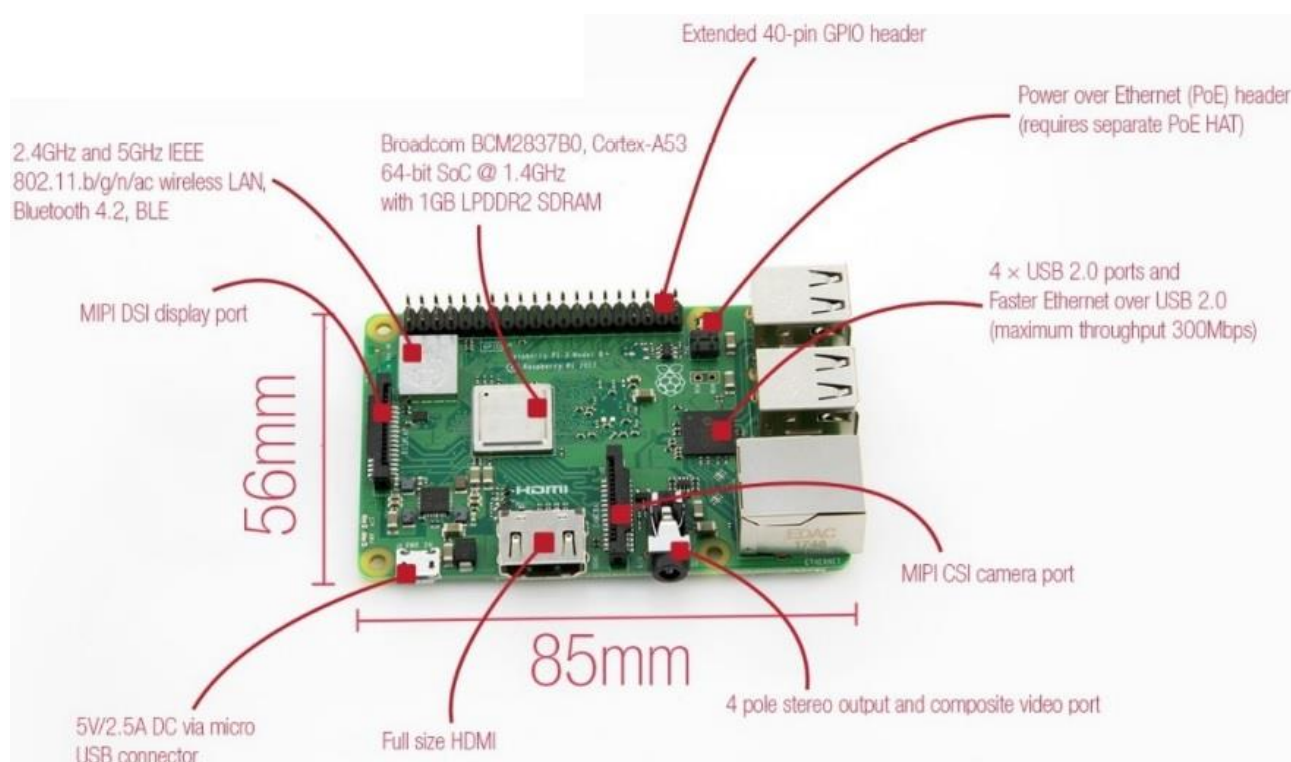
O preparo inclui a instalação, organização e criação de softwares, pastas, documentos e sistema operacional nos dispositivos computacionais e a integração entre sensores e esses dispositivos, seja por meio de cabos de cobre ou sinais de rádio.

2.1.1 Raspberry

A escolha do modelo Pi 3B+ (figura 1) se deu pelo custo-benefício. Equipada com um processador ARM operando a 1.4 gigahertz com 04 núcleos de processamento e 1024 megabytes de capacidade de armazenamento volátil (RAM).

Por meio de um cartão de memória (card storage – SD) torna-se possível a instalação de uma distribuição GNU/Linux, dentro de seus diversos sabores, foi selecionado a distribuição denominada Raspberry Pi OS (32-bit) Lite, disponibilizada pela fabricante. Essa distribuição causa um impacto (custo) computacional extremamente baixo, por não possuir interface gráfica do usuário (GUI) e nenhum programa de terceiros embutido no seu pacote de instalação.

Figura 2: Exemplificação visual simplificada dos periféricos de I/O, processamento, armazenamento e capacidade (RAM) de uma Raspberry Pi 4 (existem diferenças entre os modelos Pi 1, 2 e 3)



Fonte: <https://www.raspberrypi.org>

Desta forma, obtivemos um dispositivo capaz de computar instruções altamente funcional e poderoso (computacionalmente) para nosso propósito, resultando em consumo reduzido de energia elétrica para entregar o que lhe é solicitado.

Após instalar e configurar as funcionalidades básicas do sistema operacional (SO), por motivos de segurança é recomendado a criação de um novo usuário, contemplado com permissões de “super usuário”, e que seja desabilitado o usuário padrão do SO, chamado “pi”. Ainda na Raspberry, utilizou-se da linguagem de programação Python 2.7 e as bibliotecas PySerial, Statistics e MySQL Connector para possibilitar a comunicação com o sensor ultrassônico e a câmera. De maneira especial, ao utilizar câmeras não sejam módulos da Raspberry e possuam “Real Time Streaming Protocol” (RTSP) é indispensável a instalação da biblioteca “cv2” ou OpenCV.

Este dispositivo ainda conta com uma instância do banco de dados MySQL em operação, onde ganha o privilégio de “Master” em um sistema de replicação. Garantido o ACID (Atomicidade, Consistência, Isolamento e Durabilidade) e a independe de conexão simultânea com o servidor e realizando a sincronização enquanto houver conexão com o servidor.

2.1.2 Módulo Câmera

Também foi agregada ao Protótipo, um módulo câmera com resolução de 5 megapixel (figura 2). Tal resolução nos permite capturar imagens e gravar vídeos em qualidade de 1920x1080 a 30 quadros por segundo e 1280x720 a 60 quadros por segundo.

Figura 3: Módulo Raspberry Pi Câmera v2



Fonte: <https://www.raspberrypi.org>

2.1.3 Sensor Ultrassônico

Como uma alternativa para realizar a medição do nível de água dos ribeirões, optamos pelo sensor ultrassônico TFmini Plus (figura 3) fabricado pela Benewake LiDar. Capaz de emitir ondas sonoras há serem refletidas em uma distância máxima de 12 metros, quando direcionada para superfícies estáticas, como a de uma parede, e é reduzido seu alcance para 10 metros quando apontada para superfícies dinâmicas, como a de um riacho ou ribeirão.

Figura 4: TFmini Plus - 12m IP65



Fonte: <http://en.benewake.com/product/detail/5e746e294d839408076b6264.html>

Seu funcionamento se dá pela emissão de uma onda sonora que percorre o ambiente e ao colidir com uma superfície (estática ou dinâmica) é refletida, retornando ao emissor. Calculando o tempo gasto percorrido por esta onda no ambiente, é possível estimar, com precisão milimétrica, a distância entre o emissor e a superfície de colisão.

Na plataforma de versionamento disponibilizado pela empresa, <https://github.com/Tfmini/Tfmini-Plus>, há detalhamento técnico sobre o dispositivo.

2.1.3 Máquina Virtual no Servidor

Na sala da TI, no IFC Unidade Urbana, localiza-se o servidor da unidade. Neste há uma unidade de processamento para virtualização de sistemas, onde atualmente estão diversos sistemas operando. Um destes ganhou emprego neste projeto, operando como servidor de armazenamento e distribuição dos dados coletados pelos protótipos.

De maneira geral, no OS Ubuntu Server 16,04 LTS estão executados os serviços do MySQL, OpenVPN Server e php 7.4.9, que tornam possível traduzir os dados em informação disponibilizado para acesso externo por meio de páginas web ou consulta no banco de dados.

2.1.4 Configuração do Ambiente

Antes de iniciar qualquer configuração, em todos os dispositivos crie as seguintes pastas: `mkdir -p /opt/monitrib/image; mkdir -p /var/log/monitrib`, desta forma será possível transmitir arquivos entre os nós do túnel.

2.1.4 Cliente

O cliente, ou host, é responsável por consumir serviços disponibilizados por outro host (denominado servidor) que estejam interconectados. Neste projeto, conforme houver necessidade, o host poderá assumir o papel de servidor e vice-versa. A Raspberry, por se tratar de uma máquina destinada a permanecer em ambientes isolados, não haverá periféricos de entrada (mouse e teclado) e saída (monitor) conectados a todo momento. Para facilitar a total utilização desta, é necessário habilitar algum serviço de acesso remoto por terminal, já que nosso SO não contempla GUI. O Secure Shell, popularmente conhecido como SSH, atenderá todos nossos requisitos por ser simples, seguro, ágil e otimizado, resultando em desempenho e baixo custo computacional.

A fim de garantir segurança e privacidade, a comunicação entre os dispositivos e o servidor (onde há IP fixo) foi realizada por meio de um Túnel utilizando-se do serviço da OpenVPN. Via esta “rede privada virtual” torna-se possível trafegar dados como: SSH e seus derivados, comunicação com banco de dados, transferência de arquivos (imagens, logs, scripts, etc), como se estivessem conectados em uma rede local.

Após as configurações básicas do SO e aplicativos para controle remoto dos dispositivos, é realizada instalação da linguagem Python 2.7 e seus pacotes e/ou bibliotecas. Estas tornarão possível a codificação de instruções de alto nível para realizar a comunicação entre a Raspberry, sensor e câmera. Além de integrar os hardwares, a linguagem será responsável por inserir os dados coletados no banco de dados e sincronizar imagens capturadas com servidor, verificar conexões ativas, gerar logs das operações e executar tratativas para erros previamente previstos.

Com o propósito de garantir a atomicidade do cliente, foi habilitada a opção de sincronismo – ou replicação – entre instâncias de banco de dados, ou seja, a persistência de dados não dependerá da estabilidade e/ou garantia de conexão com o servidor para ser efetuada. Este processo é denominado Master and Slave (Mestre e Servo), tendo o cliente como master e o servidor, slave. Sempre que a conexão entre ambos for estabelecida e estabilizada, o próprio banco de dados inicializará a sincronização, clonando os dados existentes no master para o slave.

Durante os processos de adequações e calibrações do Protótipo, observou-se a necessidade de torná-lo autônomo, ou seja, em caso de alimentação pobre de energia ou reinicializações sem programação o Protótipo voltaria a operar normalmente após completa inicialização. Logo, todos os softwares necessários para o correto funcionamento do cliente foram configurados para inicialização automática: código Main (coleta de dados), OpenVPN e MySQL.

2.1.5 Servidor

O servidor é uma máquina física destinada a fornecer serviços a outros dispositivos (denominados host) conectados a uma mesma rede, nesta aplicação, será destinado a armazenar grande quantidade de dados e imagens, além de disponibilizar esses para o acesso via internet por meio de uma página web.

Após configuração apropriada do banco de dados, é possível por meio do Apache publicar uma simples página web implementa em php. Onde consta um gráfico com as coletas de nível da água e informações pertinentes ao Projeto. Em ambos os ambientes, host ou server, foram implementadas configurações similares como habilitar a função de acesso via SSH por meio de autenticação por chave pública, dispensando a necessidade de inserção de senha de autenticação a

cada tentativa de transmissão de arquivos ou execução de códigos remotamente. A criação de novos usuários, com permissões diferenciadas, para manuseio do banco de dados e por segurança, desabilitado usuário “root”.

Também foi empregada a ferramenta Git, proporcionando o versionamento total de arquivos de configuração, documentos de códigos, documentos de textos, entre outros arquivos. Desta maneira, garantiu-se a manutenção constante, descentralização e fácil auditoria de tudo que foi realizado neste projeto.

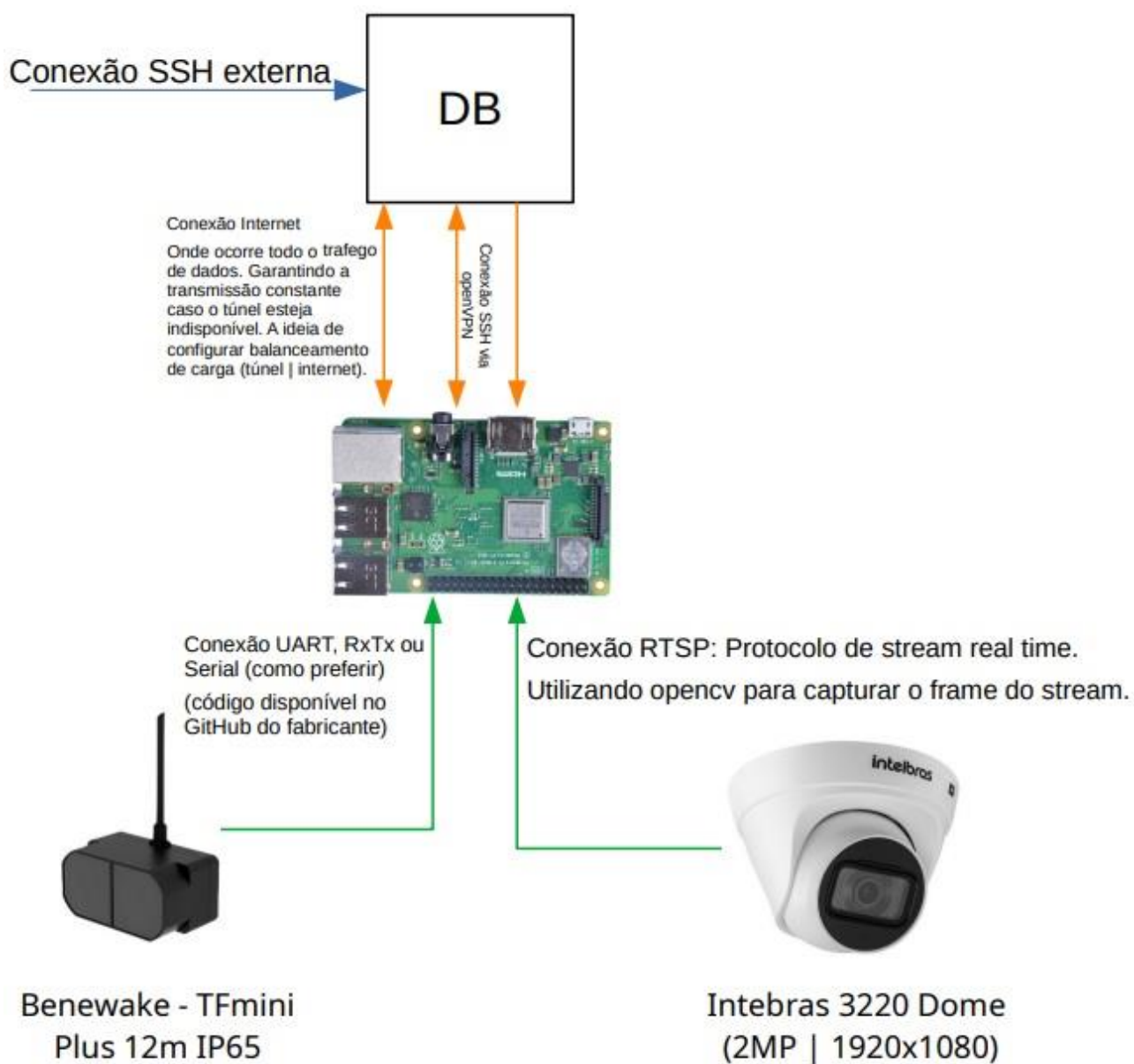
3. FUNCIONAMENTO DO PROTÓTIPO

Abaixo será explicado como os itens trabalham em conjunto para o desempenho do protótipo:

- OpenVPN: Tem a função de garantir a comunicação entre DB→Rasp para facilitar update, crashView, etc.
- Dados: Utilização de fog e, por enquanto, armazenados em MySQL (hora/data, metragem, path da imagem).
- Banco: MySQL apenas no lado DataBase. Para focar a raspberry no consumo de energia.
- Sensor ultrassônico: Manipulando os dados coletados pelo sensor ultrassônico para evitar erros (-1, 0, 10 ou 25000mm de nível), juntamente com scripts de backup de dados da memória, para o uso do Fog (realiza computação distribuída, serviços de rede e armazenamento, além da comunicação entre cloud computing data centers). Caso a raspberry dê erro, quando o algoritmo retornar, continua de onde parou.
- Imagem: Coleta da câmera via RTSP (transferência de dados em tempo real como áudio e vídeo) através do opencv (cv2) (livraria computacional com módulos de processamento de imagens). Armazenando por um momento na Rasp e enviando para BD via SSH. Garanto o envio (ls -l, grep “name”) depois é deletado. Esse processo acontece apenas na raspberry.
- Daemons: Client OpenVPN e os algoritmos viraram Daemons do sistema (programa executável em plano de fundo).

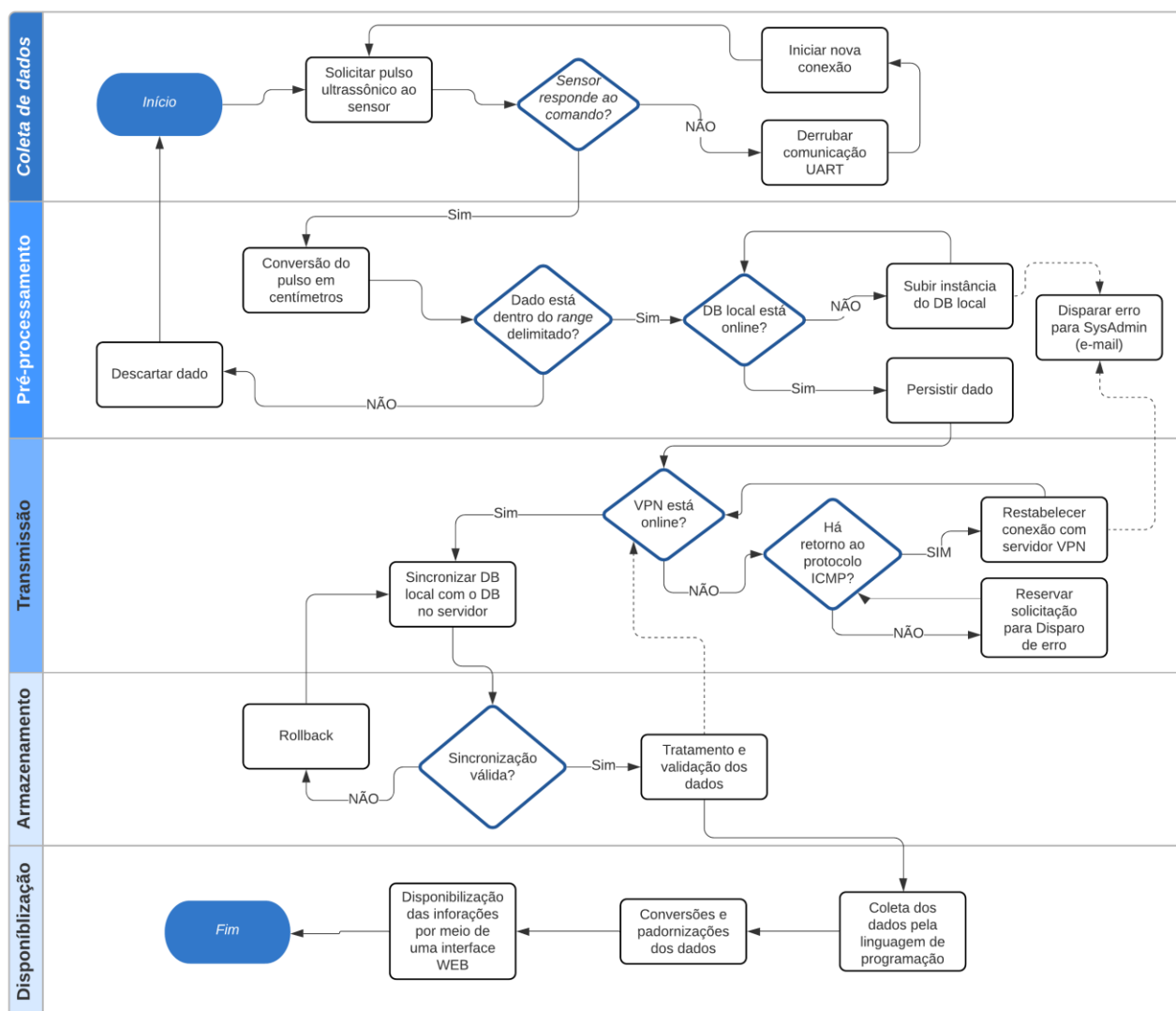
- Scripts / Algoritmos: Utilizando python 2.7, e também Shell Script.
- Conexão OpenVPN: Com objetivo de não perder comunicação por SSH com a Rasp. Dispensando IP fixo ou qualquer outra barreira.

Figura 5: Diagrama de funcionamento.



No fluxograma abaixo (figura 6) fica mais visível a ordem de funcionamento do protótipo:

Figura 6: Fluxograma funcional.



Fonte: Acervo dos Autores

O intuito dessa pesquisa é apoiar as ações de Governo Eletrônico nas prefeituras Municipais, está no grande número de cheias e enchurradas que correm nos municípios do Alto Vale do Itajaí, que precisam monitoramento dos Ribeirões para evitar a perda de vidas, que é o bem mais precioso. Nesse sentido duas delas se tornaram notícia em nível nacional e internacional, a catástrofe do Morro do Bau na região de Blumenau no ano 2008 e mais recentemente a catástrofe de Presidente

Getúlio em dezembro 2020, que com um sistema de sensores poderia ter emitido alertas a população que nunca tinham presenciado algo similar em anos de colonização.

4. RESULTADOS E DISCUSSÃO

A importância do trabalho para apoiar as ações de Governo Eletrônico nas prefeituras Municipais, está no grande número de cheias e enxurradas que correm nos municípios do Alto Vale do Itajaí. Com isso, surge a necessidade de monitoramento dos Ribeirões, pois são afluentes dos rios que atualmente possuem monitoramento realizado pela Defesa Civil. Porém, este monitoramento não auxilia os moradores próximos aos Ribeirões, que apresentam elevação de nível muito rápido por conta de sua estreita calha. Assim, é necessária a medida preventiva e de alerta antecipado para monitorar estas elevações.

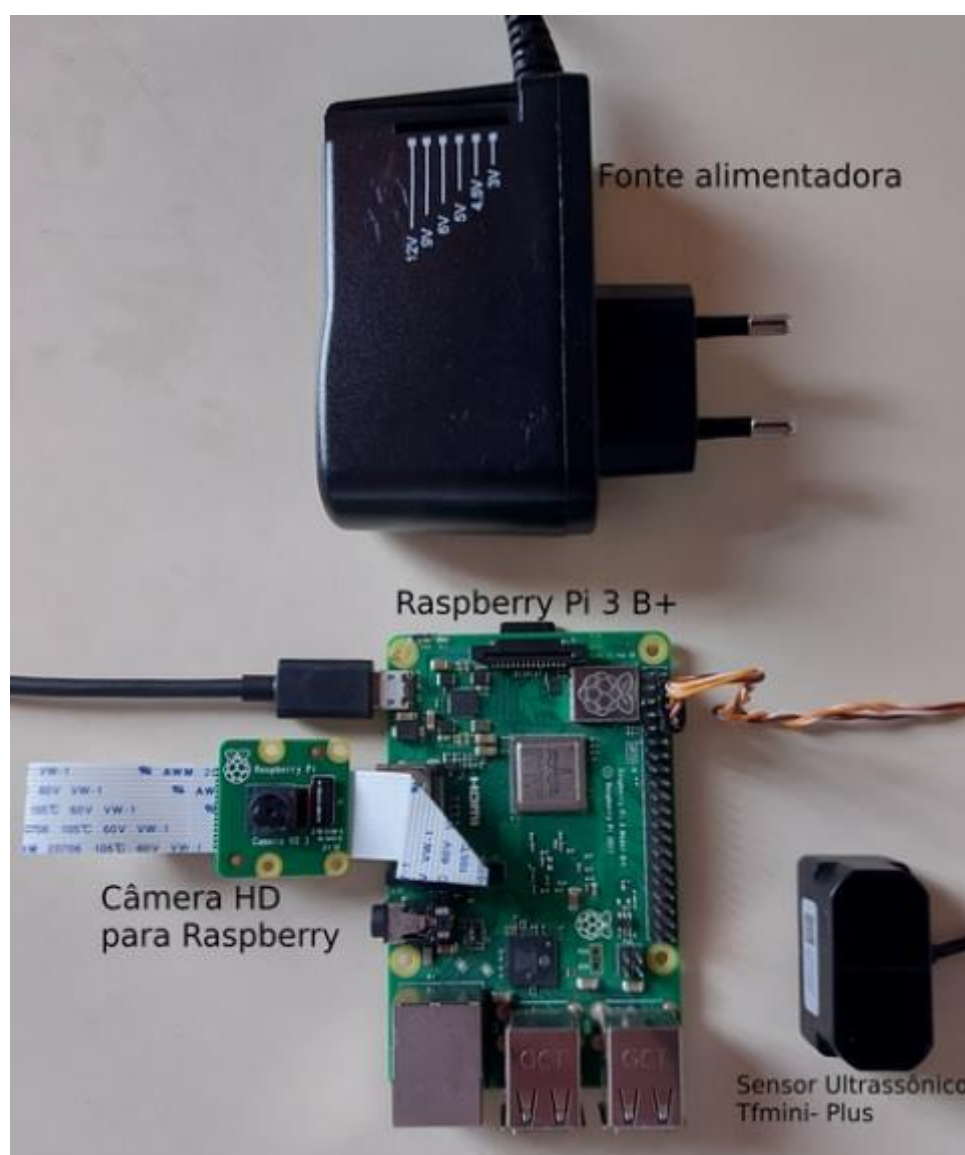
A utilização da rede de sensores atrelados a protocolos relacionados a internet das coisas, se faz necessário por suas características como: baixo tempo de resposta, baixo custo e baixo consumo de energia. O que os tornam ideais para a utilização em projetos dinâmicos, possuindo capacidade de adaptar-se às alterações como: perda de nó ou mau funcionamento, vandalismo ou mudanças de posição, com a possibilidade de integração com um gateway por uma rede externa para trafegar dados utilizando-se do modelo TCP/IP.

A Montagem de protótipo experimental com uso de tecnologia da informação de comunicação para monitoramento de Ribeirões em casos de cheias foi realizada com uso de um mini microcomputador Raspberry PI 3 B+, sensor ultrassônico Tfmini Plus e módulo de câmera HD para Raspberry (a lista completa dos dispositivos consta na tabela da figura 1). Também foi utilizado o servidor de virtualização, localizado no IFC – Unidade Urbana, para receber e processar os dados enviados pelo protótipo e disponibilizar para acesso externo, seja via web ou consulta no banco de dados.

Um ponto de coleta foi implementado e está disponível para consulta no link <http://monitrib.ifc-riodosul.edu.br/>, exemplo na figura abaixo. No protótipo foram testados dois tipos de sensores que podem ser aplicados em nossa região, tais como o sensor HC-SR04 e Tfmini Plus. Onde, segundo os fabricantes, o sensor HC possui alcance máximo de 4 metros e precisão de $\pm 0,5$

centímetro, contra 12 metros de alcance e precisão de ± 5 centímetros do Tfmini. De modo geral, ambos os sensores operam de maneira semelhante, apesar de possuírem frequências diferentes para emissão da onda ultrassônica, sendo um não ajustável e outro ajustável, respectivamente.

Figura 7: Protótipo experimental: Fonte de alimentação, câmera HD para Raspberry, Raspberry pi 3 B+ e sensor ultrassônico Tfmini Plus (conectado a Raspberry via GPIO).



Fonte: Acervo dos Autores

Desta forma, é possível destinar a aplicação adequada para cada sensor. Definindo a distância correta entre o ponto de fixação do sensor e o nível máximo em que água pode alcançar, a

fim de evitar danos ao sensor que possam ocasionar mau funcionamento ou interrupção de novas coletas.

Figura 8: Dados referentes ao nível de água coletado pelo Protótipo.



Fonte: Acervo dos Autores

Ao testar ambos os sensores no ambiente de testes com água translúcida, notou-se a uma elevada taxa de perda de eco (leitura nula) e leituras inválidas (valores discrepantes). Isso é explicado pela composição da água: nutrientes, organismos, vegetação, tipo de solo, entre outros, que não formam uma “barreira”, facilitando assim ultrapassagem da onda ultrassônica e mensurando distâncias não desejadas. Esta taxa varia de 40% a 62%, com ambos os sensores posicionados com aproximadamente a metade do seu alcance máximo (± 200 centímetros para o HC, ± 6000 centímetros para o Tfmmini).

Porém, ao alterar a composição desta água no ambiente de teste, ao ponto de torná-la não translúcida, houve uma redução nas taxas de perda de eco e leituras inválidas, permanecendo entre 0,5% e 5,7%. De maneira geral, ambos os sensores desempenham bom funcionamento quando direcionados para mensurar o nível de água do ribeirão, retornando medidas precisas e confiáveis. Pois, em sua grande maioria possuem águas de aparência não translúcida e de agitação constante.

Para a instalação do Protótipo equipado com sensor ultrassônico em águas extremamente translúcidas, recomenda-se a utilização de algum equipamento auxiliar que permaneça como alvo da onda e seja afetado pelas variações no nível da água onde esteja.

A pandemia atrapalhou todas as atividades acadêmicas, incluindo a pesquisa de campo, ainda assim, no ano de 2021 os códigos do projeto foram otimizados, e utilizando uma raspberry virtual foi possível realizar testes e verificar a eficácia do código, junto aos arquivos já capturados pelo sensor anteriormente e fazendo simulações para corrigir possíveis erros do protótipo. Para o futuro, pretendemos voltar a realizar testes de campo e captar o máximo de informações possíveis em ribeirões para aprimorar cada vez mais o protótipo.

Agradecimentos

Agradeço ao IFC Propi – Pró-reitoria Pesquisa e Inovação e Direção Campus Rio do Sul e CNPQ/MEC, pela concessão da bolsa e auxílio para aquisição dos materiais. Ao professor Jó Ueyama (USP), pelos ensinamentos e contribuições para o projeto. Ao analista de TI do IFC – UU, Fabiano, pela paciência e auxílio ao projeto. Ao time do projeto, pela parceria. Ao coordenador e coorientador de projeto, Fábio Alexandrini e André Alessandro Stein, pela oportunidade confiada.

REFERÊNCIAS

Compared with TFmini, TFminiPlus has the following advantages. Disponível em: <<https://github.com/TFmini/TFmini-Plus>>. Acesso em: agosto de 2020.

Debian Documentation. Disponível em: <<https://www.debian.org/doc/>>. Acesso em: agosto de 2020.

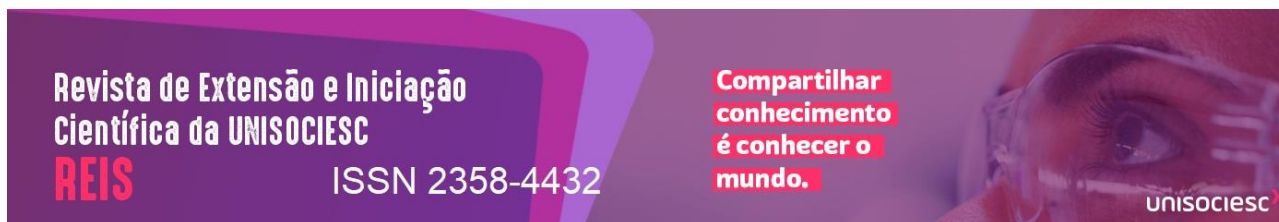
GNU Bash. Disponível em: <<https://www.gnu.org/software/bash/>>. Acesso em: agosto de 2020.

LOFFI, Leandro; ALEXANDRINI, Fábio; BRIGNOLI, Juliano T.; SOARES, Daniel Gomes; FAVERI, José Ernesto De. MONIT-RIO – TECNOLOGIA DA INFORMAÇÃO DE COMUNICAÇÃO PARA MONITORAMENTO DE RIOS EM CASOS DE CHEIAS. XIII SEGeT, 2016.

MARCELINO, Emerson Vieira; NUNES, Luci Hidalgo; KOBİYAMA, Masato. Mapeamento de risco de desastres naturais do estado de Santa Catarina. Caminhos de Geografia, v. 7, n. 17, 2006.

MySQL Documentation. Disponível em: <<https://dev.mysql.com/doc/>>. Acesso em: agosto de 2020.

OpenCV modules. <<https://docs.opencv.org/4.2.0/>>. Acesso em: agosto de 2020.



Python 2.7.18 documentation. Disponível em: <<https://docs.python.org/2.7/>>. Acesso em: agosto de 2020.

Raspberry Pi Documentation. Disponível em: < <https://www.raspberrypi.org/documentation/>>. Acesso em: agosto de 2020.

Reference manual for OpenVPN 2.4. < <https://openvpn.net/community-resources/reference-manual-for-openvpn-2-4/>> . Acesso em: agosto de 2020.

STUHLER, Jackson Alessandro; ALEXANDRINI, Fábio; ALEXANDRINI, Carla F. D.; FAVERI, José Ernesto De; ARAUJO, Thiago Souza. Utilização da Tecnologia Zigbee para Sensoriamento de Nível de Rio para Monitoramento de Cheias. IX SEGeT. 2012.

Ubuntu documentation. Disponível em: <<https://docs.ubuntu.com/>> . Acesso em: agosto de 2020.