UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROJETO DE GRADUAÇÃO

Mayara Nascimento de Oliveira

DESENVOLVIMENTO DE WEBSITE E APLICATIVO DE CELULAR PARA MONITORAMENTO DA QUALIDADE DO AR NA GRANDE VITÓRIA

Vitória-ES

Março/2017

Mayara Nascimento de Oliveira

DESENVOLVIMENTO DE WEBSITE E APLICATIVO DE CELULAR PARA MONITORAMENTO DA QUALIDADE DO AR NA GRANDE VITÓRIA

Parte manuscrita do Projeto de Graduação da aluna Mayara Nascimento de Oliveira, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheira Eletricista.

Vitória-ES

Março/2017

Mayara Nascimento de Oliveira

DESENVOLVIMENTO DE WEBSITE E APLICATIVO DE CELULAR PARA MONITORAMENTO DA QUALIDADE DO AR NA GRANDE VITÓRIA

Parte manuscrita do Projeto de Graduação da aluna Mayara Nascimento de Oliveira, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheira Eletricista.

Aprovado em 02 de Março de 2017.

COMISSÃO EXAMINADORA:

Prof. Dr. Teodiano Freire Bastos
Universidade Federal do Espírito Santo
Orientador

Profa. Dra. Jane M. Santos
Universidade Federal do Espírito Santo
Coorientador

Prof. Dr. André Ferreira
Universidade Federal do Espírito Santo
Examinador

Profa. Dra. Eliete Maria de Oliveira Caldeira
Universidade Federal do Espírito Santo
Examinador

Vitória-ES

Março/2017

Dedico este trabalho primeiramente aos meus pais que sempre acreditaram que eu fosse alcançar os meus sonhos, sendo a finalização deste curso apenas mais um deles. Também gostaria de dedicar este trabalho a todas as pessoas que, por serem do grupo de risco, sofrem constantemente com a baixa qualidade do ar da nossa cidade, e que, através deste trabalho, têm a esperança de conquistar uma melhor qualidade de vida. Obrigada!

AGRADECIMENTOS

Gostaria de agradecer aos meus familiares por compreenderem minha ausência, não somente nos anos da universidade, mas também nos anos que a precederam. Aos meus pais, minha eterna gratidão por me darem todo o suporte necessário para que um dia eu pudesse chegar no lugar em que hoje estou. Vocês, pais, mesmo com uma maneira singular de amar, marcaram minha memória com suas manifestações de amor. Fosse para chamar minha atenção, me levar para o hospital nas madrugadas e até mesmo aguentar meu choro quase que constante quando eu era apenas um bebê. Obrigada por nunca desistirem de mim (mesmo quando as professoras da creche não aguentavam mais meu choro e ficavam tristes quando eu ia para escola).

Ao resto da minha enorme família, muito obrigada. Gostaria de agradecer em especial à minha prima Cíntia por ter ido todos os dias depois do estágio à minha casa, quando eu ainda era um bebê, para que minha mãe pudesse pelo menos tomar um banho. Obrigada tia Jane e Andrea por terem me oferecido suas casas para que eu tivesse onde morar em um dos momentos mais difíceis que já passei, e tudo isso no último ano do Ensino Médio! Sem vocês eu jamais teria conseguido entrar na universidade. Aos meus primos Elen e Andrey, obrigada por todos os conselhos e risadas! Amo vocês!

Também não poderia deixar de agradecer as minhas avós, que, mesmo não estando mais entre nós, marcaram minha vida e infância. Minha avó Aleida, que sabia fazer uma torta capixaba e um escabeche como ninguém, que por muitas vezes ficou comigo e me levou à escola porque meus pais estavam trabalhando, a mulher que, já na casa dos 60 ou 70 anos (não me recordo muito bem) terminou o Ensino Fundamental. Você foi uma vencedora! Minha avó Idália, uma verdadeira guerreira, superou o preconceito e se tornou a matriarca de uma família de mulheres fortes. Mulheres que batalharam para que pudessem superar as dificuldades financeiras e graduar em uma universidade federal. Vocês são verdadeiros exemplos para mim.

Não poderia deixar de agradecer aos meus amigos, alguns com quem estudei desde a primeira série do Ensino Fundamental até o terceiro ano do Ensino Médio. Vocês foram uma verdadeira família para mim. Aos meus amigos que conheci no IFES, vocês são sensacionais e me orgulho muito de quem vocês se tornaram! A gente batalhou muito e já estamos colhendo os frutos da nossa dedicação! Aos amigos que conheci na UFES, obrigada por me ajudarem a passar por esse momento tão estressante. Se hoje estou aqui concluindo este curso, metade dos méritos eu devo a vocês. Um carinho especial aos meus

amigos Silmar, Paula e Luísa! Nossa amizade realmente veio de Deus.

Aos meus amigos do Ciências Sem Fronteiras, vocês foram a minha família no Canadá. Muito obrigada pelas noites de Domino's no ECSA e pelos bolos de aniversário que compartilhamos. Obrigada Lu por ter sido sempre presente e pelas viagens que fizemos juntas. Cat e Gabi, vocês foram verdadeiras companheiras. Ju, Raquel e Breno, muito obrigada pelas loucuras! Graças a vocês terei muitas histórias pra contar! Vocês são meu *squad*! Amo todos vocês!

Por fim, meu agradecimento a todos os meus mestres, em especial a Mario Eugênio, por todo suporte e pela conversa que tivemos dentro do ônibus a caminho do IFES. Conversa que mudou o rumo da minha vida no último ano do Ensino Médio. Muito obrigada por me fazer acreditar que era possível mesmo com todas as adversidades. Meu eterno agradecimento à professora mais linda de todas, Eliane, que, além de ser a melhor professora de máquinas elétricas da vida, ainda é uma super amiga, e sai pra tomar açaí com os alunos mesmo depois de cinco anos. E não poderia deixar de agradecer aos meus incríveis orientadores: Teodiano, que acreditou em mim para desenvolver este projeto e me encorajou em tantas outras coisas, e Jane, pela oportunidade de trabalhar neste tema!

Deus, muito obrigada por me sustentar até aqui e pelo cuidado com a minha vida. Sei que todas as coisas, até mesmo as que eu pensava serem ruins, cooperaram para que eu pudesse finalizar este curso. Deus, Você é "o cara", muito obrigada!

"You matter. And there is a void in the symphony of life when you are silent".

Jon Foreman, TEDx Talk: "Live it like a song".

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

UFES	*Universidade Federal do Espírito Santo*
RGMV	*Região Metropolitana da Grande Vitória*
IEMA	*Instituto Estadual de Meio Ambiente*
PS	*Partículas Sedimentadas*
RAMQAr	*Rede Automática de Monitoramento da Qualidade do Ar*
PTS	*Partículas Totais em Suspensão*
MP10	*Partícula com Diâmetro Inferior a 10μm*
HC	*Hidrocarbonetos*
NOX	*Nitróxidos*
OMS	*Organização Mundial de Saúde*
CONAMA	*Conselho Nacional do Meio Ambiente*
MI1	*Medida Intermediária 1*
MI2	*Medida Intermediária 2*
MI3	*Medida Intermediária 3*
MP2.5	*Partículas com Diâmetro Inferior a 2.5μm*
IQA	*Índice de Qualidade do Ar*
CSS	Cascading Styke Sheets
API	Application Programming Interface
W3C	World Wide Web Consortium
OMM	*Organização Metereológica Mundial*
SGML	Standard Generalized Markup Language
HTML	HyperText Markup Language
XHTML	Extensible HTML

XML	Extensible Markup Language
WHATWD	Web HyperText Application Technology Working Group
SEO	Search Engine Optimization
IBGE	*Instituto Brasileiro de Geografia e Estatística*
RDBMS	Relational Database Management System
SQL	Structured Query Language
MER	*Modelo Entidade Relacionamento*
DER	*Diagrama Entidade Relacionamento*
CASE	Computer-Aided Software Engineering
MVC	Model View Controller
ADK	Android Package
HAL	Hardware Abstraction Layer
SSH	Secure Shell Access
CORS	Cross-Origin Resource Sharing
NTI	*Núcleo de Tecnologia da Informação*

SUMÁRIO

1 INTRODUÇÃO

Há várias formas nas quais a poluição atmosférica antropogênica (causada pelo homem) pode se manifestar, sendo algumas visíveis e outras não. De fato, qualquer substância introduzida na atmosfera e que afeta negativamente os seres vivos e o meio ambiente é considerada poluição atmosférica. Na Região Metropolitana da Grande Vitória (RMGV) a poluição atmosférica gera incômodos e pode ser prejudicial a saúde da população em geral. A maioria das pessoas sentem-se incomodadas devido às partículas sedimentadas (PS), ou seja, poeiras visíveis que se depositam em superfícies de uso cotidiano. Contudo, há grupos de riscos formados por pessoas com asma, problemas cardíacos e doenças pulmonares que são extremamente sensíveis aos poluentes atmosféricos em geral, e, dependendo da concentração de tais poluentes, podem ter a qualidade de vida deteriorada (IEMA, 2013).

A fim de monitorar as concentrações dos principais poluentes atmosféricos e informar a população a respeito da qualidade do ar, o Instituto Estadual de Meio Ambiente (IEMA) inaugurou em 2001 a Rede Automática de Monitoramento da Qualidade do Ar (RAMQAr). Os dados coletados pela RAMQAr são divulgados diariamente em jornais da Grande Vitória e através do website do IEMA. Posteriormente, em 2009, o IEMA inaugurou a Rede Manual de Monitoramento das Partículas Sedimentadas, com vista a monitorar os índices de poeira da RMGV (IEMA, 2013).

A RAMQAr monitora os seguintes poluentes atmosféricos: partículas totais em suspensão (PTS), partículas com diâmetro inferior a 10μm (MP10), dióxido de enxofre (SO2), nitróxidos (NOX), hidrocarbonetos (HC), monóxido de carbono (CO), ozônio (O3) e, em algumas estações, partículas com diâmetro inferior a 2.5μm (MP2.5). Além dos poluentes, algumas estações também mensuram variáveis climáticas como direção do vento, temperatura, umidade, radiação solar, pressão atmosférica e precipitação.

1.1 Justificativas para o Trabalho

Os poluentes monitorados pela RAMQAr podem causar diversos efeitos adversos à saúde, incluindo problemas respiratórios e cardiovasculares, sendo que a sensibilidade a esses poluentes depende das condições de saúde e da faixa etária de cada indivíduo exposto a poluição (IEMA, 2013).

Cada poluente causa um efeito diferente à saúde e ao bem-estar da população em geral. Por exemplo, as partículas sedimentadas geram incômodos à população por serem visíveis e ficarem expostas nas superfícies de uso cotidiano. Já o SO2 pode gerar o SO3 que, em

presença de água, gera o ácido sulfúrico, sendo este o responsável pela chuva ácida. A chuva ácida, por sua vez, afeta as membranas mucosas do nariz e o trato respiratório superior, gerando alterações na função pulmonar e sintomas respiratórios em asmáticos. O dióxido de nitrogênio (NO2) reage com a água gerando ácido nítrico, sendo este o precursor do nitrato, causador da redução da capacidade de defesa do sistema respiratório, de injúrias e de inflamações (IEMA, 2013).

O ozônio, por sua vez, tem efeitos agudos sobre os sistemas cardiovasculares e pulmonares (inflamação e redução das defesas pulmonares), além de efeitos crônicos como a redução da função pulmonar e desenvolvimento de arteriosclerose e de asma. As MP10 podem causar danos na região do nariz, da nasofaringe e no sistema respiratório como um todo, e as MP2.5 podem chegar até aos bronquíolos pulmonares, gerando problemas no sistema respiratório. Por fim, o CO interfere no transporte de oxigênio pelo sangue, reduzindo a capacidade aeróbica do organismo, além de agravar doenças cardiovasculares (IEMA, 2013).

Ainda, de acordo com pesquisa realizada pelo *Institute for Health Metrics and Evaluation*, localizado em Washington, Estados Unidos, e envolvendo cerca de dois mil pesquisadores, apenas no ano de 2015 mais de quatro milhões de pessoas morreram devido a poluição do ar, número maior do que o estimado pela Organização Mundial de Saúde (OMS). A maioria das mortes, cerca de um milhão, ocorreram em países como Índia e China, porém, o Brasil encontra-se em décimo lugar, com cerca de cinquenta e duas mil mortes causadas pela má qualidade do ar (Jornal Hoje, 2017).

Devido à gravidade dos problemas gerados pela poluição atmosférica, tornou-se necessária a criação de uma legislação que regulasse a emissão de gases poluentes. No âmbito internacional, a OMS produziu diretrizes para a qualidade do ar. No âmbito nacional, os padrões de qualidade do ar são estabelecidos em resoluções do Conselho Nacional do Meio Ambiente (CONAMA), e a principal resolução foi aprovada em 1990 dispondo os dois padrões de qualidade do ar. Os padrões primários são aqueles cujas concentrações dos poluentes, se ultrapassadas, podem afetar a saúde da população. Já os padrões secundários são aqueles cujas concentrações dos poluentes se encontram abaixo do valor que prevê o mínimo efeito adverso sobre a saúde da população. Este último pode ser considerado uma meta a longo prazo, porém, cada estado brasileiro pode ter sua própria legislação e seus próprios padrões (IEMA, 2013).

O CONAMA, além de regulamentar os padrões de qualidade do ar, também regulamenta as condições de medição e os equipamentos referenciais a serem utilizados para o monitoramento, sendo que tal monitoramento está a cargo dos governos estaduais.

O Governo do Estado do Espírito Santo decretou em 2013 padrões estaduais de qualidade do ar que incluem Metas Intermediárias (MI1, MI2 E MI3), além de padrões finais baseados nas diretrizes da OMS. Como o padrão CONAMA foi estabelecido há mais de 15 anos, este tornou-se obsoleto, e as diferenças nos padrões CONAMA e OMS são gritantes. Por exemplo, o padrão CONAMA apresenta limite para a MP10 três vezes superior à diretriz da OMS. Isso explica o porquê da escolha das diretrizes da OMS pelo Governo do Estado do Espírito Santo (IEMA, 2013).

Os padrões para cada poluente foram definidos de acordo com os danos causados à saúde. Para alguns poluentes como os hidrocarbonetos (HCT e HCMN) não há diretrizes para o estabelecimento de padrões, já que estes não são legislados, mesmo assim tais poluentes são monitorados pela RAMQAr.

Todos esses fatores motivaram o desenvolvimento deste projeto de graduação que consiste no desenvolvimento de um website e de um aplicativo que permitam transmitir, da forma mais transparente possível à população, as concentrações dos poluentes em um determinado local da Grande Vitória e em um determinado horário do dia. Com isso, as pessoas de grupos sensíveis como os asmáticos e pessoas com bronquite e rinite poderiam estar alertas às concentrações dos poluentes e tomar medidas de precaução, reduzindo, assim, o número de internações e os gastos com saúde.

O aplicativo e website desenvolvidos, têm o potencial de mostrar com total transparência as concentrações de cada poluente e também o Índice de Qualidade do Ar (IQA), o qual é baseado no índice mais elevado entre todos os poluentes medidos em cada estação (IEMA, 2013).

Os benefícios deste projeto, atuando em conjunto com as redes de monitoramento e o cumprimento da legislação vigente, são diversos, incluindo uma relação de maior confiança entre governo e população, além, de, em longo prazo, diminuir gastos com saúde pública. Com o desenvolvimento deste projeto tornou-se possível seguir a tendência de outras cidades, como Santiago no Chile, Beijing na China, Londres no Reino Unido e São Paulo no Brasil, as quais têm tido uma boa experiência com esse tipo de serviço, e o feedback da população tem sido positivo.

2 EMBASAMENTO TEÓRICO

2.1 Caracterização das Estações de Monitoramento

2.1.1 Localização

A RAMQAr é composta por oito estações estrategicamente localizadas nas regiões de Laranjeiras, Jardim Camburi, Carapina, Enseada do Suá, Centro de Vitória, Cariacica, Ibes e Centro de Vila Velha (Figuras 1 e 2). Como já citado no item 1, esta rede monitora os principais poluentes atmosféricos, tais como PTS, MP10, SO2, NOX, hidrocarbonetos (HC), CO, O3 e MP2.5. Contudo, o número de estações ainda não é suficiente para monitorar todo o território da RMGV, já que apenas quatro entre os sete municípios da região possuem estações em pontos estratégicos (IEMA, 2013).

Figura 1 – *Localização das Estações RAMQAr*

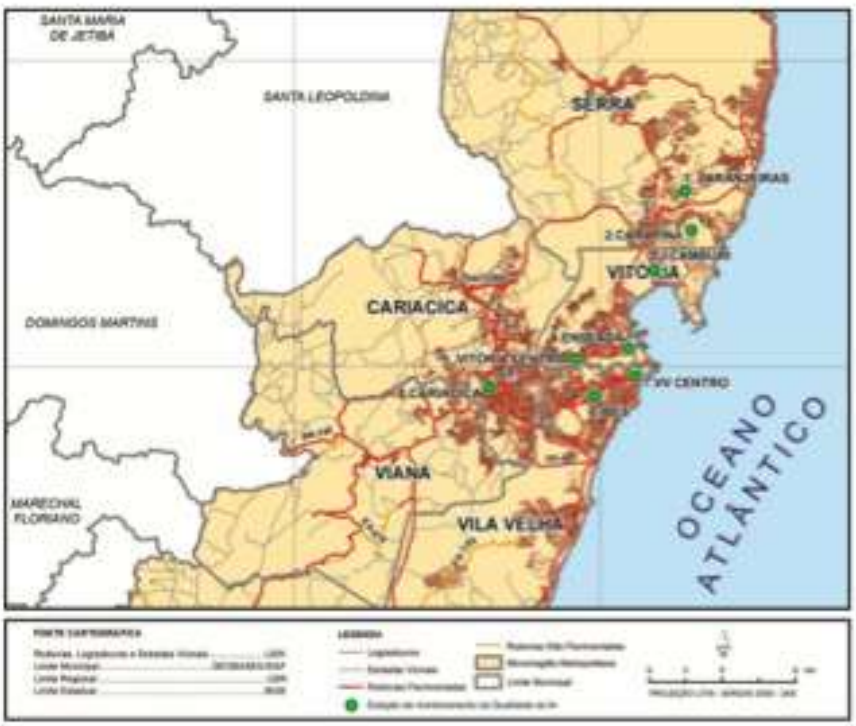
Nome da estação	Localidade/Bairro	Município
Laranjeiras – RAMQAr 01	Hospital Dório Silva / Laranjeiras	Serra
Carapina – RAMQAr 02	ArcelorMittal Tubarão / Carapina	Serra
J. Camburi – RAMQAr 03	Unidade de Saúde / Jardim Camburi	Vitória
Enseada do Suá – RAMQAr 04	Corpo de Bombeiros / Enseada do Suá	Vitória
Vitória Centro – RAMQAr 05	Ministério da Fazenda / Centro	Vitória
Ibes – RAMQAr 06	4° Batalhão da Polícia Militar / Ibes	Vila Velha
VV Centro – RAMQAr 07	Ao lado do Colégio Marista N. S. da Penha / Centro	Vila Velha
Cariacica – RAMQAr 08	CEASA / Coord. da Defesa Agropecuária / Vila Capixaba	Cariacica

Fonte: IEMA, 2013.

Além disso, as estações que medem O3 estão muito próximas de vias de tráfego intenso, o que aumenta a concentração de nitróxidos, que, por sua vez, reduzem as concentrações atmosféricas locais de O3. A proximidade dessas estações com edifícios também acaba sendo um problema já que isso prejudica o monitoramento da qualidade do ar e das variáveis meteorológicas (IEMA, 2013).

Já a Rede Manual de Monitoramento de Partículas Sedimentadas entrou em operação com nove estações de monitoramento e passou posteriormente a operar com onze estações, sendo as oito da RAMQAr, duas na comunidade da Ilha do Boi (Hotel Ilha do Boi e Clube Ítalo Brasileiro) e um ponto adicional no Centro de Vitória, que atualmente está desativado (Figuras 3 e 4).

Figura 2 – *Localização Geográfica das Estações RAMQAr*



Fonte: IEMA, 2013.

Figura 3 – *Localização das Estações da Rede Manual de Partículas Sedimentadas*

Nome da estação	Localidade/Bairro	Município
Laranjeiras – Estação 01	Hospital Dório Silva / Laranjeiras – RAMQAr 01	Serra
Carapina – Estação 02	ArcelorMittal Tubarão / Carapina – RAMQAr 02	Serra
J. Camburi – Estação 03	Unidade de Saúde / Jardim Camburi – RAMQAr 03	Vitória
Enseada do Suá – Estação 04	Corpo de Bombeiros / Enseada do Suá – RAMQAr 04	Vitória
Vitória Centro – Estação 05	Ministério da Fazenda / Centro – RAMQAr 05	Vitória
Vitória Centro (Banca) – Estação 06	Banca de Jornais Cultura / Em frente a C&A / Centro	Vitória
Ibes – Estação 07	4º Batalhão da Polícia Militar / Ibes – RAMQAr 06	Vila Velha
VV Centro – Estação 08	Ao lado do Colégio Marista N. S. da Penha / RAMQAr 07	Vila Velha
Cariacica – Estação 09	CEASA / Coord. da Defesa Agropecuária / Vila Capixaba – RAMQAr 08	Cariacica
SENAC – Estação 10	Hotel SENAC Ilha do Boi / Ilha do Boi	Vitória
Clube Ítalo – Estação 11	Clube Ítalo Brasileiro do Espírito Santo / Ilha do Boi	Vitória

Fonte: IEMA, 2013.

Figura 4 – *Localização das Estações da Rede Manual de Partículas Sedimentadas*

Fonte: IEMA, 2013.

Cada estação de monitoramento conta com quatro coletores, possibilitando, assim, a rejeição de amostras fora do padrão e a verificação dos dados de quantificação da massa depositada, para posterior caracterização química das partículas. O maior problema desta rede manual é que o recipiente deve ficar exposto por 30 dias, de acordo com a norma ASTM D1739, o que implica que a população não tenha um controle diário das Partículas Sedimentadas (IEMA, 2013).

Por essa razão nem o website e nem aplicativo aqui desenvolvidos trabalharão com os dados fornecidos pela Rede Manual, já que o objetivo é oferecer dados confiáveis da qualidade do ar a cada hora. Porém, já há um projeto para a automatização dessa Rede Manual de Monitoramento de Partículas Sedimentadas, o qual é uma parceria entre o Departamento de Engenharia Elétrica e o Departamento de Engenharia Ambiental da Universidade Federal do Espírito Santo com o IEMA, e é dirigido pelo professor Celso José Munaro. Com isso, futuramente, os dados a respeito das partículas sedimentadas poderão ser também fornecidos em tempo real.

2.1.2 Coleta de Dados

A captação de dados é feita utilizando equipamentos que captam grande quantidade de informações brutas, segundo a segundo. Essas informações são disponibilizadas através de um *datalog* que disponibiliza os dados de hora em hora. Os dados coletados utilizam o código ASCII, e cada dado contém uma série de informações como data, hora, *tag* (indica qual poluente foi medido), valor lido, *flag* e um comentário. Todas essas informações são separadas por ponto e vírgula (";").

As *flags* (Figura 5) servem para indicar se um valor lido é coerente e pode ser considerado representativo. Porém, em alguns casos, há a necessidade da verificação de um operador, por exemplo, quando há variação brusca na concentração de um poluente em uma determinada região. Neste caso, o operador deve levar em conta situações estatísticas e algumas situações que podem ser verdadeiras, mas que não descrevem a realidade do bairro como um todo. Além disso, problemas na calibração do equipamento utilizado na coleta, como a existência de *offset*, só podem ser verificados por um operador que ajusta e recalcula os dados que foram divulgados erroneamente.

Cada estação conta com um modem de transmissão de dados, sendo que o IEMA possui um coletor de dados chamado Miglis que tem a função de fazer a comunicação com cada estação via linha telefônica comum discada. O que acontece de fato é que um modem presente no IEMA "liga" para o modem das estações de hora e hora, e cada estação responde inserindo a informação no banco de dados. O Miglis faz um processo de validação nos dados e, após isso, envia todos os dados recebidos para o Atmus, que é um servidor multiusuário formado por diversos bancos de dados, que armazena todo o histórico das informações, sendo que o operador pode mudar, posteriormente, os valores já registrados.

Para a difusão das medições no *website* e aplicativo, o Miglis, que já armazena os dados em um *buffer*, deve ser reprogramado para criar um novo caminho para a duplicação da informação. Com isso, seria possível utilizar um critério de validação diferente do usado atualmente pela prestadora de serviços (empresa Ecosoft).

Figura 5 – *Descrição das* Flags *Utilizadas*

Descrição do *Flag*	*Flag*	Descrição do *Flag*	*Flag*
Válido		Falta de energia elétrica	IP
Invalidado pelo sistema	IS	Instrumento desligado	ID
Invalidado pelo equipamento	IE	Instrumento com defeito	IF
Invalidado pelo gerente	IG	Instrumento descalibrado	IO
Invalidado pelo usuário	IU	Interferência local	IL
Validado pelo usuário	VU	Valor fora da faixa de leitura	IR
Inserido pelo usuário	VI	Suspeito pelo sistema	?S
Validado pelo gerente	VG	Valor abaixo limite detecção	<
Suspeito pelo equipamento	?E	Valor acima fundo escala	>
Suspeito pelo gerente	?G	Valor calculado por fórmula	VC
Suspeito pelo usuário	?U	Editado pelo usuário	VE
Não requisitado	IN	Validado pelo sistema	VS
Instrumento em calibração	IC	Dado inexistente	!
Dado indisponível	II	Valor retificado	VR
Insuficiente % de dados	I%		

Fonte: IEMA, 2013.

2.1.3 Processamento de Dados

O método de processamento de dados utilizado neste projeto é baseado nas concentrações médias horárias para cada um dos poluentes monitorados. Assim, devem ser feitas médias móveis de 24 horas para os poluentes PTS, MP10 e SO2, médias móveis de 8 horas para os poluentes CO e O3, além de médias horárias para os poluentes CO, O3 e NO2 (é necessário fazer também a média horária do CO e O3 porque as estações estão mal localizadas e isso poderia culminar com medidas errôneas). As médias móveis são utilizadas para comparar com os padrões da legislação atual. A exposição ao poluente deve ser verificada continuamente, pois há 23 médias móveis de 24 horas em um único dia. O Índice de Qualidade do Ar (IQA) de cada estação será divulgado baseado na concentração do pior poluente na respectiva estação.

A representatividade dos dados é baseada no percentual de dados brutos válidos utilizados

no cálculo das médias, sendo que esse critério segue o padrão já utilizado em outros estados e visa a uniformizar o processamento de informações. Além disso, só devem ser divulgados dados representativos. Vale ressaltar que, nos últimos anos foi possível verificar um crescimento no número de dados faltantes no conjunto de dados fornecidos pelos equipamentos das estações (IEMA, 2013), sendo que esta insuficiência de dados válidos dificulta a avaliação da qualidade do ar. Assim, é necessário que a RAMQAr melhore a operação, de forma que a população possa ter acesso a dados representativos com mais frequência.

O IQA é comparado com uma faixa de valores, e a este é atribuído uma característica que pode ser "bom", "moderado", "ruim", "muito ruim" e péssimo (Tabela 1).

Tabela 1 – Classificação do Índice de Qualidade do Ar (IQA)

IQA	Significado
0-40 (Bom)	Atende às recomendações da OMS (Padrões Finais da legislação ambiental do ES), não afetando a saúde da população.
41-80 (Moderado)	Pessoas de grupos sensíveis (crianças, idosos e pessoas com doenças respiratórias e cardíacas) podem apresentar sintomas como tosse seca e cansaço. A população em geral não é afetada.
81 – 120 (Ruim)	A população pode apresentar sintomas como tosse seca, cansaço, ardor nos olhos, nariz e garganta. Pessoas de grupos sensíveis (crianças, idosos e pessoas com doenças respiratórias e cardíacas) podem apresentar efeitos mais sérios na saúde.
121 - 200 (Muito Ruim)	A população pode apresentar agravamento de sintomas como tosse seca, cansaço, ardor nos olhos, nariz e garganta e ainda falta de ar e respiração ofegante. Efeitos ainda mais graves à saúde podem ocorrer em grupos sensíveis (crianças, idoso e pessoas com doenças respiratórias e cardíacas).
$>$ 200 (Péssimo)	A população pode apresentar sérios riscos de manifestação de doenças respiratórias e cardiovasculares. Possibilidade de mortes prematuras em pessoas de grupos sensíveis.

Fonte: IEMA, 2013.

É importante destacar que, de acordo com os relatórios da qualidade do ar dos últimos anos, e utilizando a proposta de IQA apresentada na Tabela 1, verificou-se que a qualidade do ar na RMGV nos últimos anos é moderada na maior parte do tempo (entre 2% e 91% dos dias, dependendo da estação de monitoramento), o que implica que pessoas do grupo sensível (crianças, idosos e indivíduos com doenças cardíacas e pulmonares) possam ter sido afetadas por sintomas como tosse seca e cansaço. Nesses últimos anos, a qualidade do ar foi considerada ruim em 3% dos dias, e, assim, toda a população pode ter apresentado sintomas como tosse seca, cansaço e ardor nos olhos, nariz e garganta.

No momento só é possível comparar as concentrações dos poluentes com as diretrizes da MI1, já que os níveis máximos permitidos na MI2 e MI3 ainda não foram definidos. Por exemplo, para PS o nível máximo permitido pela MI1 é de $14 g/m^2$.mês, porém a maioria da população (entre 84% e 98%) ainda se sente incomodada com tal concentração (IEMA, 2013). Isso indica que, apesar, da legislação estadual ser restritiva quando comparada com outros estados da União, ainda há um longo caminho a percorrer.

2.2 O *Website*

2.2.1 Introdução ao HTML e ao CSS

O *website* desenvolvido neste trabalho pode ser acessado pela Internet, a qual pode ser definida como uma rede de computadores conectados que transmitem e recebem dados a uma velocidade próxima à velocidade da luz (Craig et al. , 2012).

Uma das subdivisões da Internet é a *World Wide Web*, além desta última também povoam a Internet os e-mails, os grupos e as salas de bate-papo, por exemplo. Há alguns anos apenas arquivos de texto eram transmitidos pela Internet, hoje, porém, através do mesmo canal circulam vídeos, imagens e áudios. O coração de toda Internet ,contudo, é baseada em códigos HTML (Craig et al. , 2012).

O navegador (*browser*) é o software mais utilizado em nossos computadores e é o responsável por interpretar os códigos HTML (CAELUM, 2016). Utilizando o navegador é possível acessar a Internet através dos mais diversos dispositivos, como computadores, celulares e tablets. O *browser* é chamado na linguagem computacional de cliente, pois está solicitando e recebendo um serviço provido pelo servidor (*server*). O *browser* e o *server* são as terminações de uma cadeia conectada pela Internet (Craig et al. , 2012).

HTML é a sigla para *HyperText Markup Language*, e recebeu este nome pela capacidade de interconectar documentos em diferentes locais da Web através do *hyperlink*. A esta dinâmica deu-se o nome de *hypertext*. Também recebeu a denominação de *markup language* pela capacidade de estruturar as páginas Web de uma maneira compreensível pelos humanos (Craig et al., 2012).

A linguagem HTML é marcada pelas *tags* que indicam as diferentes partes de um texto. As *tags* são escritas em um arquivo de texto e são interpretadas pelo navegador a fim de que este mostre os elementos (todo o conteúdo entre a abertura e o fechamento de uma *tag*) de uma forma compreensível ao leitor. As *tags*, porém, não são visíveis na página Web. É possível observar como as *tags* são utilizadas na Figura 5 (Craig et al., 2012).

Figura 6 – *Representação das Tags*

```
<h1>This is a Level One Heading</h1>

<p>This is a paragraph.</p>
```

Fonte: Craig et al., 2012.

Não é necessária uma licença para desenvolver arquivos HTML e esse é um dos motivos pelo qual a Internet se tornou tão poderosa: é possível compartilhar informações gratuitamente e facilmente. Contudo, há algumas regras que os programadores, clientes e servidores devem seguir a fim de proporcionar um serviço confiável ao público em geral. Tais regras são chamadas de *web standards* e são reguladas pelo *World Web Consortium* (W3C), uma organização sem fins lucrativos criada em 1994. Além da linguagem HTML o W3C também regulas as demais linguagens utilizadas na programação Web, e tal serviço é muito importante para garantir o funcionamento de páginas web em diferentes navegadores e sistemas operacionais (Craig et al., 2012)

2.2.1.1 A evolução do HTML

O HTML surgiu em 1990 baseado em uma linguagem chamada de *Standard Generalized Markup Language* (SGML) . Todavia, o HTML só foi descrito formalmente em 1993. Em 1995 surgiu a primeira versão da linguagem chamada de HTML 2.0. E as versões HTML 3.7 e HTML 4.0 vieram em 1997 (Craig et al., 2012).

Na metade dos anos de 1990, a fim de serem mais competitivos, os navegadores não seguiam as regras propostas pelo W3C rigorosamente, consequentemente, os programadores

deveriam programar versões diferentes de suas páginas web para cada tipo diferente de navegador, ou, no pior dos cenários, os *websites* só funcionavam em apenas um *browser*. Nos dias atuais isso praticamente deixou de ocorrer pois os navegadores passaram a seguir as devidas especificações (Craig et al., 2012).

Em 1999 surgiu uma atualização do HTML 4.0 chamada de HTML 4.01 que trouxe pequenas, mas significativas mudanças. O final da década de 1990 também trouxe consigo uma nova linguagem, o *Extensible HTML* (XHTML). O XHTML veio para fazer a transição entre o HTML e o *Extensible Markup Language* (XML), uma nova linguagem de programação que permitia a construção de elementos customizados, além daqueles já padronizados pela linguagem HTML (Craig et al., 2012).

De fato a linguagem XML era mais restritiva, na questão da sintaxe, do que o HTML. Assim, o XHTML 1.0 apenas tornou o HTML mais restritivo, fazendo com que ambos, navegadores e programadores, seguissem os padrões do W3C mais à risca (Craig et al., 2012).

O W3C, então, começou a desenvolver o XHTML 2.0 que prometia uma completa mudança nos padrões de programação Web, porém, esta linguagem nunca foi aceita e os programadores decidiram atualizar a linguagem HTML (Craig et al., 2012), que é a linguagem utilizada neste trabalho.

2.2.1.2 O Surgimento do HTML5

Em 2005 um grupo dentro do W3C denominado de *Web HyperText Application Technology Working Group* (WHATWG) começou a trabalhar em dois projetos: *Web Apps 1.0* e *Web Forms 2.0*. Juntos estes projetos deram origem ao HTML5 que se tornou o novo padrão HTML. O HTML5 trouxe diversas inovações incluindo novas *tags* e atributos, e, o mais importante, é compatível com todas as outra versões HTML (Craig et al., 2012).

Assim como o HTML, o HTML5 serve para estruturar o conteúdo Web, e suas principais inovações foram os elementos de navegação, menus, sumários, figuras com legenda e alguns elementos interativos. Além disso o HTML5 permitiu tocar áudios e vídeos sem a utilização de um *plug-in*, como o Adobe Flash. Outro ponto interessante do HTML5 é que ele foi desenvolvido a fim de permitir que aplicações como o Google Docs fossem criadas em cima dessa linguagem de programação (Craig et al., 2012).

2.2.1.3 Separando o Conteúdo da Apresentação

No início da década de 1990 os programadores utilizavam apenas a linguagem HTML para apresentar suas páginas Web. Ao utilizarem as *tags* era possível diferenciar títulos e parágrafos, tipo e tamanho das fontes. Porém, com a popularização da Web, designers gráficos começaram a se interessar pela estética dos *websites* e perceberam que o HTML pecava nesse quesito. Assim, em 1996, o W3C apresentou o *Cascading Style Sheets* (CSS), uma linguagem completamente nova e que tinha o objetivo de descrever como os códigos HTML deveriam aparecer na página Web. Tudo isso sem modificar a base dos códigos HTML, já que a estruturação da página foi separada de sua apresentação (Craig et al., 2012).

Ao manter esses dois aspectos da programação separados, foi possível fazer mudanças tanto na estrutura quanto no design de maneira mais fácil e sem que uma coisa impactasse a outra. Contudo, o CSS não foi bem aceito pelos programadores no início devido a alguns *bugs* da linguagem. Hoje porém, a grande maioria dos navegadores suportam os três níveis do CSS (Craig et al., 2012).

2.2.1.4 O CSS3

Como já descrito na seção anterior, a primeira versão do CSS surgiu em 1996 e sua sucessora foi apresentada ao público em 1998 e chamada de CSS2. Posteriormente, o CSS2 sofreu algumas mudanças e tornou-se o CSS 2.1, no entanto, em 2011 não havia um navegador sequer que suportasse tudo que o CSS 2.1 oferecia. Mesmo assim a W3C desenvolveu o CSS3 que era muito mais complicado que seus dois irmãos mais velhos (Craig et al., 2012).

As duas primeiras versões focavam principalmente nos tamanhos das fontes, definições de cores e posicionamento de elementos na página Web. O CSS3, por sua vez, é mais audacioso e promove mudanças de sombras, transições e animações utilizando diferentes módulos para manipular cada atributo. Os módulos são totalmente independentes uns dos outros o que facilita sua manipulação tornando o CSS3 atrativo (Craig et al., 2012).

2.2.2 O Documento HTML e CSS

Apesar da Web ser palco de diversos conteúdos como vídeos, áudios, jogos e diversas *Application Programming Interface* (APIs), a maioria do seu conteúdo é composto por documentos HTML. Uma página Web é basicamente a maneira como o navegador interpreta uma série de textos e remarcações que compõem o documento HTML (Craig et al., 2012).

É possível ter acesso ao código fonte de qualquer *website* na rede apenas utilizando um navegador regular e esta é umas das maneiras mais eficientes de aprender a programação HTML. Entretanto, é preciso ficar atento porque muitos programadores ainda hoje não seguem os padrões definidos pela W3C. Um documento HTML é composto por elementos essenciais e outros que não são essenciais, porém ambos têm grande utilidade (Craig et al., 2012).

2.2.2.1 A Estrutura do Documento HTML

O documento HTML possui os seguintes documentos essenciais (Figura 7): um *doctype* que indica o tipo de declaração do documento; um *root element* que envolve todo o documento; um *head* que é basicamente o título da página Web; e, por fim o *body* que é onde está localizado todo o conteúdo.

De fato, a Figura 7 mostra um código completo e válido em HTML.

Figura 7 – *Estrutura Básica de um Documento HTML*

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Document Title</title>
  </head>
  <body>
    <p>This is a very simple web page.</p>
  </body>
</html>
```

Fonte: Craig et al., 2012.

2.2.2.1.1 O Elemento *Doctype*

Doctype é a abreviação para *document type declaration* e serve para informar ao navegador qual o tipo de documento será processado. Apesar de estar entre os colchetes (< e >), o *doctype* não é considerado uma *tag*, mas sim uma instrução, ou seja, um tipo de comentário e por isso o símbolo "!" no início (Craig et al., 2012).

É possível descrever o *doctype* em letras maiúsculas e minúsculas, e o HTML que vem depois deste indica que o documento foi programado em HTML e não em outra *markup language*. Assim como o *doctype*, o tipo de documento também pode ser escrito tanto em letras maiúsculas e minúsculas, mas deve estar de acordo com o *root element* (que será explicado mais a frente) (Craig et al., 2012).

No início dos anos 2000, quando existia além do HTML, o XHTML, era preciso indicar, além do tipo de documento, todas as regras que este seguia, como é possível observar na Figura 8. Todavia, com o advento do HTML5 isso não é mais necessário, já que esta nova versão é compatível com todas as anteriores. Exceções ocorrem quando se deseja utilizar um característica específica de alguma versão, assim, estas devem ser declaradas no *doctype* (Craig et al., 2012).

Figura 8 – *Declaração das Regras no XHTML*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

Fonte: Craig et al., 2012.

O *doctype* deve aparecer na primeira linha do código fonte, caso contrário, os navegadores irão processar o documento de uma maneira diferente (Craig et al., 2012).

Posteriormente surgiu o *doctype switch* a fim de solucionar um dos problemas já citados neste trabalho. De fato, no início os navegadores não seguiam as especificações do W3C para processar os documentos HTML. Com o surgimento do CSS isso ficou ainda pior porque as especificações para o CSS eram bastante vagas. Consequentemente, *websites* eram processados de maneiras diferentes em cada navegador e muitas vezes sequer funcionavam em navegadores diferentes. Porém, com o desenvolvimento do CSS os *browsers* começaram a ter mais suporte para esta linguagem e começaram a seguir as especificações. Com isso, milhões de páginas Web que haviam sido programadas para seguir as especificações de cada navegador seriam quebradas, mas o *doctype switch* veio para contornar essa situação (Craig et al., 2012).

O *doctype switch* foi a maneira que os navegadores encontraram para distinguir um documento HTML atual de um mais antigo. O que acontece é que se o *doctype* é declarado de maneira clara e na primeira linha do código fonte, então o navegador irá processar este documento como um documento atual. Caso contrário, se há mais especificações antes do *doctype* então o navegador irá considerar este documento como antigo e será mais tolerante ao processá-lo (Craig et al., 2012).

2.2.2.1.2 O *Root Element*: HTML

O HTML, também chamado de *root element*, envolve todo o documento e é a base do que chama-se a árvore do documento. A partir dele derivam-se todas as outras partes, sendo que pode apenas existir um *head* e um *body* dentro deste elemento. Uma curiosidade a respeito do *root element* é que mesmo se o programador esquecer de declará-lo, o próprio navegador irá gerá-lo, já que o *doctype* implica sua existência (Craig et al., 2012).

Contudo, a fim de facilitar o trabalho do navegador, é interessante declarar o *root element* juntamente com uns atributos adicionais como o *lang* e *dir*. Esses atributos servem, respectivamente, para definir a linguagem que será utilizada no documento e o sentido de leitura (no caso do português lê-se da esquerda para a direita). No mundo globalizado tais atributos tornaram-se essenciais já que há diversas nações falando diferentes línguas ao redor do globo (Craig et al., 2012).

2.2.2.1.3 O Elemento *Head*

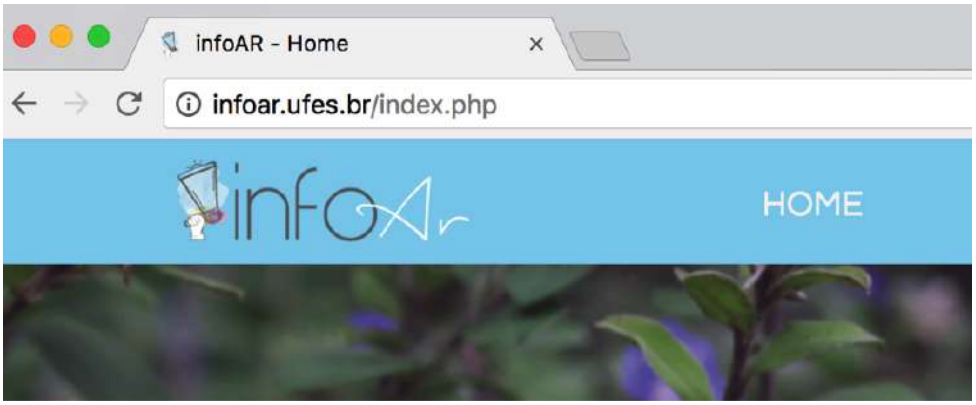
O elemento *head* funciona basicamente como um cabeçalho. Ele não é visível no *website*, porém contém informações importantes para o navegador. É nesse elemento que são declarados os locais onde encontram-se os *scripts* e *style sheets*, fazendo também a conexão com qualquer outro recurso utilizado pelo código-fonte. Ou seja, é recomendável que alguns dos *metadata elements* sejam descritos no *head*, pois não podem ser descritos em nenhuma outra parte do documento (Craig et al., 2012).

Necessariamente o *head* deve aparecer logo após o *root element*, e, assim como este último, pode ser automaticamente gerado pelo navegador, porém as informações sobre a *metadata* serão omitidos (Craig et al., 2012).

2.2.2.1.4 O Elemento *Title*

O *title* é um elemento derivado do *head* e, como o próprio nome já diz, é o título do documento. Este é um elemento obrigatório, porém só pode ser declarado uma única vez. O título aparece tanto nas abas dos navegadores, quanto na janela de títulos ou históricos (Craig et al., 2012). A Figura 9 mostra onde o título aparece no navegador, no *website* (infoar.ufes.br) desenvolvido neste trabalho.

Figura 9 – *Localização dos Título no Navegador Chrome*



Fonte: Produção do próprio autor.

O título é importante porque muitas vezes ele pode aparecer em ferramentas de buscas, como o Google, porém o objetivo principal é informar ao leitor o conteúdo da página que ele está prestes a ler. O elemento *title* também pode mostrar a hierarquia do *website* e permitir que o leitor se localize com facilidade (Craig et al., 2012).

Por fim, este elemento só pode aparecer dentro do elemento *head* e só pode conter elementos de texto, fazendo parte do conteúdo de *metadata* (Craig et al., 2012).

2.2.2.1.5 O Elemento *Meta*

O termo *metadata* significa de fato "dados de outros dados". Este elemento não é visível para os usuários, mas carrega informações importantes, como quando o documento foi publicado, qual software foi utilizado, quem criou e outras informações para facilitar o processamento pelo navegador (Craig et al., 2012).

O elemento *meta* pode incluir vários atributos. Um dos elementos mais utilizados é o *charset* que significa *Character Enconding System*, e auxilia o navegador a lidar com os

caracteres utilizados no documento. De fato, o *charset* mais utilizado é o UTF-8 que significa *UCS Transformation Format, 8-bit*. Esse atributo não é essencial no código fonte porque o servidor já possui um *charset*, porém, se o programador optar por utilizá-lo, ambos devem ser compatíveis (Craig et al., 2012).

Outros atributos são o *name* e o *http-equiv* que informam o nome ou a categoria dos dados. Além desses também há o *content* que atribui um valor ao dado (Craig et al., 2012).

Os sites de busca geralmente utilizam programas chamados de *robots, spiders* ou *crawlers* que varrem praticamente todo o conteúdo na Web e utilizam os elementos *meta* para auxiliá-los na varredura. Um programador pode tanto impedir que os *robots* mostrem o *website* em sites de busca, como também encorajá-los a divulgar a página, tudo isso utilizando os devidos atributos do elemento *meta* (Craig et al., 2012). Ambos os casos podem ser vistos nas Figuras 10 e 11.

Figura 10 – *Atributos Para Não Indexar a Sites de Busca*

```
<meta name="robots" content="noindex, nofollow">
```

Fonte: Craig et al., 2012.

Figura 11 – *Atributos Para Indexar a Sites de Busca*

```
<meta name="robots" content="index, follow">
```

Fonte: Craig et al., 2012.

Alguns programadores utilizam *keywords* para o atributo *name* e várias palavras relacionadas ao conteúdo do *website* para o atributo *content*, com o objetivo de facilitar as buscas. Contudo, essas palavras devem ser poucas e estritamente relacionados ao conteúdo da página Web para não prejudicar o *Search Engine Optimization* (SEO) (Craig et al., 2012).

2.2.2.1.6 O Elemento *Link*

O elemento *link* serve para relacionar o código fonte com outros recursos. Este elemento deve aparecer junto com o elemento *head* e seus principais atributos são o *rel* e o *href*. Tais atributos servem, respectivamente, para indicar o tipo de relação entre o documento e o recurso, e para indicar qual a URL do recurso. Um dos principais valores atribuídos ao *rel* é o *stylesheet*, fazendo referência ao CSS (Craig et al., 2012). A maneira como esses atributos são declarados pode ser visto na Figura 12 .

Figura 12 – *Declaração dos Atributos do Elemento Link*

```
<!DOCTYPE html>
<html lang="en-US" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Power Outfitters Superhero Costume and Supply Co.</title>
    <link rel="stylesheet" type="text/css" href="/css/styles.css">
    <link rel="icon" type="image/ico" href="/favicon.ico">
  </head>
  <body>
    <h1>Welcome, Heroes!</h1>
    <p>Power Outfitters offers top of the line merchandise at
    rock-bottom prices for the discerning costumed crime-fighter.</p>
  </body>
</html>
```

Fonte: Craig et al., 2012.

Outro atributo do elemento *link* é o *type*, o qual indica o tipo de arquivo que está sendo utilizado como recurso. A definição deste atributo facilita o processamento dos recursos pelo navegador, já que arquivos de texto e vídeo, por exemplo, são processados de maneira diferente. Como pode ser visto na Figura 12, o valor do atributo *type* é representado por um *type/subtype*. Para arquivos CSS, por exemplo, o valor atribuído é *text/css*, já para arquivos JavaScript, o valor atribuído é *text/javascript* (Craig et al., 2012).

Quando o recurso utilizado é um arquivo CSS, ou seja, um *stylesheet*, um novo atributo chamado de *media* pode ser utilizado. Tal atributo serve para diferenciar maneiras como a página Web deve ser mostrada em diferentes ocasiões, por exemplo, como o *website* deve ser mostrado na tela de um computador e como ele deve ser impresso (Craig et al., 2012).

No HTML5 é possível dar diversos valores para o atributo *media*, como pode ser visto na Figura 13, e, graças a esse atributo, é possível criar *responsive websites*, ou seja, páginas Web que se ajustam a diferentes dispositivos e navegadores (Craig et al., 2012). A teoria envolvendo *responsive websites* será explicada com mais detalhes posteriormente.

Figura 13 – *Definição do Atributo* Media *Para Diferentes Tamanhos de Tela*

```
<link rel="stylesheet" href="wide.css" media="screen and (min-device-
width: 600px)">
<link rel="stylesheet" href="narrow.css" media="screen and (max-
device-width: 600px)">
```

Fonte: Craig et al., 2012.

2.2.2.1.7 O Elemento *Style*

O elemento *style* é escrito em todo o seu conteúdo em CSS, e serve para definir o *design* da página Web. Na Figura 14 é mostrada a definição do atributo *style*. Por estar dentro de um código escrito em linguagem HTML, tal informação é considerada como um *stylesheet* interno. Essa configuração só é interessante para *websites* que possuem uma única página, pois quando há várias páginas compartilhando o mesmo *design*, é mais vantajoso criar um *stylesheets* externo e utilizar tal recurso através do elemento *link* (Craig et al., 2012).

Figura 14 – *Definição do Elemento* Style

```
<style type="text/css">
  body { background-color: ivory; color: navy; }
  h1 { font-size: 1.6em; color: crimson; }
</style>
```

Fonte: Craig et al., 2012.

Antes do HTML5, o elemento *style* só poderia ser declarado dentro do elemento *head*, contudo, essa nova versão do HTML trouxe um atributo chamado *scope* que permite definir o *style* em qualquer parte do elemento *body*, modificando o *design* dos elementos vizinhos. Todavia, não é recomendável a utilização do atributo *scope*, pois este mistura estrutura e apresentação, o que, como já exlicado, não é vantajoso. A solução para este problema é a utilização do atributo *class* para modificar o *design* de partes mais específicas do *website* utilizando o *stylesheet* (Craig et al., 2012).

2.2.2.1.8 O Elemento *Script* e o JavaScript

O elemento *script*, assim como o elemento *style*, deve ser totalmente escrito em uma *scripting language* (na maioria das vezes JavaScript), ou deve "linkar" o código fonte a um *script* cuja URL é indicada pelo atributo *src*. Os *scripts* servem, principalmente, para adicionar animações e outros elementos interativos (Craig et al., 2012). Na Figura 15 é possível ver como o elemento *script* é definido dentro do texto.

Como no caso do *style*, não é recomendável criar um código em JavaScript dentro do código HTML, e sim fazer um *link* utilizando o atributo *src*. O valor desse atributo será uma URL com extensão .js (Craig et al., 2012). Na Figura 16 é possível ver como se utiliza o atributo *src*.

Figura 15 – *Definição do Elemento* Script *Dentro do Código HTML*

```
<script>
  window.onload = function() {
    alert("Excelsior!");
  }
</script>
```

Fonte: Craig et al., 2012.

Figura 16 – *Declaração do Atributo* scr

```
<script src="/scripts/excelsior.js"></script>
```

Fonte: Craig et al., 2012.

Quando se utiliza o atributo *src*, o elemento *script* deve estar vazio, ou seja, sem nenhum outro código em Javascript, caso contrário o navegador irá ignorar o *script* interno ao código. Quanto ao posicionamento, o elemento *script* geralmente se encontra dentro do elemento *head* (o que é recomendável), mas também pode estar localizado dentro do elemento *body*. Além disso, é possível que haja mais de um *script* em um mesmo código HTML, porém um grande número desse elemento não é recomendável, devido à dificuldade de processamento (Craig et al., 2012).

Para entender melhor o elemento *script*, é preciso compreender também como as *script languages* funcionam. Em específico, será descrito o JavaScript, o qual, diferentemente de outras linguagens, não executa seus próprios comandos, mas apenas dá instruções ao navegador. Suas duas funções são: manipular dinamicamente documentos HTML e facilitar a comunicação *browser*-servidor através da manipulação de *cookies*. Apesar de manipular os mesmos elementos que o CSS, o JavaScript é uma linguagem complicada e que praticamente não existe sem a linguagem HTML (Craig et al., 2012).

2.2.2.1.9 O Elemento *Body*

O último elemento a ser descrito nesta seção é o *body* que tem como objetivo separar o conteúdo visível pelos usuários do conteúdo de *metadata*, servindo assim como um contenedor (contêiner). Assim como o elemento *head*, é possível omitir as *tags* que marcam o início e o fim desse elemento, mas como nos outros casos, isso não é recomendável. Não há nenhum atributo essencial para esse elemento, porém é muito comum utilizar os

atributos *class* para identificar uma página específica, e *home* para indicar qual a página inicial (Craig et al., 2012).

Ao utilizar atributos dentro do elemento *body*, é possível relacioná-los com o *stylesheet*. Dessa maneira, independentemente de onde o atributo aparecer dentro do *body*, ele terá as características específicas definidas pelo código CSS (Craig et al., 2012). Na Figura 17 é possível verificar qual a fonte definida para os elementos h1, por exemplo.

Figura 17 – Stylesheet *Para Definir Alguns Atributos Dentro do* Body

```
h1 {
  font-size: 20px;
}

.landing h1 {
  font-size: 26px;
}

#home h1 {
  font-size: 34px;
}
```

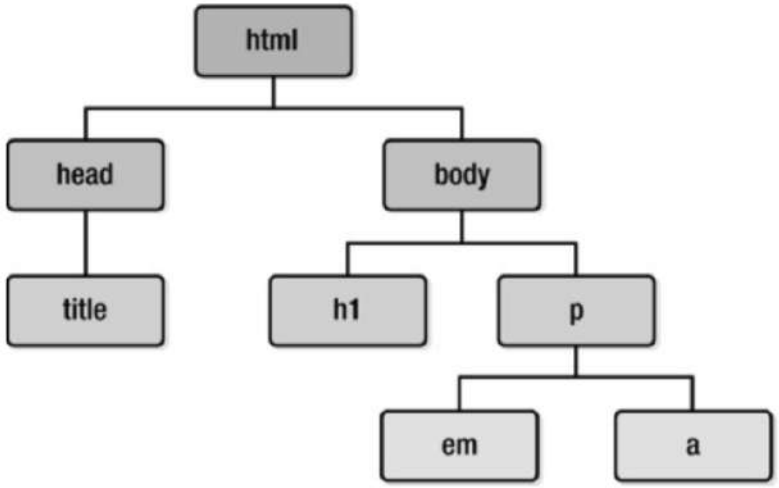
Fonte: Craig et al., 2012.

2.2.2.2 Diagrama de Árvore do Código HTML

Para facilitar a visualização da estrutura de um documento HTML, é possível utilizar um diagrama de árvore como o mostrado na Figura 18. A partir do *root element* derivam-se todos os demais elementos, como o *head* e o *body*. Em suma, no *head* encontram-se o conteúdo de *metadata* e o título, mas apenas este último é visível ao leitor. No elemento *body* encontra-se todo o conteúdo visível ao leitor (craig et al., 2012).

Um *website* é, de fato, um conjunto de páginas Web que seguem a mesma estrutura da Figura 18 e contendo os elementos citados na seção anterior. Todos os arquivos devem estar dentro de uma mesma pasta, e a forma de organização das pastas depende de como a aplicação necessita dos recursos para funcionar corretamente (CAELUM, 2016).

Figura 18 – *Diagrama de Árvore de um Código Simples*



Fonte: Craig et al., 2012.

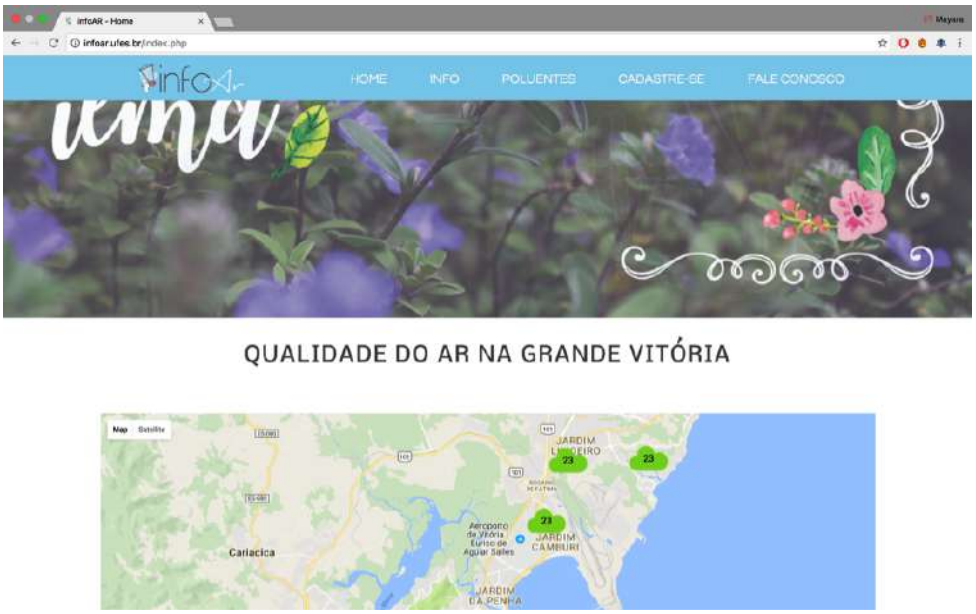
2.2.2.3 *Responsive Website*

Dados do Instituto Brasileiro de Geografia e Estatísticas (IBGE) apontam que, pela primeira vez, o número de usuários que utilizam o celular para acessar a Internet é maior do que o número de usuários que utilizam o computador para o mesmo fim (IBGE, 2016). Assim, a necessidade de desenvolver *responsive websites*, ou seja, *websites* que se adaptam a diversos dispositivos, se tornou evidente, o que é um dos objetivos deste trabalho.

Quando um *website* não é *responsive*, o usuário terá acesso a uma versão minimizada do que é mostrado nas telas dos computadores, já que o *browser* irá processar o documento HTML da mesma maneira. Contudo, essa é uma tarefa tediosa, já que o leitor deverá ficar ajustando o *zoom* a fim de ler o conteúdo desejado. Assim, a tendência de mercado, hoje, é possuir uma versão diferente dependendo do tamanho da tela do dispositivo, a fim de facilitar a navegação. As figuras 19 e 20 mostram como um *responsive website* se adequa a diversos formatos: tela do computador e tela do celular, respectivamente.

Os dois principais componentes para se construir um *responsive website* são o *viewport meta tag* e *media queries*.

Figura 19 – Layout *do* Website *no Computador*



Fonte: Produção do próprio autor.

Figura 20 – Layout *do* Website *no Celular*

Fonte: Produção do próprio autor.

2.2.2.3.1 *Viewport Meta Tag*

Esse foi o primeiro mecanismo criado para solucionar o problema do *zoom* nas páginas Web. Sua definição pode ser vista na Figura 21. O que ocorre é que essa *viewport meta*

tag impede que o conteúdo do *website* seja escalado (Firdaus et al., 2016). A Figura 22 mostra, à esquerda, um *website* no qual a *viewport meta tag* foi utilizado, e, à direita uma página Web regular.

Figura 21 – *Definição do* Viewport Meta Tag

```
<meta name="viewport" content="width=device-width,
initial-scale=1">
```

Viewport meta tag

Fonte: Firdaus et al., 2012.

Figura 22 – *Comparação:* Viewport Meta Tag *vs.* Website *Convencional*

Fonte: Firdaus et al., 2012.

Em termos técnicos, o *viewport*, que, diferentemente da tela do dispositivo, é apenas a área que o navegador utiliza para mostrar o *website*, é definido com o tamanho da tela do dispositivo. Ao definir uma escala de um para um, o conteúdo não terá sua dimensão diminuída em dipostivos menores, como ocorria em *websites* sem essa característica (Firdaus et al., 2016).

2.2.2.3.2 *Media Queries*

Como já citado anteriormente, um dos atributos do elemento *link* é o *media*. Em conjunto com o CSS3, é possível, utilizando esse atributo, definir características específicas de acordo

com a largura do *viewport* do dispositivo (Firdaus et al., 2016). Na Figura 23 é possível ver um código em CSS3 em que a fonte do parágrafo é modificada quando o *viewport* fica menor do que 480 pixels.

Figura 23 – Media Queries *no Código CSS3*

```
p {
font-size: 16px;
}
@media screen and (max-width: 480px) {
p {
    font-size: 14px;
}
}
```

Fonte: Firdaus et al., 2012.

2.2.2.3.3 *Responsive Frameworks*

Os *responsive frameworks* são a base para se construir *responsive websites*. Eles contam com uma série de classes, *layouts*, botões e barras de navegação. Com o *layout* pré-definido, é possível desenvolver um *responsive website* mais rápido (Firdaus et al., 2016).

As vantagens de se utilizar um *framework* são diversas, incluindo compatibilidade com diversos *browsers*, diminuindo, assim, o trabalho com programação, e também possuem extensa documentação de fácil acesso, o que pode auxiliar novos desenvolvedores. Contudo, os códigos dos *frameworks* são bem complexos, já que precisam incluir diversos cenários nos quais é possível desenvolver um *layout*, cabendo ao programador definir qual cenário será utilizado (Firdaus et al., 2016).

Neste trabalho foi utilizado um *framework* popular chamada de *Foundation*, o qual foi criado por uma agência de *design* da Califórnia chamada de ZURB. Esse *framework*, além de utilizar as bases do CSS, também utiliza vários *plug-ins* JQuery para adicionar interatividade à página Web. Muitas empresas, como a McAfee, desenvolveram seus *websites* utilizando esse *framework*, demonstrando que esse é um *framework* confiável (Firdaus et al., 2016).

2.2.2.4 A Linguagem PHP

A Internet é um ambiente interativo que utiliza formulários para coletar informações em *websites* dinâmicos. Essas informações podem ser utilizadas para desenvolver outras páginas Web ou são armazenas em bancos de dados. A linguagem HTML não provê as funcionalidades necessárias para que uma página Web dinâmica funcione corretamente, para isso é preciso utilizar uma *scripting language* chamada de PHP (Valade, 2006).

O PHP é uma linguagem que possui uma sintaxe parecida com a Linguagem C, porém, suas funcionalidades são voltadas ao desenvolvimento de *websites* dinâmicos. Por ser um *software* livre, é compatível com a maioria dos sistemas operacionais, além disso possui funções próprias para manipular o banco de dados MySQL (Valade, 2006).

O código PHP pode tanto estar inserido dentro de um documento HTML ou ser desenvolvido em um novo arquivo que será "linkado" ao documento HTML. Em ambos os casos, porém, o código PHP deve ser descrito entre as *tags* <?php e ?>, e, para que funcione corretamente, deve trabalhar em conjunto com o servidor do *website*. O servidor, ao verificar a existência de um código PHP, envia o mesmo para o *software* PHP que fará o processamento e gerará uma saída. Assim, o servidor manda para o navegador o código HTML e a saída PHP gerada. O navegador, por sua vez, mostra o *website* ao usuário (Valade, 2006).

Como os códigos PHP e HTML podem estar descritos em um mesmo arquivo, é aconselhável utilizar a extensão *.php*, ao invés da extensão *.html*, nos arquivos que serão armazenados no servidor. A Figura 24 mostra como os códigos PHP devem ser definidos.

Figura 24 – *Definição dos Códigos PHP*

Fonte: Valade, 2006.

Quando um código PHP está integrado ao documento HTML, é possível ter várias seções de código PHP seguidos por código HTML e vice-versa. Todavia, todo o código PHP deverá estar entre as *tags* que marcam seu início e fim, caso contrário esse não será processado pelo servidor. O programador também deve estar atento, pois os valores de

variáveis definidos em uma seção PHP serão mantidos na próxima seção (Valade, 2006).

É sabido que um código PHP é composto por várias declarações que instruem o *software* PHP a realizar uma ação, e a ordem de execução das instruções é de cima para baixo, de forma sequencial (Valade, 2006).

As declarações PHP podem ser simples ou complexas. As declarações simples são aquelas que contêm apenas uma instrução PHP, e o seu final é marcado pelo caractere ";". O *software* PHP irá ler todas as instruções até encontrar a *tag* que marca o final da seção (?>). Já as declarações complexas são aquelas que contêm mais de uma instrução, como é o caso das instruções *if* e *while*. Além disso, as declarações simples podem ser combinadas para gerar um bloco, assim, todas as declarações serão executadas simultaneamente. Essa funcionalidade é importante para facilitar a programação de declarações complexas (Valade, 2006).

Para as saídas também há dois tipos possíveis: *echo* e *print*. A sintaxe de ambos pode ser vista na Figura 25. A diferença entre os dois é que o *echo* pode imprimir várias *strings* de uma só vez, enquanto o *print* funciona como uma função e só pode imprimir uma *string* por vez. A maneira como o *browser* irá interpretar essas saídas irá depender se elas possuirão ou não HTML *tags* (Valade, 2006).

Figura 25 – *Sintaxe das saídas* Echo *e* Print

```
echo outputitem1, outputitem2, outputitem3
...;

$var = print outputitem;
```

Fonte: Valade, 2006.

Além das declarações e saídas já citadas anteriormente, a linguagem PHP provê muitas outras funcionalidades, como a manipulação de vetores e a criação de funções, porém, como a programação é bem parecida com a da Linguagem C, esses detalhes não serão abordados nesta seção. Para mais detalhes da programação é possível analisar os códigos desenvolvidos nos Anexos.

2.2.3 O Banco de Dados MySQL

O banco de dados, também chamado de *database*, se refere ao grupo de arquivos que contém as informações do *website*. As informações contidas no banco de dados podem ser

mostradas na página Web ou apenas armazenadas. O MySQL, muito utilizado em *websites* dinâmicos, é considerado um *Relational Database Management System* (RDBMS), ou seja, um sistema que organiza os dados em tabelas. A Microsoft oferece uma versão gratuita do MySQL e esse *software* é compatível com a maioria dos sistemas operacionais (Valade, 2006).

Além do seu servidor, o MySQL conta com diversos programas e softwares que, em conjunto, são capazes de lidar com qualquer manipulação que se deseja fazer no banco de dados. As operações no banco de dados são feitas interpretando-se instruções em uma fila, e a linguagem utilizada é a *Structured Query Language* (SQL) (Valade, 2006).

A linguagem SQL é baseada no idioma inglês e é relativamente fácil de aprender quando comparada com outras linguagens de programação. Cada instrução se inicia com uma palavra, também chamada de *query*, que indica a ação a ser realizada. As principas *queries* são CREATE, SELECT, DELETE, INSERT, UPDATE e ALTER (Valade, 2006).

A estrutura do MySQL, como já citado, é baseado em tabelas. As tabelas têm como foco um objeto, também chamado de entidade, que ocupa as linhas. Já as colunas, por sua vez, representam os atributos das entidades. É possível que diferentes tabelas se relacionem entre si, e isso será explicado a seguir (Valade, 2006).

2.2.3.0.1 Modelo Entidade e Relacionamento e Diagrama Entidade e Relacionamento

O Modelo Entidade Relacionamento (MER) é uma ferramenta da Engenharia de Software utilizada para caracterizar as entidades e seus atributos, além de relacioná-los entre si. Esse modelo conceitual ajuda a modelar a estrutura que um banco de dados deve possuir. A maneira de construir esse modelo será explicado adiante (Rodrigues, 2016).

Primeiro é preciso definir as entidades, que podem ser físicas ou lógicas. As entidades físicas são aquelas que existem no mundo real, como um cliente. Já as entidades lógicas surgem da interação entre entidades físicas e não são tangíveis. Para facilitar o modelo, as entidades devem receber nomes que remetem às suas funções, como cliente ou produto (Rodrigues, 2016).

De acordo com o motivo da sua existência, as entidades podem ser classificadas como fortes, fracas e associativas. As entidades fortes são aquelas que não dependem de nenhuma outra entidade para existir. Já as entidades fracas dependem de uma outra entidade para existir, pois, por si só, sua existência não faz sentido. E as entidades associativas surgem

quando há a necessidade de relacionar uma entidade com um relacionamento. No MER não é permitido que uma entidade seja associada a um relacionamento diretamente, assim, um relacionamento vira uma entidade associativa para se associar a outras entidades (Rodrigues, 2016).

Após definir as entidades, é preciso definir o relacionamento entre elas que pode ser de "um para um", de "um para muitos" e de "muitos para muitos". No relacionamento "um para um", uma entidade só pode se relacionar com apenas uma única outra entidade. Por exemplo, em uma universidade, cada aluno só pode ter um número de matrícula e vice-versa. Já no relacionamento "um para muitos" uma entidade pode se relacionar com várias outras entidades, porém, as entidades relacionadas só podem se relacionar com a entidade principal. Um exemplo ocorre quando se têm várias estações de monitoramento que podem ter vários usuários cadastrados, mas o usuário só pode estar cadastrado em uma estação. E, por fim, no relacionamento "muitos para muitos" uma entidade pode se relacionar com várias entidades, da mesma maneira que uma entidade já relacionada pode se relacionar com outras entidades (Rodrigues, 2016).

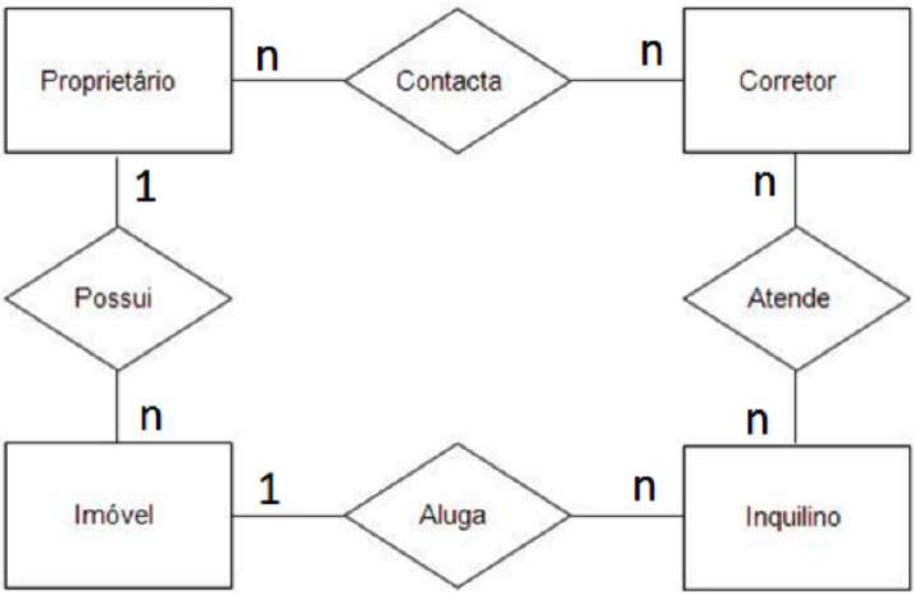
Os atributos, que caracterizam as entidades, também podem ser classificados de acordo com sua função. Os atributos descritivos fazem relação a uma característica particular da entidade, como o nome. Já os atributos nominativos tem como função identificar a entidade. E os atributos referenciais indicam a relação de uma entidade com outra (Rodrigues, 2016).

Além de serem classificados quanto à função, os atributos também podem ser classificados quanto à estrutura. Os atributos simples são aqueles que definem apenas uma característica da entidade, e os atributos compostos são aqueles que requerem várias características para descrever uma entidade (Rodrigues, 2016).

Após caracterizar o cenário é possível desenvolver o MER, que, em sua forma gráfica é chamada de Diagrama de Entidade e Relacionamento (DER). A regra para a construção do diagrama é simples, as entidades devem ser apresentadas em retângulos, seus atributos em elipses e seus relacionamentos em losangos. Os losangos são ligados as entidades por linhas que devem indicar também o tipo de relacionamento ("um para um", "um para muitos"ou "muitos para muitos") (Rodrigues, 2016). Na Figura 26 é possível analisar um DER simples.

Para auxiliar no desenvolvimento do DER há na Internet *softwares* chamados de *Computer-Aided Software Engineering* (CASE). A ferramenta CASE permite a criação de diagramas de forma simples e em um ambiente de fácil utilização. À partir desse diagrama a administração de um banco de dados se torna menos trabalhosa.

Figura 26 – *Exemplo simples de um DER*



Fonte: Rodrigues, 2016.

2.3 Desenvolvimento de Aplicativos para Celular

2.3.1 Aplicativo para o Sistema Operacional iOS

A fim de desenvolver um aplicativo iOS, é necessário ter um computador produzido pela Apple, também chamado de *Macbook*. Assim, é possível adquirir o *software* Xcode que é gratuito e disponível para desenvolvedores da Apple, sendo que a linguagem utilizada é a *Start Developing iOS Apps (Swift)*. O Xcode já vem configurado para que seja possível emular o aplicativo tanto em um computador quanto em um iPhone. Contudo, o fato de o aplicativo funcionar no dispositivo não quer dizer que ele está pronto para ser divulgado na App Store. Os aplicativos precisam ter uma série de customizações para garantir a melhor experiência possível ao usuário (Apple, 2016).

Para serem mostrados de maneira correta nos dispositivos iOS, os aplicativos devem utilizar uma série de recursos e *metadata* pré-determinados. Um desse recursos é o *info.plist* que contém toda a *matedata* do aplicativo e é gerado automaticamente pelo Xcode de acordo com as configurações definidas pelo programador. As informações de *metadata* são utilizadas pelo sistema operacional para que este possa interagir com o aplicativo corretamente, além disso, é possível modificar o conteúdo do *info.plist*, se necessário (Apple, 2016).

O programador também deve declarar qual o *hardware* necessário para rodar o aplicativo. Com essas informações a App Store pode determinar se um determinado dispositivo é capaz, ou não, de suportar o aplicativo. Além disso, é preciso especificar vários tamanhos de ícones que serão utilizados em diferentes situações, como na *home screen* do aparelho, nas ferramentas, ou quando o usuário for realizar uma busca na App Store. Outro item necessário é a *launch image*, uma imagem que irá aparecer enquanto o sistema operacional carrega o conteúdo do aplicativo (Apple, 2016).

A segurança do usuário também deve ser uma prioridade para o programador, já que a maioria dos dispositivos armazenam dados pessoais que não devem ser acessados por aplicativos ou usuários externos. Assim, é importante sempre pedir autorização do usuário para acessar dados e ser transparente quanto ao uso dos mesmos (Apple, 2016).

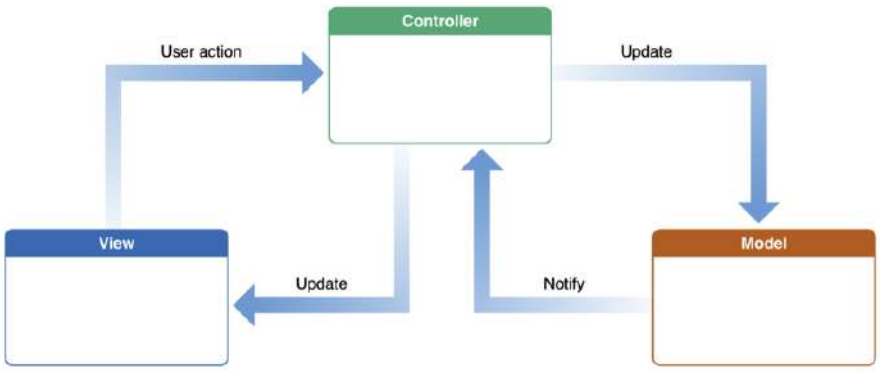
O sistema operacional iOS oferece *frameworks* que são a base que um aplicativo necessita para rodar. É função do programador, porém, fornecer os códigos que irão customizar essa base e gerar o aplicativo com as funcionalidades desejadas. Com isso, é importante compreender um pouco mais dos *frameworks* iOS que são baseados em padrões, como o *Model View Controller* (MVC) (Apple, 2016).

No padrão MVC os objetos podem ter três diferentes funções: *model, view* e *controller*, e, além de determinar a função de cada objeto, esse padrão também determina a maneira

como esses objetos se comunicam. Ao conjunto de objetos com a mesma função dá-se o nome de *layer*, sendo que os *layers* são separados uns dos outros por fronteiras.

As vantagens do padrão MVC são diversas, por exemplo, a interface do aplicativo será melhor definida e será mais fácil produzir novas versões. Além disso, o padrão MVC é o elemento central dos chamados aplicativos *Cocoa*, que são baseados em uma API orientada a objetos nativa do sistema operacional iOS (Apple, 2016). Na Figura 27 é possível verificar como o padrão MVC funciona.

Figura 27 – *Funcionamento do padrão MVC*



Fonte: Apple, 2016.

Outro elemento muito importante no desenvolvimento de aplicativos iOS é a função *Main*. Essa função é a porta de entrada de qualquer aplicativo, e, no caso do iOS, essa função é gerada automaticamente pelo Xcode. Salvo raras exceções essa função nunca deve ser modificada. O padrão adotado pode ser visto na Figura 28.

Figura 28 – *Função* Main *Padrão*

```
#import <UIKit/UIKit.h>
#import "AppDelegate.h"

int main(int argc, char * argv[])
{
    @autoreleasepool {
        return UIApplicationMain(argc, argv, nil, NSStringFromClass([AppDelegate class]));
    }
}
```

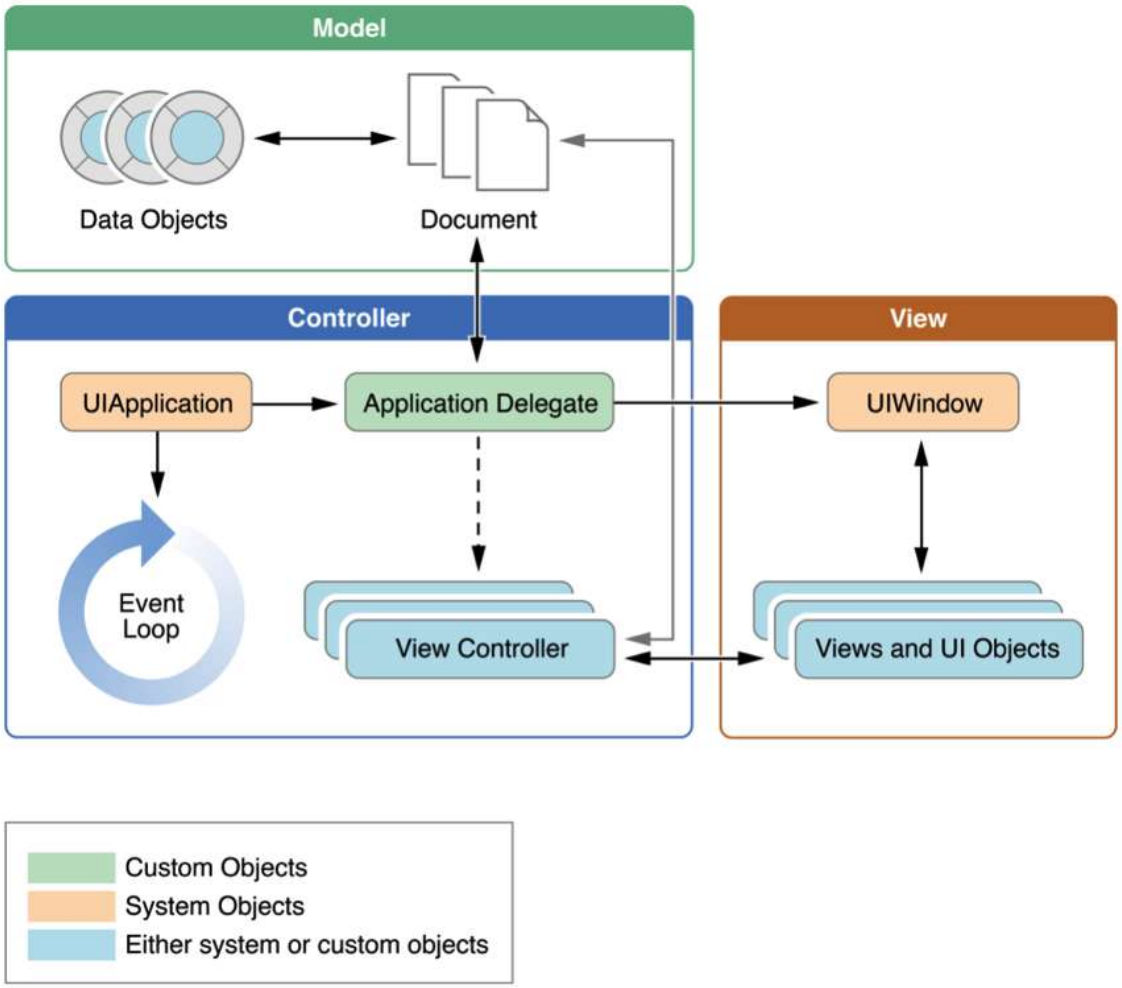
Fonte: Apple, 2016.

O objetivo principal da função *main* é administrar o *UIKit framework* através da função *UIApplicationMain*. Essa função irá criar os objetos do aplicativo, além de carregar a interface disponível no *storyboard* e chamar o código desenvolvido pelo programador.

2.3.1.1 A Estrutura do Aplicativo

A função *UIApplicationMain* gera vários objetos-chave (Figura 29) e também é responsável por começar a rodar o aplicativo. Um dos objetos criados é o *UIApplication*, que é responsável pela interação do sistema com outros objetos. Como a inicialização do aplicativo é sempre igual, o que difere um aplicativo do outro é a maneira como os dados são manipulados e mostrados ao usuário (Apple, 2016). A Tabela 2 explica as funções dos objetos-chave criados.

Figura 29 – *Objetos-chave de um Aplicativo iOS*



Fonte: Apple, 2016.

Tabela 2 – Função dos Objetos em um Aplicativo iOS

Objeto	Descrição

Objeto *UIApplication*	O *UIApplication* administra os eventos de *loop* e outros eventos de alto-nível. Além disso é responsável por relatar as transições e alguns eventos especiais, como a notificação.
Objeto *AppDelegate*	É o coração do código e está presente em todos os aplicativos. Junto com o *UIApplication*, é responsável pela inicialização do aplicativo, lidando também com as transições e alguns eventos especiais.
Documentos e Objetos *Data Model*	Os objetos de *data model* são responsáveis por armazenar todo o conteúdo do aplicativo, sendo, assim, exclusivos de cada aplicativo. É possível utilizar os documentos para organizar os arquivos de *data model*. Grupos são criados a fim de que os dados de um mesmo arquivo continuem juntos.
Objetos *View Controller*	É responsável pelo conteúdo que será carregado na tela do dispositivo. Controla todas as *views* e *subviews* de modo que estas são instaladas na janela do aplicativo
Objeto *UIWindow*	Coordena a apresentação das *views* nas janelas dos aplicativos. Geralmente um aplicativo tem apenas umas *UIWindow* que apresenta o conteúdo do aplicativo na tela principal. Contudo, outras *UIWindow* podem ser adicionadas se o conteúdo for mostrado em uma tela externa

Objetos de *View*, Objetos de Controle e Objetos de *Layers*	Objetos de *view* são responsáveis por arrastar conteúdos para uma área específica a fim de esse conteúdo seja mostrado na tela do aplicativo. Já os objetos de controle são responsáveis por implementar objetos como botões e áreas de texto. E os objetos de *layers* são objetos de dados e que representam o conteúdo visual.

Fonte: Apple, 2016.

2.3.1.2 O *Loop* Principal

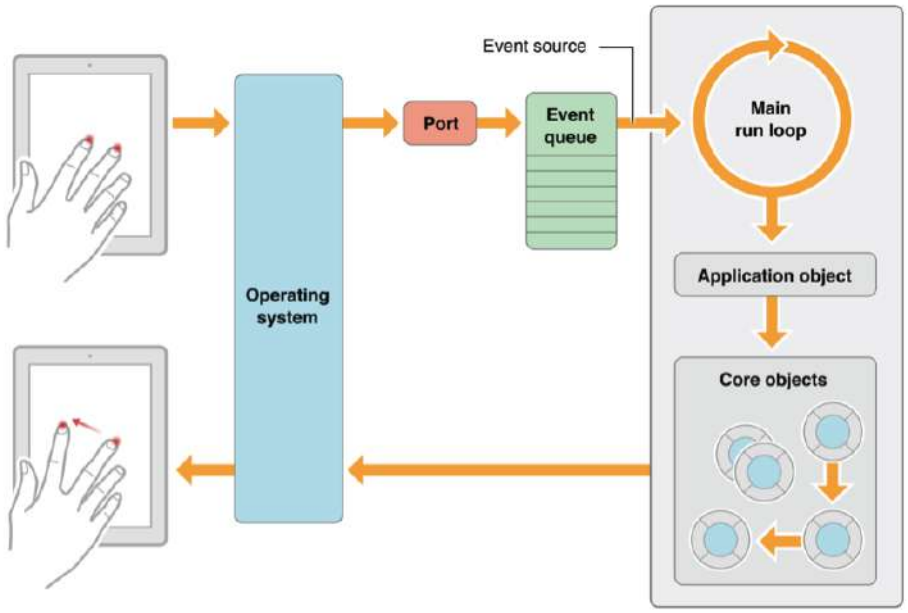
O *loop* principal é responsável por processar todos os eventos gerados pelo usuário, e também deve garantir que esses eventos sejam processados na ordem em que foram recebidos (Figura 30). Há vários tipos de eventos que podem ocorrer em um aplicativo iOS, incluindo o *touch*. A maioria desses eventos são recebidos através do *loop* principal, mas alguns precisam ser enviados a objetos. Por exemplo, os eventos de *touch* e de controle remoto devem ser enviados a *responder objects*, a fim de serem processados. Geralmente os eventos são enviados a *responder objects* específicos, porém, se esse objeto não for capaz de fazer o processamento, ele pode enviar o evento para um outro *responder object* através da *responder chain* (Apple, 2016).

2.3.1.3 Estados de Execução do Aplicativo

O aplicativo iOS só pode adquirir cinco diferentes estados de execução: "inativo", "ativo", "no *background*", "não está rodando"e "suspenso". O aplicativo muda de um estado para o outro dependendo das ações que ocorrem dentro do sistema (Apple, 2016).

Um aplicativo não está rodando quando ele ainda não foi "lançado" ou quando ele foi fechado pelo sistema operacional. Já o estado inativo ocorre quando o aplicativo está em primeiro plano, porém esse não recebe eventos. Isso acontece geralmente nos momentos de transição. Por outro lado, o aplicativo estará ativo se estiver em primeiro plano e recebendo eventos, e, quando o aplicativo é fechado, ele passa rapidamente pelo estado

Figura 30 – *Processamento de eventos no* Loop *Principal*



Fonte: Apple, 2016.

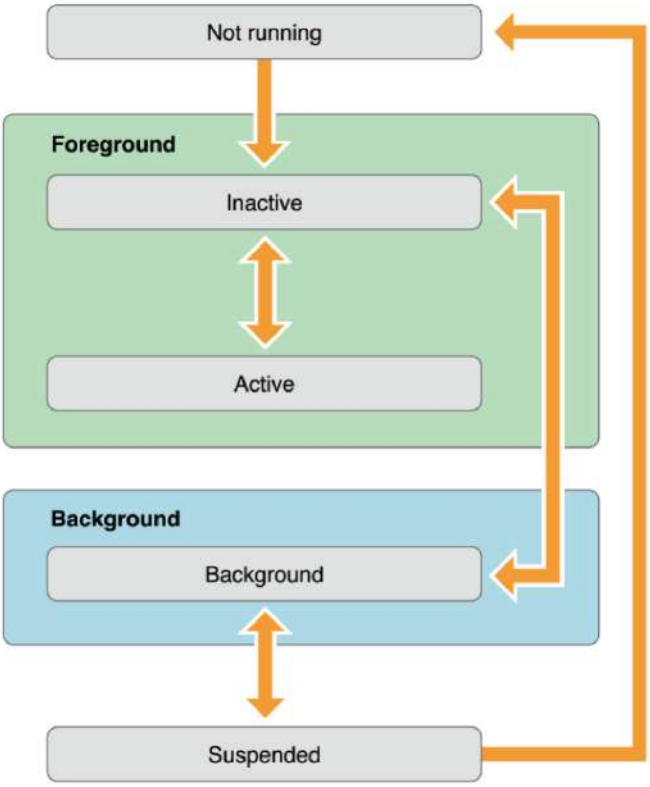
de *background* em que, apesar de estar no *background* o aplicativo continua executando o código, e depois vai para o estado de suspensão. No estado suspenso, o aplicativo se encontra no *background*, porém não executa nenhum código. As mudanças de estado estão descritas na Figura 31.

O aplicativo deve estar preparado para ser fechado a qualquer momento e não deve deixar nenhuma tarefa crítica para ser realizada nesse instante. É muito comum que o sistema operacional feche o *app* forçadamente, seja para abrir memória para outros aplicativos ou mesmo quando o aplicativo deixa de responder aos eventos corretamente. O programador deve estar ciente de que, no momento em que o aplicativo é fechado, nenhuma notificação é enviada, e o processo é apenas "morto".

2.3.1.4 A Linguagem Swift

A linguagem de programação *Swift* é *open source*, ou seja, gratuita, e sua documentação está disponível em Swift.org. Além de colaboradores dentro da Apple, o desenvolvimento dessa linguagem contou, e ainda conta, com colaboradores externos que buscam principalmente tornar o *software Swift* mais rápido e seguro. Atualmente o *Swift* é suportado pelas plataformas Apple e Linux (Apple, 2016).

Figura 31 – *Mudança de Estados do Aplicativo iOS*



Fonte: Apple, 2016.

No quesito transição de linguagens ou versões, o *Swift* é bem flexível e o Xcode já fornece todas as ferramentas para fazer a transição de um código implementado em uma versão mais antiga para uma versão mais nova. Além disso, oferece suporte para os *frameworks* mais utilizados, e pode coexistir, em um mesmo projeto, com códigos em *Objective-C*, a fim de facilitar a transição (Apple, 2016).

Como resultado de uma longa pesquisa em linguagens de programação, o *Swift*, quando comparado com o *Objective-C*, é uma linguagem mais limpa (Figura 32), assim, o trabalho de ler um código e mantê-lo é facilitado. Além disso, a memória é administrada automaticamente, facilitando o trabalho do desenvolvedor (Apple, 2016).

Desde a sua criação, o *Swift* tinha como objetivo ser uma linguagem de programação rápida, para tanto, utiliza um compilador de alta performance chamado de LLVM, que transforma o código *Swift* em um código nativo a fim de utilizar ao máximo o *hardware* disponível. A sintaxe, baseada no inglês, também foi otimizada para que as instruções fossem descritas da maneira mais óbvia possível (Apple, 2016).

A linguagem *Swift* é a sucessora das linguagens C e *Objective-C*, e, assim, também oferece

Figura 32 – *Código* Swift *Simples*

```
extension String {
    var banana : String {
        let shortName = String(characters.dropFirst(1))
        return "\(self) \(self) Bo B\(shortName) Banana Fana Fo F\(shortName)"
    }
}

let bananaName = "Jimmy".banana     // "Jimmy Jimmy Bo Bimmy Banana Fana Fo Fimmy"
```

Fonte: Apple, 2016.

primitivas de baixo nível como operadores e controle de fluxo. Também pode ser orientada a objetos, fornecendo elementos como as classes, sendo assim compatível com a API *Cocoa* (Apple, 2016).

Por questões de segurança o *Swift* eliminou algumas classes que continham códigos inseguros, além disso checa automaticamente se vetores estão sofrendo com *overflow* e, como já citado, administra a memória automaticamente. Outra característica interessante é que esse linguagem não permite que um objeto seja declarado *nil*, ou seja, nulo, prevenindo assim que o aplicativo sofra com problemas devido ao longo tempo de compilação do código (Apple, 2016).

2.3.1.4.1 O *JavaScript Object Notation (JSON)*

Quando o aplicativo precisa se comunicar com uma aplicação Web, as informações vindas do servidor geralmente são disponibilizadas no formato *JSON*. Para converter *JSON* para *Swift*, é possível utilizar *frameworks*, como o Foundation, que converte os dados para alguns tipos específicos, como vetores ou *strings*. Contudo, nem sempre o tipo de dado recebido é conhecido, e essa conversão pode se tornar trabalhosa (Apple, 2016).

Ao utilizar o Foundation *framework*, é necessário utilizar o *JSONSerialization*, o qual utiliza a classe *jsonObject(with:options)* (Figura 33), retornando um objeto do tipo *Any*. Se houver algum problema de recebimento do arquivo, essa classe irá retornar um erro. Nas opções da classe é possível utilizar operadores como o *if* para extrair uma informação com valor já conhecido em uma constante, por exemplo, para extrair um vetor, o valor da opção deve ser *Any*. Essa situação pode ser vista na Figura 34.

Figura 33 – *Extraindo valores JSON com Foundation* framework

```
import Foundation

let data: Data // received from a network request, for example
let json = try? JSONSerialization.jsonObject(with: data, options: [])
```

Fonte: Apple, 2016.

Figura 34 – *Extração com Foundation de um Objeto do Tipo Vetor*

```
if let array = jsonWithArrayRoot as? [Any] {
    if let firstObject = array.first {
        // access individual object in array
    }

    for object in array {
        // access all objects in array
    }

    for case let string as String in array {
        // access only string values in array
    }
}
```

Fonte: Apple, 2016.

Outra maneira de extrair dados *JSON* em *Swift* é utilizando o modelo MVC e convertendo os dados em objetos específicos do domínio do aplicativo, baseado no *layer* de modelos (Apple, 2016).

Isso acontece quando se tem um objeto já definido no aplicativo e que já tenha seus atributos definidos. No caso, o objeto é um Restaurante e seus atributos são: localização, definida por latitude e longitude, nome e refeições. Nesse caso, é preciso programar um inicializador que irá receber as informações do *JSON* em um objeto do tipo *Any*, e esse mesmo inicializador irá transformar essa informação nos atributos desejados (Figura 35). Contudo, se o aplicativo se comunicar com mais de um servidor Web, as informações *JSON* podem ser representadas de maneiras diferentes, neste caso, é necessário um inicializador para cada tipo de representação possível (Apple, 2016).

Também é possível converter automaticamente um modelo *JSON* em *Swift* utilizando uma ferramenta disponível do *Swift* chamada de *Mirror*. Todavia, uma das desvantagens

Figura 35 – *Inicializador para converter dados* JSON *em* Swift

```
extension Restaurant {
    init?(json: [String: Any]) {
        guard let name = json["name"] as? String,
            let coordinatesJSON = json["coordinates"] as? [String: Double],
            let latitude = coordinatesJSON["lat"],
            let longitude = coordinatesJSON["lng"],
            let mealsJSON = json["meals"] as? [String]
        else {
            return nil
        }

        var meals: Set<Meal> = []
        for string in mealsJSON {
            guard let meal = Meal(rawValue: string) else {
                return nil
            }

            meals.insert(meal)
        }

        self.name = name
        self.coordinates = (latitude, longitude)
        self.meals = meals
    }
}
```

Fonte: Apple, 2016.

desse método é que tratar as exceções torna-se complicado e fica mais difícil de encontrar erros no código (Apple, 2016).

Assim, o programador deve avaliar qual método de extração deve ser utilizado para otimizar a conversão de dados no aplicativo.

2.3.2 Aplicativo para o Sistema Operacional Android

A linguagem de programação utilizada para desenvolver aplicativos Android é o Java. Porém, é possível compilar qualquer arquivo ou recursos, independente da extensão, desde que estes estejam no *Android Package* (ADK). É também através do ADK que os dispositivos Android instalam os aplicativos (Android, 2016).

O sistema operacional Android é um sistema multi-usuário dentro do Linux, e a segurança dos aplicativos é baseada nesta característica. Os aplicativos Android são como usuários do sistema Linux e são nomeados com uma *ID* única conhecida apenas pelo sistema operacional, e, assim, um aplicativo só poderá acessar os arquivos que tiverem a mesma *ID*. Além disso, cada aplicativo é rodado em uma máquina virtual (VM) diferente e tem seu próprio processo dentro do Linux. Se algum dos componentes do aplicativo precisar ser executado, o processo será aberto, e, este processo será fechado ao final da execução ou se o sistema operacional precisar de mais memória (Android, 2016).

Ademais, o Android se utiliza do princípio de privilégio mínimo, ou seja, um aplicativo só poderá ter acesso ao conteúdo necessário para seu funcionamento, não podendo acessar nenhuma parte do sistema que não tenha sido autorizada previamente. Para acessar os dados dos usuários, assim como no sistema operacional iOS, o aplicativo deve pedir acesso explicitamente e ser transparente quanto ao uso dos dados (Android, 2016).

Em algumas exceções, quando os aplicativos possuem o mesmo certificado dentro da Play Store, eles podem compartilhar da mesma *ID*, da mesma VM e do mesmo processo. Nesse caso um aplicativo terá acesso ao conteúdo do outro (Android, 2016).

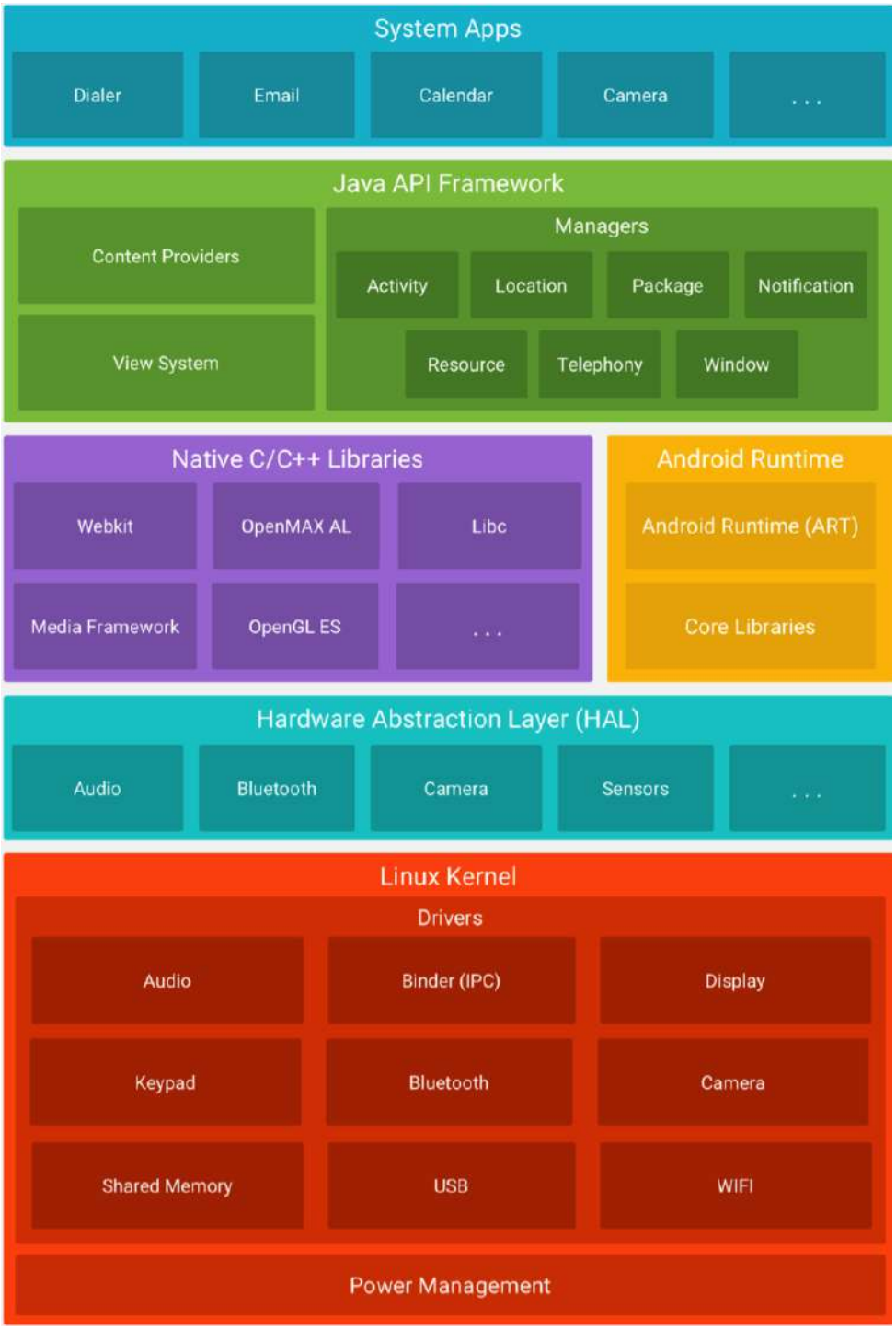
2.3.2.1 A Arquitetura do Sistema Operacional

A Figura 36 mostra o diagrama com os principais componentes da plataforma Android.

O Android, além de ser um *software* baseado no Linux, é também desenvolvido no formato de pilha. O primeiro elemento da pilha é o kernel Linux, já que esta é a base do Android. As vantagens de utilizar esse kernel são diversas, incluindo questões de segurança e facilidade de fabricação (Android, 2016).

Acima do Linux kernel encontra-se o *Hardware Abstraction Layer* (HAL) que consiste em diferentes bibliotecas que fornecem a interface necessária para diferentes componentes do

Figura 36 – *Principais Componentes da Plataforma Android*



Fonte: Android, 2016.

hardware, como a câmera. Quando uma parte do *hardware* é acionada, o sistema carrega a biblioteca referente àquele componente específico (Android, 2016).

O Android *Runtime* vem logo após e tem a função de rodar diferentes máquinas virtuais utilizando pouca memória. Isso é feito ao rodar arquivos especialmente desenvolvidos para

Android e que têm a função de minimizar o uso da memória (Android, 2016).

No mesmo nível que o Android *Runtime* encontram-se as bibliotecas nativas do C e do C++, necessárias já que alguns componentes do Android, como o HAL, foram desenvolvidos utilizando tais linguagens de programação. Contudo, alguns aplicativos também requerem o uso da linguagem C e C++ e, assim, é possível ter acesso a essas bibliotecas através do *Android NDK* (Android, 2016).

O Java API *framework* oferece aos programadores todas as funcionalidades do sistema operacional Android escrito na linguagem Java. Já os aplicativos do sistema são literalmente os aplicativos nativos do sistema que têm a função de servir ao usuário, além de serem a base de outros aplicativos (Android, 2016).

2.3.2.2 Componentes do Aplicativo

Os componentes são as partes principais de um aplicativo Android, sendo através desses que o sistema ou usuário pode acessar o aplicativo. Há quatro tipos de componentes, *activities, services, content providers* e *broadcast receivers*, que possuem diferentes funções e ciclos de vida (Android, 2016).

Uma *activity* é representada por uma única tela com interface para o usuário, sendo o ponto de interação entre aplicativo e usuário. Um aplicativo geralmente tem várias *activities*, que, apesar de trabalharem de maneira coerente para melhorar a experiência do usuário, são independentes umas das outras. É possível que um aplicativo tenha acesso a uma *activity* de um outro *app*, desde que isso tenha sido permitido previamente. As *activities* são implementadas como uma subclasse dentro da classe *Activity* (Android, 2016).

Uma das funções da *activity* é regular o acesso do usuário à tela, de maneira que o processo que hospeda essa *activity* não seja parado. Também devem saber quais *activities* serão utilizadas posteriormente para que os processos dessas continuem funcionando. No caso do sistema operacional fechar o aplicativo inesperadamente, as *activities* devem auxiliar o sistema a retornar ao estado anterior (Android, 2016).

Outro importante componente é o *service*, que tem a função de manter o aplicativo rodando no *background*, não fornecendo interface com o usuário. Para que haja interação com o usuário o *service* precisa chamar um outro componente, como uma *activity*, por exemplo.

Há duas maneiras como o *service* pode lidar com os aplicativos rodando no *background*. Se

o usuário estiver ciente de que há um aplicativo rodando ao fundo, como por exemplo, uma música, então o sistema operacional deve fazer um grande esforço para que este aplicativo continue rodando. Todavia, se o usuário não tiver ciência de que um aplicativo está rodando ao fundo, então o sistema operacional tem maior liberdade para, se preciso, fechar tal aplicativo para gerar memória RAM e rodar os aplicativos que o usuário deseja (Android, 2016).

É possível que o *service* de um aplicativo seja relacionado com o de outro aplicativo, a fim de mostrar dependência entre os dois processos, ou simplesmente mostrar que o usuário tem interesse nos dois processos simultaneamente. Por essa característica, os *services* são muito utilizados para construir, por exemplo, *wallpapers* dinâmicos. Nesse caso, o aplicativo implementa o *service* e o sistema operacional conecta-se com este quando o aplicativo deve estar em funcionamento (Android, 2016).

Os *broadcast receivers*, por sua vez, são responsáveis por permitir que o sistema entregue eventos a aplicativos que não estão sendo utilizados pelo usuário, fazendo com que aplicativos que não estejam rodando possam também responder a uma mensagem de *broadcast*. A maioria dos *broadcasts* são gerados pelo sistema, como por exemplo para informar que a bateria está baixa, porém, alguns aplicativos também podem enviar mensagens de *broadcast* para informar o *download* de um arquivo. Apesar de não fornecer interface com o usuário, alguns *broadcasts* podem aparecer na barra notificação (Android, 2016).

Os últimos componentes a serem descritos são os *content providers*, que permitem que os aplicativos armazenem informações dentro do sistema operacional ou em qualquer outro local de armazenamento, incluindo a Web. Assim, é possível que outros aplicativos, se autorizados, adicionem ou modifiquem esses dados. Além disso, os *content providers* também auxiliam os aplicativos a lerem e escreverem dados que são exclusivos do aplicativo (Android, 2016).

Outra peculiaridade do Android é que qualquer aplicativo pode iniciar o componente de um outro aplicativo. Nesse caso, o sistema operacional irá iniciar o processo relacionado ao componente que deseja-se utilizar em outro aplicativo, e, assim, haverá pelo menos dois processos rodando ao mesmo tempo. Isso mostra que os aplicativos são processados separadamente, ou seja, não há um ponto de entrada único para os aplicativos. Resumindo, não há uma função *Main*, e é o sistema que inicia o novo processo, já que o aplicativo não tem permissão para isso (Android, 2016)

2.3.2.3 Ativação dos Componentes

Os componentes *activity, service* e *broadcast receivers* são ativados por uma mensagem assíncrona chamada de *intent*. Essas mensagens são criadas dentro do objeto *Intent* e podem ativar tanto um componente específico (*intent* explícito), quanto informar a ação que deve ser tomada pelo componente (*intent* implícito). No último caso o sistema deverá encontrar um componente habilitado a realizar tal ação, e, se houver mais de um habilitado, a escolha fica a cargo do usuário (Android, 2016).

No caso das *activities* e dos *services*, a mensagem *intent* informa a ação que deverá ser tomada por esses componente; já no caso dos *broadcast receivers*, essa mensagem é basicamente a mensagem do *broadcast* (Android, 2016).

Por questões de segurança os *content providers* não trocam mensagens diretamente com o componente requisitando a informação. Ao invés disso, esse componente só é ativado quando recebe um pedido do *ContentResolver* (Android, 2016).

2.3.2.4 O arquivo *Manifest*

Para que um aplicativo rode corretamente, o sistema operacional deve conhecer seus componentes e estes são declarados no *manifest file*. O *manifest* é o centro de todo aplicativo, e, além de conter os componentes, também deve identificar as permissões necessárias para rodar o aplicativo, declarar qual versão Android é requerida pelo aplicativo, declarar qual *hardware* é necessário para rodar o aplicativo e declarar as bibliotecas que deverão ser utilizadas (Android, 2016). Na Figura 37 é possível ver como as *activities* são declaradas no *manifest*.

Figura 37 – *Declaração de Componentes no* Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest ... >
    <application android:icon="@drawable/app_icon.png" ... >
        <activity android:name="com.example.project.ExampleActivity"
                  android:label="@string/example_label" ... >
        </activity>
        ...
    </application>
</manifest>
```

Fonte: Android, 2016.

Também é no *manifest file* que os *intent filters* devem ser definidos juntamente com os componentes. Esses atributos são importantes porque é através deles que o sistema é capaz de verificar se um componente é habilitado ou não para realizar uma ação determinada por uma mensagem *intent* implícita (Android, 2016).

A definição de *hardware* e *software* necessários para rodar um determinado aplicativo também é importante porque há diversos dispositivos diferentes rodando o sistema operacional Android, e, através dessas declarações (Figura 38), é possível que o Google Play filtre quais usuários poderão utilizar o aplicativo desenvolvido (Android, 2016).

Figura 38 – *Declaração de Especificações de* Hardware *e* Software

```
<manifest ... >
    <uses-feature android:name="android.hardware.camera.any"
                  android:required="true" />
    <uses-sdk android:minSdkVersion="7" android:targetSdkVersion="19" />
    ...
</manifest>
```

Fonte: Android, 2016.

No caso do código da Figura 38, se o dispositivo não tiver câmera, então ele não terá acesso ao aplicativo. O mesmo ocorrerá se a versão do Android for inferior a 2.1 (API 7) (Android, 2016).

2.3.2.5 Recursos dos Aplicativos

São considerados recursos quaisquer arquivos independentes do código fonte e que têm relação com a apresentação do aplicativo, como as imagens. É necessário fornecer um leque grande de opções para cada recurso, de modo que a interface do aplicativo se adeque aos diversos tamanhos de tela dos dispositivos Android (Android, 2016).

Para cada recurso adicionado ao projeto, o compilador irá associar um número inteiro chamado de *ID*. Assim, é possível fazer referência aos recursos de maneira mais rápida e fácil. Além disso, o fato de separar os recursos do código fonte facilita na mudança de algumas características sem alterar o código fonte (Android, 2016).

2.3.2.6 *JSON*

Por fim, como no caso do sistema operacional iOS, o *software* Android precisará fazer uma conversão da linguagem *JSON* para a linguagem Java quando quiser ter acesso a dados provenientes de uma aplicação Web (servidor). Contudo, como a linguagem Java é orientada a objetos, essa conversão será feita através da manipulação de quatro classes: *JSONArray, JSONObject, JSONStringer* e *JSONTokenizer*.

Para converter as informações em *JSON* é preciso criar um objeto na classe *JSONObject* e atribuir uma *String* em *JSON* a esse objeto (Figura 39).

Figura 39 – *Declaração do* JSONObject

```
String in;
JSONObject reader = new JSONObject(in);
```

Fonte: Produção do próprio autor.

Após isso basta fazer a leitura utilizando o método *getJSONObject* (Figura 40) e fazer as manipulações necessárias utilizando um dos métodos da Tabela 3.

Figura 40 – *Utilização do* getJSONObject

```
JSONObject sys  = reader.getJSONObject("sys");
country = sys.getString("country");

JSONObject main  = reader.getJSONObject("main");
temperature = main.getString("temp");
```

Fonte: Produção do próprio autor.

Tabela 3 – Métodos para Extração de Informações em *JSON*

Método	Descrição
getString(String name)	Retorna o valor da *string* especificada.
get(String name)	Retorna o valor da *string* especificada em forma de objeto.
getBoolean(String name)	Retorna o valor booleano da *string* especificada.
getDouble(String name)	Retorna o valor *double* da *string* especificada.
getInt(String name)	Retorna o valor inteiro da *string* especificada.
getLong(String name)	Retorna o valor *long* da *string* especificada.
length()	Retorna o número de valores/nomes no objeto especificado.
names()	Retorna um vetor cotendo os nomes das *strings* dentro do objeto.

Fonte: Produção do próprio autor.

3 METODOLOGIA E ETAPAS DE DESENVOLVIMENTO

3.1 O Desenvolvimento do Website

3.1.1 Código HTML e CSS Comum a Todas as Páginas do *Website*

Todas as páginas do *website* contam com um *design* similar, assim, foi utilizado um *stylesheet* externo em conjunto com o elemento *link* para fazer referência aos arquivos. A princípio, o IEMA iria disponibilizar verba para contratar um *Web designer* e a aluna ficaria responsável pela semântica do trabalho (códigos HTML e PHP). Contudo, nenhuma verba foi disponibilizada durante a execução do projeto e a aluna teve que se aventurar nessa área de programação. A fim de facilitar, e com o intuito de criar um *responsive website*, a aluna utilizou o Foundation *framework*, e a seguir será descrito como o código foi desenvolvido.

Como já citado no capítulo anterior, um documento HTML, que segue as especificações do W3C, deve iniciar o arquivo com o elemento *Doctype*. A seguir vem o *root element*, o qual, no caso, será HTML, seguido pelos elementos *head* e *body*.

No elemento *head* foram definidos os elementos de *metadata*, e, para criar um *responsive website*, foi utilizado a *viewport meta tag*, ou seja, ao elemento *meta* foram atribuídos os atributos *name*, com valor de *viewport*, e *content*, com o valor de *"width = device-width, initial-scale = 1"*, como mostrado na Figura 21. Após isso é preciso definir o título, o qual varia de acordo com a página. Para mais informações a respeito do título é possível acessar o *website* no domínio www.infoar.ufes.br e observar as abas do navegador.

Após isso, ainda dentro do elemento *head*, foi definido o elemento *link*, fazendo referência aos três diferentes arquivos CSS: um chamado de *style.css*, que trata da apresentação da página em geral, como fontes de texto e definição de classes; e outros dois arquivos chamados de *jquery.bxslider.css* e *nav.css*, que tratam do *image slider* e da barra de navegação, respectivamente. Os dois últimos arquivos fazem parte do Foundation *framework* e, assim, não foram desenvolvidos pela aluna; as únicas mudanças feitas foram para alteração da cor da fonte e do *background*.

Enfim inicia-se o elemento *body*, e juntamente com ele, o trabalho do programador. Primeiramente, foi adicionado o elemento *script* e utilizado o atributo *src*, fazendo referência aos arquivos JavaScript para a barra de navegação. Após, foi declararada a classe *nav-outer* (já desenvolvida no *stylesheet nav.css*), sendo que essa classe faz referência ao fundo azul

da barra de navegação, ocupando toda a largura da tela (Figura 41). Depois, foi preciso utilizar uma outra classe, também já definida no *stylesheet*, chamada de *nav-wrap*, que engloba todos os outros elementos da barra de navegação, incluindo a logo e o texto. Assim, a fim de adicionar a logo e o texto, foi preciso declarar uma classe, utilizando a *tag nav*, chamada de *navigation*, a qual também está descrita no *stylesheet*.

Dentro da classe *navigation* declarou-se a classe "logo", e, a essa classe, associou-se a imagem desejada. Posteriormente, foi definida a classe *nav*, e, ao atributo *nav-menu-style*, associou-se o valor *yoga*, o qual está descrito no arquivo JQuery do Foundation e é responsável pelo *design* da barra de navegação. Na classe *nav* é criada uma lista não ordenada (*tag*) associada à classe *nav-menu*, e, dentro desta classe, são listados os *links* para as páginas do *website* utilizando o elemento *href*, o qual aponta para os códigos-fonte de cada página. No caso do infoAR, as páginas são: "Home", "Info", "Poluentes", "Cadastre-se" e "Fale Conosco" (Figura 41).

Figura 41 – *Apresentação das Classes* nav-outer *e* nav-wrap *nos Navegadores*

Fonte: Produção do próprio autor.

Ainda dentro do elemento *body* foi criado um novo elemento *script*, que, através do atributo *src*, aponta para o arquivo com os códigos JavaScript para o *image slider*. Assim, foi criada a classe *slide-wrap*, a qual, de maneira similar à classe *nav-wrap*, engloba todo o conteúdo do *image slider*.

Com isso, utilizou-se a *tag section* e criou-se uma classe chamada *slider*. Nesta foi criada uma lista não ordenada e uma classe chamada de *slider1*, onde foram listadas as imagens que compõem o *image slider* (Figuras 42 e 43). A fim de configurar o *image slider*, foi criado um *script* interno com um pequeno código JavaScript, a fim de que, no momento da transição das imagens, houvesse uma característica chamada de *fading*.

Após isso foram adicionados o "título"(utilizando a *tag* h1) e as funcionalidades de cada página, as quais serão explicadas em detalhes adiante.

Para finalizar, há, na versão para computadores, uma seção com informações dos desenvolvedores e do IEMA. Essa seção foi produzida utilizando uma das funcionalidades do Foundation *framework* chamada de *parallax*, a qual está descrita no *stylesheet*. O interessante do *parallax* é que, ao utilizar o *scroll bar*, a imagem ao fundo da mensagem se movimenta, enquanto o texto permanece imóvel.

Figura 42 – *Primeira imagem do* Image Slider

Fonte: Produção do próprio autor.

Figura 43 – *Segunda imagem do* Image Slider

Fonte: Produção do próprio autor.

Para tanto, criou-se, utilizando a *tag section*, uma classe chamada de *parallax -1*, e, dentro dessa classe, foi criada a classe *parallax-inner* que engloba todo o conteúdo de texto. A fim de que houvesse duas colunas de texto (uma para o "Sobre Nós" e outra para o IEMA), foi criada a classe *one-half* cuja apresentação foi programada no *stylesheet style.css*. A Figura 44 mostra a seção com *parallax*, sendo que seu conteúdo é omitido na versão para celular.

Ao fim temos o rodapé que informa o ano de desenvolvimento e a presença de direitos autorais (Figura 45). Para criá-lo foi utilizada o elemento *footer* junto com a *tag* que representa o parágrafo (<p>). Após isso os elementos *body* e *root* são finalizados a fim de terminar o documento HTML.

Figura 44 – *Seção com a Funcionalidade* Parallax

Fonte: Produção do próprio autor.

Figura 45 – *Rodapé da Página*

Fonte: Produção do próprio autor.

O procedimento descrito até agora está presente nos códigos HTML de todas as páginas e, por isso, foi descrito detalhadamente. Na Figura 46 é possível ver o fluxograma desse código.

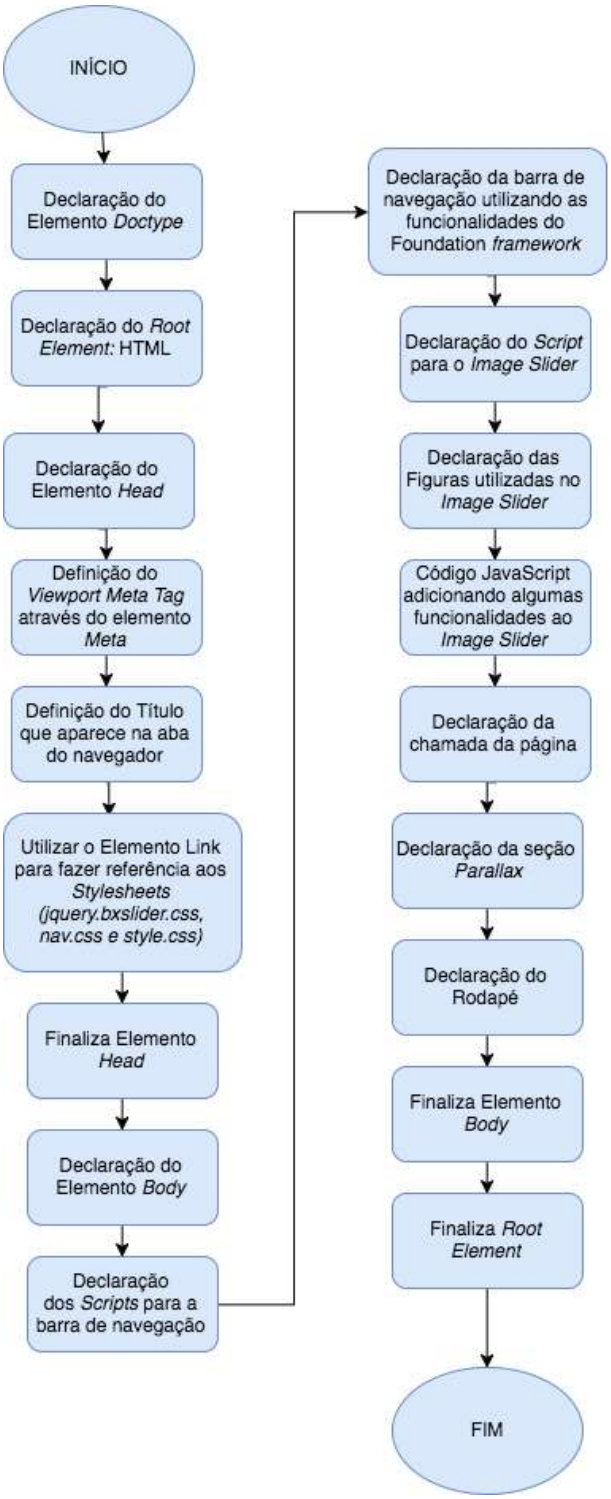
Para mais detalhes a respeito do *layout*, basta acessar o *website* em: www.infoar.ufes.br. Além disso, tanto as imagens utilizadas quanto a logo foram produzidas por um publicitário que cobrou R$200,00 pelo serviço. O recurso utilizado para o pagamento veio de um dos projetos da professora Jane em parceria com o IEMA.

Apesar de utilizar o *framewrok* Foundation, o qual já possui os códigos JQuery e JavaScript desenvolvidos, o programador teve que alterar o arquivo *style.css* (código CSS), a fim de configurar as *tags*, os atributos e o elemento *media* para o *responsive website*.

Para os elementos *head* e *body* as características básicas definidas foram: cor de fundo, fonte utilizada e tamanho da fonte. E, com o intuito de utilizar uma fonte customizada, foi necessário importar no início do código as fontes desejadas (Figura 47). Já para as *tags*, algumas outras características precisaram ser definidas, como alinhamento, *padding*, que são espaços até a margem, e a própria margem, a qual é a distância até o *viewport* (Figura 48).

Após isso, foi preciso definir as diversas classes com as seguintes características: largura, alinhamento e margem (Figura 49). No caso da classe *parallax*, também foi preciso definir

Figura 46 – *Fluxograma do Código Base HTML*



Fonte: Produção do próprio autor.

a imagem que seria utilizada no *background* (Figura 50).

A fim de explicar como o elemento *media* funciona, foi desenvolvido um fluxograma, o qual explica como o *website* irá se ajustar a diversos tamanhos de tela (Figura 51). Com

Figura 47 – *Importar Fontes Customizadas*

```
@import url(https://fonts.googleapis.com/css?family=Hind);
@import url(https://fonts.googleapis.com/css?family=Nobile);
```

Fonte: Produção do próprio autor.

Figura 48 – *Definição dos Elementos e* Tags

```
head{
    background:#87CEEB ;
}

body{

    background-color: #FFF;
    font-family:'Hind', sans-serif;
    font-size: 18px;
    position: relative;
}

h1{

    font-family:'Nobile', sans-serif;
    text-align: center;
    font-size: 175%;
    color: #4A4444;
    text-transform: uppercase;
    letter-spacing: 3px;
    padding:3% 0; /*top&bottom left&right*/

}
```

Fonte: Produção do próprio autor.

Figura 49 – *Definição das Classes no Arquivo CSS*

```
.one-half{

    width:44%; /*não é 50% por causa da margem*/
    float: left;
    margin: 2% 0 3% 4%; /* top/right/bottom/left*/
    text-align: center;
}

.one-third{
    width:28%;
    float:left;
    margin: 2% 0 3% 4%;
    text-align: center;
}
```

Fonte: Produção do próprio autor.

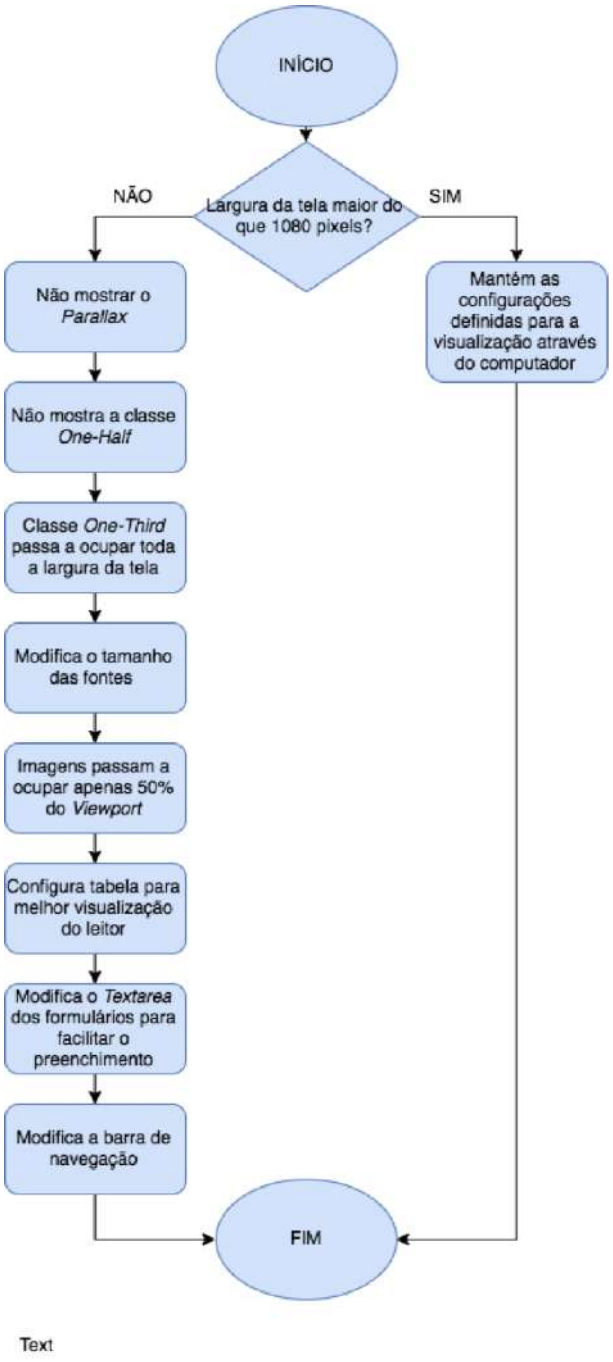
Figura 50 – *Definição da Classe* Parallax

```
.parallax-1{
    margin-top:5%;
    background: url("img/parallax.png") repeat fixed 100%;
    text-align:center;
}
```

Fonte: Produção do próprio autor.

isso, as características comuns a todas as páginas do *website* foram explicadas. A seguir serão explicadas as particularidades de cada página e como essas foram programadas.

Figura 51 – *Fluxograma do Atributo* Media *no Código CSS*



Fonte: Produção do próprio autor.

3.1.2 Manipulação do Banco de Dados

Antes de explicar as funcionalidades de cada página, é preciso explicar como ocorre a manipulação do banco de dados, já que muitas páginas, e os aplicativos, se utilizam desse recurso. Primeiramente, será explicado como foram definidas as tabelas dentro do MySQL e como essas tabelas se relacionam entre si. A teoria utilizada para tanto foi o Modelo de

Entidades e Relacionamentos. Após isso, será indicado como os dados das concentrações dos poluentes foram armazenados no banco de dados.

3.1.2.1 Criação do Banco de Dados Utilizando o MER

Para a criação do banco de dados, primeiro foram definidas as entidades, as quais, para este projeto, são: Pessoa, Estação e Dados. A entidade Pessoa representa as pessoas que podem se cadastrar no *website* para receber alertas a respeito da qualidade do ar em uma determinada região, assim, seus atributos são: Nome, email, endereço, data de aniversário, condição de saúde e uma ID (única para cada pessoa).

Já para a entidade Estação foram estabelecidos os atributos nome e ID, e para a entidade Dados foram estabelecidos os atributos ID da estação, poluente e concentração.

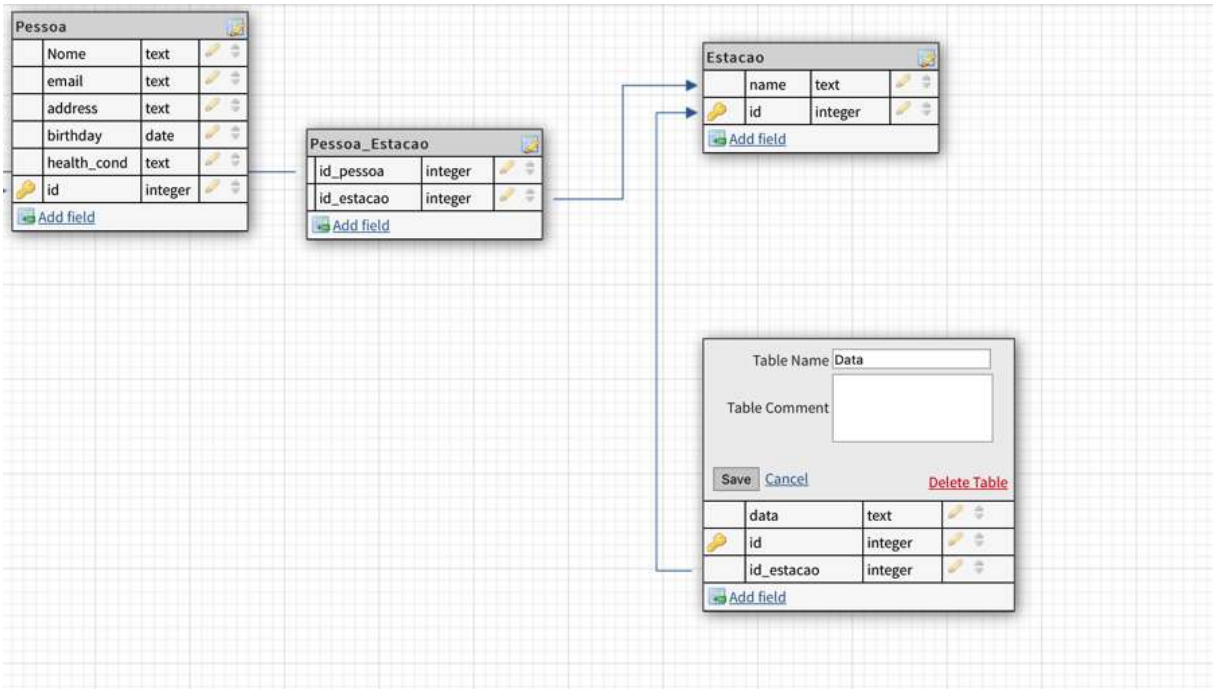
Após isso, foi necessário estabelecer o relacionamento entre as entidades. A entidade Dados possui um relacionamento "muitos para um" com a entidade Estação, não possuindo relacionamento com a entidade Pessoa. Já a entidade Estação possui um relacionamento "um para muitos" com a entidade Pessoa. Para facilitar o sistema de envio de e-mails para as pessoas cadastradas, foi criada uma outra tabela que relaciona a ID da pessoa com a ID da estação.

Assim, com o auxílio de um *software* CASE, foi criado um diagrama para o banco de dados, o qual é mostrado na Figura 52.

3.1.2.2 Armazenando Dados Coletados no Banco de Dados

A priori, o objetivo era ter acesso direto ao "Miglis", ou seja, ao banco de dados do IEMA, contudo, isso não foi possível e a autora se utilizou de alguns poucos arquivos obtidos, com extensão .txt, para construir um único arquivo que informasse as concentrações dos poluentes em cada estação. O formato do arquivo é o mesmo que o utilizado pelo IEMA e pode ser visto na Figura 53. Neste projeto, E0X representa uma estação específica e P0X um poluente específico. Assim, E01 representa a estação de Laranjeiras, E02 a de Carapina, E03 a de Jardim Camburi, E04 a da Enseada do Suá, E05 a do Centro de Vitória, E06 a do Ibes, E07 a do Centro de Vila Velha, E08 a de Cariacica e E09 a de Cidade Continental. E, de forma semelhante, P01 representa as MP10, P03 o SO2, P04 o NO2, P07 o CO, P08 o O3 e P18 as MP2.5.

Figura 52 – *DER do Banco de Dados Projetado*



Fonte: Produção do próprio autor.

Figura 53 – *Formato do Arquivo Utilizado na Leitura*

```
*#dd/mm/yyyy#hh:mm#;#.#MIG46-07/0008-55#
13/01/2015 00:30;E01P03;19.4301986694336;;;
13/01/2015 00:30;E01P04;201.45002746582;;;
13/01/2015 00:30;E01P07;328.545288085938;;;
13/01/2015 00:30;E01P08;25.2625617980957;;;
13/01/2015 00:30;E01P01;45.7401504516602;;;
13/01/2015 00:30;E02P01;17.5403594970703;;;
```

Fonte: Produção do próprio autor.

Os números subsequentes representam as concentrações de cada poluente, e, a fim de classificar uma certa concentração como “boa”, “moderada”, “ruim”, “muito ruim” ou “péssima” foi tomado como base os valores mostrados na Figura 54.

Contudo, a concentração de CO analisada na tabela está na escala ppm, enquanto que a concentração de CO recebida das estações vem na escala μ/m^3. Assim, foi preciso fazer a conversão utilizando a Equação 3.1, na qual que Y representa a concentração em μ/m^3, e 28.01g/mol representa a massa molecular do CO.

$$X(ppm) = \frac{(Y * 0.001) * (24.45)}{28.01} \tag{3.1}$$

Figura 54 – *Classificação dos Poluentes de Acordo com a Concentração*

Índice/ Classificação	MP10 (µg/m³) 24h	MP2,5 (µg/m³) 24h	O3 (µg/m³) 8h	CO (ppm) 8h	NO2 (µg/m³) 1h	SO2 (µg/m³) 24h
0 - 40 (Boa)	0 - 50	0 - 100	0 - 100	0 - 9	0 - 200	0 - 20
41 - 80 (Moderada)	> 50 - 100	> 100 - 130	> 100 - 130	> 9 - 11	> 200 - 240	> 20 - 40
81-120 (Ruim)	> 100 - 150	> 130 - 160	> 130 - 160	> 11 - 13	> 240 - 320	> 40 - 365
121 - 200 (Muito Ruim)	> 150 - 250	> 160 - 200	> 160 - 200	> 13 - 15	> 320 - 1130	> 365 - 800
> 200 (Péssima)	> 250	> 200	> 200	> 15	> 1130	> 800

Fonte: IEMA, 2013.

Enfim, para calcular o IQA é preciso fazer uma regra de três relacionando o valor da concentração obtida e as extremidades de sua classificação. Por exemplo, na Equação 3.2 é mostrado como se calcula o IQA para o poluente MP10 e concentração (Y) abaixo de $50\mu/m^3$.

$$IQA = \frac{Y * 40}{50} \tag{3.2}$$

Essa equação varia de acordo com cada poluente; mais detalhes podem ser vistos no código em Anexo.

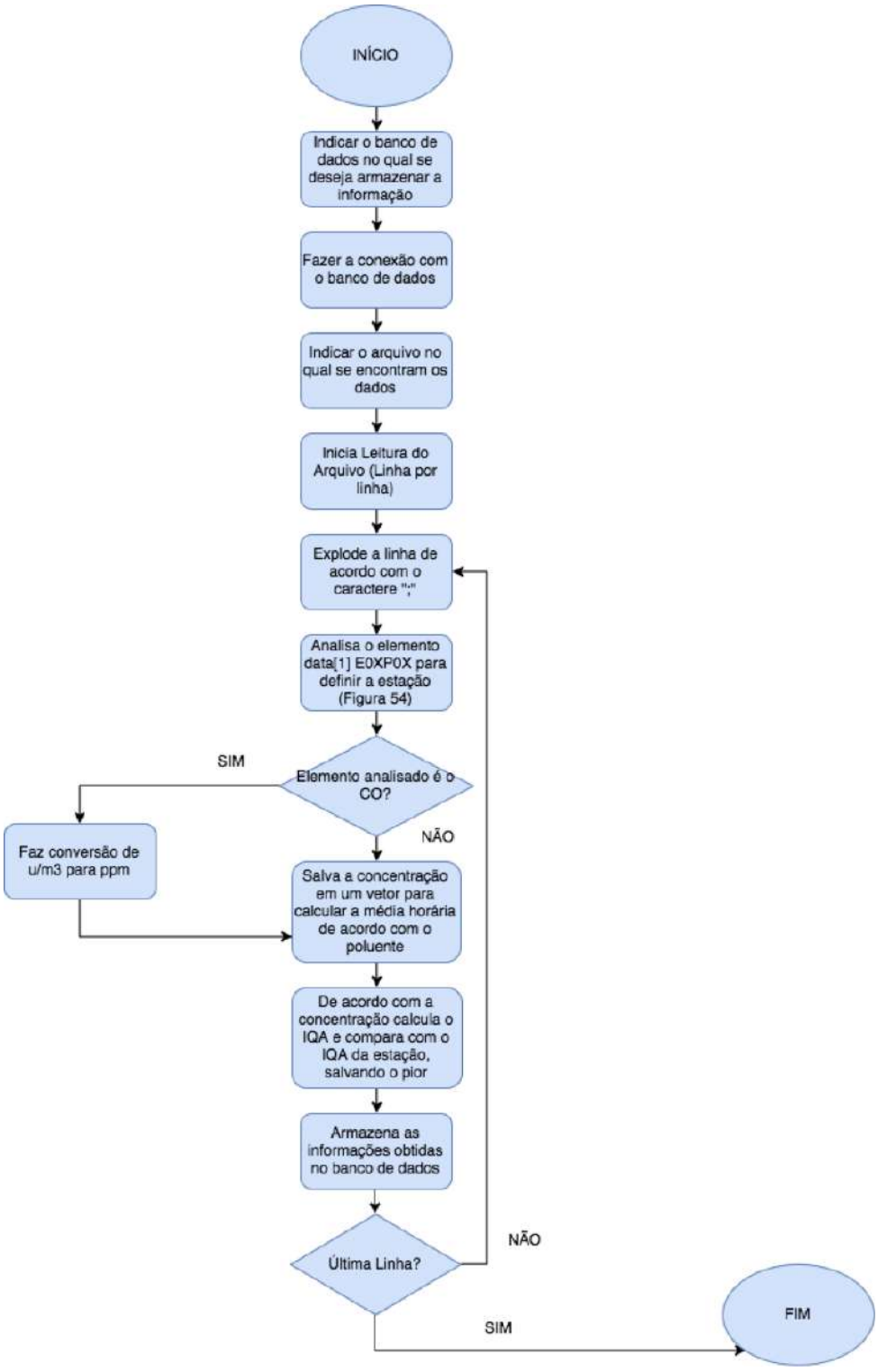
A fim de fazer a leitura desse arquivo e salvar as informações no banco de dados, foi preciso criar um código PHP, já que essa é uma funcionalidade dinâmica. Os fluxogramas das Figura 55 e 56 explicam o código PHP desenvolvido. Para mais detalhes basta consultar os Anexos.

Considerando que o arquivo contendo os dados vindos das estações deva ser atualizado de hora em hora, então, esse código também deve ser processado pelo *software* PHP de hora em hora. Para isso, é preciso fazer uso de um processo chamado de *Cron Table (cronTab)* que tem como função facilitar o agendamento de tarefas repetitivas.

No *Cron Table* é preciso criar um *cron job*, o qual possui cinco colunas representando operadores cronológicos e, após isso, é preciso descrever o caminho até o arquivo que será executado.

Os operadores cronológicos são os minutos, horas, dias (de um mês), meses e dias da

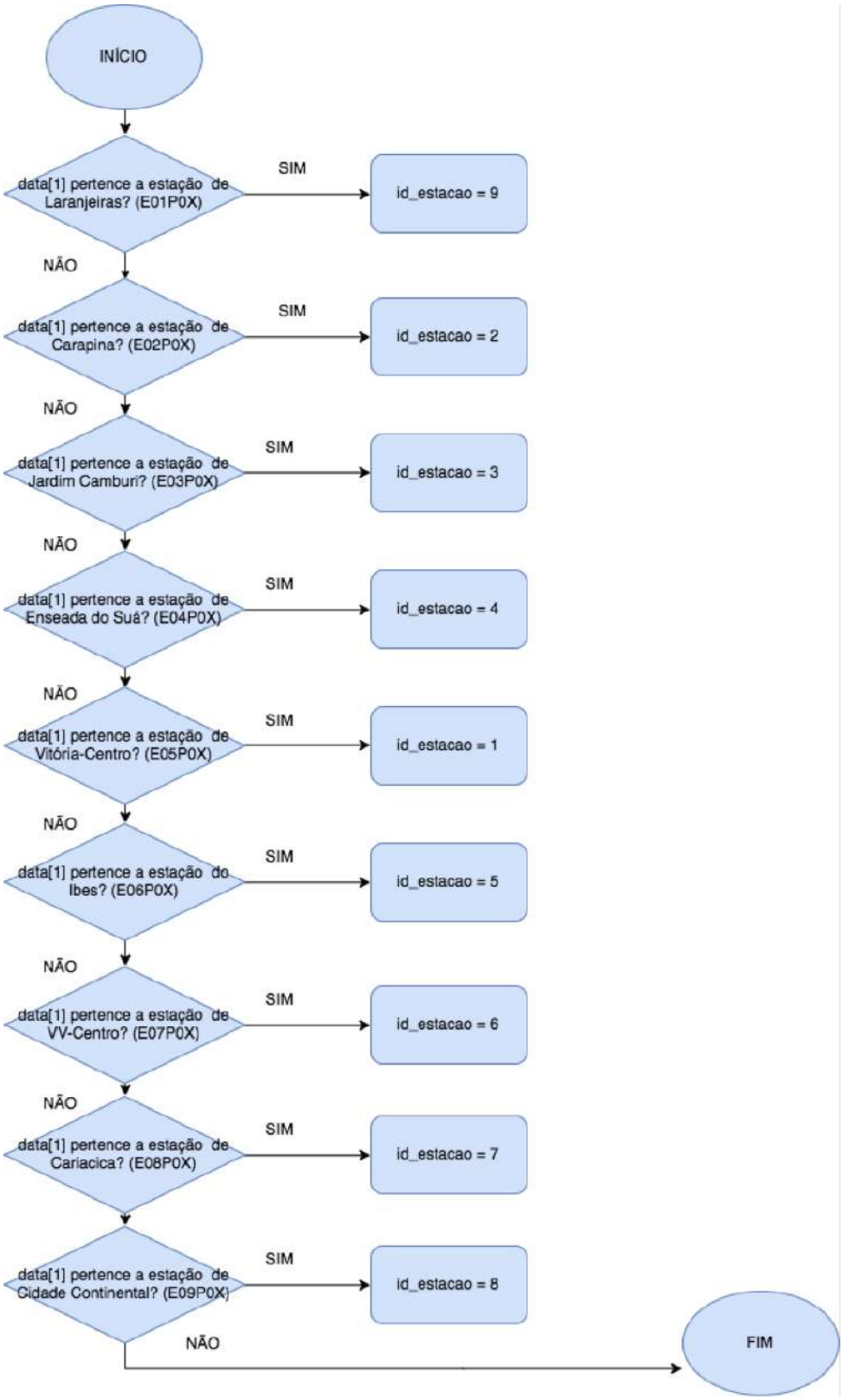
Figura 55 – *Fluxograma do Código PHP Desenvolvido para Armazenar os Dados no Banco de Dados*



Fonte: Produção do próprio autor.

semana. Os minutos são representados por um "vetor" variando de [0-59], as horas variam de [0-23], os dias de [1-31], os meses [1-12] e os dias da semana também são numéricos variando de [0-6]. O caractere "*" é muito utilizado pois pode representar qualquer valor,

Figura 56 – *Fluxograma Para Verificar a Estação*



Fonte: Produção do próprio autor.

sendo utilizado em qualquer um dos cinco operadores cronológicos.

Uma característica *default* do *cronTab* é que ele envia uma notificação de email toda vez que uma tarefa agendada foi executada. Para que isso não ocorra é preciso redirecionar esse comando pra o dispositivo */dev/null*.

Para que o *cronTab* possa funcionar corretamente é preciso que o programa *cron daemon* esteja rodando no *background* do servidor Web, pois esse é o responsável por chamar os *cron Jobs* quando estes são agendados. Todo o agendamento é armazenado em um arquivo chamado de *crontab* que lista todas as tarefas.

Além disso, o processo *Cron Table* só funciona em máquinas rodando o sistema operacional Linux/Unix. Assim, se o servidor estiver hospedado em uma máquina Windows, será preciso utilizar o *Windows Task Scheduler*. Neste projeto, como o *website* está hospedado no servidor da UFES, as máquinas virtuais utilizadas utilizam o sistema Linux. Também deve ser possível acessar o servidor utilizando o *Secure Shell Access* (SSH).

Para criar um arquivo *crontab*, basta listar todos os *cron Jobs* em um arquivo com extensão .txt e salvá-lo no servidor. No caso do servidor da UFES, é preciso pedir aos funcionários de TI que o faça.

Para transformar os dados armazenados dentro do MySQL em dados JSON, é preciso utilizar um código PHP (disponível nos Anexos) que também deve ser processado de hora em hora, ou seja, é preciso utilizar o processo *CronTab*.

3.1.3 Particularidades da Página Inicial

Na página inicial a particularidade é o mapa contendo as nove estações da RAMQAr, as quais são representadas pelo ícone de nuvem, cuja cor varia de acordo com o IQA. O mapa utilizado é proveniente do Google Maps e é um elemento *script* do tipo JavaScript. Há uma extensa documentação sobre a inserção do Google Maps em *websites* e algumas funções já prontas, cabendo ao programador a customização.

Primeiramente, no código HTML (elemento *script*), foi preciso iniciar o mapa utilizando a função *initMap*. Também foram definidas as localizações das estações com base nas respectivas latitudes e longitudes. Para que o mapa focasse na ilha de Vitória, foi preciso definir, além do elemento central, que, no caso, foi a UFES, o *zoom* para que as estações ficassem em evidência (Figura 57). Para que o mapa aparecesse na tela, foi preciso criar um objeto *infowindow* utilizando a classe *google.maps.InfoWindow*.

Figura 57 – *Inicialização do Mapa no Código HTML*

```
<div id="map"></div>
  <script type= "text/javascript">
    function initMap() {
      var ufes = {lat: -20.27, lng: -40.30};
      var locations = [
          ['Laranjeiras', -20.196, -40.245],
          ['Carapina', -20.226, -40.259],
          ['J.Camburi', -20.255, -40.270],
          ['Enseada do Sua', -20.313, -40.291],
          ['Centro-Vitoria', -20.321, -40.333],
          ['Ibes', -20.348, -40.317],
          ['VV-Centro', -20.336, -40.291],
          ['Cariacica', -20.342, -40.402],
          ['Cidade Continental', -20.225, -40.218],
      ];
      var map = new google.maps.Map(document.getElementById('map'), {
        zoom: 12,
        center: ufes
      });

       var infowindow = new google.maps.InfoWindow();
```

Fonte: Produção do próprio autor.

Após isso o objeto ícone foi definido, sendo que esses ícones foram desenhados pela autora utilizando o *software* Sketch. A classe para os ícones, também chamados de *markers*, é a *google.maps.Marker*, na qual é possível configurar a posição dos ícones, a imagem e o texto mostrado dentro das nuvens (Figura 58).

Figura 58 – *Customização dos Ícones*

```
  for (i = 0; i < locations.length; i++) {
   marker = new google.maps.Marker({
   position: new google.maps.LatLng(locations[i][1], locations[i][2]),
   icon: 'img/verde.png',
   label:iqa,
   map: map
});
    google.maps.event.addListener(marker, 'click', (function(marker, i)
        {
   return function() {
     infowindow.setContent(locations[i][0]);
     infowindow.open(map, marker);
   }
})(marker, i));
```

Fonte: Produção do próprio autor.

Além de definir o mapa no código HTML, também foi preciso configurá-lo no código CSS, já que foi criada uma ID específica para o elemento mapa. As principais características definidas foram: a largura e a altura que este ocuparia na página Web (Figura 59).

O *layout* do mapa na tela do computador e na tela do celular é mostrado, respectivamente, nas Figuras 60 e 61.

Figura 59 – *Definição do Mapa no Código CSS*

```
.map{
    position: absolute;
        top: 0;
        left: 0;
        width: 100% ;
        height: 100% ;
}
```

Fonte: Produção do próprio autor.

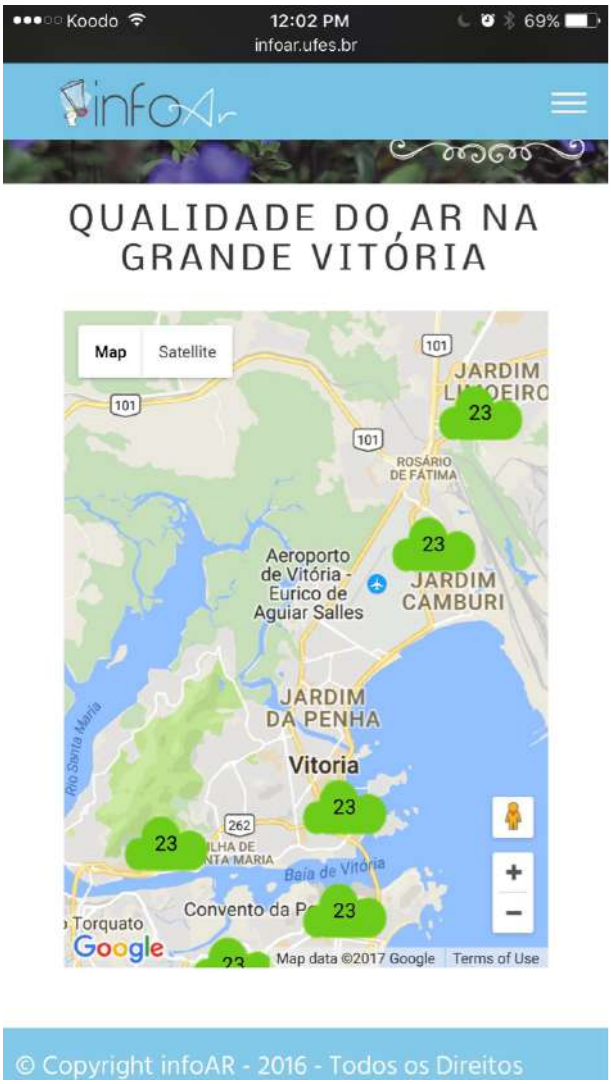
Figura 60 – *Mapa no* Website

Fonte: Produção do próprio autor.

Além disso, ainda na página inicial, é preciso mostrar o IQA de cada estação. Como já explicado anteriormente, a cada hora um código PHP irá rodar, a fim de salvar os dados no banco de dados MySQL. Assim, foi necessário importar esses dados do banco de dados e mostrar no mapa. Para o elemento *Google Maps* é possível importar dados localizados no mesmo domínio do *website*, utilizando a classe *map.data.loadGeoJson()*, ou fora deste, utilizando o *Cross-Origin Resource Sharing* (CORS). Para o desenvolvimento deste projeto, os dados coletados estavam alocados no mesmo domínio.

Contudo, a informação coletada pela classe *map.data.loadGeoJson* é apenas relativa à geografia de um determinado *marker*, e, a fim de utilizar algum outro tipo de dado (no nosso caso o IQA), foi preciso utilizar a classe *google.maps.Data*. Além disso, a fim de determinar como os dados deveriam ser mostrados aos usuários, foi preciso utilizar o elemento *Data Layer*, já fornecido pelo Google, e a classe *Data.setStyle()*. Os fluxogramas nas Figuras 62 e 63 explicam os códigos HTML e CSS, respectivamente, utilizados para

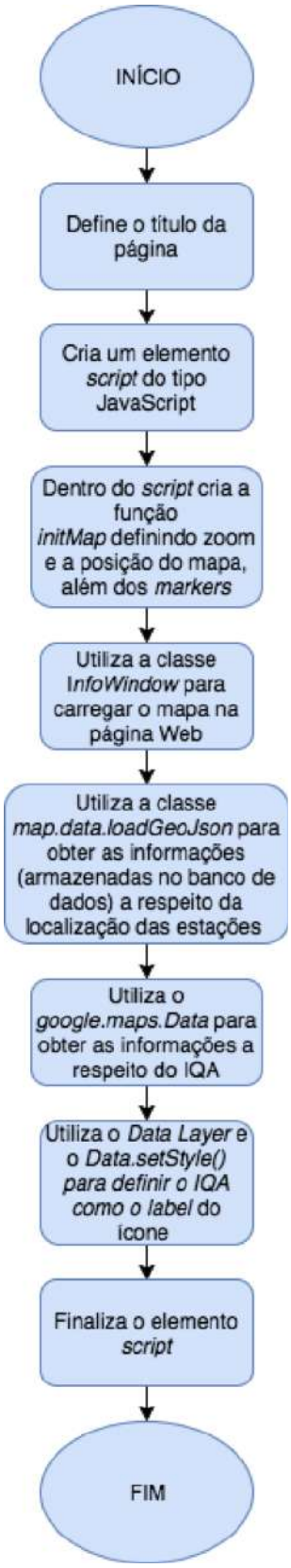
Figura 61 – *Mapa em um* Smartphone



Fonte: Produção do próprio autor.

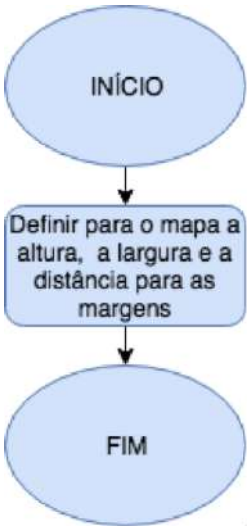
estilizar a página inicial.

Figura 62 – *Fluxograma do Código HTML das Particularidades da Página Inicial*



Fonte: Produção do próprio autor.

Figura 63 – *Fluxograma do Código CSS das Particularidades da Página Inicial*



Fonte: Produção do próprio autor.

3.1.4 Particularidades da Página Contendo as Informações

A página contendo as informações, como o próprio nome já diz, tem o objetivo de informar ao usuário o significado do IQA e das cores das nuvens que são mostradas no mapa. A relação entre o IQA e os efeitos na saúde são mostrados em uma tabela, e, a fim de obter melhores informações, é possível acessar uma cartilha desenvolvida pela Dra. Jane M. Santos, professora do Departamento de Engenharia Ambiental da UFES, e coorientadora deste trabalho. Além disso, é explicado resumidamente como o IQA é calculado, informando também a existência de aplicativos para iOS e Android.

Para definir uma tabela em um código HTML é preciso utilizar a *tag table*, sendo que, cada linha é definida pela *tag tr*, e cada célula é definida pela *tag td*. Além disso, a fim de definir o cabeçalho é preciso utilizar a *tag th* (Figura 64).

Essa tabela deve ter o formato ajustado em *responsive websites* para facilitar a interação com o usuário. Assim, no código CSS, a desenvolvedora optou por omitir o cabeçalho e modificar as cores das células com o intuito de diferenciá-las, e também optou por apresentar as células uma embaixo da outra. A maneira como a tabela é exibida em *desktops* pode ser vista na Figura 65, enquanto a Figura 66 mostra a tabela na tela de um celular.

O acesso à cartilha é feito através de um botão, o qual mostra o documento que está armazenado dentro do domínio do website. Para tanto, foi preciso utilizar a *tag form* e definir o método como *get*, além de indicar onde o arquivo está localizado dentro do

Figura 64 – *Definição da Tabela no Código HTML*

```
<table>
    <thead>
        <tr>
            <th>&Iacute;ndice/Classifica&ccedil;&atilde;o</th>
            <th>Significado</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td><img src="img/bom.png"><br> BOM </td>
            <td>Atendendo as recomenda&ccedil;&otilde;es da OMS (Padr&otilde;es finais da
            legisla&ccedil;&atilde;o final do ES), para que a sa&uacute;de da popula&ccedil;&
            atilde;o seja preservada ao m&aacute;ximo, em rela&ccedil;&atilde;o aos danos
            causados pela polui&ccedil;&atilde;o atmosf&eacute;rica </td>
        </tr>
```

Fonte: Produção do próprio autor.

Figura 65 – *Tabela na Tela de um* Desktop

ÍNDICE/CLASSIFICAÇÃO	SIGNIFICADO
0-40 BOM	Atendendo as recomendações da OMS (Padrões finais da legislação final do ES), para que a saúde da população seja preservada ao máximo, em relação aos danos causados pela poluição atmosférica
41-80 MODERADO	Pessoas de grupos sensíveis (crianças, idosos e pessoas com problemas respiratórios ou cardíacos)podem apresentar sintomas como tosse seca e cansaço. A população em geral não é afetada
81-120 RUIM	Toda a população pode apresentar sintomas como tosse seca, cansaço, ardor nos olhos, nariz e garganta. Pessoas dos grupos sensíveis (crianças, idosos e pessoas com problemas respiratórios ou cardíacos) podem apresentar efeitos mais sérios na saúde
121-200 MUITO RUIM	Toda a população pode apresentar agravamento dos sintomas como tosse seca, cansaço, ardor nos olhos, nariz e garganta e ainda falta de ar e respiração ofegante. Efeitos ainda mais graves à saúde de grupos sensíveis (crianças, idosos e pessoas com problemas respiratórios ou cardíacos)
>200 PÉSSIMO	Toda a população pode apresentar sérios riscos de manifestação de doenças respiratórias e cardiovasculares. Aumento de mortes prematuras em pessoas dos grupos sensíveis
SEM REPRESENTATIVIDADE	Dados sem representatividade

Fonte: Produção do próprio autor.

servidor. Ao clicar no botão, definido pela *tag input*, o usuário tem a opção de fazer o download do conteúdo. A semântica do código HTML para essa funcionalidade é mostrada na Figura 67 enquanto o conteúdo mostrado pelo *browser* é mostrado na Figura 68.

Além disso, ao final da página, é informada a existência dos aplicativos para iOS e Android. O objetivo é facilitar o download dos aplicativos pelo usuário, o qual deveria apenas clicar na imagem representando as lojas para ser direcionado à URL dos aplicativos dentro das mesmas (Figura 69). Infelizmente, até o presente momento, não foi possível publicar os aplicativos dentro das lojas devido ao problema no acesso ao banco de dados do IEMA.

Figura 66 – *Tabela na Tela de um Celular*

Fonte: Produção do próprio autor.

Figura 67 – *Semântica para Desenvolver o Botão de Acesso à Cartilha*

```
<h3>Para mais informa&ccedil;&otilde;es veja nossa cartilha clicando no bot&atilde;o abaixo!</h3>
<form class = "form" method = "get" action = "downloads/Guia_Qualidade_Ar.pdf">
<p class = "submit">
            <input type = "submit" style = "background-color:#87CEEB" value = "Cartilha">
        </p>

</form>
```

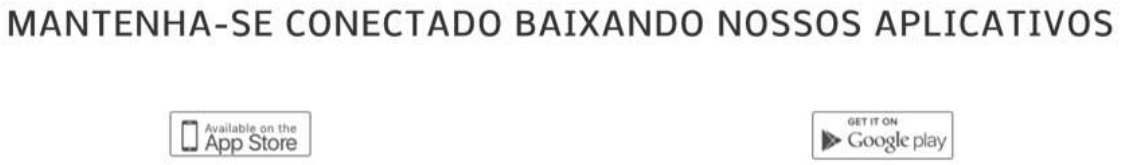
Fonte: Produção do próprio autor.

Os fluxogramas nas Figuras 70 e 71 explicam as particularidades dos códigos HTML e CSS, respectivamente, desenvolvidos para a página contendo as informações.

Figura 68 – *Conteúdo Apresentado pelo Navegador*

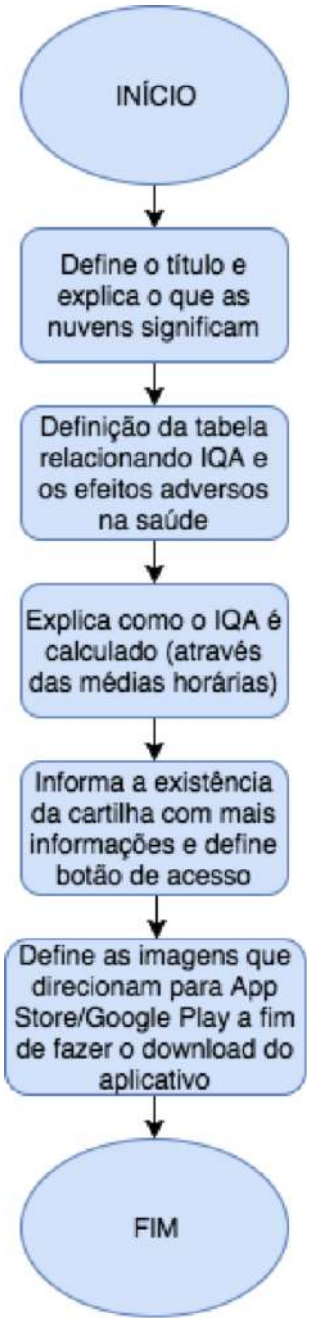
Fonte: Produção do próprio autor.

Figura 69 – *Imagens Direcionando para as Lojas (App Store e Google Play)*



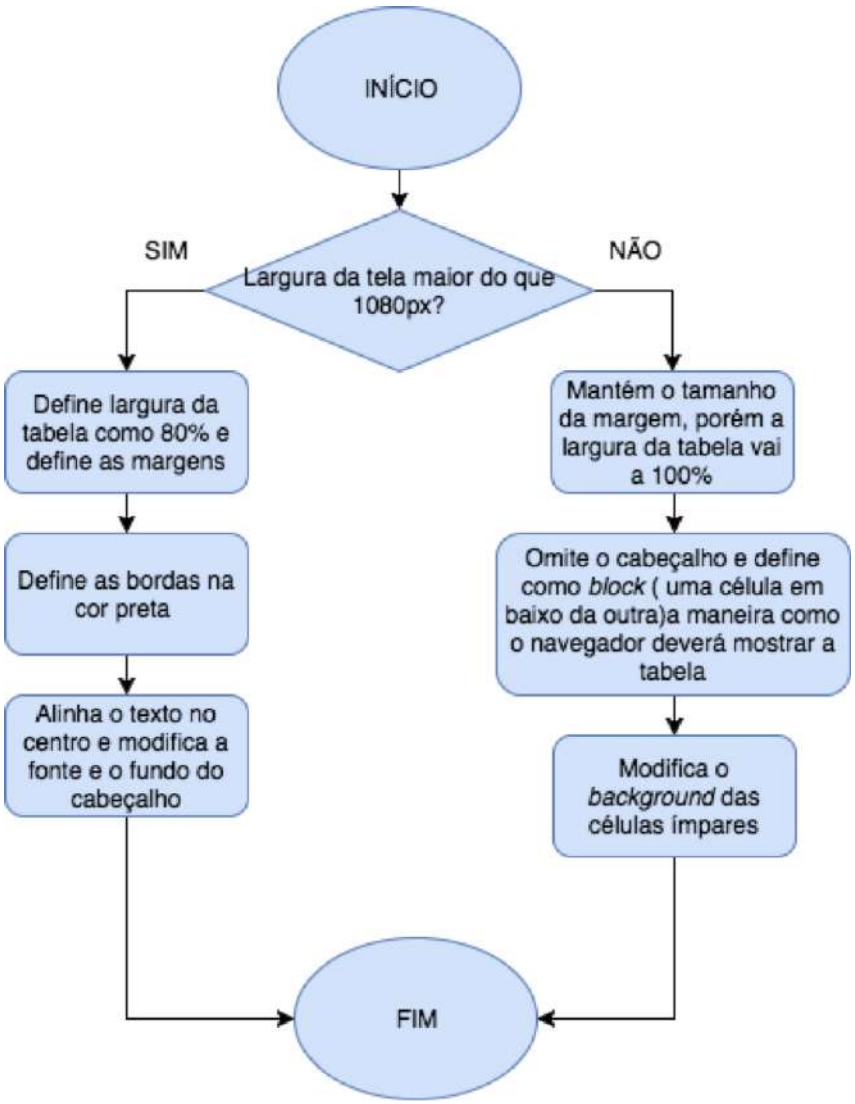
Fonte: Produção do próprio autor.

Figura 70 – *Fluxograma do Código HTML das Particularidades da Página de Informações*



Fonte: Produção do próprio autor.

Figura 71 – *Fluxograma do Código CSS das Particularidades da Página de Informações*



Fonte: Produção do próprio autor.

3.1.5 Particularidades da Página Contendo Informações dos Poluentes

Essa página tem o objetivo de indicar, para cada estação, a concentração dos poluentes na última hora. Assim, cada estação é representada por uma tabela dinâmica que irá receber dados de um arquivo JSON.

A leitura do JSON é feita utilizando o elemento *script* do tipo JavaScript. No arquivo JavaScript foi criada uma função (*createTableFromJSON()*) para transformar os dados no arquivo JSON em uma tabela. Primeiramente, foram extraídos os dados referentes ao cabeçalho da tabela (Figura 72).

Figura 72 – *Extração Dados Para o Cabeçalho da Tabela*

```
var col = [];
        for (var i = 0; i < concent.length; i++) {
            for (var key in concent[i]) {
                if (col.indexOf(key) === -1) {
                    col.push(key);
                }
            }
        }
```

Fonte: Produção do próprio autor.

No código da Figura 72 a variável *concent* representa todo o conteúdo armazenado no arquivo JSON.

Após isso foi possível criar uma tabela em JavaScript, ou seja, dinâmica, utilizando o método *createElement("table")* (Figura 73). Com a tabela criada, é possível adicionar o conteúdo extraído anteriormente ao cabeçalho utilizando o comando *createElement("th")* (Figura 73).

Figura 73 – *Criação da Tabela Dinâmica e Adição do Cabeçalho*

```
var table = document.createElement("table");

var tr = table.insertRow(-1);

        for (var i = 0; i < col.length; i++) {
            var th = document.createElement("th");
            th.innerHTML = col[i];
            tr.appendChild(th);
        }
```

Fonte: Produção do próprio autor.

Finalmente foi possível adicionar todo o conteúdo do JSON dentro da tabela utilizando o comando *insertCell()*, e também criar as bordas da tabela (Figura 74).

Figura 74 – *Adição do conteúdo JSON na Tabela*

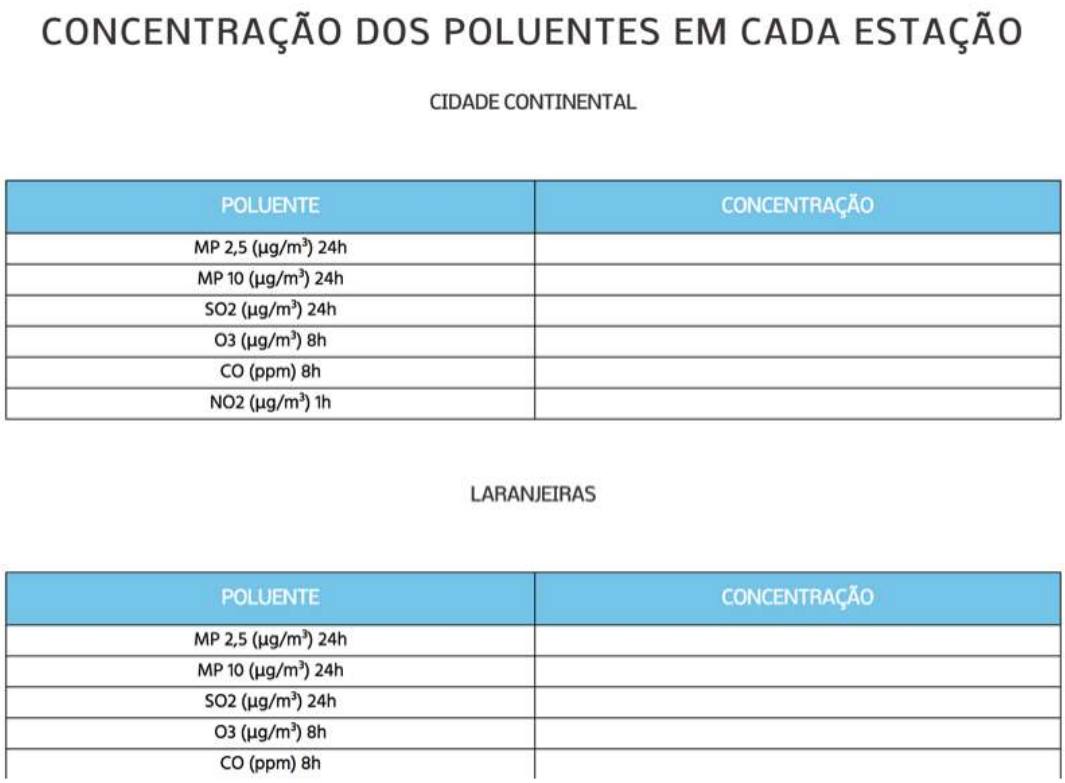
```
for (var i = 0; i < concent.length; i++) {

    tr = table.insertRow(-1);

    for (var j = 0; j < col.length; j++) {
        var tabCell = tr.insertCell(-1);
        tabCell.innerHTML = concent[i][col[j]];
    }
}

var divContainer = document.getElementById("showData");
divContainer.innerHTML = "";
divContainer.appendChild(table);
```

Fonte: Produção do próprio autor.

Assim, fora do elemento *script*, basta chamar a função *CreateTableFromJSON()* para que o navegador mostre as tabelas dinâmicas, sendo que o mesmo procedimento deve ser feito para cada uma das estações. Além disso, utilizando o código CSS, foi possível modificar as cores de fundo e a fonte, de maneira que esta tabela seguisse o mesmo padrão da tabela de informações. Da mesma forma, para o *responsive website* as dimensões da tabela foram modificadas e o cabeçalho omitido, a fim de facilitar a navegação. As Figuras 75 e 76 mostram, respectivamente, o *layout* das tabelas na tela de um computador e na tela de um celular.

O fluxograma na Figuras 77 explica as particularidades do código HTML desenvolvido para a página contendo as concentrações dos poluentes. O código CSS, por tratar apenas da estilização da tabela, é muito similar ao código CSS da seção anterior, e por isso não será explicado novamente.

Figura 75 – *Layout das Tabelas dos Poluentes no* Desktop



CONCENTRAÇÃO DOS POLUENTES EM CADA ESTAÇÃO

CIDADE CONTINENTAL

POLUENTE	CONCENTRAÇÃO
MP 2,5 (µg/m³) 24h	
MP 10 (µg/m³) 24h	
SO2 (µg/m³) 24h	
O3 (µg/m³) 8h	
CO (ppm) 8h	
NO2 (µg/m³) 1h	

LARANJEIRAS

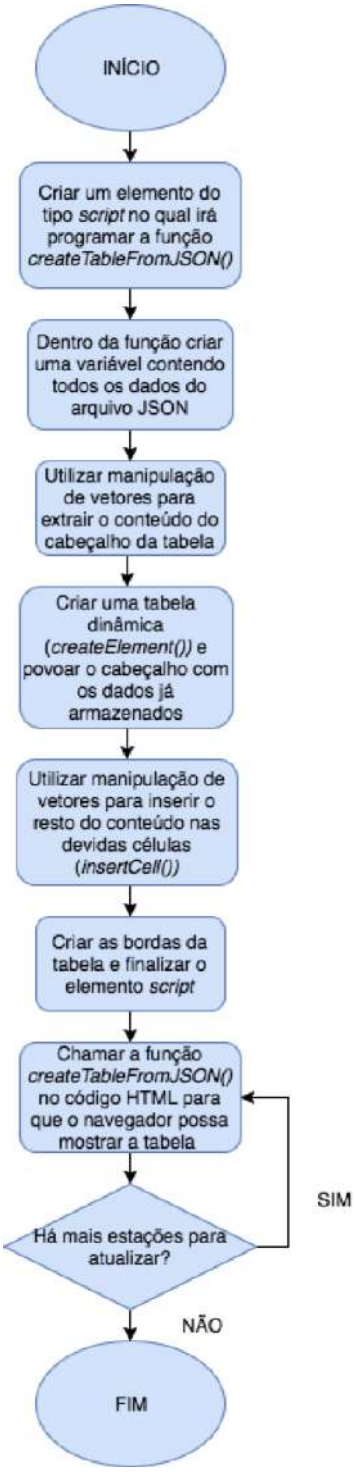
POLUENTE	CONCENTRAÇÃO
MP 2,5 (µg/m³) 24h	
MP 10 (µg/m³) 24h	
SO2 (µg/m³) 24h	
O3 (µg/m³) 8h	
CO (ppm) 8h	

Fonte: Produção do próprio autor.

Figura 76 – *Layout das Tabelas no Celular*

Fonte: Produção do próprio autor.

Figura 77 – *Fluxograma do Código HTML das Particularidades da Página com as Concentrações dos Poluentes*



Fonte: Produção do próprio autor.

3.1.6 Particularidades da Página Cadastre-se

A página que oferece ao usuário a possibilidade de se cadastrar para receber e-mails é basicamente um formulário, e, assim, é uma página Web dinâmica e que precisa utilizar a linguagem PHP para salvar os dados com sucesso no banco de dados MySQL.

As informações necessárias para fazer o cadastro são nome, e-mail, endereço, data de nascimento, estação a qual deseja-se receber informações e a condição de saúde (se possui problemas respiratórios ou cardíacos). Com isso, é possível identificar as pessoas que fazem parte do grupo de risco, e assim, mesmo se a qualidade do ar estiver moderada, essas pessoas irão receber e-mails de alerta.

No código HTML foi preciso criar os espaços nos quais as pessoas poderão informar os dados. Para o nome, email e endereço, isso foi feito utilizando a *tag input* do tipo texto. Além disso, todos os dados devem estar inclusos dentro da *tag form* que aponta para o arquivo PHP (*save_user.php*), o qual irá processar toda a informação (Figura 78).

Figura 78 – *Código HTML para Informar Dados de Texto*

```
<form class ="form" method="post" action="save_user.php">
<section class = "one-third">
    <p class = "name">
        <input type = "text" name="name" id="name" placeholder="Maria Silva"/>
        <label for= "name">Nome*</label>
    </p>
</section>
```

Fonte: Produção do próprio autor.

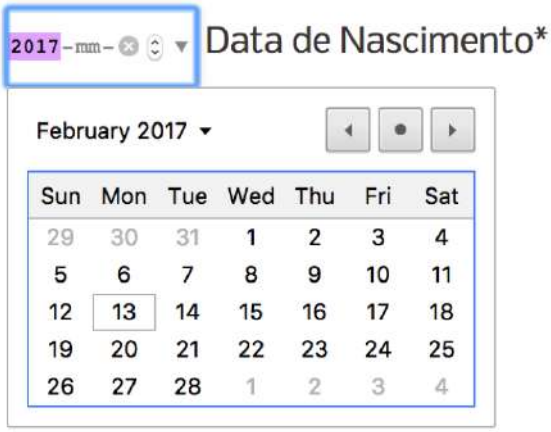
Já para a data de nascimento, como é um arquivo do tipo "data", é possível informar a mesma tanto digitando, quanto utilizando um calendário disponível nos navegadores (Figura 79). A única diferença em relação ao código apresentado na Figura 78 é que a *tag input* é do tipo *date*.

No caso das estações, e também das condições de saúde, foi utilizada a *tag select*, assim, o usuário não precisa digitar tal informação, e sim escolher entre as opções apresentadas. O código HTML é mostrado na Figura 80, e na Figura 81 é possível ver como o navegador interpreta o código e o mostra ao usuário.

Por fim foi criado um botão utilizando a *tag input*, porém do tipo *submit*, que, ao ser clicado irá chamar o código PHP para processar as informações recebidas.

No código PHP a primeira coisa a se fazer é informar qual o banco de dados que deseja-se

Figura 79 – *Informar a Data de Nascimento no Navegador*



Fonte: Produção do próprio autor.

Figura 80 – *Utilização da* Tag Form *para Chamar o Arquivo PHP*

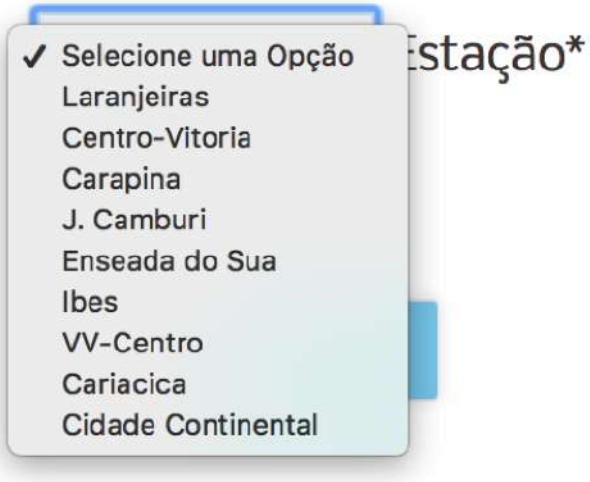
```
<section class="one-third">
    <p class = "estacao">
    <select name="estacao" style= "height: 40px">
    <option value="opcao"/>Selecione uma Op&ccedil;&atilde;o</option>
    <option value="0"/>Laranjeiras</option>
    <option value="1"/>Centro-Vitoria</option>
    <option value="2"/>Carapina</option>
    <option value="3"/>J. Camburi</option>
    <option value="4"/>Enseada do Sua</option>
    <option value="5"/>Ibes</option>
    <option value="6"/>VV-Centro</option>
    <option value="7"/>Cariacica</option>
    <option value="8"/>Cidade Continental</option>
    </select>
    <label for="estacao">Esta&ccedil;&atilde;o*</label>
</section>
```

Fonte: Produção do próprio autor.

utilizar. Após isso, por questões de segurança, é preciso verificar o conteúdo digitado pelo usuário. Esse passo é de extrema importância, pois, com um pouco de conhecimento, um *hacker* pode digitar certos valores que podem, ou tornar público todo o conteúdo armazenado no banco de dados, ou simplesmente deletar o mesmo.

Para o nome só são aceitas letras maiúsculas e minúsculas e os espaços. Caso o usuário digite qualquer outro tipo de caractere, uma mensagem de erro será gerada e mostrada no

Figura 81 – *Escolha das Estações no Navegador*



Fonte: Produção do próprio autor.

topo do botão de enviar (Figura 82). O código PHP que confere o conteúdo submetido é mostrado na Figura 83.

Figura 82 – *Mensagem de Erro para Nome não válido*

Fonte: Produção do próprio autor.

Figura 83 – *Código PHP que faz o Processamento do Nome*

```php
$name= test_input($_POST['name']);
if (!preg_match("/^[a-zA-Z ]*$/", $name)) {
  $_SESSION['STR_ERRO']  = 'Formato de nome inválido. Apenas letras e espaços em branco são permitidos';
  header("Location: ./cadastrese.php#cadastrese");
  exit(1);
}
```

Fonte: Produção do próprio autor.

O código PHP confere os dados de endereço e data de nascimento da mesma forma, porém, no primeiro caso, só são permitidos números, letras maiúsculas e minúsculas e espaços em branco, enquanto no segundo caso só são permitidos números e a barra. Em ambos os casos, contudo, será mostrada uma mensagem de erro seguindo o padrão da Figura 82.

No caso do e-mail, a fim de conferir se é um e-mail válido, foi utilizada uma função chamada de *filter_var*, a qual processa o conteúdo recebido, e, se este não for válido, retorna o valor 0. O código PHP utilizado para isso é mostrado na Figura 84.

Figura 84 – *Código PHP para Conferir o Email*

```php
$email= test_input($_POST['email']);
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
   $_SESSION['STR_ERRO']  = 'Formato de email inválido';
   header("Location: ./cadastrese.php#cadastrese");
   exit(1);
}
```

Fonte: Produção do próprio autor.

No caso das estações e da condição de saúde, apesar do usuário ter as opções já prontas, é possível que este acesse o código fonte do *website* através do navegador e mude os valores pré-determinados pelo programador. Assim, se o valor recebido for diferente, uma mensagem de erro também será gerada e esses dados não serão armazenados no banco de dados. A Figura 85 mostra como o código PHP confere esses valores.

Figura 85 – *Código PHP para Conferir os Dados das Estações e Condições de Saúde*

```php
$station = test_input($_POST['estacao']);
if(($station != "0") && ($station != "1") && ($station != "2") && ($station != "3")&& ($station != "4") &&
    ($station != "5") && ($station != "6") && ($station != "7") && ($station != "8") && ($station != "opcao
    ")){
     $_SESSION['STR_ERRO']  =   ' A estação selecionada é inválida';
     header("Location: ./cadastrese.php#cadastrese");
    exit(1);
}
```

Fonte: Produção do próprio autor.

Enfim, se todos os valores submetidos forem válidos, o código PHP irá armazenar os dados no banco de dados MySQL. Nesse caso em específico os valores serão adicionados em duas tabelas, uma com os dados dos usuários e uma outra tabela que relaciona o usuário com sua estação (Figura 86). Esta última é importante, pois facilita a programação na hora de enviar os e-mails. Se todos os dados foram recebidos com sucesso, uma mensagem é gerada para informar ao usuário (Figura 87). As Figuras 88 e 89 mostram, respectivamente, as tabelas Pessoa e Pessoa_Estacao dentro do banco de dados MySQL.

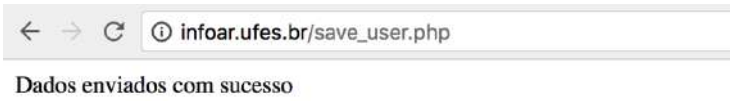
Para essa página não foi preciso modificar o código CSS, já que todos os elementos faziam parte da classe *one-third* que já foi explicada anteriormente. Assim, os fluxogramas das Figuras 90 e 91 representam, respectivamente os códigos HTML e PHP.

Figura 86 – *Armazenamento de Dados no Banco de Dados*

```
$sql1 = "INSERT INTO Pessoa (Nome, email, address, birthday, health_cond)
        VALUES ('$name','$email','$address', '$birthdate', '$health_cond')";

$sql2 = "INSERT INTO Pessoa_Estacao(id_pessoa, id_estacao)
        VALUES (last_insert_id(), '$station')";

mysql_select_db('infoar');
```

Fonte: Produção do próprio autor.

Figura 87 – *Mensagem: Dados enviados com sucesso*



Fonte: Produção do próprio autor.

Figura 88 – *Tabela Pessoa no Banco de Dados*

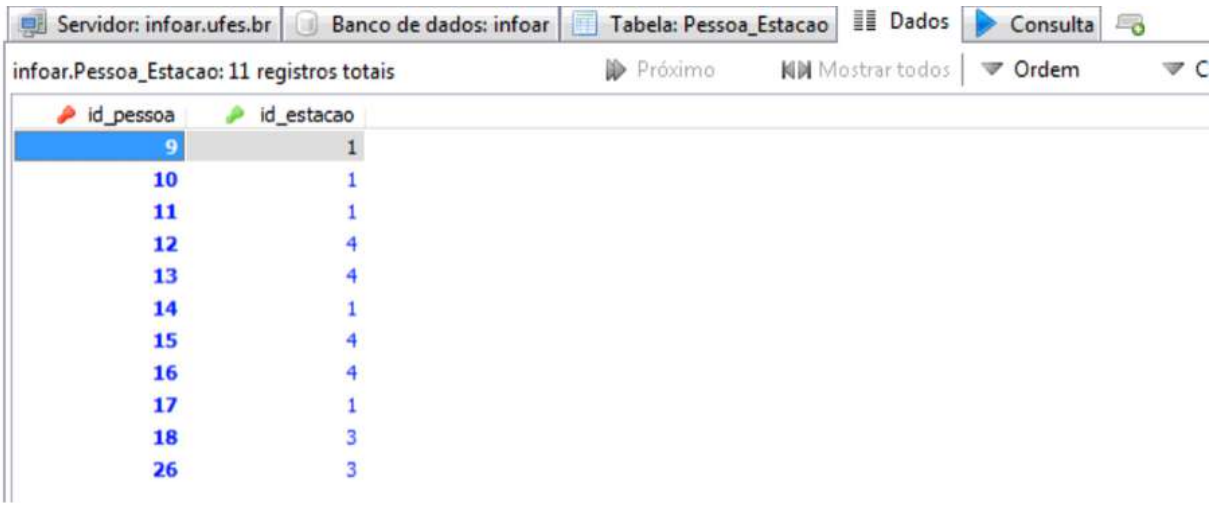
Servidor: infoar.ufes.br | Banco de dados: infoar | Tabela: Pessoa | Dados | Consulta

infoar.Pessoa: 13 registros totais — Próximo | Mostrar todos | Ordem | Colunas (6/6) | Filtr

Nome	email	address	birthday	health_cond	id
Solange Oliveira	sololiveira65@hotmail.com	Rua Marcondes de Souza, n62, apt 302	1965-01-24	Cardiacos	10
Vitor	vdmascarenhas@gmail.com	Rua Dukla de Aguiar, 80	2016-11-03	Respiratorio	13
Orlando Teixeira	orlandotoliveira@gmail.com	Rua Marcondes de Souza, n 62, apt 302	1956-12-16	Nenhum	11
Vitor	vdmascarenhas@gmail.com	Avenida Maruipe, n 100	1992-05-03	Respiratorio	12
Mayara Oliveira	mayara1@ualberta.ca	Rua Marcondes de Souza, n62, apt 302	1993-12-27	Nenhum	9
Yago Oliveira	yago6090@gmail.com	Rua Marcondes de Souza, n62, apt 302	2003-03-15	Nenhum	14
Vitor	vdmascarenhas@gmail.com	Rua Marcondes de Souza, n62, apt 302	2003-03-15	Respiratorio	15
Vitor	vdmascarenhas@gmail.com	R. Dukla de Aguiar	1992-05-03	Respiratorio	16
David Rossolatos	davidrossolatos@outlook.com		1989-10-06	Nenhum	17
Mayara Oliveira	mayara1@ualberta.ca	Rua Marcondes de Souza n62 apt 302	2016-11-08	Ambos	26
Mayara Oliveira234	mayara1@ualberta.ca		1993-12-27	Nenhum	23
Mayara Oliveira	mayara1@ualberta.ca	Rua Marcondes de Souza n62 apt 302	1993-12-27	Cardiacos	24
Mayara Oliveira	mayara1@ualberta.ca	Rua Marcondes de Souza n62 apt 302	1993-12-27	Cardiacos	25

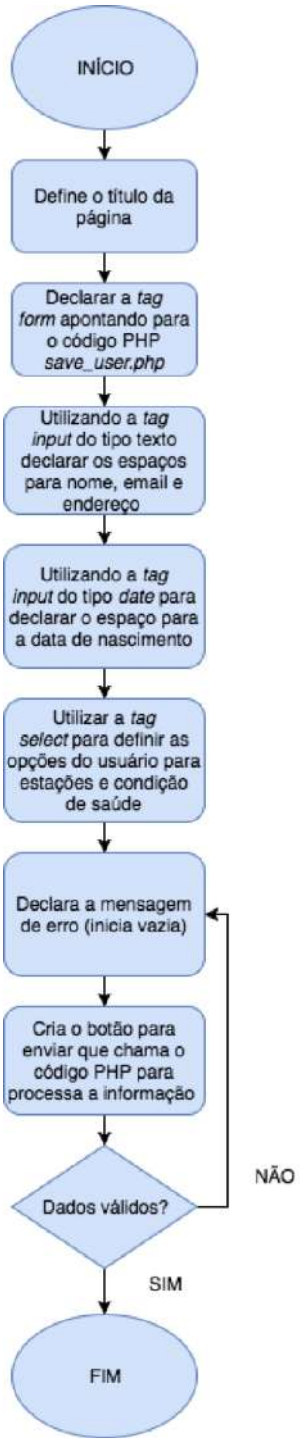
Fonte: Produção do próprio autor.

Figura 89 – *Tabela Pessoa_Estacao no Banco de Dados*



Fonte: Produção do próprio autor.

Figura 90 – *Fluxograma do Código HTML da Página de Cadastro*



Fonte: Produção do próprio autor.

Figura 91 – *Fluxograma do Código PHP da Página de Cadastro*

Fonte: Produção do próprio autor.

3.1.7 Particularidades da Página Fale Conosco

A página de Fale Conosco oferece ao usuário a possibilidade de fazer reclamações ou sugestões a respeito do *website* . Essa página foi montada da mesma maneira que a página Cadastre-se, ou seja, utilizando a *tag input* do tipo texto para montar o formulário. As exceções são o botão de enviar, que foi desenvolvido utilizando a *tag input* do tipo *submit*, e a *tag form* que neste caso aponta para o arquivo PHP *send_email.php* (Figura 92).

Figura 92 – *Utilização da* Tag Form *para Chamar o Arquivo PHP*

```
<form class ="form" method = "post" action="send_email.php">
<section class = "one-third">
    <p class = "name">
        <input type = "text" name="name" id="name" placeholder="Maria Silva"/>
        <label for= "name">Nome*</label>
    </p>
</section>
```

Fonte: Produção do próprio autor.

Ao clicar no botão de submeter, o código PHP irá começar a rodar, e o primeiro passo é chamar dois arquivos chamados de *class.phpmailer.php* e *class.smtp.php* (Figura 93). Após isso é preciso indicar o email para o qual se deseja enviar a mensagem (no caso, "infoarapp@gmail.com"). O assunto da mensagem enviada será sempre o mesmo (Dúvidas e Sugestões Website), e o conteúdo será baseado nos dados submetidos pelo usuário (Figura 94).

Figura 93 – *Incluindo Arquivos do PHPMailer*

```
require_once('phpmailer/class.phpmailer.php');
require_once('phpmailer/class.smtp.php');
```

Fonte: Produção do próprio autor.

Figura 94 – *Definição do Email e do Conteúdo*

```
$email_to = "infoarapp@gmail.com";
$email_subject = "Duvidas e Sugestoes Website";

$name = ($_POST['name']);
$email_from = ($_POST['email']);
$phone = ($_POST['phone']);
$text = ($_POST['text']);
```

Fonte: Produção do próprio autor.

Por questões de segurança também é necessário verificar o conteúdo submetido pelo usuário. Primeiramente foi verificado se todos os campos haviam sido preenchidos, depois

verificou-se se o e-mail era válido, e, por fim, se o comentário possuía mais do que dois caracteres. Por fim, utilizando a função *smtpmailer* (Figura 95), o e-mail será enviado, e se, a operação foi concluída com sucesso, uma mensagem será gerada para informar ao usuário (Figura 96). A Figura 97 mostra como o e-mail é recebido.

Figura 95 – *Função* smtpmailer

```php
function smtpmailer($to, $from, $from_name, $subject, $body) {
    global $error;
    $mail = new PHPMailer();  // create a new object
    $mail->IsSMTP(); // enable SMTP
    $mail->SMTPDebug = 0;  // debugging: 1 = errors and messages, 2 = messages only
    $mail->SMTPAuth = true;  // authentication enabled
    $mail->SMTPSecure = 'ssl'; // secure transfer enabled REQUIRED for GMail
    $mail->Host = 'smtp.gmail.com';
    $mail->Port = 465;
    $mail->Username = 'infoarapp@gmail.com';
    $mail->Password = 'infoar2016';
    $mail->SetFrom($from, $from_name);
    $mail->Subject = $subject;
    $mail->Body = $body;
    $mail->AddAddress($to);
    if(!$mail->Send()) {
        $error = 'Mail error: '.$mail->ErrorInfo;
        print($error);
        return false;
    } else {
        $error = 'Message sent!';
        return true;
    } print($error);
    print('aqui');
}
```

Fonte: Produção do próprio autor.

Figura 96 – *Mensagem Informando ao Usuário o Sucesso no envio*

Fonte: Produção do próprio autor.

Figura 97 – *E-mail recebido*

Fonte: Produção do próprio autor.

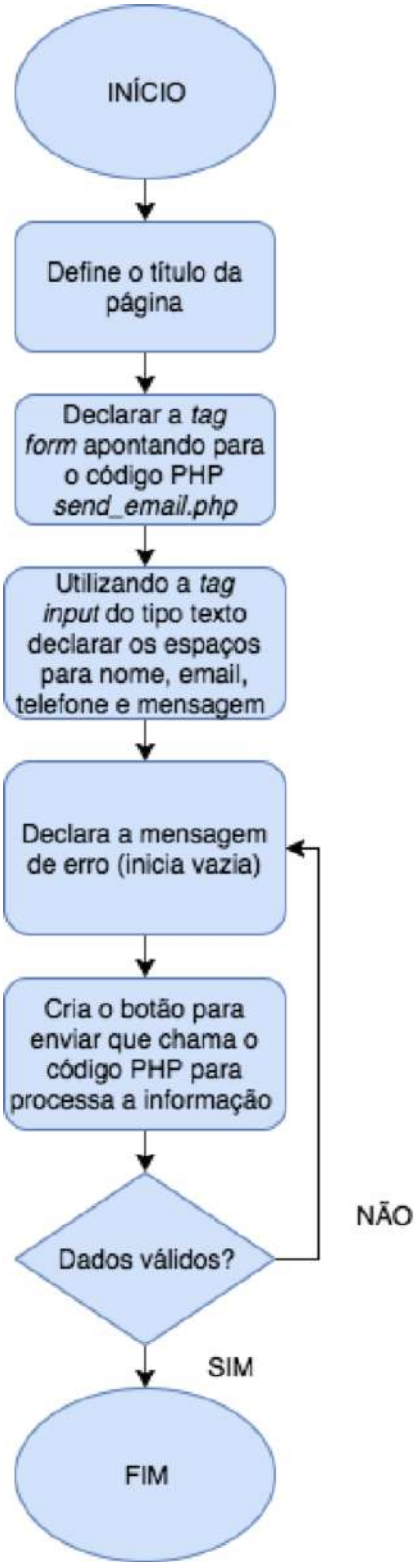
O e-mail que será enviado para as pessoas cadastradas utilizará a mesma função *smptmailer*; a diferença serão os valores atribuídos a cada variável. Porém, a fim de enviar os e-mail

diariamente, é necessário utilizar o *CronTable*, e isso só pode ser feito pelos funcionários do Núcleo de Tecnologia da Informação (NTI) da UFES.

Os fluxogramas que explicam os códigos HTML e PHP da página Fale Conosco estão, respectivamente, nas Figuras 98 e 99.

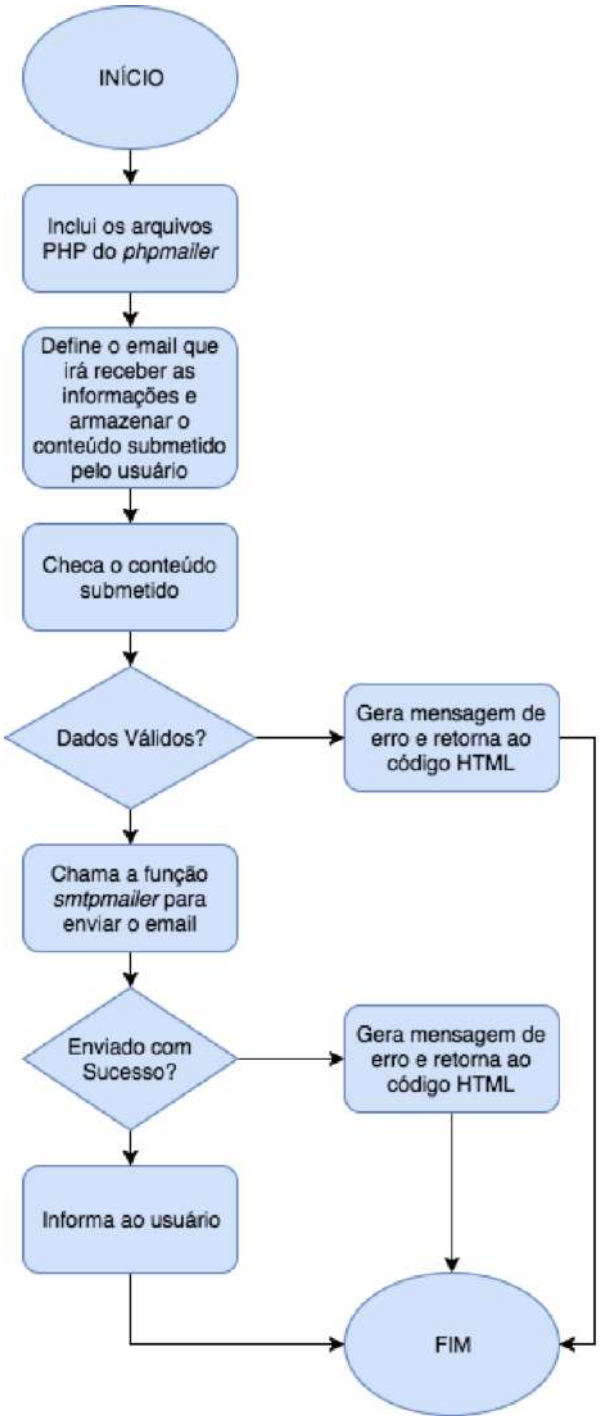
Com isso, todas as páginas do *website* foram explicadas, incluindo as particularidades de cada uma. Para mais informações é possível consultar os Anexos que contêm os códigos completos.

Figura 98 – *Fluxograma do Código HTML para a Página Fale Conosco*



Fonte: Produção do próprio autor.

Figura 99 – *Fluxograma do Código PHP para a Página Fale Conosco*



Fonte: Produção do próprio autor.

3.2 Desenvolvimento dos Aplicativos

3.2.1 Desenvolvimento do Aplicativo para iOS

O aplicativo infoAR para iOS é um aplicativo *Cocoa*, ou seja, foi criado com base em uma API orientada a objetos nativa do sistema operacional. Como já citado anteriormente, o arquivo *AppDelegate* roda antes do aplicativo abrir, e é nele que estão declaradas todas as bibliotecas, sendo também responsável pelas notificações, as quais vêm do *web service*. Esse código deve ser muito rápido para não afetar a experiência do usuário.

Além disso, cada janela do aplicativo precisa ter seu próprio *ViewController*, e, no *Main.storyboard* é possível acessar todos os *ViewControllers* e também definir a ordem em que aparecerão. Também é no *Main.storyboard*, que os elementos que vão aparecer na tela são definidos. Como não é possível programar em *xml* no sistema operacional iOS, é necessário definir *constraints* a fim de que todos os elementos apareçam na tela do celular de acordo com o que foi planejado.

Para utilizar a API *Cocoa* foi necessário acessar o *website cocoapods.org* que possui várias bibliotecas, as quais podem ser instaladas, facilitando o trabalho do programador. Por exemplo, para fazer a comunicação com o JSON, basta baixar a biblioteca *Alamofire* e fazer as devidas configurações. O Google Maps também foi definido utilizando a API *Cocoa*, e, para fazer isso, foi necessário utilizar o terminal do computador. Após instalar a biblioteca desejada, foi necessário localizar o arquivo do aplicativo e utilizar o comando *pod init* para gerar o *Podfile* (Figura 100).

Figura 100 – *Importar biblioteca*

```
import UIKit
import GoogleMaps
```

Fonte: Produção do próprio autor.

Após isso, a *ViewController* da página inicial foi gerada, porém, para que o mapa pudesse aparecer na tela, foi necessário mudar a classe da *view* no *Main.storyboard* para *GMSMapView*. Com isso, foi possível acessar a subclasse *GMSMarker* (Figura 101) para definir os *markers*, e, a fim de focar na região em que os *markers* estão localizados, foi necessário utilizar o comando *self.mapview*. Isso é necessário, pois pode haver mais de um mapa na mesma tela. Para acessar a localização do usuário, é necessário pedir permissão através do arquivo *info.plist* e também do *AppDelegate* (Figura 102), além disso, também é preciso conferir se foi possível acessar tal informação (Figura 103).

Figura 101 – *Utilização da Subclasse* GMSMarker

```
 var cont: Int = 0
for location in locations{
    let coordinate = CLLocationCoordinate2DMake(location[0] , location[1] )
     let marker = GMSMarker(position:coordinate )
     marker.map = mapView
     marker.title = names[cont];
     marker.icon = UIImage(named: "verde")
cont += 1;
}
```

Fonte: Produção do próprio autor.

Figura 102 – *Autorização para Acessar a Localização do Usuário*

```
func locationManager(_ manager: CLLocationManager, didChangeAuthorization status:
    CLAuthorizationStatus) {

    if status == .authorizedWhenInUse{
        locationManager.startUpdatingLocation()

        mapView.isMyLocationEnabled = true // do googlemaps
    }
}
```

Fonte: Produção do próprio autor.

Figura 103 – *Verificação da Localização*

```
func locationManager(_ manager: CLLocationManager, didUpdateLocations locations: [CLLocation]) {
    if let location = locations.first{
       usersLocation = location
        mapView.camera = GMSCameraPosition(target: usersLocation!.coordinate, zoom: 13, bearing:0,
            viewingAngle: 0)
        locationManager.stopUpdatingLocation()
    }
    else {
        let alert = UIAlertController(title: "Erro", message: "Localização não obtida",
            preferredStyle: .alert)
        let action = UIAlertAction(title: "OK", style: .default, handler: nil)

        alert.addAction(action)
        self.present(alert, animated: true, completion: nil)
    }
}
```

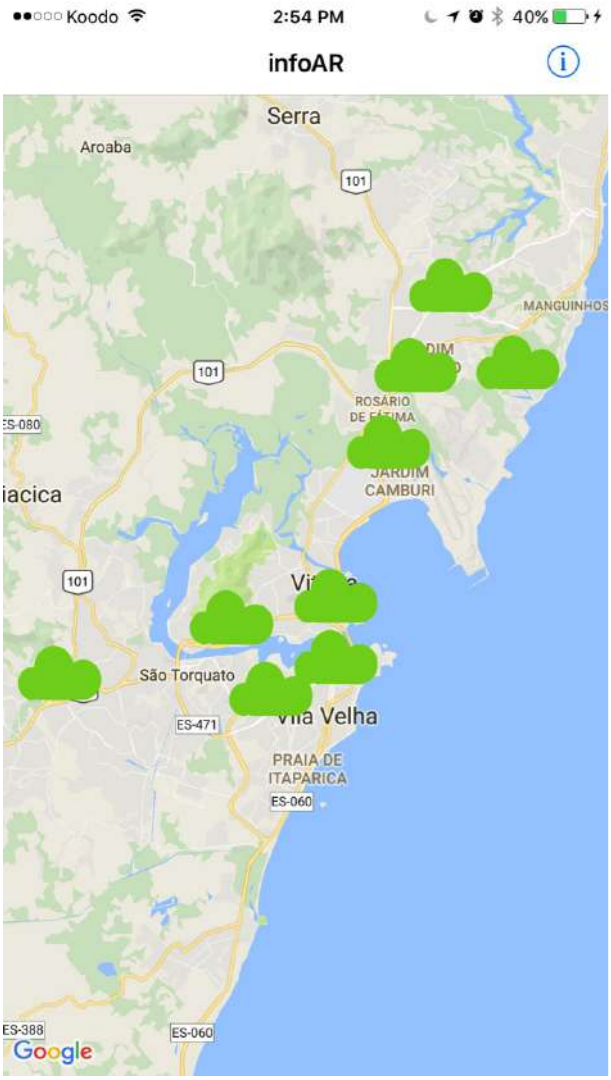
Fonte: Produção do próprio autor.

A função *locationManager* (Figura 102) é o manuseador de localização e cria uma variável para manipular essa informação. Essa função não é chamada sempre, mas sim quando o aplicativo é aberto para pedir permissão ao usuário para ter acesso à sua localização. O *locationManager* possui duas funções principais: verificar se o usuário autorizou o acesso a informação, e armazenar a localização. Para esta última função também é necessário utilizar o *CLLocationManager* (Figura 103). A atualização da localização só acontece se houver uma variação muito brusca da mesma, e, como várias localizações são recebidas em um pequeno intervalo de tempo, apenas a primeira é armazenada utilizando o comando

locations.first (Figura 103).

Ainda no *ViewController* do mapa há a função *viewDidLoad* que serve para carregar o mapa na tela. A Figura 104 mostra como o mapa aparece na tela do celular.

Figura 104 – *Google Maps no Aplicativo iOS*

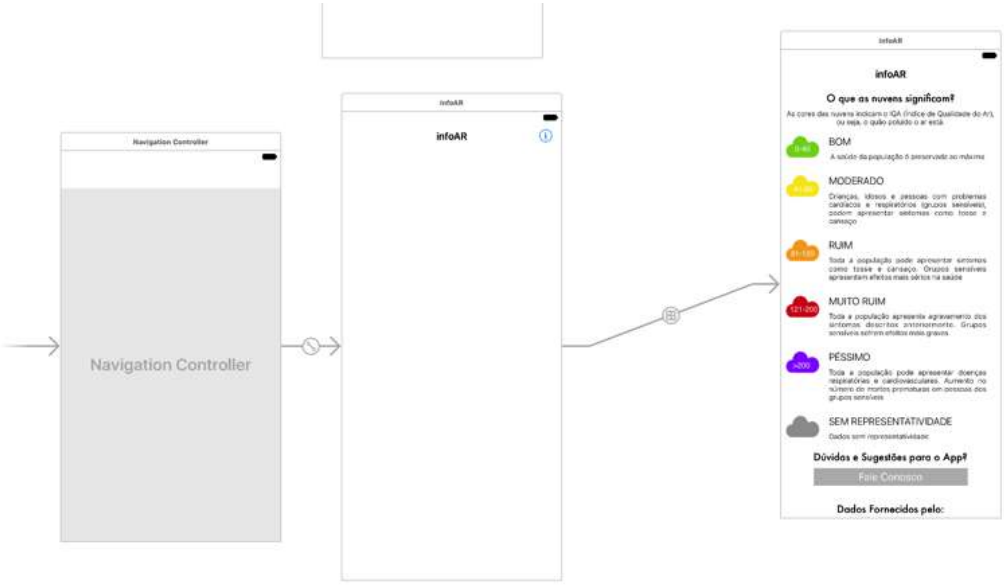


Fonte: Produção do próprio autor.

Após isso foi possível criar uma nova *ViewCotroller* para a página contendo as informações. A maneira como as *ViewControllers* se relacionam dentro do *Main.storyboard* é mostrada na Figura 105.

O *Navigation Controller* mostrado na Figura 105 não é visualizado pelo usuário, sendo apenas uma tela "mãe" responsável pela navegação e pelos comandos de *"push"*. No aplicativo infoAR há apenas um comando no canto superior direito na tela do Google Maps e que serve para acessar a tela de informação.

Figura 105 – Main.storyboard *com todas as* ViewControllers *do Aplicativo*



Fonte: Produção do próprio autor.

Como a página de informações é composta basicamente por elementos de imagem e texto, ela foi desenvolvida quase em toda sua totalidade dentro do *Main.storyboard* utilizando os comandos *TableViewController*, o qual gera uma tabela de elementos semelhantes, e o *ScrollView*, o qual possibilita ao usuário deslizar a tela para baixo e para cima e também para que essa página possa se adaptar aos diferentes tamanhos de tela.

As nuvens que povoam a tela de informação foram desenhadas pela autora utilizando o *software* Sketch e todas as imagens criadas são vetorizadas. Após a fase de desenvolvimento os elementos foram armazenados dentro da pasta contendo os códigos do aplicativo. Tanto as imagens quanto os textos foram inseridos na tela através do comando *drag* e utilizando os *constraints* para definir a localização exata. As Figuras 106 e 107 mostram como a tela de informação é visualizada pelo usuário.

Figura 106 – *Parte Superior da Tela de Informação*

Fonte: Produção do próprio autor.

Figura 107 – *Parte Inferior da Tela de Informação*

Fonte: Produção do próprio autor.

Ao final dessa tela foi criado um botão para que a página de email pudesse ser aberta a fim de que o usuário possa deixar sugestões. As funcionalidades desse botão foram definidos no arquivo *InfoViewController* (Figura 108).

Figura 108 – *Função para Abrir a Tela do Email*

```
@IBAction func faleconosco(_ sender: AnyObject) {
    let mailComposeViewController = configuredMailComposeViewController()
    if MFMailComposeViewController.canSendMail(){
        self.present(mailComposeViewController,animated: true, completion: nil)
    }else {
        self.showSendMailErrorAlert()
    }
}

func configuredMailComposeViewController() -> MFMailComposeViewController{
    let mailComposerVC = MFMailComposeViewController()
    mailComposerVC.mailComposeDelegate = self

    mailComposerVC.setToRecipients(["infoarapp@gmail.com"])
    mailComposerVC.setSubject("Mensagem do aplicativo iOS")
    mailComposerVC.setMessageBody("Dúvidas e Sugestões para o aplicativo:", isHTML: false)

    return mailComposerVC

}
```

Fonte: Produção do próprio autor.

A primeira função da Figura 108 (faleconosco) tem a função de chamar o *MailComposeViewController* e defini-lo como a tela principal; já a segunda função configura essa *ViewController* indicando o e-mail para o qual a sugestão deve ser enviada. As Figuras 109 e 110 mostram, respectivamente, a página que se abre a clicar no botão Fale Conosco e o email recebido em infoarapp@gmail.com.

Figura 109 – *Página para Enviar o Email no Aplicativo iOS*

Fonte: Produção do próprio autor.

Figura 110 – *E-mail enviado pelo Aplicativo iOS*

Fonte: Produção do próprio autor.

Por fim, foi necessário fazer a comunicação com o JSON para atualizar o IQA nos ícones. Como já citado anteriormente isso foi feito utilizando a biblioteca *Cocoa*, chamada de *Alamofire*, e utilizando a classe *Response JSON Handler*. Essa classe utiliza o *responseJSONSerializer* para transformar os dados armazenados no servidor em qualquer tipo de dado utilizando o *JSONSerialization.ReadingOptions* (Figura 111). Com isso é possível armazenar os dados e atualizar o IQA de cada estação.

Figura 111 – *Utilização da biblioteca Alamofire para armazenar os dados JSON*

```
Alamofire.request("https://httpbin.org/get").responseJSON { response in
debugPrint(response)

if let json = response.result.value {
print("JSON: \(json)")
}
```

Fonte: Produção do próprio autor.

Assim, todas as funcionalidades do aplicativo iOS foram explicadas; para mais informações, é possível consultar os códigos nos Anexos.

3.2.2 Desenvolvimento do Aplicativo Android

O aplicativo Android deve ter as mesmas funcionalidades que o aplicativo iOS. Assim, será composto de duas *Activities*, uma com o Google Maps e que mostrará as estações e seus respectivos IQAs, e outra com a página de informação. Além disso, o aplicativo fará uso de uma *Activity* já existente, que é o do aplicativo de e-mail.

A fim de utilizar o Google Maps, é preciso instalar o Google Play Services no aplicativo e ativar diversas permissões, como para o acesso da localização. Além disso, é preciso gerar uma API *Key* disponibilizada pelo Google, e dentro do aplicativo checar se o Play Service está disponível para uso. Com isso, é possível adicionar os *map fragments* e inicializar o mapa.

Para este trabalho optou-se por utilizar a versão 2 do Google Maps, apesar da versão 3 já estar disponível. Isso porque essa requer apenas a versão *4.2 Jelly Bean* do sistema operacional, assim, será mais acessível à maioria da população.

Após fazer o *download* do Google Services, foi preciso indicar no *build.gradle* que o Google Services deve ser compilado junto com o aplicativo. Com isso, é possível pedir as permissões no *AndroidManifest*. A primeira permissão é para que o aplicativo possa acessar a Internet e também saber qual o estado da rede, a fim de determinar se o Google Services pode ser

acessado ou não. Também é necessário, toda vez que o Google Maps é aberto, salvar os dados recebidos em um arquivo externo, e tal processo requer permissão (Figura 112).

Figura 112 – *Permissões no* AndroidManifest

```
<uses-permission android:name="android.permission.INTERNET"/> //permissao par
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/> //p
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/> /
```

Fonte: Produção do próprio autor.

Ainda no *AndroidManifest*, o programador criou uma permissão a fim de que só o aplicativo em desenvolvimento pudesse utilizar a API *Key* disponibilizada pelo Google, a qual é definida através da *tag meta-data*. Também foi necessário pedir permissão para ler as informações do Google Services, e, a fim de acessar o Google Maps, é preciso requerer tal característica (Figura 113).

Figura 113 – *Criação da Permissão e Requerimento do Acesso ao Google Maps*

```
<permission android:name="com.example.mayaraoliveira.permission.MAPS_RECEIVE"
            android:protectionLevel="signature" /> // cria autorizacao para o aplicativo receber o mapa

<uses-permission android:name="com.example.mayaraoliveira.permission.MAPS_RECEIVE"/> //pede permissao
<uses-permission android:name="com.google.android.providers.gsf.permissions.READ_GSERVICES"/>

<uses-feature android:glEsVersion="0x00020000"
            android:required="true"/> // utilizar o googlemaps na versao certa
```

Fonte: Produção do próprio autor.

Como o aplicativo utiliza o Play Services, é necessário verificar se o usuário tem o Google Play instalado do dispositivo, e, para isso foi utilizada o comando *GoogleApiAvailability.getInstance()* dentro do arquivo *MainActivity.java*, e, se o retorno for *true*, então o dispositivo possui o Play Services. Contudo, se o retorno for *false*, uma mensagem será mostrada na tela do dispositivo, informando ao usuário a necessidade de baixar o Play Services (Figura 114).

No *MainActivity.java* é preciso criar um objeto para receber o Google Maps, e para isso é preciso adicionar um *fragment* dentro do arquivo *activity_main.xml* (Figura 115).

Finalmente, o objeto do tipo GoogleMap foi criado, sendo apenas necessário uma função chamada de *initMap()* para inicializar o mapa. Dentro dessa função foi preciso acessar o fragmento do mapa e relacioná-lo ao objeto Google Maps. Após essa etapa foi possível especificar os *markers* para cada estação utilizando o comando *addMarker* e também modificar o *zoom* do mapa (Figura 116).

Figura 114 – *Verificação do Play Services*

```
public boolean googleServicesAvailable(){
    GoogleApiAvailability api = GoogleApiAvailability.getInstance();
    int isAvailable = api.isGooglePlayServicesAvailable(this);
    if(isAvailable == ConnectionResult.SUCCESS){
       return true;
    } else if(api.isUserResolvableError(isAvailable)){
        Dialog dialog = api.getErrorDialog(this,isAvailable,0);
        dialog.show();
    } else {
        Toast.makeText(this, "Não foi possível se conectar com o play services", Toast.LENGTH_LONG).show();
    }
    return false;
}
```

Fonte: Produção do próprio autor.

Figura 115 – *Declaração do Fragmento do Mapa*

```
<fragment
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:name="com.google.android.gms.maps.MapFragment"
    android:id="@+id/mapFragment"
    android:layout_centerHorizontal="true"
    android:layout_alignParentTop="true" />
```

Fonte: Produção do próprio autor.

Figura 116 – *Definição das Posições das Estações*

```
// Add a marker in Sydney and move the camera
LatLng laranjeiras = new LatLng(-20.196, -40.245);
LatLng carapina = new LatLng(-20.226, -40.259);
LatLng jcamburi = new LatLng(-20.255, -40.270);
LatLng enseada = new LatLng(-20.313, -40.291);
LatLng centrovix = new LatLng(-20.321, -40.333);
LatLng ibes = new LatLng(-20.348, -40.317);
LatLng vvcentro = new LatLng(-20.336, -40.291);
LatLng cariacica = new LatLng(-20.342, -40.402);
LatLng continental = new LatLng(-20.225, -40.218);

mMap.addMarker(new MarkerOptions().position(laranjeiras).title("Laranjeiras"));
mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(laranjeiras,12));
```

Fonte: Produção do próprio autor.

O próximo passo foi mostrar a localização do usuário no mapa, e, para isso foi necessário adicionar mais duas permissões no *AndroidManifest* (Figura 117): uma para ter acesso à localização precisa e outra para ter acesso a uma localização não tão exata assim.

Figura 117 – *Permissão para Acessar a Localização do Usuário*

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

Fonte: Produção do próprio autor.

Assim, utilizando o comando *setMyLocationEnabled()* no *MainActivity*, é possível obter a localização do usuário. Contudo, em algumas versões do Android as permissões definidas no *AndroidManifest* não são suficientes, sendo necessário utilizar a função *checkSelfPermission* que já é nativa do Android (Figura 118).

Figura 118 – *Acessar a Localização do Usuário*

```
if (ActivityCompat.checkSelfPermission(this, android.Manifest.permission.ACCESS_FINE_LOCATION)
        != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
        android.Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {

    return;
}
mGoogleMap.setMyLocationEnabled(true);
```

Fonte: Produção do próprio autor.

Assim, foi criada a primeira *Activity* do aplicativo na plataforma Android, a qual pode ser vista na Figura 119.

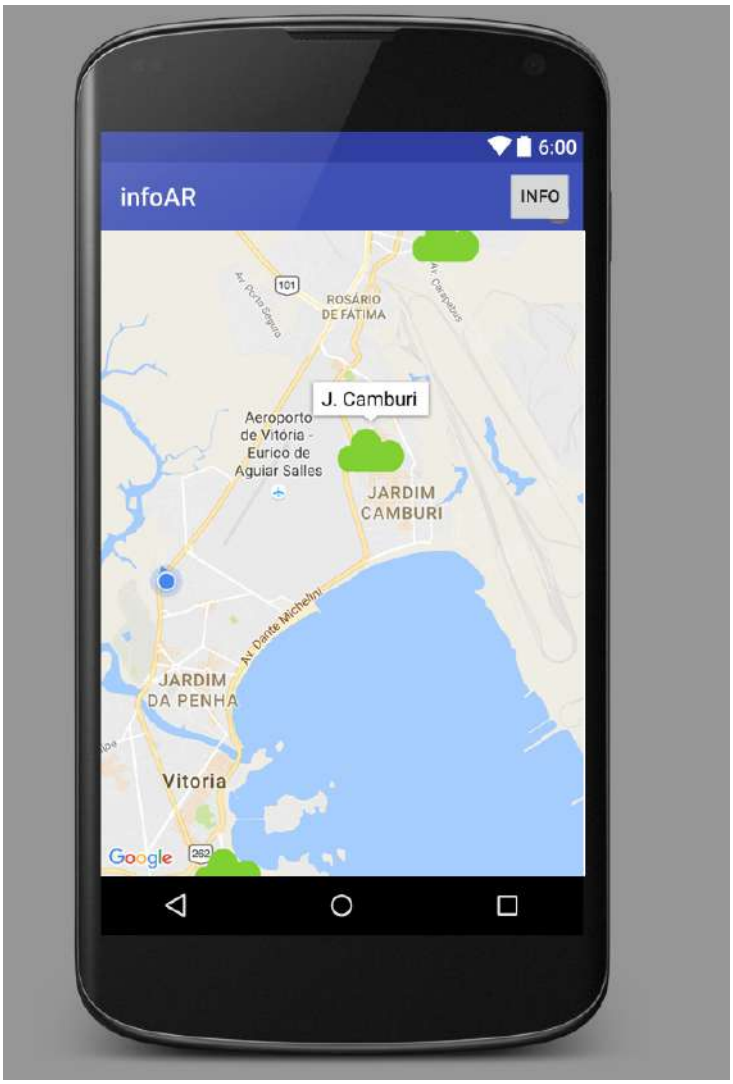
Posteriormente foi criada a segunda *Activity*, e, para tanto, foi necessário criar um novo arquivo *.xml* e um novo arquivo em Java. Para que o usuário pudesse ter acesso à página das informações, foi necessário criar um botão no arquivo *activity_main.xml* apenas utilizando o comando *drag*. Assim, foi possível povoar a página de informações da mesma maneira que foi feito no aplicativo iOS.

Porém, diferente da programação para iOS, em que é possível utilizar os *constraints* para que os elementos se adaptem aos vários tipos de tela, nos aplicativos Android o programador é responsável por carregar imagens em diversos tamanhos para assegurar que o usuário terá acesso ao conteúdo planejado, sendo essa uma das grandes dificuldades da programação para Android.

No documento Java da segunda *Activity*, para que o conteúdo apareça, foi necessário utilizar o comando *extends* para transformar o arquivo de fato em uma *Activity*, além disso, também foi necessário desenvolver a função *onCreate()* para que o conteúdo fosse visível ao usuário (Figura 120).

Também foi preciso declarar a nova *Activity* dentro do arquivo *AndroidManifest*, e, no

Figura 119 – *Primeira* Activity *do Aplicativo na Plataforma Android*



Fonte: Produção do próprio autor.

Figura 120 – *Configuração do Arquivo Java para a Segunda* Activity

```
public class display extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.isplay);

}
```

Fonte: Produção do próprio autor.

arquivo *activity_main*, foi necessário descrever o que acontecerá quando o botão for apertado. Para tanto, foi criada uma função que manipula um objeto da classe *Intent* a fim de que a nova *Activity* seja chamada (Figura 121). Essa nova *Activity* é mostrada na Figura 122.

Figura 121 – *Definir a Ação do Botão de Acesso a Página de Informações*

```
public void onButtonClick(View v)
{
    if(v.getId() == R.id.ibutton)
    {
        Intent i = new Intent(MainActivity.this, display.class);
        startActivity(i);
    }
}
```

Fonte: Produção do próprio autor.

Figura 122 – Activity *Contendo as Informações*

Fonte: Produção do próprio autor.

Na *Activity* contendo as informações, além de elementos de imagem e texto, há também o botão "Fale Conosco"para que seja possível receber o *feedback* do usuário. Para isso, o aplicativo infoAR irá chamar o aplicativo de email nativo do sistema operacional Android.

Por conseguinte, dentro do arquivo Java da segunda *Activity*, foram criadas duas funções: uma para verificar se o botão foi pressionado e outra para chamar o aplicativo de e-mail do Android. A fim de enviar o e-mail, foi necessário criar um objeto da classe *Intent*, o qual carregará todo o conteúdo digitado para o aplicativo de e-mail. Também foi necessário selecionar a ação de enviar o e-mail e definir o destinatário, além de informar o assunto e o título da mensagem de email (Figura 123).

Figura 123 – *Função para Enviar o Email*

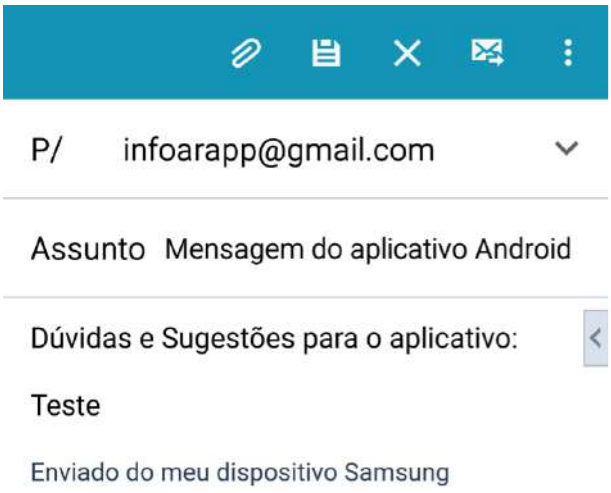
```
protected void sendEmail() {
    Log.i("Enviar email", "");
    String[] TO = {"infoarapp@gmail.com"};
    String[] CC = {""};
    Intent emailIntent = new Intent(Intent.ACTION_SEND);

    emailIntent.setData(Uri.parse("mailto:"));
    emailIntent.setType("text/plain");
    emailIntent.putExtra(Intent.EXTRA_EMAIL, TO);
    emailIntent.putExtra(Intent.EXTRA_CC, CC);
    emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Mensagem do aplicativo Android");
    emailIntent.putExtra(Intent.EXTRA_TEXT, "Dúvidas e Sugestões para o aplicativo:");
```

Fonte: Produção do próprio autor.

As Figuras 124 e 125 mostram, respectivamente, o aplicativo aberto ao clicar no botão de Fale Conosco e o e-mail recebido em infoarapp@gmail.com.

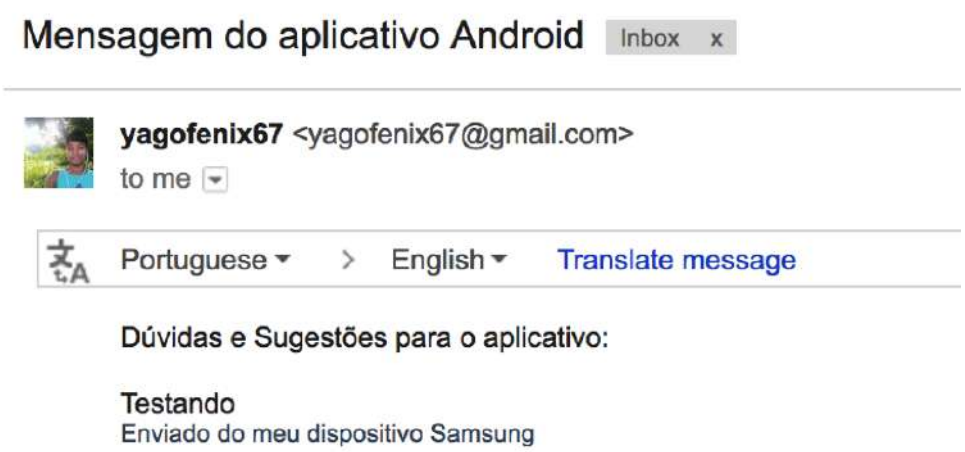
Para finalizar o aplicativo Android, ainda é preciso fazer a conexão com o JSON, utilizando a classe *JSONObject*, atividade que será desenvolvida futuramente.

Figura 124 – *Aplicativo do E-mail aberto pelo infoAR*



Fonte: Produção do próprio autor.

Figura 125 – *E-mail Recebido*



Fonte: Produção do próprio autor.

4 RESULTADOS E CONCLUSÕES

O estudo realizado ao longo deste projeto de graduação mostrou que o desenvolvimento de um *website* e de um aplicativo para transmitir, de maneira transparente, à população as concentrações dos poluentes em um determinado local da Grande Vitória e em um determinado horário do dia, é viável.

A importância do desenvolvimento de um aplicativo nesse contexto foi importante porque dados apresentados no decorrer do trabalho mostraram, que, pela primeira vez, o número de brasileiros que acessam a Internet pelo celular é maior do que os número de brasileiros que o fazem através do computador. O *website*, portanto, veio para complementar as informações já disponibilizadas pelos aplicativos, oferecendo ao usuário a possibilidade de adquirir um pouco mais de conhecimento sobre o assunto.

Contudo, o desenvolvimento do *website* e do aplicativo em duas plataformas diferentes se tornou um desafio. O primeiro desafio foi desenvolver o conhecimento sobre qualidade do ar, a fim de que fosse possível desenvolver um projeto com informações consistentes e corretas. Isso só foi possível com a ajuda da coorientadora Jane Meri dos Santos, do Departamento de Engenharia Ambiental, que, além de ser uma das autoras do Relatório da Qualidade do Ar utilizado no desenvolvimento do trabalho, também foi responsável pela criação do método de validação (médias horárias e representatividade) e da cartilha que relaciona o IQA aos efeitos adversos à saúde, baseando-se na diretriz da OMS. O processo de estudo da qualidade do ar perdurou inicialmente por cerca de um mês, porém, durante todo o desenvolvimento do trabalho as dúvidas em relação ao cálculo do IQA foram sanadas pela coorientadora.

Na proposta inicial do trabalho, o qual é um parceria entre os Departamentos de Engenharia Elétrica e Engenharia Ambiental da UFES com o IEMA, ficara definido que o IEMA seria responsável por disponibilizar os computadores para o desenvolvimento do trabalho, além do *nobreak*, verba para contratar um *Web designer* e o acesso ao banco de dados e servidor. Contudo, por questões burocráticas, a aluna não teve acesso a nenhum desses recursos. A ajuda proveniente do IEMA veio dos funcionários do TI, que foram sempre muito solícitos, mas que não detinham a autoridade necessária para solucionar os problemas.

Assim, a aluna iniciou o desenvolvimento do *website*, e, esse processo se tornou mais trabalhoso do que o esperado, já que não haveria a possibilidade de contratar um *Web designer*. Além disso, a fim de desenvolver as páginas Web dinâmicas, era necessário o acesso a um servidor, e, após uma espera de quase seis meses para que isso fosse disponibilizado pelo IEMA, a aluna conseguiu que o site fosse hospedado no servidor da UFES, além de

também conseguir o acesso ao banco de dados. No fim, o *website* foi hospedado no domínio *www.infoar.ufes.br*, o qual está disponível para acesso, e, por ser um domínio institucional, gera mais credibilidade ao projeto.

O desenvolvimento do aplicativo para as plataformas iOS e Android também foi um desafio, apesar de haver extensa documentação sobre o assunto. Isso porque a aluna não dominava as linguagens de programação, e, além disso, os compiladores são atualizados e, consequentemente, muda-se a declaração de classes e funções importantes. Ainda, para o aplicativo rodar na plataforma Android, há necessidade de se adequar as diversas versões es os diferentes formatos de tela tornou a programação ainda mais trabalhosa.

Infelizmente o objetivo principal deste trabalho não foi fatalmente alcançado, já que não foi possível acessar o banco de dados do IEMA, o qual armazena os dados em tempo real. Assim, não foi possível disponibilizar os aplicativos nas lojas já que, para tanto, seria necessário pagar US$100.00 para a AppStore e US$25.00 para a GooglePlay, o que, não faria sentido sem os dados atualizados. Mesmo assim, a simulação realizada pela aluna mostra que é possível proporcionar esse serviço à população capixaba.

Além disso, o desenvolvimento do trabalho proporcionou a aluna a ampliação do conhecimento em diversas linguagens de programação, como HTML, CSS, PHP, *Swift* e *Objective-C*, além da manipulação de banco de dados. A aluna também desenvolveu características como automotivação, humildade, e aprendeu sobre a importância do *network* no desenvolvimento de qualquer trabalho.

Enfim, apesar do trabalho ainda precisar ser finalizado, um longo caminho já foi traçado, a fim de disponibilizar informações transparentes à população e que culmine com a melhora da qualidade de vida no Estado do Espírito Santo.

5 TRABALHOS FUTUROS

Como sugestões de trabalhos futuros, citam-se:

- Conseguir acesso ao banco de dados do IEMA para que seja possível atualizar o *website* de hora e hora e divulgar o resultado deste trabalho para a população em geral. Como já citado no decorrer do trabalho, isso será feito utilizando o *CronTable*;
- Verificar a representatividade dos dados provenientes das estações;
- Conectar o aplicativo Android com o JSON;
- Testar o aplicativo na plataforma Android em diferentes dispositivos para certificar que este funciona corretamente;
- Disponibilizar os aplicativos nas lojas AppStore e GooglePlay;
- Fazer uma validação de pelo menos dois meses com os usuários, e utilizar o *feedback* para corrigir falhas e *bugs*, além de desenvolver novas versões que deverão ser disponibilizadas nas lojas;
- Realizar parceria com outro aluno do Departamento de Engenharia Ambiental para que seja possível desenvolver um algoritmo que irá prever a qualidade do ar em um espaço de dois dias, assim como já é feito pelo *website AirText* da cidade de Londres, o qual foi uma das inspirações para este projeto de graduação.

6 REFERÊNCIAS BIBLIOGRÁFICAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023**: Informação e documentação – referências – elaboração. Rio de Janeiro, 2002a.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6024**: Informação e documentação – numeração progressiva das seções de um documento escrito – apresentação. Rio de Janeiro, 2003a.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6027**: Informação e documentação – sumário – apresentação. Rio de Janeiro, 2003b.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6028**: Informação e documentação – resumo – apresentação. Rio de Janeiro, 2003c.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 10520**: Informação e documentação – citações em documentos – apresentação. Rio de Janeiro, 2002b.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 14724**: Informação e documentação – trabalhos acadêmicos – apresentação. Rio de Janeiro, 2011.

ESPÍRITO SANTO. Instituto Estadual de Meio Ambiente. **Relatório da qualidade do ar da Grande Vitória 2013**. Vitória, 2014.

CERC. **AirText air quality**. Disponível em: < http://www.airtext.info/> . Acesso em 20 de março de 2016.

SISTEMA NACIONAL DE CALIDAD DEL AIRE EN CHILE. **Real time air quality index (AQI)**. Disponível em: < http://aqicn.org/city/chile/las-condes/> . Acesso em 04 de abril de 2016.

US EMBASSY. **Beijing air quality**. Disponível em: < http://aqicn.org/city/beijing/us-embassy/>. Acesso em 04 de abril de 2016.

GOVERNO DO ESTADO DE SÃO PAULO. **Aplicativo qualidade do ar**. Disponível em: < http://www.saopaulo.sp.gov.br/spnoticias/lenoticia2.php?id=245532> . Acesso em 16 de abril de 2016.

CAELUM. **Apostila do Curso WD-43 – Desenvolvimento web com HTML,**

CSS e JavaScript. Disponível em: < https://www.caelum.com.br/apostila-html-css-javascript/> . Acesso em 13 de maio de 2016.

MAZZA, Lucas. **HTML5 e CSS3: Domine a web do futuro**. 1 ed. São Paulo: Casa do Código, 2012.

LOPES, Sérgio. **A web mobile: programe para um mundo de muitos dispositivos**. 1 ed. São Paulo: Casa do Código, 2013.

ADAMS, C. ; BOLTON, J. ; JOHNSON, D. ; STEV. **A arte e a ciência da CSS**. 1 ed. Rio de Janeiro: Bookman, 2009.

CRAIG, C. ; GARBER, J. **Foundation HTML5 with CSS3**. 1 ed. Berkley: Apress, 2012.

FIRDAUS, T. ; FRAIN, B. ; LAGRONE, B. **HTML and CSS3: Building Responsive Websites**. 1 ed: Packt Publishing, 2016.

VALADE, Janet. **PHP and MySQL: Your Visual Blueprint for creating dynamics, database-driven Websites**. 1ed: Visual, 2006

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Estatísticas do Uso de Celular no Brasil**. disponível em: < http://agenciabrasil.ebc.com.br/economia/noticia/2016-04/celular-e-principal-meio-de-acesso-internet-na-maioria-dos-lares>. Acesso em 20 de janeiro de 2017.

RODRIGUES, JOEL. **Modelo Entidade Relacionamento (MER) e Diagrama Entidade Relacionamento (DER)**. Disponível em: < http://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332 >. Acesso em 21 de janeiro de 2017.

APPLE. **App Programming Guide for iOS**. Disponível em: < https://developer.apple.com/library/ >. Acesso em 22 de janeiro de 2017.

APPLE. **The Powerful Programming Language: Swift**. Disponível em: < https://developer.apple . Acesso em 22 de janeiro de 2017.

ANDROID. **Developers Guide**. Disponível em: < https://developer.android.com/guide/index.html >. Acesso em 23 de janeiro de 2017.

GLOBO. **Estudo mostra que poluição do ar matou mais de quatro milhões de pessoas em 2015**. Disponível em: < http://g1.globo.com/jornal-hoje/videos/t/edicoes/v/estudo-mostra-que-poluicao-do-ar-matou-mais-de-quatro-milhoes-de-pessoas-em-2015/5655283/ >. Acesso em 18 de fevereiro de 2017.

Anexos

ANEXO A – CÓDIGOS DO WEBSITE

A.1 Código CSS Utilizado em Todas as Páginas

```
1
2  @import url(https://fonts.googleapis.com/css?family=Hind);
3  @import url(https://fonts.googleapis.com/css?family=Nobile);
4  * {
5
6          margin:0; /* para acabar com os espacos éat as margens*/
7          border:0;
8          padding:0;
9  }
10
11
12 head{
13         background:#87CEEB ;
14 }
15
16 body{
17
18         background-color: #FFF;
19         font-family:'Hind', sans-serif;
20         font-size: 18px;
21         position: relative;
22 }
23
24 h1{
25
26         font-family:'Nobile', sans-serif;
27         text-align: center;
28         font-size: 175%;
29         color: #4A4444;
30         text-transform: uppercase;
31         letter-spacing: 3px;
32         padding:3% 0; /*top&bottom left&right*/
33
34 }
35
```

```
36 h3{
37
38         font-family:'Nobile', sans-serif;
39         text-align: center;
40         color: #4A4444;
41         text-transform: uppercase;
42         letter-spacing: 1%;
43         margin-bottom:5%;
44 }
45
46 h4{
47         font-size:0.6;
48         font-family:'Nobile', sans-serif;
49         text-align: center;
50         color: #4A4444;
51         text-transform: uppercase;
52         letter-spacing: 1%;
53         margin:0;
54         padding:0;
55 }
56
57 p{
58
59         padding: 2%;
60         color:#4A4444;
61         text-align: justify, center;
62 }
63
64 img{
65
66         max-width:100%;
67         height:auto;
68 }
69
70 #banner-wrapper{
71         max-width:1280px;
72         margin:0 auto;
73 }
74
75 /*---Start Image Slider Style---*/
76 .slider{
77
```

```
78          width:100%;
79  }
80
81  .slider1 img{
82          min-width:100%;
83          margin:0 auto;
84  }
85
86  /*colocando as setas nas fotos*/
87  .slider .bx-wrapper .bx-controls-direction a {
88
89          outline: 0 none;
90          position: absolute;
91          text-indent: -9999px;
92          top:40%;
93          height:71px;
94          width: 40px;
95          z-index: -1; /*abaixo da imagem*/
96          transition: all 0.7s; /*muda a imagem quando coloca em
               cima*/
97  }
98
99  .map{
100         position: absolute;
101         top: 0;
102         left: 0;
103         width: 100% ;
104         height: 100% ;
105 }
106 /*.slider .bx-wrapper:hover .bx-controls-direction a {
107
108         z-index:5;
109 }
110
111 /*.slider .bx-wrapper .bx-prev{
112         background: #000 url("img/left-arrow-white.png") no
               repeat 8px 13px;
113         left:0px;
114         opacity:0.3;
115 }
116
117 .slider .bx-wrapper .bx-prev:hover{
```

```
118         opacity:0.6;
119 }
120
121 .slider .bx-wrapper .bx-next{
122         background: #000 url("img/right-arrow-white.png") no
               repeat 10px 12px;
123         right:10px;
124         opacity:0.3;
125 }
126
127 .slider .bx-wrapper .bx-next:hover{
128         opacity:0.6;
129 }*/
130
131 /*---End Image Slider Style---*/
132
133 .one-half{
134
135         width:44%; /*ãno é 50% por causa da margem*/
136         float: left;
137         margin: 2% 0 3% 4%; /* top/right/bottom/left*/
138         text-align: center;
139 }
140
141 .one-third{
142         width:28%;
143         float:left;
144         margin: 2% 0 3% 4%;
145         text-align: center;
146 }
147
148 .left-col{
149
150         width:60%;
151         float: left;
152         margin:4% 0% 4% 4%;
153 }
154
155 .sidebar{
156         width: 26%;
157         float: right;
158         margin: 4% 4%;
```

```
159 }
160
161 .sidebar img {
162         opacity: 0.7; /*imagem mais transparente*/
163 }
164
165 /* -- Start Parallax Section --*/
166
167 .clearfix{ /*indicar onde um termina e o outro comeca*/
168         clear:both;
169 }
170
171 .parallax-1{
172         margin-top:5%;
173         background: url("img/parallax.png") repeat fixed 100%; /*
              mantenha display 100% da tela*/
174         text-align:center;
175 }
176
177 .parallax-inner{ /*mostra as bordas*/
178         padding-top:90px;
179         padding-bottom:300px;
180 }
181
182 /*parallax-2{
183         background: url("https://www.w3newbie.com/wp-content/
              uploads/black-pattern.jpg") repeat fixed 100%; /*
              mantenha display 100% da tela*/
184         /*text-align:center;*/
185
186
187 .parallax-inner h3, .parallax-inner p{
188         color:#FFFFFF; /*um pouco mais escuro do que o branco
              regular*/
189         text-shadow: -1px 0 black, 0 1px black, 1px 0 black, 0 -1
              px black;
190         margin-top: 5%;
191
192 }
193
194 /* -- End Parallax Section --*/
195
```

```
196 footer{
197 
198         background: #87CEEB;
199         opacity:0.9;
200         width: 100%;
201         margin-top: 5%;
202         padding:1% 0;
203         overflow: auto;
204 
205 }
206 
207 footer p{
208                 color: #FFF;
209 }
210 
211 table{
212         border-collapse: collapse;
213         width: 80%;
214         margin-left: 10%;
215         margin-right: 10%;
216         margin-top: 5%;
217         margin-bottom: 5%;
218 }
219 
220 table, th, td {
221         border: 1px solid black;
222 
223 }
224 
225 th,td{
226         text-align: center;
227 }
228 
229 th{
230         font-family: "Nobile";
231         text-transform: uppercase;
232         height: 50px;
233         background-color: #87CEEB;
234         color:#FFF;
235 }
236 
237 .form{
```

```
238         font-family:"Nobile", sans-serif;
239         text-align: center;
240         margin-top: 5%;
241         margin-bottom: 5%;
242         color:#999;
243 }
244
245 input{
246         height: 40px;
247         width:100px;
248         border: 1px solid #E5E5E5;
249         color: #999;
250         box-shadow: rgba(0, 0, 0, 0.1) 0px 0px 8px;
251         -moz-box-shadow: rgba(0, 0, 0, 0.1) 0px 0px 8px;
252         -webkit-box-shadow: rgba(0, 0, 0, 0.1) 0px 0px 8px;
253         border-radius: 3px;
254         -moz-border-radius: 3px;
255         -webkit-border-radius: 3px;
256 }
257
258 textarea{
259         height: 150px;
260         width:400px;
261         margin-left: 5%;
262         border: 1px solid #E5E5E5;
263         color: #999;
264         box-shadow: rgba(0, 0, 0, 0.1) 0px 0px 8px;
265         -moz-box-shadow: rgba(0, 0, 0, 0.1) 0px 0px 8px;
266         -webkit-box-shadow: rgba(0, 0, 0, 0.1) 0px 0px 8px;
267         max-width:400px;
268         line-height: 15px;
269         border-radius: 3px;
270         -moz-border-radius: 3px;
271         -webkit-border-radius: 3px;
272 }
273
274 input:hover, textarea:hover, select:hover,
275 input:focus, textarea:focus, select:focus{
276         border-color: 1px solid #C9C9C9;
277         box-shadow: rgba(0, 0, 0, 0.2) 0px 0px 8px;
278         -moz-box-shadow: rgba(0, 0, 0, 0.2) 0px 0px 8px;
279         -webkit-box-shadow: rgba(0, 0, 0, 0.2) 0px 0px 8px;
```

```
280
281 }
282
283 .submit input{
284         font-family: "Nobile", sans-serif;
285         text-transform: uppercase;
286         float: center
287         width:100px;
288         height:40px
289         background:#87CEEB;
290         color:#FFF;
291         border-radius: 3px;
292         -moz-border-radius: 3px;
293         -webkit-border-radius: 3px;
294 }
295
296 .birthday input{
297         height: 40px;
298 }
299
300 select{
301         height: 40px;
302 }
303
304 .labels{
305         color:#FFF;
306 }
307
308 .icons{
309         width:44%; /*ãno é 50% por causa da margem*/
310         float: left;
311         margin: 2% 0 3% 4%; /* top/right/bottom/left*/
312         text-align: center;
313  }
314
315 /*---MEDIA QUERIES--*/
316
317 @media screen and (max-width: 1080px) {
318
319         .slider .bx-wrapper .bx-controls{
320                 display: none;
321         }
```

```
322         .parallax-inner{
323                 display: none;
324         }
325
326         .one-third{
327                 width:100%;
328                 margin: 4% 0;
329                 float:center;
330         }
331
332         .one-half{
333                 display: none;
334         }
335
336         h1{
337                 font-size: 125%;
338                 float:center;
339         }
340
341         .left-col{
342                 width:50%;
343                 margin: 0 0 3% 0;
344         }
345
346         .sidebar{
347                 width:50%;
348                 margin:0;
349         }
350
351         img{
352                 width: 50%;
353         }
354
355         h3{
356                 padding-top: 3%;
357                 float: center;
358         }
359
360         textarea{
361                 width: 100%;
362                 margin: 4% 0;
363         }
```

```
364
365 table {width:%; float: center;}
366 thead {display: none;}
367 tr:nth-of-type(2n) {background-color: inherit;}
368 tr td:first-child {background: #f0f0f0; font-weight:bold;font-
    size:1.3em;}
369 tbody td {display: block;  text-align:center;}
370 tbody td:before {
371     content: attr(data-th);
372     display: block;
373     text-align:center;
374   }
375 }
376 }
```

A.1.1 Código CSS para o *Navigation Bar* - Foundation *Framework*

```
1
2 @import url(https://fonts.googleapis.com/css?family=Questrial);
3
4 .nav-outer {
5         width: 100%;
6         height: 67px;
7         background: #87CEEB;
8         position: fixed !important;
9         z-index: 100000000;
10 }
11 .nav-wrap {
12         max-width: 1120px;
13         margin: 0 auto;
14 }
15 nav {
16         margin-top: 0;
17         background: #87CEEB;
18         font-family: 'Questrial', sans-serif;
19 }
20 .nav ul {
21         overflow:hidden;
22         list-style:none;
23 }
24 .nav-button:hover {
25         cursor:pointer;
```

```
}
.navigation {
        clear: both;
        width: 100%;
        position: relative;
}
.nav a {
        color: #F5F5F5;
        text-transform: uppercase;
        text-decoration:none;
}
body .nav .nav-menu li a {
        display:inline-block;
        margin-top: 10px;
        padding:15px 20px;
        color: #F5F5F5;
        font-size: 19px;
}
.nav.yoga .nav-menu li.active a {
        color: #F5F5F5;
        text-decoration: underline;
}
.nav.yoga .nav-menu li a:hover {
        color: #F5F5F5;
        text-decoration: underline;
}
.nav.yoga .nav-toggled {
        min-height:36px;
        border-radius:6px;
        margin-top: -7px;
}
.nav.yoga .nav-toggled-controls {
        display:block;
        height:40px;
        text-align:left;
        position:relative;
}
.nav.yoga .nav-toggled-title {
        position:relative;
        top:9px;
        left:15px;
        font-size:16px;
```

```
68 }
69 .nav.yoga .nav-button {
70         display:block;
71         position:absolute;
72         right:15px;
73         top:8px;
74 }
75 .nav.yoga .nav-button span {
76         display:block;
77         margin-top:4px;
78         height:2px;
79         background: #FFF;
80         width:24px;
81 }
82 .nav-toggled-controls{
83         border-bottom: 0px solid #FFF;
84 }
85 .nav.yoga .nav-toggled ul li a {
86         display:block;
87         width:100%;
88         background-color: #87CEEB;
89         text-align:left;
90         padding:10px 0px 10px 15px;
91         border-bottom:1px solid #FFF;
92         text-align: center;
93         font-size: 16px;
94         -webkit-box-sizing: border-box;
95         -moz-box-sizing: border-box;
96         box-sizing: border-box;
97 }
98 .nav.yoga .nav-toggled ul li ul a {
99         font-size: 15px;
100        text-transform: none;
101 }
102 .nav.yoga .nav-toggled ul li.active a {
103        background-color: #F5F5F5;
104        color: #505E67;
105 }
106 .nav.yoga .nav-toggled ul li a:hover {
107        background: #F5F5F5;
108        color: #505E67;
109 }
```

```
.nav.yoga .nav-toggled ul li {
        position: relative;
}
.toggle-sm:after {
        position: absolute;
        right: 15px;
        top: 10px;
        font-size: 18px;
        line-height: 25px;
}
.toggle-sm:after {
        content: '+';
        cursor: pointer;
}
.toggle-sm.open:after {
        content: '-';
        cursor: pointer;
}
.nav .nav-menu {
        text-align: center;
        overflow: visible;
        min-height: 56px;
        padding-left: 0;
        margin: 0;
}
@media (min-width: 823px) {
        .nav {
                float: right;
        }
        .logo {
        float: left;
        }
}
@media screen and (max-width: 822px) {
        body .nav .nav-menu li a {
                margin-top: 4px;
        }
        .logo img {
                max-width: 135px !important;
                margin-bottom: 3px;
        }
        .nav-outer {
```

```
152                 height: 21px;
153         }
154         .nav.yoga .nav-toggled-controls {
155                 top: -40px;
156                 margin-bottom: -36px;
157         }
158         .nav-clear {
159                 padding-top: 37px;
160                 margin-bottom: -15px;
161         }
162 }
163 .logo img {
164         max-width: 175px;
165         height: auto;
166         margin-top: 7px;
167         margin-left: 30px;
168         margin-bottom: 5px;
169 }
170 .navigation:after {
171         content: "";
172         display: table;
173         clear: both;
174 }
175 .nav .nav-menu li {
176         display:inline-block;
177         padding:0px;
178         margin:0px !important;
179         position: relative;
180 }
181 .nav-button:hover {
182         cursor:pointer;
183 }
184 .nav .nav-toggled ul {
185         display:none;
186         margin:0px ;
187         padding:0px ;
188         position: relative;
189 }
190 .nav .nav-menu > li > ul {
191         position: absolute;
192         z-index: 10000000000;
193         top: 50px;
```

```
194         text-align: left;
195         background: #505E67;
196         width: 100%;
197         padding-left: 0;
198         display: none;
199 }
200 .nav.yoga .nav-menu > li > ul a {
201         padding: 11px 15px;
202         font-size: 18px;
203         font-weight: normal;
204         text-transform: none;
205 }
206 .nav .nav-menu > li > ul  li {
207         position: relative;
208         width: 100%;
209         text-align: center;
210 }
211 .nav .nav-menu > li > ul > li  ul {
212         position: absolute;
213         right: -100%;
214         top: 0;
215         width: 100%;
216         padding-left: 0;
217         background: #505E67;
218 }
219 .nav .nav-menu > li > ul > li  ul a {
220         white-space: nowrap;
221 }
222 .nav ul {
223         overflow: visible;
224 }
225 .has-children {
226         position: relative;
227 }
228 .has-children:after {
229         content: '+';
230         color: #FFF;
231         position: absolute;
232         right: 8px;
233         top: 50%;
234         transform: translateY(-50%);
235         -webkit-transform: translateY(-50%);
```

```
236         -moz-transform: translateY(-50%);
237         -o-transform: translateY(-50%);
238         -ms-transform: translateY(-50%);
239         cursor: pointer;
240 }
241 .has-children:hover:after {
242         content: '-';
243         color: #FFF;
244         cursor: pointer;
245 }
246 body .nav .nav-menu li a {
247         padding: 15px 30px;
248 }
249 .clicker {
250         width: 45px;
251         height: 45px;
252         position: absolute;
253         right: 0;
254         z-index: 30000;
255 }
256 .nav-clear {
257         clear: both;
258         padding-top: 67px;
259 }
260 /*--- End Navigation Style --*/
261 /*--------------------MEDIA!!!---------------*/
262 @media screen and (max-width: 768px) {
263         #banner-wrapper {
264                 position: relative;
265                 -ms-overflow-x: hidden;
266                 overflow-x: hidden;
267         }
268         .nav-menu {
269                 display: none;
270         }
271 }
```

A.1.2 Código JQuery - Foundation *Framework*

```
1
2 /**
3  * BxSlider v4.1.2 - Fully loaded, responsive content slider
```

```
 * http://bxslider.com
 *
 * Written by: Steven Wanderski, 2014
 * http://stevenwanderski.com
 * (while drinking Belgian ales and listening to jazz)
 *
 */


/** RESET AND LAYOUT
===================================*/

.bx-wrapper {
max-width: 1020px;
        position: relative;
        padding: 0;
        *zoom: 1;
}

.bx-wrapper img {
        max-width: 1020px;
        display: block;
}

/** THEME
====================================*/

.bx-wrapper .bx-viewport {
        left: 0px;
        background: #fff;

        /*fix other elements on the page moving (on Chrome)*/
        -webkit-transform: translatez(0);
        -moz-transform: translatez(0);
        -ms-transform: translatez(0);
        -o-transform: translatez(0);
        transform: translatez(0);
}

.bx-wrapper .bx-pager,
.bx-wrapper .bx-controls-auto {
        position: absolute;
```

```
46          bottom: -30px;
47          max-width: 1020px;
48  }
49
50  /* LOADER */
51
52  .bx-wrapper .bx-loading {
53          min-height: 50px;
54          background: url("img/bx_loader.gif") center center no-
                repeat #fff;
55          height: 100%;
56          max-width: 1020px;;
57          position: relative;
58          top: 0;
59          left: 0;
60          z-index: 2000;
61  }
62
63  /* PAGER */
64
65  .bx-wrapper .bx-pager {
66          text-align: center;
67          font-size: .85em;
68          font-family: Arial;
69          font-weight: bold;
70          color: #666;
71          padding-top: 20px;
72  }
73
74  .bx-wrapper .bx-pager .bx-pager-item,
75  .bx-wrapper .bx-controls-auto .bx-controls-auto-item {
76          display: inline-block;
77          *zoom: 1;
78          *display: inline;
79  }
80
81  .bx-wrapper .bx-pager.bx-default-pager a {
82          background: #666;
83          text-indent: -9999px;
84          display: block;
85          width: 10px;
86          height: 10px;
```

```
87          margin: 0 5px;
88          outline: 0;
89          -moz-border-radius: 5px;
90          -webkit-border-radius: 5px;
91          border-radius: 5px;
92  }
93
94  .bx-wrapper .bx-pager.bx-default-pager a:hover,
95  .bx-wrapper .bx-pager.bx-default-pager a.active {
96          background: #000;
97  }
98
99  /* DIRECTION CONTROLS (NEXT / PREV) */
100
101 .bx-wrapper .bx-prev {
102         left: 10px;
103         background: url("img/controls.png") no-repeat 0 -32px;
104 }
105
106 .bx-wrapper .bx-next {
107         right: 10px;
108         background: url("img/controls.png") no-repeat -43px -32px
              ;
109 }
110
111 .bx-wrapper .bx-prev:hover {
112         background-position: 0 0;
113 }
114
115 .bx-wrapper .bx-next:hover {
116         background-position: -43px 0;
117 }
118
119 .bx-wrapper .bx-controls-direction a {
120         position: relative;
121         top: 50%;
122         margin-top: -16px;
123         outline: 0;
124         width: 32px;
125         height: 32px;
126         text-indent: -9999px;
127         z-index: 9999;
```

```
128 }
129
130 .bx-wrapper .bx-controls-direction a.disabled {
131         display: none;
132 }
133
134 /* AUTO CONTROLS (START / STOP) */
135
136 .bx-wrapper .bx-controls-auto {
137         text-align: center;
138 }
139
140 .bx-wrapper .bx-controls-auto .bx-start {
141         display: block;
142         text-indent: -9999px;
143         width: 10px;
144         height: 11px;
145         outline: 0;
146         background: url("img/controls.png") -86px -11px no-repeat
              ;
147         margin: 0 3px;
148 }
149
150 .bx-wrapper .bx-controls-auto .bx-start:hover,
151 .bx-wrapper .bx-controls-auto .bx-start.active {
152         background-position: -86px 0;
153 }
154
155 .bx-wrapper .bx-controls-auto .bx-stop {
156         display: block;
157         text-indent: -9999px;
158         width: 9px;
159         height: 11px;
160         outline: 0;
161         background: url("img/controls.png") -86px -44px no-repeat
              ;
162         margin: 0 3px;
163 }
164
165 .bx-wrapper .bx-controls-auto .bx-stop:hover,
166 .bx-wrapper .bx-controls-auto .bx-stop.active {
167         background-position: -86px -33px;
```

```
168 }
169
170 /* PAGER WITH AUTO-CONTROLS HYBRID LAYOUT */
171
172 .bx-wrapper .bx-controls.bx-has-controls-auto.bx-has-pager .bx-
    pager {
173         text-align: left;
174         width: 80%;
175 }
176
177 .bx-wrapper .bx-controls.bx-has-controls-auto.bx-has-pager .bx-
    controls-auto {
178         right: 0;
179         width: 35px;
180 }
181
182 /* IMAGE CAPTIONS */
183
184 .bx-wrapper .bx-caption {
185         position: absolute;
186         bottom: 0;
187         left: 0;
188         background: #666\9;
189         background: rgba(80, 80, 80, 0.75);
190         width: 100%;
191 }
192
193 .bx-wrapper .bx-caption span {
194         color: #fff;
195         font-family: Arial;
196         display: block;
197         font-size: .85em;
198         padding: 10px;
199 }
```

A.1.3 Código JavaScript - Foundation *Framework*

```
1
2 function responsiveMobileMenu() {
3         $('.nav').each(function() {
4
5
```

```
6
7                   // $(this).children('ul').addClass('nav-menu');
                       // mark main menu list
8
9
10                  var $style = $(this).attr('nav-menu-style'); //
                       get menu style
11                  if (typeof $style == 'undefined' || $style ==
                       false) {
12                        $(this).addClass('yoga'); // set yoga
                             style if style is not defined
13                  } else {
14                        $(this).addClass($style);
15                  }
16
17
18                  /*      width of menu list (non-toggled) */
19                  var $width = 0;
20                  $(this).find('ul li').each(function() {
21                        $width += $(this).outerWidth();
22                  });
23
24                  // if modern browser
25
26                  if ($.support.leadingWhitespace) {
27                        // $(this).css('max-width', $width * 1.2
                             + 'px');
28                  }
29                  //
30                  else {
31                        $(this).css('width', $width * 1.2 + 'px')
                             ;
32                  }
33
34          });
35  }
36
37  function getMobileMenu() {
38
39          /*      build toggled dropdown menu list */
40
41          $('.nav').each(function() {
```

```
42                  var menutitle = $(this).attr("nav-menu-title");
43                  if (menutitle == "") {
44                          menutitle = "";
45                  } else if (menutitle == undefined) {
46                          menutitle = "";
47                  }
48                  var $menulist = $(this).children('.nav-menu').
                        html();
49                  var $menucontrols = "<div class='nav-toggled-
                        controls'><div class='nav-toggled-title'>" +
                        menutitle + "</div><div class='nav-button'><
                        span> </span><span> </span><span>&
                        nbsp;</span></div></div>";
50                  $(this).prepend("<div class='nav-toggled nav-
                        closed'>" + $menucontrols + "<ul>" + $menulist
                        + "</ul></div>");
51
52          });
53  }
54
55  function adaptMenu() {
56
57          /*      toggle menu on resize */
58
59          $('.nav').each(function() {
60                  // var $width = $(this).css('max-width');
61                  // $width = $width.replace('px', '');
62                  var $width = 768;
63                  if ($(this).parent().width() < $width * 1.05) {
64                          $(this).children('.nav-menu').hide(0);
65                          $(this).children('.nav-toggled').show(0);
66                  } else {
67                          $(this).children('.nav-menu').show(0);
68                          $(this).children('.nav-toggled').hide(0);
69                  }
70          });
71
72  }
73
74  $(function() {
75
76          responsiveMobileMenu();
```

```
77          getMobileMenu();
78          adaptMenu();
79
80          /* slide down mobile menu on click */
81
82          $('.nav-toggled, .nav-toggled .nav-button').click(
               function() {
83                  if ($(this).is(".nav-closed")) {
84                          $(this).find('> ul').stop().show(300);
85                          $(this).removeClass("nav-closed");
86                  } else {
87                          $(this).find(' > ul').stop().hide(300);
88                          $(this).addClass("nav-closed");
89                  }
90
91          });
92
93  });
94  /*      hide mobile menu on resize */
95  $(window).resize(function() {
96          adaptMenu();
97  });
98
99  $(document).ready(function() {
100         //test for touch events support and if not supported,
               attach .no-touch class to the HTML tag.
101         if (!("ontouchstart" in document.documentElement)) {
102                 document.documentElement.className += " no-touch"
                       ;
103         } else {
104                 document.documentElement.className += " touch";
105         }
106
107
108         var $submenus = $('.nav.yoga .nav-menu li > ul');
109         $submenus.hide();
110         if ($('html').hasClass('no-touch')) {
111                 $submenus.parent().hover(function() {
112                         $(this).find('> ul').slideToggle('fast');
113                 });
114         } else {
115                 $submenus.parent().click(function(e) {
```

```
116
117                         $(this).find('> ul').slideToggle('fast');
118                         e.preventDefault();
119                         e.stopPropagation();
120                 });
121         }
122
123         $submenus.parent().addClass('has-children');
124
125
126         var $submenusSm = $('.nav.yoga .nav-toggled li > ul');
127         $submenusSm.hide();
128         $submenusSm.parent().prepend('<div class="clicker"></div>
              ');
129         $submenusSm.parent().children('a').addClass('toggle-sm');
130         var clicker = $('.clicker');
131         clicker.on('click', function(e){
132                 e.stopPropagation();
133                 $(this).parent().find('> ul').slideToggle();
134                 $(this).parent().children('a').toggleClass('open'
                    );
135         });
136
137 });
```

A.2 Código HTML da Página Inicial

```
1
2
3 <!DOCTYPE html>
4 <head>
5         <meta name="viewport" content="width=device-width,
             initial-scale=1"> <!-- for mobile version -->
6         <title> infoAR - Home</title> <!-- what apperars when you
              search on Google or on the top of the tab -->
7         <link href="css/jquery.bxslider.css" rel="stylesheet" />
8         <link href="style.css" rel="stylesheet">
9         <link href="css/nav.css" rel="stylesheet">
10        <link rel="shortcut icon" type="image/png" href="img/
             icone.png"/> <!-- logo 40x40 -->
11        <style>
```

```
12          #map {
13          height: 450px;
14          width: 80%;
15          margin: 2% 4% 10% 10%;-->
16
17         }
18      </style>
19  </head>
20  <body>
21          <a name="home">
22  <!--- Start Navigation -->
23                  <script src="js/jquery-1.11.2.min.js"></script>
24                  <script src="js/main.js"></script> <!--- For
                        Navigation -->
25
26  <!---ADD NAVIGATION HERE-->
27  <div class= "nav-outer">
28  <div class="nav-wrap">
29          <nav class="navigation">
30                          <div class="logo"><img src = "img/
                                logobranco.png"></div>
31                                  <div class='nav' nav-menu-style =
                                        "
32                                  yoga">
33                                          <ul class = "nav-menu">
34                                                  <li><a href= "
                                                        index.php">Home
                                                        </a></li> <!--
                                                        linkar com
                                                        outras webpages
                                                        -->
35                                                  <li><a href="info
                                                        .php">Info</a
                                                        ></li>
36                                                  <li><a href="
                                                        poluentes.php">
                                                        Poluentes</a></
                                                        li>
37                                                  <li><a href="
                                                        cadastrese.php"
                                                        >Cadastre-se</a
                                                        ></li>
```

```
38                                                  <li><a href="
                                                       faleconosco.php
                                                       ">Fale Conosco
                                                       </a></li>
39                                  </div>
40          </nav>
41  </div>
42  </div>
43          <div class="nav-clear"></div>
44  <!--- End Navigation -->
45          <div class="clearfix"></div>
46  <!--- Start Slider -->
47                  <script src="js/jquery.bxslider.min.js"></script
                       ><!--For Image Slider-->
48                  <div class="slide-wrap">
49
50  <!--ADD IMAGE SLIDES HERE-->
51  <section class="slider">
52          <ul class="slider1">
53                  <li><img src="img/capa1.png"></li>
54                  <li><img src="img/capa2.png"></li>
55          </ul>
56  </section>
57                  </div>
58                  <script type="text/javascript">
59                          $('.slider1').bxSlider({
60                                  mode: 'fade',
61                                  captions: false,
62                                  auto:true,
63                                  pager:false,
64
65                          });
66                          $('.slider2').bxSlider({
67                                  pager:false,
68                                  captions: true,
69                                  maxSlides: 3,
70                                  minSlides: 1,
71                                  slideWidth: 230,
72                                  slideMargin: 10
73                          });
74
75                  </script>
```

```
76
77
78  <!--- End Slider -->
79  <!--<a name="info">-->
80          <!--Start Banner Wrapper-->
81          <div id="banner-wrapper">
82          <h1>Qualidade do Ar na Grande Vit&oacute;ria</h1>
83          </div>
84
85
86          <!--Google Maps-->
87   <div id="map"></div>
88      <script type= "text/javascript">
89        function initMap() {
90          var ufes = {lat: -20.27, lng: -40.30};
91          var locations = [
92                  ['Laranjeiras', -20.196, -40.245],
93                  ['Carapina', -20.226, -40.259],
94                  ['J.Camburi', -20.255, -40.270],
95                  ['Enseada do Sua', -20.313, -40.291],
96                  ['Centro-Vitoria', -20.321, -40.333],
97                  ['Ibes', -20.348, -40.317],
98                  ['VV-Centro', -20.336, -40.291],
99                  ['Cariacica', -20.342, -40.402],
100                 ['Cidade Continental', -20.225, -40.218],
101         ];
102         var map = new google.maps.Map(document.getElementById('
               map'), {
103           zoom: 12,
104           center: ufes
105         });
106
107          var infowindow = new google.maps.InfoWindow();
108
109         var marker, i, iqa
110
111         //var MarkerWithLabel = require('markerwithlabel')(google
               .maps);
112
113         iqa = "23"
114
115        for (i = 0; i < locations.length; i++) {
```

```
116         marker = new google.maps.Marker({
117         position: new google.maps.LatLng(locations[i][1],
               locations[i][2]),
118         icon: 'img/verde.png',
119         label:iqa,
120         map: map
121       });
122          google.maps.event.addListener(marker, 'click', (function
               (marker, i) {
123         return function() {
124           infowindow.setContent(locations[i][0]);
125           infowindow.open(map, marker);
126         }
127       })(marker, i));
128     }
129       }
130     </script>
131     <script async defer
132     src="https://maps.googleapis.com/maps/api/js?key=
          AIzaSyA8MaWUjzVeEzHQ2WVIG3rrDBKfaEzqrXY&callback=initMap">
133     </script>
134
135 <!-- End Banner Wrapper-->
136 <div class="clearfix"></div>
137 <!--Start Parallax Section-->
138         <section class="parallax-1">
139                 <div class="parallax-inner">
140                         <section class="one-half">
141                                 <h3>Sobre N&oacute;s</h3>
142                                 <p> Esse projeto busca atender
                                    uma demanda do IEMA dentro de
                                    um plano de colabora&ccedil;&
                                    atilde;o cont&iacute;nua com o
                                    grupo de pesquisadores dos
                                    Departamentos de Engenharia El&
                                    eacute;trica e Engenharia
                                    Ambiental da Universidade
                                    Federal do Esp&iacute;rito
                                    Santo (UFES). O desenvolvimento
                                     deste website e dos
                                    aplicativos fazem parte do
                                    Projeto de Gradua&ccedil;&
```

```
                                    atilde;o da aluna do curso de
                                    Engenharia El&eacute;trica
                                    Mayara Nascimento de Oliveira.
                                    A aluna &eacute; orientanda do
                                    Prof. Dr. Teodiano Freire
                                    Bastos do Departamento de
                                    Engenharia El&eacute;trica e da
                                     Prof &ordf;. Dr &ordf;. Jane M
                                    . Santos do Departamento de
                                    Engenharia Ambiental </p>
143                         </section>
144                         <section class="one-half">
145                                 <h3>IEMA</h3>
146                                 <p> O IEMA &eacute; o Instituto
                                    Estadual de Meio Ambiente e
                                    Recursos H&iacute;dricos
                                    respons&aacute;vel pela execu&
                                    ccedil;&atilde;o das pol&iacute
                                    ;ticas estaduais de meio
                                    ambiente. Suas atribui&ccedil;&
                                    otilde;es incluem o
                                    monitoramento, fiscaliza&ccedil
                                    ;&atilde;o, pesquisa, trabalhos
                                     de educa&ccedil;&atilde;o
                                    ambiental e o licenciamento de
                                    empreendimentos que realizam
                                    atividades potencialmente
                                    poluidoras </p>
147                         </section>
148                 </div>
149         </section>
150
151 <!--End Parallax Section-->
152 <div class="clearfix"></div>
153 <footer>
154 <p> &copy; Copyright infoAR - 2016 - Todos os Direitos Reservados
        </p>
155
156 </footer>
157
158 </body>
159 </html>
```

A.3 Código HTML da Página de Informação

```
1
2  <!DOCTYPE html>
3  <head>
4          <meta name="viewport" content="width=device-width,
              initial-scale=1"> <!-- for mobile version -->
5          <title> infoAR - Info</title> <!-- what apperars when you
               search on Google or on the top of the tab -->
6          <link href="css/jquery.bxslider.css" rel="stylesheet" />
7          <link href="style.css" rel="stylesheet">
8          <link href="css/nav.css" rel="stylesheet">
9          <link rel="shortcut icon" type="image/png" href="img/
              icone.png"/> <!-- logo 40x40 -->
10         <style>
11         #map {
12         height: 450px;
13         width: 80%;
14         margin: 2% 4% 10% 10%;-->
15
16        }
17     </style>
18 </head>
19 <body>
20 <!--- Start Navigation -->
21                 <script src="js/jquery-1.11.2.min.js"></script>
22                 <script src="js/main.js"></script> <!--- For
                      Navigation -->
23
24 <!---ADD NAVIGATION HERE-->
25 <div class= "nav-outer">
26 <div class="nav-wrap">
27         <nav class="navigation">
28                         <div class="logo"><img src = "img/
                               logobranco.png"></div>
29                                 <div class='nav' nav-menu-style =
                                       "
30                                 yoga">
31                                         <ul class = "nav-menu">
32                                                 <li><a href= "
                                                       index.php">Home
                                                       </a></li> <!--
```

```
                                                    linkar com
                                                    outras webpages
                                                    -->
33                                                  <li><a href="info
                                                    .php">Info</a
                                                    ></li>
34                                                  <li><a href="
                                                    poluentes.php">
                                                    Poluentes</a></
                                                    li>
35                                                  <li><a href="
                                                    cadastrese.php"
                                                    >Cadastre-se</a
                                                    ></li>
36                                                  <li><a href="
                                                    faleconosco.php
                                                    ">Fale Conosco
                                                    </a></li>
37                                  </div>
38          </nav>
39  </div>
40  </div>
41          <div class="nav-clear"></div>
42  <!--- End Navigation -->
43          <div class="clearfix"></div>
44  <!--- Start Slider -->
45                  <script src="js/jquery.bxslider.min.js"></script
                        ><!--For Image Slider-->
46                  <div class="slide-wrap">
47
48  <!--ADD IMAGE SLIDES HERE-->
49  <section class="slider">
50          <ul class="slider1">
51                  <li><img src="img/capa1.png"></li>
52                  <li><img src="img/capa2.png"></li>
53          </ul>
54  </section>
55                  </div>
56                  <script type="text/javascript">
57                          $('.slider1').bxSlider({
58                                  mode: 'fade',
59                                  captions: false,
```

```
60                                  auto:true,
61                                  pager:false,
62
63                          });
64                          $('.slider2').bxSlider({
65                                  pager:false,
66                                  captions: true,
67                                  maxSlides: 3,
68                                  minSlides: 1,
69                                  slideWidth: 230,
70                                  slideMargin: 10
71                          });
72
73                  </script>
74
75
76 <!--- End Slider -->
77 <!--<a name="info">-->
78         <!--Start Banner Wrapper-->
79         <div id="banner-wrapper">
80         <h1>O que as nuvens significam?</h1>
81         </div>
82         <h3>As cores das nuvens representam o IQA (&Iacute;ndice
              de Qualidade do Ar),ou seja, o qu&atilde;o polu&iacute;
              do o ar est&aacute;</h3>
83
84 <!--<div style="overflow-x:auto;"> -->
85         <table>
86         <thead>
87                 <tr>
88                         <th>&Iacute;ndice/Classifica&ccedil;&
                              atilde;o</th>
89                         <th>Significado</th>
90                 </tr>
91         </thead>
92         <tbody>
93                 <tr>
94                         <td><img src="img/bom.png"><br> BOM </td>
95                         <td>Atendendo as recomenda&ccedil;&otilde
                              ;es da OMS (Padr&otilde;es finais da
                              legisla&ccedil;&atilde;o final do ES),
                              para que a sa&uacute;de da popula&
```

```
                                ccedil;&atilde;o seja preservada ao m&
                                aacute;ximo, em rela&ccedil;&atilde;o
                                aos danos causados pela polui&ccedil;&
                                atilde;o atmosf&eacute;rica </td>
96                  </tr>
97                  <tr>
98                          <td><img src="img/moderado.png"><br>
                                MODERADO </td>
99                          <td> Pessoas de grupos sens&iacute;veis (
                                crian&ccedil;as, idosos e pessoas com
                                problemas respirat&oacute;rios ou card&
                                iacute;acos)podem apresentar sintomas
                                como tosse seca e cansa&ccedil;o. A
                                popula&ccedil;&atilde;o em geral ãno &
                                eacute; afetada </td>
100                 </tr>
101
102                 <tr>
103                         <td><img src="img/ruim.png"><br> RUIM </
                                td>
104                         <td>Toda a popula&ccedil;&atilde;o pode
                                apresentar sintomas como tosse seca,
                                cansa&ccedil;o, ardor nos olhos, nariz
                                e garganta. Pessoas dos grupos sens&
                                iacute;veis (crian&ccedil;as, idosos e
                                pessoas com problemas respirat&oacute;
                                rios ou card&iacute;acos) podem
                                apresentar efeitos mais s&eacute;rios
                                na sa&uacute;de </td>
105                 </tr>
106
107                 <tr>
108                         <td><img src="img/muito ruim.png"><br>
                                MUITO RUIM </td>
109                         <td>Toda a popula&ccedil;&atilde;o pode
                                apresentar agravamento dos sintomas
                                como tosse seca, cansa&ccedil;o, ardor
                                nos olhos, nariz e garganta e ainda
                                falta de ar e respira&ccedil;&atilde;o
                                ofegante. Efeitos ainda mais graves &
                                agrave; sa&uacute;de de grupos sens&
                                iacute;veis (crian&ccedil;as, idosos e
```

```
                                pessoas com problemas respirat&oacute;
                                rios ou card&iacute;acos)</td>
110                 </tr>
111
112                 <tr>
113                         <td><img src="img/pessimo.png"><br> P&
                                Eacute;SSIMO </td>
114                         <td>Toda a popula&ccedil;&atilde;o pode
                                apresentar s&eacute;rios riscos de
                                manifesta&ccedil;&atilde;o de doen&
                                ccedil;as respirat&oacute;rias e
                                cardiovasculares. Aumento de mortes
                                prematuras em pessoas dos grupos sens&
                                iacute;veis  </td>
115                 </tr>
116
117                 <tr>
118                         <td><img src="img/represent.png"><br> SEM
                                 REPRESENTATIVIDADE </td>
119                         <td>Dados sem representatividade </td>
120                 </tr>
121         </tbody>
122 </table>
123 <!--</div>-->
124
125
126 <div class="clearfix"></div>
127 <h1>Como o IQA &eacute; calculado?</h1>
128
129 <h3>Primeiro &eacute; preciso definir como a concentra&ccedil;&
    atilde;o de cada poluente &eacute; calculada: </h3>
130
131 <section class="one-third">
132 <h4>Para MP2,5 , MP10 e SO2</h4>
133 <p>Ser&atilde;o calculadas m&eacute;dias m&oacute;veis de 24
    horas</p>
134 </section>
135
136 <section class="one-third">
137 <h4>Para CO e O3</h4>
138 <p>Ser&atilde;o calculadas m&eacute;dias m&oacute;veis de 8 horas
    </p>
```

```
</section>

<section class="one-third">
<h4>Para NO2</h4>
<p>Ser&aacute; calculada a m&eacute;dia hor&aacute;ria</p>
</section>

<div class="clearfix"></div>
<h3> O IQA ser&aacute; baseado na concentra&ccedil;&atilde;o do
   pior poluente</h3>
<div class="clearfix"></div>

<div class="clearfix"></div>
<h3>Para mais informa&ccedil;&otilde;es veja nossa cartilha
   clicando no bot&atilde;o abaixo!</h3>
<form class = "form" method = "get" action = "downloads/
   Guia_Qualidade_Ar.pdf">
<p class = "submit">
                        <input type = "submit" style = "
                           background-color:#87CEEB" value = "
                           Cartilha">
                </p>

</form>


<div class="clearfix"></div>
<h1> Mantenha-se conectado baixando nossos aplicativos</h1>
<section class = "icons">
        <img src = "img/app-icon.png">
</section>

<section class = "icons">
        <img src = "img/google-play-icon.png">
</section>

<div class="clearfix"></div>
<!--Start Parallax Section-->
        <section class="parallax-1">
                <div class="parallax-inner">
                        <section class="one-half">
                                <h3>Sobre N&oacute;s</h3>
```

```
176                             <p> Esse projeto busca atender
                                   uma demanda do IEMA dentro de
                                   um plano de colabora&ccedil;&
                                   atilde;o cont&iacute;nua com o
                                   grupo de pesquisadores dos
                                   Departamentos de Engenharia El&
                                   eacute;trica e Engenharia
                                   Ambiental da Universidade
                                   Federal do Esp&iacute;rito
                                   Santo (UFES). O desenvolvimento
                                    deste website e dos
                                   aplicativos fazem parte do
                                   Projeto de Gradua&ccedil;&
                                   atilde;o da aluna do curso de
                                   Engenharia El&eacute;trica
                                   Mayara Nascimento de Oliveira.
                                   A aluna &eacute; orientanda do
                                   Prof. Dr. Teodiano Freire
                                   Bastos do Departamento de
                                   Engenharia El&eacute;trica e da
                                    Prof &ordf;. Dr &ordf;. Jane M
                                   . Santos do Departamento de
                                   Engenharia Ambiental </p>
177                         </section>
178                         <section class="one-half">
179                             <h3>IEMA</h3>
180                             <p> O IEMA &eacute; o Instituto
                                   Estadual de Meio Ambiente e
                                   Recursos H&iacute;dricos
                                   respons&aacute;vel pela execu&
                                   ccedil;&atilde;o das pol&iacute
                                   ;ticas estaduais de meio
                                   ambiente. Suas atribui&ccedil;&
                                   otilde;es incluem o
                                   monitoramento, fiscaliza&ccedil
                                   ;&atilde;o, pesquisa, trabalhos
                                    de educa&ccedil;&atilde;o
                                   ambiental e o licenciamento de
                                   empreendimentos que realizam
                                   atividades potencialmente
                                   poluidoras </p>
181                         </section>
```

```
182                 </div>
183         </section>
184         <div class="clearfix"></div>
185 
186 <footer>
187 <p> &copy; Copyright infoAR - 2016 - Todos os Direitos Reservados
        </p>
188 
189 </footer>
190 
191 </body>
192 </html>
```

A.4 Código HTML da Página dos Poluentes

```
1 
2 <!DOCTYPE html>
3 <head>
4         <meta name="viewport" content="width=device-width,
            initial-scale=1"> <!-- for mobile version -->
5         <title> infoAR - Poluentes</title> <!-- what apperars
            when you search on Google or on the top of the tab -->
6         <link href="css/jquery.bxslider.css" rel="stylesheet" />
7         <link href="style.css" rel="stylesheet">
8         <link href="css/nav.css" rel="stylesheet">
9         <link rel="shortcut icon" type="image/png" href="img/
            icone.png"/> <!-- logo 40x40 -->
10        <style>
11        #map {
12        height: 450px;
13        width: 80%;
14        margin: 2% 4% 10% 10%;-->
15 
16       }
17    </style>
18 </head>
19 <body>
20        <a name="home">
21 <!--- Start Navigation -->
22                <script src="js/jquery-1.11.2.min.js"></script>
23                <script src="js/main.js"></script> <!--- For
```

```
                    Navigation -->
24
25 <!---ADD NAVIGATION HERE-->
26 <div class= "nav-outer">
27 <div class="nav-wrap">
28         <nav class="navigation">
29                         <div class="logo"><img src = "img/
                               logobranco.png"></div>
30                             <div class='nav' nav-menu-style =
                                   "
31                             yoga">
32                                     <ul class = "nav-menu">
33                                             <li><a href= "
                                                index.php">Home
                                                </a></li> <!--
                                                linkar com
                                                outras webpages
                                                -->
34                                             <li><a href="info
                                                .php">Info</a
                                                ></li>
35                                             <li><a href="
                                                poluentes.php">
                                                Poluentes</a></
                                                li>
36                                             <li><a href="
                                                cadastrese.php"
                                                >Cadastre-se</a
                                                ></li>
37                                             <li><a href="
                                                faleconosco.php
                                                ">Fale Conosco
                                                </a></li>
38                             </div>
39         </nav>
40 </div>
41 </div>
42         <div class="nav-clear"></div>
43 <!--- End Navigation -->
44         <div class="clearfix"></div>
45 <!--- Start Slider -->
46                 <script src="js/jquery.bxslider.min.js"></script
```

```
                    ><!--For Image Slider-->
47                  <div class="slide-wrap">
48
49  <!--ADD IMAGE SLIDES HERE-->
50  <section class="slider">
51          <ul class="slider1">
52                  <li><img src="img/capa1.png"></li>
53                  <li><img src="img/capa2.png"></li>
54          </ul>
55  </section>
56                  </div>
57                  <script type="text/javascript">
58                          $('.slider1').bxSlider({
59                                  mode: 'fade',
60                                  captions: false,
61                                  auto:true,
62                                  pager:false,
63
64                          });
65                          $('.slider2').bxSlider({
66                                  pager:false,
67                                  captions: true,
68                                  maxSlides: 3,
69                                  minSlides: 1,
70                                  slideWidth: 230,
71                                  slideMargin: 10
72                          });
73
74                  </script>
75
76  <script>
77  var col = [];
78          for (var i = 0; i < concent.length; i++) {
79              for (var key in concent[i]) {
80                  if (col.indexOf(key) === -1) {
81                      col.push(key);
82                  }
83              }
84          }
85
86   var table = document.createElement("table");
87
```

```
88    var tr = table.insertRow(-1);
89
90          for (var i = 0; i < col.length; i++) {
91              var th = document.createElement("th");
92              th.innerHTML = col[i];
93              tr.appendChild(th);
94          }
95
96          for (var i = 0; i < concent.length; i++) {
97
98              tr = table.insertRow(-1);
99
100             for (var j = 0; j < col.length; j++) {
101                 var tabCell = tr.insertCell(-1);
102                 tabCell.innerHTML = concent[i][col[j]];
103             }
104         }
105
106         var divContainer = document.getElementById("showData");
107         divContainer.innerHTML = "";
108         divContainer.appendChild(table);
109  </script>
110
111 <!--- End Slider -->
112 <!--<a name="info">-->
113         <!--Start Banner Wrapper-->
114         <div id="banner-wrapper">
115         <h1>Concentra&ccedil;&atilde;o dos Poluentes em cada esta
               &ccedil;&atilde;o</h1>
116
117 <h4>Cidade Continental</h4>
118
119 <table>
120         <thead>
121                 <tr>
122                         <th>Poluente</th>
123                         <th>Concentra&ccedil;&atilde;o</th>
124                 </tr>
125         </thead>
126         <tbody>
127                 <tr>
128                         <td>MP 2,5 (&mu;g/m&sup3;) 24h</td>
```

```
129                         <td>  </td>
130                 </tr>
131                 <tr>
132                         <td> MP 10 (&mu;g/m&sup3;) 24h</td>
133                         <td> </td>
134                 </tr>
135
136                 <tr>
137                         <td> SO2 (&mu;g/m&sup3;) 24h</td>
138                         <td> </td>
139                 </tr>
140
141                 <tr>
142                         <td>O3 (&mu;g/m&sup3;) 8h</td>
143                         <td> </td>
144                 </tr>
145
146                 <tr>
147                         <td>CO (ppm) 8h</td>
148                         <td> </td>
149                 </tr>
150
151                 <tr>
152                         <td>NO2 (&mu;g/m&sup3;) 1h</td>
153                         <td> </td>
154                 </tr>
155         </tbody>
156 </table>
157
158 <h4> Laranjeiras</h4>
159 <table>
160         <thead>
161                 <tr>
162                         <th>Poluente</th>
163                         <th>Concentra&ccedil;&atilde;o</th>
164                 </tr>
165         </thead>
166         <tbody>
167                 <tr>
168                         <td>MP 2,5 (&mu;g/m&sup3;) 24h</td>
169                         <td>  </td>
170                 </tr>
```

```
171                 <tr>
172                         <td> MP 10 (&mu;g/m&sup3;) 24h</td>
173                         <td> </td>
174                  </tr>
175
176                  <tr>
177                         <td> SO2 (&mu;g/m&sup3;) 24h</td>
178                         <td> </td>
179                  </tr>
180
181                  <tr>
182                         <td>O3 (&mu;g/m&sup3;) 8h</td>
183                         <td> </td>
184                  </tr>
185
186                  <tr>
187                         <td>CO (ppm) 8h</td>
188                         <td> </td>
189                  </tr>
190
191                  <tr>
192                         <td>NO2 (&mu;g/m&sup3;) 1h</td>
193                         <td> </td>
194                  </tr>
195         </tbody>
196 </table>
197
198 <h4>Jardim Camburi</h4>
199 <table>
200         <thead>
201                  <tr>
202                         <th>Poluente</th>
203                         <th>Concentra&ccedil;&atilde;o</th>
204                  </tr>
205         </thead>
206         <tbody>
207                  <tr>
208                         <td>MP 2,5 (&mu;g/m&sup3;) 24h</td>
209                         <td>  </td>
210                  </tr>
211                 <tr>
212                         <td> MP 10 (&mu;g/m&sup3;) 24h</td>
```

```
213                         <td> </td>
214                 </tr>
215
216                 <tr>
217                         <td> SO2 (&mu;g/m&sup3;) 24h</td>
218                         <td> </td>
219                 </tr>
220
221                 <tr>
222                         <td>O3 (&mu;g/m&sup3;) 8h</td>
223                         <td> </td>
224                 </tr>
225
226                 <tr>
227                         <td>CO (ppm) 8h</td>
228                         <td> </td>
229                 </tr>
230
231                 <tr>
232                         <td>NO2 (&mu;g/m&sup3;) 1h</td>
233                         <td> </td>
234                 </tr>
235         </tbody>
236 </table>
237
238 <h4>Enseada do Su&aacute;</h4>
239 <table>
240         <thead>
241                 <tr>
242                         <th>Poluente</th>
243                         <th>Concentra&ccedil;&atilde;o</th>
244                 </tr>
245         </thead>
246         <tbody>
247                 <tr>
248                         <td>MP 2,5 (&mu;g/m&sup3;) 24h</td>
249                         <td>  </td>
250                 </tr>
251                <tr>
252                         <td> MP 10 (&mu;g/m&sup3;) 24h</td>
253                         <td> </td>
254                 </tr>
```

```
255
256                 <tr>
257                         <td> SO2 (&mu;g/m&sup3;) 24h</td>
258                         <td> </td>
259                 </tr>
260
261                 <tr>
262                         <td>O3 (&mu;g/m&sup3;) 8h</td>
263                         <td> </td>
264                 </tr>
265
266                 <tr>
267                         <td>CO (ppm) 8h</td>
268                         <td> </td>
269                 </tr>
270
271                 <tr>
272                         <td>NO2 (&mu;g/m&sup3;) 1h</td>
273                         <td> </td>
274                 </tr>
275         </tbody>
276 </table>
277
278 <h4>Centro-Vit&oacute;ria</h4>
279 <table>
280         <thead>
281                 <tr>
282                         <th>Poluente</th>
283                         <th>Concentra&ccedil;&atilde;o</th>
284                 </tr>
285         </thead>
286         <tbody>
287                 <tr>
288                         <td>MP 2,5 (&mu;g/m&sup3;) 24h</td>
289                         <td>  </td>
290                 </tr>
291                <tr>
292                         <td> MP 10 (&mu;g/m&sup3;) 24h</td>
293                         <td> </td>
294                 </tr>
295
296                 <tr>
```

```
297                         <td> SO2 (&mu;g/m&sup3;) 24h</td>
298                         <td> </td>
299                 </tr>
300
301                 <tr>
302                         <td>O3 (&mu;g/m&sup3;) 8h</td>
303                         <td> </td>
304                 </tr>
305
306                 <tr>
307                         <td>CO (ppm) 8h</td>
308                         <td> </td>
309                 </tr>
310
311                 <tr>
312                         <td>NO2 (&mu;g/m&sup3;) 1h</td>
313                         <td> </td>
314                 </tr>
315         </tbody>
316 </table>
317
318 <h4>Carapina</h4>
319 <table>
320         <thead>
321                 <tr>
322                         <th>Poluente</th>
323                         <th>Concentra&ccedil;&atilde;o</th>
324                 </tr>
325         </thead>
326         <tbody>
327                 <tr>
328                         <td>MP 2,5 (&mu;g/m&sup3;) 24h</td>
329                         <td>  </td>
330                 </tr>
331                <tr>
332                         <td> MP 10 (&mu;g/m&sup3;) 24h</td>
333                         <td> </td>
334                 </tr>
335
336                 <tr>
337                         <td> SO2 (&mu;g/m&sup3;) 24h</td>
338                         <td> </td>
```

```
339                 </tr>
340
341                 <tr>
342                         <td>O3 (&mu;g/m&sup3;) 8h</td>
343                         <td> </td>
344                 </tr>
345
346                 <tr>
347                         <td>CO (ppm) 8h</td>
348                         <td> </td>
349                 </tr>
350
351                 <tr>
352                         <td>NO2 (&mu;g/m&sup3;) 1h</td>
353                         <td> </td>
354                 </tr>
355         </tbody>
356 </table>
357
358
359 <h4>VV-Centro</h4>
360 <table>
361         <thead>
362                 <tr>
363                         <th>Poluente</th>
364                         <th>Concentra&ccedil;&atilde;o</th>
365                 </tr>
366         </thead>
367         <tbody>
368                 <tr>
369                         <td>MP 2,5 (&mu;g/m&sup3;) 24h</td>
370                         <td>  </td>
371                 </tr>
372                <tr>
373                         <td> MP 10 (&mu;g/m&sup3;) 24h</td>
374                         <td> </td>
375                 </tr>
376
377                 <tr>
378                         <td> SO2 (&mu;g/m&sup3;) 24h</td>
379                         <td> </td>
380                 </tr>
```

```
381
382                 <tr>
383                         <td>O3 (&mu;g/m&sup3;) 8h</td>
384                         <td> </td>
385                 </tr>
386
387                 <tr>
388                         <td>CO (ppm) 8h</td>
389                         <td> </td>
390                 </tr>
391
392                 <tr>
393                         <td>NO2 (&mu;g/m&sup3;) 1h</td>
394                         <td> </td>
395                 </tr>
396         </tbody>
397 </table>
398
399
400 <h4> Ibes</h4>
401 <table>
402         <thead>
403                 <tr>
404                         <th>Poluente</th>
405                         <th>Concentra&ccedil;&atilde;o</th>
406                 </tr>
407         </thead>
408         <tbody>
409                 <tr>
410                         <td>MP 2,5 (&mu;g/m&sup3;) 24h</td>
411                         <td>  </td>
412                 </tr>
413                <tr>
414                         <td> MP 10 (&mu;g/m&sup3;) 24h</td>
415                         <td> </td>
416                 </tr>
417
418                 <tr>
419                         <td> SO2 (&mu;g/m&sup3;) 24h</td>
420                         <td> </td>
421                 </tr>
422
```

```
423                 <tr>
424                         <td>O3 (&mu;g/m&sup3;) 8h</td>
425                         <td> </td>
426                 </tr>
427
428                 <tr>
429                         <td>CO (ppm) 8h</td>
430                         <td> </td>
431                 </tr>
432
433                 <tr>
434                         <td>NO2 (&mu;g/m&sup3;) 1h</td>
435                         <td> </td>
436                 </tr>
437         </tbody>
438 </table>
439
440
441 <h4> Cariacica </h4>
442 <table>
443         <thead>
444                 <tr>
445                         <th>Poluente</th>
446                         <th>Concentra&ccedil;&atilde;o</th>
447                 </tr>
448         </thead>
449         <tbody>
450                 <tr>
451                         <td>MP 2,5 (&mu;g/m&sup3;) 24h</td>
452                         <td>  </td>
453                 </tr>
454                <tr>
455                         <td> MP 10 (&mu;g/m&sup3;) 24h</td>
456                         <td> </td>
457                 </tr>
458
459                 <tr>
460                         <td> SO2 (&mu;g/m&sup3;) 24h</td>
461                         <td> </td>
462                 </tr>
463
464                 <tr>
```

```
465                         <td>O3 (&mu;g/m&sup3;) 8h</td>
466                         <td> </td>
467                 </tr>
468
469                 <tr>
470                         <td>CO (ppm) 8h</td>
471                         <td> </td>
472                 </tr>
473
474                 <tr>
475                         <td>NO2 (&mu;g/m&sup3;) 1h</td>
476                         <td> </td>
477                 </tr>
478         </tbody>
479 </table>
480 </div>
481
482 <div class="clearfix"></div>
483 <!--Start Parallax Section-->
484     <section class="parallax-1">
485         <div class="parallax-inner">
486             <section class="one-half">
487                 <h3>Sobre N&oacute;s</h3>
488                 <p> Esse projeto busca atender uma demanda do
                       IEMA dentro de um plano de colabora&ccedil;&
                       atilde;o cont&iacute;nua com o grupo de
                       pesquisadores dos Departamentos de Engenharia
                       El&eacute;trica e Engenharia Ambiental da
                       Universidade Federal do Esp&iacute;rito Santo (
                       UFES). O desenvolvimento deste website e dos
                       aplicativos fazem parte do Projeto de Gradua&
                       ccedil;&atilde;o da aluna do curso de
                       Engenharia El&eacute;trica Mayara Nascimento de
                        Oliveira. A aluna &eacute; orientanda do Prof.
                        Dr. Teodiano Freire Bastos do Departamento de
                       Engenharia El&eacute;trica e da Prof &ordf;. Dr
                        &ordf;. Jane M. Santos do Departamento de
                       Engenharia Ambiental </p>
489             </section>
490             <section class="one-half">
491                 <h3>IEMA</h3>
492                 <p> O IEMA &eacute; o Instituto Estadual de Meio
```

```
                    Ambiente e Recursos H&iacute;dricos respons&
                    aacute;vel pela execu&ccedil;&atilde;o das pol&
                    iacute;ticas estaduais de meio ambiente. Suas
                    atribui&ccedil;&otilde;es incluem o
                    monitoramento, fiscaliza&ccedil;&atilde;o,
                    pesquisa, trabalhos de educa&ccedil;&atilde;o
                    ambiental e o licenciamento de empreendimentos
                    que realizam atividades potencialmente
                    poluidoras </p>
493             </section>
494         </div>
495     </section>
496
497 <!--End Parallax Section-->
498 <div class="clearfix"></div>
499 <footer>
500 <p> &copy; Copyright infoAR - 2016 - Todos os Direitos Reservados
        </p>
501
502 </footer>
503
504 </body>
505 </html>
```

A.5 Código HTML da Página para se Registrar

```
1
2 <?php
3 session_start();
4  ?>
5
6 <!DOCTYPE html>
7 <head>
8         <meta name="viewport" content="width=device-width,
             initial-scale=1"> <!-- for mobile version -->
9         <title> infoAR - Cadastre-se</title> <!-- what apperars
             when you search on Google or on the top of the tab -->
10        <link href="css/jquery.bxslider.css" rel="stylesheet" />
11        <link href="style.css" rel="stylesheet">
12        <link href="css/nav.css" rel="stylesheet">
13        <link rel="shortcut icon" type="image/png" href="img/
```

```
            icone.png"/> <!-- logo 40x40 -->
14          <style>
15          #map {
16          height: 450px;
17          width: 80%;
18          margin: 2% 4% 10% 10%;-->
19
20         }
21      </style>
22  </head>
23  <body>
24          <a name="home">
25  <!--- Start Navigation -->
26                  <script src="js/jquery-1.11.2.min.js"></script>
27                  <script src="js/main.js"></script> <!--- For
                        Navigation -->
28
29  <!---ADD NAVIGATION HERE-->
30  <div class= "nav-outer">
31  <div class="nav-wrap">
32          <nav class="navigation">
33                          <div class="logo"><img src = "img/
                                logobranco.png"></div>
34                                  <div class='nav' nav-menu-style =
                                        "
35                                  yoga">
36                                          <ul class = "nav-menu">
37                                                  <li><a href= "
                                                        index.php">Home
                                                        </a></li> <!--
                                                        linkar com
                                                        outras webpages
                                                        -->
38                                                  <li><a href="info
                                                        .php">Info</a
                                                        ></li>
39                                                  <li><a href="
                                                        poluentes.php">
                                                        Poluentes</a></
                                                        li>
40                                                  <li><a href="
                                                        cadastrese.php"
```

```
                                                    >Cadastre-se</a
                                                    ></li>
41                                                <li><a href="
                                                    faleconosco.php
                                                    ">Fale Conosco
                                                    </a></li>
42                                  </div>
43          </nav>
44  </div>
45  </div>
46          <div class="nav-clear"></div>
47  <!--- End Navigation -->
48          <div class="clearfix"></div>
49  <!--- Start Slider -->
50                  <script src="js/jquery.bxslider.min.js"></script
                       ><!--For Image Slider-->
51                  <div class="slide-wrap">
52
53  <!--ADD IMAGE SLIDES HERE-->
54  <section class="slider">
55          <ul class="slider1">
56                  <li><img src="img/capa1.png"></li>
57                  <li><img src="img/capa2.png"></li>
58          </ul>
59  </section>
60                  </div>
61                  <script type="text/javascript">
62                          $('.slider1').bxSlider({
63                                  mode: 'fade',
64                                  captions: false,
65                                  auto:true,
66                                  pager:false,
67
68                          });
69                          $('.slider2').bxSlider({
70                                  pager:false,
71                                  captions: true,
72                                  maxSlides: 3,
73                                  minSlides: 1,
74                                  slideWidth: 230,
75                                  slideMargin: 10
76                          });
```

```
77
78                  </script>
79
80
81  <!--- End Slider -->
82  <!--<a name="info">-->
83          <!--Start Banner Wrapper-->
84          <div id="banner-wrapper">
85
86
87          <h1><a href="#cadastrese" style="color: #4A4444; text-
                decoration: none;">Cadastre-se para receber alertas!</a
                ></h1>
88          <a name = "cadastrese">
89          <h4>escolha a esta&ccedil;&atilde;o mais pr&oacute;xima
                de sua casa e previna-se dos efeitos adversos causados
                pela baixa qualidade do ar</h4>
90
91          <form class ="form" method="post" action="save_user.php">
92          <section class = "one-third">
93                  <p class = "name">
94                          <input type = "text" name="name" id="name
                                " placeholder="Maria Silva"/>
95                          <label for= "name">Nome*</label>
96                  </p>
97  </section>
98  <section class = "one-third">
99          <p class = "email">
100                         <input type = "text" name="email" id="
                               email" placeholder="mail@example.com"/>
101                         <label for= "email">Email*</label>
102                 </p>
103 </section>
104
105 <section class = "one-third">
106         <p class = "address">
107                         <input type = "text" name="address" id="
                               address" placeholder="R./Av."/>
108                         <label for= "address">Endere&ccedil;o</
                               label>
109                 </p>
110 </section>
```

```
111
112 <section class = "one-third">
113         <p class = "birthday">
114                         <input type = "date" name= "birthdate"
                               placeholder= "DD/MM/AAAA" style = "
                               width:100px"></>
115                         <!--<input type = "text" name= "month"
                               placeholder= "MM" style = "width:25px"
                               ></>
116                         <input type = "text" name= "year"
                               placeholder= "AAAA" style = "width:70px
                               "></>-->
117                         <label for="text">Data de Nascimento*</
                               label>
118                 </p>
119 </section>
120
121 <section class="one-third">
122         <p class = "estacao">
123         <select name="estacao" style= "height: 40px">
124         <option value="opcao"/>Selecione uma Op&ccedil;&atilde;o
              </option>
125         <option value="0"/>Laranjeiras</option>
126         <option value="1"/>Centro-Vitoria</option>
127         <option value="2"/>Carapina</option>
128         <option value="3"/>J. Camburi</option>
129         <option value="4"/>Enseada do Sua</option>
130         <option value="5"/>Ibes</option>
131         <option value="6"/>VV-Centro</option>
132         <option value="7"/>Cariacica</option>
133         <option value="8"/>Cidade Continental</option>
134         </select>
135         <label for="estacao">Esta&ccedil;&atilde;o*</label>
136 </section>
137
138 <section class="one-third">
139         <p class = "health">
140         <select name="health" style= "height: 40px">
141         <option value="opcao"/>Selecione uma Op&ccedil;&atilde;o
              </option>
142         <option value="Respiratorio"/>Problemas Respirat&oacute;
              rios</option>
```

```
143         <option value="Cardiacos"/>Problemas Card&iacute;acos</
               option>
144         <option value="Ambos"/>Ambos</option>
145         <option value="Nenhum"/>Nenhum</option>
146         </select>
147         <label for="health">Condi&ccedil;&atilde;o de Sa&uacute;
               de*</label>
148 </section>
149
150
151 <div class="clearfix" style= "color: #FF0000;">
152 <?php if( !empty($_SESSION['STR_ERRO'])){
153         print($_SESSION['STR_ERRO']);
154         $_SESSION['STR_ERRO'] = ""; } ?></div>
155                 <p class = "submit">
156                         <input type = "submit" style = "
                            background-color:#87CEEB" value = "
                            Enviar">
157                 </p>
158
159                 </form>
160                 </div>
161
162 <div class="clearfix"></div>
163 <!--Start Parallax Section-->
164         <section class="parallax-1">
165                 <div class="parallax-inner">
166                         <section class="one-half">
167                                 <h3>Sobre N&oacute;s</h3>
168                                 <p> Esse projeto busca atender
                                    uma demanda do IEMA dentro de
                                    um plano de colabora&ccedil;&
                                    atilde;o cont&iacute;nua com o
                                    grupo de pesquisadores dos
                                    Departamentos de Engenharia El&
                                    eacute;trica e Engenharia
                                    Ambiental da Universidade
                                    Federal do Esp&iacute;rito
                                    Santo (UFES). O desenvolvimento
                                     deste website e dos
                                    aplicativos fazem parte do
                                    Projeto de Gradua&ccedil;&
```

```
                                    atilde;o da aluna do curso de
                                    Engenharia El&eacute;trica
                                    Mayara Nascimento de Oliveira.
                                    A aluna &eacute; orientanda do
                                    Prof. Dr. Teodiano Freire
                                    Bastos do Departamento de
                                    Engenharia El&eacute;trica e da
                                     Prof &ordf;. Dr &ordf;. Jane M
                                    . Santos do Departamento de
                                    Engenharia Ambiental </p>
169                         </section>
170                         <section class="one-half">
171                                 <h3>IEMA</h3>
172                                 <p> O IEMA &eacute; o Instituto
                                    Estadual de Meio Ambiente e
                                    Recursos H&iacute;dricos
                                    respons&aacute;vel pela execu&
                                    ccedil;&atilde;o das pol&iacute
                                    ;ticas estaduais de meio
                                    ambiente. Suas atribui&ccedil;&
                                    otilde;es incluem o
                                    monitoramento, fiscaliza&ccedil
                                    ;&atilde;o, pesquisa, trabalhos
                                     de educa&ccedil;&atilde;o
                                    ambiental e o licenciamento de
                                    empreendimentos que realizam
                                    atividades potencialmente
                                    poluidoras </p>
173                         </section>
174                 </div>
175         </section>
176
177
178         <div class="clearfix"></div>
179 <footer>
180 <p> &copy; Copyright infoAR - 2016 - Todos os Direitos Reservados
        </p>
181
182 </footer>
183
184 </body>
185 </html>
```

A.5.1 Código PHP para Salvar os Dados

```php
<?php

session_start();

error_reporting(E_ALL);
ini_set('display_errors', 1);


$servername = 'localhost';
$username = 'infoar';
$password = 'baaIC!30E';
$dbname = 'infoar';

$health_cond = $station = "opcao";
$name = $email = $birthdate = "";

$name= test_input($_POST['name']);
if (!preg_match("/^[a-zA-Z ]*$/", $name)) {
  $_SESSION['STR_ERRO']  = 'Formato de nome áinvlido. Apenas
     letras e çespaos em branco ãso permitidos';
  header("Location: ./cadastrese.php#cadastrese");
  exit(1);
}

$email= test_input($_POST['email']);
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
  $_SESSION['STR_ERRO']  = 'Formato de email áinvlido';
  header("Location: ./cadastrese.php#cadastrese");
  exit(1);
}

$address= test_input($_POST['address']);
if (!preg_match("/^[0-9a-zA-Z. ]+$/",$address)) {
  $_SESSION['STR_ERRO']  = 'Formato de çendereo áinvlido. Apenas
     letras, únmeros e çespaos em branco ãso permitidos';
  header("Location: ./cadastrese.php#cadastrese");
  exit(1);
}

$birthdate= test_input($_POST['birthdate']);
```

```
39 $birthdate = implode("-", array_reverse(explode("/", $birthdate))
      );
40 if(!preg_match('/^([0-9\-]+)$/', $birthdate)){
41          $_SESSION['STR_ERRO']  = 'Data de áaniversrio áinvlida';
42           header("Location: ./cadastrese.php#cadastrese");
43         exit(1);
44 }
45
46 $health_cond= test_input($_POST['health']);
47 if(($health_cond != "Respiratorio") && ($health_cond != "
      Cardiacos") && ($health_cond != "Ambos") && ($health_cond != "
      Nenhum") &&($health_cond != "opcao")){
48          $_SESSION['STR_ERRO']  =  'çãCondio de úsade áinvlida';
49          header("Location: ./cadastrese.php#cadastrese");
50         exit(1);
51 }
52
53 $station = test_input($_POST['estacao']);
54 if(($station != "0") && ($station != "1") && ($station != "2") &&
       ($station != "3")&& ($station != "4") && ($station != "5") &&
      ($station != "6") && ($station != "7") && ($station != "8") &&
      ($station != "opcao")){
55          $_SESSION['STR_ERRO']  =   ' A çãestao selecionada éá
              invlida';
56          header("Location: ./cadastrese.php#cadastrese");
57         exit(1);
58 }
59
60
61
62 if((empty($name)) || (empty($email)) || (empty($birthdate)) || (
      $health_cond == "opcao") || ($station=="opcao")){
63         $_SESSION['STR_ERRO']  =  'Desculpe, áh problemas no
              áformulrio. Confira suas çõinformaes e preencha os
              campos óobrigatrios adequadamente';
64         header("Location: ./cadastrese.php#cadastrese");
65         exit(1);
66 }
67
68
69 function test_input($data){
70         $data = trim($data); // Remove espacos em branco na
```

```
            frente e atras da string
71          $data = stripslashes($data); //Remova aspas de uma string
72          $data = htmlspecialchars($data); // Converte caracteres
               speciais em entidades html
73          return $data;
74  }
75
76
77
78
79
80  $conn = mysql_connect($servername, $username, $password);
81  if($conn){
82          echo '';
83  }else {
84          die("A ãconexo falhou:".mysql_connect_error());
85  }
86
87  $sql1 = "INSERT INTO Pessoa (Nome, email, address, birthday,
       health_cond)
88                  VALUES ('$name','$email','$address', '$birthdate
                       ', '$health_cond')";
89
90  $sql2 = "INSERT INTO Pessoa_Estacao(id_pessoa, id_estacao)
91                  VALUES (last_insert_id(), '$station')";
92
93  mysql_select_db('infoar');
94
95  if(!mysql_query($sql1,$conn)){
96
97          die ('Erro:'.mysql_error());
98
99  }else {
100         echo '';
101 }
102
103 if(!mysql_query($sql2,$conn)){
104
105         die ('Erro:'.mysql_error());
106
107 }else {
108         echo 'Dados enviados com sucesso';
```

```
109 }
110
111 mysql_close($conn);
112
113
114 ?>
```

A.6 Código HTML da Página Fale Conosco

```
1
2 <?php
3 session_start();
4 ?>
5
6 <!DOCTYPE html>
7 <head>
8         <meta name="viewport" content="width=device-width,
             initial-scale=1"> <!-- for mobile version -->
9         <title> infoAR - Fale Conosco</title> <!-- what apperars
             when you search on Google or on the top of the tab -->
10        <link href="css/jquery.bxslider.css" rel="stylesheet" />
11        <link href="style.css" rel="stylesheet">
12        <link href="css/nav.css" rel="stylesheet">
13        <link rel="shortcut icon" type="image/png" href="img/
             icone.png"/> <!-- logo 40x40 -->
14        <style>
15        #map {
16        height: 450px;
17        width: 80%;
18        margin: 2% 4% 10% 10%;-->
19
20       }
21    </style>
22 </head>
23 <body>
24        <a name="home">
25 <!--- Start Navigation -->
26                <script src="js/jquery-1.11.2.min.js"></script>
27                <script src="js/main.js"></script> <!--- For
                     Navigation -->
28
```

```
<!---ADD NAVIGATION HERE-->
<div class= "nav-outer">
<div class="nav-wrap">
        <nav class="navigation">
                        <div class="logo"><img src = "img/
                           logobranco.png"></div>
                                <div class='nav' nav-menu-style =
                                    "
                                yoga">
                                        <ul class = "nav-menu">
                                                <li><a href= "
                                                   index.php">Home
                                                   </a></li> <!--
                                                   linkar com
                                                   outras webpages
                                                   -->
                                                <li><a href="info
                                                   .php">Info</a
                                                   ></li>
                                                <li><a href="
                                                   poluentes.php">
                                                   Poluentes</a></
                                                   li>
                                                <li><a href="
                                                   cadastrese.php"
                                                   >Cadastre-se</a
                                                   ></li>
                                                <li><a href="
                                                   faleconosco.php
                                                   ">Fale Conosco
                                                   </a></li>
                                </div>
        </nav>
</div>
</div>
        <div class="nav-clear"></div>
<!--- End Navigation -->
        <div class="clearfix"></div>
<!--- Start Slider -->
                <script src="js/jquery.bxslider.min.js"></script
                   ><!--For Image Slider-->
                <div class="slide-wrap">
```

```
52
53 <!--ADD IMAGE SLIDES HERE-->
54 <section class="slider">
55         <ul class="slider1">
56                 <li><img src="img/capa1.png"></li>
57                 <li><img src="img/capa2.png"></li>
58         </ul>
59 </section>
60                 </div>
61                 <script type="text/javascript">
62                         $('.slider1').bxSlider({
63                                 mode: 'fade',
64                                 captions: false,
65                                 auto:true,
66                                 pager:false,
67
68                         });
69                         $('.slider2').bxSlider({
70                                 pager:false,
71                                 captions: true,
72                                 maxSlides: 3,
73                                 minSlides: 1,
74                                 slideWidth: 230,
75                                 slideMargin: 10
76                         });
77
78                 </script>
79
80
81 <!--- End Slider -->
82         <!--Start Banner Wrapper-->
83         <div id="banner-wrapper">
84         <h1><a href="#faleconosco" style="color: #4A4444; text-
              decoration: none;">D&uacute;vidas e Sugest&otilde;es
              para o Website?</a></h1>
85         <a name = "faleconosco">
86         <h3>Deixe sua Mensagem</h3>
87
88         <form class ="form" method = "post" action="send_email.
              php">
89         <section class = "one-third">
90                 <p class = "name">
```

```
91                          <input type = "text" name="name" id="name
                               " placeholder="Maria Silva"/>
92                          <label for= "name">Nome*</label>
93                  </p>
94  </section>
95  <section class = "one-third">
96          <p class = "email">
97                          <input type = "text" name="email" id="
                               email" placeholder="mail@example.com"/>
98                          <label for= "email">Email*</label>
99                  </p>
100 </section>
101 <section class = "one-third">
102         <p class = "phone">
103                         <input type = "text" name="phone" id="
                               phone" placeholder="DDD XXXXXXXXX"/>
104                         <label for= "phone">Telefone*</label>
105                 </p>
106 </section>
107         <p class = "text">
108                         <textarea name= "text" placeholder= "
                               Escreva uma mensagem" maxlength="1000"
                               ></textarea>
109                         <label for="text">Mensagem*</label>
110                 </p>
111
112 <div class="clearfix" style= "color: #FF0000;">
113 <?php if( !empty($_SESSION['STR_ERRO'])){
114         print($_SESSION['STR_ERRO']);
115         $_SESSION['STR_ERRO'] = ""; } ?></div>
116                 <p class = "submit">
117                         <input type = "submit"  style = "
                               background-color:#87CEEB" value = "
                               Enviar">
118                 </p>
119                 </form>
120                 </div>
121
122 <div class="clearfix"></div>
123 <!--Start Parallax Section-->
124         <section class="parallax-1">
125                 <div class="parallax-inner">
```

```
126                         <section class="one-half">
127                                 <h3>Sobre N&oacute;s</h3>
128                                 <p> Esse projeto busca atender
                                       uma demanda do IEMA dentro de
                                       um plano de colabora&ccedil;&
                                       atilde;o cont&iacute;nua com o
                                       grupo de pesquisadores dos
                                       Departamentos de Engenharia El&
                                       eacute;trica e Engenharia
                                       Ambiental da Universidade
                                       Federal do Esp&iacute;rito
                                       Santo (UFES). O desenvolvimento
                                        deste website e dos
                                       aplicativos fazem parte do
                                       Projeto de Gradua&ccedil;&
                                       atilde;o da aluna do curso de
                                       Engenharia El&eacute;trica
                                       Mayara Nascimento de Oliveira.
                                       A aluna &eacute; orientanda do
                                       Prof. Dr. Teodiano Freire
                                       Bastos do Departamento de
                                       Engenharia El&eacute;trica e da
                                        Prof &ordf;. Dr &ordf;. Jane M
                                       . Santos do Departamento de
                                       Engenharia Ambiental </p>
129                         </section>
130                         <section class="one-half">
131                                 <h3>IEMA</h3>
132                                 <p> O IEMA &eacute; o Instituto
                                       Estadual de Meio Ambiente e
                                       Recursos H&iacute;dricos
                                       respons&aacute;vel pela execu&
                                       ccedil;&atilde;o das pol&iacute
                                       ;ticas estaduais de meio
                                       ambiente. Suas atribui&ccedil;&
                                       otilde;es incluem o
                                       monitoramento, fiscaliza&ccedil
                                       ;&atilde;o, pesquisa, trabalhos
                                        de educa&ccedil;&atilde;o
                                       ambiental e o licenciamento de
                                       empreendimentos que realizam
                                       atividades potencialmente
```

```
                                        poluidoras </p>
133                         </section>
134                 </div>
135         </section>
136
137
138         <div class="clearfix"></div>
139 <footer>
140 <p> &copy; Copyright infoAR - 2016 - Todos os Direitos Reservados
        </p>
141
142 </footer>
143
144 </body>
145 </html>
```

A.6.1 Código PHP para Enviar os Emails

```
1  <?php
2
3  session_start();
4
5  require_once('phpmailer/class.phpmailer.php');
6  require_once('phpmailer/class.smtp.php');
7
8  //error_reporting(E_ALL);
9  //ini_set('display_errors', 1);
10
11 if(isset($_POST['email'])){
12
13         $email_to = "infoarapp@gmail.com";
14         $email_subject = "Duvidas e Sugestoes Website";
15
16         $name = ($_POST['name']);
17         $email_from = ($_POST['email']);
18         $phone = ($_POST['phone']);
19         $text = ($_POST['text']);
20
21
22         if((empty($name))|| (empty($email_from))|| (empty($phone)
               )||(empty($text))) {
23                 $_SESSION['STR_ERRO']  = 'Desculpe, áh problemas
```

```
                    no áformulrio. Confira suas çõinformaes e
                    preencha todos os campos adequadamente';
24                  header("Location: ./faleconosco.php#faleconosco"
                       );
25                 exit(1);
26          }
27
28          $error_message = "";
29
30      $email_exp = '/^[A-Za-z0-9._%-]+@[A-Za-z0-9.-]+\.[A-Za-z
           ]{2,4}$/';
31
32  if(!preg_match($email_exp,$email_from)) {
33
34  $_SESSION['STR_ERRO']  = 'O email digitado ãno é valido.<br />';
35  header("Location: ./faleconosco.php#faleconosco");
36  exit(1);
37
38  }
39
40      $string_exp = "/^[A-Za-z .'-]+$/";
41
42
43  if(strlen($text) < 2) {
44
45      $_SESSION['STR_ERRO']  =  'Seu ácomentrio ãno é valido.<br />'
           ;
46          header("Location: ./faleconosco.php#faleconosco");
47          exit(1);
48  }
49
50  /*if(strlen($error_message) > 0) {
51
52      print($error_message);
53      exit(1);
54
55  }*/
56
57          $email_message = " Os detalhes do áformulrio seguem abaixo
               :\n\n";
58
59          function clean_string($string) {
```

```
60
61        $bad = array("content-type","bcc:","to:","cc:","href");
62
63        return str_replace($bad,"",$string);
64
65      }
66
67      $email_message.= "Nome: ".clean_string($name)."\n";
68      $email_message.= "Email ".clean_string($email_from)."\n";
69      $email_message.= "Telefone ".clean_string($phone)."\n";
70      $email_message.= "Mensagem: ".clean_string($text)."\n";
71
72
73  function smtpmailer($to, $from, $from_name, $subject, $body) {
74          global $error;
75          $mail = new PHPMailer();  // create a new object
76          $mail->IsSMTP(); // enable SMTP
77          $mail->SMTPDebug = 0;  // debugging: 1 = errors and
               messages, 2 = messages only
78          $mail->SMTPAuth = true;  // authentication enabled
79          $mail->SMTPSecure = 'ssl'; // secure transfer enabled
               REQUIRED for GMail
80          $mail->Host = 'smtp.gmail.com';
81          $mail->Port = 465;
82          $mail->Username = 'infoarapp@gmail.com';
83          $mail->Password = 'infoar2016';
84          $mail->SetFrom($from, $from_name);
85          $mail->Subject = $subject;
86          $mail->Body = $body;
87          $mail->AddAddress($to);
88          if(!$mail->Send()) {
89                  $error = 'Mail error: '.$mail->ErrorInfo;
90                  print($error);
91                  return false;
92          } else {
93                  $error = 'Message sent!';
94                  return true;
95          } print($error);
96          print('aqui');
97  }
98
99
```

```
100 if(smtpmailer($email_to, $email_from, $name, $email_subject,
        $email_message)){
101         echo 'Sua mensagem foi enviada com sucesso!';
102 } else {
103         echo 'Houve um problema ao enviar sua mensagem, por favor
                tente novamente';
104 }
105
106
107 }
108
109 ?>
```

ANEXO B – BANCO DE DADOS

B.1 Arquivo Com Dados dos Poluentes

```
1
2  *#dd/mm/yyyy#hh:mm#;#.#MIG46-07/0008-55#
3  13/01/2015 00:30;E01P03;19.4301986694336;;;
4  13/01/2015 00:30;E01P04;201.45002746582;;;
5  13/01/2015 00:30;E01P07;328.545288085938;;;
6  13/01/2015 00:30;E01P08;25.2625617980957;;;
7  13/01/2015 00:30;E01P01;45.7401504516602;;;
8  13/01/2015 00:30;E02P01;17.5403594970703;;;
9  13/01/2015 00:30;E02P03;2.63093328475952;;;
10 13/01/2015 00:30;E02P04;21.7109775543213;;;
11 13/01/2015 00:30;E02P07;67.4326324462891;;;
12 13/01/2015 00:30;E02P08;28.8119964599609;;;
13 13/01/2015 00:30;E03P01;23.4591102600098;;;
14 13/01/2015 00:30;E03P03;2.69021677970886;;;
15 13/01/2015 00:30;E03P04;9.6090784072876;;;
16 13/01/2015 00:30;E03P07;30.0829429626465;;;
17 13/01/2015 00:30;E04P01;12.2343053817749;;;
18 01/12/2016 00:30;E04P03;4.27809286117554;;;
19 01/12/2016 00:30;E04P04;26.4096069335938;;;
20 13/01/2015 00:30;E04P07;182.311935424805;;;
21 13/01/2015 00:30;E04P08;43.720947265625;;;
22 01/12/2016 00:30;E05P01;31.1123943328857;;;
23 01/12/2016 00:30;E05P03;158.282623291016;;;
24 01/12/2016 00:30;E05P04;134.500030517578;;;
25 01/12/2016 00:30;E05P07;289.716583251953;;;
26 01/12/2016 00:30;E06P01;31.1126937866211;;;
27 13/01/2015 00:30;E06P03;41.4441337585449;;;
28 13/01/2015 00:30;E06P04;17.2288970947266;;;
29 13/01/2015 00:30;E06P07;575.034606933594;;;
30 13/01/2015 00:30;E06P08;18.2139282226563;;;
31 13/01/2015 00:30;E07P01;45.7401504516602;;;
32 13/01/2015 00:30;E07P03;19.4301986694336;;;
33 13/01/2015 00:30;E07P04;215.114379882813;;;
34 13/01/2015 00:30;E07P07;326.286560058594;;;
35 13/01/2015 00:30;E07P08;27.7326793670654;;;
```

```
13/01/2015 00:30;E08P01;65.5516357421875;;;
13/01/2015 00:30;E08P03;0.145851016044617;;;
13/01/2015 00:30;E08P04;11.8252630233765;;;
13/01/2015 00:30;E08P07;177.525863647461;;;
13/01/2015 00:30;E08P08;40.3788070678711;;;
```

B.2 Código PHP para Armazenar Informações no Banco de Dados

```
1
2  <?php
3
4  //session_start();
5
6  error_reporting(E_ALL);
7  ini_set('display_errors', 1);
8
9
10 $servername = 'localhost';
11 $username = 'infoar';
12 $password = 'baaIC!30E';
13 $dbname = 'infoar';
14
15 print('HELLO');
16 $conn = mysql_connect($servername, $username, $password);
17 if($conn){
18         echo '';
19 }else {
20         die("A ãconexo falhou:".mysql_connect_error());
21 }
22
23 $file = file('Arquivo_Teste_Inicial.txt'); // vai ler os arquivos
       em um vetor
24 $count = count($file); // conta os numeros de caracteres no
      arquivo
25
26 if($count > 0){ // se o arquivo nao estiver vazio
27
28         $i = 1;
29         foreach($file as $row){
30                 $data = explode(';', $row);
31                 if(($data[1] == "E01P01") || ($data[1] == "E01P03
```

```
                    ") || ($data[1] == "E01P04") ||($data[1] == "
                    E01P07") || ($data[1] == "E01P08") || ($data[1]
                     == "E01P18")) {
32                      $id_estacao = 9;
33                  } else if (($data[1] == "E02P01") || ($data[1] ==
                     "E02P03") || ($data[1] == "E02P04") ||($data
                    [1] == "E02P07") || ($data[1] == "E02P08") || (
                    $data[1] == "E02P18")) {
34                      $id_estacao = 2;
35                  } else if (($data[1] == "E03P01") || ($data[1] ==
                     "E03P03") || ($data[1] == "E03P04") ||($data
                    [1] == "E03P07") || ($data[1] == "E03P08") || (
                    $data[1] == "E03P18")) {
36                      $id_estacao = 3;
37                  } else if (($data[1] == "E04P01") || ($data[1] ==
                     "E04P03") || ($data[1] == "E04P04") ||($data
                    [1] == "E04P07") || ($data[1] == "E04P08") || (
                    $data[1] == "E04P18")) {
38                      $id_estacao = 4;
39                  } else if (($data[1] == "E05P01") || ($data[1] ==
                     "E05P03") || ($data[1] == "E05P04") ||($data
                    [1] == "E05P07") || ($data[1] == "E05P08") || (
                    $data[1] == "E05P18")) {
40                      $id_estacao = 1;
41                  } else if (($data[1] == "E06P01") || ($data[1] ==
                     "E06P03") || ($data[1] == "E06P04") ||($data
                    [1] == "E06P07") || ($data[1] == "E06P08") || (
                    $data[1] == "E06P18")) {
42                      $id_estacao = 5;
43                  } else if (($data[1] == "E07P01") || ($data[1] ==
                     "E07P03") || ($data[1] == "E07P04") ||($data
                    [1] == "E07P07") || ($data[1] == "E07P08") || (
                    $data[1] == "E07P18")) {
44                      $id_estacao = 6;
45                  } else if (($data[1] == "E08P01") || ($data[1] ==
                     "E08P03") || ($data[1] == "E08P04") ||($data
                    [1] == "E08P07") || ($data[1] == "E08P08") || (
                    $data[1] == "E08P18")) {
46                      $id_estacao = 7;
47                  } else if (($data[1] == "E09P01") || ($data[1] ==
                     "E09P03") || ($data[1] == "E09P04") ||($data
                    [1] == "E09P07") || ($data[1] == "E09P08") || (
```

```
                    $data[1] == "E09P18")) {
48                        $id_estacao = 8;
49                  }
50
51
52                  if(($data[1] == "E01P01") || ($data[1] == "E02P01
                    ") || ($data[1] == "E03P01") || ($data[1] == "
                    E04P01") || ($data[1] == "E05P01") || ($data[1]
                     == "E06P01") || ($data[1] == "E07P01") || (
                    $data[1] == "E08P01") || ($data[1] == "E09P01")
                    ) {
53                        $poluentes = "Particulas Inalaveis- 10um"
                             ;
54                        if(($data[2] <50)){
55                              $iqa = ($data[2]*40)/50;
56                        } else if(($data[2] >50) && ($data[2]
                             <100)){
57                              $iqa = ($data[2]*80)/100;
58                        } else if(($data[2] >100) && ($data[2]
                             <150)){
59                              $iqa = ($data[2]*120)/150;
60                        }  else if(($data[2] >150)){
61                              $iqa = ($data[2]*200)/250;
62                        }
63
64                  }
65                  else if(($data[1] == "E01P03") || ($data[1] == "
                    E02P03") || ($data[1] == "E03P03") || ($data[1]
                     == "E04P03") || ($data[1] == "E05P03") || (
                    $data[1] == "E06P03") || ($data[1] == "E07P03")
                     || ($data[1] == "E08P03") || ($data[1] == "
                    E09P03")) {
66                        $poluentes = "SO2";
67                        if(($data[2] <20)){
68                              $iqa = ($data[2]*40)/20;
69                        } else if(($data[2] >20) && ($data[2]
                             <40)){
70                              $iqa = ($data[2]*80)/40;
71                        } else if(($data[2] >40) && ($data[2]
                             <365)){
72                              $iqa = ($data[2]*120)/365;
73                        }  else if(($data[2] >365)){
```

```
74                                  $iqa = ($data[2]*200)/800;
75                          }
76
77                  }
78                  else if(($data[1] == "E01P04") || ($data[1] == "
                        E02P04") || ($data[1] == "E03P04") || ($data[1]
                         == "E04P04") || ($data[1] == "E05P04") || (
                        $data[1] == "E06P04") || ($data[1] == "E07P04")
                         || ($data[1] == "E08P04") || ($data[1] == "
                        E09P04")) {
79                          $poluentes = "NO2";
80                          if(($data[2] <200)){
81                                  $iqa = ($data[2]*40)/200;
82                          } else if(($data[2] >200) && ($data[2]
                                <240)){
83                                  $iqa = ($data[2]*80)/240;
84                          } else if(($data[2] >240) && ($data[2]
                                <320)){
85                                  $iqa = ($data[2]*120)/320;
86                          }  else if(($data[2] >320)){
87                                  $iqa = ($data[2]*200)/1130;
88                          }
89
90                  }       else if(($data[1] == "E01P07") || ($data
                        [1] == "E02P07") || ($data[1] == "E03P07") || (
                        $data[1] == "E04P07") || ($data[1] == "E05P07")
                         || ($data[1] == "E06P07") || ($data[1] == "
                        E07P07") || ($data[1] == "E08P07") || ($data[1]
                         == "E09P07")) {
91                          $poluentes = "CO";
92                          $data[2] = (($data[2]*0.001)*(24.45))/28;
93                          if(($data[2] <9)){
94                                  $iqa = ($data[2]*40)/9;
95                          } else if(($data[2] >9) && ($data[2] <11)
                                ){
96                                  $iqa = ($data[2]*80)/11;
97                          } else if(($data[2] >11) && ($data[2]
                                <13)){
98                                  $iqa = ($data[2]*120)/13;
99                          }  else if(($data[2] >13)){
100                                 $iqa = ($data[2]*200)/15;
101                         }
```

```
102
103             }       else if(($data[1] == "E01P08") || ($data
                    [1] == "E02P08") || ($data[1] == "E03P08") || (
                    $data[1] == "E04P08") || ($data[1] == "E05P08")
                     || ($data[1] == "E06P08") || ($data[1] == "
                    E07P08") || ($data[1] == "E08P08") || ($data[1]
                     == "E09P08")) {
104                 $poluentes = "O3";
105                 if(($data[2] <100)){
106                         $iqa = ($data[2]*40)/100;
107                 } else if(($data[2] >100) && ($data[2]
                        <130)){
108                         $iqa = ($data[2]*80)/130;
109                 } else if(($data[2] >130) && ($data[2]
                        <160)){
110                         $iqa = ($data[2]*120)/160;
111                 }  else if(($data[2] >160)){
112                         $iqa = ($data[2]*200)/200;
113                 }
114
115             }       else if(($data[1] == "E01P18") || ($data
                    [1] == "E02P18") || ($data[1] == "E03P18") || (
                    $data[1] == "E04P18") || ($data[1] == "E05P18")
                     || ($data[1] == "E06P18") || ($data[1] == "
                    E07P18") || ($data[1] == "E08P18") || ($data[1]
                     == "E09P18")) {
116                 $poluentes = "Particulas Inalaveis - 2.5"
                        ;
117                 if(($data[2] <100)){
118                         $iqa = ($data[2]*40)/100;
119                 } else if(($data[2] >100) && ($data[2]
                        <130)){
120                         $iqa = ($data[2]*80)/130;
121                 } else if(($data[2] >130) && ($data[2]
                        <160)){
122                         $iqa = ($data[2]*120)/160;
123                 }  else if(($data[2] >160)){
124                         $iqa = ($data[2]*200)/200;
125                 }
126
127             }
128
```

```
129                 $data_query = "INSERT INTO Data (id_estacao,
                        poluente, concentracao, iqa) VALUES ('
                        $id_estacao, $poluentes', '$data[2]', '$iqa') "
                        ;
130
131                 $data_query .= $i < $count ? ',' : '';
132                 $i++;
133         }
134         mysql_query($data_query) or die (mysql_error());
135 }
136
137 ?>
```

B.3 Código para Criar o arquivo JSON

```
1
2 <?php
3
4 //session_start();
5
6 error_reporting(E_ALL);
7 ini_set('display_errors', 1);
8
9
10 $servername = 'localhost';
11 $username = 'infoar';
12 $password = 'baaIC!30E';
13 $dbname = 'infoar';
14
15 $conn = mysql_connect($servername, $username, $password);
16 if($conn){
17         echo '';
18 }else {
19         die("A ãconexo falhou:".mysql_connect_error());
20 }
21
22 $sql = "select * from Data";
23 $result = mysql_query($conn, $sql) or die("Error in Selecting " .
       mysql_error($conn));
24
25 $emparray = array();
```

```
26      while($row =mysqli_fetch_assoc($result)) {
27          $emparray[] = $row;
28      }
29
30  $fp = fopen('empdata.json', 'w');
31      fwrite($fp, json_encode($emparray));
32      fclose($fp);
33
34  mysql_close($conn);
35
36  ?>
```

ANEXO C – CÓDIGOS APLICATIVO IOS

C.1 Arquivo com a Localização das Estações

```
1
2  //
3  //  Config.swift
4  //  infoAR
5  //
6  //  Created by Mayara Oliveira on 2016-10-26.
7  //  Copyright © 2016 Mayara. All rights reserved.
8  //
9
10 import Foundation
11
12   let locations = [[-20.196, -40.245], [-20.226,
        -40.259],[-20.255, -40.270],[-20.313, -40.291],[-20.321,
        -40.333],[-20.348, -40.317],[-20.336, -40.291],[-20.342,
        -40.402], [-20.225, -40.218]]
13
14    var names: [String ] = ["Laranjeiras", "Carapina", "J. Camburi
        ", "Enseada do áSu", "Centro-óVitria", "Ibes", "VV-Centro",
        "Cariacica", "Cidade Continental"]
```

C.2 Arquivo *AppDelegate*

```
1
2  //
3  //  AppDelegate.swift
4  //  infoAR
5  //
6  //  Created by Mayara Oliveira on 2016-10-26.
7  //  Copyright © 2016 Mayara. All rights reserved.
8  //
9
10 import UIKit
11 import GoogleMaps
12
13 @UIApplicationMain
```

```
14 class AppDelegate: UIResponder, UIApplicationDelegate {
15
16     var window: UIWindow?
17
18
19     func application(_ application: UIApplication,
           didFinishLaunchingWithOptions launchOptions: [
           UIApplicationLaunchOptionsKey: Any]?) -> Bool {
20         GMSServices.provideAPIKey("AIzaSyD08jp-xGnEQI-
               T_1Pk24g3iOuxwIY45qY")
21         return true
22     }
23
24     func applicationWillResignActive(_ application: UIApplication
           ) {
25         // Sent when the application is about to move from active
               to inactive state. This can occur for certain types of
               temporary interruptions (such as an incoming phone
               call or SMS message) or when the user quits the
               application and it begins the transition to the
               background state.
26         // Use this method to pause ongoing tasks, disable timers
               , and invalidate graphics rendering callbacks. Games
               should use this method to pause the game.
27     }
28
29     func applicationDidEnterBackground(_ application:
           UIApplication) {
30         // Use this method to release shared resources, save user
               data, invalidate timers, and store enough application
               state information to restore your application to its
               current state in case it is terminated later.
31         // If your application supports background execution,
               this method is called instead of
               applicationWillTerminate: when the user quits.
32     }
33
34     func applicationWillEnterForeground(_ application:
           UIApplication) {
35         // Called as part of the transition from the background
               to the active state; here you can undo many of the
               changes made on entering the background.
```

```
36      }
37
38      func applicationDidBecomeActive(_ application: UIApplication)
            {
39          // Restart any tasks that were paused (or not yet started
               ) while the application was inactive. If the
               application was previously in the background,
               optionally refresh the user interface.
40      }
41
42      func applicationWillTerminate(_ application: UIApplication) {
43          // Called when the application is about to terminate.
               Save data if appropriate. See also
               applicationDidEnterBackground:.
44      }
45
46
47  }
```

C.3 Arquivo *ViewController.swift*

```
1
2   //
3   //  ViewController.swift
4   //  infoAR
5   //
6   //  Created by Mayara Oliveira on 2016-10-26.
7   //  Copyright © 2016 Mayara. All rights reserved.
8   //
9
10  import UIKit
11  import GoogleMaps
12
13  var usersLocation: CLLocation?
14
15  class ViewController: UIViewController, GMSMapViewDelegate,
        CLLocationManagerDelegate {
16
17
18      @IBOutlet weak var mapView: GMSMapView!
19
```

```
20      var locationManager = CLLocationManager()
21
22      override func viewDidLoad() {
23          super.viewDidLoad()
24      mapView.delegate = self
25       locationManager.delegate = self
26          locationManager.requestWhenInUseAuthorization()
27
28          var cont: Int = 0
29         for location in locations{
30             let coordinate = CLLocationCoordinate2DMake(location
                   [0] , location[1] )
31              let marker = GMSMarker(position:coordinate )
32              marker.map = mapView
33              marker.title = names[cont];
34              marker.icon = UIImage(named: "verde")
35          cont += 1;
36          }
37
38      let camera = GMSCameraPosition.camera(withLatitude: -20.27,
           longitude: -40.30, zoom: 12)
39      mapView.camera = camera
40
41      /*let marker = GMSMarker()
42          marker.position = CLLocationCoordinate2DMake(-20.27,
               -40.30)
43          marker.title = "Sydney"
44          marker.snippet = "Australia"
45          marker.map = mapView*/
46
47
48      }
49
50      override func prepare(for segue: UIStoryboardSegue, sender:
           Any?) {
51          let backItem = UIBarButtonItem()
52          backItem.title = "Voltar"
53          navigationItem.backBarButtonItem = backItem
54      }
55
56      func locationManager(_ manager: CLLocationManager,
           didChangeAuthorization status: CLAuthorizationStatus) {
```

```
57
58          if status == .authorizedWhenInUse{
59              locationManager.startUpdatingLocation()
60
61              mapView.isMyLocationEnabled = true // do googlemaps
62          }
63      }
64
65      func locationManager(_ manager: CLLocationManager,
           didUpdateLocations locations: [CLLocation]) {
66          if let location = locations.first{
67              usersLocation = location
68              mapView.camera = GMSCameraPosition(target:
                   usersLocation!.coordinate, zoom: 13, bearing:0,
                   viewingAngle: 0)
69              locationManager.stopUpdatingLocation()
70          }
71          else {
72              let alert = UIAlertController(title: "Erro", message:
                    "çãLocalizao ãno obtida", preferredStyle: .alert)
73              let action = UIAlertAction(title: "OK", style: .
                   default, handler: nil)
74
75              alert.addAction(action)
76              self.present(alert, animated: true, completion: nil)
77          }
78      }
79
80      func createMarker(number: String, place:
           CLLocationCoordinate2D, name: String){
81          let marker = GMSMarker(position: place)
82          marker.title = name
83          let uiview = UIView(frame: CGRect(x: 0, y: 0, width:
               56.0, height: 36.0))
84          let imageview = UIImageView(image: UIImage(named: "verde"
               ))
85
86          let label = UILabel()
87          label.frame = CGRect(x: 0, y: 0, width: 56.0, height:
               36.0)
88          label.textAlignment = .center
89          label.backgroundColor = .clear
```

```
90          label.textColor = .white
91          label.font = label.font.withSize(11)
92          label.text = number
93
94          uiview.addSubview(imageview)
95          uiview.addSubview(label)
96
97          marker.icon = self.imageForView(view: uiview)
98          marker.title = name
99          marker.map = mapView
100     }
101
102
103     func imageForView(view: UIView)->UIImage{
104         UIGraphicsBeginImageContextWithOptions(view.frame.size,
               false, UIScreen.main.scale)
105
106         if let currentContext = UIGraphicsGetCurrentContext(){
107             view.layer.render(in:currentContext)
108                 let image =
                       UIGraphicsGetImageFromCurrentImageContext()
109                 UIGraphicsEndImageContext()
110                 return image!
111         }
112
113         else{
114             return UIImage()
115         }
116
117     }
118
119     //func evaluateNumber( number: Double )
120
121     //func  IQA
122
123     override func didReceiveMemoryWarning() {
124         super.didReceiveMemoryWarning()
125         // Dispose of any resources that can be recreated.
126     }
127
128
129 }
```

C.4 Arquivo *InfoViewController.swift*

```
1
2  //
3  //  InfoViewController.swift
4  //  infoAR
5  //
6  //  Created by Mayara Oliveira on 2016-11-04.
7  //  Copyright © 2016 Mayara. All rights reserved.
8  //
9
10 import UIKit
11 import MessageUI
12
13 class InfoViewController: UIViewController,
      MFMailComposeViewControllerDelegate {
14
15     override func viewDidLoad() {
16         super.viewDidLoad()
17
18         // Do any additional setup after loading the view.
19     }
20
21     @IBAction func faleconosco(_ sender: AnyObject) {
22         let mailComposeViewController =
              configuredMailComposeViewController()
23         if MFMailComposeViewController.canSendMail(){
24             self.present(mailComposeViewController,animated: true
                  , completion: nil)
25         }else {
26             self.showSendMailErrorAlert()
27         }
28     }
29
30     func configuredMailComposeViewController() ->
          MFMailComposeViewController{
31         let mailComposerVC = MFMailComposeViewController()
32         mailComposerVC.mailComposeDelegate = self
33
34         mailComposerVC.setToRecipients(["infoarapp@gmail.com"])
35         mailComposerVC.setSubject("Mensagem do aplicativo iOS")
36         mailComposerVC.setMessageBody("úDvidas e õSugestes para o
```

```
            aplicativo:", isHTML: false)
37
38          return mailComposerVC
39
40      }
41
42
43      func showSendMailErrorAlert(){
44          let sendMailErrorAlert = UIAlertView (title:"ãNo foi
               ípossvel enviar o email", message: "Seu dispositivo ãno
                ôpde enviar o email. Por favor cheque as çõconfiguraes
                e tente de novo", delegate: self, cancelButtonTitle: "
               OK")
45              sendMailErrorAlert.show()
46      }
47
48      func mailComposeController(_ controller:
           MFMailComposeViewController, didFinishWith result:
           MFMailComposeResult, error: Error?) {
49          controller.dismiss(animated: true, completion: nil)
50      }
51
52
53      override func didReceiveMemoryWarning() {
54          super.didReceiveMemoryWarning()
55          // Dispose of any resources that can be recreated.
56      }
57
58
59      /*
60      // MARK: - Navigation
61
62      // In a storyboard-based application, you will often want to
           do a little preparation before navigation
63      override func prepare(for segue: UIStoryboardSegue, sender:
           Any?) {
64          // Get the new view controller using segue.
               destinationViewController.
65          // Pass the selected object to the new view controller.
66      }
67      */
68
```

```
69 }
```

ANEXO D – CÓDIGOS APLICATIVO ANDROID

D.1 Arquivo *AndroidManifest.xml*

```
1
2  <?xml version="1.0" encoding="utf-8"?>
3  <manifest xmlns:android="http://schemas.android.com/apk/res/
       android"
4      package="com.example.mayaraoliveira.infoar">
5
6      <uses-permission android:name="android.permission.INTERNET"/>
           //permissao para usar a internet
7      <uses-permission android:name="android.permission.
           ACCESS_NETWORK_STATE"/> //permissao para verificar se a
           internet esra sendo usada
8      <uses-permission android:name="android.permission.
           WRITE_EXTERNAL_STORAGE"/> // permissao para escrever
9
10     <permission android:name="com.example.mayaraoliveira.
           permission.MAPS_RECEIVE"
11                 android:protectionLevel="signature" /> // cria
                       autorizacao para o aplicativo receber o mapa
12
13     <uses-permission android:name="com.example.mayaraoliveira.
           permission.MAPS_RECEIVE"/> //pede permissao
14     <uses-permission android:name="com.google.android.providers.
           gsf.permissions.READ_GSERVICES"/>
15
16     <uses-feature android:glEsVersion="0x00020000"
17                 android:required="true"/> // utilizar o
                       googlemaps na versao que ele quer
18
19     <application
20         android:allowBackup="true"
21         android:icon="@mipmap/ic_launcher"
22         android:label="@string/app_name"
23         android:supportsRtl="true"
24         android:theme="@style/AppTheme">
25         <activity android:name=".MainActivity">
```

```
26              <intent-filter>
27                  <action android:name="android.intent.action.MAIN"
                        />
28
29                  <category android:name="android.intent.category.
                        LAUNCHER" />
30              </intent-filter>
31          </activity>
32
33          <meta-data android:name="com.google.android.maps.v2.
                API_KEY"
34              android:value="
                    AIzaSyBaEcoOMEAgaPp9meFeSC1L4xHEHFalpqg" /> //
                    verificando com API KEY
35
36          <meta-data android:name="com.google.android.gms.version"
37              android:value="@integer/google_play_services_version"
                    />
38
39      </application>
40
41  </manifest>
```

D.2 Arquivo *activity_main.xml*

```
1
2   <?xml version="1.0" encoding="utf-8"?>
3   <RelativeLayout xmlns:android="http://schemas.android.com/apk/res
        /android"
4       xmlns:tools="http://schemas.android.com/tools"
5       android:layout_width="match_parent"
6       android:layout_height="match_parent"
7       android:paddingBottom="0dp"
8       android:paddingLeft="0dp"
9       android:paddingRight="0dp"
10      android:paddingTop="0dp"
11      tools:context="com.example.mayaraoliveira.infoar.MainActivity
            ">
12
13      <TextView
14          android:layout_width="wrap_content"
```

```
15          android:layout_height="wrap_content"
16          android:text="Hello World!"
17          android:id="@+id/textView" />
18
19      <fragment
20          android:layout_width="match_parent"
21          android:layout_height="match_parent"
22          android:name="com.google.android.gms.maps.MapFragment"
23          android:id="@+id/mapFragment"
24          android:layout_centerHorizontal="true"
25          android:layout_alignParentTop="true" />
26  </RelativeLayout>
```

D.3 Arquivo *MainActivity.Java*

```
1
2   package com.example.mayaraoliveira.infoar;
3
4   import android.support.v4.app.FragmentActivity;
5   import android.os.Bundle;
6
7   import com.google.android.gms.maps.CameraUpdateFactory;
8   import com.google.android.gms.maps.GoogleMap;
9   import com.google.android.gms.maps.OnMapReadyCallback;
10  import com.google.android.gms.maps.SupportMapFragment;
11  import com.google.android.gms.maps.model.LatLng;
12  import com.google.android.gms.maps.model.MarkerOptions;
13
14  public class infoAR extends FragmentActivity implements
        OnMapReadyCallback {
15
16      private GoogleMap mMap;
17
18      @Override
19      protected void onCreate(Bundle savedInstanceState) {
20          super.onCreate(savedInstanceState);
21          setContentView(R.layout.activity_info_ar);
22          // Obtain the SupportMapFragment and get notified when
              the map is ready to be used.
23          SupportMapFragment mapFragment = (SupportMapFragment)
              getSupportFragmentManager()
```

```
24                  .findFragmentById(R.id.map);
25          mapFragment.getMapAsync(this);
26      }
27
28
29      /**
30       * Manipulates the map once available.
31       * This callback is triggered when the map is ready to be
             used.
32       * This is where we can add markers or lines, add listeners
             or move the camera. In this case,
33       * we just add a marker near Sydney, Australia.
34       * If Google Play services is not installed on the device,
             the user will be prompted to install
35       * it inside the SupportMapFragment. This method will only be
              triggered once the user has
36       * installed Google Play services and returned to the app.
37       */
38      @Override
39      public void onMapReady(GoogleMap googleMap) {
40          mMap = googleMap;
41
42          // Add a marker in Sydney and move the camera
43          LatLng laranjeiras = new LatLng(-20.196, -40.245);
44          LatLng carapina = new LatLng(-20.226, -40.259);
45          LatLng jcamburi = new LatLng(-20.255, -40.270);
46          LatLng enseada = new LatLng(-20.313, -40.291);
47          LatLng centrovix = new LatLng(-20.321, -40.333);
48          LatLng ibes = new LatLng(-20.348, -40.317);
49          LatLng vvcentro = new LatLng(-20.336, -40.291);
50          LatLng cariacica = new LatLng(-20.342, -40.402);
51          LatLng continental = new LatLng(-20.225, -40.218);
52          mMap.addMarker(new MarkerOptions().position(laranjeiras).
                title("Laranjeiras"));
53          mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(
                laranjeiras,12));
54      }
55  }
```

D.4 Arquivo *InfoActivity.xml*

```
1
2  package com.example.tutorialspoint;
3
4  import android.net.Uri;
5  import android.os.Bundle;
6  import android.app.Activity;
7  import android.content.Intent;
8  import android.util.Log;
9  import android.view.Menu;
10 import android.view.View;
11 import android.widget.Button;
12 import android.widget.Toast;
13
14
15
16 public class display extends Activity {
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_main);
21
22         Button startBtn = (Button) findViewById(R.id.sendEmail);
23         startBtn.setOnClickListener(new View.OnClickListener() {
24             public void onClick(View view) {
25                 sendEmail();
26             }
27         });
28     }
29
30     protected void sendEmail() {
31         Log.i("Enviar email", "");
32         String[] TO = {"infoarapp@gmail.com"};
33         String[] CC = {""};
34         Intent emailIntent = new Intent(Intent.ACTION_SEND);
35
36         emailIntent.setData(Uri.parse("mailto:"));
37         emailIntent.setType("text/plain");
38         emailIntent.putExtra(Intent.EXTRA_EMAIL, TO);
39         emailIntent.putExtra(Intent.EXTRA_CC, CC);
40         emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Mensagem do
               aplicativo Android");
41         emailIntent.putExtra(Intent.EXTRA_TEXT, "úDvidas e
```

```
            õSugestes para o aplicativo:");
42
43          try {
44              startActivity(Intent.createChooser(emailIntent, "
                    Enviando"));
45              finish();
46              Log.i("Email enviado", "");
47          } catch (android.content.ActivityNotFoundException ex) {
48              Toast.makeText(MainActivity.this, "Aplicativo ãno
                    instalado", Toast.LENGTH_SHORT).show();
49          }
50      }
51  }
```