

JHONSTON ROBSON DALCIN
MARCELO PESSE
RICARDO BONFÁ ALBERICO

DEMÉTER: SISTEMA DE IRRIGAÇÃO INTELIGENTE

SÃO PAULO
2012

JHONSTON ROBSON DALCIN
MARCELO PESSE
RICARDO BONFÁ ALBERICO

DEMÉTER: SISTEMA DE IRRIGAÇÃO INTELIGENTE

Projeto apresentado à Escola Politécnica da USP como parte dos requisitos das disciplinas PSI2591-Projeto de Formatura I e PSI2594-Projeto de Formatura II, para obtenção do título de bacharel em engenharia.

Área de Concentração:
Sistemas Eletrônicos

Orientador:
Prof. Dr. José Kleber Cunha Pinto

SÃO PAULO

2012

DEDICATÓRIA

Dedicamos este trabalho a nossas famílias, amigos e mestres, que tanto nos ensinaram e ajudaram ao longo de nossa jornada.

RESUMO

Motivado pelo uso excessivo, e por vezes desnecessário, de água no planeta, aliado às previsões de escassez desse recurso no futuro, o projeto propõe uma solução de baixo custo para o monitoramento de informações, visando aumentar a eficiência desse recurso em plantações de pequeno e médio porte. Será desenvolvido um sistema que faça medições em tempo real da umidade do solo e, por meio de uma rede de transmissão de dados por ZigBee, envie relatórios de umidade para uma central que possibilite a irrigação apenas dos setores onde for constatada a necessidade de mais água, através de qualquer sistema de irrigadores utilizados, configurados em um módulo especial do sistema. Todo esse registro, bem como outras informações relevantes para o cultivo, serão reunidas em um Sistema de Informação, desenvolvido com uma interface amigável, auxiliando assim, o usuário a administrar seu agronegócio de forma prática e remota.

Palavras chave: Água. Agricultura. Automação. Sensores de umidade. Rede ZigBee. Sistema de informação

ABSTRACT

Motivated by overuse, and sometimes unnecessary use of water on the planet, coupled with forecasts of a shortage of this resource in the future, this project proposes a low cost solution for information monitoring, to enhance the efficiency of this resource in small and medium size plantations. A system that makes real-time measurements of soil moisture will be developed and, through a network of data transmission by ZigBee, send reports of moisture to a central computer, acting in order to irrigate only the sectors where more water is necessary. This whole record, and other relevant information to cultivation will be assembled in an Information System, developed with a user-friendly interface, thus helping the user to manage their practice and agribusiness remotly.

Keywords: Water. Agriculture. Automation. Humidity sensors. ZigBee network. Information system.

LISTA DE ILUSTRAÇÕES

Figura 1 - Diagrama de elementos do sistema.....	12
Figura 2 - LeoStick.....	13
Figura 3 - Esquemáticos do ADC.....	14
Figura 4 - Principais Módulos do Mercado	17
Figura 5 - Topologia Hexagonal.....	18
Figura 6 - Diagrama de Classes	20
Figura 7 - Plano de Implementação	21
Figura 8 – Metodologia Scrum	22
Figura 9 – Lista de Requisitos e Priorização	23
Figura 10 - Placas desenvolvidas no primeiro protótipo	25
Figura 11 - Primeiro Protótipo	26
Figura 12 – Funcionamento MVC	27
Figura 13 – Circuito final	32
Figura 14 – Página de login do Sistema.....	33
Figura 15 – Página Inicial do Sistema	34
Figura 16 – Página da Fazenda do Sistema	34
Figura 17 – Página do Setor do Sistema.....	35
Figura 18 – Código da View do Setor	36
Figura 19 – Código do Controller do Setor.....	36
Figura 20 – Código do Models do Setor.....	37
Figura 21 – Ambiente de testes	38
Figura 22 – Visualização dos setores	39
Figura 23 – Tela do Site de Acompanhamento	41

LISTA DE TABELAS

Tabela 1 - Distribuição da Água no Planeta	7
Tabela 2 - Matriz de comparação dos critérios.....	11
Tabela 3 - Matriz de decisão.....	11
Tabela 4 - Tipos de Sensores	15
Tabela 5 - Matriz de decisão para topologia	16
Tabela 6 - Matriz de comparação dos módulos.....	16

LISTA DE ABREVIATURAS, DEFINIÇÕES E SÍMBOLOS

ADC	Analogic/Digital Converter ou Conversor Analógico/Digital
API	<i>Application Programming Interface</i> (ou Interface de Programação de Aplicativos) é um conjunto de rotinas e padrões estabelecidos por um software para a utilização das suas funcionalidades por aplicativos que não pretendem envolver-se em detalhes da implementação do software, mas apenas usar seus serviços.
Boot-loader	Carregador inicial do sistema
BV	<i>Business Value</i> ou valor da funcionalidade para o cliente
CSS	CascadingStyleSheets. É uma linguagem de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação
Django	Framework web
Framework	Modelo de dados para simular um domínio
HTML	HyperTextMarkupLanguage, que significa Linguagem de Marcação de Hipertexto
Javascript	Linguagem de programação
Metodologia Scrum	Metodologia de desenvolvimento ágil
Mockup	Conjunto de dados utilizados para testes em softwares
Mosfets	MetalOxideSemiconductorFieldEffectTransistor, ou transistor de efeito de campo
MVV	ModelViewController
MySQL	Banco de dados
Planning Poker	Método de atribuição de notas por múltiplas pessoas ao mesmo tempo
PMBok	Guia de boas práticas em gestão de projetos, desenvolvido pelo PMI
PMI	Project Management Institute
Python	Linguagem de programação versátil
Rede mesh	Rede de dados sem fio em que cada nó é também um transmissor
ROI	<i>ReturnonInvestiment</i> ou <i>Retorno do Investimento</i>
Sensor TDR	Time Domain Reflectometers
SI	Sistema de Informação
SP	<i>Storie Points</i> ou dificuldade do time para implementar uma funcionalidade
Sprints	<i>Ciclos de implementação</i>
USB	Universal Serial Bus. Tipo de conexão para computadores.
ZigBee	Padrão de rede sem fio de baixa potência

SUMÁRIO

1-Introdução.....	7
2 - Objetivos	9
2.1 – Objetivos Gerais	9
2.2 – Objetivos Específicos.....	9
3 – Referencial Teórico.....	10
4 - Metodologia	10
4.1 - Concepção	10
4.2 - Design	12
4.2.1 - Microcontrolador	12
4.2.2 – Conversor Analógico Digital (ADC)	13
4.2.3 - Sensor de Umidade	14
4.2.4 - Controlador de Bateria.....	15
4.2.5 - Transceptores.....	15
4.2.6 - Sistema de Informação	19
4.3 - Construção do protótipo.....	24
5 - Atividades realizadas	27
5.1 - Plano de Integração e Depuração	27
5.2 - Plano de Testes.....	28
5.2.1 - Dispositivo (unidade)	29
5.2.2 –Integração.....	30
5.2.3 - Aceitação.....	30
6– Resultados.....	32
7– Informações Gerenciais	39
8 - Conclusão	42
9 - Referências	43

1-Introdução

O aumento exacerbado da utilização de recursos naturais em nosso planeta é hoje um dos principais problemas que temos que resolver com relação ao meio ambiente, pois afeta não só o estado atual e degradado em que a natureza se encontra, como também a perspectiva de recursos disponíveis para as gerações seguintes.

Os minérios e combustíveis fósseis foram, há algum tempo atrás, a grande preocupação da humanidade em termos de recursos que se esgotariam, em um relativo curto prazo, em nome do progresso e desenvolvimento. Graças ao avanço tecnológico e aos constantes investimentos para fontes mais eficientes, menos nocivas e mais abundante de energia, temos hoje diversas alternativas para suprir nossa necessidade energética e provavelmente vivemos no fim de um ciclo em que a principal riqueza de uma nação era sua reserva de petróleo.

Voltamos agora nossa atenção, portanto, a um recurso muito mais vital e aparentemente mais abundante em nosso planeta: a água. Apesar 75% da superfície da Terra ser recoberta por massas líquidas, a água doce não representa mais do que 3% desse total (conforme tabela abaixo), e dessa, apenas um terço é acessível.

Tabela 1 - Distribuição da Água no Planeta (Fonte: Wetzel, 1983)

<u>Local</u>	<u>Volume (km³)</u>	<u>Percentual do total (%)</u>
Oceanos	1.370.000	97,61
Calotas polares e geleiras	29.000	2,08
Água Subterrânea	4.000	0,29
Água doce de lagos	125	0,009
Água salgada de lagos	104	0,008
Água misturada no solo	67	0,005
Rios	1,2	0,00009
Vapor d'água na atmosfera	14	0,0009

A quantidade de água doce produzida pelo seu ciclo natural é hoje basicamente a mesma que em 1950 e que deverá permanecer inalterada até 2050.

É por isso que deve-se preservar bem este recurso e fazer seu uso de forma consciente para que as próximas gerações possam também seu uso, garantindo sua sobrevivência e seu desenvolvimento, pois segundo a ONU (2007) até 2025, se os atuais padrões de consumo se mantiverem, duas em cada três pessoas no mundo vão sofrer escassez moderada ou grave de água.

A aplicação que mais faz o uso desse recurso é o cultivo de alimentos na agricultura, com cerca de 65% a 70% do total, sendo que a agricultura irrigada representa 40% da agricultura total mundial (UPF, 2007). Para ter uma dimensão da quantidade utilizada, tomamos como exemplo a cana-de-açúcar, que para a produção de uma tonelada consome-se 600 toneladas de água. Já para a soja são necessários 2000 litros de água, para produção de um quilograma de alimento (Conab, 2007).

Este documento visa apresentar de forma elaborada o planejamento e as atividades desenvolvidas nas disciplinas PSI2591 - Projeto de Formatura 1 e PSI2594 - Projeto de Formatura 2, bem como a demonstração aplicação de como os mecanismos e metodologias ensinados durante a disciplina foram utilizados no projeto e como este se relaciona com as demais disciplinas oferecidas no curso de Engenharia de Sistemas Eletrônicos.

2 - Objetivos

2.1 – Objetivos Gerais

O projeto conceitual foi desenvolvido a partir do problema da falta de água no planeta. Como a agricultura é o principal destino de toda a água doce utilizada no mundo e, em grande parte não realiza um racionamento adequado, o projeto Deméter pretende desenvolver uma solução de baixo custo para os problemas de irrigação em pequenas e médias propriedades rurais de produção agrícola, amenizando assim impactos ambientais provenientes do uso excessivo e até mesmo desnecessário de água em certos casos. A solução poderá ser estendida de acordo com a área plantada e fornecerá ao agricultor responsável pela plantação, diversas outras informações relevantes para o monitoramento e controle da lavoura, acarretando assim em economia de recursos naturais utilizados, redução de custos, aumento da produtividade e melhoria na qualidade do produto.

2.2 – Objetivos Específicos

O produto final deverá atuar de modo a coletar, transmitir e interpretar os dados necessários para uma irrigação específica para cada setor da plantação, de acordo com a sua respectiva situação de umidade atual.

Como parâmetros iniciais para a busca da solução podemos citar:

- a solução deve ser independente da rede elétrica presente na região aplicação;
- o mal funcionamento de um módulo de aquisição de dados não deve acarretar no mal funcionamento de outro módulo e sua transmissão de dados;
- o responsável pela plantação deve ser capaz de acompanhar remotamente os dados referentes às suas terras;
- as informações referentes a períodos anteriores poderão ser armazenadas.

3 – Referencial Teórico

Através de conversas com agrônomos foi possível verificar que atualmente no Brasil são utilizados quatro principais sistemas de irrigação, o método de pivô central, gotejamento, aspersão e inundação.

- o método de pivô central é mais utilizado em grandes fazendas e é mais indicado para as culturas de grãos;
- o método de inundação é utilizado em culturas específicas, como o arroz inundado;
- o método de gotejamento é utilizado em culturas com alta rentabilidade como fruticultura e horticultura;
- o método de aspersão é utilizado em fazendas de médio porte em diferentes culturas.

4 - Metodologia

4.1 - Concepção

Através de um brainstorming entre os integrantes da equipe, foram levantadas possíveis soluções para amenizar o consumo de água em plantações de pequeno e médio porte. Foram elas:

1. Sensoreamento da umidade do solo

Utilização de sensores de umidade e rede de comunicação wireless para irrigação específica do local onde constatada a carência de água.

2. Temporizadores

Sistema de irrigação temporizado, com horários pré-programados específicos.

3. Aplicação de culturas hidropônicas

Implementação de métodos alternativos de plantio.

4. Implementação alteração genética

Utilização de alimentos transgênicos que consumam menos água

5. Máquina de Osmose Reversa

Dispositivo de transforma água salgada em água própria para o consumo

Crítérios de Seleção

I. Investimento Inicial

- II. Custo de Manutenção
- III. Facilidade de operação
- IV. Escalabilidade
- V. Capacidade de Diversificação de Cultura
- VI. Efetividade
- VII. Complexidade de projeto

Tabela 2 - Matriz de comparação dos critérios

	Jhonston	Marcelo	Ricardo	Média
Investimento Inicial	3	4	2	3
Custo de Manutenção	1	2	2	2
Facilidade de operação	3	5	3	4
Escalabilidade	3	3	3	3
Capacidade de Diversificação de Cultura	4	2	2	3
Efetividade	4	5	5	5
Inovação	5	4	5	5
Complexidade de projeto	3	3	4	3

Definição da melhor solução:

Em decisão conjunta através do método do "planningpoker", a equipe chegou às seguintes notas para cada solução proposta:

Tabela 3 - Matriz de decisão

<u>Crítérios/Projetos</u>	1	2	3	4	5
Investimento Inicial	4	5	2	1	2
Custo de Manutenção	3	4	2	1	1
Facilidade de operação	3	5	2	4	1
Escalabilidade	4	5	3	5	1
Capacidade de Diversificação de Cultura	4	4	2	1	5
Efetividade	5	2	5	3	4
Inovação	5	2	3	5	5
Complexidade de projeto	3	5	4	1	1
<u>Nota Ponderada Final</u>	107.3	101.0	81.7	77.7	73.0

4.2 - Design

A solução proposta consiste em um sistema de irrigação inteligente controlado por um Sistema de Informação (SI), que permite a monitoração e o controle dos elementos envolvidos na rede além de outras informações relevantes para o acompanhamento remoto da lavoura.

Inicialmente a rede de dados principal do sistema de irrigação será composta por três elementos chave: Módulos microcontrolados alimentados por energia solar, acoplados a sensores de umidade, responsáveis pela aquisição dos dados no solo; uma rede mesh utilizando módulos ZigBee, responsável pela transmissão desses dados para um servidor e uma rede de Sprinklers, de controle à distância, responsáveis por atuar de forma automática de acordo com os dados recebidos.

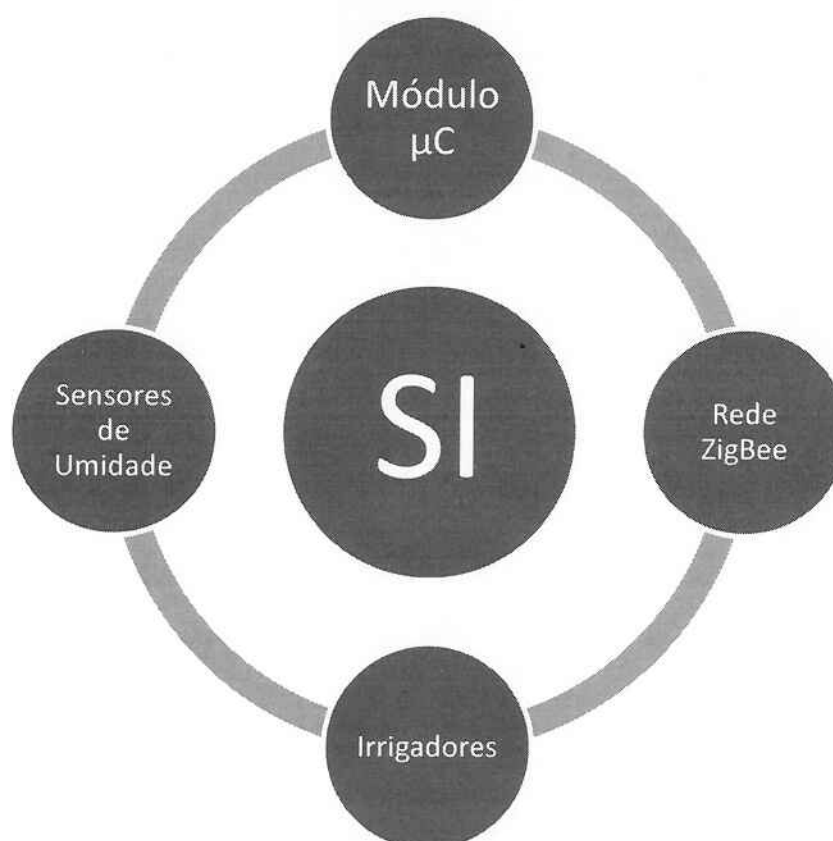


Figura 1 - Diagrama de elementos do sistema

4.2.1 - Microcontrolador

Utilizou-se a plataforma open-source Arduino Leonardo como nosso controlador, a plataforma Arduino é um misto de microcontrolador, boot-loader e

linguagem de programação, possuindo a vantagem de possuir diversas bibliotecas disponíveis e pela sua facilidade de aprendizado.

A escolha foi feita também com base na familiaridade do grupo com a plataforma e também a disponibilidade de kit de desenvolvimento.

Para o projeto preliminar será utilizado uma placa Arduino vendida pelo freeelectronics com o nome de LeoStick, conforme apresentado na Figura 2.

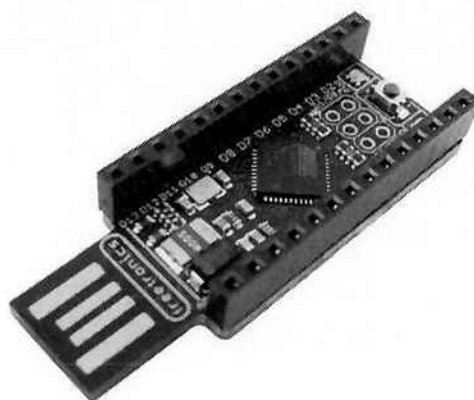


Figura 2 - LeoStick

A placa Arduino Leonardo utiliza um microcontrolador ATmega32u4 fabricado pela Atmel e possui controlador USB integrado.

4.2.2–Conversor Analógico Digital (ADC)

Como o sensor de umidade utilizado recomenda um ADC de no mínimo 12bits e o ADC interno do microcontrolador só funciona com 10bits, decidiu-se pela utilização de um ADC externo delta-sigma fabricado pela Microchip, capaz de medir uma tensão entre mais e menos 2.048V, conforme esquemáticos a seguir.

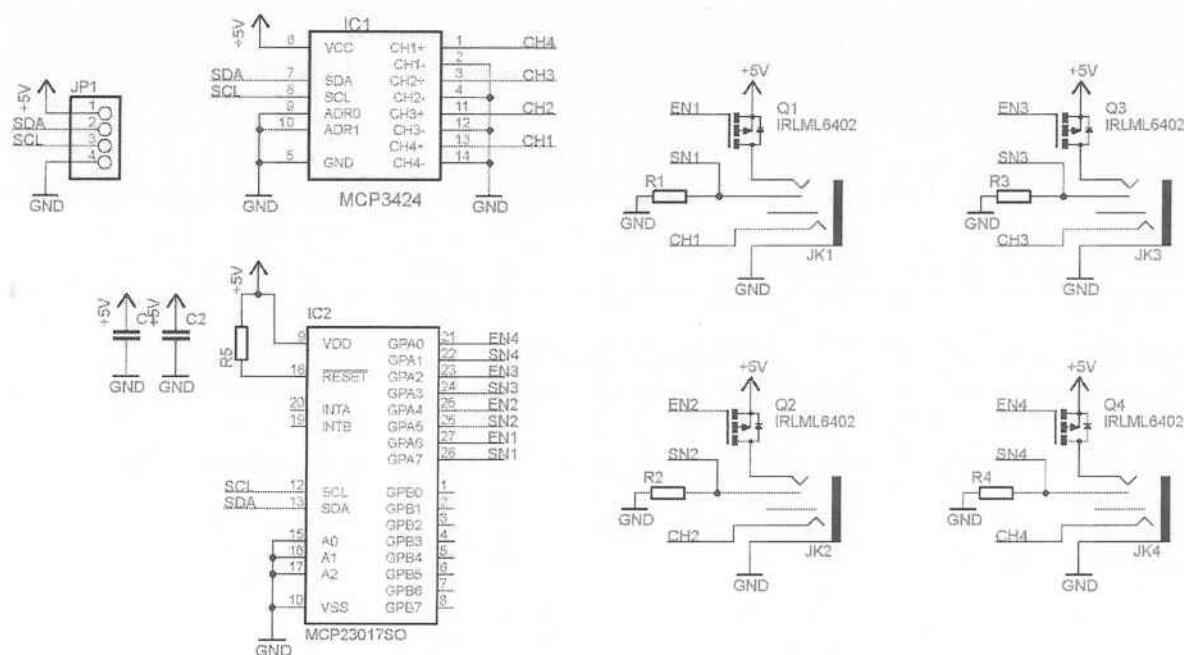


Figura 3 - Esquemáticos do ADC

Como o fabricante do sensor também recomenda que o sensor seja ligado somente no momento da leitura para evitar interferência em outros equipamentos, utilizou-se um conjunto de mosfets para controlar a alimentação dos sensores.

Como seriam necessários vários mosfets, decidimos utilizar um expensor de porta da Microchip para controlar todos os mosfets além de detectar quando algum dos sensores esta presente.

4.2.3 - Sensor de Umidade

Foram pesquisados diversos tipos e qualidades de sensores de umidade. Seus princípios de funcionamento diferem, sendo cada qual recomendado para uma aplicação específica.

Foram pesquisados os diversos tipos de sensores de umidade do solo e foi criada a seguinte tabela com pontos fortes e fracos de cada sensor.

Tabela 4 - Tipos de sensores

	Pontos Fortes	Pontos Fracos
Resistivo	- Baixíssimo custo - Ampla faixa disponível	- Calibração periódica - Altamente influenciável pela salinidade do solo
Capacitivo	- Resposta rápida - Baixo custo	- Influenciado pela temperatura e tipo do solo
TDR	- Comumente utilizado na indústria	- Alto custo - Alta complexidade
Tensiometro	- Baixo Custo - Praticidade	- Não recomendado para terrenos arenosos ou muito secos - Requer manutenção frequente
Sonda de Nêutron	- Altamente confiável - Mede uma grande quantidade de solo	- Método radioativo - Alto custo - Requer calibração

Dado principalmente ao requisito de criar um produto de baixo custo, os sensores que melhor preenchem os requisitos de engenharia são os sensores resistivos e os capacitivos.

4.2.4 - Controlador de Bateria

Supondo que o modulo só precise funcionar durante um intervalo pequeno de tempo e com intervalos grandes de tempo entre uma ativação e outra, o consumo de energia deve ser relativamente baixo.

A equipe optou então por manter o modulo energizado utilizando um sistema de bateria e células solares.

Por ter se apresentado muito custoso, após um levantamento preliminar, fora dos limites orçamentários do projeto e também por não se tratar do enfoque do produto, esse módulo foi considerado como secundário para a implementação do projeto e ficará apenas como indicação de uso, complementar ao produto desenvolvido.

4.2.5 - Transceptores

Primeiramente foi discutido que tipo de topologia de rede seria utilizado, foram listados as principais topologias e foi feita uma matriz de decisão para decidir a mais adequada.

Tabela 5 - Matriz de decisão para topologia

	Facilidade de Implementação	Capacidade de Recuperação	Adição de novos módulos	Alcance	Média Final
Mesh	1	4	5	5	3,75
Tree	3	2	5	4	3,5
Star	5	3	5	1	3,5
Ring	3	1	1	3	2

Embora, como pode ser visto na tabela anterior, a rede Mesh tenha uma grande dificuldade de implementação, alguns transceptores já a possuem implementada ou oferecem uma implementação para alguns tipos de microcontroladores.

Resumidamente, uma rede mesh é uma rede onde cada nó funciona também como um retransmissor de dados dos outros nós.

Uma rede mesh é capaz de realizar *self-healing*, ou seja, caso algum módulo perca a conexão ou pare de funcionar, a rede deve ser capaz de se reorganizar de modo a lidar com a queda do módulo.

Para a escolha do módulo transceptor a ser utilizado, criamos a tabela a seguir com a comparação de alguns módulos proeminentes no mercado, representados na Figura 4.

Tabela 6 - Matriz de comparação dos módulos

Nome	Banda	Potencia de Transmissão (dBm)	Corrente TX@3.3V (mA)	Corrente RX@3.3V (mA)	Alcance linha direta (m)	Implementação Mesh
NRF24L01+ PA + LNA	2,4GHz	20	115	45	1100	Não
NRF24L01+	2,4GHz	0	11,3	13,5	100	Não
Xbee-PRO	2,4GHz	18	250	55	1600	Sim
Xbee	2,4GHz	0	45	50	90	Sim
RFM12B	433MHz	0	15	11	200	Não
RFM22B	433MHz	20	85	18,5	-	Não

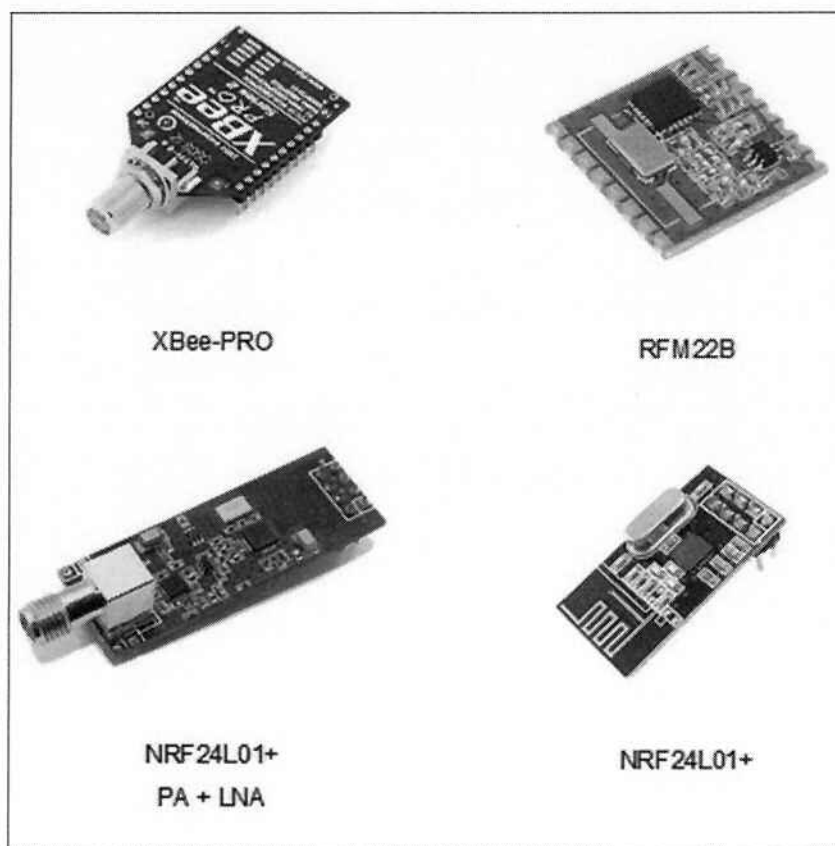


Figura 4 - Principais Módulos do Mercado

Diante do alcance limitado dos transceptores, realizamos alguns cálculos referentes a área efetivamente coberta por uma rede sem fio utilizando topologia mesh.

Considerando um sistema de células hexagonais, podemos considerar que o alcance máximo do transceptor equivale a d , onde d é a distância entre o centro da célula hexagonal a outra célula, conforme a figura a seguir.

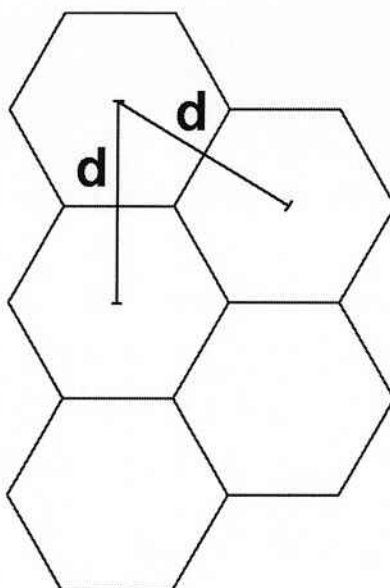


Figura 5 - Topologia Hexagonal

A área do hexágono é dada por $A = \frac{\sqrt{3}}{2} d^2$

Considerando $d = 100\text{m}$, temos $A = 8660 \text{ m}^2$

Já para $d = 1000\text{m}$, temos $A = 866.025 \text{ m}^2$

Para uma plantação de 10 hectares, seriam necessários cerca de 11.547 transceptores com alcance de 100m e 115 transceptores com alcance de 1000m.

Portanto, verifica-se a necessidade um transceptor de longo alcance e de fácil implementação.

Porém, um transceptor com alcance maior implica em um preço maior. Dado o limitado recurso financeiro, e a compatibilidade entre alguns transceptores de baixo e longo alcance, iremos utilizar para o protótipo um transceptor de baixo alcance que possa ser substituído por um de longo alcance facilmente.

Pela facilidade de implementação, decidimos utilizar o modulo XBee que já possui uma rede mesh previamente implementada, além de contar com algumas facilidades, como código corretor de erro e modo de baixo consumo.

Utilizaremos o modulo XBee comum com potencia de 0 dBm na saída da antena e alcance estimado de 100m para testes e para a construção do protótipo.

O módulo XBeePRO com potencia de +20 dBm e alcance estimado de 1.000m fica como sugestão de implementação final devido às suas já mencionadas diferenciações.

Para cálculos de desempenho de bateria e painéis solares, utilizaremos o modulo XBeePRO como referencia, já que o mesmo consome muito mais potencia.

4.2.6 - Sistema de Informação

Para funcionamento completo do sistema de monitoramento de umidade dos solos será implementado um sistema de informação (SI) que obtenha os dados enviados pelos sensores, trate-os e os apresente de uma maneira clara e informativa para o usuário. Para maior entendimento do software, segue o diagrama de classes gerado até o momento, figura 6, e o plano de implementação, figura 7.

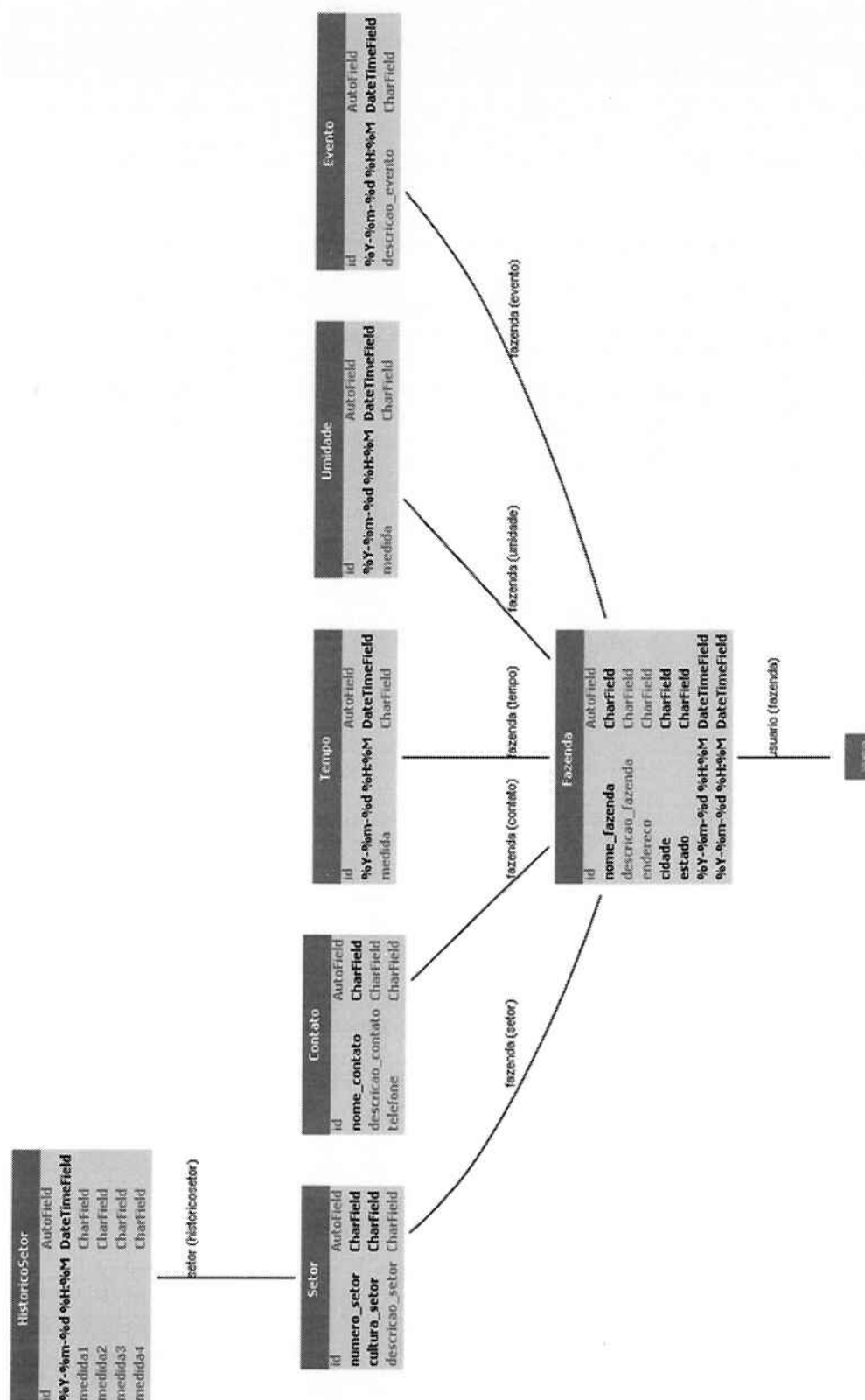


Figura 6 - Diagrama de Classes

Plano de Implementação - Projeto Demeter

Logado	Tela	URL	View	GET/POST	Parâmetros	Model	Comentários
nao	login	/login/	login	GET/POST	-	Usuario	Figura ao fundo Figura ao fundo Fazendas em lista
sim	inicio	/inicio/	inicio	GET/POST	-	Fazenda	
sim	cria_fazenda	/add_fazenda/	cria_fazenda	GET/POST	-	Fazenda	
sim	fazenda	/rem_fazenda/<id>/	rem_fazenda	GET	id_fazenda	Fazenda	Mapa com setores Descrição dos setores abaixo
sim	fazenda	/fazenda/<id>/	fazenda	GET/POST	id_fazenda	Fazenda; Setor; Contato; Tempo; Umidade; ar; Calendario	Valores importantes visíveis
sim	cria_setor	/fazenda/<id>/add_setor/	cria_setor	GET/POST	id_fazenda	Fazenda; Setor	Poder editar no mapa com Google Maps API
sim	setor	/fazenda/<id>/rem_setor/<id>/	rem_setor	GET	id_fazenda & id_setor	Fazenda; Setor	
sim	setor	/fazenda/<id>/setor/<id>/	setor	GET/POST	id_fazenda & id_setor	Fazenda; Setor	
sim	historico_setor	/fazenda/<id>/setor/<id>/historico/	historico_setor	GET	id_fazenda & id_setor	Fazenda; Setor; HistoricoSetor	
sim	cria_contato	/fazenda/<id>/add_contato/	cria_contato	GET/POST	id_fazenda & id_setor	Fazenda; Contato	
sim	contatos	/fazenda/<id>/rem_contato/<id>/	rem_contato	GET	id_fazenda & id_contato	Fazenda; Contato	
sim	contatos	/fazenda/<id>/contatos/	contatos	GET	id_fazenda	Fazenda; Contato	
sim	contato	/fazenda/<id>/contato/<id>/	contato	GET/POST	id_fazenda & id_contato	Contato	Lista de contatos
sim	tempo	/fazenda/<id>/tempo/	tempo	GET	id_fazenda	Fazenda; Tempo	Widget de ClimaTempo
sim	umidade_ar	/fazenda/<id>/umidade_ar/	umidade_ar	GET	id_fazenda	Fazenda; UmidadeAr	Widget de ClimaTempo
sim	calendario	/fazenda/<id>/calendario/	calendario	GET	id_fazenda	Fazenda	
sim	cria_evento	/fazenda/<id>/calendario/<id>/cria_evento/	add_evento	GET/POST	id_fazenda	Fazenda; Evento	
sim	evento	/fazenda/<id>/calendario/<id>/evento/<id>/	evento	GET/POST	id_fazenda & id_evento	Fazenda; Evento	
sim	evento	/fazenda/<id>/calendario/<id>/rem_evento/<id>/	rem_evento	GET	id_fazenda & id_evento	Fazenda; Evento	
sim	-	/fazenda/<id>/calendario/<id>/lembra_evento/lembra_evento	lembra_evento	GET	id_fazenda & id_evento	Fazenda; Evento	Envio SMTP

Figura 7 - Plano de Implementação

Para tal, o SI deve ser bem especificado e organizado seguindo uma metodologia que garanta os itens acima. Assim, foi escolhida a metodologia Scrum, baseada em métodos ágeis. Ela foi escolhida pois permite que o processo de desenvolvimento do software seja flexível e iterativo, diferentemente das metodologias tradicionais que acabam burocratizando e engessando correções ou pedidos do cliente.

Ela consiste em basicamente fazer ciclos, ou *sprints*, de especificação seguidos de implementação. Cada *sprint* de uma a duas semanas e gera um subproduto que será entregue ao cliente e estará pronto para uso. Assim, o cliente já pode verificar se os requisitos foram cumpridos e, caso não, já solicitar uma correção ou alteração, evitando que todas elas se acumulem no final do projeto e o prazo final seja prejudicado. Podemos ter uma noção melhor de como funciona essa metodologia através da figura 8.

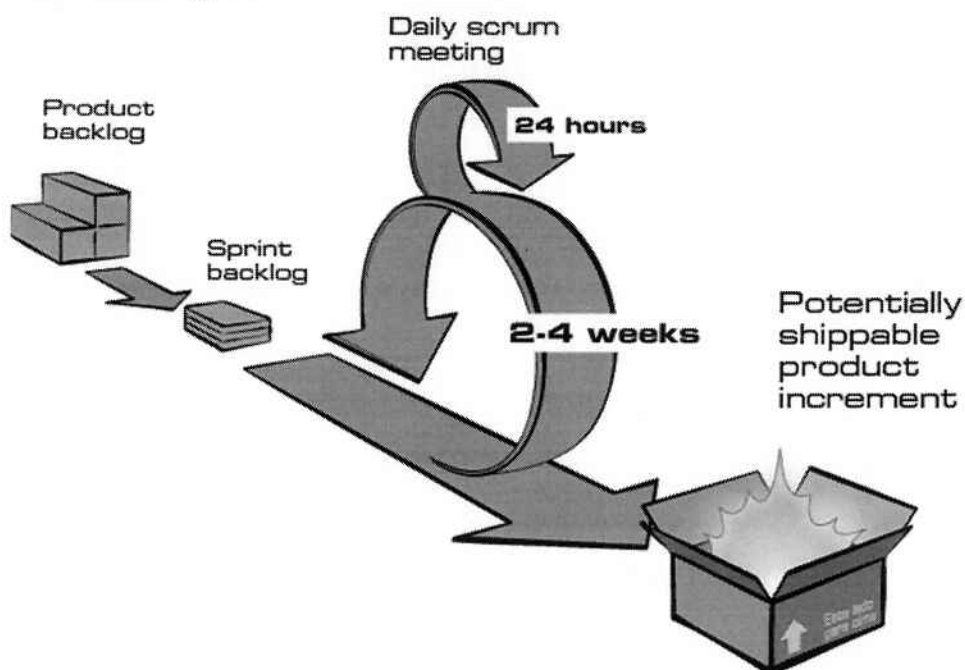


Figura 8 – Metodologia Scrum

As funcionalidades entregues em cada *sprint* são definidas pelo ROI (*Return on Investment*) atrelado a cada uma delas. O ROI é definido pela divisão do BV (*Business Value* = valor da funcionalidade para o cliente) pelo SP (*Storie Points* = dificuldade do time para implementar essa mesma funcionalidade). Abaixo, na figura

9, a lista das funcionalidades definidas até o momento e seus respectivos valores de ROI, BV, SP e também de horas estimadas para sua implementação.

Funcionalidades - Projeto Demeter	(1,2,3,5,8,13)	(1,2,3,5,8,13)	(BV/SP)	
Requisitos Funcionais	Business Value	Storie Points	ROI	Horas estimadas
Usuário pode logar no sistema	13	1	13	2
Usuário pode criar, ver, editar informações e remover fazendas	13	8	1,625	16
Usuário pode criar, ver, editar informações e remover setores de fazendas	8	13	0,61538	26
Usuário pode verificar a umidade do solo dos setores de suas fazendas	13	8	1,625	16
Usuário pode ver estatísticas sobre os dados de umidade do solo por setor	3	8	0,375	16
Usuário pode ver gráficos com os dados de umidade do solo por setor	5	5	1	10
Usuário pode verificar a temperatura atual, histórico e previsão do tempo para a região de sua fazenda	3	5	0,6	10
Usuário pode verificar a umidade do ar atual, histórico e a previsão dos próximos dias para a região de	3	5	0,6	10
Usuário pode criar, ver, editar informações e remover contatos de suas fazendas	5	3	1,66667	6
Usuário pode ter um calendário para cada fazenda	5	2	2,5	4
Usuário pode criar, ver, editar e remover eventos de seus calendários	5	5	1	10
Usuário pode ser avisado por email quando um evento de seus calendários se aproxima	3	3	1	6
Requisitos Não-funcionais				
Recebimento de dados no formato enviado pelos sensores de umidade	13	8	1,625	8
Banco de dados sincronizado com a web	5	8	0,625	8
Acessível por dispositivos móveis	3	8	0,375	8
Fácil visualização dos mapas	8	5	1,6	5
Fácil edição de setores através dos mapas	5	13	0,38462	13
Interface amigável	8	5	1,6	5
Interface moderna	5	5	1	5

Figura 9 – Lista de Requisitos e Priorização

4.3 - Construção do protótipo

Realizou-se a construção do primeiro protótipo utilizando a fabricação caseira de placas de circuito impresso, com o objetivo de testar alguns aspectos do circuito não compreendidos totalmente e também para realizar testes iniciais em alguns aspectos básicos do circuito, como consumo de energia, distancia máxima de transmissão, etc.

Optou-se pela construção modular do circuito, de forma que cada bloco essencial do circuito foi separado em sua própria placa. Essa abordagem oferece algumas vantagens muito interessantes nesse estado inicial do circuito, como a possível reutilização dos módulos em outros circuitos futuros, facilidade de correção de alguns erros de projeto, etc.

Depois de todos os módulos terem sido testados individualmente, trabalhou-se na integração dos módulos no software e na criação de um protótipo mais próximo do produto final.

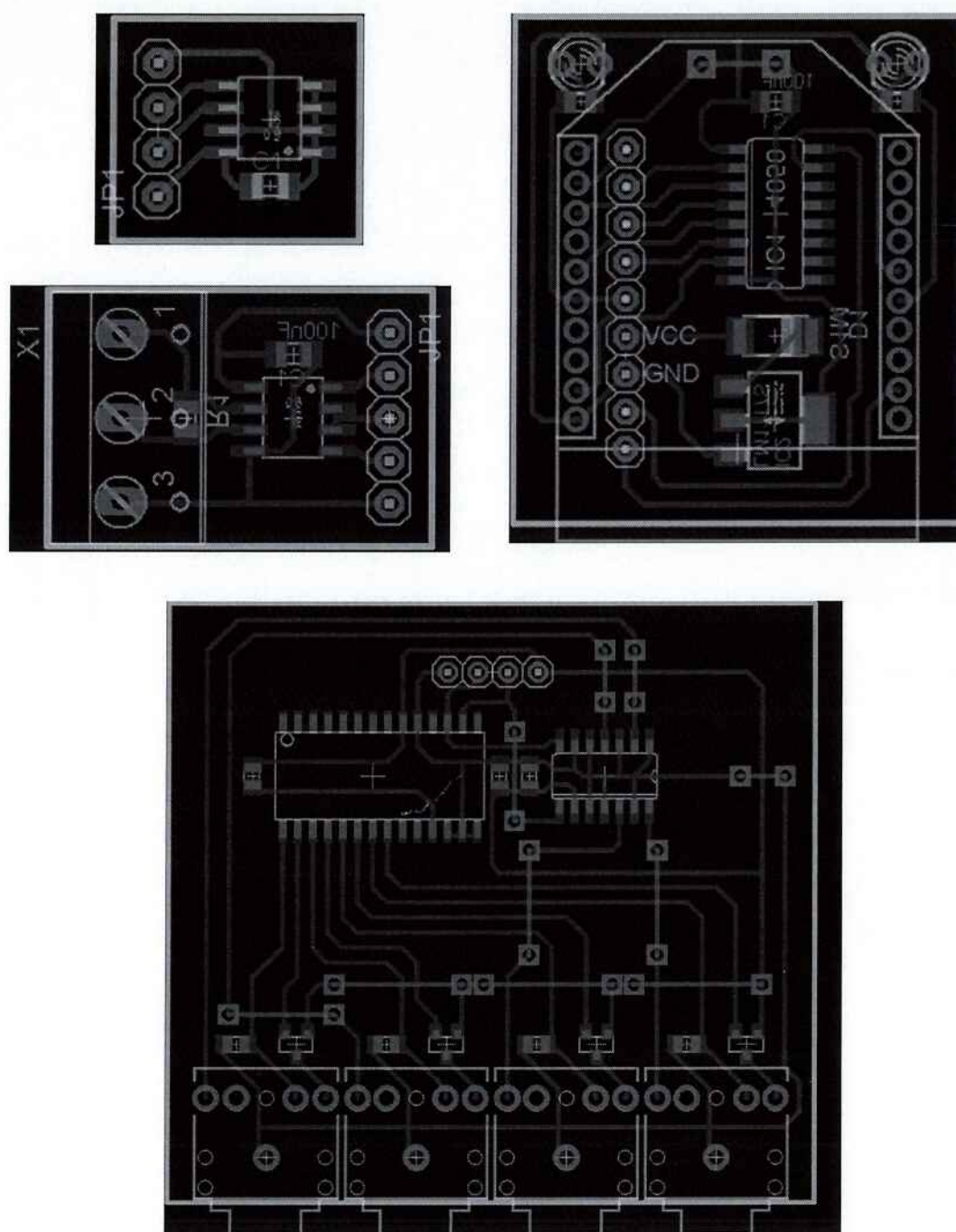


Figura 10 - Placas desenvolvidas no primeiro protótipo

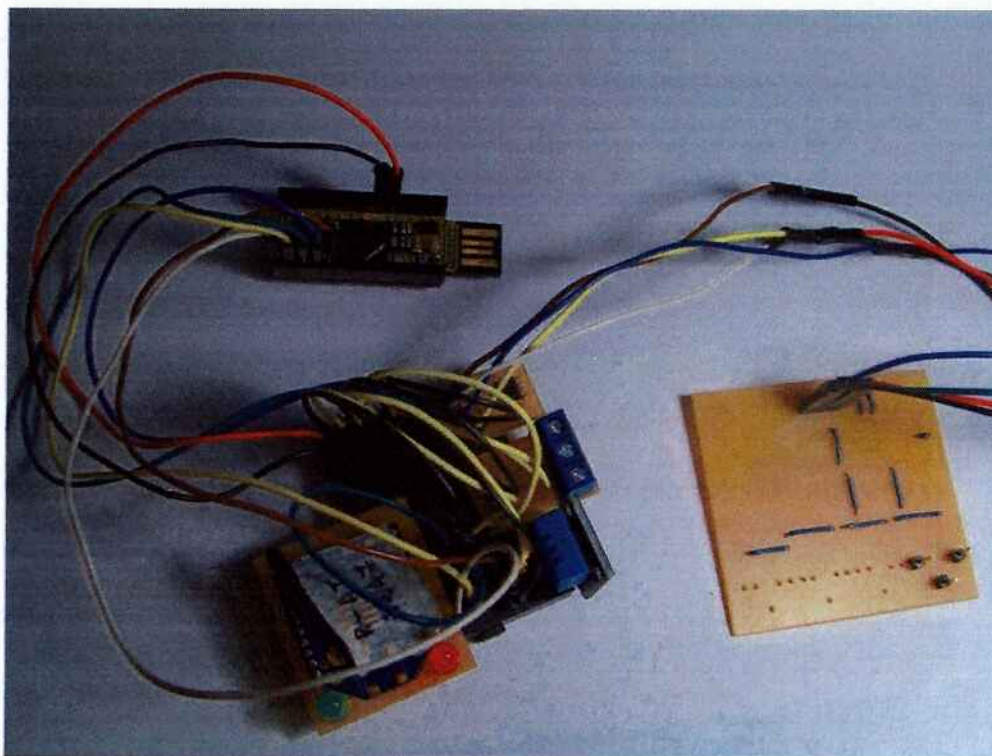


Figura 11 - Primeiro Protótipo

O core do sistema foi desenvolvido a partir da linguagem de programação Python, uma linguagem bastante versátil e que é usada pelo framework web Django, também usado na implementação do SI. Com o Django rodando no servidor, podemos usar as classes criadas em Python através do acesso às tabelas do banco de dados MySQL, e também podemos utilizar suas funções e dados que serão mostrados para o usuário. Essa parte do sistema é comumente chamada de server-side.

Para o usuário poder usar o sistema temos as páginas web foram criadas em HTML em meio à códigos Django e são basicamente estruturadas a partir de classes CSS, para termos um layout apresentável e bem definido para qualquer browser e mecanizadas a partir de códigos javascript, que permitem a dinamização do conteúdo apresentado. Essas páginas permitem a interação do usuário com o core do sistema. Nelas o usuário chama funções do Server-side para acessar e modificar o banco de dados, facilitando o uso do sistema para o usuário. Parte comumente chamada de client-side.

Isso é programado com base em um modelo muito utilizado no mercado, o ModelViewController (MVC). Nesse modelo separamos o código em três categorias: Model: código que implementa as classes do sistema e como o banco de dados será

organizado; View: código que implementa o que o usuário vê na tela; Controller: código que implementa as funções de obtenção, tratamento e retorno dos dados para as demais partes. Isso pode ser melhor entendido através da figura 12.

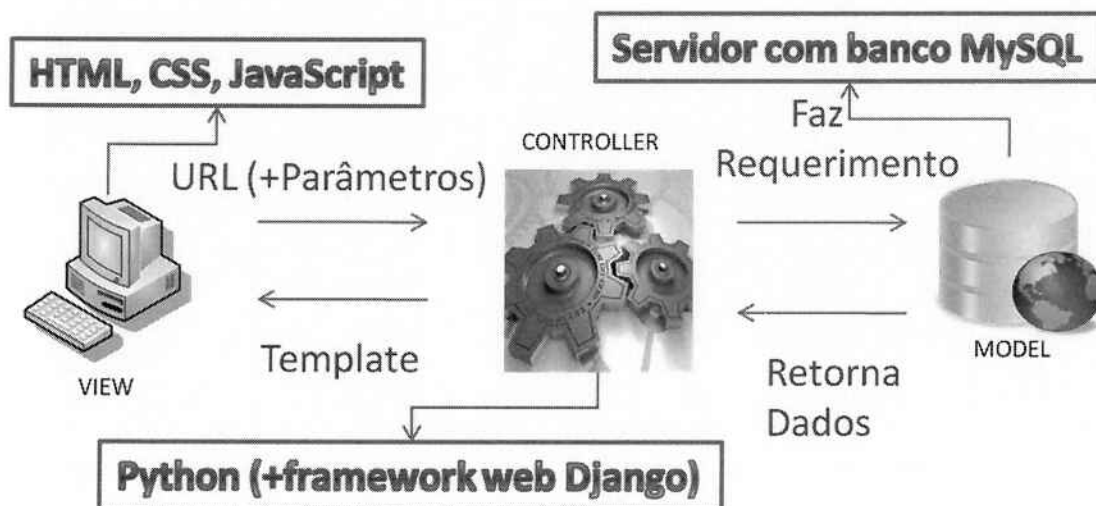


Figura 12 – Funcionamento MVC

No protótipo do projeto foi utilizado o software já em sua versão final. Caso surjam novas ideias e upgrades para o SI, os mesmos só serão implementados se o projeto sair de protótipo e ir para a fase comercial.

5 - Atividades realizadas

Com exceção das atividades que o grupo optou por retirar do projeto por diversos motivos já citados anteriormente, como a integração com o controlador de bateria e os dispositivos de irrigação, todas atividades foram realizadas conforme planejado, cada qual contribuindo para obtenção de resultados que levaram ao sucesso do trabalho.

5.1 - Plano de Integração e Depuração

Os principais pontos de integração do firmware são os seguintes:

- integração entre a aquisição de dados e a transmissão de dados;
- integração entre os diferentes métodos de aquisição de dados;
- integração com o banco de dados presente no microcontrolador;
- integração entre o scheduler e os diferentes processos.

Os sistemas de aquisição, transmissão de dados e obtenção deles pelo SI, até a sua visualização pelo usuário são feitos de forma independente, necessitando assim a integração de todos para o funcionamento do projeto como um todo.

A última parte do projeto consistiu na junção dessas partes para o pleno funcionamento do produto.

5.2 - Plano de Testes

Difícilmente o código de um software sai sem alguma falha logo na primeira tentativa de implementação, dessa forma é altamente necessário ter um plano de testes em todo desenvolvimento de software e reservar recursos para o mesmo no planejamento inicial.

Essas falhas podem surgir por diversos motivos, o mais comum é quando a implementação foi feita de modo incorreto com erro no próprio algoritmo, temos também muitas falhas provenientes de erros de especificação, que é quando se pensa que determinada função deve ter certo retorno, mas na real deve ter outro. Outro tipo menos comum de falha é quando, por limitações de hardware ou software, é simplesmente impossível de se executar o código.

Para se testar um sistema temos que elaborar os casos de testes, que consistem em definir as entradas que devem ser colocadas em cada função ou funcionalidade do mesmo e saber qual é resposta esperada para cada uma delas. Podemos perceber assim que quanto maior o sistema, mais casos de testes teremos e mais maçante será esse processo. Assim, temos que evitar testes descartáveis e focar nos que tem maior probabilidade de erro e impacto no funcionamento do sistema.

No caso do sistema do projeto Deméter não foi feito um documento tão detalhado a esse ponto. Para o teste desse sistema de informação foi criado um banco de dados povoado com dados falsos feitos somente para testes, o que é comumente chamado de *"mockup"*, e a partir disso serão testadas todas as funcionalidades, pois será conhecido o que devemos ter como respostas.

Outro ponto a ser levantado é que mensagens de erro especificando onde o erro ocorreu foram adicionadas em meio a pontos específicos do código, além de que o framework utilizado no desenvolvimento já provê uma ferramenta na qual essas falhas são facilmente identificadas.

5.2.1 - Dispositivo (unidade)

Além de testes em bancadas, foram realizados os seguintes testes em campo para o desenvolvimento do hardware:

- teste de distancia máxima de transmissão do sinal;
- curva de resposta do sensor de umidade do solo;
- verificação do consumo de energia do dispositivo em situação próxima da real;
- teste do sensor de temperatura e umidade do ar.

As funcionalidades testadas em software foram:

- usuário pode logar no sistema;
- usuário pode criar, ver, editar informações e remover fazendas;
- usuário pode criar, ver, editar informações e remover setores de fazendas;
- usuário pode verificar a umidade do solo dos setores de suas fazendas;
- usuário pode ver estatísticas sobre os dados de umidade do solo por setor;
- usuário pode ver gráficos com os dados de umidade do solo por setor;
- usuário pode verificar a temperatura atual, histórico e previsão do tempo para a região de sua fazenda;
- usuário pode verificar a umidade do ar atual, histórico e a previsão dos próximos dias para a região de sua fazenda;
- usuário pode criar, ver, editar informações e remover contatos de suas fazendas;
- usuário pode ter um calendário para cada fazenda;
- usuário pode criar, ver, editar e remover eventos de seus calendários;
- usuário pode ser avisado por e-mail quando um evento de seus calendários se aproxima.

5.2.2 –Integração

A integração dos módulos de hardware ocorreram sempre duas a duas, como por exemplo sensor/ADC, ADC/Microcontrolador e Microcontrolador/Transceptor, de forma a garantir o funcionamento e a correta comunicação entre as partes envolvidas em cada teste e assim o funcionamento da rede completa em um teste geral entre os módulos.

Em relação à integração entre cada módulo do sistema web, os mesmos são feitos em conjunto e sua integração já ocorre automaticamente com o teste de cada novo módulo. Podemos citar entre as integrações feitas:

- módulo de fazendas;
- módulo de setores;
- módulo de medições;
- módulo de histórico;
- módulo de contatos;
- módulo de mapas;
- módulo de previsão do tempo.

5.2.3 - Aceitação

A aceitação deu-se através da verificação de testes com propósitos específicos, confirmando a atuação do sistema em determinadas condições e atendendo às funcionalidades propostas.

Os integrantes da equipe e o orientador validaram os testes, posteriormente reportando aos professores da disciplina e demais membros da banca avaliadora do projeto.

Para o SI ser dado como aceito alguns critérios foram cumpridos. Dever-se-ia ter no mínimo as seguintes funções completamente funcionais:

- usuário pode logar no sistema:
 - teste básico de login com senha correta e incorreta;
 - deve ter um campo para recuperação de senha;
- usuário pode criar, ver, editar informações e remover fazendas:
 - duas fazendas não podem ter o mesmo nome para um mesmo usuário;
 - duas fazendas não podem ocupar a mesma posição no mapa;

- usuário pode criar, ver, editar informações e remover setores de fazendas:
 - dois setores não podem ter o mesmo nome para um mesmo usuário;
 - dois setores não podem ocupar a mesma posição no mapa;
- usuário pode verificar a umidade do solo dos setores de suas fazendas:
 - os dados devem ser adquiridos a partir da última atualização do banco de dados;
- usuário pode ver estatísticas sobre os dados de umidade do solo por setor:
 - o tratamento dos dados de um setor deve ser o mesmo dos demais;
- usuário pode ver gráficos com os dados de umidade do solo por setor:
 - a umidade será diferenciada por cores nos gráficos.

Além disso, o sistema teria que:

- conseguir receber os dados no formato enviado pelos sensores:
 - pode ser usada uma interface com outros softwares intermediários;
- sincronizar seu banco de dados com o servidor na internet:
 - o banco deve ser atualizado no mínimo a cada uma hora.

6– Resultados

Como resultados das atividades acima descritas, obteve-se um sistema que mede, transmite, interpreta e mostra de uma maneira intuitiva a situação da umidade do solo, nos diferentes setores da fazenda do cliente.

Realizados os testes preliminares de funcionamento do protótipo, o circuito foi encaminhado para um profissional especializado realizar a confecção do mesmo. O produto final, conforme apresentado na Figura 13, possui todos os componentes anteriormente citados, três entradas para sensores de umidade do solo, uma entrada sensor de temperatura, conector para o transceptor utilizado e entrada para a fonte de alimentação. Foram feitas três cópias do circuito, para realização de testes e por segurança.

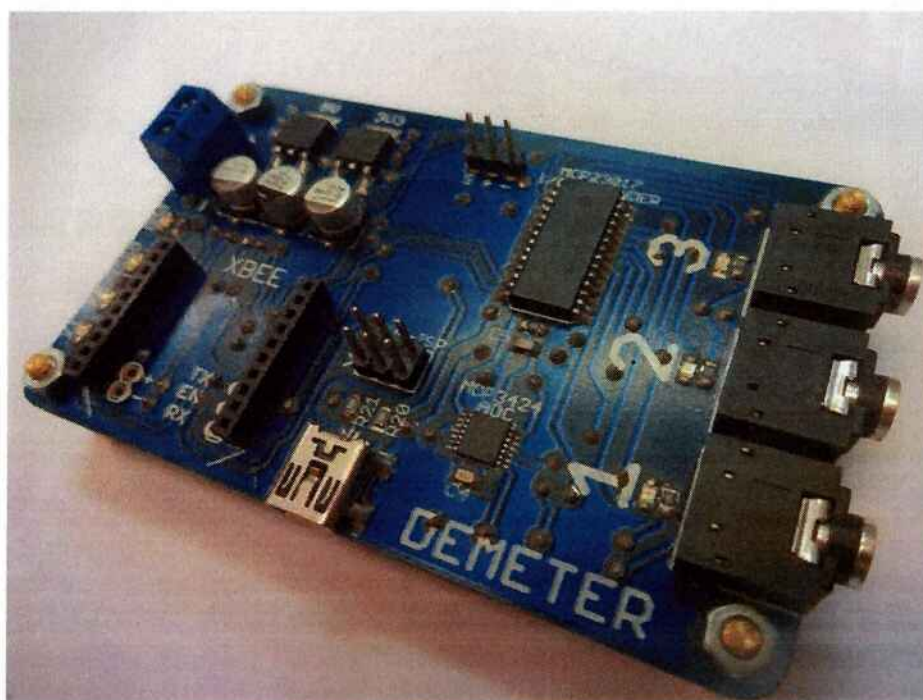


Figura 13 – Circuito final

Em relação ao Sistema de Informação, que atende essa última etapa de interpretação e visualização dos dados, as atividades foram basicamente de procurar entender o problema a ser solucionado com o SI e após isso a programação do código propriamente dito, sendo a metodologia já mencionada. Com isso, temos como resultado as telas abaixo e também o código, que poderá também ser utilizado para futuros projeto.

Na tela de login (figura 14), o usuário tem que entrar com seu nome de usuário e senha para poder acessar os dados de suas fazendas. Essa funcionalidade garante a possibilidade de acessar o sistema remotamente com segurança aos dados do cliente.

A imagem mostra a interface de login do 'Sistema Deméter'. No topo, o título 'Sistema Deméter' está em uma fonte preta e negrita. Abaixo dele, há um formulário branco com uma borda cinza. O formulário contém dois campos de entrada: 'Usuário:' com um ícone de pessoa no final da barra, e 'Senha:' com um ícone de cadeado no final da barra. À direita do campo de senha, há um link azul 'Esqueceu a senha?'. Na base do formulário, há uma opção 'Manter conectado' com uma caixa de seleção vazia e um botão azul 'Entrar' com um ícone de seta para a direita.

Figura 14 – Página de login do Sistema

Logo que o usuário efetua o login ele é levado para a página inicial, que contém todas as suas fazendas e as informações mais importantes das mesmas, além de ter acesso ao menu lateral esquerdo, no qual pode acessar também seus contatos e a página de previsão do tempo, e do menu superior, no qual pode configurar sua conta e acessar suas mensagens e notificações. A tela citada aqui pode ser vista na figura 15.

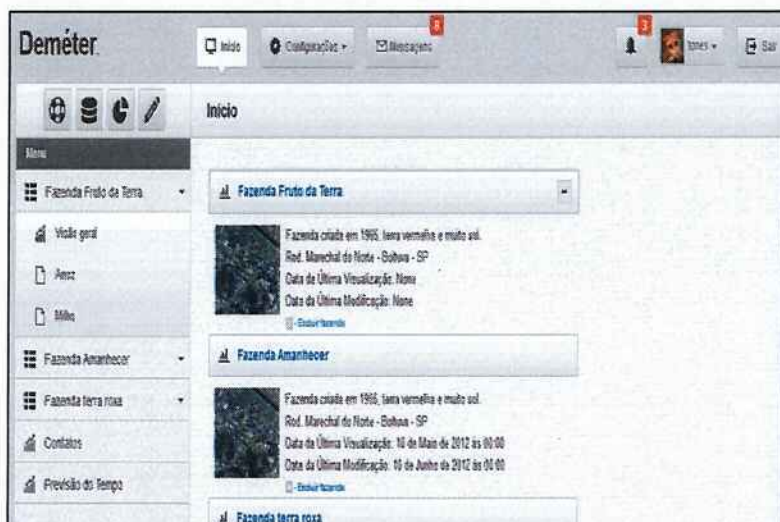


Figura 15 – Página Inicial do Sistema

Ao clicar em uma fazenda na página inicial, o usuário é levado para a página daquela fazenda específica e lá pode acessar mais dados da mesma e também tem a opção de editá-los, como podemos ver na figura 16.

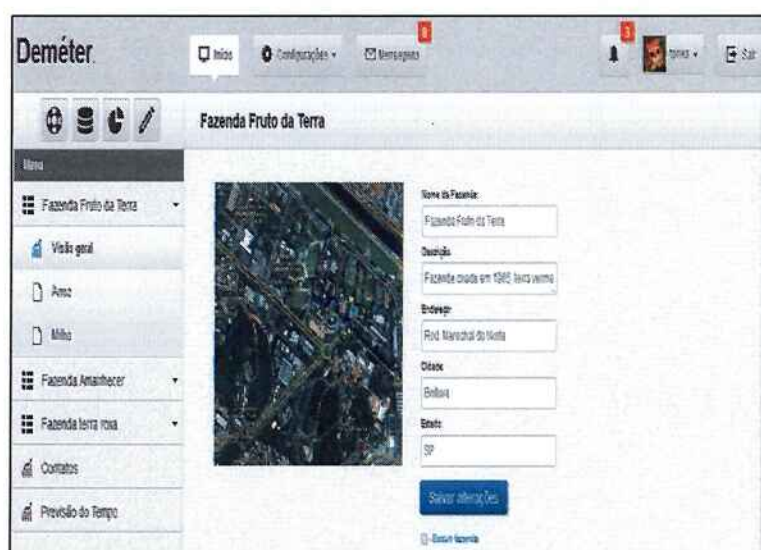


Figura 16 – Página da Fazenda do Sistema

Dentro de cada fazenda temos diversos setores, onde cada setor representa um sensor. Os setores também têm dados específicos, como tipo de cultura, localização e também os dados das medições anteriores realizadas pelos sensores. Essas medidas são mostradas em gráficos na página dos sensores, como podemos ver na figura 17.



Figura 17 – Página do Setor do Sistema

O módulo de histórico não foi implementado no protótipo apresentado, uma vez que para testar o sistema necessita-se apenas de medidas em períodos pequenos de tempo e uma futura atualização do sistema com essa funcionalidade é de fácil implementação.

Já para o módulo de atualização com dados em tempo real sobre a previsão do tempo, com dados atuais sobre temperatura e umidade, e o posterior armazenamento desses dados, a não implementação no protótipo desenvolvido foi devido à não ligação direta entre esses dados e o hardware construído. Nesse módulo, seria necessária ainda a integração com um sistema de algum site que fornecesse os dados requeridos e uma eventual autorização para uso dessa informação.

Outros módulos anteriormente descritos, como os relacionados com a criação de um calendário para auxiliar na organização dos eventos de cada fazenda, acabaram não sendo priorizados na implementação do software, pois tratam-se de adaptações de códigos já utilizados em outras plataformas e que não se relacionam diretamente com o equipamento desenvolvido para a medição de umidade do solo, foco principal do projeto.

Como dito anteriormente, temos também os códigos que podem ser reutilizados para futuros projetos. Abaixo encontra-se, a título de exemplo, os

códigos da pagina do setor. Conforme já explicado na parte de design, o código é dividido em View, figura 18, Controller, figura 19 e Model, figura 20.

```

        </div><!-- End .box -->
    </div><!-- End .span8 -->

    <div class="span4">
        <div class="sparkStats">
            <form method="POST" action="/fazenda/{{fazenda.id}}/setor/{{setor.id}}/">
                <div class="fazenda_form_box">
                    <span class="fazenda_form_text">Setor número:</span>
                    <span class="fazenda_form_text">{{setor_form.numero_setor}}</span>
                </div>
                <div class="fazenda_form_box">
                    <span class="fazenda_form_text">Cultura do Setor:</span>
                    <span class="fazenda_form_text">{{setor_form.cultura_setor}}</span>
                </div>
                <div class="fazenda_form_box">
                    <span class="fazenda_form_text">Descrição do Setor:</span>
                    <span class="fazenda_form_text">{{setor_form.descricao_setor}}</span>
                </div>
            </form>
        </div><!-- End .sparkStats -->
    </div><!-- End .span4 -->
</div><!-- End .row -->

```

Figura 18 – Código da View do Setor

```

#SETOR
@login_required
def cria_setor(request, id_fazenda):
    usuario = Usuario.objects.get(username=request.user.username)
    fazenda = Fazenda.objects.get(id=id_fazenda)
    setor = Setor()
    setor.fazenda = fazenda
    if request.method == 'POST':
        setor_form = SetorForm(request.POST, instance=setor)
        if setor_form.is_valid():
            setor = setor_form.save()
            messages.add_message(request, messages.SUCCESS,
                                url = reverse('setor', args=[fazenda.id, setor.id])
            return HttpResponseRedirect(url)
        else:
            messages.add_message(request, messages.ERROR, '
            return render_to_response('cria_setor.html', local:
    else:
        setor_form = SetorForm(instance=setor)
        return render_to_response('cria_setor.html', local:

@login_required
def setor(request, id_fazenda, id_setor):
    usuario = Usuario.objects.get(username=request.user.username)
    fazenda = Fazenda.objects.get(id=id_fazenda)
    setor = Setor.objects.get(id=id_setor)
    if request.method == 'POST':
        setor_form = SetorForm(request.POST, instance=setor)
        if setor_form.is_valid():
            setor = setor_form.save()
            messages.add_message(request, messages.SUCCESS,
                                url = reverse('setor', args=[fazenda.id, setor.id])

```

Figura 19 – Código do Controller do Setor

```

class Setor(models.Model):
    fazenda = models.ForeignKey(Fazenda)
    numero_setor = models.CharField("Número do Setor", max_
    cultura_setor = models.CharField("Cultura do Setor", ma
    descricao_setor = models.CharField("Descrição do Setor"

    def __unicode__(self):
        return self.nome_setor

class HistoricoSetor(models.Model):
    setor = models.ForeignKey(Setor)
    data_medida = models.DateTimeField("Data da Medida")
    id_modulo = models.CharField("ID do Módulo", max_length=
    medida1 = models.CharField("Medida 1", max_length=1024,
    medida2 = models.CharField("Medida 2", max_length=1024,
    medida3 = models.CharField("Medida 3", max_length=1024,
    medida4 = models.CharField("Medida 4", max_length=1024,

```

Figura 20 – Código do Models do Setor

A interface do sistema foi desenvolvida de forma a se mostrar moderna, completa e ainda assim amigável ao usuário. Pacotes de *layout* foram adquiridos na rede, a um preço bem acessível, de modo com que a equipe priorizasse a programação do sistema à questões de design e estéticas. Com isso foi possível basear-se no formato dos sistemas e sites mais utilizados na atualidade e implementar uma interface adequada.

A visualização dos mapas foi estabelecida pela utilização de uma API (*Application Programming Interface* – ou Interface de Programação de Aplicativos) do site Google Maps, disponível para utilização de desenvolvedores de sistemas e softwares. Contudo, a edição de setores através dos mapas ficou comprometida devido à dificuldade em encontrar uma ferramenta adequada e por não ser crucial ao protótipo. Com isso, convencionou-se que os setores iriam ser delimitados automaticamente pelo sensor utilizado, em que cada sensor corresponderia a um setor da plantação.

Para a integração dos sistemas de hardware e software, através de software de interface em C#, foram implementados as rotinas de recebimento de dados no formato enviado pelos sensores de umidade e o banco de dados sincronizado com a internet. Para este último, foi também utilizado servidor com Python, Django e MySQL devidamente configurados conforme detalhado anteriormente.

Para garantir o acesso por dispositivos móveis, fez-se o uso de html baseado no bootstrap (responsive html), que permite adaptação ao tamanho da tela com deslizamento e dos blocos do layout.

Realizada a integração do software com o hardware, foram realizados testes em ambiente controlado, conforme a Figura 21, onde uma jardineira foi preenchida com terra própria para o cultivo de plantas e vegetais, deixando-a secar ao sol para no momento do teste umedece-la de forma a deixar uma extremidade completamente úmida e a outra completamente seca, formando um gradiente na região central.

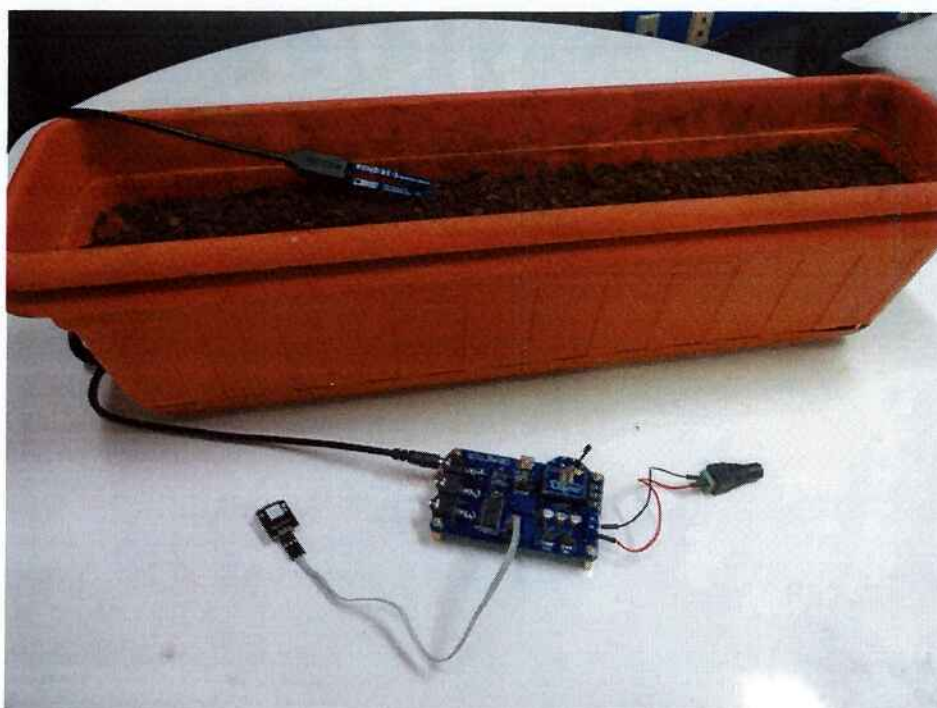


Figura 21 – Ambiente de testes

Com isso, pôde-se fazer os teste da aquisição dos dados através do sensor e transmissão dos mesmos através da rede.

O teste de distância resultou em aproximadamente 80m como alcance máximo para o transceptor utilizado em uma região totalmente aberta. Estima-se que efetivamente se possa utilizar este sensor para distâncias de até 50m, para que a taxa de erros esteja dentro do aceitável.

Uma vez recebidos, os dados foram interpretados pelo SI e dispostos graficamente nas regiões previamente delimitadas como alcance do sensor. O ponto central de cada setor, correspondente a localização de cada sensor, deve ser

implementado através de coordenadas no momento da instalação do sistema e aparecerá automaticamente com um mapa no SI, conforme a Figura 22.

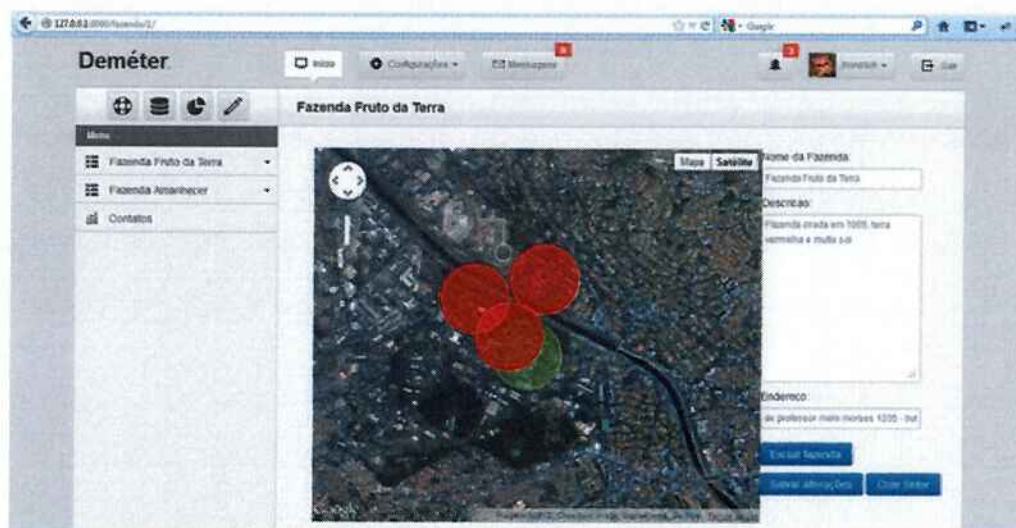


Figura 22 – Visualização dos setores

A cor de cada setor indica a necessidade ou não de se irrigar cada setor. Ela é definida através de uma comparação com um valor de referência da umidade do solo estabelecido para cada setor de forma independente, uma vez que varia de acordo com a cultura existente naquele setor. A cor vermelha indica que a umidade daquele setor está abaixo do valor de referência e a cor verde indica que a umidade está acima do valor de referência.

7– Informações Gerenciais

Durante a gestão do projeto, todos os tópicos de interesse nas nove áreas do conhecimento indicadas pelo PMBoK (Escopo, RH, Aquisições, Custo, Qualidade, Comunicação, Risco, Tempo e Integração) (PMI, 2008) vêm sendo monitoradas para garantir um bom andamento do projeto.

Como mencionado anteriormente, algumas tarefas inicialmente consideradas para o projeto acabaram se mostrando inviáveis de serem realizadas, como a fonte de alimentação por célula solar, por exemplo. Logo, os integrantes da equipe e o orientador decidiram por deixá-las como opções de aplicação em conjunto com o produto do projeto, garantindo assim uma pequena alteração no escopo, que não desviasse do propósito inicial do projeto.

A gestão de recursos do projeto vem sendo feita de modo que, os membros da equipe dividam suas atividades de acordo com sua especialidade e disponibilidade, uma vez que todos possuem atribuições acadêmicas e profissionais distintas durante este semestre. Desta forma, não há superalocação e o trabalho pode ser dividido para que o projeto tenha prosseguimento.

Quanto aos recursos de materiais e equipamentos, os mesmos vêm sendo adquiridos quando necessários, uma vez cabendo dentro do orçamento do projeto através do gerenciamento das aquisições do projeto. O levantamento dos fornecedores corretos e seus diferenciais no mercado mostram-se de extrema importância nesse âmbito e são feitos de forma criteriosa dentro do projeto. De forma prioritária, têm se tentado obter os componentes eletrônicos para o projeto por meio de empréstimos, como os módulos ZigBee, que gentilmente foi cedido pelos profs. Marcelo Zuffo, Sergio Kofuji e Ramona Straube, do departamento de Sistemas Eletrônicos da Poli-USP. O outro componente prioritário do projeto, o sensor de umidade capacitivo, será adquirido através de uma importadora. Decorrente de um atraso na liberação de verba para aquisição do mesmo, o estoque desse componente junto à fornecedora acabou, tendo a previsão de reposição apenas para metade do mês de novembro. Por este motivo, buscaram-se alternativas para este componente através de modelos similares, como o que foi obtido para realização de testes junto ao prof. Dr. Carlos Cugnasca. O sensor encomendado foi entregue apenas três semanas antes da apresentação final do projeto, o que acarretou um atraso não previsto no cronograma.

O custo do projeto ficou totalizado com os componentes adquiridos e a mão de obra terceirizada que fez-se necessária para a fabricação das placas.

Os principais componentes adquiridos e seus respectivos preços foram:

- sensor de umidade Echo EC-5: R\$ 500
- kit de desenvolvimento XBee: (obtido via empréstimo com o departamento do PSI na Poli-USP) R\$ 95
- componentes em geral: R\$ 89
- manufatura das placas de circuito impresso: R\$ 60 (Considerando a fabricação de uma placa. Considerando-se a fabricação de mais unidades o valor unitário deverá ser menor)

Para a gestão da comunicação do projeto, todos os membros participam do grupo de e-mails criado especialmente para este fim, tendo sido este, juntamente com as ligações telefônicas e reuniões presenciais, os canais mais utilizados pelos integrantes para discutir os eventos do projeto. Com o propósito de informar os demais stakeholders do projeto e todos os outros interessados, foi criado um website através da ferramenta google sites, onde todas as informações a respeito da equipe, da motivação, do andamento e detalhamento das atividades podem ser encontradas. A página em questão está hospedada do google e pode ser acessada através do domínio: <https://sites.google.com/site/projetodemeter/>



Figura 23 – Tela do Site de Acompanhamento

As atividades são acompanhadas semanalmente e foram atualizadas no cronograma do projeto para um melhor acompanhamento do projeto.

8 - Conclusão

Como resultado, obteve-se um sistema completo e funcional, que pode ser aplicado a fazendas de pequeno e médio porte.

Algumas ressalvas devem ser feitas no que se refere ao alcance do módulo ZigBee utilizado, que apresentou uma distância máxima de transmissão de apenas 100m, pouco para as dimensões de uma fazenda média. Existem outros módulos no mercado, porém com maior custo, que alcançam distâncias muito maiores.

Outro componente que pode ser substituído acaba sendo o sensor de umidade do solo, já que, devido à importação, representa a maior parte do custo do módulo e deixa assim o produto final com valor elevado. Outras alternativas poderiam ser buscadas com o uso de sensores não tão precisos e com uma menor área de abrangência. O desenvolvimento de um sensor capacitivo de precisão para umidade do solo com tecnologia nacional é ainda possível, utilizando-se os conhecimentos adquiridos durante o curso de engenharia de sistemas eletrônicos, porém não será abordado neste projeto por falta de tempo hábil para a elaboração de um projeto de qualidade.

O preço final de cada módulo sensor/transmissor fica por volta de R\$750, considerando-se os preços pagos durante o projeto. Esse valor ainda é muito elevado para que o sistema possa ser amplamente comercializado no país, tendo em vista que grande parte dos agricultores não teria como arcar com tais custos, mesmo visando otimizar seu cultivo e reduzir os gastos com água.

É necessário ainda um desenvolvimento de componentes de mais baixo custo para que o sistema possa ser comercialmente viável e seja assim amplamente aplicado em qualquer tipo de plantação.

Os integrantes da equipe puderam ter um contato direto com um projeto de engenharia completo, colocando em prática, muitos dos conhecimentos e competências adquiridas ao longo do curso de sistemas eletrônicos. Dessa forma puderam também contribuir diretamente para o avanço da pesquisa e tecnologia nacionais em prol de uma causa socioambiental de fundamental importância.

Fica a cargo de gerações futuras um aperfeiçoamento do que aqui foi desenvolvido para que a consciência da necessidade cada vez maior de racionamento desse recurso tão importante para vida seja feita e que este projeto possa também servir de inspiração para que outros trabalhos com esse propósito sejam realizados.

9 - Referências

GANDOLFO, M. A.; SAUER, A. L.; DE JESUS, F. T.; AFONSO, M – Demanda de Água Atual e Futura nas Aplicações de Agroquímicos

Distribuição da água no planeta

<http://educacao.uol.com.br/geografia/agua-potavel- apenas-3-das-aguas-sao-doces.jhtm> - Acessado em 10/04/2012

Embrapa – tipos de irrigação

<http://sistemasdeproducao.cnptia.embrapa.br> – Acessado em 31/07/2012

Sobre a Google Maps API

<https://developers.google.com/maps/?hl=pt-BR> – Acessado em 29/10/2012

Manifesto Ágil (Guia)

<http://manifestoagil.com.br/> - Acessado em 13/05/2012

Página do Projeto

<https://sites.google.com/site/projetodemeter/>

PMI – Project Management Institute

<http://www.pmi.org> - Acessado em 02/05/2012

USB/DC/Solar LipoCharger Design notes

<http://ladyada.net/products/usbdcsolarlipo/design.html> - Acessado em 15/04/2012

Utilização da água

<http://www.mananciais.org.br/2011/09/ usos-da-agua/> - Acessado em 10/04/2012