

UNIVERSIDADE FEDERAL DO PARANÁ

JOHN WESLEY BOMFIM FARIA
SIMEON THOMAS BOMFIM DE FARIA

SISTEMA DE MONITORAMENTO REMOTO DE SOLOS

CURITIBA

2019

JOHN WESLEY BOMFIM FARIA
SIMEON THOMAS BOMFIM DE FARIA

SISTEMA DE MONITORAMENTO REMOTO DE SOLOS

Projeto apresentado como Trabalho de Conclusão de Curso, de Engenharia Elétrica com Ênfase em Sistemas Eletrônicos Embarcados.

CURITIBA
2019

TERMO DE APROVAÇÃO

**JOHN WESLEY BOMFIM FARIA
SIMEON THOMAS BOMFIM DE FARIA**

SISTEMA DE MONITORAMENTO REMOTO DE SOLOS

Projeto apresentado como Trabalho de
Conclusão de Curso, de Engenharia Elé-
trica com Ênfase em Sistemas Eletrônicos
Embarcados.

, pela seguinte banca examinadora:

Henri Frederico Eberspacher
Orientador

Rodrigo Jardim Riella
Avaliador

Ricardo Schumacher
Avaliador

Curitiba, 05 de Dezembro de 2019.

AGRADECIMENTOS

Agradecemos, primeiramente a Deus por estar sempre presente durante toda a caminhada acadêmica. Agradecemos a Universidade Federal do Paraná, e seu corpo docente, pela estrutura e conhecimento que nos foi compartilhado. Agradecemos a nossos pais, Hélio e Railda, pelo amor e incentivo incondicional, sendo os maiores responsáveis de chegarmos até aqui. E por fim, a todos que direta e indiretamente fizeram parte da nossa formação, o nosso muito obrigado.

A persistência é o menor caminho do êxito.

Charles Chaplin

LISTA DE ILUSTRAÇÕES

FIGURA 2.1 – TIPOS DE MOVIMENTOS DE MASSA	19
FIGURA 2.2 – MODELOS DE HORIZONTES DE PERFIL DO SOLO.	20
FIGURA 2.3 – DESLIZAMENTO DE TERRA SOB NA REGIÃO DE NOVA FRIBURGO-RJ EM 13 DE JANEIRO DE 2011.	22
FIGURA 2.4 – DESLIZAMENTO DE TERRA SOBRE ESTRADA.	23
FIGURA 2.5 – ILUSTRAÇÃO REPRESENTANDO UM DESLIZAMENTO DO TIPO ROTACIONAL.	24
FIGURA 2.6 – MODELO HIDROMECAÂNICO DO COMPORTAMENTO DO SOLO EM ENCOSTAS	25
FIGURA 2.7 – DEFORMAÇÃO DO SOLO DESVIADA DO PADRÃO DE COM- PRESSÃO BASE DO TALUDE.	26
FIGURA 2.8 – DINÂMICA DE ESCOAMENTO DE ÁGUA NO SOLO.	26
FIGURA 2.9 – ETAPAS DOS MOVIMENTOS DE MASSA.	27
FIGURA 2.10 – ÁREA FRACA, ÁREA CISALHADA E SUPERFÍCIE DE CISA- LHAMENTO.	29
FIGURA 3.1 – ESTRUTURA PROGRAMAÇÃO ORIENTADA A OBJETOS	32
FIGURA 3.2 – ESTRUTURA PROGRAMAÇÃO ESTRUTURADA	33
FIGURA 3.3 – DIAGRAMA REPRESENTANDO O CONCEITO DE HERANÇA .	35
FIGURA 3.4 – REPRESENTAÇÃO DE UM PROCESSO DE REGRESSÃO. . . .	47
FIGURA 4.1 – DIAGRAMA REPRESENTANDO A ARQUITETURA GERAL DO PROJETO	49
FIGURA 4.2 – DIAGRAMA REPRESENTANDO A ARQUITETURA FUNCIONAL DO PROJETO.	50
FIGURA 4.3 – MÓDULO MPU6050.	51
FIGURA 4.4 – HIGRÔMETRO.	52
FIGURA 4.5 – SENSOR DE CHUVA.	53
FIGURA 4.6 – PLACA ESP32.	54
FIGURA 4.7 – DIAGRAMA REPRESENTANDO A ARQUITETURA DOS COM- PONENTES DO PROJETO	56

FIGURA 4.8 – DIAGRAMA REPRESENTANDO AS TECNOLOGIAS PRESEB- TES EM CADA UM DOS COMPONENTES DO PROJETO	58
FIGURA 5.1 – MONTAGEM DOS DISPOSITIVOS DO PROJETO.	62
FIGURA 5.2 – INSTANCIA DA CLOUDMQTT.	62
FIGURA 5.3 – CLOUD MQTT RECEBENDO MENSAGEM DO COLETOR COM AS LEITURAS DOS SENSORES.	63
FIGURA 5.4 – ESTACA ONDE SERÃO FIXADOS OS SENSORES.	64
FIGURA 5.5 – PLATAFORMA DE SIMULAÇÃO DO PROJETO.	64
FIGURA 5.6 – PLATAFORMA DE SIMULAÇÃO DO PROJETO.	65
FIGURA 5.7 – PROTÓTIPO DA PÁGINA INICIAL DA APLICAÇÃO.	66
FIGURA 5.8 – PROTÓTIPO DA PÁGINA DA APLICAÇÃO EXIBINDO AS LEI- TURAS FEITAS EM TEMPO REAL.	67
FIGURA 5.11 – PROTÓTIPO DA PÁGINA DA APLICAÇÃO NO EXIBINDO O ARMAZENAMENTO DE DADOS.	68
FIGURA 5.9 – PROTÓTIPO DA PÁGINA INICIAL DA APLICAÇÃO PARA DIS- POSITIVOS MÓVEIS.	69
FIGURA 5.10 – MODELO DE RECEPÇÃO DOS DADOS DO COLETOR.	70

LISTA DE TABELAS

1	CLASSIFICAÇÕES DE MOVIMENTOS DE MASSA (VARNES, 1978).	16
2	TIPOS DE MOVIMENTOS DE MASSA (VARNES, 1978)	18

LISTA DE ABREVIATURAS E SIGLAS

WWW	World Wide Web
Web	Teia ou rede
CSS	Cascade Style Sheet
HTML	Hyper Text Markup Language
I/O	Input/Output
IoT	Internet of Things
LDR	Light Dependent Resistor
IAEG	International Association of Engineering
DMP	Digital Motion Processor
JSON	JavaScript Object Notation
XML	Extensible Markup Language
LCD	Liquid Crystal Display
PWA	Progressive Web App
MQTT	Message Queuing Telemetry Transport

SUMÁRIO

1	INTRODUÇÃO	12
1.1	MOTIVAÇÕES PARA O TEMA DO TRABALHO	13
1.2	OBJETIVOS	14
1.2.1	Objetivo Geral	14
1.2.2	Objetivos Específicos	14
1.3	ESTRUTURA DO TRABALHO	14
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	TEORIA SOBRE DESLIZAMENTO DE SOLOS	16
2.2	MODELAGEM DE SOLOS PARA DESLIZAMENTOS	20
2.3	DESLIZAMENTO DE ENCOSTAS	21
2.3.1	Processo de Ruptura e Deslocamento de Massa	27
2.3.1.1	Etapa de Pré-Ruptura	27
2.3.1.2	Etapa de Ruptura	28
2.4	DESLOCAMENTO DE TALUDE	30
3	TECNOLOGIAS	31
3.1	PROGRAMAÇÃO ORIENTADA A OBJETOS	32
3.2	PROTOCOLO HTTP	35
3.3	PROTOCOLO MQTT	36
3.4	PROTOCOLO I2C	37
3.5	LINGUAGENS DE PROGRAMAÇÃO	37
3.5.1	MicroPython	37
3.5.2	JavaScript	37
3.5.3	HTML	38
3.5.4	CSS	38
3.6	FRAMEWORKS	38
3.7	PROGRESSIVE WEB APPS	39
3.8	SERVICE WORKERS	40
3.9	NODE.JS	41

3.10	APEXCHARTS	42
3.11	OBJECT RELATIONAL MAPPER - ORM	42
3.12	WEBSOCKETS	43
3.13	WEB COMPONENTS	43
3.14	DESENVOLVIMENTO POR CAMADAS	43
3.15	GIT	44
3.16	BANCO DE DADOS MONGODB	45
3.17	COMPUTAÇÃO EM NUVEM	46
3.18	ANÁLISE DE REGRESSÃO	46
4	PROJETO	48
4.1	ARQUITETURA DO PROJETO	48
4.1.1	Escolha dos Sensores	50
4.2	ESCOLHA DO COLETOR	53
4.3	PROGRESSIVE WEB APPS	55
4.4	ARQUITETURA DE COMPONENTES	55
4.5	ESCOHA DAS TECNOLOGIAS	56
4.5.1	Azure	56
4.5.2	Azure App Service	56
4.5.3	Armazenamento de Dados	57
4.6	ARQUITETURA DAS TECNOLOGIAS	57
4.7	REQUISITOS FUNCIONAIS	58
4.8	COMUNICAÇÃO ENTRE SENSORES E COLETOR	58
4.9	COMUNICAÇÃO COLETOR E CLOUD MQTT	59
4.10	PROGRESSIVE WEB APP COM VUE.JS	59
4.11	GERAÇÃO DE GRÁFICOS	59
4.12	ALGORITMOS DE PREDIÇÃO	60
4.12.1	TensorFlow	60
5	DESENVOLVIMENTO	61
5.1	MONTAGEM DOS DISPOSITIVOS	61
5.2	COMUNICAÇÃO ENTRE COLETOR E CLOUD MQTT	61
5.3	PLATAFORMA FÍSICA DO PROJETO	63

	11
5.4 PRODUÇÃO DA PAGINA <i>WEB</i> PARA EXPOSIÇÃO DE DADOS	65
5.5 ARMAZENAMENTO DE DADOS	68
6 CONCLUSÃO	71
REFERÊNCIAS	75

1 INTRODUÇÃO

Na Pedologia (Estudo do solo), os profissionais da área, como Geólogos e a Engenheiros Agrônomos precisam de parâmetros para criar soluções no âmbito ambiental, social e agrônoma. Para facilitar a aquisição desses dados que tem grande utilidade, este projeto tem como objetivo, criar uma ferramenta de auxílio a esses profissionais que necessitam de dados do solo, a qual irá captar grandezas, interpretar, e por fim expor em uma página *Web* e armazenando esses dados para que seja possível obter análises estatísticas e temporais detalhadas sobre solos. Nos últimos anos estão sendo realizados estudos com base na literatura de geologia para que seja possível realizar a predição de eventos que afetam direta e indiretamente o ser humano com base em modelos matemáticos, e informações obtidas por meio de sensorização. A ferramenta que será desenvolvida neste trabalho pode ser uma grande aliada no que se trata de deslizamento de solos, que nos centros urbanos causam grande prejuízo material, ambiental e mortes.

Os deslizamentos de terra afetam a população e o meio ambiente. Nos últimos anos, o Brasil vem registrando diversos deslizamentos de encostas como por exemplo os eventos registrados em Santa Catarina em 2008, Angra do Reis e Morro do Bumba em 2010, e Teresópolis e Petrópolis em 2011 e 2012. Tais eventos demonstram que o poder público carece da instrumentalização e realização de medidas de contingenciamento para a minimização dos prejuízos sociais e ambientais causados, compreensão dos fatores desencadeantes e geração de um modelo de predição para tais eventos.

O sistema de monitoramento irá captar informações do solo e climáticas relevantes e expor em uma página *Web* na qual o profissional terá uma visão geral dos dados em tempo real, análises estatísticas das medições por meio de gráficos, e emissão de alertas para parâmetros que indiquem perigo de deslizamento de terra ou anomalias nas propriedades do solo.

Os dados das predições serão baseados em estudos já realizados sobre o assunto levando em conta os principais fatores desencadeantes dos deslizamentos de solo em geral, e deve ser ajustado a cada tipo de solo e de cada região para obtenção de análises com menor índice de incerteza e um dispositivo com maior confiabilidade ao usuário.

1.1 MOTIVAÇÕES PARA O TEMA DO TRABALHO

No Brasil, eventos de deslizamentos de encostas e morros, ceifam vidas e causam grande prejuízo social e ambiental. Se tais eventos forem previstos, os prejuízos sociais e ambientais podem ser minimizados. Atualmente o poder público e profissionais da área de geotecnia não possuem muitas opções de ferramentas com a função de monitoramento de áreas de risco de desabamento e assim realizar estudos de fatores ambientais, geográficos e climáticos que desencadeiam esses eventos.

Com o uso de sensores, é possível obter informações sobre fenômenos ambientais como por exemplo o movimento de deslizamento de solos. Com auxílio da comunicação e da eletrônica embarcada, é possível fazer a integração do sistema de sensores com uma estação de monitoramento em tempo real, que irá acumular e expor esses dados úteis ao usuário.

Para se obter uma avaliação de risco, é necessário se obter dados referentes a estabilidade, por meio de modelos matemáticos, e parâmetros climáticos e físicos da geografia do ambiente, que são em geral fatores que influenciam diretamente nos eventos de deslizamento de solos. Porém, o acesso a esses dados em tempo real se torna ineficiente por falta de uma ferramenta unificada com monitoramento em tempo real e histórico de um ambiente com exposição de todos esses dados de forma clara com uma interface otimizada.

Através da tecnologia e sensoriamento, é possível produzir e fornecer aos profissionais e ao poder público uma ferramenta unificada de predição e análise de áreas com riscos de deslizamentos de encostas. Conseqüentemente, a ferramenta irá gerar um histórico de cada ponto de coleta de informações, gerando com os dados coletados análises estatísticas trazendo aos profissionais as informações como a umidade do solo, índice pluviométrico, ângulo da encosta que geraram tal evento e com essas informações gerar padrões que indicam alerta de deslizamentos que serão de grande valia ao poder público, defesa civil e principalmente à população localizada nessas áreas de risco.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Desenvolver um sistema de monitoramento para encostas baseada na tecnologia de sensorização que seja capaz de gerar dados de eventos úteis, bem como o respectivo painel de controle para visualização e acompanhamento.

1.2.2 Objetivos Específicos

Dentre os principais objetivos específicos, deste Trabalho de Conclusão de Curso são:

1. Desenvolver o mínimo produto viável do dispositivo de monitoramento;
2. Criar uma interface de visualização de dados multiplataforma;
3. Gerar gráficos com os dados obtidos;
4. Modelar uma arquitetura distribuída e escalável para a aplicação;
5. Avaliar a viabilidade de implementação de um algoritmo de predição de eventos de desabamento.

1.3 ESTRUTURA DO TRABALHO

O capítulo 1 (um) exibe a INTRODUÇÃO no qual é exibido o problema, os objetivos gerais e específicos e as razões para escolha do tema do trabalho. O capítulo 2 (dois) apresenta a FUNDAMENTAÇÃO TEÓRICA, expõe a teoria sobre deslizamento de massas, modelagens para esses deslizamentos e fala das tecnologias e conceitos que serão usadas para a resolução do problema de deslizamento de encostas.

No capítulo 3 (três) é apresentado as TECNOLOGIAS, onde são relatadas as linguagens de programação, conceitos e paradigmas usados para no dispositivo. O capítulo 4 (quatro) apresenta o PROJETO onde são relatadas a arquitetura, requisitos funcionais, e detalhes de comunicação.

No capítulo 4 (quatro) é apresentado o DESENVOLVIMENTO do trabalho, onde é relatada metodologia e como foram empregadas os conhecimentos científicos

do assunto e as tecnologias envolvidas de forma cronológica. e por fim é feita uma CONCLUSÃO, onde é abordada as conclusões dos resultados obtidos no projeto.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados a teoria sobre deslizamentos de solos, movimentos característicos, fatores decisivos e etapas presentes nesses eventos. Os tópicos presentes nessa capítulo são:

- Teoria sobre deslizamento de solos;
- Modelagem de solos para deslizamentos;
- Deslizamento de encostas;
- Processo de Ruptura e Deslocamento de Massa;
- Deslocamento de Talude.

2.1 TEORIA SOBRE DESLIZAMENTO DE SOLOS

has relacionadas a taludes e as características geológicas dessas falhas. A classificação proposta por (VARNES, 1978) é uma das mais utilizadas no mundo, e é considerada a classificação oficial da *International Association of Engineering Geology* - IAEG.

TABELA 1 – CLASSIFICAÇÕES DE MOVIMENTOS DE MASSA (VARNES, 1978).

TIPO DE MOVIMENTO	TIPO DE MATERIAL		
	ROCHA	SOLO	
		Predominante Grosso	Predominante Fino
Queda	Queda de rocha (Fall)	Queda de detritos (Debris)	Queda de solo
Tombamento (Topple)	Tombamentos de rocha	Tombamentos de detritos (Debris)	Tombamentos de solo
Escorregamento (Slide)	Escorregamento em rocha	Escorregamento de detritos (Debris)	Escorregamento de solo (Slide)
Espalhamento (Spread)	Espalhamento de rocha	Espalhamento de detritos	Espalhamento de solo
Escoamento ou Corrida (Flow)	Corrida de rocha	Corrida de detritos (Debris)	Corrida de lama

Fonte: (SILVA, 2007)

A classificação proposta por Varnes baseia-se no tipo de material transportado pelas encostas, e no tipo de movimento. Os materiais são classificados como rocha (*rock*), solos (*earth*) e detritos (*debris*), e os movimentos são classificados em quedas (*falls*), tombamentos (*topples*), escorregamentos (*slides*), espalhamentos (*spreads*), e escoamentos (*flows*) que são divididos de acordo com o conteúdo e velocidade do deslizamento (VARNES, 1978). A tabela 1 representa os tipos e as classificações de movimentos, usada pela IAEG (VARNES, 1978).

Na tabela 2 é representada as características dos principais tipos de movimentos de solo que foram propostas por Varnes em 1978.

A figura 2.1 representam os tipos de movimentos de massa classificados de maneira individual. Os movimentos são as quedas, corridas de massa, deslizamentos rotacional e translacional, rolamentos, e as subsidência e colapsos (VARNES, 1978).

Nelas é possível observar a diferença visual de cada tipo de deslocamento de solo e suas características físicas. Nesse trabalho em especial vai ser tratado do caso representado na Figura 2.3 que são os deslizamentos rotacionais e translacionais. Nas figuras é possível observar a diferença entre os tipos de movimentos de massa e suas características geográficas (VARNES, 1978).

Todos os anos ocorrem em todo o planeta, deslizamentos de solos, sob condições adversas de climas e condições do solo. Os mecanismos desencadeadores dos fenômenos de deslizamento são climáticos, adversidades humanas relativas a suas intervenções no meio ambiente tais como aberturas de estrada e escavações sem orientação de órgãos de fiscalização e profissionais habilitados como engenheiros e geólogos. Eventos como terremotos e abalos sísmicos de altas intensidades, podem também se tornar fatores desencadeantes para deslizamentos de solos (FILHO, 2012).

"As chuvas atuam como o principal agente físico no desencadeamento de deslizamentos no Brasil" (LIMA, 2002). Os grandes acidentes registrados ocorreram durante períodos chuvosos com características variando de acordo com a região. A água da chuva quando infiltrada no solo aumenta os valores de peso e pressão, reduzindo a resistência característica do solo. A água da chuva é um dos fatores responsáveis pelas variações dos níveis dos lençóis freáticos, que é uma das causas dos deslocamentos de encostas no Brasil (LIMA, 2002).

É importante a compreensão dos parâmetros hidrológicos e pluviométricos

TABELA 2 – TIPOS DE MOVIMENTOS DE MASSA (VARNES, 1978)

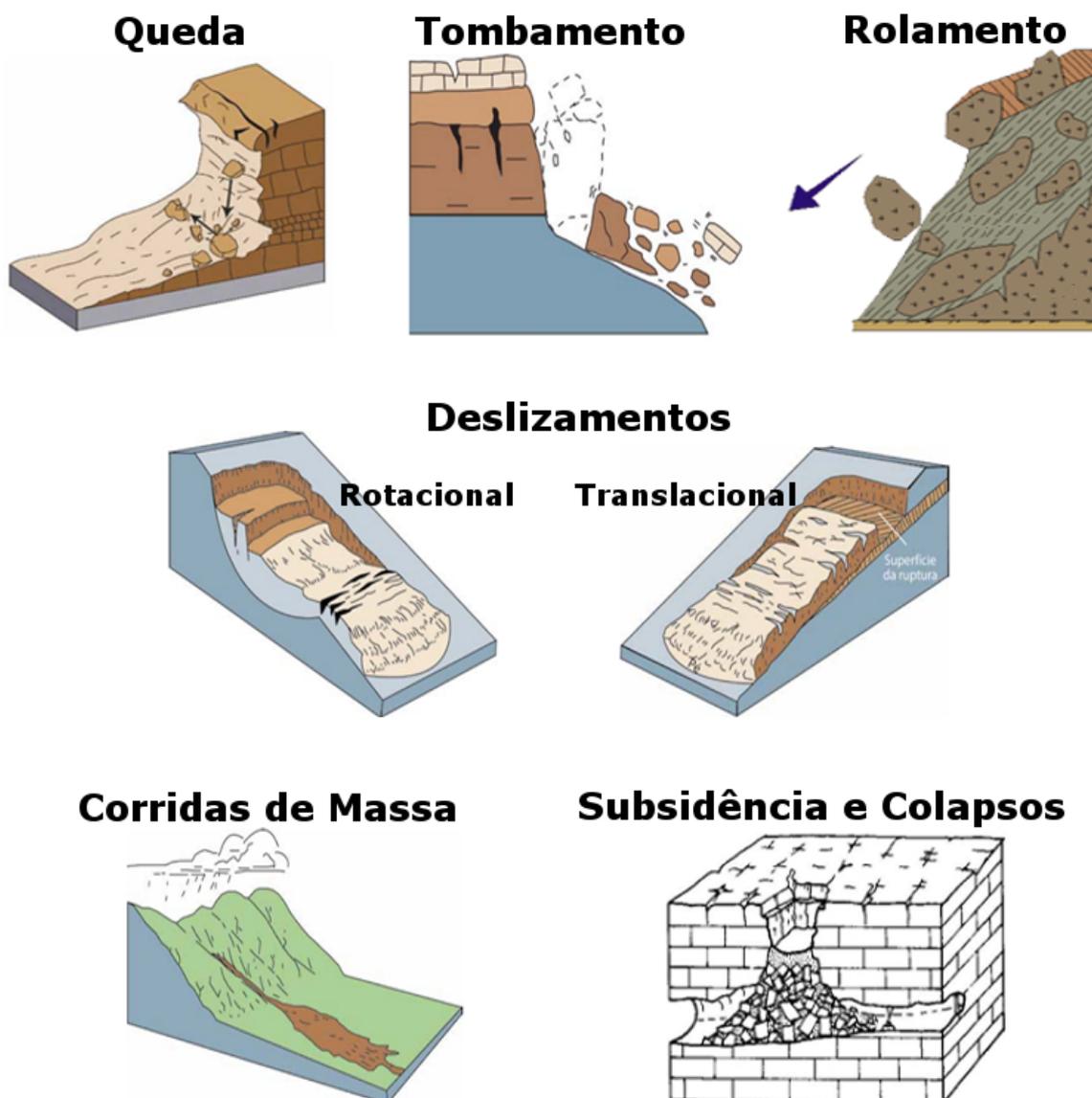
TIPO DE MOVIMENTO	CARCTERÍSTICA DO MOVIMENTO
Quedas (falls)	Os materiais se desprendem das encostas pela ação da gravidade, o movimento é de tipo queda livre com velocidade alta (m/s), atingindo grandes distâncias, Nas encostas íngremes o movimento é geralmente queda livre, e nas menos íngremes o movimento é de rolamento.
Escorregamentos (slides)	É o movimento que ocorre geralmente sobre uma superfície de ruptura. Os primeiros sinais são visíveis através de rupturas e fissuras na superfície do solo.
Tombamentos (topples)	O movimento inicia através de uma rotação de um bloco da encosta em torno de um eixo. Este movimento é condicionado a existência de planaltos frágeis.
Espalhamento (spread)	Ocorre em materiais mais pesados sobre camadas mais leves, gerando fissuras e fraturas transversais a direção de movimento. O movimento é repentino e se dá pela perda de resistência do material que se abriga na parte inferior (mais leve), como o efeito da liquefação do sedimento, ou também escoamento plástico, como ocorre nas argilas. O material da parte superior (mais pesado) pode sofrer movimentos de escoamento, desintegração, translação e rotação.
Escoamentos (flows)	São formas de escoamento de caráter hidrodinâmico causados pela perda de resistência do material, devido ao excesso de água. A massa passa a se comportar como um líquido viscoso e se desloca.

Fonte: (SILVA, 2007)

como fatores que desencadeiam deslizamentos de solos, com o monitoramento de processos hidrológicos e a compreensão dos mecanismos desencadeadores de deslizamento e ruptura de solos. É possível desenvolver modelos de previsão de deslizamento de encostas, sendo esses modelos gerados a partir de dados históricos obtidos pelo monitoramento (LIMA, 2002).

A definição de escalas temporais e espaciais é um dos grandes fatores causadores de problemas de monitoramento dos sinais de comportamento dinâmico das encostas. Por isso, a definição dessas escalas temporais e espaciais nesse projeto vai depender diretamente dos fenômenos observados em ambiente de simulação. A definição destes fenômenos determina as escalas e componentes de frequência de eventos para que seja possível determinar padrões nos dados que permitam o reco-

FIGURA 2.1 – TIPOS DE MOVIMENTOS DE MASSA



Fonte: (CEMADEN, 2016)

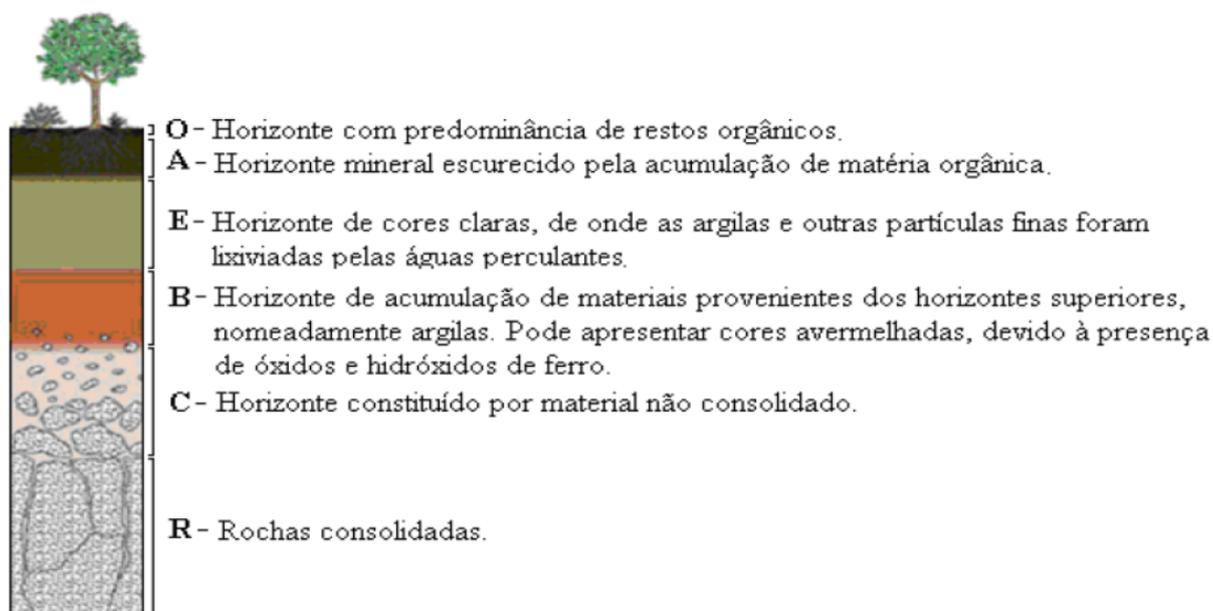
nhecimento dos tipos de fenômenos do solo que podem ser indicativos de possíveis eventos de deslizamento. As questões que devem ser consideradas são: que processos hidrológicos devem ser mensurados; quais as propriedades do solo que devem ser caracterizadas; quais os instrumentos mais adequados para o monitoramento, qual o intervalo de tempo ideal entre as amostragens dos sinais dos sensores (POPE, 1984).

A formação dos solos é resultante da ação combinada de 3 fatores sendo eles o clima que condiciona a ação da água da chuva e da temperatura, os materiais de formação do solo condicionados pela circulação interna da água sendo o conteúdo mineral e composição, e os organismos animais e vegetais que interferem no microclima,

modificando características químicas e físicas do solo (SILVA, 2007).

Os solos são constituídos de milhões de partículas de diferentes materiais de diversos tamanhos, como cascalhos, areias, salites e argilas. Parte destas partículas tem estrutura de grãos simples ou múltiplos de grãos agregados de matéria orgânica ou argila. Os espaços vazios entre as partículas do solo são denominados poros e podem estar totalmente ou parcialmente preenchidos com água. Os solos se encontram em extratos bem definidos e organizados, esse tipo de organização é estabelecida durante o processo de sua formação (NETTO, 2009), sendo estruturado por camadas que podem ser representadas pelo modelo ilustrado na figura 2.2:

FIGURA 2.2 – MODELOS DE HORIZONTES DE PERFIL DO SOLO.



Fonte: (LEROUEIL, 1996)

A estrutura exata do solo é definida pela presença ou ausência de cada horizonte, inclinação em relação ao campo gravitacional, e composição físico-química do solo.

2.2 MODELAGEM DE SOLOS PARA DESLIZAMENTOS

Há uma grande quantidade de configurações de solo que exigem uma modelagem específica para cada caso ou região a ser estudada e apesar da complexidade, a literatura disponibiliza modelos matemáticos capazes de descrever o comportamento

do solo. Como modelo difundido para deslizamentos rasos, pode de citar o modelo matemático de Shalstab (DIETRICH, 1998), que combina um modelo de estabilidade de encostas e de fatores hidrológicos, com objetivo de definir áreas suscetíveis a escorregamentos e deslizamento de terra.

O modelo de Shalstab resulta na equação 2.1 que leva em conta a chuva como fator crítico para deslizamento de solos e encostas (GUIMARÃES DAVID MONTGOMERY, 2003) e permite se obter a relação entre a quantidade de chuva crítica em milímetros e a transmissividade do solo à ser analisado.

$$\frac{Q}{T} = \left(\frac{\sin\theta}{\frac{a}{b}}\right) \left[\frac{C'}{\rho g z \cos^2\theta \tan\phi}\right] + \frac{s}{w} \left(1 - \frac{\tan\theta}{\tan\phi}\right) \quad (2.1)$$

Onde:

Q - Chuva crítica necessária para ruptura;

T - Transmissividade do solo (produto entre a condutividade hidráulica saturada e a espessura do solo);

a - Área de contribuição;

b - Comprimento da área de contorno;

C - Coesão efetiva do solo;

θ - Declividade local;

w - Densidade da água;

g - Aceleração da gravidade;

z - Espessura do solo;

s - Densidade global do solo saturado;

ϕ - Ângulo de atrito do solo.

2.3 DESLIZAMENTO DE ENCOSTAS

Engenheiros, Geólogos e profissionais da área de geotecnia e engenharia se apoiam em definições de definições de deslizamentos específicos com pequenas

diferenças. A diversidade de definições nessa área reflete a complexidade que esses eventos envolvem sendo necessário diversas áreas de estudo associadas a fenômenos de deslizamento. Os deslizamentos de encostas são processos diferentes da erosão laminar, ravinas e voçorocas que se definem pelo movimento gravitacional descendente para o lado externo da encosta, fazendo o transporte de materiais com auxílio da água, relevo, rochas e presentes no ambiente (LEROUEIL, 1996). Deslizamento é um termo genérico para se referir e descrever movimentos de massa na forma de solos, rochas ou materiais orgânicos sob o efeito de fenômenos climáticos, principalmente a chuva. O termo deslizamento também é usado para descrever a formação geológica resultante de um dos movimentos de terra. Além de deslizamento há outros termos para descrever tais eventos, como por exemplo, deslocamento ou ruptura de talude e deslocamento de massa, sendo esses termos aplicados a todas as proporções de evento (LEROUEIL, 1996).

Nas figuras 2.3 e 2.4 são expostos exemplos de deslizamentos de encostas com soterramento de casas e estradas. Nas duas situações as construções se encontravam muito próximas das encostas.

FIGURA 2.3 – DESLIZAMENTO DE TERRA SOB NA REGIÃO DE NOVA FRIBURGO-RJ EM 13 DE JANEIRO DE 2011.



Fonte: (CEA, 2011)

O deslizamento de encostas é caracterizado pelo movimento de descida de solo, rocha, ou ambos em declive, que ocorre a ruptura de uma superfície. Essa ruptura pode ser do tipo ruptura curva (escorregamento rotacional) ou plana (escorregamento translacional) (BOBROWSKY, 2008).

Para o tipo translacional de movimento de solo, o modelo de Shalstab é aplicável no caso de solos rasos. Deve-se observar, que em alguns casos os deslizamentos podem envolver outros tipos de movimentos, tanto desencadeamento da ruptura, quanto em eventos posteriores se as propriedades forem alteradas durante o movimento de material (MICHEL, 2011). Na figura 2.5 é possível observar um corte que sofreu um deslizamento do tipo rotacional.

FIGURA 2.4 – DESLIZAMENTO DE TERRA SOBRE ESTRADA.



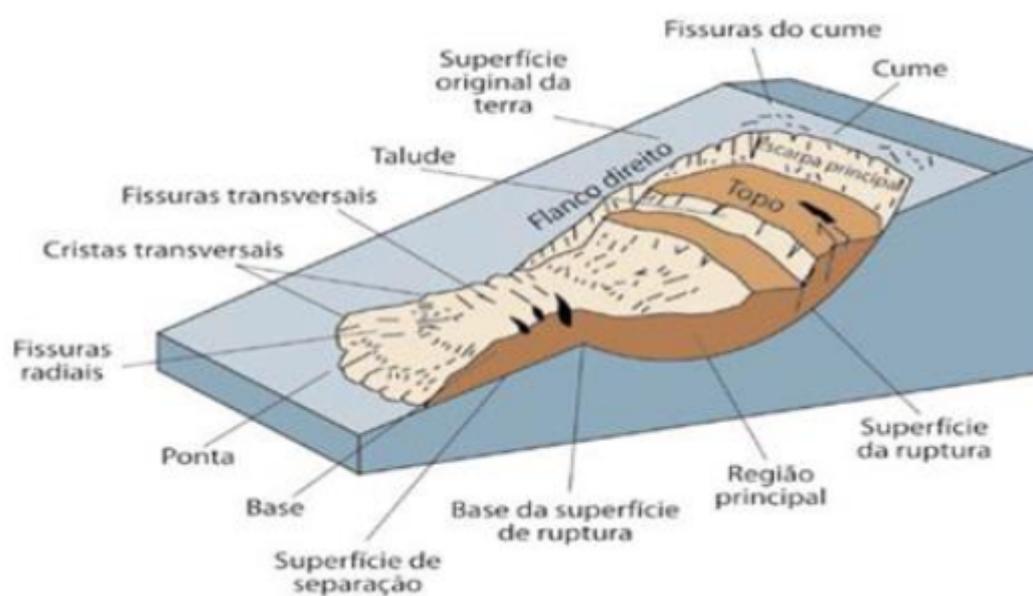
Fonte: (BOBROWSKY, 2008)

Quando ocorrem deslizamentos existem riscos que são caracterizados por perdas socioambientais resultantes da interação entre os perigos naturais, prejuízos ao meio ambiente, e perdas materiais. A equação 2.2 representa o risco sobre a população da maneira à seguir:

$$R = f(H, Pop, V) \quad (2.2)$$

Onde R representa o risco sobre a população, Pop representa a população, H representa o perigo (Eventos, fenômenos ou processos naturais que potencialmente

FIGURA 2.5 – ILUSTRAÇÃO REPRESENTANDO UM DESLIZAMENTO DO TIPO ROTACIONAL.



Fonte: (BOBROWSKY, 2008)

causam dano socioambientais) e V a venerabilidade (Capacidade que grupos ou indivíduos tem de lidar com perigos naturais).

Com base na estimativa de risco dada pela equação de R , é possível planejar medidas específicas para amenizar os efeitos e danos dos deslizamentos de encostas, avaliando a velocidade do movimento, extensão territorial onde os efeitos produzem danos. A determinação das variáveis e das unidades usadas dependem do tipo de ambiente natural, população envolvida e de mecanismos de contingenciamento utilizados, entre outros fatores (GOERL, 2011).

O objetivo desse trabalho não é definir a função f da equação 2.2, e sim desenvolver um sistema de monitoramento genericamente definido, por meio da construção do sensor e coleta de dados de um ambiente real, gerando um banco de dados de monitoramento suficiente para a compreensão dos fenômenos de deslizamentos e seus desdobramentos e efeitos internos na dinâmica do solo, sem a utilização de complexos modelos matemáticos pré definidos de processos geodinâmicos.

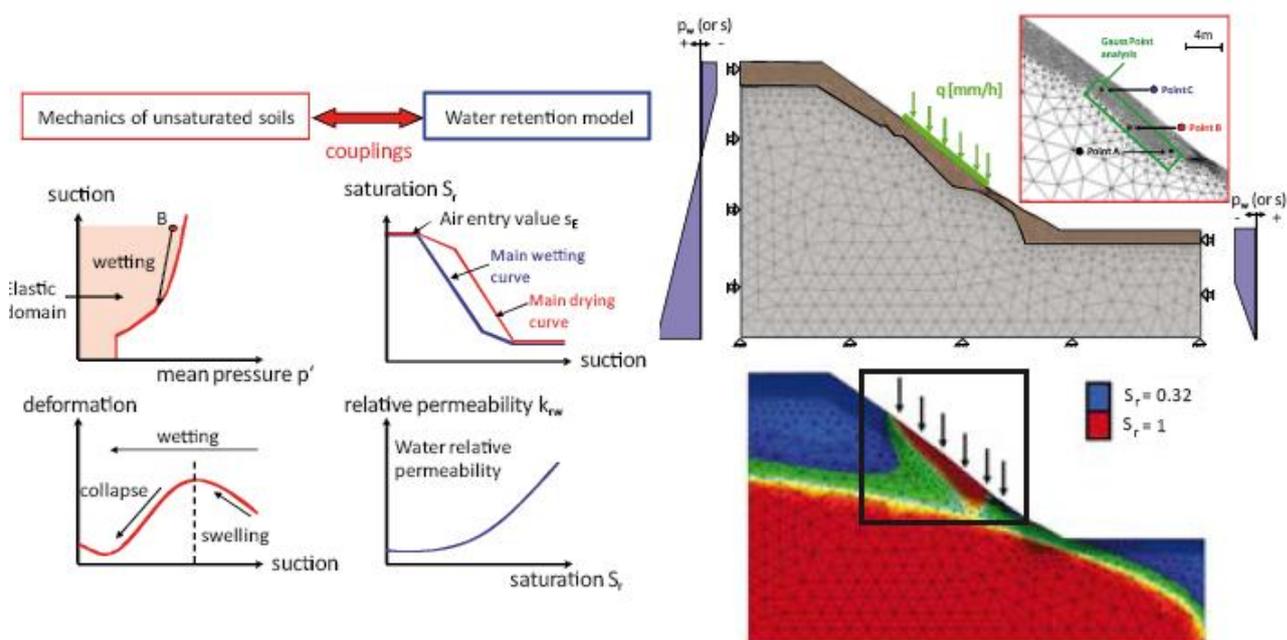
Há modelos hidromecânicos na literatura que tem como intuito prever o ponto crítico de deslizamento de solos sobre condições de permeabilidade e infiltração de chuva (ALVIM, 2011), um desses modelos pode ser observado na figura 2.6.

Deformações que estão fora do desvio padrão que indica a estabilidade são

apresentados na figura 2.7. Estas deformações são observadas na iminência do evento, com compressão das regiões no montante da massa de solo deslocada (BOBROWSKY, 2008).

É importante considerar que a saturação de água no solo contribui para o deslizamento e aumento dos riscos, devido ao aumento do peso do solo e a redução da resistência de cisalhamento nas superfícies de ruptura que facilita o desencadeamento de eventos de deslizamento (BOBROWSKY, 2008).

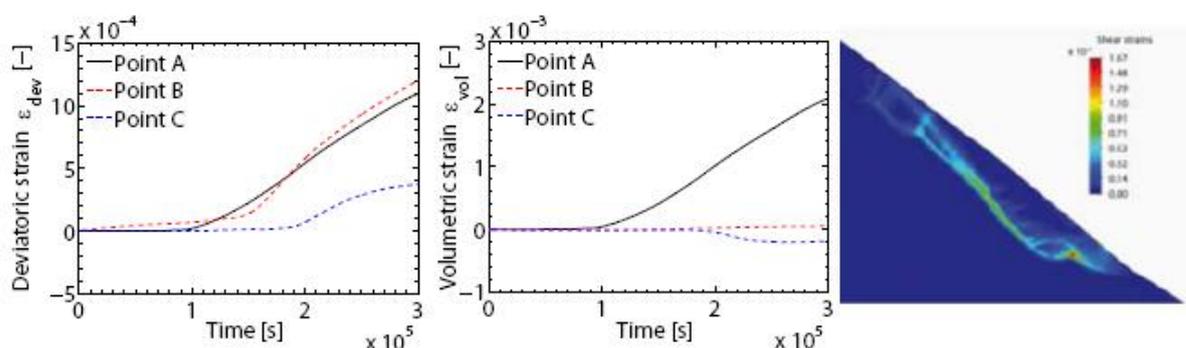
FIGURA 2.6 – MODELO HIDROMECAÂNICO DO COMPORTAMENTO DO SOLO EM ENCOSTAS



Fonte: (ALVIM, 2011).

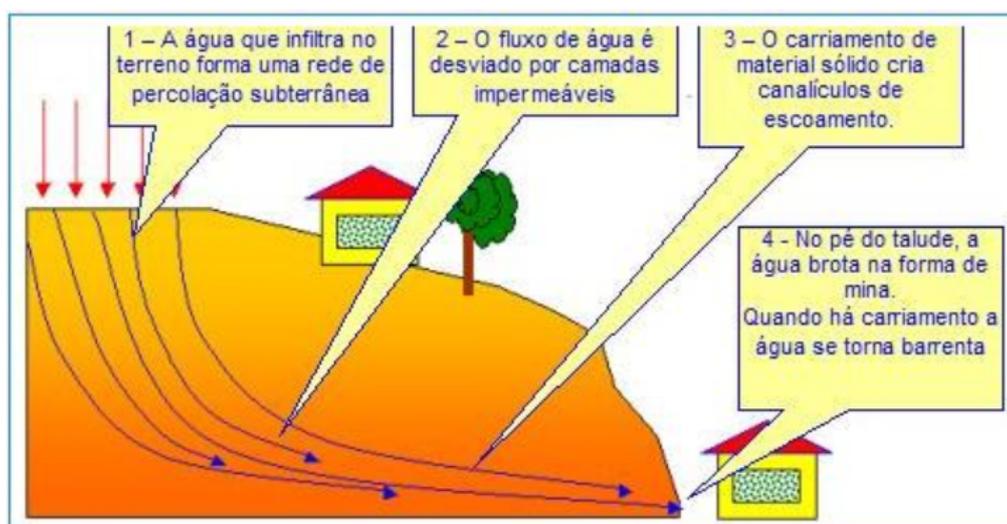
As características de saturação por água e drenagem do solo são decisivas para o controle e prevenção dos deslizamentos servindo como indicativo para eventos de deslizamento de solo (ALVIM, 2011). Com isso, conclui-se que a tecnologia de sensoriamento é útil para a caracterização de escoamentos de água que ocorrem na estrutura interna do solo. A figura 2.12 mostra a dinâmica de escoamento de água no solo.

FIGURA 2.7 – DEFORMAÇÃO DO SOLO DESVIADA DO PADRÃO DE COMPRESSÃO BASE DO TALUDE.



Fonte: (ALVIM, 2011).

FIGURA 2.8 – DINÂMICA DE ESCOAMENTO DE ÁGUA NO SOLO.



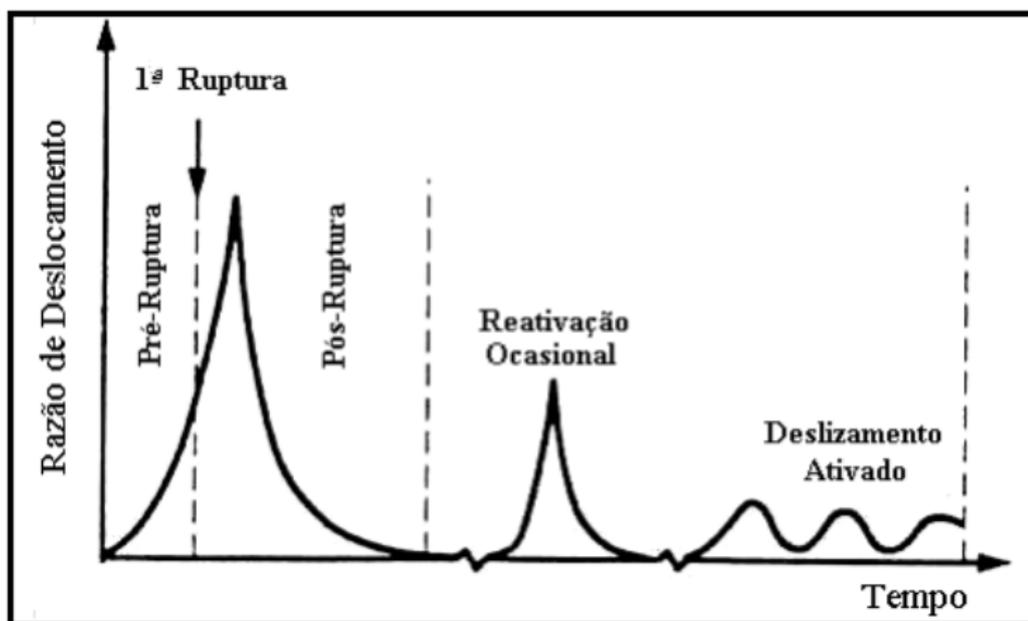
Fonte: (ALVIM, 2011).

Como não existem modelos genéricos que descrevam de forma unificada os tipos de terrenos e seus respectivos processos de desestabilização e os fatores envolvidos, deverão ser realizados estudos de diversos tipos de terrenos e encostas (morros), a fim de escolher as configurações de terreno que serão usados como objeto de estudo inicial com seus respectivos modelos descritivos da dinâmica de deslizamento de solo.

2.3.1 Processo de Ruptura e Deslocamento de Massa

Os movimentos de massa se organizam em 4 etapas, que são representados na figura 2.13. As etapas de deslocamento de massa são: pré-ruptura, ruptura, pós-ruptura e reativação (LEROUEIL, 1996).

FIGURA 2.9 – ETAPAS DOS MOVIMENTOS DE MASSA.



Fonte: (LEROUEIL, 1996).

2.3.1.1 Etapa de Pré-Ruptura

A primeira etapa é a de pré-ruptura, que inclui todo o processo de deformação e tem interferência de fatores como: deformações eletro-plásticas, acompanhado pelas mudanças nas tensões efetivas do solo, e por razões de deslocamento de massa, e aumento da velocidade de deslocamento, ao se aproximar do instante de ruptura, tendo como consequência uma deformação gradual e propagação da zona cisalhamento na massa do solo durante o período que se extender o evento de deslizamento do solo (SILVA, 2007).

Escorregamentos em taludes precariamente estáveis nas etapas de pré ruptura estão presentes em argilas moles, argilas rijas e em massas de rocha. O tempo de duração da etapa de pré-ruptura depende diretamente das características do solo e da tensão inicial do solo que pode ser extremamente variável (LEROUEIL, 1996).

A ruptura progressiva pode ocorrer em solos com diferença de resistência de pico e de resistência de pico e resistência de pós-pico do solo. As tensões que desencadeiam o cisalhamento geralmente não são uniformes, e quando as tensões cisalhantes alcançam a resistência de pico, ocorre a ruptura localizada. Como consequência, o solo rompido passa a suportar uma menor tensão cisalhante gerando um aumento de deformação. Quando a tensão cisalhante não é mais suportada pelos elementos que foram rompidos, essa tensão é repassada para os elementos vizinhos que tendem a romper em seguida. A transferência de carga cisalhante continuara até que o somatório das tensões de ruptura na superfície supere o valor de resistência do solo, e nesse momento ocorrerá uma ruptura generalizada. Após a ruptura generalizada, parte do solo atinge a resistência de pico e outra parte fica submetida a grandes deformações e modificações na sua resistência residual (LEROUEIL, 1996).

Para que aconteça o surgimento de uma ruptura progressiva é necessária a presença de algumas características como a fragilidade dos solos, a não uniformidade na distribuição das tensões de cisalhamento e tensões cisalhantes em locais que excedam a resistência de pico do solo (LEROUEIL, 1996).

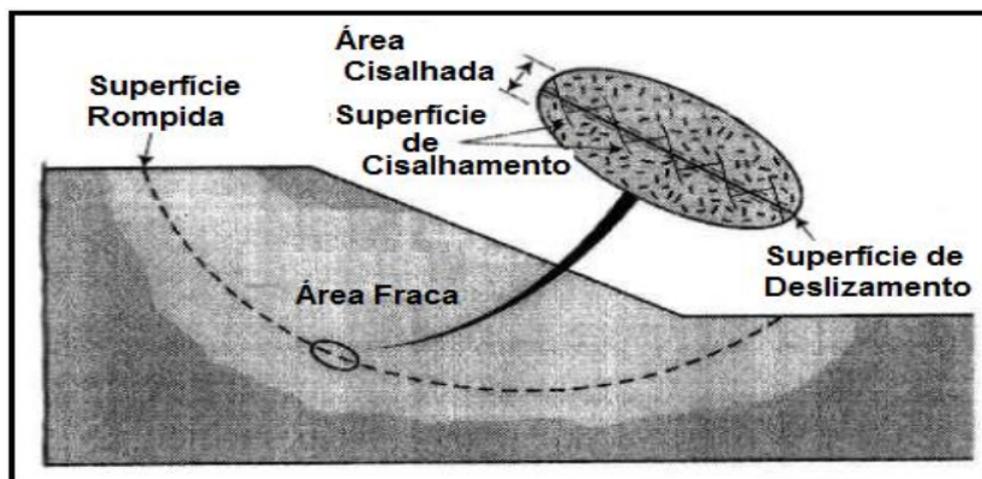
Em determinado momento do processo, pode haver equilíbrio na ruptura progressiva. Para que isso aconteça é necessário que o processo de ruptura reinicie ou se modifique. Os elementos que levam a essa situação são relacionados a uma variação na geometria do problema da dinâmica e variação das tensões cisalhantes (LEROUEIL, 1996). a diminuição da tensão normal efetiva, com consequência da diminuição do pico de resistência do solo, Variação na resistência residual e erosão na base ou no topo do talude (LEROUEIL, 1996).

2.3.1.2 Etapa de Ruptura

A etapa de ruptura é caracterizada pela formação da superfície de cisalhamento na massa de solo ou de uma área. É um processo complexo resultante de alguns fatores como por exemplo a velocidade da deformação e efeito de ruptura progressiva e descontinuidades (LEROUEIL, 1996).

A camada de solo em torno da superfície de cisalhamento que perde suas propriedades durante o processo de ruptura é denominada Área Fraca. A figura 2.10 é uma representação da área fraca, área cisalhada e superfície de cisalhamento.

FIGURA 2.10 – ÁREA FRACA, ÁREA CISALHADA E SUPERFÍCIE DE CISALHAMENTO.



Fonte: (LEROUEIL, 1996).

Primeiramente ocorre a formação da área cisalhante e em seguida se desenvolve a superfície de cisalhamento. As três áreas distintas de cisalhamento são: Área Fraca, onde o solo atinge a ruptura total tornando-se desestruturado, área cisalhante, que é adjacente à superfície e ao cisalhamento, com espessura variante em décímetros, dependendo do material e dos deslocamentos envolvidos, e superfície de cisalhamento, ao longo da qual existe a possibilidade de orientação de partículas durante o deslizamento (LEROUEIL, 1991).

O que caracteriza a formação inicial de superfícies de cisalhamento simples e isoladas, é a tensão de cisalhamento dos solos no processo de escorregamento, atuando mais intensamente em solos que compõe a zona cisalhante da área (SILVA, 2007).

A primeira etapa descreve o momento da pré ruptura e todo o processo que conduz a deformação à ruptura. A ruptura se caracteriza pelos seguintes fatores: Deformações Elasto-Plásticas, acompanhadas pelas mudanças nas tensões efetivas e por razões de deslocamento de massa, associada à ruptura progressiva (SILVA, 2007), Aumento da velocidade de deslocamento ao se aproximar da ruptura, tendo como consequência uma formação e propagação gradual da zona de cisalhamento na massa do solo (LIMA, 2002).

No fator 2, a energia potencial é liberada transformando-se em energia cinética. Nesse momento uma onda sônica é liberada e propaga-se pelo solo e esta onda poderá ser percebida por um conjunto de sensores acústicos, que podem fazer parte

do dispositivo para captação do evento em fase de pré-ruptura.

2.4 DESLOCAMENTO DE TALUDE

Define-se Talude como uma superfície inclinada em relação a horizontal que delimita uma massa de solo, sendo ele natural ou artificial. Sob condições específicas, uma porção de material pode deslocar-se desencadeando um processo que pode ser denominado genericamente por movimento de massa ao longo de uma superfície chamada de superfície de ruptura. A quantidade de água presente no solo é um fator agravante no que se trata do deslocamento de taludes, por aumentar o peso específico do solo o que influencia diretamente na perda da cimentação e resistência entre as partículas do solo (BITTENCOURT, 2018).

3 TECNOLOGIAS

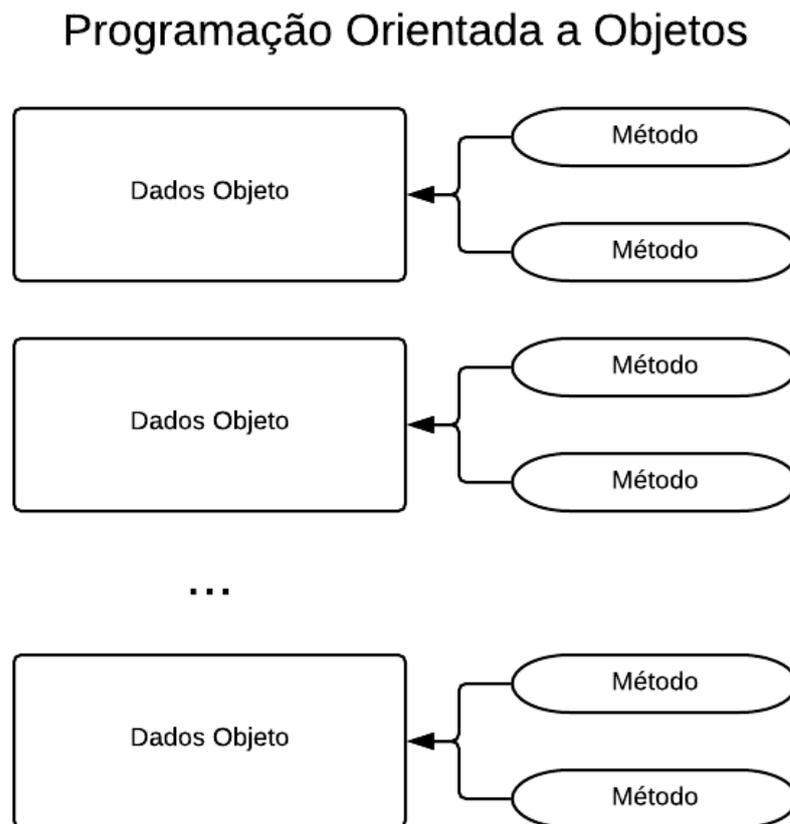
Esse capítulo apresenta as tecnologias e todos os conceitos envolvidos na produção do dispositivo de monitoramento de solos. Nele estão descritos linguagens de programação, bibliotecas, plataformas, nuvens públicas, e conceitos. O capítulo está dividido em:

- Programação Orientada a Objetos;
- Protocolo HTTP;
- Protocolo MQTT;
- Protocolo I2C;
- Linguagem de programação;
- Frameworks;
- Progressive Web Apps;
- Service Workers;
- NodeJS;
- Bibliotecas;
- Object Relational Mapper - ORM;
- Web Sockets;
- Web Components;
- Desenvolvimento por camadas;
- GIT.

3.1 PROGRAMAÇÃO ORIENTADA A OBJETOS

A programação orientada a objetos pode ser definida como um padrão de desenvolvimento que é seguido por linguagens como o Java e o C++. Existem diferenças entre a Programação Estruturada, que é o método usado na linguagem C, e a orientada a objetos. Na Programação Estruturada, as funções ou procedimentos são aplicadas globalmente na aplicação, já na orientação a objetos, existem métodos que são aplicados aos dados de cada objeto. Na figura 3.1 pode ser visto um diagrama que representa a estrutura de dados usada na Programação Orientada a Objetos (HORSTMANN, 2009).

FIGURA 3.1 – ESTRUTURA PROGRAMAÇÃO ORIENTADA A OBJETOS

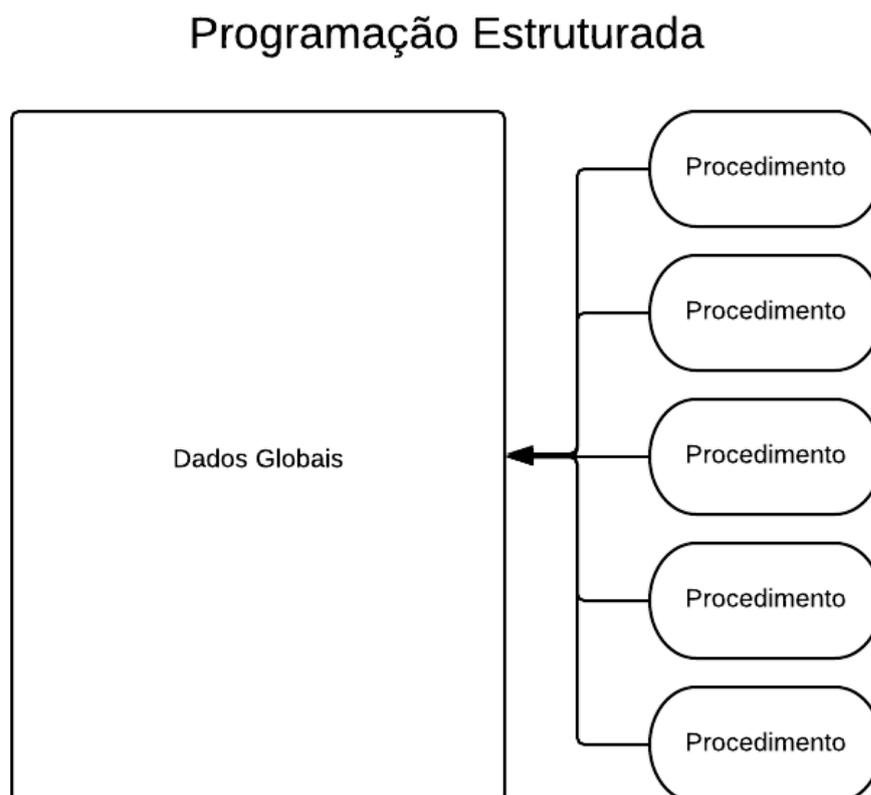


Fonte: (AUTORIA PRÓPRIA)

No início da computação, os computadores manipulavam tipos de variáveis primitivas com números e caracteres, e a medida que esses programas se tornaram maiores e mais complexos, passaram a manipular um número cada vez maior dessas

variáveis trazendo um alto grau de desorganização, detalhes e erros presentes nos códigos. A figura 3.2 representa a estrutura de dados da Programação Estruturada (HORSTMANN, 2009).

FIGURA 3.2 – ESTRUTURA PROGRAMAÇÃO ESTRUTURADA



Fonte: (AUTORIA PRÓPRIA)

Para resolver esse problema, entra o conceito de Programação Orientada a Objetos, que diminui a complexidade encapsulando cálculos e operações rotineiras criando abstrações de software ou funções que podem ser usadas sem a preocupação com os procedimentos necessários para que essas operações e cálculos sejam feitos. A figura 2.15 representa a estrutura da programação orientada a objetos (HORSTMANN, 2009).

A programação estruturada, quando feita corretamente possui um desempenho superior em processamento em relação à programação orientada a objetos, pelo motivo de ser executado sem concorrência, porém, devido ao poder do desempenho dos computadores na atualidade, e o fato do desempenho não ser prioridade em algumas

aplicações a orientação a objetos se difundiu gradativamente. Dentre as principais vantagens da Programação Orientada a Objeto estão a capacidade da representação do sistema próxima da realidade, e reutilização de códigos e organização do *software* (HORSTMANN, 2009).

Abstração

Abstração, é um pilar da Programação Orientada a Objetos, nas qual consiste em trabalhar com um objeto, preocupando-se somente com suas principais características sem se apegar a pontos acidentais, ou seja, abstrair um item do mundo real e transformá-lo em uma classe. O maior objetivo da abstração é simplificar o processo de programação (HORSTMANN, 2009).

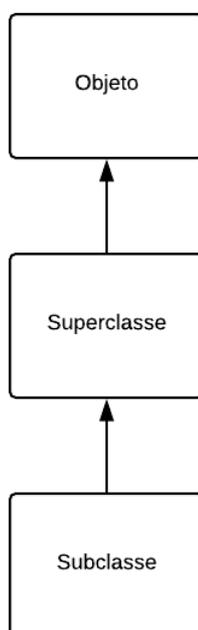
Herança

Herança é um mecanismo de aprimoramento de classes. Se for necessário implementar uma nova classe dentro do código, e já existir uma classe que representa um conceito mais geral, então a nova classe pode herdar atributos dessa classe existente. A herança é um mecanismo para estender as classes existentes, chamadas de superclasses, adicionando novos métodos e campos para uma ramificação menor chamada de subclasse (HORSTMANN, 2009). A figura 3.3 é um diagrama que representa a estruturação do mecanismo de herança.

Encapsulamento

O Encapsulamento é um mecanismo de restrição de acesso a alguns componentes do objeto, assim, os dados de uma classe são ocultos, tornando-os acessíveis somente por meio de métodos. O objetivo do encapsulamento é agrupar dados que se relacionam os agrupando, reduzindo assim as colisões entre nomes de variáveis e reunindo os métodos relacionados às suas propriedades. Por conta do encapsulamento, o conhecimento detalhado da implementação interna da classe é desnecessário do ponto de vista do objeto, sendo necessário o conhecimento somente para os métodos internos da classe. As principais vantagens do encapsulamento é que o processo de desenvolvimento de sistemas é acelerado e simplificado, já que os usuários dos objetos não precisam necessariamente saber como eles são constituídos internamente,

FIGURA 3.3 – DIAGRAMA REPRESENTANDO O CONCEITO DE HERANÇA



Fonte: (AUTORIA PRÓPRIA)

a segurança do código e a implementação de um comportamento pode ser modificada radicalmente sem que haja impacto no resto do programa (HORSTMANN, 2009).

POLIMORFISMO

Polimorfismo é o princípio que duas ou mais classes derivadas de uma classe que representam um conceito mais geral podem ter métodos com a mesma identificação e assinatura, porém com comportamentos distintos, específicos para cada classe. De maneira geral, o polimorfismo representa como os objetos de uma classe podem se comportar de maneiras diferentes (HORSTMANN, 2009).

3.2 PROTOCOLO HTTP

O protocolo *HyperText Transfer Protocol* (HTTP), é um protocolo de aplicação usado para transferência de documentos que englobam várias mídias como o HTML, JSON, XML e arquivos texto. O HTTP segue um modelo cliente-servidor clássico, no qual o cliente envia uma requisição e aguarda até que receba uma resposta. Sua principal aplicação é na comunicação entre navegadores *Web* e servidores possuindo grande flexibilidade (MOZILLA, 2018d).

Método de solicitação GET

O método GET solicita ao servidor que envie a página (ou objeto, no caso mais genérico; na prática, apenas um arquivo). A grande maioria das solicitações a servidores da *Web* tem a forma de métodos GET (TANENBAUN, 2003).

Método de solicitação POST

Em computação, POST é um dos muitos métodos de requisição suportados pelo protocolo HTTP. Esse método de requisição, foi projetado para solicitar que o servidor *Web* aceite os dados anexados no corpo da mensagem de requisição para armazenamento. Ele é normalmente utilizado quando se faz o *upload* de um arquivo ou envia-se um formulário *Web* completo (TANENBAUN, 2003).

3.3 PROTOCOLO MQTT

Para dispositivos que utilizam *Internet of Things* (IoT) a conexão à internet é um requisito, pois com ela, é possível que os dispositivos trabalhem entre si e com outros serviços de *backend*. Desenvolvido com base na pilha TCP/IP, o *Message Queue Telemetry Transport* (MQTT) tornou-se padrão para comunicações de IoT (YUAN, 2017).

O MQTT foi desenvolvido pela IBM na década de 90, com objetivo de vincular sensores em pipelines de petróleo a satélites. O MQTT é um protocolo de mensagens com suporte à sincronização assíncrona, leve e flexível que permite a sua implementação em hardware e redes com baixo desempenho (YUAN, 2017). O padrão de troca de mensagens do MQTT é o *publish/subscriber* (Publicador/subscritor). Nesse padrão quando se deseja receber uma determinada informação, é feita uma requisição para o broker, que é um elemento de rede que gere as publicações e subscrições, que é um intermediário no processo de comunicação. Para publicar informações ou responder essas requisições, também fazem uso do *broker* (intermediador), enviando as informações requisitadas (YUAN, 2017).

O *broker* traz como vantagem o desacoplamento entre as partes comunicantes do sistema, o que não é possível em modelos de comunicação do tipo cliente/servidor. A identificação das mensagens no MQTT se dá através de tópicos, onde os elementos

de rede enviam vários tópicos para o *broker*, e os elementos subscritores podem colher as informações requisitadas (YUAN, 2017).

3.4 PROTOCOLO I2C

O protocolo I2C foi desenvolvido pela Philips em 1996, ele é altamente difundido, e conecta dispositivos de controle inteligentes como por exemplo microcontroladores, microprocessadores e circuitos de uso geral como drivers LCD, e módulos de sensores (FILHO, 2018).

O protocolo I2C possui muitas vantagens, dentre elas a organização funcional em blocos, a não necessidade de desenvolvimento de interfaces, diagnóstico de falhas simples, facilidade de desenvolvimento, imunidade a ruídos, baixo consumo de corrente (FILHO, 2018).

3.5 LINGUAGENS DE PROGRAMAÇÃO

3.5.1 MicroPython

O MicroPython é uma implementação enxuta e eficiente da linguagem de programação Python 3, incluindo um subconjunto da biblioteca padrão do Python, e é otimizada para rodar em microcontroladores e em ambientes restritos.

O MicroPython foi originalmente criado pelo programador australiano e físico Damien George com objetivo de rodar na PyBoard que é uma placa compacta de circuito eletrônico (MICROPYTHON, 2018).

3.5.2 JavaScript

O JavaScript é uma linguagem de programação interpretada que foi inicialmente implementada como parte dos navegadores *Web* para que *scripts* pudessem ser executados do lado do cliente, por isso é chamada de *client-side*, ela é utilizada para controlar o HTML e o CSS para manipular uma página. A linguagem foi criada por Brendan Eich como uma forma de adicionar programas a páginas da *Web*. O JavaScript é usado em muitas plataformas como além de navegadores *Web*, como por exemplo bancos de dados e aplicativos para computadores e celulares (HAVERBEKE, 2014).

3.5.3 HTML

O HTML é uma linguagem de marcação que foi criada por Tim Berners-Lee em 1991. A sigla HTML significa *Hyper Text Markup Language*, que traduzida para o português significa Linguagem de Marcação de Hipertexto. Esta linguagem permite a utilização de um conjunto de comandos para exibição de conteúdo e formatação de layout de páginas para Internet (MAJER, 2009).

Hiper Text refere-se a atalhos conhecidos por links que fazem a conexão de uma página a outra dentro de um *Web* site ou entre *Web* sites. Os links são o aspecto fundamental da internet. Markup refere-se a marcação, que é usado para envolver texto, imagens, tabelas e outros conteúdos para renderizar no navegador (MOZILLA, 2018c).

HTML Canvas

O HTML é um elemento HTML que pode ser usado para desenhar gráficos via código. Além de desenhar gráficos, ele pode fazer composição de fotos e criar animações e até fazer processamento e renderização de vídeo em tempo real (MOZILLA, 2018a).

3.5.4 CSS

A sigla CSS significa *Cascade Style Sheet* que pode ser traduzido para o português como Folha de Estilo em Cascata. O CSS é uma tecnologia usada na criação de páginas de internet que foi criada em 1993 com objetivo de alterar as formas de apresentação de uma página fugindo das limitações que os navegadores traziam.

Com o surgimento do CSS, cores, tamanhos de fontes, e diversos atributos visuais puderam ser alteradas trazendo maior gama de possibilidades de apresentação de uma página na internet (MAJER, 2009).

3.6 FRAMEWORKS

Bulma

O Bulma é um *framework* CSS baseado na tecnologia *flexbox*, que facilita a organização dos elementos quando o layout precisa ser visualizado em diversos tamanho de tela em diferentes dispositivos, e possui uma alta compatibilidade entre os

navegadores. O pacote possui elementos comuns como formulários, menus, tabelas, títulos, barras de progresso e um sistema de *grid* (PRATES, 2016).

O Bulma é gratuito, disponibilizado sobre a licença de código aberto e é baseado em outro framework chamado Bootstrap (PRATES, 2016).

Express.js

O Express é um *framework* para Node.js minimalista e flexível que contém um conjunto de recursos para facilitar a criação de aplicações *Web*. O Express é a estrutura *Web* mais popular no Node.js e é esta implícito em outras bibliotecas populares do Node.js. O Express oferece métodos para especificação de funções para cada comando em HTTP e uma rota, também possui métodos para especificar o modelo da função e definir um modelo para renderizar uma resposta (MOZILLA, 2018b).

Vue.js

Vue.JS é uma biblioteca javascript de código aberto para o desenvolvimento de componentes reativos para interfaces *Web*, ou seja, são trechos de código que contém marcação, estilo e comportamento (html, css e javascript) que juntos podem compor interfaces reaproveitáveis. O Vue.JS permite que os dados e a camada de apresentação sejam vinculados facilmente (REIS, 2016).

O Vue.JS pode ser utilizados em qualquer projeto de tecnologia, pois se integra facilmente a outras linguagens de programação como o Ruby, PHP, Python, Node e Java (REIS, 2016).

3.7 PROGRESSIVE WEB APPS

Uma *Progressive Web App* (PWA) pode ser definida como uma evolução das páginas da *Web* regulares, e dos aplicativos móveis combinando as os recursos oferecidos pelos navegadores e as vantagens do uso de um celular. Uma PWA distingue-se de uma aplicação nativa pelas suas capacidades multiplataforma, podendo ser executada em vários sistemas operacionais e plataformas, pois é executada na *Web* através dos navegadores (MOTA, 2018).

Uma PWA é progressiva e responsiva, adaptando-se ao dispositivo em que está a ser executada, independente de conectividade, segura, atual e semelhante a

aplicações nativa, pois o processo de atualização e manutenção é semelhante ao de uma página Web instalável no dispositivo (MOTA, 2018).

Atualmente, mesmo com a grande variedade de aplicativos disponíveis para dispositivos móveis, grande parte dos utilizadores se restringem a utilização de algumas dezenas de aplicações referentes a produtos de grandes empresas como o Facebook e outras redes sociais, Google, Microsoft ou Amazon, ou seja, a maioria das aplicações disponíveis não são usadas pelos usuários, especialmente aplicações destinadas a fins muito específicos e desenvolvidas por empresas de menor porte. Essas aplicações podem ser definidas como “aplicações de ocasião” que são aplicações que existem, porém são pouco utilizadas até mesmo pelos usuários dos serviços e produtos de uma empresa (MOTA, 2018).

Como proposta de solução dessa realidade é o desenvolvimento de PWAs no lugar de aplicações nativas. Como vantagem as PWAs apresentam além das já citadas facilidades de desenvolvimento multiplataforma e manutenção, no lado do usuário, ocupam menos espaço de armazenamento no dispositivo, e são mais descartáveis para serviços de utilização espontânea ou ocasional, tendo a opção de fixa-la ou não no seu dispositivo através do endereço, contendo no final a mesma interface e funcionalidades próximas de uma aplicação nativa (MOTA, 2018).

3.8 SERVICE WORKERS

O service Worker é um script executado em segundo plano pelo navegador, que possibilitam recursos que não precisam de uma página ou uma interface com o usuário. Os Service Workers possuem recursos como notificações push e sincronização e permitem experiências off-line com uma aplicação (GAUNT, 2018).

O Service Worker é um "Javascript Worker" e por isso consegue acessar o DOM diretamente e se comunicar diretamente com as páginas que controla através de mensagens enviadas pela interface chamadas de DomMessage, se comportando como um proxy de rede programável, situado entre a aplicação Web, o navegador e a rede que permite controlar como as solicitações de rede da página são tratadas. Além disso, ele faz cache dos recursos trafegados de maneira granular, trazendo controle total sobre o comportamento da aplicação em determinadas situações, como por exemplo, quando não há conexão de rede (GAUNT, 2018).

Por questões de segurança o Service Worker funciona somente em sites servidos via HTTPS, para não haver possibilidade de um agente externo modificar as requisições de navegação (GAUNT, 2018).

3.9 NODE.JS

Node.js é uma plataforma que tem como base o JavaScript, que para construção de aplicações de rede com maior facilidade. O Node.js tem como objetivo usar um modelo de I/O (*Input/Output*), que é traduzido como entrada e saída de dados, assíncrona, o que o torna eficiente e ideal para aplicações em tempo real com troca intensa de dados através de dispositivos distribuídos.

O Node.js torna menor a complexidade de se modelar aplicações assíncronas devido a seus I/O serem Orientados a Evento trazendo grande facilidade para aplicações de tempo real (HAVERBEKE, 2014).

NPM

NPM é a abreviação de *Node Package Manager* que pode ser traduzido como Gerenciador de Pacotes Node. O NPM é um repositório online de código aberto para Node.js, que auxilia no controle de versão e no gerenciamento de dependências (NODEBR, 2016b).

O NPM conta com mais de 35000 pacotes, sendo eles bibliotecas a aplicações de código aberto que podem ser instalados usando somente uma linha de comando no terminal do computador bastando somente uma conexão com a internet para que a instalação seja concluída, sem a necessidade que os arquivos de dependências sejam salvos não ocupando espaço na aplicação e reduzindo o numero de arquivos do sistema (NODEBR, 2016b).

Vue-router

Um componente essencial numa *Single Page Application* é o router, que é responsável por exibir ou ocultar um ou mais elementos dependendo da URL que acessa a página (VEDOVELLI, 2016a).

A maior vantagem do Vue-router é que os elementos de interface já estão presentes no pré carregamento da página, assim, a velocidade de carregamento é

maior, trazendo ao usuário a sensação de uma aplicação instalada (VEDOVELLI, 2016a).

Vuex

O Vuex é um padrão de gerenciamento de estado e biblioteca para aplicações Vue.js. Quando uma aplicação com grande fluxo de dados é construída é comum que o controle de fluxo de informação seja perdido. Esses dados não são somente os que o usuário envia ao servidor mas também os que controlam a exibição de interfaces. Conforme o tamanho de uma aplicação aumenta é difícil gerenciar esses dados e todos os locais onde eles tem influência (VEDOVELLI, 2016b).

O Vuex utiliza a abordagem *Single Source of Truth* que pode ser traduzido como "Fonte única da Verdade", que estrutura os dados de maneira que eles não possam ser duplicados em nenhum lugar da aplicação (VEDOVELLI, 2016b).

3.10 APEXCHARTS

ApexCharts é uma biblioteca moderna JavaScript de código aberto para geração de gráficos modernos. Ela possui uma API simples, gráficos de visuais leves e detalhados com uma ampla gama de opções de personalização (TOWN, 2019).

O ApexCharts fornece integrações para o VueJs e React nativas facilitando o desenvolvimento (TOWN, 2019).

3.11 OBJECT RELATIONAL MAPPER - ORM

O object relational Mapper é uma técnica de mapeamento objeto relacional que permite fazer uma realização dos objetos com os dados que eles representam. Nos dados são usados princípios matemáticos com a linguagem SQL para orientar ao banco o que fazer, e na programação são usados classes, métodos e fundamentos da engenharia de software para orientar como serão feitas as funcionalidades de um programa. O ORM é o intercessor entre esses dois escopos, pois faz o mapeamento da classe para o banco de dados, e cada tipo de ORM tem sua maneira de gerar um SQL referente a inserção do objeto, que corresponde a uma tabela no banco de dados e realizar a operação (DEV MEDIA, 2011).

A vantagem do uso do ORM se dá pela produtividade, pois com ele não é necessário escrever códigos SQL, pois os mesmos são escritos pelo próprio ORM (DEVMEDIA, 2011).

3.12 WEBSOCKETS

WebSockets é uma tecnologia avançada que torna possível abrir uma sessão de comunicação interativa entre o navegador do usuário e um servidor. Com esta API, você pode enviar mensagens para um servidor e receber respostas orientadas a eventos sem ter que consultar o servidor para obter uma resposta (MOZILLA, 2018e). O objetivo desta tecnologia é fornecer um mecanismo para o navegador baseado em aplicativos que precisam de comunicação bidirecional com servidores que não dependem da abertura de várias conexões HTTP, como por exemplo o *XMLHttpRequest* (FETTE, 2018).

3.13 WEB COMPONENTS

Web Components são a combinação de 4 especificações que são os *Templates*, *Custom Elements*, *Shadow DOM*, e os *Imports*. Sendo cada uma delas individualmente úteis e quando combinadas são realmente importantes (EIS, 2014).

3.14 DESENVOLVIMENTO POR CAMADAS

No desenvolvimento *Web* existem três camadas de desenvolvimento, sendo elas informação, formatação, e comportamento. A informação é atrelada ao código HTML, formatação que é feita pela linguagem CSS, e o comportamento que é controlar elementos do HTML com JavaScript (HAVERBEKE, 2014).

O conceito de desenvolvimento em camadas organiza o código e divide cada parte do desenvolvimento facilitando o tratamento das informações de cada área (HAVERBEKE, 2014).

Custom Elements

Com o crescimento da *Web*, o HTML passou a não possuir todos os elementos necessários para cobrir todas as necessidades. O *custom elements* permite que os

desenvolvedores criem tags HTML, aprimorem as tags HTML existentes, ou ampliem os componentes criados por outros desenvolvedores (EIS, 2014).

Shadow DOM

O *Shadow DOM* é um recurso que permite encapsular os elementos de um componente, ou seja, torna a fonte de um componente inacessível, cria um escopo para o CSS que permite que os elementos sejam exibidos dentro do componente sem fazer parte do mesmo. O Shadow DOM foi projetado como uma ferramenta para a criação de aplicativos baseados em componentes. Assim, ele traz soluções para problemas comuns no desenvolvimento da *Web*, suas principais vantagens são a simplificação do CSS e aumento de produtividade. (EIS, 2014).

Templates

O HTML Templates é o recurso que permite a definição de modelo HTML. Os Templates são acessíveis para o navegador e seu conteúdo não é processado (EIS, 2014).

Imports

O HTML Imports é um recurso que importa uma página HTML completa, incluindo seu Javascript e o CSS. A página importada é processada paralelamente enquanto a página que o importou continuar normalmente seu processamento (EIS, 2014).

3.15 GIT

O Git é um sistema de controle de versão distribuído, ele foi desenvolvido inicialmente por Linus Torvalds e usado no desenvolvimento do kernel do Linux. Através do Git é possível trabalhar sempre em um mesmo diretório podendo fazer alterações e gravando a documentação e comentários. Com o registro do que é feito é possível que o desenvolvedor possa voltar a fase anterior quando houverem problemas (POZZEBOM, 2015).

Além disso, é possível criar áreas separadas para realizar testes ou projetos diferentes, podendo desfazer alterações que causaram problemas, voltando a versão

anterior. Com o Git também é possível trabalhar em arquivos de forma independente, sem alterar o arquivo principal antes de uma publicação oficial (POZZEBOM, 2015).

3.16 BANCO DE DADOS MONGODB

O MongoDB é uma ideia de banco de dados orientado à documentos, que pode ser definido com o conceito de dados autocontidos e auto descritivos, e isso implica que o documento em si já define como ele deve ser apresentado e qual o significado dos dados apresentados na sua estrutura (DEVMEDIA, 2017).

Os bancos de dados Orientados a Documentos têm como característica conter todas as informações em um único documento, sem esquemáticos de dados e possuir identificadores únicos universais (UUID), possibilitando a consulta de dados através de métodos avançados de agrupamento e filtragem e permitir redundância e inconsistência (DEVMEDIA, 2017).

Esses bancos de dados são chamados de Bancos NoSQL, (Not Only SQL). Esse termo é devido à ausência do Structured Query Language (SQL), pois de forma geral, esse banco de dados não tem características do modelo relacional de banco de dados e nem a linguagem SQL. O MongoDB tem como característica ser código-fonte aberto, licenciado pela a Affero General Public License (GNU AGPL), possuir alto desempenho, não possuir esquemas, ser escrito em C++, multiplataforma e ser formado por um conjunto de aplicativos JSON (DEVMEDIA, 2017).

BIBLIOTECA MONGOOSE

O Mongoose é uma biblioteca do Nodejs que proporciona uma plataforma baseada em esquemas para modelagem de dados de uma aplicação. Essa biblioteca possui sistema de conversão de tipos, validação, criação de consultas e funções para lógicas de negócios (NODEBR, 2016a).

O Mongoose possui um mapeamento de objetos semelhante ao ORM (Object Relational Mapping), ou Object Data Mapping (ODM), no caso do Mongoose, traduzindo os dados do banco de dados para objetos JavaScript para que possam ser utilizados pela aplicação (NODEBR, 2016a).

3.17 COMPUTAÇÃO EM NUVEM

A computação em nuvem é a entrega de serviços de computação, tais como servidores, armazenamento, banco de dados, rede, software, análise e inteligência pela internet que é a nuvem, oferecendo inovação com mais rapidez, flexibilidade de recursos e economia em escala (AZURE, 2017).

Entre as principais vantagens da computação em nuvem estão o custo, pois elimina a necessidade de compra de hardware e instalação e execução de datacenters locais, como racks de servidores e eletricidade com disponibilidade permanente para energia e todo suporte envolvido, a velocidade, que pode ser dimensionada flexibilizando os recursos computacionais usados, e a segurança que os provedores em nuvem oferecem com a tecnologia e controle que fortalece a proteção de dados, aplicações e infraestrutura contra possíveis ameaças (AZURE, 2017).

3.18 ANÁLISE DE REGRESSÃO

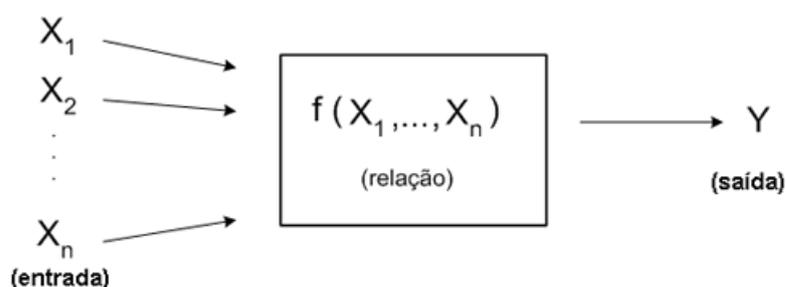
Em diversos problemas das áreas médica, biológica, industrial, química entre outras, é de grande interesse verificar se duas ou mais variáveis estão relacionadas de alguma forma. Para expressar esta relação é muito importante estabelecer um modelo matemático. Este tipo de modelagem é chamado de regressão, e ajuda a entender como determinadas variáveis influenciam outra variável, ou seja, verifica como o comportamento de uma(s) variável(is) pode mudar o comportamento de outra (ACTION, 2018).

A Análise de Regressão possibilita encontrar uma relação razoável entre as variáveis de entrada e saída, por meio de relações empíricas. A utilização desta abordagem necessita de coleta de dados e do uso de métodos estatísticos de Análise de Regressão Linear. A coleta de dados permite conhecer a natureza da relação entre as variáveis e realizar estudos capazes de acomodar situações inesperadas, como por exemplo, variabilidade na matéria prima, temperatura ambiente, máquina e operadores (ACTION, 2018).

Para encontrar a relação de apenas uma variável de entrada com a variável resposta é usada a Regressão Linear Simples. E para relacionar a variável resposta com mais de uma variável regressora, a Regressão Linear Múltipla é utilizada (ACTION, 2018).

Caso a variável resposta seja uma variável categórica, ou seja, a variável apresenta como possíveis realizações uma qualidade (ou atributo) e não mais uma mensuração, utilizamos o Modelo de Regressão Logística. A figura 3.4 representa o processo para obtenção dessas relações de regressão (ACTION, 2018).

FIGURA 3.4 – REPRESENTAÇÃO DE UM PROCESSO DE REGRESSÃO.



Fonte: (ACTION, 2018)

Modelos de Regressão são construídos com os objetivos:

- (i) Predição - Uma vez que esperamos que grande parte da variação da variável de saída seja explicada pelas variáveis de entrada, podemos utilizar o modelo para obter valores de Y correspondentes a valores de X que não estavam entre os dados. Esse procedimento é chamado de predição e, em geral, usamos valores de X que estão dentro do intervalo de variação estudado. A utilização de valores fora desse intervalo recebe o nome de extrapolação e deve ser usada com muito cuidado, pois, o modelo adotado pode não ser correto fora do intervalo estudado. Acredita-se que a predição seja a aplicação comum dos modelos de regressão;
- (ii) Estimação de parâmetros - Dado um modelo e um conjunto de dados referente às variáveis resposta e preditoras, estimar parâmetros ou ajustar um modelo aos dados significa obter valores ou estimativas para os parâmetros, por algum processo, tendo por base o modelo e os dados observados;
- (iii) Inferência - O ajuste de um modelo de regressão em geral tem por objetivos básicos, além de estimar os parâmetros, realizar inferências sobre eles, tais como, testes de hipóteses e intervalos de confiança.
- (iv) Inferência - O ajuste de um modelo de regressão em geral tem por objetivos básicos, além de estimar os parâmetros, realizar inferências sobre eles, tais como, testes de hipóteses e intervalos de confiança.

4 PROJETO

Nesse capítulo são apresentadas todas as características de projetos desse trabalho, como arquitetura, escolha dos dispositivos e uso das tecnologias dentro do projeto:

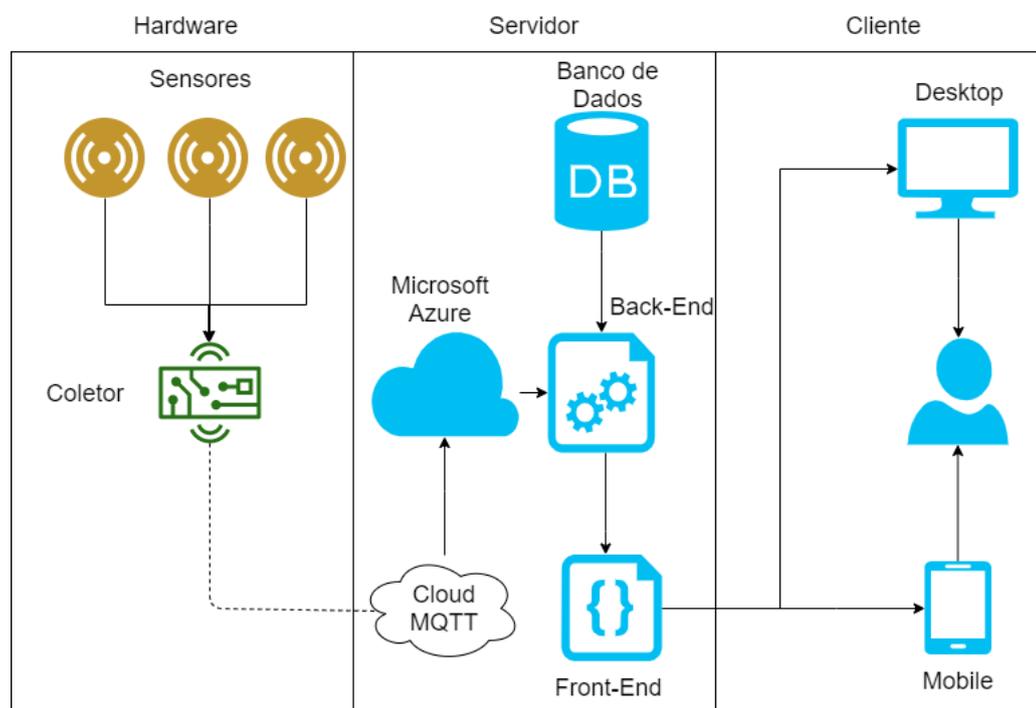
4.1 ARQUITETURA DO PROJETO

O fluxo do projeto segue um fluxo que se inicia na captação dos dados a partir dos sensores, que são enviados para o microcontrolador (Coletor) que envia os dados obtidos das leituras a Cloud MQTT, que por sua vez envia os dados para o Azure (Nuvem), onde estão implementados os servidores e o banco. o servidor busca os dados das recebidos e envia para a página *Web* para que seja feita a exposição dos dados, e envia para o banco de dados.

Quando o usuário da aplicação solicitar a visualização dos dados de leituras feitas em momentos anteriores, a página *Web* irá mandar uma solicitação ao servidor que irá solicitar armazenamento esses dados, o banco de dados irá devolver esses dados ao servidor que entregará ao usuário os dados solicitados.

A Arquitetura Geral do projeto é subdividida em 3 partes, sendo elas o hardware, servidor e o cliente. Na parte de hardware existem os sensores e o coletor, enquanto na de servidor existe a Cloud MQTT, nuvem publica com o back-end, front-end e o banco de dados, e por fim, o cliente que consiste nas plataformas de visualização dos dados que são desktop e mobile. A figura 4.1 representa o fluxo e arquitetura geral presente no projeto do sistema de monitoramento remoto de solos.

FIGURA 4.1 – DIAGRAMA REPRESENTANDO A ARQUITETURA GERAL DO PROJETO



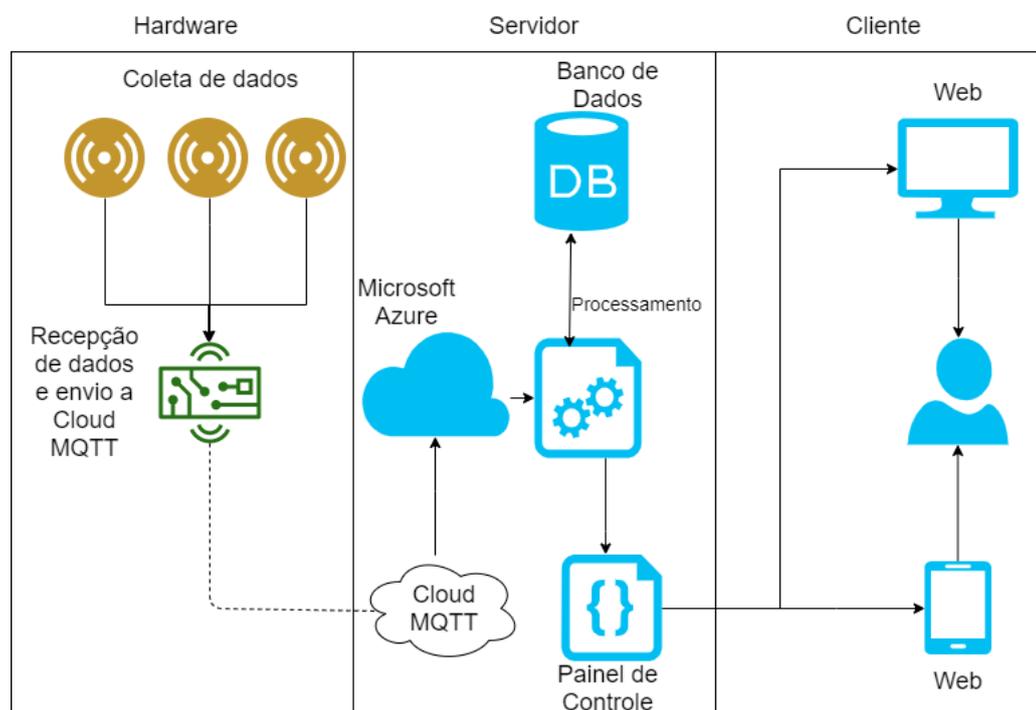
Fonte: (AUTORIA PRÓPRIA)

Plataforma do Projeto

A primeira decisão no desenvolvimento da aplicação foi a escolha da plataforma aonde iriam ser expostos os dados obtidos. Durante muito tempo as aplicações desktop, supriram as necessidades dos usuários de *software* nas mais diversas áreas, porém, nos últimos anos, com o avanço da tecnologia móvel, foi optado a plataforma *Web*. A plataforma *Web* possibilita a versatilidade de dispositivos e facilidade no acesso e atualização, sem que seja necessária a instalação de software nos dispositivos de acesso.

Do ponto de vista funcional, a arquitetura continua sendo dividida em 3 áreas assim como na arquitetura geral referenciada pela Figura 4.1. Na arquitetura funcional a parte de hardware fica responsável pela coleta de dados do solo, recepção dos dados e envio a Cloud MQTT. A parte do servidor é responsável pela parte visual do painel de controle, processamento e armazenamento em nuvem, e o cliente é o responsável por exibir os dados via interfaces *Web* em dispositivos desktop e mobile. A figura 4.2

FIGURA 4.2 – DIAGRAMA REPRESENTANDO A ARQUITETURA FUNCIONAL DO PROJETO



Fonte: (AUTORIA PRÓPRIA)

4.1.1 Escolha dos Sensores

Para captar informações pertinentes ao escopo do projeto, foi feito o mapeamento dos dados básicos que são necessários para o sistema de monitoramento de encostas. Os dados que foram pontuados são a umidade do solo, ângulo do solo ou encosta, e variação de aceleração nos eixos. Para captar esses dados foram escolhidos inicialmente dois módulos sensores que são o higrômetro, que irá captar a grandeza de umidade do solo, e o giroscópio que irá captar a posição do sensor nos 3 eixos (X, Y e Z) e a aceleração. Além disso o giroscópio fornece informações como a taxa de giro, inclinação, aceleração em módulo e nos 3 eixos e temperatura do ambiente em unidades Celsius, Fahrenheit e Kelvin.

Como as chuvas atuam como principal agente físico desencadeante para deslizamentos no Brasil, para fazer a quantificação e análise foi escolhido um sensor de chuva, o qual irá captar e avaliar a intensidade da chuva como moderada, média ou torrencial. O nível de chuva por determinado tempo será usado como forma de predição e geração de alertas.

Dentre os parâmetros que o sensor consegue captar, foram escolhidos para expor na página Web a umidade do solo, temperatura ambiente, a posição nos 3 eixos

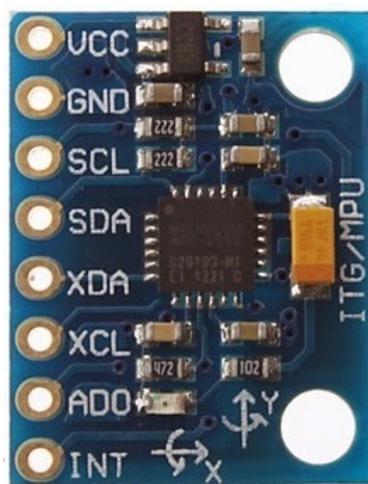
e inclinação e a aceleração nos 3 eixos. A ideia inicial é que o acompanhamento do parâmetro de posição dará ao usuário informações sobre o deslocamento da encosta em grandes períodos, e as medidas de aceleração sobre informar em momento real e gerar alertas sobre eventos de deslizamento, pois eles gerarão picos de aceleração que indicam a ocorrência de eventos de deslizamentos.

Giroscópio MPU6050

O módulo MPU6050 é um giroscópio de alta precisão, acelerômetro, e sensor de temperatura de 6 eixos ou 6 graus de liberdade fornecendo 6 saídas sendo delas 3 do giroscópio e 3 do acelerômetro, além da saída de temperatura que fornece valores com precisão de -40 a 85 °C.

O módulo MPU conta com um recurso chamado *Digital Motion Processor* (DMP), que é um acelerador de *hardware* e cuida dos cálculos do sensor. As principais vantagens desse sensor, é o baixo custo e consumo de energia, tornando o ideal para projetos de monitoramento (ALEXANDRE, 2015). A figura 4.3 representa o sensor MPU 6050.

FIGURA 4.3 – MÓDULO MPU6050.



Fonte: (ALEXANDRE, 2015)

Higrômetro

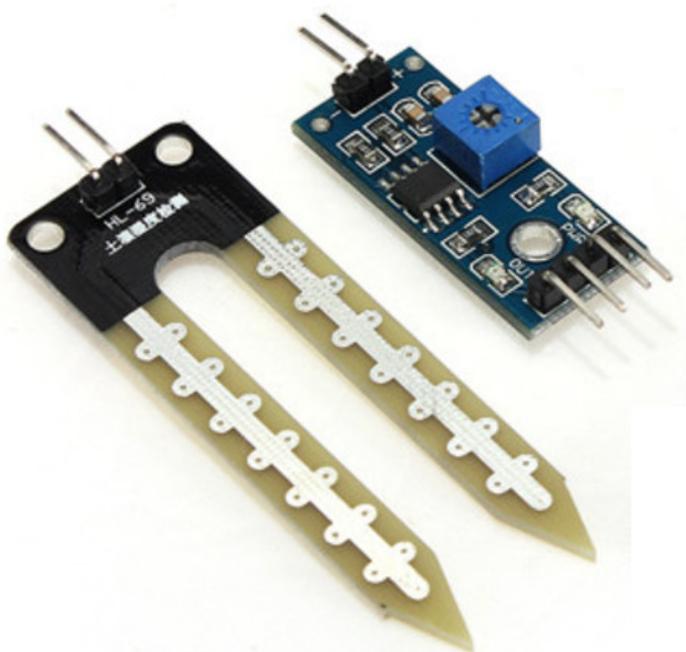
O sensor de umidade do solo, ou higrômetro, consiste em 2 partes, sendo uma delas a sonda que fica em contato direto do solo, e um módulo comparador LM393 que

lê os dados do sensor e envia para o microcontrolador. O módulo comparador possui duas saídas, sendo uma delas a D0, que varia entre 1 e 0 dependendo da umidade, e uma saída analógica A0, que possibilita maior precisão nas medições e possui um potenciômetro, no qual é possível fazer a calibração do sensor (THOMSEN, 2016).

O sensor tem uma tensão de operação de 3,3 a 5 V com corrente de saída de até 100mA, possui LEDs para indicação da tensão e da saída digital com dimensão da sonda de 6x2 centímetros quadrados (THOMSEN, 2016).

Na figura (THOMSEN, 2016) é possível observar o módulo comparador e as sondas do higrômetro (THOMSEN, 2016).

FIGURA 4.4 – HIGRÔMETRO.



Fonte: (THOMSEN, 2016)

Sensor de Chuva

O sensor de Chuva consiste em 2 partes, sendo uma delas uma placa de circuito impresso com as trilhas expostas para detecção da água, e a outra um módulo comparador LM393 que lê os dados do sensor e envia para o microcontrolador (THOMSEN, 2016).

O módulo comparador possui duas saídas, sendo uma delas a D0, que varia entre 1 e 0 dependendo da umidade, e uma saída analógica A0, que possibilita maior

precisão nas medições e possui um potenciômetro, no qual é possível fazer a calibração do sensor (THOMSEN, 2016).

O sensor tem uma tensão de operação de 3,3 a 5 V com corrente de saída de até 100mA, possui LEDs para indicação da tensão e da saída digital com dimensão da placa de 5x4 centímetros quadrados. Na figura ?? é possível observar o sensor de chuva e o módulo comparador (THOMSEN, 2016).

FIGURA 4.5 – SENSOR DE CHUVA



Fonte: (THOMSEN, 2016)

4.2 ESCOLHA DO COLETOR

Para atender a arquitetura geral proposta, é necessário que o coletor de dados seja capaz de interagir com a rede de internet sem fio para que seja possível fazer o envio dos dados provenientes das leituras para a Cloud MQTT. Além disso, é necessário que o dispositivo seja de baixo custo e boa confiabilidade visando o melhor custo-benefício.

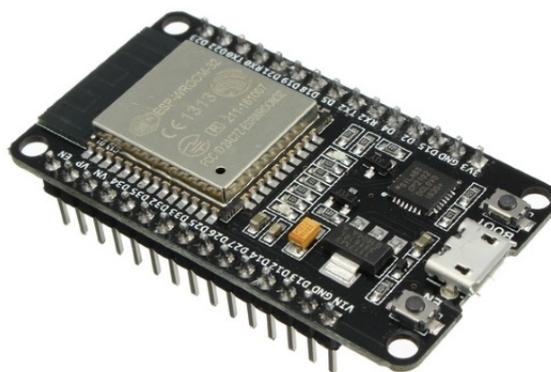
ESP32

Desenvolvido pela empresa Espressif, o microcontrolador ESP32 apresenta-se como um meio inovador no desenvolvimento de projetos automatizados. Além do módulo de comunicação Wi-Fi, apresenta um sistema com processador Dual Core, Bluetooth híbrido e múltiplos sensores embutidos, tornando a construção de sistema como internet das coisas (IoT) muito mais simples e compacto.

O ESP32 dispõe de memória SRAM de 520 KB, memória flash de 16MB, dois núcleos de processamento em arquitetura Tensilica LX6 capaz de operar em 240 MHz, Bluetooth híbrido (clássico e BLE), tensão operacional de 2,3V á 3,6V, conector externo antena IPEX/PCB, redução de problemas com instabilidade, quantidade maior de portas programáveis, múltiplos sensores, cristal oscilador 32KHZ e criptografia com chave RSA de 4096 bits, ou seja, flexibilidade operacional e maior segurança na construção de sistemas automatizados.

Na pinagem de um ESP32, podemos encontrar 34 portas GPIO, 3 SPI, 2 I2S, 18 canais ADC, 3 UART, 10 pinos de leitura capacitiva e PWM. Na figura 4.6 podemos observar a placa do ESP32.

FIGURA 4.6 – PLACA ESP32.



Fonte: (CURTOCIRCUITO, 2018)

4.3 PROGRESSIVE WEB APPS

Devido a escolha do interpretador ter como um dos motivos a portabilidade de exposição de dados em diferentes plataformas como navegadores de computadores e de *smartphones*, na produção do leiaute da página foi levado em conta parâmetros de responsividade da página seguir padrões do conceito de *Progressive Web App* (PWA), que torna a interface *Web* mais agradável visualmente em outras resoluções de telas, que no caso seriam de *smartphones*. Com o PWA foi possível implementar aprimorar a interação do aplicativo em dispositivos móveis por meio da responsividade e o menu de navegação *SideNav*, que indica rotas opcionais para parâmetros da página que serão implementados futuramente.

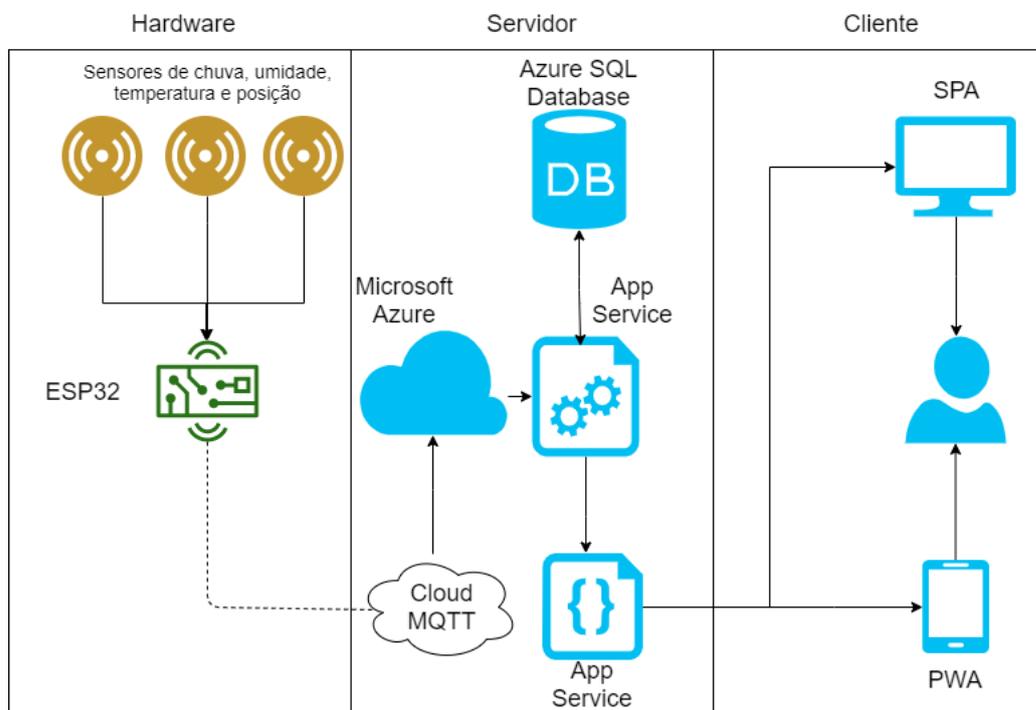
Além das vantagens já citadas, a PWA possui uma grande vantagens quando se trata de dispositivos móveis como o smartphone, que é a possibilidade de acessar a aplicação mesmo sem internet, e salva-la na tela inicial como atalho de modo semelhante a um aplicativo nativo de celular. Com isso o usuário tem a interação igual à de um aplicativo comum, com a vantagem de não usar a memória de seu dispositivo.

Para implementação da PWA e facilitar a produção da layout, para diferentes dimensões de telas, foi usado o framework Bulma, que é uma biblioteca CSS de código aberto que possui modelos prontos de menus, botões e barras.

4.4 ARQUITETURA DE COMPONENTES

Os componentes que fazem parte do projeto são na parte do hardware são os sensores que são os de intensidade da chuva, umidade, temperatura, aceleração nos 3 eixos e posição nos 3 eixos, todos ligados a um ESP32 que envia sem fio os dados adquiridos a Cloud MQTT, Após isso esses dados são enviados a nuvem Azure, onde haverá um App Service para o front-end e back end da aplicação salvando os dados no Azure SQL Database. Todos esses dados serão consumidos pelo cliente por meio de uma Single Page Application no desktop ou uma *Progressive Web App* em dispositivos mobile. A figura 4.7 representa os componentes de cada item presente na arquitetura do projeto.

FIGURA 4.7 – DIAGRAMA REPRESENTANDO A ARQUITETURA DOS COMPONENTES DO PROJETO



Text

Fonte: (AUTORIA PRÓPRIA)

4.5 ESCOHA DAS TECNOLOGIAS

4.5.1 Azure

Apresentado em 2008, o Windows Azure foi lançado em 2010 e renomeado para Microsoft Azure em 2014. Hoje, o serviço concentra toda a plataforma de nuvem da Microsoft — desde a infraestrutura de cloud computing para hospedar sistemas empresariais até ferramentas e recursos que expandem a capacidade produtiva de negócios em todos os setores (AZURE, 2017).

Em termos gerais, o Microsoft Azure funciona como um apoio tecnológico ao negócio. Ele oferece, ao mesmo tempo, a infraestrutura necessária para rodar seu sistema empresarial (computação, armazenamento, lógica, gestão financeira etc) e as ferramentas necessárias para transformar esse poder em produtividade (AZURE, 2017).

4.5.2 Azure App Service

O Azure App Service é uma opção de PaaS (Plataforma como Serviço) que oferece mecanismos para hospedar aplicações web escaláveis, em diferentes lingua-

gens, capazes de serem integradas a outros serviços da plataforma (RODRIGUES, 2017).

Uma das principais características do App Service é a facilidade que ele oferece para os desenvolvedores realizarem o deploy (publicação) de suas aplicações. Esse processo pode ser feito a partir do Visual Studio, para quem usa esse IDE, mas também pode ser realizado de outras maneiras e de forma alinhada a práticas de DevOps. Por exemplo, é possível publicar sua aplicação a partir de um repositório no GitHub, Bitbucket ou Visual Studio Team Services quando um commit for feito em determinada branch (RODRIGUES, 2017).

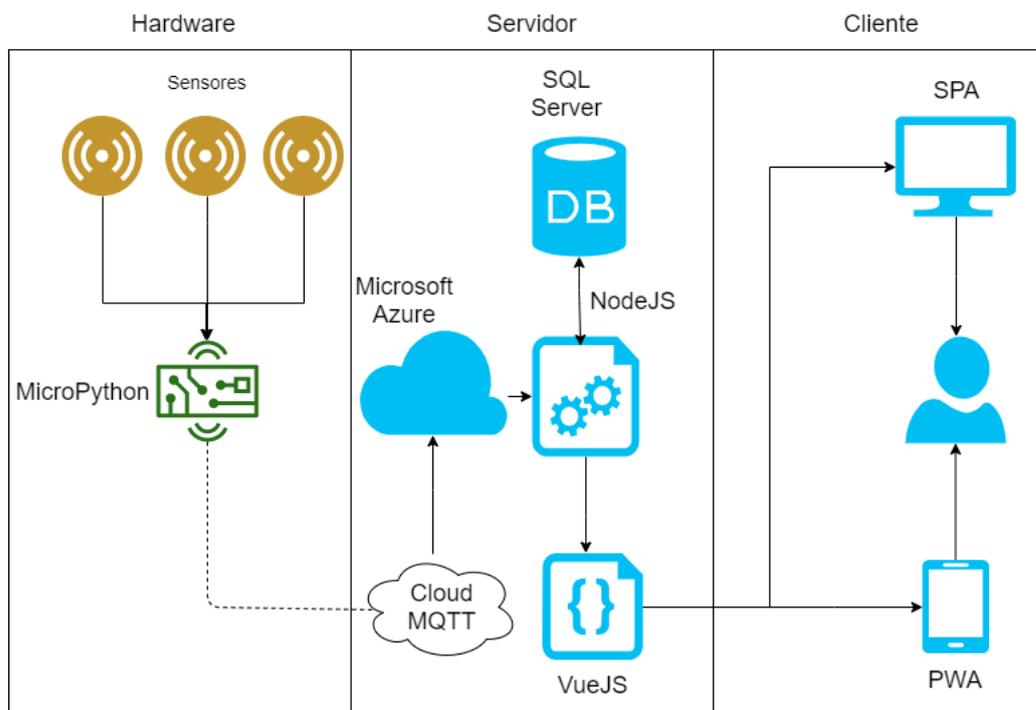
4.5.3 Armazenamento de Dados

Para realizar o armazenamento dos dados coletados foi escolhido o banco de dados Azure SQL Server. O SQL Server é um banco de dados de uso relacional om ele, você pode criar uma camada de armazenamento de dados altamente disponível e de alto desempenho para os aplicativos e soluções no Azure

4.6 ARQUITETURA DAS TECNOLOGIAS

As tecnologias que serão empregadas na parte do hardware sera a aquisição dos dados no coletor com a linguagem de programação MicroPython, que enviara os dados sem fio para a Cloud MQTT que ira fazer a recepção e envio das informações. Na parte de servidor, existe a Cloud MQTT e a nuvem pública, a nuvem usada sera a Microsoft Azure, que terá um back-end em NodeJS, e um front com o VueJS, e o armazenamento com o SQL Server. A figura 4.8 representa qual é a tecnologia presente em cada um dos componentes da arquitetura do projeto do Sistema de Monitoramento Remeto de Solos.

FIGURA 4.8 – DIAGRAMA REPRESENTANDO AS TECNOLOGIAS PRESENTES EM CADA UM DOS COMPONENTES DO PROJETO



Text

Fonte: (AUTORIA PRÓPRIA)

4.7 REQUISITOS FUNCIONAIS

O projeto do Sistema de Monitoramento Remoto de Solos visa atender os seguintes requisitos funcionais:

- Realizar leituras dos dados de umidade do solo, intensidade da chuva, posição e aceleração dos sensores;
- Integrar os elementos adequadamente garantindo o envio dos dados;
- Empregar comunicação sem fio entre elementos;
- Persistir os dados em nuvem;
- Apresentar graficamente os dados de modo útil no painel de controle.

4.8 COMUNICAÇÃO ENTRE SENSORES E COLETOR

Para realizar a conexão entre o ESP32 e os sensores, foi desenvolvido um código usando a linguagem MicroPython que roda no microcontrolador. Para os senso-

res de umidade do solo e de intensidade de chuva, foram usados a entrada analógica sendo, usando o conversor analógico digital presentes no microcontrolador.

Já para o MPU6050, foi feita a aquisição das leituras com auxílio do protocolo I2C com auxílio de um arquivo de código que consiste em uma classe com as configurações específicas do hardware do sensor. O código chama as funções da classe com as especificações e a classe fica responsável de criar um cabeçalho de todos os dados obtidos e devolver ao código do ESP32.

4.9 COMUNICAÇÃO COLETOR E CLOUD MQTT

No código do microcontrolador, foi importada um módulo MQTT que faz o auxílio da conexão com a Cloud MQTT. A comunicação entre o microcontrolador é feita por meio de 3 passos, sendo o primeiro, a conexão com a internet, onde é feita uma autenticação básica com nome da rede Wi-Fi e senha. O segundo passo consiste em fazer as configurações do parâmetros do MQTT onde são definidos a autenticação, porta, nome do dispositivo visto pela Cloud MQTT e host. O terceiro passo é uma função que envia os dados a cloud por meio de um método POST do protocolo HTTP que publica as leituras feitas pelo microcontrolador.

Para o desenvolvimento do painel de controle foi desenvolvido

4.10 PROGRESSIVE WEB APP COM VUE.JS

Para facilitar a criação da PWA, foi usado um modelo existente na aplicação Vue.js. A utilização do modelo fez com quem a PWA fosse desenvolvida paralelamente ao desenvolvimento do resto da aplicação, trabalhando automaticamente na responsividade e aparência do aplicativo hospedado no navegador.

4.11 GERAÇÃO DE GRÁFICOS

A geração de gráficos é feita diretamente pela biblioteca ApexCharts, que recebe do servidor os valores obtidos dos sensores, e exibe nos gráficos sendo o eixo vertical a magnitude da leitura obtida, e o eixo horizontal uma linha temporal que registra valores de horas, minutos, segundos e a data que foi efetuada a medição. O parâmetro temporal serve como base para organização do armazenamento do banco de dados por data.

4.12 ALGORITMOS DE PREDIÇÃO

Para avaliar a viabilidade da implementação de um algoritmo de predição de eventos de deslizamentos de solos, foi feita uma pesquisa sobre quais interfaces e abstrações o mercado de *Machine Learning* dispoe. Com a abstração a complexidade da produção de código diminui, porem para que haja uma alta eficacia do algoritmo é necessário um trabalho paralelo de filtragem e acerto de um modelo eficiente, que gerará resultados mais acertivos.

4.12.1 TensorFlow

O TensorFlow é uma biblioteca de software de código aberto que foi desenvolvida originalmente pela Google, com objetivo de auxiliar em estudos de *Machine Learning* e *Deep Learning*. Foi disponibilizado em 2011 e teve grande aceitação do mercado se tornando a biblioteca padrão para desenvolvimento de aplicações com *Machine Learning* (ACADEMY, 2018).

5 DESENVOLVIMENTO

Esse capítulo apresenta como foi feito o desenvolvimento do sistema de monitoramento remoto de solos, a construção de sua plataforma, a montagem e ligação entre os dispositivos, produção do software e do painel de visualização. O capítulo está dividido conforme:

- Montagem dos dispositivos
- Plataforma Física do Projeto;
- Sensorização analógica;
- Produção da página Web para exposição de dados;
- Armazenamento de dados

5.1 MONTAGEM DOS DISPOSITIVOS

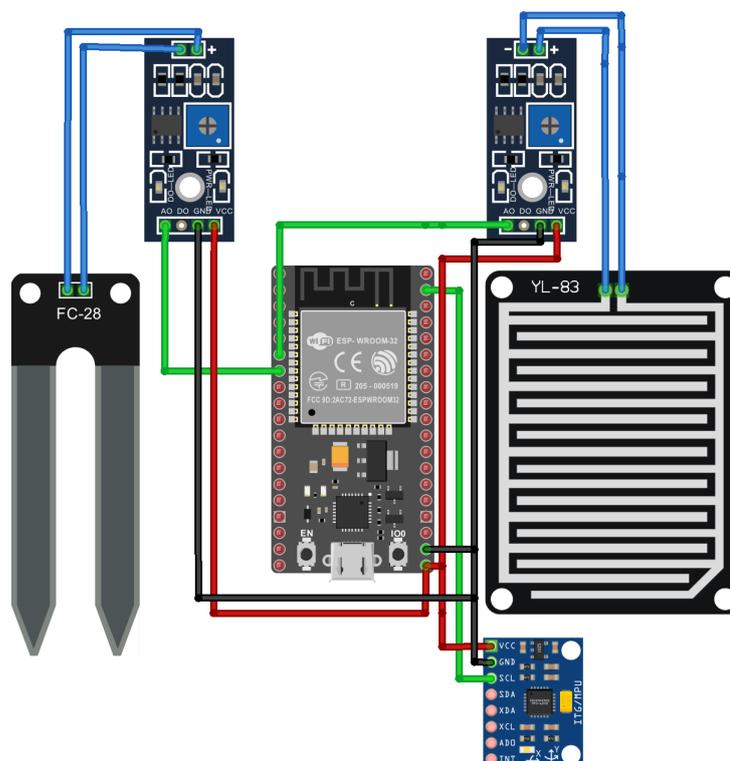
A ligação entre o dispositivo coletor de dados e os sensores foi feita com auxílio da alimentação presente no microcontrolador, com objetivo de reduzir os custos e o peso de uma alimentação externa. Somente o dispositivo coletor terá uma alimentação externa que pode ser via cabo USB ou até mesmo uma bateria de acordo com a necessidade.

Os sensores de umidade do solo e de intensidade da chuva, usam a interface analógica e o módulo MPU6050 usa a interface I2C para se conectar ao microcontrolador. A figura 5.1 representa o esquemático de ligação entre os sensores e o coletor.

5.2 COMUNICAÇÃO ENTRE COLETOR E CLOUD MQTT

Para realizar a comunicação entre o coletor e a Cloud MQTT, foi feita com o auxílio do *broker* CloudMQTT, as leituras efetuadas pelo coletor são colocadas em um cabeçalho padrão contendo o nome do dispositivo, hora e os dados obtidos pelos sensores, e enviados a CloudMQTT. A CloudMQTT recebe esses dados e envia para

FIGURA 5.1 – MONTAGEM DOS DISPOSITIVOS DO PROJETO.



Fonte: (AUTORIA PRÓPRIA)

o Azure. A instancia configurada do *broker* pode ser vista na figura 5.2 e o formato mensagem enviada pelo coletor chegando ao *broker* no formato representado pela figura 5.3.

FIGURA 5.2 – INSTANCIA DA CLOUDMQTT.

Instance info

Server	tailor.cloudmqtt.com	
User	enlwkcio	Restart
Password	👁️	🔄
Port	17302	
SSL Port	27302	
Websockets Port (TLS only)	37302	
Connection limit	5	

Active Plan



Cute Cat

Upgrade Instance

Fonte: (AUTORIA PRÓPRIA)

FIGURA 5.3 – CLOUD MQTT RECEBENDO MENSAGEM DO COLETOR COM AS LEITURAS DOS SENSORES.



Topic	Message
ESP1	{"time": [2019, 11, 23, 0, 39, 10, 5, 327], "soil": 4095, "sensor": "ESP1", "accel": {"GyZ": 143, "GyY": 35, "GyX": -212, "Tmp": 26.74176, "AcZ": 16524, "AcY": 188, "AcX": -376}, "rain": 4069}
ESP1	{"time": [2019, 11, 23, 0, 40, 10, 5, 327], "soil": 4095, "sensor": "ESP1", "accel": {"GyZ": 127, "GyY": 52, "GyX": -232, "Tmp": 26.64765, "AcZ": 16520, "AcY": 180, "AcX": -272}, "rain": 4072}
ESP1	{"time": [2019, 11, 23, 0, 41, 10, 5, 327], "soil": 4095, "sensor": "ESP1", "accel": {"GyZ": 151, "GyY": 56, "GyX": -215, "Tmp": 26.78882, "AcZ": 16508, "AcY": 156, "AcX": -440}, "rain": 4077}
ESP1	{"time": [2019, 11, 23, 0, 42, 10, 5, 327], "soil": 4095, "sensor": "ESP1", "accel": {"GyZ": 164, "GyY": 51, "GyX": -209, "Tmp": 26.74176, "AcZ": 16552, "AcY": 136, "AcX": -436}, "rain": 4074}

Fonte: (AUTORIA PRÓPRIA)

5.3 PLATAFORMA FÍSICA DO PROJETO

Foi desenvolvido o dispositivo que se assemelha a uma estaca, onde vão ser fixados os sensores de umidade, temperatura e o giroscópio. Como se trata de um protótipo e foi priorizado parâmetros como viabilidade e baixo custo, por isso, inicialmente os sensores foram montados em um cano de *Polyvinyl chloride* PVC de 25 milímetros de 30 centímetros de comprimento que foi pintado na cor preta. Em sua extremidade superior foi fixado o giroscópio e sensor de temperatura, e na sua extremidade inferior foi fixado o higrômetro que irá captar a umidade do solo. e a figura 5.4 o protótipo do sensor já contendo fixado o sensor de umidade higrômetro.

Com objetivo de criar um ambiente para simulação de tamanho reduzido, aumentando assim a mobilidade e facilitando a apresentação do projeto, foi montado em uma placa de *Medium-Density Fiberboard* (MDF) de com medidas de 60 centímetros de largura por 60 centímetros de comprimento. Nessa placa foi fixada uma caixa que de aproximadamente 20 centímetros de largura por 30 de comprimento que irá ser ocupada com terra para que sejam feitas as devidas medidas, e o resto da placa será

FIGURA 5.4 – ESTACA ONDE SERÃO FIXADOS OS SENSORES.



Fonte: (AUTORIA PRÓPRIA)

ocupada pela placa microcontroladora, sensor de chuva e o computador que irá captar os dados. As figuras 5.5 e 5.6 representam a plataforma do projeto.

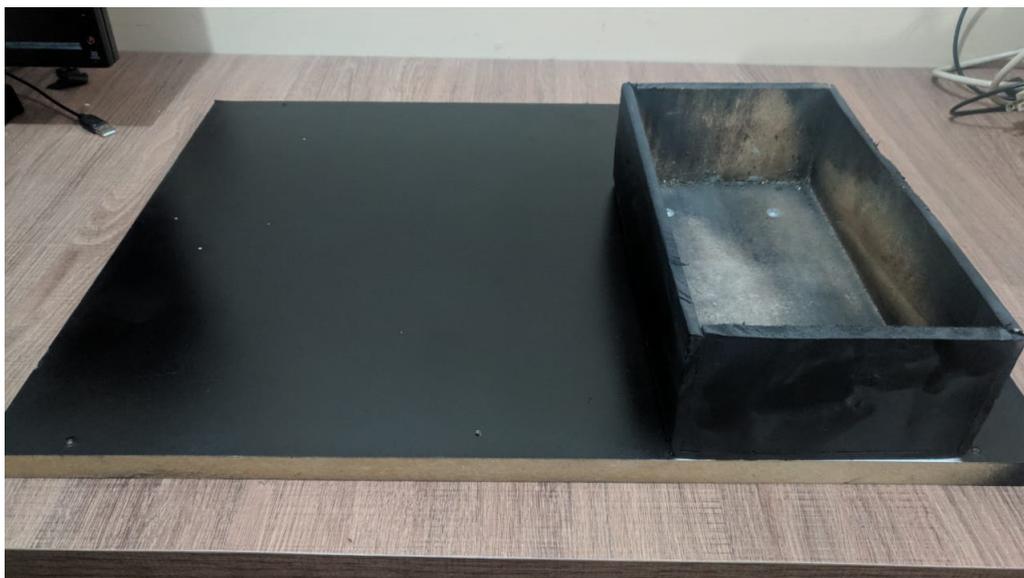
FIGURA 5.5 – PLATAFORMA DE SIMULAÇÃO DO PROJETO.



Fonte: (AUTORIA PRÓPRIA)

Após a produção da plataforma, foi realizada a fixação dos sensores, microcontrolador e placa perfurada na superfície por meio de parafusos. A conexão entre os componentes do projeto foi feita por meio de fios do tipo *jumpers* e sua alimentação através de uma fonte de 5 Volts alimentada via entrada USB do computador.

FIGURA 5.6 – PLATAFORMA DE SIMULAÇÃO DO PROJETO.



Fonte: (AUTORIA PRÓPRIA)

5.4 PRODUÇÃO DA PÁGINA WEB PARA EXPOSIÇÃO DE DADOS

Foi produzida uma página *Web* a qual foi nomeada como *Geo Control*, que recebe os dados coletados pelos sensores analógicos e expõe de forma clara ao usuário. A página possui título, subtítulo, menu de navegação, espaço para textos explicativos sobre os dados coletados, um rodapé o qual informa informações da instituição de ensino que está produzindo esse trabalho no qual podem ser colocadas informações adicionais ou divulgação de páginas úteis relacionados. Para a exposição dos gráficos dos dados coletados pelos sensores, foram criadas 5 rotas além da *home* que exibe os dados coletados de maneira geral sem gerar gráficos.

A página seguiu conceitos de responsividade e PWA, os quais geraram uma melhor adaptação visual para outras resoluções de monitores e telas. Em resoluções menores com por exemplo a de *smartphones*, a disposição das informações se altera, fazendo que a barra de menu se torne oculta lateralmente, e os textos e informações disponíveis na página sejam expostas de forma diferente em relação a resoluções de navegadores de computador. A responsividade tornou a visualização da página em *smartphones* similar a de aplicativos de celular em um navegador *Web*. Com isso, também é possível salvar um atalho do aplicativo na tela inicial do celular, a abri-lo com visualização semelhante a de um aplicativo, sem a presença da caixa de endereço que está presente no navegador. Além disso, na página foi redigido um texto que resume

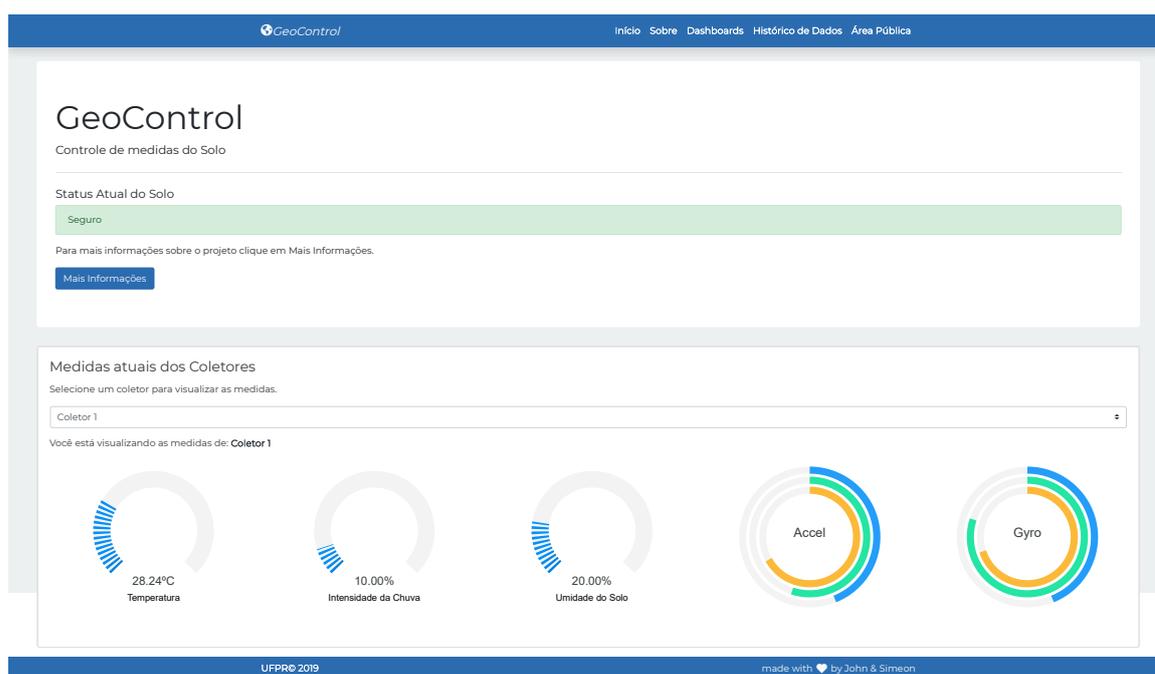
a importância dessa ferramenta para a sociedade em geral e como ela pode evitar catástrofes de eventos de deslizamento de terra.

As leituras obtidas são salvas numa tabela contida na aba chamada histórico de dados

Por fim foi implementado um algoritmo de comunicação entre a página e o micro controlador que recebe os dados captados pelos sensores do protótipo e atualiza automaticamente as leituras obtidas enviando esses valores coletados também aos gráficos.

Na figura 5.7 é possível visualizar a página de início da aplicação exibindo as leituras obtidas pelos sensores em tempo real, sendo elas a umidade do solo, temperatura, aceleração em modulo e nos 3 eixos, inclinação, nível de chuva, e giroscópio.

FIGURA 5.7 – PROTÓTIPO DA PÁGINA INICIAL DA APLICAÇÃO.

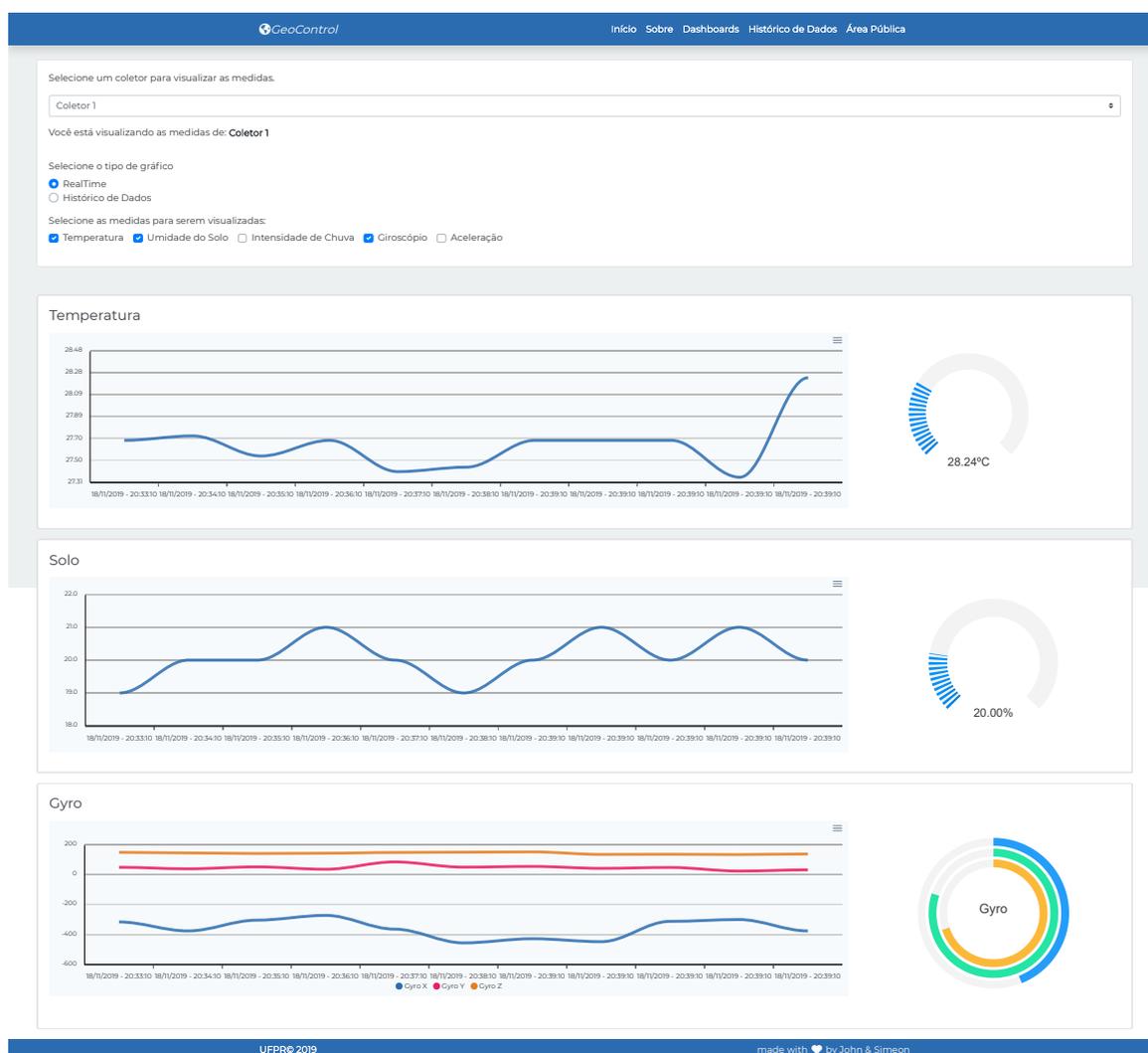


Fonte: (AUTORIA PRÓPRIA)

Na figura 5.8 é possível observar os gráficos de umidade, temperatura que estão contidos na rota Dashboard que agrupa além deles, os gráficos de temperatura, nível de chuva, aceleração e inclinação. Os gráficos são gerados dinamicamente em tempo real. Os gráficos são parte da proposta de visualização dos dados não somente em tempo real, mas também das medições por períodos de tempo, auxiliando o usuário a obter análises que servirão como base de dados históricos juntamente com os valores

contidos nas tabelas para estabelecer parâmetros de fatores de risco e geração de alertas a população e os órgãos competentes como a policia, governo e defesa cível.

FIGURA 5.8 – PROTÓTIPO DA PÁGINA DA APLICAÇÃO EXIBINDO AS LEITURAS FEITAS EM TEMPO REAL.



Fonte: (AUTORIA PRÓPRIA)

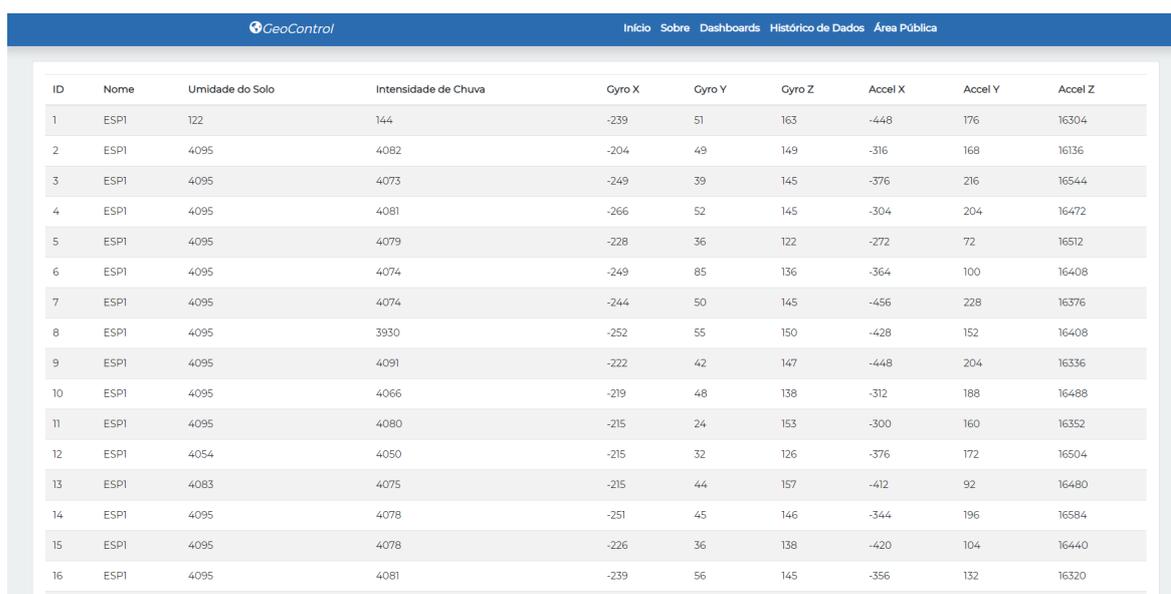
Na figura 5.9 é possível observar a responsividade da página e os conceitos de PWA aplicados em *smartphones*, geraram uma melhor adaptação da página ao navegador tomando um disposição similar a um aplicativo de *smartphone*, porém implementado no navegador. Na figura 4.2 especificamente, é exposta a visão geral da página na resolução de *smartphones* da tela inicio, onde contem as medidas dos sensores atualizando em tempo real contando com as mesmas funcionalidades da aplicação exposta no computador, porém com tamanho de tela reduzido.

5.5 ARMAZENAMENTO DE DADOS

Existe um modelo que recebe os dados do coletor e envia para o banco de dados SQL no seguinte formato presente na figura 5.10:

Para o armazenamento de dados foi implementado o banco de dados que é chamado pelo ORM que irá receber os dados obtidos juntamente com o horário de leitura para que posteriormente esses dados possam ser usados como uma base de predição e contingenciamento de deslizamentos de encostas. Além dos dados obtidos por meio dos sensores, o banco de dados lê e armazena os momentos que houveram eventos de deslizamentos de encostas, assim, o usuário pode verificar qual foi fator desencadeante de cada um dos eventos e assim por meio deles gerar os alertas.

FIGURA 5.11 – PROTÓTIPO DA PÁGINA DA APLICAÇÃO NO EXIBINDO O ARMAZENAMENTO DE DADOS.

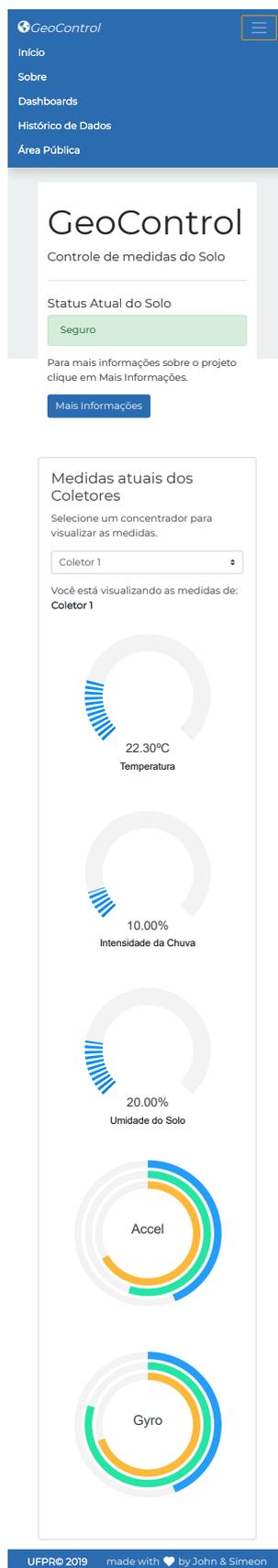


ID	Nome	Umidade do Solo	Intensidade de Chuva	Gyro X	Gyro Y	Gyro Z	Accel X	Accel Y	Accel Z
1	ESPI	122	144	-239	51	163	-448	176	16304
2	ESPI	4095	4082	-204	49	149	-316	168	16136
3	ESPI	4095	4073	-249	39	145	-376	216	16544
4	ESPI	4095	4081	-266	52	145	-304	204	16472
5	ESPI	4095	4079	-228	36	122	-272	72	16512
6	ESPI	4095	4074	-249	85	136	-364	100	16408
7	ESPI	4095	4074	-244	50	145	-456	228	16376
8	ESPI	4095	3930	-252	55	150	-428	152	16408
9	ESPI	4095	4091	-222	42	147	-448	204	16336
10	ESPI	4095	4066	-219	48	138	-312	188	16488
11	ESPI	4095	4080	-215	24	153	-300	160	16352
12	ESPI	4054	4050	-215	32	126	-376	172	16504
13	ESPI	4083	4075	-215	44	157	-412	92	16480
14	ESPI	4095	4078	-251	45	146	-344	196	16584
15	ESPI	4095	4078	-226	36	138	-420	104	16440
16	ESPI	4095	4081	-239	56	145	-356	132	16320

Fonte: (AUTORIA PRÓPRIA)

O projeto foi disponibilizado de forma pública no Azure DevOps por meio do seguinte Link.

FIGURA 5.9 – PROTÓTIPO DA PÁGINA INICIAL DA APLICAÇÃO PARA DISPOSITIVOS MÓ-VEIS.



Fonte: (AUTORIA PRÓPRIA)

FIGURA 5.10 – MODELO DE RECEPÇÃO DOS DADOS DO COLETOR.

```
const sensor = sequelize.define('reads', {
  id: {
    type: Sequelize.INTEGER,
    primaryKey: true,
    autoIncrement: true,
  },
  name: {
    type: Sequelize.TEXT
  },
  soil: {
    type: Sequelize.INTEGER
  },
  rain: {
    type: Sequelize.INTEGER
  },
  temp: {
    type: Sequelize.REAL
  },
  gyx: {
    type: Sequelize.INTEGER
  },
  gyy: {
    type: Sequelize.INTEGER
  },
  gyz: {
    type: Sequelize.INTEGER
  },
  acx: {
    type: Sequelize.INTEGER
  },
  acy: {
    type: Sequelize.INTEGER
  },
  acz: {
    type: Sequelize.INTEGER
  },
  time: {
    type: Sequelize.DATE
  }
})
```

Fonte: (AUTORIA PRÓPRIA)

6 CONCLUSÃO

No decorrer desse trabalho, que visa produzir uma ferramenta para monitoramento de grandezas do solo, foi possível inicialmente ter progresso na criação do protótipo do dispositivo, onde foram definidos os sensores de algumas grandezas que serão úteis, como por exemplo o giroscópio acelerômetro e o sensor de umidade, que irão respectivamente controlar o ângulo da encosta, variação na aceleração se houver um evento, e a umidade do solo.

Com auxílio do framework Vue.js o trabalho de produção da aplicação foi facilitado, pois o mesmo conta com modelos que implementam a reatividade da estilização da página em vários tamanhos de tela, também com isso foi facilitado o desenvolvimento em HTML e CSS, produção de barras de organização do site e a PWA que funciona em dispositivos móveis tanto como aplicação no navegador, como um aplicativo nativo.

Com auxílio do Arduino, que neste projeto é programado em Javascript, e das tecnologias NodeJS com uso dos *frameworks* Johnny Five e Express, foi implementado um algoritmo de comunicação entre os sensores e a página *Web* que usa a tecnologia *Web Socket*

Com a união da página e do algoritmo de comunicação foi implementado um sistema de monitoramento em tempo real, o qual além de mostrar os dados captados, gera gráficos trazendo uma visualização e análise temporal dos dados. Os gráficos são responsivos, sendo atualizados automaticamente em um intervalo específico trazendo uma dinâmica e interação entre usuário e a ferramenta maior. Por meio desses gráficos gerados e dos dados obtidos em tempo real, foi possível viabilizar a implementação do sistema de alerta de deslizamento de encostas, e registrar esses eventos e todos os seus parâmetros em um banco de dados, nele todos os dados obtidos como por exemplo grandezas de umidade, temperatura, orientação, e aceleração são salvas juntamente com seu horário e servem de base para que essas informações sejam exibidas, por meio de gráficos ou tabelas.

Para a execução e sucesso do projeto foi implementado o uso de metodologias de engenharia de software como por exemplo a refatoração de código, que traz vantagens como por exemplo o aumento da produtividade, controle de versão através do Git, tendo como principal vantagem a possibilidade de retornar a uma versão anterior de

desenvolvimento, e os testes de software no momento de desenvolvimento, que traz vantagens como a facilidade de localizar a fonte de comportamentos inesperados no software.

Foi desenvolvida uma plataforma física para o projeto visando a simulação do protótipo que contem uma placa de MDF, na qual esta fixada uma caixa que irá conter terra. e um cano de PVC de 30 milímetros de comprimento contendo 30 centímetros de largura no qual foram fixados os sensores, microcontrolador, e a fonte que auxilia na utilização da ferramenta.

Quanto à calibração dos sensores para diminuir a taxa de erro do sistema, foram escolhidos sensores com facilidade de calibração para facilitar os ajustes, como por exemplo o higrômetro e o sensor de chuva que podem ser calibrados por meio de um potenciômetro.

Foi possível adquirir o conhecimento necessário sobre os desabamentos e deslizamentos de terra para que se pudesse ser elaborado uma ferramenta efetiva para o contingenciamento e predição de catástrofes atingindo assim o objetivo maior que é diminuir as perdas humanas, ambientais com uma ferramenta de baixo custo e usar a tecnologia a favor da sociedade para resolução desse problema.

Por fim, retomando os objetivos específicos:

1. Desenvolver o mínimo produto viável do dispositivo de monitoramento:

Foi possível por meio de dispositivos de alta disponibilidade no mercado e baixo custo, desenvolver uma ferramenta que atender as especificação proposta de trazer um dispositivo de menor custo, aliado a uma interface moderna com potencial de escalabilidade.

O dispositivo implementa o sistema de controle de maneira geral, podendo a partir deste dispositivo mínimo, gerar demandas de mais sensores e até dispositivos semelhantes aplicados a outros problemas presentes na sociedade.

2. Criar uma interface de visualização de dados multiplataforma:

O painel de controle conta com a tecnologia *Progressive Web App* que juntamente com o framework VueJS trouxe o conceito de reatividade a pagina, ou seja, os elementos gráficos se adaptam ao tamanha da tela de visualização.

Além disso, a PWA, pode ser salva na tela inicial dos dispositivos Mobile e funcionar como um aplicativo nativo, usando o mesmo código da aplicação Web. Esse conceito, trouxe a vantagem do reaproveitamento de código e a não necessidade de fazer manutenções de código para dispositivos moveis.

3. Gerar gráficos com os dados obtidos:

Por meio de toda a flexibilidade presente nas ferramentas de desenvolvimento *Web*, e auxilio da biblioteca ApexCharts, foi possível gerar graficos e indicadores das medições em tempo real na página.

A alta compatibilidade da biblioteca com a linguagem Javascript e o Framework usado para a construção do painel de controle, facilitaram a implementação e trouxeram o bom funcionamento dessa funcionalidade.

4. Modelar uma arquitetura distribuída e escalável para a aplicação:

Para implementação de um dispositivo distribuído foi desenvolvida uma plataforma que realiza a segregação dos dispositivos coletores. A ausência de fio na comunicação entre o coletor e a nuvem MQTT faz que o conceito de distribuição esteja aplicado a esse protótipo.

O uso da nuvem pública traz como vantagens a possibilidade de escalabilidade da aplicação sem ter que alterar a infra estrutura presente, confiabilidade e disponibilidade da aplicação.

5. Avaliar a viabilidade de implementação de um algoritmo de predição de eventos de desabamento.

Os modelos de predição múltiplos, para que sejam eficientes demandam de um alto nível de treinamento para que se obtenham os resultados acertivos. Atualmente as bibliotecas de aprendizado maquina abstraem a complexidade presente na produção de códigos desses modelos.

Porém, para que o sistema ganhe maturidade, é necessário um volume alto de dados de treinamento juntamente com a filtragem dos dados que são relevantes para um evento de deslizamento de solos. Para geração desses dados é necessário fazer que o dispositivo enfrente eventos de deslizamentos em um elevado numero de vezes o que torna inviável fazer o treinamento em campo.

O sistema de monitoramento remoto de solos tem viabilidade de implementação de um algoritmo de predição de eventos de deslizamentos de solo desde que o modelo seja treinado empiricamente.

REFERÊNCIAS

ACADEMY, D. S. *O QUE É O TENSORFLOW MACHINE INTELLIGENCE PLATFORM?* 2018. <<http://datascienceacademy.com.br/blog/o-que-e-o-tensorflow-machine-intelligence-platform/>>. Citado na página 60.

ACTION. *Análise de Regressão*. 2018. <<http://www.portalaction.com.br/analise-de-regressao>>. Citado 2 vezes nas páginas 46 e 47.

ALEXANDRE, S. *Acelerômetro e Giroscópio MPU6050*. 2015. <<https://www.arduinoecia.com.br/2015/04/acelerometro-giroscopio-mpu-6050.html>>. Citado na página 51.

ALVIM, R. *Tecnologias para prevenção de deslizamentos de terra*. [S.l.: s.n.], 2011. Citado 3 vezes nas páginas 24, 25 e 26.

AZURE, M. *O que é computação em nuvem?* 2017. <<https://azure.microsoft.com/pt-br/overview/what-is-cloud-computing/>>. Citado 2 vezes nas páginas 46 e 56.

BITTENCOURT, D. M. A. *Estabilidade de Taludes*. 2018. <http://professor.pucgoias.edu.br/SiteDocente/admin/arquivosUpload/17430/material/GEO_II_13_Estabilidade%20de%20Taludes.pdf>. Acesso em 28 mar. 2018. Citado na página 30.

BOBROWSKY, L. M. H. P. *The Landslide handbook - A guide to understanding landslides*. [S.l.]: United States Geological Survey, 2008. Citado 3 vezes nas páginas 23, 24 e 25.

CEA. *Movimento de Massa*. 2011. <<https://centrodeestudosambientais.wordpress.com/tag/regiao-serrana/>>. Citado na página 22.

CEMADEN. *Movimento de Massa*. 2016. <<https://www.cemaden.gov.br/deslizamentos/>>. Citado na página 19.

CURTOCIRCUITO. *CONHECENDO O ESP32*. 2018. <<https://www.curtocircuito.com.br/blog/conhecendo-esp32/>>. Citado na página 54.

DEV MEDIA. *ORM : Object Relational Mapper*. 2011. <<https://www.devmedia.com.br/orm-object-relational-mapper/19056>>. Citado 2 vezes nas páginas 42 e 43.

_____. *Introdução ao MongoDB*. 2017. <<https://www.devmedia.com.br/introducao-ao-mongodb/30792>>. Citado na página 45.

DIETRICH, D. R. M. W. E. A digital terrain model for mapping shallow landslide potential. 1998. Citado na página 21.

EIS, D. *Por que VueJS é uma boa opção?* 2014. <<https://tableless.com.br/web-components-introducao/>>. Citado 2 vezes nas páginas 43 e 44.

FETTE, I. *RFC 6455*. 2018. <<https://tools.ietf.org/html/rfc6455>. Acesso em 7 mai. 2018. Citado na página 43.

FILHO, A. C. da C. *Modelo experimental baseado na declividade da encosta para avaliação de risco de deslizamento*. Tese (Doutorado) — Universidade Católica de Pernambuco, 2012. Citado na página 17.

FILHO, R. P. D. A. *Protocolo de Comunicação I2C*. 2018. <<http://www3.eletronica.org/artigos/protocolo-de-comunicacao-i>>. Citado na página 37.

GAUNT, M. *Service Workers: uma Introdução*. 2018. <<https://developers.google.com/web/fundamentals/primers/service-workers/?hl=pt-br>>. Citado 2 vezes nas páginas 40 e 41.

GOERL, R. F. *Elaboração e Aplicação de Índices de Fragmentação e Conectividade da Paisagem para análise de bacias hidrográficas (Development and Application of Landscape Fragmentation and Connectivity Index to Watershed Analysis)*. [S.l.: s.n.], 2011. Citado na página 24.

GUIMARÃES DAVID MONTGOMERY, H. M. N. F. R. A. T. G. O. A. C. J. R. F. Parameterization of soil properties for a model of topographic controls on shallow landsliding. 2003. Citado na página 21.

HAYERBEKE, M. *Eloquent JavaScript*. [S.l.]: No Starch Press., 2014. Citado 3 vezes nas páginas 37, 41 e 43.

HORSTMANN, C. *Conceitos de Computação com Java*. [S.l.]: Bookman., 2009. Citado 4 vezes nas páginas 32, 33, 34 e 35.

LEROUEIL, S. *Discussion on the paper "The general and congruent effects of structure in natural soils and weak rocks"*. [S.l.: s.n.], 1991. Citado na página 29.

_____. *Ageotechnical characterization of slope movements*. [S.l.]: International symposium of Landslides, 1996. Citado 5 vezes nas páginas 20, 22, 27, 28 e 29.

LIMA, A. F. *Comportamento geomecânico e análise de estabilidade de uma encosta da formação barreiras na área urbana da cidade do Recife*. 2002. Citado 3 vezes nas páginas 17, 18 e 29.

MAJER, C. *Introdução ao Desenvolvimento WEB*. [S.l.: s.n.], 2009. Citado na página 38.

MICHEL, G. P. *Estudo de escorregamento na bacia do rio Cunha, Rio dos cedros/SC com os modelos de Shalstab e Inmap*. [S.l.: s.n.], 2011. Citado na página 23.

MICROPYTHON. 2018. <<https://micropython.org>>. Citado na página 37.

MOTA, E. *O que são as PWA e de que modo mudarão a forma como usamos os nossos dispositivos?* 2018. <<https://pplware.sapo.pt/software/pwa-mudarao-forma-usamos-dispositivos/>>. Citado 2 vezes nas páginas 39 e 40.

MOZILLA, C. individuais e. *Canvas | MDN*. 2018. <https://developer.mozilla.org/pt-BR/docs/Web/Guide/HTML/Canvas_tutorial. Acesso em 18 out. 2018. Citado na página 38.

_____. *Express / MDN*. 2018. <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction. Acesso em 28 mar. 2018. Citado na página 39.

_____. *HTML / MDN*. 2018. <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>. Acesso em 25 mar. 2018. Citado na página 38.

_____. *HTTP / MDN*. 2018. <<https://developer.mozilla.org/pt-BR/docs/Web/HTTP>. Acesso em 26 mar. 2018. Citado na página 35.

_____. *WebSockets / MDN*. 2018. <<https://developer.mozilla.org/pt-BR/docs/WebSockets>. Acesso em 7 mai. 2018. Citado na página 43.

NETTO, E. L. C. *Natural Hazards and Human-Exacerbated Disasters in Latin America*. [S.l.]: Elsevier Science, 2009. Citado na página 20.

NODEBR. *Por que VueJS é uma boa opção?* 2016. <<http://nodebr.com/nodejs-e-mongodb-introducao-ao-mongoose/> >. Citado na página 45.

_____. *Service Workers: uma Introdução*. 2016. <<http://nodebr.com/o-que-e-a-npm-do-nodejs/> >. Citado na página 41.

POPE, A. M. G. *The incorporation of soil water models into geotechnical studies of landslide behaviour*. [S.l.]: Landslide Symposium, 1984. Citado na página 19.

POZZEBOM, R. *O que GitHub?* 2015. <<https://www.oficinadanet.com.br/post/14791-o-que-github> >. Citado 2 vezes nas páginas 44 e 45.

PRATES, G. *Bulma: framework CSS baseado em flexbox*. 2016. <<https://tableless.com.br/bulma-framework-css-baseado-em-flexbox/> >. Citado na página 39.

REIS, V. *Por que VueJS é uma boa opção?* 2016. <<http://vuejs-brasil.com.br/por-que-vuejs-e-uma-boa-opcao/> >. Citado na página 39.

RODRIGUES, J. *Conhecendo o Azure - Parte 1: App Service*. 2017. <<https://medium.com/@joelrodrigues/azure-web-apps-conhecendo-o-azure-a8bccc997121> >. Citado na página 57.

SILVA, M. M. da. *Estudo geológico-geotécnico de uma encosta com problemas de instabilidade no Município de Camaragibe - PE*. Tese (Doutorado) — Universidade Federal de Pernambuco, 2007. Citado 5 vezes nas páginas 16, 18, 20, 27 e 29.

TANENBAUN, A. S. *Redes de Computadores*. [S.l.: s.n.], 2003. Citado na página 36.

THOMSEN, A. *O sensor de umidade do solo higrômetro*. 2016. <<https://www.filipeflop.com/blog/monitore-sua-planta-usando-arduino/>. Citado 2 vezes nas páginas 52 e 53.

TOWN, D. *Apexcharts and Vue*. 2019. <<https://www.drewtown.dev/post/apexcharts-and-vue/> >. Citado na página 42.

VARNES, D. J. *Landslide types and processes*. 1978. Citado 4 vezes nas páginas 7, 16, 17 e 18.

VEDOVELLI, F. *Vue-router*. 2016. <<https://vuejs-brasil.com.br/vue-router/>>. Citado 2 vezes nas páginas 41 e 42.

_____. *Vuex*. 2016. <<http://vuejs-brasil.com.br/vuex/>>. Citado na página 42.

YUAN, M. *Conhecendo o MQTT*. 2017. <<https://www.ibm.com/developerworks/br/library/iot-mqtt-why-good-for-iot/index.html>>. Citado 2 vezes nas páginas 36 e 37.