



CENTRO UNIVERSITÁRIO DE BRASÍLIA- UniCEUB  
FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS – FATECS  
CURSO DE ENGENHARIA DA COMPUTAÇÃO

AYRTON GISSONI ALMEIDA

# ADEGA CLIMATIZADA PARA VINHOS UTILIZANDO A CÉLULA DE PELTIER

BRASÍLIA – DF

1º SEMESTRE DE 2013

AYRTON GISSONI ALMEIDA

# ADEGA CLIMATIZADA PARA VINHOS UTILIZANDO A CÉLULA DE PELTIER

Trabalho apresentado ao Centro  
Universitário de Brasília (UniCEUB) como  
pré-requisito para a obtenção do grau de  
Bacharel em Engenharia de Computação.

Orientadora: Prof.<sup>a</sup>. MSc. Maria Marony  
Sousa Farias

Brasília

Julho, 2013

AYRTON GISSONI ALMEIDA

# ADEGA CLIMATIZADA PARA VINHOS UTILIZANDO A CÉLULA DE PELTIER

Trabalho apresentado ao Centro  
Universitário de Brasília (UniCEUB) como  
pré-requisito para a obtenção do grau de  
Bacharel em Engenharia de Computação.

Orientadora: Prof.<sup>a</sup>. MSc. Maria Marony  
Sousa Farias

Este Trabalho foi julgado adequado para a obtenção do Título de Engenheiro de  
Computação, e aprovado em sua forma final pela Faculdade de Tecnologia e  
Ciências Sociais Aplicadas - FATECS.

---

Prof. Abiezer Amarília Fernandes  
Coordenador do Curso

**Banca Examinadora:**

---

Prof.<sup>a</sup>. MSc. Maria Marony Sousa Farias  
Orientadora

---

Prof. MSc. Luís Cláudio Lopes de Araújo  
UniCEUB

---

Prof. Dr. Miguel Archanjo Bacellar Goes Telles Junior  
UniCEUB

---

Prof.<sup>a</sup>. MSc. Valéria de Assis Vasconcelos  
UniCEUB

*Dedico este trabalho à minha querida família,  
a minha maravilhosa namorada e amigos,  
por todo o apoio dado à minha pesquisa.*

Ayrton Gissoni Almeida

## **AGRADECIMENTOS**

*Agradeço em primeiro lugar a Deus, por estar sempre comigo em todos os momentos de minha vida me dando sabedoria e forças para correr atrás dos meus objetivos. A Nossa Senhora, que me abençoou ao fazer este trabalho. Ao meu anjo da guarda que sempre está cuidando de mim.*

*Agradeço a minha mãe Yara Gissoni Almeida, por sempre me incentivar a estudar, por ter me dado a chance de cursar a faculdade e chegar onde eu estou, e a minha família, meu pai Paulo Felinto de Almeida e minha irmã Rebecca Gissoni Almeida que me ajudou muito neste projeto.*

*A minha maravilhosa namorada Talita Silva Cortês, que eu amo muito e que está comigo em todos os momentos, sendo sempre minha companheira.*

*Aos meus grandes amigos da minha turma Gilberto, Rogério, Tomás, Marcos e Luís que sempre estiveram todos unidos nessa grande jornada da faculdade.*

*Aos meus professores maravilhosos que me ajudaram muito, especialmente ao professor Luciano Duque, Fabiano Mariath e Francisco Javier.*

*Aos meus amigos José Carlos, Gilberto de Paula, Tomás da Silva, Felipe Belluco e Marcos Paulino pela ajuda e paciência no desenvolvimento deste projeto que se tornou fundamental para a sua completa realização.*

*A minha querida orientadora Maria Marony, pela compreensão e imensa ajuda.*

*A todos;*

*Muito obrigado!*

*Ayrton Gissoni Almeida*

*"Seja você quem for, seja qual for a posição social que você tenha na vida, a mais alta ou a mais baixa, tenha sempre como meta muita força, muita determinação e sempre faça tudo com muito amor e com muita fé em Deus, que um dia você chega lá. De alguma maneira você chega lá."*

Ayrton Senna

# SUMÁRIO

CAPÍTULO 1 INTRODUÇÃO .....	16
1.1 Contextualização do Projeto .....	16
1.2 Objetivo Geral .....	17
1.3 Objetivos Específicos.....	18
1.4 Motivação.....	18
1.5 Resultados Esperados .....	18
1.6 Escopo .....	19
1.7 Estrutura do trabalho .....	20
CAPÍTULO 2 REFERENCIAL TEÓRICO.....	21
2.1 O Vinho e suas Formas de Acondicionamento .....	21
2.1.1 História do Vinho.....	21
2.1.2 O Vinho.....	23
2.1.3 Principais Cepas.....	24
2.1.4 Rótulos.....	25
2.1.5 A Guarda do Vinho .....	26
2.1.6 A Importância da Temperatura .....	27
2.1.7 Modelo Atual de Adega Climatizada .....	28
2.1.8 Limitações do Modelo Atual .....	30
2.2 Fenômenos de Transporte de Calor.....	31
2.2.1 A Transferência de Calor .....	31
2.2.1.1 Condução.....	32
2.2.1.2 Convecção .....	33
2.2.1.3 Radiação .....	34
2.2.2 Calorimetria.....	35
2.2.2.1 Calor Sensível .....	36
2.2.2.2 Calor Latente .....	36
2.3 Fenômenos Termoelétricos .....	37
2.3.1 Efeito Seebeck .....	37
2.3.2 Efeito de Peltier.....	39
2.3.3 Efeito Thomson .....	41
2.4 Célula de Peltier .....	42

2.4.1	Funcionamento da Célula de Peltier .....	42
2.4.2	Cuidados na Montagem do Módulo de Peltier .....	44
2.4.3	Estágios do Módulo de Peltier .....	45
2.4.4	Dimensionamento da Célula de Peltier por Carga Térmica .....	46
2.4.4.1	Carga Ativa .....	46
2.4.4.2	Carga Passiva .....	47
CAPÍTULO 3 DESENVOLVIMENTO DO PROTÓTIPO DA ADEGA CLIMATIZADA .....		50
3.1	Modelo de Adega Climatizado Proposto.....	50
3.2	Descrição dos Materiais Utilizados .....	52
3.2.1	Arduino.....	52
3.2.2	Estrutura da Adega.....	55
3.2.3	Sensor de Temperatura.....	57
3.2.4	Módulo de Relés.....	58
3.2.5	Célula de Peltier .....	59
3.2.5.1	Dimensionamento da Célula de Peltier por Calorimetria .....	59
3.2.5.2	Dimensionamento da Célula de Peltier por Carga Térmica .....	61
3.2.5.3	A Escolha da Célula de Peltier .....	63
3.2.6	Fonte .....	64
3.2.7	Cooler com Dissipador.....	64
3.3	Montagem do <i>Hardware</i> .....	65
3.4	<i>Software</i> de Controle de Temperatura - Arduino .....	70
CAPÍTULO 4 DESENVOLVIMENTO DO <i>SOFTWARE</i> DE CADASTRAMENTO DOS VINHOS.....		72
4.1	Descrição da Arquitetura do <i>Software</i> .....	72
4.1.1	Modelo .....	73
4.1.1.1	DTO ( <i>Data Transfer Object</i> ).....	74
4.1.1.2	DAO ( <i>Data Access Object</i> ) .....	75
4.1.1.3	FACADE.....	77
4.1.2	Controle.....	78
4.1.2.1	BEAN.....	79
4.1.3	Visão .....	81
4.2	Banco de Dados .....	82
4.2.1	Modelo de Entidade de Relacionamento.....	82
4.2.2	JDBC e Classe de Conexão .....	84
4.3	Interface da Aplicação .....	85

4.3.1	Descrição das Telas do Sistema.....	86
CAPÍTULO 5 TESTES E RESULTADOS .....		92
5.1	Testes Realizados para o Protótipo.....	92
5.2	Teste do <i>Software</i> de Cadastramento dos vinhos .....	92
5.3	Dificuldades Encontradas / Soluções .....	95
5.3.1	Corrente elétrica .....	95
5.3.2	Eficiência da Célula de Peltier .....	95
5.3.3	Ruídos no Sensor LM35.....	96
5.3.4	Quantidade de Beans .....	96
5.3.5	Escopo dos Beans .....	97
5.3.6	Componentes do <i>Framework</i> .....	97
5.3.7	Comunicação Java e Arduino .....	98
5.4	Resultados Obtidos .....	98
5.4.1	Temperatura dos Compartimentos.....	98
5.4.2	Protótipo .....	101
5.5	Produto Gerado.....	103
CAPÍTULO 6 CONSIDERAÇÕES FINAIS.....		104
6.1	Conclusão .....	104
6.2	Proposta para Futuros Projetos .....	106
REFERÊNCIAS.....		108
APÊNDICE - A.....		109
APÊNDICE - B .....		116
Cenário 1	.....	116
Cenário 2	.....	117
Cenário 3	.....	117
Cenário 4	.....	117
Cenário 5	.....	118
Cenário 6	.....	118
Cenário 7	.....	119
Cenário 8	.....	120
Cenário 9	.....	120
APÊNDICE – C .....		122
ANEXO – A .....		123
ANEXO – B .....		125

ANEXO - C.....	131
ANEXO - D.....	134

## LISTA DE FIGURAS

FIGURA 1.1 - ADEGA FRANCESA (FONTE: O GRANDE LAROUSSE DO VINHO, 2012) .....	17
FIGURA 2.1 - PRINCIPAIS TIPOS DE VINHOS (AUTOR: AYRTON GISSONI).....	24
FIGURA 2.2 - CABERNET-SAUVIGNON E CHARDONNAY (FONTE: O GRANDE LAROUSSE DO VINHO, 2012).....	25
FIGURA 2.3 - RÓTULO DE UM VINHO DE BORDEAUX (FONTE: O GRANDE LAROUSSE DO VINHO, 2012).....	26
FIGURA 2.4 - ADEGA CLIMATIZADA (FONTE: SITE DAS LOJAS AMERICANAS, AMERICANAS.COM.BR) .....	27
FIGURA 2.5 - ESQUEMA DE UMA ADEGA QUE UTILIZA UM COMPRESSOR (FONTE: HTTP://PORDENTRODAADEGA.COM.BR) .....	29
FIGURA 2.6 - ADEGA TERMOELÉTRICA (FONTE:HTTP://PORDENTRODAADEGA.COM.BR) .....	29
FIGURA 2.7 - LIMITES DO MODELO ATUAL (FONTE:HTTP://WWW.COLOMBO.COM.BR/PRODUTO/ELETRODOMESTICOS/ADEGA-16-GARRAFAS- ADE160-CADENCE).....	30
FIGURA 2.8 -MECANISMOS DE TRANSFERÊNCIA DE CALOR (FONTE: HTTP://FISICA.UFPR.BR/GRIMM/AOSMETEO/CAP2/CAP2-9.HTML).....	31
FIGURA 2.9 - TRANSMISSÃO DE CALOR (FONTE: RAMALHO, NICOLAU, TOLEDO - 2006).....	35
FIGURA 2.10 – EXPERIÊNCIA DE SEEBECK (DESENHADO POR AYRTON GISSONI) .....	38
FIGURA 2.11 - EXPERIÊNCIA DE PELTIER (DESENHADO POR AYRTON GISSONI) .....	39
FIGURA 2.12 – REPRESENTAÇÃO ESQUEMÁTICA DA CÉLULA DE PELTIER (FONTE: HTTP://WWW.MSPC.ENG.BR/ELETRN/PELTIER_110.SHTML) .....	40
FIGURA 2.13 – ESQUEMA DOS EFEITOS DE SEEBECK, PELTIER E THOMSON (FONTE: DISSERTAÇÃO DE MESTRADO, INPE, VÂNIA APARECIDA DA COSTA).....	41
FIGURA 2.14 - ESQUEMA DA CÉLULA DE PELTIER (FONTE: SITE DO FABRICANTE, HTTP://WWW.PELTIER.COM.BR/FAQS.HTM).....	43
FIGURA 2.15 - VISÃO INTERNA DA CÉLULA DE PELTIER E DO EFEITO DE PELTIER (FONTE: SITE DO FABRICANTE, HTTP://WWW.PELTIER.COM.BR/FAQS.HTM).....	44
FIGURA 2.16 - ESQUEMA DE MONTAGEM DA CÉLULA DE PELTIER (FONTE: SITE DO FABRICANTE, HTTP://WWW.PELTIER.COM.BR/FAQS.HTM).....	45
FIGURA 2.17 - MÓDULOS DE PELTIER (FONTE: MONOGRAFIA: CONDICIONADOR DE AR PARA CABINE DE CAMINHÃO, UTILIZANDO CÉLULAS DE PELTIER, DANIEL DE SOUZA, 2011).....	46
FIGURA 3.1 - VISÃO GERAL DA ADEGA CLIMATIZADA (AUTOR: AYRTON GISSONI) .....	52
FIGURA 3.2 – ARDUINO MEGA ATMEGA1280 (AUTOR: AYRTON GISSONI) .....	54
FIGURA 3.3 - AMBIENTE DE DESENVOLVIMENTO DO ARDUINO (AUTOR: AYRTON GISSONI) .....	55
FIGURA 3.4 - ESTRUTURA DA ADEGA (AUTOR: AYRTON GISSONI).....	56
FIGURA 3.5 - ESTRUTURA DO PROTÓTIPO (AUTOR: AYRTON GISSONI) .....	57
FIGURA 3.6 - DATASHEET LM35 (FONTE: NATIONAL SEMICONDUCTOR) .....	58
FIGURA 3.7 - MÓDULO DE RELÉS (AUTOR: AYRTON GISSONI) .....	59
FIGURA 3.8 - CÉLULA DE PELTIER (AUTOR: AYRTON GISSONI) .....	63
FIGURA 3.9 - COOLER COM DISSIPADOR (AUTOR: AYRTON GISSONI) .....	65
FIGURA 3.10 - DISSIPADOR DA FACE QUENTE (FONTE: HTTP://WWW.PELTIER.COM.BR/PRODUTOS.HTM).....	65
FIGURA 3.11 - ESQUEMA DE MONTAGEM DO <i>HARDWARE</i> (AUTOR: AYRTON GISSONI) .....	66
FIGURA 3.12 - ESQUEMA ELÉTRICO DE CLIMATIZAÇÃO (AUTOR: AYRTON GISSONI) .....	68
FIGURA 3.13 - ESQUEMA ELÉTRICO DE CONTROLE DE TEMPERATURA (AUTOR: AYRTON GISSONI) .....	69
FIGURA 3.14 - LIGAÇÃO DO SENSOR LM35 (AUTOR: AYRTON GISSONI).....	70
FIGURA 4.1 - ARQUITETURA MVC (AUTOR: AYRTON GISSONI).....	73
FIGURA 4.2 - CÓDIGO JAVA DTO (AUTOR: AYRTON GISSONI).....	75
FIGURA 4.3 - CÓDIGO JAVA DAO (AUTOR: AYRTON GISSONI) .....	76
FIGURA 4.4 - CÓDIGO JAVA MÉTODO BUSCARTODOS (AUTOR: AYRTON GISSONI) .....	77
FIGURA 4.5 - CÓDIGO JAVA FACADE (AUTOR: AYRTON GISSONI).....	78
FIGURA 4.6 - CÓDIGO JAVA MÉTODO INSERIR (AUTOR: AYRTON GISSONI) .....	79

FIGURA 4.7 - CÓDIGO JAVA CLASSE GARRAFABEAN (AUTOR: AYRTON GISSONI).....	80
FIGURA 4.8 - CÓDIGO JAVA DA CLASSE GARRAFAALTERARBEAN (AUTOR: AYRTON GISSONI).....	81
FIGURA 4.9 - MER (AUTOR: AYRTON GISSONI).....	83
FIGURA 4.10 - CÓDIGO JAVA CLASSE CONEXÃO (AUTOR: AYRTON GISSONI) .....	85
FIGURA 4.11 - TELA INICIAL (AUTOR: AYRTON GISSONI).....	86
FIGURA 4.12 - TELA INSERIR GARRAFA (AUTOR: AYRTON GISSONI) .....	87
FIGURA 4.13 - TELA PESQUISAR GARRAFA (AUTOR: AYRTON GISSONI).....	88
FIGURA 4.14 - TELA VISUALIZAR GARRAFA (AUTOR: AYRTON GISSONI) .....	88
FIGURA 4.15 - TELA DE EXCLUSÃO (AUTOR: AYRTON GISSONI).....	89
FIGURA 4.16 - TELA ALTERAR GARRAFA (AUTOR: AYRTON GISSONI) .....	90
FIGURA 4.17 - TELA ADEGA GARRAFA (AUTOR: AYRTON GISSONI) .....	91
FIGURA 5.1 - INSERIR VINHO NO BANCO (AUTOR: AYRTON GISSONI).....	93
FIGURA 5.2 - PESQUISA DO VINHO (AUTOR: AYRTON GISSONI).....	93
FIGURA 5.3 - ALTERAÇÃO DO VINHO NO BD (AUTOR: AYRTON GISSONI) .....	93
FIGURA 5.4 - VINHO INSERIDO NA ADEGA (AUTOR: AYRTON GISSONI) .....	94
FIGURA 5.5 - EXCLUSÃO DO VINHO NO BD (AUTOR: AYRTON GISSONI).....	94
FIGURA 5.6 - TEMPERATURA DO COMPARTIMENTO 1 (AUTOR: AYRTON GISSONI).....	99
FIGURA 5.7 - TEMPERATURA DO COMPARTIMENTO 2 (AUTOR: AYRTON GISSONI).....	100
FIGURA 5.8 - TEMPERATURA DO COMPARTIMENTO 3 (AUTOR: AYRTON GISSONI).....	101
FIGURA 5.9 - VISTA POSTERIOR DO PROTÓTIPO (AUTOR: AYRTON GISSONI).....	102
FIGURA 5.10 - VISTA FRONTAL DO PROTÓTIPO (AUTOR: AYRTON GISSONI).....	102

## LISTA DE TABELAS

TABELA 1- CONDUTIVIDADE TÉRMICA DOS MATERIAIS A 300K (FONTE: BENNETT) .....	33
TABELA 2 - CONDUTIBILIDADE TÉRMICA.....	48
TABELA 3 - COEFICIENTE DE TRANSFERÊNCIA DE CALOR CONVECTIVO .....	49
TABELA 4 - ESPECIFICAÇÕES ARDUINO MEGA ATMEGA1280 .....	53
TABELA 5 - MATERIAIS/CUSTOS .....	103

## LISTA DE EQUAÇÕES

EQUAÇÃO 2.1 - EQUAÇÃO DE FOURIER .....	32
EQUAÇÃO 2.2 - EQUAÇÃO DO PRINCÍPIO DAS TROCAS DE CALOR .....	35
EQUAÇÃO 2.3 - EQUAÇÃO FUNDAMENTAL DA CALORIMETRIA .....	36
EQUAÇÃO 2.4 - EQUAÇÃO DO CALOR LATENTE .....	36
EQUAÇÃO 2.5 - COEFICIENTE DE SEEBECK .....	38
EQUAÇÃO 2.6 - CALOR DE PELTIER.....	40
EQUAÇÃO 2.7 - EQUAÇÃO FUNDAMENTAL DA TERMOELETRICIDADE .....	42
EQUAÇÃO 2.8 - EQUAÇÃO DA CARGA ATIVA .....	47
EQUAÇÃO 2.9 - CARGA PASSIVA DE RADIAÇÃO .....	47
EQUAÇÃO 2.10 - CARGA PASSIVA DE CONVECÇÃO .....	47
EQUAÇÃO 2.11 - CARGA PASSIVA DE CONDUÇÃO .....	48
EQUAÇÃO 2.12 - CARGA PASSIVA COMBINADA.....	49

## RESUMO

Neste trabalho, foi proposta a construção de uma adega climatizada que serve para armazenar com qualidade os vinhos durante o seu envelhecimento. Para isso, foi criado um modelo de adega de três compartimentos para manter guardados os diferentes tipos de vinhos e um sistema que permite o controle da climatização e localização dos mesmos. Na construção do protótipo da adega climatizada, foi utilizada a célula de Peltier para fazer a climatização, o microcontrolador Arduino para o controle de temperatura, a linguagem Java e o banco de dados MySQL para fazer o sistema de cadastro de vinhos.

**Palavras chaves:** Adega Climatizada, Arduino, Climatização, Controle, Microcontrolador, Peltier, Temperatura, Vinho, Cadastro de Vinho, Java, Banco de Dados, MVC.

## **ABSTRACT**

In this work, it was proposed to build a wine cellar for storing quality wines during aging. For this, we created a model of cellar three compartments to keep kept the different types of wines and a system that allows control of climate and location thereof. In building the prototype of the wine cellar, we used the Peltier cell to the climate, the Arduino microcontroller to control the temperature and the Java language and the MySQL database to make the registration system for wines.

**Keywords:** Heated Cellar, Arduino, Climate, Control, Microcontroller, Peltier, Temperature, Wine, Wine Master, Java, Database, MVC.

# CAPÍTULO 1 INTRODUÇÃO

## 1.1 Contextualização do Projeto

Uma das maiores preocupações dos apreciadores de vinhos se refere a sua guarda, pois com o tempo o vinho jovem evolui seus aromas, muda sua cor, seu sabor e se torna mais complexo. Um vinho jovem de boa qualidade e procedência pode se beneficiar do envelhecimento e atingir o seu apogeu (o ápice de qualidade).

Segundo Lepré, para o que um vinho possa envelhecer, é necessário que ele seja armazenado em boas condições. Para isso utiliza-se a adega que deve seguir algumas condições e tomar alguns cuidados:

- A temperatura deverá permanecer numa faixa entre 6 graus Celsius a 18 graus Celsius, evitando sempre alterações bruscas;
- A posição de guarda do vinho deve ser na horizontal, para que o vinho permaneça em contato com a rolha, evitando seu ressecamento, pois o ressecamento da rolha acaba permitindo a entrada de oxigênio no recipiente do vinho, causando sua oxidação;
- A umidade ideal é próxima de 65%. Um local muito úmido pode fazer com que os rótulos se descolem e a rolha seja embolorada, estragando assim o vinho. Um local com pouca umidade pode fazer com que a rolha resseque e conseqüentemente o vinho seja oxidado;
- A vibração pode estragar a qualidade do vinho e fazer com que ele envelheça rapidamente;
- A luz desencadeia reações químicas que degradam e estragam o vinho, por isso deve-se armazenar os vinhos preferencialmente em locais escuros ou de pouca luminosidade;
- Possuir uma boa ventilação, sem correntes de ar, para que seja feita a circulação de ar e a convecção de temperatura;

(GRANDE LAROUSSE DO VINHO, 2012).

Uma adega climatizada serve para manter uma temperatura adequada para a devida preservação da qualidade dos vinhos, cuidando de sua conservação, aroma e sabor. Os vinhos de guarda devem ser armazenados preferencialmente em uma adega climatizada, pois exigem maiores cuidados com sua guarda e envelhecimento, mas mesmo os vinhos jovens de consumo imediato também são beneficiados ao serem guardados em adega climatizada, como por exemplo: durante as estações mais quentes do ano.

Na Figura 1.1 é mostrada um dos exemplos de uma adega antiga francês, as quais eram feitas de pedra e localizadas no subsolo.



Figura 1.1 - Adega Francesa (Fonte: O Grande Larousse do Vinho, 2012)

## 1.2 Objetivo Geral

O objetivo deste trabalho é o de projetar e construir uma adega climatizada que possui três compartimentos e um *software* que auxilia no cadastro e localização dos diferentes tipos de vinhos: tinto, branco e espumante. A adega utiliza como base para a climatização a célula de Peltier cuja variação de temperatura é controlada automaticamente com variabilidade máxima de mais ou menos três graus Celsius.

### 1.3 Objetivos Específicos

1. Projetar um *Hardware* capaz de controlar a temperatura de cada compartimento da adega via célula de Peltier;
2. Desenvolver um *software* para cadastramento dos vinhos utilizando a linguagem Java e o banco de dados MySQL;
3. Realizar testes operacionais do *hardware* a fim de se buscar uma melhor eficiência para a climatização, utilizando a célula de Peltier.

### 1.4 Motivação

Este projeto tem como motivação a construção de uma adega climatizada de três compartimentos e de uso pessoal, pois o usuário tem a necessidade de acomodar os diferentes tipos de vinho, os quais possuem uma diferença de temperatura de guarda que deve ser obedecida para que o vinho envelheça corretamente. O usuário desprende um enorme tempo ao procurar garrafa por garrafa de vinhos em sua adega, visto que essa não tem um sistema que melhora a eficiência na hora de se procurar, identificar e cadastrar o vinho que se deseja.

### 1.5 Resultados Esperados

Espera-se com este projeto construir um protótipo de adega climatizada que utilize para a climatização a célula de Peltier, que contenha três compartimentos separados e isolados termicamente para a guarda de vinhos tintos, brancos e espumantes e que tenha um sistema que permita o cadastro de cada garrafa de vinho que seja guardado na adega. O custo do projeto seja viável do ponto de vista econômico, tendo como foco usuários domésticos.

Espera-se que a climatização do vinho feita pela célula de Peltier atinja a temperatura desejada pelo usuário e possa fazer a climatização de cada compartimento, mantendo a temperatura do vinho com uma variação de no máximo três graus Celsius.

## 1.6 Escopo

O escopo deste projeto consiste em construir um protótipo que contenha três compartimentos separados e isolados termicamente para a guarda de vinhos tintos, brancos e espumantes. O escopo não contempla a guarda de vinhos rosés, sobremesa e fortificados (do porto). A eventual guarda de vinhos rosés será feita no compartimento destinado aos brancos, pois parte da faixa de temperatura de armazenamento desses dois tipos de vinhos coincidem.

A adega climatizada é feita observando a amplitude de variação térmica de Brasília, que varia normalmente entre 14 graus Celsius a 30 graus Celsius. O tempo de climatização não é rápido a ponto de ser praticamente instantâneo (como o de um congelador freezer), pois não é bom para o vinho ter mudanças bruscas de temperatura. Por isso a climatização é feita de forma mais lenta (levando um tempo maior).

A umidade é desprezada para o tratamento da climatização, visto que o uso de aparelhos umidificadores e desumidificadores para construção do protótipo torna o projeto inviável pelo seu custo.

O protótipo deve ser instalado em ambientes que estejam fora de fontes de calor e fora do alcance direto dos raios solares, pois essas fontes de calor prejudicam a eficiência da adega.

O tempo de abertura e fechamento da porta para armazenamento e retirada dos vinhos da adega, deverá ser feito em curto espaço de tempo, para evitar o aquecimento dos compartimentos do protótipo. Para efeito de climatização, é sempre considerado que a porta do protótipo esteja sempre fechada.

O escopo do protótipo contém três compartimentos para armazenar os vinhos tintos, brancos e espumantes. Como se trata de um protótipo (pequena escala e de uso doméstico), cada compartimento armazena somente 2 garrafas, possuindo assim 6 garrafas na adega climatizada. O *software* simula a realidade do protótipo, possuindo a mesma quantidade de garrafas para cada compartimento.

O *software* não aborda tratamentos de erros realizados pelos usuários, como erros de digitação e erros em cadastrar o vinho com informações erradas. Para o protótipo será utilizado somente o computador (*notebook*) para rodar a

aplicação de cadastramento dos vinhos e controle de temperatura, bem como exibir seus dados. Caso venha ser desenvolvido um modelo comercial desta adega, um *tablet* poderá ser utilizado acoplado a sua estrutura física.

## 1.7 Estrutura do trabalho

A estrutura do trabalho se divide nos seguintes capítulos:

O Capítulo 2 será apresentado o referencial teórico sobre os vinhos, suas formas de acondicionamento e a importância da temperatura; sobre fenômenos físicos de transporte de calor como a condução, convecção e radiação, estudo da calorimetria; fenômenos termoelétricos e efeito de Peltier; a célula de Peltier, como funciona e como dimensionar sua carga térmica.

No Capítulo 3 é apresentado o desenvolvimento do protótipo, mostrando descrição da construção e a descrição dos materiais utilizados.

No Capítulo 4 é apresentado o desenvolvimento da construção do sistema de cadastramento de vinhos para a adega climatizada, utilizando a arquitetura de desenvolvimento MVC.

O Capítulo 5 apresenta os testes realizados no protótipo, resultados, principais dificuldades encontradas e soluções.

No Capítulo 6 são apresentadas as considerações finais como a conclusão do trabalho e as propostas para futuros projetos.

## **CAPÍTULO 2 REFERENCIAL TEÓRICO**

Este capítulo apresenta as bases teóricas para a resolução do problema apresentado no capítulo anterior, apresentando os principais conceitos sobre os vinhos e suas formas de acondicionamento, ressaltando a importância da temperatura; o referencial teórico sobre os fenômenos físicos que envolvem a transmissão de calor; os fenômenos termoelétricos, tendo como principal fenômeno o efeito de Peltier e conceitos teóricos sobre o funcionamento da célula de Peltier, como também ferramentas para o cálculo do dimensionamento da carga da célula de Peltier.

### **2.1 O Vinho e suas Formas de Acondicionamento**

#### **2.1.1 História do Vinho**

Não é possível dizer exatamente quem e quando começou a produzir o vinho. O fato é que os arqueólogos encontraram sementes, que têm mais de 7000 anos, que provinham de uvas cultivadas no Cáucaso, ao leste do Mar Negro. Pesquisadores encontraram no Egito e na Síria vestígios da espécie *Vitis vinifera*, datando de 4000 a.C. Fatos que indicam que o vinho é uma bebida milenar de grande importância histórica.

O primeiro ponto marcante do vinho na história foi em 2500 a.C, com a importância dada ao vinho pelas civilizações grega e romana, onde o vinho tornou-se um dos elementos expressivos, pelo seu papel religioso e ritualístico, na figura do deus do vinho: Dionísio para os gregos e Baco para os romanos.

O Império Romano, desde sua ascensão, deu grande impulso ao consumo do vinho. O cristianismo, por sua vez, adotou o vinho em seu ritual eucarístico, consolidando o uso do vinho já adotado pelos romanos.

No início da Idade Média, com a decadência do Império Romano, a Europa viveu uma fase de grande turbulência e somente a Igreja Católica necessitava do vinho e poderia assegurar sua continuidade. Foi nesse período que o vinho encontrou sua maior aliada: a Igreja, onde coube aos monges católicos

continuar a produção do vinho, cuja tarefa foi desempenhada com grande perspicácia, pois os monges não se contentaram somente em produzir o vinho, mas desenvolveram diversas técnicas, fazendo com que os vinhos não fossem mais direcionados somente aos rituais da igreja, como também ao comércio.

Na Renascença, no século XVIII, período de paz na Europa, surgiu na França e Inglaterra uma nova classe social (a burguesia) que possuía dinheiro e bom gosto e começaram a demandar por vinhos de melhor qualidade.

A partir de 1854, o cientista francês Louis Pasteur, dedicou-se ao estudo de diferentes tipos de fermentação, onde decifrou fenômenos naturais responsáveis pela transformação da uva em vinho, fazendo com que a produção do vinho passasse do empirismo à ciência. Nesse período houve um avanço surpreendente onde começou o uso de garrafas de vidro e rolha de cortiça para armazenar o vinho e torná-lo mais durável.

A partir de 1870, surgiu a maior tragédia e mais devastadora da história do vinho, a praga filoxera (*Phylloxera vastatrix*) que é um pulgão que faz a vinha morrer de forma lenta. Esta praga chegou dos barcos vindos da América do Norte que levavam para à Europa mudas de variedades de plantas americanas. Praticamente toda a Europa foi atingida tendo suas vinhas destruídas. Após 40 anos de devastações, foi encontrada uma solução: o enxerto das vinhas em cepas americanas, as quais eram insensíveis ao pulgão.

Aos poucos a produção do vinho foi sendo retomada, aprimorada e difundida na Europa e em outros continentes, e começaram a serem adotadas legislações de rigoroso controle da produção e comercialização do vinho. Os grandes e famosos vinhedos conseguiram atender a crescente demanda por vinho, que se transformou em um fenômeno de moda no mundo inteiro, graças às ótimas colheitas do final do século XIX.

O século XX e XXI marca um novo panorama mundial de produção do vinho: de um lado a Europa “velho mundo”, que produz vinhos famosos e de alta qualidade; de outro lado o “novo mundo”, contendo países como a África do Sul, Argentina, Austrália, Brasil Chile, Estado Unidos e outros, que começam uma disputa buscando oferecer bons vinhos e preços acessíveis a todas as classes

sociais. (GRANDE LAROUSSE DO VINHO, 2012; SUZAMARA SANTOS, 2006; PATRICIO TAPIA, 2010).

### 2.1.2 O Vinho

O vinho é uma bebida alcoólica proveniente da fermentação alcoólica do sumo da uva (*Vitis vinifera*). O vinho é o resultado de uma transformação do açúcar da uva em álcool, onde ocorre a conversão da glicose e frutose em álcool etílico e gás carbônico. Através de diferentes processos de vinificação, podem-se obter os principais tipos de vinhos:

- **Vinho branco:** A uva branca passa por um processo de esmagamento muito delicado, onde as cascas e sementes são desprezadas. Os vinhos brancos são feitos praticamente do mosto (parte líquida da uva). Os vinhos brancos em geral são frescos, possuem cor dourada, sabor frutado, maior acidez que os tintos e seus aromas remetem a frutas verdes, minerais e herbáceos;
- **Vinho tinto:** Ao contrário do vinho branco, no vinho tinto as cascas das uvas tintas e sementes passam pelo processo de esmagamento, onde contribuem com a intensidade da cor e complexidade do vinho. Os vinhos tintos em geral possuem cor vermelho rubi intenso, taninos acentuados, sabor de frutas maduras e seu aroma nos remetem a especiarias;
- **Vinho rosé:** Sua vinificação leva o mosto em contato com a casca da uva por um tempo controlado, garantindo assim o tingimento e a sua coloração que varia de cor alaranjada a cor púrpura. A maioria dos vinhos rosés são refrescantes e possuem aromas muito frutados referentes a frutas vermelhas maduras;
- **Champanhe e espumantes:** São vinhos brancos que possuem gás carbônico obtido de uma segunda fermentação. Suas principais características são: grande presença de gás carbônico que formam as bolhas (perlage) e espuma, cor amarelo palha, aroma de frutas tropicais cítricas, é cremoso e fresco.

Na Figura 2.1 são mostrados os principais tipos de vinho, da esquerda para direita, o primeiro é um vinho branco; o segundo, tinto; terceiro, rosé; quarto, espumante.



**Figura 2.1 - Principais tipos de vinhos (Autor: Ayrton Gissoni)**

Diversos fatores influenciam na qualidade do vinho: o *terroir* (é considerado a carteira de identidade do vinho, representa o conjunto de clima, solo, relevo, altitude, exposição ao Sol e ambiente), as decisões tomadas pelo enólogo (profissional responsável por fazer o vinho), a cepa utilizada, o método de vinificação, o amadurecimento e envelhecimento (onde ocorre a guarda) do vinho. (GRANDE LAROUSSE DO VINHO, 2012; SUZAMARA SANTOS, 2006; PATRICIO TAPIA, 2010).

### **2.1.3 Principais Cepas**

No mundo do vinho existem diversos tipos de cepas (variedade da uva), como as *Vitis vinifera* que é uma espécie de vinha que dá origem ao conjunto de cepas mais populares do mundo. A França, responsável por ser um dos primeiros países pioneiros na produção do vinho e na abertura do mercado, é a líder na produção dos melhores vinhos e é na França que os outros países se espelham na

hora de produzir seus vinhos e competir no mercado. As principais cepas produzidas no mundo são:

- Tintas: Cabernet-Sauvignon, Cabernet Franc, Gamay, Merlot, Pinot Noir, Syrah (Shiraz), Tempranillo, Zinfandel, Malbec, Carmenère entre outras.
- Brancas: Chardonnay, Gewurztraminer, Muscat, Riesling, Sauvignon Blanc, Sémillon entre outras.

Na Figura 2.2 são mostradas as duas cepas de excelência para os vinhos tintos e brancos. A esquerda está a cepa considerada a “rainha dos tintos”: a Cabernet-Sauvignon. A direita está a cepa considerada a “rainha dos brancos”: Chardonnay. (Grande Larousse do Vinho, 2012)



Figura 2.2 - Cabernet-Sauvignon e Chardonnay (Fonte: o Grande Larousse do Vinho, 2012)

#### 2.1.4 Rótulos

O rótulo do vinho é a identidade do vinho e se localiza na frente da garrafa. O rótulo possui informações diversas sobre o vinho: vinhedo, uva, safra, teor alcoólico, método que foi produzido, conteúdo da garrafa, produtor, região e outras informações. Algumas informações adicionais estão localizadas no contrarrótulo, tais informações como, por exemplo, história do vinho e de sua vinícola, o importador e exportador, descrição do vinho (seus aromas e seu sabor) e etc. (GRANDE LAROUSSE DO VINHO, 2012)

A maioria dos países possuem leis rigorosas sobre a produção, conservação e distribuição do vinho. Na Figura 2.3 é mostrado um rótulo da região

de Bordeaux que contêm algumas informações regulamentadas pela legislação Francesa.

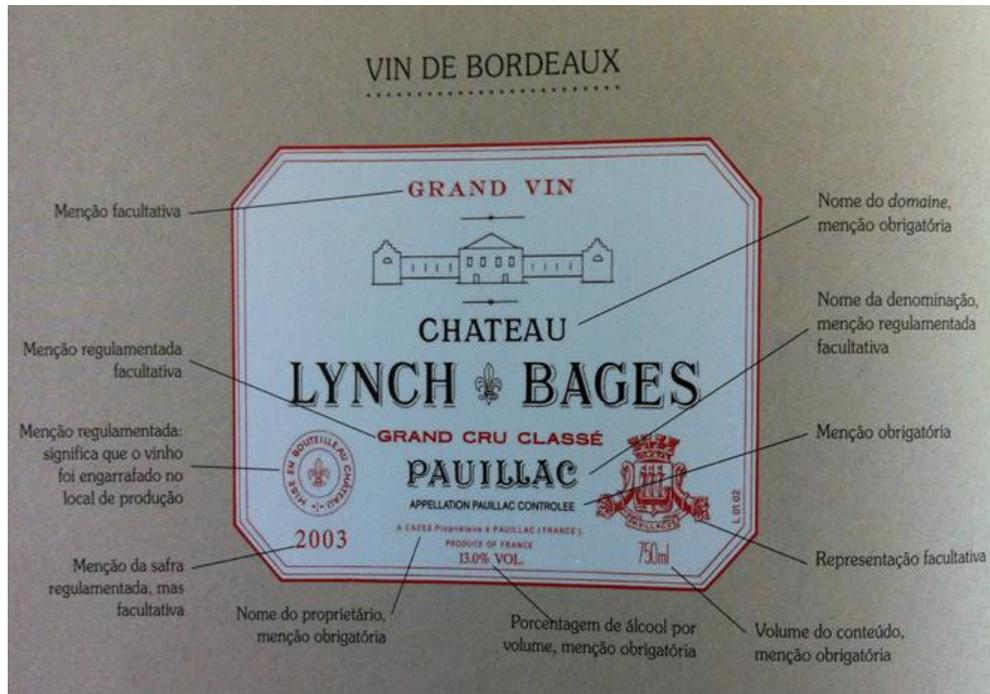


Figura 2.3 - Rótulo de um Vinho de Bordeaux (Fonte: o Grande Larousse do Vinho, 2012)

### 2.1.5 A Guarda do Vinho

Após a vinificação, amadurecimento (período em que o vinho fica armazenado em barricas de carvalho a fim de afinar e desenvolver suas qualidades) e engarrafamento, chega-se ao envelhecimento que é o período que o vinho descansa, mudando suas propriedades ao longo do tempo, para futuramente atingir seu apogeu, tornando seus aromas mais intensos e sabor mais rico.

É na etapa de envelhecimento do vinho que se faz sua guarda, deixando-o em condições ideais para descanso e evolução ao longo do tempo. A guarda do vinho deve ser feita em local apropriado como, por exemplo, uma adega climatizada. Na Figura 2.4 é mostrada uma adega climatizada de uso pessoal.



**Figura 2.4 - Adega Climatizada (Fonte: Site das Lojas Americanas, americanas.com.br)**

Ao guardar o vinho em uma adega, deve-se levar em conta as condições de armazenamento e tomar precauções quanto a temperatura, posição da garrafa, umidade, vibração, luminosidade e ventilação. (GRANDE LAROUSSE DO VINHO, 2012; SUZAMARA SANTOS, 2006; PATRICIO TAPIA, 2010).

### **2.1.6 A Importância da Temperatura**

Um dos principais fatores que mais influencia na qualidade do vinho é a temperatura. É a temperatura o elemento capaz de fazer maravilhosas garrafas de vinhos ficarem totalmente estragadas, pois uma pequena variação da temperatura muda todo o sabor, podendo-o mascarar, delatar, destruir ou santificar.

Ao fazer a guarda de um vinho deve-se manter a temperatura estável (evitando alterações bruscas de temperatura) em uma faixa que varia conforme o tipo de vinho, pois a guarda do vinho em um local quente demais acelera a maturação do vinho e um local frio demais impede sua evolução. A temperatura deve ser mantida:

- Para os tintos: entre 14 graus Celsius a 18 graus Celsius;
- Para os brancos: entre 8 graus Celsius a 12 graus Celsius;
- Para os Champanhes e espumantes: entre 6 graus Celsius a 8 graus Celsius.

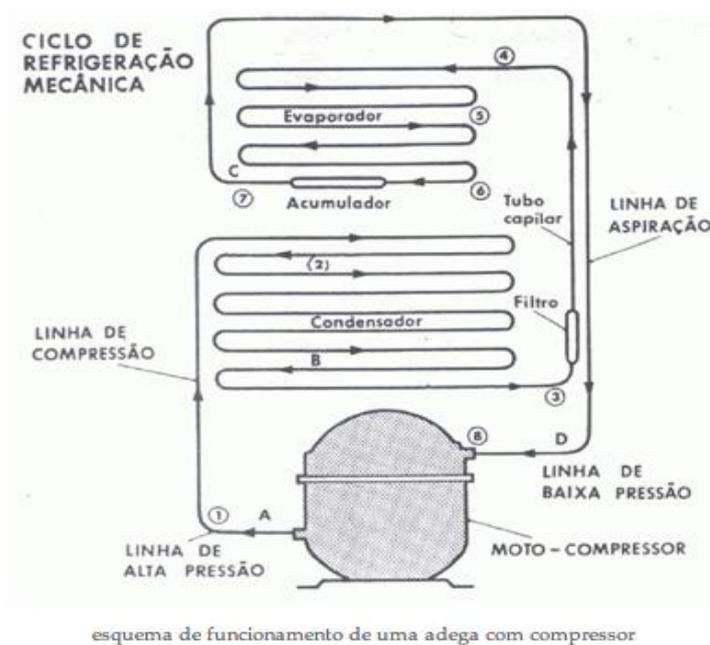
A temperatura também é muito importante na hora de servir o vinho, porque o vinho “se abre” permitindo aos componentes aromáticos volatilizar-se e

mostrar seus aromas (buquê) quando está na temperatura adequada ao seu estilo (tintos, brancos, rosés e espumantes). Quando a temperatura de serviço do vinho está alta, o vinho acaba desequilibrando e apresentando defeitos, pois o calor acentua a presença do álcool que mascara seus aromas. Quando a temperatura de serviço está baixa demais, o vinho não “se abre”, não mostra seus aromas e a acidez (se aliada a fruta torna o vinho refrescante e agradável) não se torna tão interessante. (GRANDE LAROUSSE DO VINHO, 2012; SUZAMARA SANTOS, 2006; PATRICIO TAPIA, 2010).

### **2.1.7 Modelo Atual de Adega Climatizada**

Atualmente existem dois principais tipos de adegas climatizadas para uso pessoal: o primeiro tipo utiliza um sistema de refrigeração mecânica através do uso de um compressor; o segundo tipo utiliza pastilhas termoelétricas para se fazer a climatização do vinho.

O sistema de refrigeração mecânica utilizando um compressor consegue refrigerar até uma temperatura de menos 20 graus Celsius, funcionando basicamente como um refrigerador. Entre suas vantagens está a capacidade de conseguir refrigerar uma adega contendo mais garrafas em menor tempo que o segundo método. Na Figura 2.5 é mostrado um esquema de funcionamento de uma adega utilizando o compressor.



**Figura 2.5 - Esquema de uma adega que utiliza um compressor (Fonte: <http://pordentrodaadega.com.br>)**

O sistema de climatização utilizando as pastilhas de Peltier das adegas termoelétricas de uso pessoal não tem a função de refrigerar (congelar) o vinho e sim apenas climatizá-lo. Entre suas vantagens está a capacidade de ocupar um espaço muito pequeno. Na Figura 2.6 é mostrada uma adega termoelétrica.



**Figura 2.6 - Adega Termoelétrica (fonte:<http://pordentrodaadega.com.br>)**

O sistema de climatização utilizando as pastilhas de Peltier das adegas termoelétricas de uso pessoal tem as seguintes vantagens sobre o seu concorrente:

- Ocupa um menor espaço e torna o produto mais leve;
- Não faz barulho;
- Não faz vibrações;
- Mais fácil implementação;
- Não polui;
- É eficiente para controle de temperatura;
- Não possui gás *fréon*;
- Seu custo é menor.

### 2.1.8 Limitações do Modelo Atual

Tanto o sistema de climatização das adegas que utilizam as pastilhas termoelétricas, conforme mostrado na Figura 2.7, quanto o que utiliza o compressor, não possuem diferentes compartimentos para climatizar os diferentes tipos de vinho e nem um sistema que faça o controle e cadastro dos vinhos, permitindo sua rápida localização e identificação.



**Figura 2.7 - Limites do modelo atual**  
(fonte: <http://www.colombo.com.br/produto/Eletrrodomesticos/Adega-16-Garrafas-ADE160-Cadence>)

## 2.2 Fenômenos de Transporte de Calor

### 2.2.1 A Transferência de Calor

Na física, a transferência de calor é um fenômeno que descreve a troca de calor entre corpos. Esse calor é um tipo de transferência de energia, onde um corpo cede (libera) calor e o outro recebe (absorve) este calor.

O fenômeno de transporte do calor consiste basicamente no transporte de energia térmica que se dá pela diferença de temperatura entre esses corpos. Essa diferença de temperatura (gradiente de temperatura) atua como a força motriz do sistema, pois na área da física que trata da transferência de calor, essa transferência ocorre sem que seja exercido o elemento trabalho (diferentemente do caso da termodinâmica).

A transferência de calor é classificada de três diferentes formas: condução, convecção e radiação (irradiação). Na Figura 2.8 é mostrado um exemplo dos três tipos de transferência de calor:

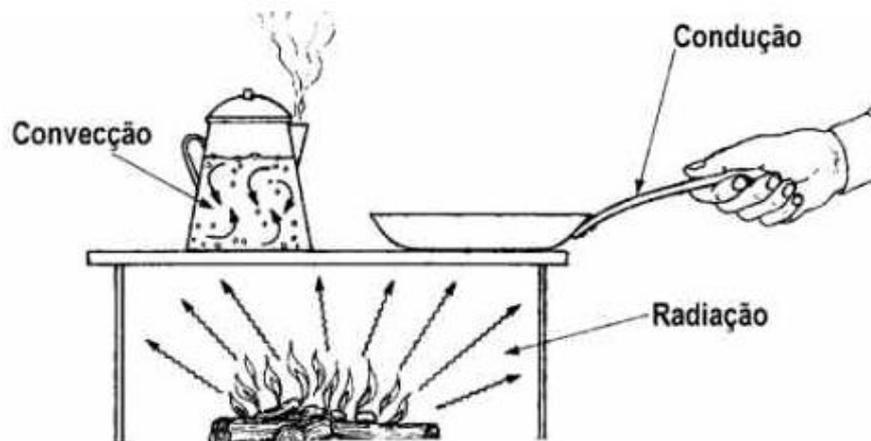


Figura 2.8 -Mecanismos de Transferência de Calor (Fonte: <http://fisica.ufpr.br/grimm/aposmeteo/cap2/cap2-9.html>)

Essas formas de transferência de calor raramente atuam sozinhas, elas podem ocorrer simultaneamente como no caso da atmosfera, porém na análise dos fenômenos que envolvem essas transferências de calor, podemos focar nossos cálculos em apenas em uma das formas predominantes, desprezando os efeitos das menores. (BENNETT, 2008; TIPLER, 2009).

### 2.2.1.1 Condução

A condução térmica é uma forma de transferência de calor que pode ocorrer entre sólidos, líquidos e gases, onde a energia é passada de partícula para partícula, átomos para átomos, moléculas para moléculas adjacentes dentro de uma mesma substância ou entre substâncias que estão em contato físico direto.

O fluxo de energia é dado pelo gradiente de temperatura (variação de temperatura) em nível microscópico. Ocorre sempre pela transferência de calor pelas partículas, átomos e moléculas que possuem maior energia cinética e interagem colidindo com seus vizinhos, que possuem menor quantidade de energia, transferindo a eles parte dessa energia cinética.

A capacidade com que diversos tipos de substâncias podem transferir calor varia consideravelmente, sendo que na maioria das vezes os sólidos são melhores condutores de calor que os líquidos, que por sua vez são melhores que os gases. Como por exemplo, a transferência de calor ocorre melhor entre os metais, pois estes são excelentes condutores de calor, enquanto o ar é um péssimo condutor de calor.

A Equação de Fourier é a lei básica utilizada no cálculo da condução do calor e é dada pela Equação 2.1:

$$Q_x = -k \cdot A \cdot \frac{\Delta T}{\Delta x} \quad (2.1)$$

Onde:

- $Q_x$  é a taxa de transferência de calor por condução direta na direção  $x$ , dado em Watts;
- $k$  é a constante de condutividade térmica, dada em  $W/(m \cdot K)$ .
- $A$  é a área da seção transversal na direção  $x$ , dada em  $m^2$ ;
- $\frac{\Delta T}{\Delta x}$  é o gradiente de temperatura na direção  $x$ , dado em Kelvin.

A equação de Fourier possui um sinal de negativo, pois vem da segunda lei da termodinâmica, onde a relação  $\frac{\Delta T}{\Delta x}$  é uma relação negativa visto que o gradiente de temperatura estabelece que o fluxo de transferência de energia ocorre das

regiões de temperaturas mais elevadas para as regiões de temperaturas menos elevadas. (BENNETT, 2008; TIPLER, 2009).

A constante de condutividade térmica ( $k$ ) depende de cada tipo de material e é utilizada como parâmetro dependendo de cada tipo de aplicação. Na Tabela 1 é mostrada a condutividade térmica dos materiais a 300K. Note que quanto maior o  $k$ , melhor condutor de calor o material. E quanto menor o  $k$ , pior é a condução de calor deste material.

**Tabela 1- Condutividade térmica dos materiais a 300K (Fonte: Bennett)**

Material	Condutibilidade Térmica (k)
Cobre (puro)	399
Ouro (puro)	317
Alumínio (puro)	237
Ferro (puro)	80,2
Aço Carbono (1%)	43
Aço Inoxidável (18/8)	15,1
Vidro	0,81
Plásticos	0,2 – 0,3
Água (líquido)	0,6

Fonte: BENNETT, 2008

### 2.2.1.2 Convecção

A convecção térmica é uma forma de transferência de calor que ocorre somente em líquidos e gases, onde o transporte é feito através de movimentações do próprio fluido. As movimentações ocorrem através das diferenças de massa específica (densidades) presentes no fluido.

A convecção é a transferência de calor que é responsável pelas grandes correntes no oceano, bem como a circulação do ar atmosférico. Simplificando em poucas palavras, a convecção ocorre quando, por exemplo, em um sistema de ar condicionado, o ar de cima quando refrigerado aumenta sua densidade e fica mais pesado que o ar quente (embaixo), então o ar frio desce e desloca empurrando para

cima (através do empuxo) o ar quente, que sobe e é refrigerado iniciando novamente o ciclo.

Existem dois tipos de convecção: a natural (explicada acima) e a forçada, a qual utiliza aspiradores e bombas para fazer o deslocamento do fluido. (BENNETT, 2008; TIPLER, 2009).

### **2.2.1.3 Radiação**

A radiação ou irradiação é uma forma de transferência de calor que ocorre através da propagação, na velocidade da luz, de ondas eletromagnéticas. Para que ocorra a radiação não é necessária a existência de matéria no meio de propagação, podendo esta ser o único tipo de transmissão de calor que pode ocorrer em um espaço vazio (a vácuo).

Todos os tipos de corpos emitem e absorvem a radiação eletromagnética. Quando um corpo aquece ele transfere calor na forma de radiação, como por exemplo, o Sol que aquece a Terra. Um corpo em equilíbrio térmico com sua vizinhança emite e absorve a radiação em uma mesma taxa.

Ondas de rádio e televisão, ondas de raios X e ondas de infravermelho são alguns exemplos de ondas eletromagnéticas, cujas diferenças entre elas são a sua frequência e seu comprimento de onda. (BENNETT, 2008; TIPLER, 2009).

## 2.2.2 Calorimetria

A calorimetria é a parte da física que estuda o calor como energia térmica em trânsito, em que tal forma de energia se dá pela diferença de temperatura entre os corpos de um determinado sistema em estudo.

Em um sistema isolado termicamente (dentro de um calorímetro), considere a existência de dois corpos distintos, A e B, com temperaturas diferentes entre si. Verifica-se a transmissão de energia térmica, ou seja, o calor, que flui do corpo com maior temperatura (grande quantidade de calor) para o corpo com menor temperatura. Quando a temperatura dos dois corpos se igualarem, a transferência de calor acaba e se atinge o equilíbrio térmico. Na Figura 2.9 é mostrado como ocorre a transferência de calor entre dois corpos até se atingir o equilíbrio térmico.

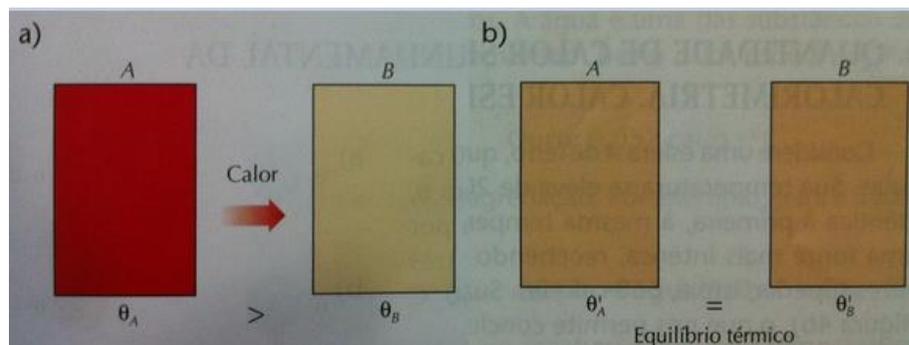


Figura 2.9 - Transmissão de Calor (Fonte: Ramalho, Nicolau, Toledo - 2006)

Na calorimetria, quando um corpo recebe calor, aumenta sua temperatura e sua quantidade de calor é positiva e quando um corpo cede calor, diminui sua temperatura e sua quantidade de calor é negativa. De acordo com o princípio das trocas de calor (Equação 2.2), a soma algébrica de todas as quantidades de calor  $Q$  de um sistema (calor recebido mais calor cedido) tem que se igualar. (TIPLER, 2009; RAMALHO, 2007).

$$\sum Q = 0 \quad (2.2)$$

### 2.2.2.1 Calor Sensível

Quando um corpo recebe calor, pode acontecer duas coisas: a mudança de estado físico e a mudança de temperatura. Quando ocorre a mudança de estado físico, o corpo recebe calor latente, e, quando ocorre a mudança de temperatura, o corpo recebe o calor sensível. (TIPLER, 2009; RAMALHO, 2007).

Para o calor sensível, a quantidade de calor recebida ou cedida pelo corpo é diretamente proporcional a sua diferença (variação) de temperatura e de sua massa. A Equação 2.3 é utilizada para determinar a quantidade de calor, esta equação é conhecida como equação fundamental da calorimetria:

$$Q_s = m \cdot c \cdot \Delta T \quad (2.3)$$

Onde:

- $Q_s$  é a quantidade de calor sensível, dado em  $J$  (Joules);
- $m$  é a massa do corpo, dada em  $g$  gramas;
- $c$  é o calor sensível que é um coeficiente de proporcionalidade dado pela característica do material, sua é  $cal/g \cdot ^\circ C$ ;
- $\Delta T$  é a diferença de temperatura em  $^\circ C$ .

### 2.2.2.2 Calor Latente

A quantidade de calor latente recebida ou cedida pelo corpo é diretamente proporcional a sua massa e a constante  $L$  que depende de cada material e de cada mudança de fase deste material. Nesse tipo de troca de calor, não ocorre mudança de temperatura, somente a mudança de estado físico. A Equação 2.4 é utilizada para determinar a quantidade de calor latente: (TIPLER, 2009; RAMALHO, 2007).

$$Q_L = m \cdot L \quad (2.4)$$

Onde:

- $Q_L$  é a quantidade de calor latente, dado em  $J$  (Joules);
- $m$  é a massa do corpo, dada em  $g$  gramas;
- $L$  é a constante de calor latente e sua unidade no é  $cal/g$ .

## 2.3 Fenômenos Termoelétricos

A termoelétrica estuda os fenômenos termoelétricos que associam o fluxo de calor com corrente elétrica com o objetivo de gerar energia elétrica ou térmica. Um dispositivo termoelétrico é utilizado para geração de energia, quando existe uma diferença de temperatura (gradiente de temperatura) entre seus lados frio e quente, este dispositivo cria uma corrente elétrica. Um dispositivo termoelétrico também é utilizado para geração de energia térmica, onde a direção do fluxo de aquecimento é determinada pelo sentido da passagem da corrente elétrica.

Um dispositivo termoelétrico pode ser utilizado para gerar energia elétrica, energia térmica, aquecer e esfriar objetos e ambientes, e medir temperatura. Eles são utilizados em diversas aplicações como, por exemplo: fontes alternativas de energia, conversão de energia, refrigeração, calefação, climatização, geração de energia elétrica para satélites, para missões espaciais, para bóias marítimas, para estações de telecomunicação e outras grandes infinidades de aplicações.

Os fenômenos termoelétricos têm baixa eficiência em metais e alta eficiência em materiais semicondutores como o silício e o germânio, pois sua condutividade térmica aumenta de acordo com o aumento da temperatura. Para aumentar a eficiência dos semicondutores ao trabalhar em altas temperaturas, se utiliza da técnica de dopagem desses materiais.

Os efeitos termoelétricos se dividem em três, são eles: Efeito de Seebeck, efeito de Peltier e efeito de Thomson. (COSTA 2010; SOUZA, 2011; MARQUES, 2011).

### 2.3.1 Efeito Seebeck

Em 1821, o físico russo-alemão, Thomas Johann Seebeck observou que ao submeter dois materiais distintos, condutores ou semicondutores, a temperaturas diferentes (gradiente de temperatura), ligados por meio de uma junta de medição (ou junta quente), ocorre o aparecimento da diferença de potencial entre a extremidade fria e a extremidade quente. Para um circuito fechado a diferença de potencial gera

uma corrente elétrica que depende do sentido do gradiente térmico em relação a junção. Na Figura 2.10 é mostrado o experimento realizado por Seebeck.

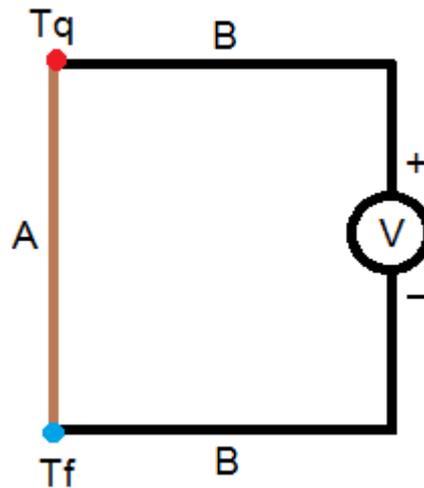


Figura 2.10 – Experiência de Seebeck (Desenhado por Ayrton Gissoni)

O estudo realizado pelo físico Seebeck possibilitou a criação do componente termopar, que é um sensor de temperatura amplamente utilizado nos variados processos de medição de temperatura. O termopar é constituído de dois metais puros ou ligas homogêneas, unidos por uma extremidade formando um circuito elétrico fechado, por onde fluirá uma corrente elétrica, para gerar uma força eletromotriz (FEM) que quando conectada a um instrumento de leitura consegue fazer a medição da temperatura. (COSTA 2010; SOUZA, 2011; MARQUES, 2011).

O coeficiente de Seebeck é conhecido como potência termoelétrica e depende dos materiais e das diferenças de temperatura entre a fonte quente e a fonte fria. Para um sistema onde o coeficiente de Seebeck é constante, podemos utilizar a seguinte Equação 2.5:

$$V = (S_b - S_a) \times (T_q - T_f) \quad (2.5)$$

Onde:

- $V$  é a tensão elétrica;
- $S_a$  e  $S_b$  são os coeficientes de Seebeck para os materiais A e B;
- $T_q$  e  $T_f$  são as temperaturas da fonte quente e da fonte fria respectivamente.

### 2.3.2 Efeito de Peltier

Após 13 anos, em 1834, o físico francês Jean Charles Athanase Peltier observou o efeito reverso ao efeito de Seebeck que recebeu o nome de efeito de Peltier em sua homenagem.

O efeito de Peltier é um fenômeno termoelétrico que produz um gradiente de temperatura quando se aplica uma tensão em um circuito elétrico fechado que gera uma corrente que atravessa um corpo formado por uma junção de condutores ou semicondutores distintos, onde dependendo do sentido de passagem da corrente elétrica, uma das junções absorve o calor e a outra o libera. Na Figura 2.11 é mostrado o experimento realizado por Peltier.

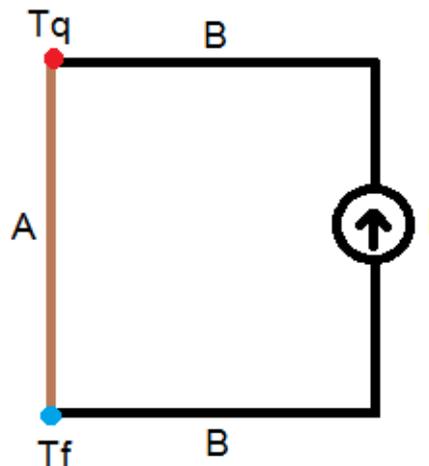


Figura 2.11 - Experiência de Peltier (Desenhado por Ayrton Gissoni)

As junções consistem em materiais semicondutores extrínsecos do tipo P e N. Através do método de dopagem altera-se a quantidade de elétrons da camada de valência desses semicondutores, o tipo P possui falta de elétrons (elemento positivo), enquanto o tipo N possui excesso de elétrons (elemento negativo). Na Figura 2.12 é mostrada uma representação esquemática da célula de Peltier.

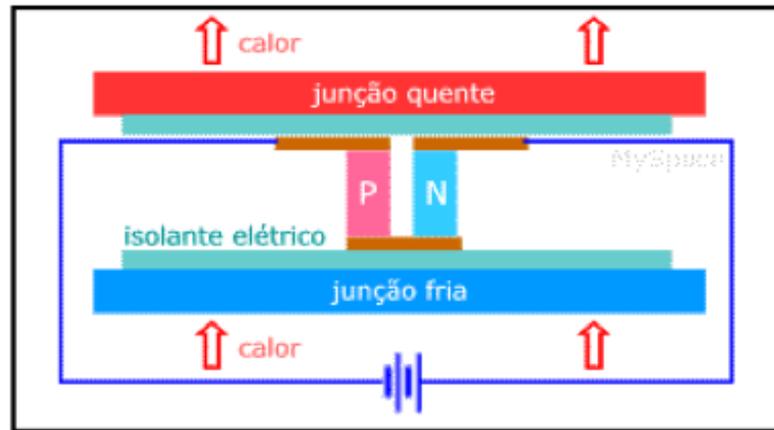


Figura 2.12 – Representação Esquemática da Célula de Peltier (Fonte: [http://www.mspsc.eng.br/elettrn/peltier\\_110.shtml](http://www.mspsc.eng.br/elettrn/peltier_110.shtml))

Quando submetido a uma tensão, dependendo do sentido da corrente elétrica gerada, um dos lados aquece enquanto o outro lado esfria. Para a junção quente, a corrente elétrica flui do lado positivo para o lado negativo e para a junção fria, a corrente elétrica flui do lado negativo para o positivo.

A transferência do fluxo de calor para os elementos P e N se dá da seguinte forma: O elemento positivo P absorve calor elevando seu nível de energia que o faz migrar para o lado quente que possui uma menor quantidade de energia. Através de ventiladores, coolers e dissipadores, retira-se calor desse elemento que retorna ao lado P reiniciando o ciclo. (COSTA 2010; SOUZA, 2011; MARQUES, 2011).

De acordo com a Equação 2.6, o calor de Peltier gerado pelas junções pode ser calculado da seguinte forma:

$$Q = (\pi_A - \pi_B) \times I \quad (2.6)$$

Onde:

- $Q$  é a taxa de transferência de calor de Peltier absorvida pela junção fria por unidade de tempo;
- $\pi_A$  e  $\pi_B$  são respectivamente os coeficientes de Peltier dos materiais A e B;
- $I$  é a corrente aplicada ao circuito, dada em Amperes (A).

### 2.3.3 Efeito Thomson

Mais tarde, em 1851, o físico-matemático e engenheiro britânico nascido na Irlanda, William Thomson explicou como se relacionavam os efeitos de Seebeck e Peltier, mostrando que um metal (condutor ou semiconductor) submetido a uma corrente elétrica com um gradiente de temperatura (diferença de temperatura) entre suas faces, poderia produzir frio ou calor apenas dependendo do sentido com que a corrente elétrica passasse por esse metal.

Na Figura 2.13 é mostrado os esquemas dos efeitos de Seebeck, Peltier e Thomson, onde o calor produzido é representado por  $Q$  e sua variação por  $\Delta Q$ ;  $T$ ,  $T_1$  e  $T_2$  representam as temperaturas dos materiais e  $\Delta T$  sua variação;  $\Delta V$  representa a diferença de potencial e  $I$  representa a corrente elétrica. (COSTA 2010; SOUZA, 2011; MARQUES, 2011)

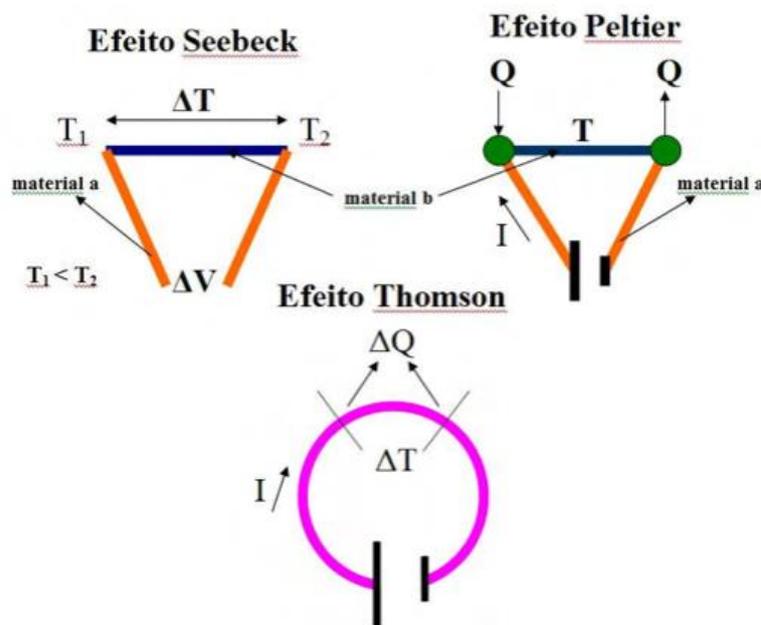


Figura 2.13 – Esquema dos efeitos de Seebeck, Peltier e Thomson (Fonte: Dissertação de Mestrado, INPE, Vânia Aparecida da Costa)

Thomson formulou a equação conhecida como equação fundamental da termoeletricidade, que mostra a relação entre a quantidade de energia absorvida ou fornecida por um metal e a respectiva corrente elétrica que passa por esse metal. A Equação 2.7 descrita por Thomson é:

$$Q = \rho \cdot J^2 - \mu \cdot J \cdot \frac{\partial T}{\partial x} \quad (2.7)$$

Onde:

- $Q$  é o calor absorvido ou fornecido por um condutor metálico, dado em Watts;
- $\rho$  é a resistividade do material, dada em  $\Omega \cdot m$ ;
- $\frac{\partial T}{\partial x}$  é o gradiente de temperatura ao longo do metal, dado em Celsius;
- $J$  é a densidade da corrente elétrica;
- $\mu$  é o coeficiente de Thomson que depende do tipo de metal.

## 2.4 Célula de Peltier

A célula de Peltier ou módulo de Peltier é uma pastilha termoelétrica que utiliza o efeito de Peltier para aplicações de resfriamento ou aquecimento nos setores de eletrônica, automotivo, industrial e militar, como: resfriamento de microprocessadores, mini geladeiras e pequenos refrigeradores, aquecedor e resfriador de assentos (na indústria automobilística), *coolers* eletrônicos, desumidificador, adegas climatizadas de uso pessoal e outras aplicações.

As principais vantagens da utilização da célula de Peltier são: ausência de barulho e vibração, alta precisão para o controle da temperatura, alta durabilidade (não necessita de uma manutenção frequente), ausência de peças móveis, não possui gás *fréon*, seu tamanho é reduzido e ocupa pouco espaço, não polui o meio ambiente, possui grande aplicabilidade e seu custo é reduzido se comparado ao seu principal concorrente tecnológico (compressor). (SITE DO FABRICANTE [HTTP://WWW.PELTIER.COM.BR](http://www.peltier.com.br); SOUZA, 2011; MARQUES, 2011)

### 2.4.1 Funcionamento da Célula de Peltier

A célula de Peltier é formada por semicondutores (telureto de bismuto) do tipo-p e tipo-n agrupado em pares e soldados em duas placas de cerâmica,

formando uma espécie de “sanduíche”, conforme mostrado na Figura 2.14, ligadas termicamente em paralelo e eletricamente em série.

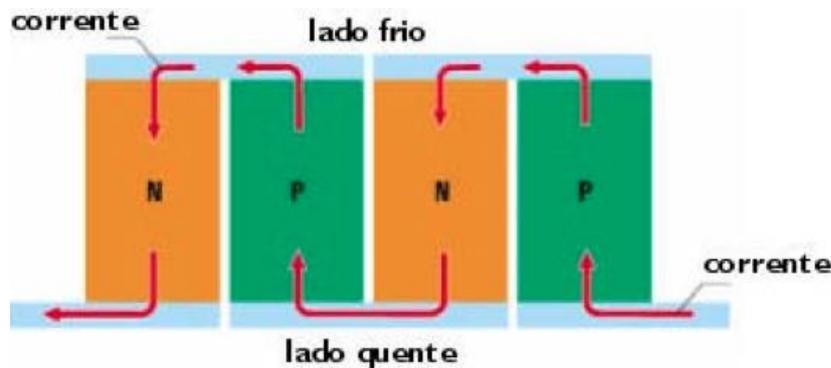


Figura 2.14 - Esquema da Célula de Peltier (Fonte: Site do Fabricante, <http://www.peltier.com.br/faqs.htm>)

As superfícies fria e quente formam as faces da célula de Peltier que são feitas de cerâmica, pois a cerâmica possui um bom isolamento elétrico e alta condutibilidade térmica. Essas superfícies de cerâmica são de extrema importância para que haja condução de calor entre os elementos (semicondutores) os quais elas se conectam.

De acordo com o efeito de Peltier, o resfriamento e aquecimento ocorrem quando aplicamos uma diferença de potencial na célula de Peltier, que gera uma corrente elétrica responsável por gerar um gradiente de temperatura entre as faces. É por causa desse gradiente de temperatura entre as faces da célula, que ocorre o fluxo de calor, onde o resfriamento de Peltier fará o calor mover de um lado para o outro das faces. Na Figura 2.15 é mostrada uma visão interna da célula de Peltier, bem como do efeito de Peltier. (SITE DO FABRICANTE [HTTP://WWW.PELTIER.COM.BR](http://www.peltier.com.br); SOUZA, 2011; MARQUES, 2011).

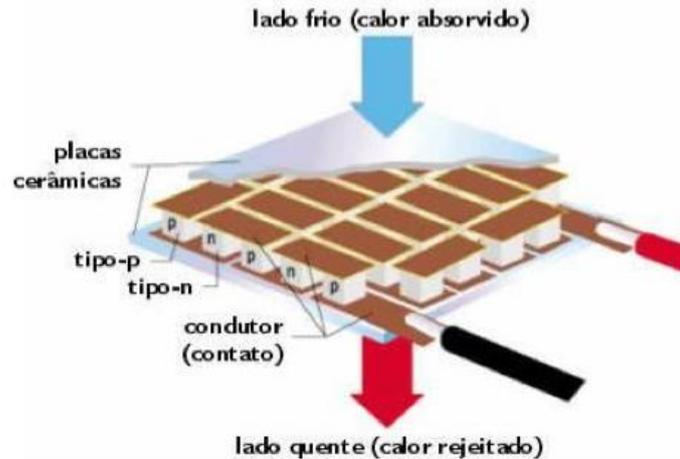


Figura 2.15 - Visão interna da Célula de Peltier e do efeito de Peltier (Fonte: Site do Fabricante, <http://www.peltier.com.br/faqs.htm>)

## 2.4.2 Cuidados na Montagem do Módulo de Peltier

O uso de dissipadores de calor e ventiladores (ou *coolers*) são de extrema importância para evitar o superaquecimento e conseqüentemente a destruição da célula de Peltier, uma vez que as pastilhas tem a função de transferirem calor de um lado para o outro de suas faces, tornando necessário o uso do dissipador para coletar calor (em modo de aquecimento) ou para dissipar calor (em modo de resfriamento).

Conforme na Figura 2.16 é mostrado um esquema de montagem do módulo de Peltier utilizando dissipadores e ventiladores, de modo a evitar a degradação da pastilha pelo superaquecimento. Para aumentar a transferência de calor, é recomendado o uso de pasta térmica entre a pastilha e o dissipador.

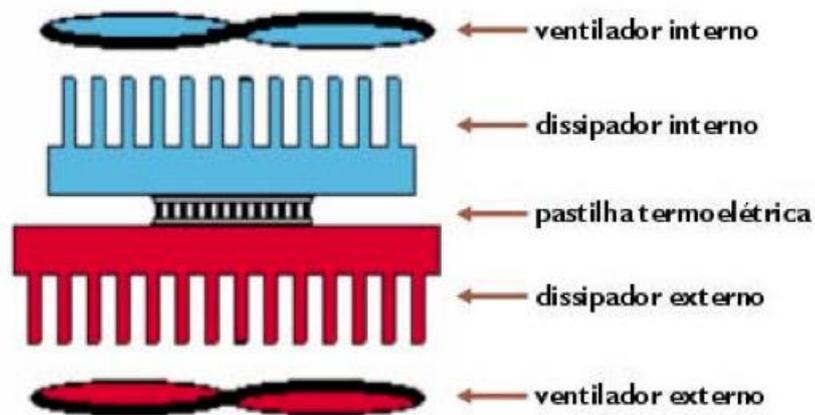


Figura 2.16 - Esquema de Montagem da Célula de Peltier (Fonte: Site do Fabricante, <http://www.peltier.com.br/faqs.htm>)

Outro cuidado que deve ser tomado ao escolher a célula de Peltier é observar o  $Q_{max}$ , que é o limite de quantidade de calor que a pastilha pode transferir ao aplicar a máxima diferença de potencial  $V_{max}$  e a máxima corrente  $I_{max}$  produzindo a máxima diferença de temperatura  $dT_{max}$  entre as faces da pastilha. (SITE DO FABRICANTE [HTTP://WWW.PELTIER.COM.BR](http://WWW.PELTIER.COM.BR); SOUZA, 2011; MARQUES, 2011).

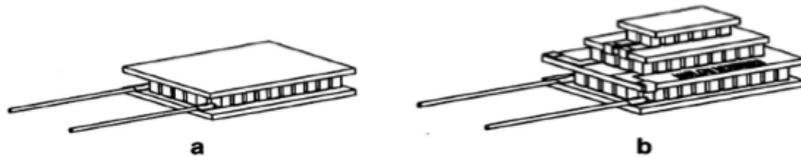
No Anexo A são mostrados os modelos de células de Peltier do fabricante Hebeiltd disponíveis no mercado.

### 2.4.3 Estágios do Módulo de Peltier

Atualmente a célula de Peltier transfere um calor máximo de 250W utilizando um único módulo (módulo de estágio simples). Para aumentar a transferência de calor é utilizado o módulo multi-estágios, onde são utilizadas várias pastilhas ao mesmo tempo empilhadas umas sobre as outras, de forma que a parte fria de uma fique voltada para a parte quente da seguinte.

Todavia, quando o gradiente de temperatura entre as fases fria e quente da pastilha termoelétrica não ultrapassar 60°C de diferença, o módulo de estágio simples é o mais recomendado, porém é aconselhado utilizar o módulo multi-estágios quando se quer obter uma diferença entre essas faces for maior que 60°C. (SITE DO FABRICANTE [HTTP://WWW.PELTIER.COM.BR](http://WWW.PELTIER.COM.BR); SOUZA, 2011).

Na Figura 2.17-A é mostrado o módulo de estágio simples e na Figura 2.17-B é mostrado o módulo múlti-estágios.



**Figura 2.17 - Módulos de Peltier (Fonte: Monografia: Condicionador de ar para Cabine de Caminhão, Utilizando Células de Peltier, Daniel de Souza, 2011)**

#### **2.4.4 Dimensionamento da Célula de Peltier por Carga Térmica**

O dimensionamento da célula de Peltier adequada para o projeto tem como base a definição das reais necessidades de resfriamento, isto inclui a correta determinação da carga térmica do sistema e da sua quantidade de calor a ser drenada do sistema.

Os tipos da carga térmica podem ser: ativa, passiva ou uma combinação das duas. A carga ativa está associada ao calor dissipado pela aplicação sendo resfriada que geralmente equivale à potência de entrada do sistema. As cargas térmicas passivas são cargas de radiação, convecção ou condução.

Com base no site oficial de Peltier, as técnicas para se calcular as cargas térmicas ativas e passivas só são aplicáveis à aplicações estáveis, sendo que para cargas térmicas transitórias que envolvem fatores mais complexos tem-se a necessidade de uma análise mais detalhada do projeto. Nem sempre todas as fórmulas para cálculo de cargas são necessárias para se desenvolver um projeto, pois de acordo com o fabricante, para sistemas mais simples é até mais fácil experimentar com algumas pastilhas para ver qual oferece o melhor desempenho. (SITE DO FABRICANTE [HTTP://WWW.PELTIER.COM.BR](http://www.peltier.com.br)).

##### **2.4.4.1 Carga Ativa**

A Equação 2.8 é a equação geral para a dissipação de potência de uma carga ativa:

$$Q_{ativo} = V \cdot I \quad (2.8)$$

Onde:

- $Q_{ativo}$  é a carga térmica ativa, dada em watts ( $W$ );
- $V$  é voltagem aplicada ao sistema resfriado, dado em volts ( $V$ );
- $I$  é a corrente da aplicação, dada em ampères ( $A$ ).

#### 2.4.4.2 Carga Passiva

A carga passiva de Radiação é considerada desprezível para um sistema que é operado em um ambiente gasoso, pois a magnitude das outras cargas passivas tende a ser bem maior. As cargas de radiação são importantes e levadas em conta para o cálculo de dimensionamento, em sistemas de altas diferenças de temperaturas, a vácuo e que contém pequenas cargas ativas. A Equação 2.9 é a equação fundamental para cargas térmicas de radiação:

$$Q_{rad} = F \cdot e \cdot S \cdot A \cdot (T_{amb}^4 - T_c^4) \quad (2.9)$$

Onde:

- $Q_{rad}$  é a carga térmica de radiação, dada em watts;
- $F$  é fator de forma (para o pior caso = 1);
- $e$  é a emissividade (para o pior caso = 1);
- $S$  é a constante de Stefan-Boltzman ( $5.667 \times 10^{-8} w/m^2K^4$ );
- $A$  é a área resfriada em  $m^2$ ;
- $T_{amb}$  é a temperatura ambiente, dada em Kelvin;
- $T_c$  é a temperatura do lado frio, dada em Kelvin .

A carga passiva de convecção para as células de Peltier são normalmente o resultado de convecção natural ou forçada através de uma ventoinha. A Equação 2.10 é a equação fundamental para o cálculo da carga de convecção é:

$$Q_{conv} = h \cdot A \cdot (T_{ar} - T_f) \quad \text{Equação (2.10)}$$

Onde:

- $Q_{conv}$  é a carga térmica de convecção em watts;
- $h$  é o coeficiente de transferência de calor convectivo ( $w/m^2°C$ ), o valor para uma placa horizontal a uma pressão de 1 atm é de  $21.7 w/m^2°C$ ;
- $A$  é a área exposta em  $m^2$ ;
- $T_{ar}$  é a temperatura do ar ambiente em  $°C$ ;
- $T_f$  é a temperatura da superfície fria em  $°C$ .

A carga passiva de condução ocorre pelo contato direto de moléculas de uma região com temperatura elevada para uma região de baixa temperatura. A Equação 2.11 é a equação fundamental para dimensionar a carga de condução é:

$$Q_{cond} = \frac{k.A}{L} . \Delta T \quad (2.11)$$

Onde:

- $Q_{cond}$  é a carga térmica condutiva em watts;
- $k$  é a condutividade térmica do material ( $w/m°C$ );
- $A$  é a área de seção transversal material em  $m^2$ ;
- $L$  é o comprimento do trajeto de calor em  $m$ ;
- $\Delta T$  é a diferença de temperatura no trajeto de calor em  $°C$ .

A Tabela 2 mostra, segundo o fabricante Peltier, os valores para condutividade térmica:

**Tabela 2 - Condutibilidade Térmica**

Material	Condutividade Térmica $w/m°C$
Alumínio	250
Cobre	386
Ouro	315
Isopor	0,031

<http://www.peltier.com.br>

A carga passiva combinada por convecção e condução pode ser calculada pela seguinte Equação 2.12:

$$Q_{passivo} = \frac{A \cdot \Delta T}{\frac{x}{k} + \frac{1}{h}} \quad (2.12)$$

Onde:

- $Q_{passivo}$  é a carga térmica passiva em watts;
- $A$  é a área total exposta (externa) do recipiente em  $m^2$ ;
- $x$  é a grossura da isolação em  $m$ ;
- $k$  é a condutividade térmica da isolação ( $w/m^\circ C$ );
- $h$  é o coeficiente de transferência de calor convectivo dado em ( $w/m^2^\circ C$ );
- $\Delta T$  é a diferença de temperatura em  $^\circ C$ .

A Tabela 3 mostra, segundo o fabricante Peltier, os valores para o coeficiente de transferência de calor convectivo:

**Tabela 3 - Coeficiente de Transferência de Calor Convectivo**

Processo	Coeficiente de calor convectivo ( $w/m^2^\circ C$ )
Convecção Natural	2 - 25
Convecção Força	25 - 250

<http://www.peltier.com.br>

## **CAPÍTULO 3 DESENVOLVIMENTO DO PROTÓTIPO DA ADEGA CLIMATIZADA**

Este capítulo apresenta a aplicação dos conceitos teóricos, tratados no capítulo 2, envolvidos na construção do protótipo da adega climatizada. Será apresentado o modelo de adega climatizada, o desenvolvimento do protótipo, mostrando descrição da construção e a descrição dos materiais utilizados.

### **3.1 Modelo de Adega Climatizado Proposto**

O modelo proposto de adega climatizada termoelétrica para uso pessoal melhorará a eficiência na hora de climatizar e localizar cada um dos tipos de vinhos, empregando um novo modelo de funcionamento que possui três compartimentos para a guarda dos diferentes tipos de vinhos: tinto, branco e espumante, onde será feita a utilização de um computador que através de uma aplicação fará o controle de temperatura de cada um dos compartimentos, como também um sistema de cadastramento e identificação de cada um dos vinhos, para que posteriormente o usuário possa fazer uma localização de forma rápida do vinho desejado.

A adega climatizada contém três compartimentos isolados termicamente entre si, o primeiro compartimento (de cima para baixo) será utilizado para a guarda dos vinhos tintos e sua temperatura é mantida entre 14 graus Celsius a 18 graus Celsius. O segundo compartimento (o compartimento do meio) será utilizado para a guarda dos vinhos brancos (podendo também fazer a guarda dos rosés) e sua temperatura é mantida entre 8 graus Celsius a 12 graus Celsius. O terceiro compartimento (localizado na parte mais baixa) será utilizado para a guarda dos vinhos espumantes e sua temperatura é mantida entre 6 graus Celsius a 8 graus Celsius.

A adega climatizada possui uma aplicação (que será rodada em um computador) responsável pelo controle de temperatura e pelo sistema de cadastramento e localização dos vinhos. Essa aplicação será rodada em um computador pessoal (*notebook*), pois se trata de um protótipo (baixo custo), mas

para comercialização (produto final) o ideal é utilizar um computador *touch screen* (como um *tablet*) colocando-o acoplado a estrutura física da adega.

A aplicação faz o cadastramento e localização utilizando para isso uma interface feita em Java onde será colocado e visualizado os dados sobre os vinhos existentes na adega, como também os vinhos já consumidos. Para isso a aplicação conta com um banco de dados (MySQL) que irá receber, tratar, armazenar e exibir informações sobre cada um dos vinhos selecionados.

O protótipo contém três compartimentos para armazenar os vinhos tintos, brancos e espumantes. Como se trata de um protótipo (pequena escala), cada compartimento armazena somente 2 garrafas, possuindo assim 6 garrafas na adega climatizada. O *software* simula a realidade do protótipo, possuindo a mesma quantidade de garrafas para cada compartimento.

A mesma aplicação (mesmo *software* de cadastramento) faz a climatização da adega, onde foram utilizados sensores de temperatura que verificam a temperatura de cada um dos compartimentos e transferem esses dados ao microcontrolador Arduino. É através da aplicação que é selecionada a temperatura de cada um dos compartimentos.

O microcontrolador Arduino é o elemento responsável por fazer o monitoramento e controle da temperatura de cada um dos compartimentos e levar esses dados ao computador que os exibi através da aplicação. Na Figura 3.1 é mostrado uma visão geral da adega climatizada proposta.

## ADEGA CLIMATIZADA PARA VINHOS UTILIZANDO A CÉLULA DE PELTIER

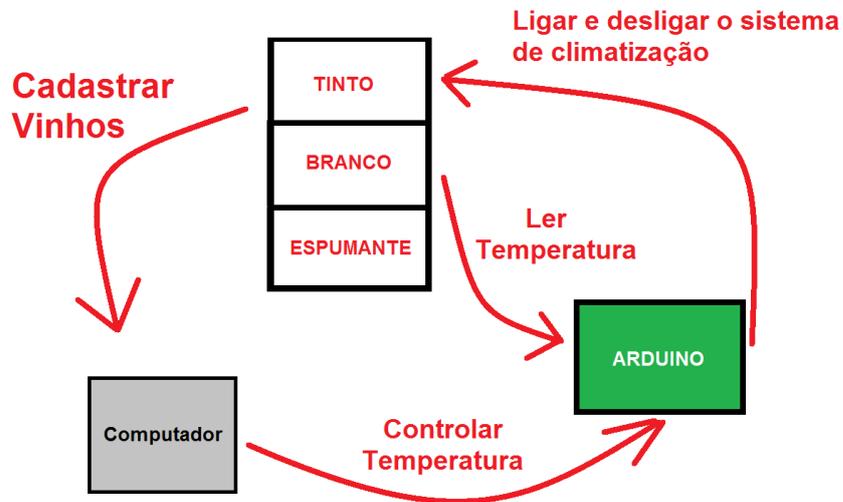


Figura 3.1 - Visão Geral da Adega Climatizada (Autor: Ayrton Gissoni)

## 3.2 Descrição dos Materiais Utilizados

### 3.2.1 Arduino

Para este projeto foi escolhido o microcontrolador Arduino Mega ATmega 1280 que é uma plataforma de computação física que possui um microcontrolador de placa única com suporte embutido de entrada e saída.

O Arduino surgiu na Itália, com o objetivo de criar um dispositivo simples e prático para fazer controle de entrada e saída de dados. O Arduino é uma plataforma de desenvolvimento que possui *software* e *hardware open source*, ou seja, de distribuição livre, onde qualquer pessoa interessada poderá basear-se nos códigos e nas placas podendo assim fazer a sua própria placa. Uma característica do Arduino é ser multi-plataforma, ou seja, roda em Windows, Mac e Linux.

Utilizando o Arduino é possível fazer diversas aplicações interagindo com o ambiente através da criação de programas que manipulem suas entradas e saídas, convertendo medidas do mundo real em sinais elétricos.

Utilizando a entrada e saída do Arduino é possível conectar diversos sensores como o sensor de temperatura, luminosidade, umidade, pressão, distância e outros dispositivos como os *displays*, LED, interruptores, motores e placas extensíveis como GSM, 3G, *Ethernet*, *Bluetooth*, relés entre outros.

O Arduino Mega possui 54 entradas/saídas digitais, das quais 14 podem ser utilizadas como saídas PWM, 16 entradas analógicas, 4 portas seriais (UART). O Arduino Mega utiliza um oscilador de 16MHz, possui conector de força, conector ICSP, botão de *reset* e uma interface de comunicação USB.

No Anexo B é mostrado o *datasheet* do Arduino ATmega 1280. Na Tabela 4 é mostrada a especificação do Arduino Mega ATmega 1280.

**Tabela 4 - Especificações Arduino Mega ATmega1280**

Microcontrolador	ATmega1280
Tensão de Alimentação	5V
Tensão de Entrada (recomendado)	5-12V
Tensão de Entrada (limites)	6-20V
Portas Digitais I/O	54
Portas Analógicas de Entrada	16
Corrente DC por pino	40mA
Corrente DC por pino 3.3V	50mA
Memória Flash	128KB
SRAM	8KB
EEPROM	4KB
Clock	16MHz

Fonte: [www.Arduino.cc/](http://www.Arduino.cc/)

Na Figura 3.2 é mostrada a placa Arduino Mega ATmega 1280.

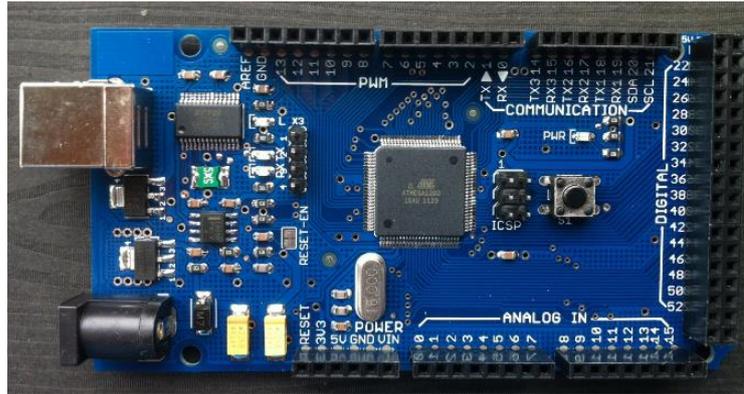
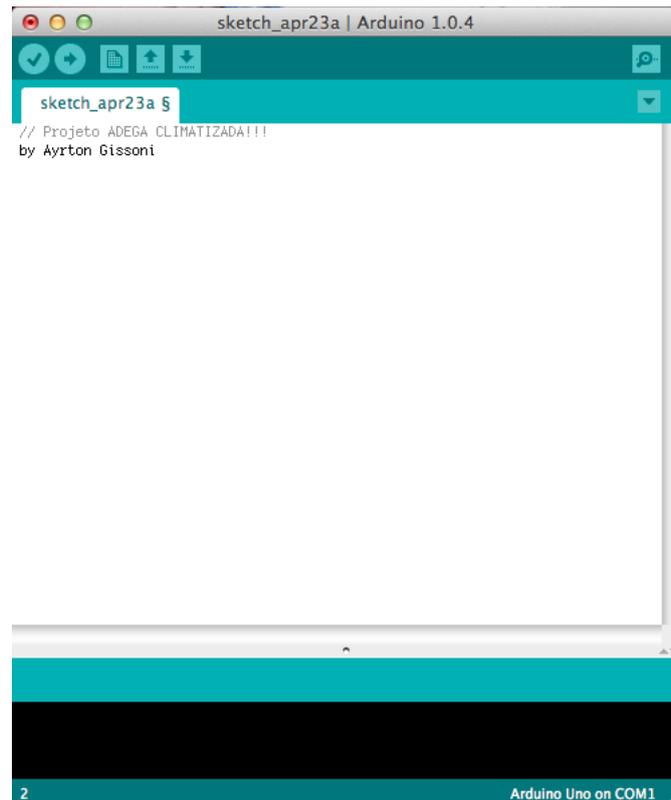


Figura 3.2 – Arduino Mega ATmega1280 (Autor: Ayrton Gissoni)

Para a programação na placa Arduino é necessário utilizar a IDE desenvolvida em *software* aberto, por meio da IDE podemos programar, fazer o *debug* do código e com o código já compilado é possível fazer o *upload* e gravá-lo na placa.

O ambiente de desenvolvimento IDE do Arduino, utiliza a linguagem de programação Wiring, que é um tipo de linguagem variante do C e do C++. O ambiente de desenvolvimento já possui algumas bibliotecas específicas, é possível também adicionar outras bibliotecas ou até mesmo criá-las. Na Figura 3.3 é mostrado o ambiente de desenvolvimento para o Arduino.



**Figura 3.3 - Ambiente de Desenvolvimento do Arduino (Autor: Ayrton Gissoni)**

A escolha do Arduino se deve pela facilidade com que o Arduino pode ser conectado com outros dispositivos e sensores, por ele ser multi-plataforma, por possuir uma linguagem simples e acessível, por ser de fácil aprendizado, pois possui uma quantidade enorme de material de estudo, indo de livros, vídeo aulas a fóruns e por ser uma placa de alta disponibilidade no mercado.

### **3.2.2 Estrutura da Adega**

Para a elaboração e dimensionamento da adega, foi considerado uma garrafa de vinho do modelo *Bordeaux* 750ml que possui uma altura de 31 cm e um diâmetro da base de 7 cm. O protótipo é feito de madeira e isopor e foi dimensionado de acordo com a Figura 3.4.

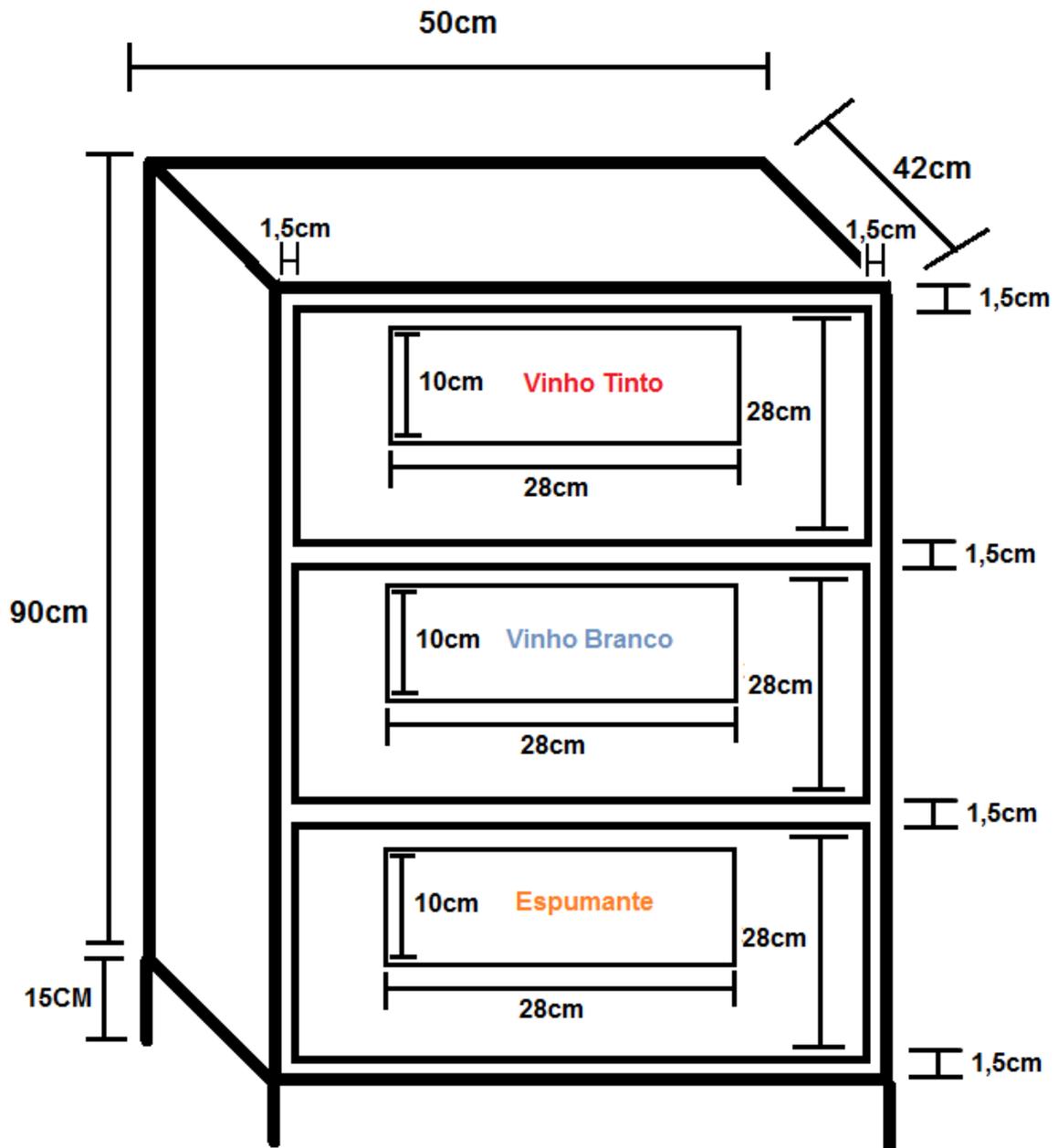


Figura 3.4 - Estrutura da Adega (Autor: Ayrton Gissoni)

As dimensões internas de cada compartimento sofreram alteração devido ao aumento do isolamento feito com o material isolante isopor e a diminuição do volume interno do protótipo a fim de se obter temperaturas mais baixas e melhorar a eficiência. Na Figura 3.5 é mostrado como é a estrutura do protótipo.



Figura 3.5 - Estrutura do Protótipo (Autor: Ayrton Gissoni)

### 3.2.3 Sensor de Temperatura

Para esse projeto foi escolhido como sensor de temperatura o sensor LM35. O sensor de temperatura LM35 é fabricado pela empresa National Semiconductor e é um sensor de alta precisão, pois consegue fornecer com boa exatidão dos valores de temperatura em graus Celsius com variações de  $\frac{1}{4}^{\circ}\text{C}$  a  $\frac{3}{4}^{\circ}\text{C}$  dentro da faixa de temperatura de  $-55^{\circ}\text{C}$  à  $150^{\circ}\text{C}$ .

O sensor LM35 possui uma grande vantagem em relação aos outros sensores encontrados no mercado, pois este sensor além de não precisar ser calibrado, apresenta uma saída linear de tensão de 10mV para cada grau Celsius, dispensando assim uma conversão de escala de Kelvin para Celsius.

O sensor LM35 possui diferentes tipos de encapsulamentos, sendo adotado para esse projeto o encapsulamento TO-92, que se parece com um transistor, é alimentado por uma tensão de 4-20VDC e GND que necessita de uma corrente de 60 $\mu\text{A}$  e seu auto aquecimento ao ar livre é de aproximadamente  $0,1^{\circ}\text{C}$ .

No Anexo C é mostrado o *datasheet* do sensor LM35. Suas principais características:

- É calibrado diretamente em graus Celsius (centígrados);

- Linear fator de escala de + 10,0 mV / °C;
- 0,5 °C precisão (quando a temperatura é superior a 25 °C);
- Faixa de temperatura de -55 °C a 150 °C;
- Indicado para aplicações remotas;
- Baixo custo;
- Opera de 4 a 20 Volts;
- Necessita de uma corrente de 60µA;
- Baixo auto-aquecimento, 0,1 °C (no ar);
- Impedância de saída baixa, de 0,1W para 1mA de carga.

Na Figura 3.6 é mostrado o diagrama de conexões do sensor de temperatura LM35.

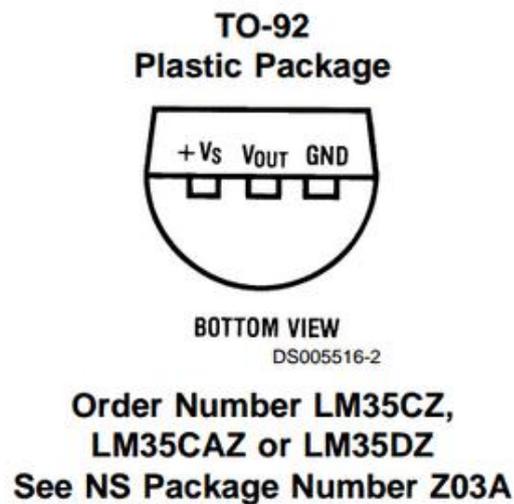


Figura 3.6 - Datasheet LM35 (Fonte: National Semiconductor)

### 3.2.4 Módulo de Relés

Para esse projeto foi utilizado um módulo de relés, que é uma placa extensível que pode ser conectada ao Arduino e é utilizada para fazer a função de ligar e desligar as Células de Peltier. O módulo de relés contém 4 canais de relés e suas principais características são:

- Corrente dos contatos de 10A em 250V AC ou 30V DC;
- Tensão de alimentação de 5V;

- Entradas opto isoladas;
- Entradas compatíveis com nível TTL (5V);
- Baixa corrente de entrada para chaveamento, podendo ser conectado direto ao Arduino.

Na Figura 3.7 é mostrado o módulo de relés.



Figura 3.7 - Módulo de Relés (Autor: Ayrton Gissoni)

### 3.2.5 Célula de Peltier

Para a escolha correta da célula de Peltier, foram utilizados dois métodos de dimensionamento. O primeiro método é baseado em princípios de calorimetria, como a transferência de calor entre os corpos envolvidos no sistema e o segundo consiste em dimensionar a carga térmica total que é a soma da carga térmica ativa e passiva onde envolve os fenômenos da irradiação, condução e convecção.

#### 3.2.5.1 Dimensionamento da Célula de Peltier por Calorimetria

Esse método consiste em determinar a quantidade de calor entre os corpos envolvidos no sistema, no caso, para facilitar os cálculos considerou-se um sistema ideal, onde o calorímetro não trocará calor com o meio externo. Dentro do

compartimento da adega tem-se a célula de Peltier acoplada ao módulo frio (*cooler* e dissipador), a qual é responsável por fazer a climatização do compartimento; contendo duas garrafas de vinho e ar entre as garrafas.

De acordo com o princípio das trocas de calor (Equação 2.2), todo calor recebido tem que ser igual ao cedido.

$$\sum Q = 0$$

$$Q_{frio} + Q_{quente} = 0$$

$$Q_{frio} = -Q_{quente}$$

$$Q_{Peltier} = -(Q_{vinho} + Q_{ar})$$

$$Q_{Peltier} = -(m_{vinho} \cdot c_{vinho} \cdot \Delta T + m_{ar} \cdot c_{ar} \cdot \Delta T)$$

Para o cálculo da quantidade de calor do vinho foi adotado:

- $m = 1kg$  massa de uma garrafa de vinho;
- $c = 0,960 \frac{kcal}{kg \cdot ^\circ C}$ ;

Para o sistema foi definido as temperaturas:

- $T_f = 6^\circ C$  que é a temperatura final, a mais fria que se deseja atingir no sistema (menor temperatura do espumante);
- $T_i = 25^\circ C$  que é a temperatura inicial, ou seja, a temperatura ambiente;
- $\Delta T = T_f - T_i = 6 - 25 = -19^\circ C$ .

Calculando a quantidade de calor do vinho segundo Equação 2.3:

$$Q_{vinho} = m_{vinho} \cdot c_{vinho} \cdot \Delta T = 2 \times 1 \times 0,96 \cdot (-19) = -36,48kcal$$

Para o cálculo da massa de ar foi adotado:

- $densidade_{ar} = \frac{massa_{ar}}{Volume_{ar}} = 1,184 kg/m^3$  a  $25^\circ C$  e  $1atm$ ;
- $c = 0,240 cal/g \cdot ^\circ C$ .

Para o cálculo da quantidade de calor do ar foi calculado:

$$Volume = 280 \times 100 \times 310mm^3 = 0,00868m^3$$

$$massa_{ar} = Volume_{ar} \times densidade_{ar} = 0,00868 \times 1,184 = 0,01028kg$$

Calculando a quantidade de calor do ar:

$$Q_{ar} = m_{ar} \cdot c_{ar} \cdot \Delta T = 0,01028 \times 0,24 \times (-19) = -0,047cal$$

Como a quantidade de calor do ar é muito pequena se comparada a quantidade de calor do vinho, podemos desprezá-la para efeito dos cálculos.

Voltando ao cálculo do calor de Peltier:

$$Q_{Peltier} = -(-36,48 + 0) = 36,48Kcal$$

Para calculo da potência da célula de Peltier a ser utilizada, adotou 1 caloria como sendo 4,18 Joules:

$$Potência = \frac{Trabalho}{Tempo}$$

$$Trabalho = Q_{Peltier} \times 4,18 = 152,49kJ$$

Para que a adega seja climatizada em 1hora, ou 3600 segundos a potência da célula será de:

$$Potência = \frac{152,49 \times 10^3}{3600} = 42,36W$$

Para que a adega seja climatizada em 40 minutos, ou 2400 segundos a potência da célula será de:

$$Potência = \frac{152,49 \times 10^3}{2400} = 63,54W$$

### 3.2.5.2 Dimensionamento da Célula de Peltier por Carga Térmica

Esse método consiste em determinar a quantidade de calor com base no dimensionamento da carga térmica do sistema e da quantidade de calor a ser drenada do sistema. Os tipos da carga térmica podem ser: ativa, passiva ou uma combinação das duas.

O primeiro passo consiste em determinar as temperaturas de funcionamento que são:

- $T_{ambiente} = 25^{\circ}C$ ;

- $T_{fria} = 6^{\circ}C$  que é a temperatura mais baixa do terceiro compartimento;
- $\Delta T = 60^{\circ}C$  que é a diferença de temperatura entre as faces da célula de Peltier;
- Conseqüentemente a temperatura da superfície quente da célula de Peltier será  $T_{Quente} = 66^{\circ}C$ .

O segundo passo consiste em determinar as cargas térmicas envolvidas:

$$Q_{Térmica} = Q_{Radiação} + Q_{Condução} + Q_{Convecção} + Q_{ativa}$$

Para esse projeto a carga térmica de radiação será desprezada, pois não se trata de um sistema de altas temperaturas a vácuo.

Para o cálculo foi utilizado a Equação 2.8 da carga ativa que é a carga do cooler da parte interna da adega, será adotada a corrente de 0,2A e uma tensão de 12V.

$$Q_{ativa} = V.I = 12 \times 0,2 = 2,4W$$

Para o cálculo da carga térmica de condução e convecção será utilizado a combinação desses dois fenômenos:

$$Q_{Térmica} = Q_{passiva} + Q_{ativa}$$

Utilizando a Equação 2.12

$$Q_{passivo} = \frac{A \cdot \Delta T}{\frac{x}{k} + \frac{1}{h}}$$

Para o cálculo da área externa utilizou as dimensões externas do isopor de 450 X 251 X 371 mm.

$$A = (2 \times 450 \times 251 + 2 \times 450 \times 371 + 2 \times 251 \times 371) \cdot 10^{-6} = 0,746m^2$$

Cálculo da carga passiva segundo Equação 2.12:

$$Q_{passivo} = \frac{A \cdot \Delta T}{\frac{x}{k} + \frac{1}{h}} = \frac{0,746 \times 66}{\frac{0,03}{0,037} + \frac{1}{250}} = \frac{49,236}{0,811 + 0,004} = \frac{49,236}{0,815} = 60,42W$$

Então:

$$Q_{Térmica} = Q_{passiva} + Q_{ativa} = 60,42 + 2,4 = 62,82W$$

### 3.2.5.3 A Escolha da Célula de Peltier

Com base nos dois métodos de dimensionamento, para climatizar um compartimento da adega, precisamos de acordo com o método da calorimetria, uma célula de Peltier de 63,54W para climatizar em 40 minutos. De acordo com o dimensionamento por carga térmica, precisamos de uma célula de Peltier de 62,82W para climatizar um compartimento da adega.

Para compra da célula de Peltier foi encontrado o modelo TEC1-12706 (no Anexo D é mostrado o *datasheet da célula de Peltier*) com potência de 63W e que tem um custo de aproximadamente 30 reais no MercadoLivre e o modelo HTC40-07F-15.4 com potência de 62,2W e um custo de 80 reais no site <http://www.peltier.com.br>.

Para esse projeto foi utilizada para fazer a função de climatização da adega climatizada a Célula de Peltier. O modelo adotado foi o TEC1-12706, por ter um custo mais baixo. Este modelo possui comprimento e largura de 4cm e altura de 3,2mm. A corrente elétrica máxima que a Célula de Peltier pode suportar é de 6 Ampères, a tensão máxima é de 15,4V e a potência máxima é de 63W, pois essa pastilha termoelétrica abrange os valores de 62,82W a 63,54W que foram dimensionados para o projeto.

Para a construção do protótipo será utilizado um módulo de estágio simples para cada um dos três compartimentos da adega, totalizando assim a utilização de três Células de Peltier. Na Figura 3.8 é mostrada a Célula de Peltier.

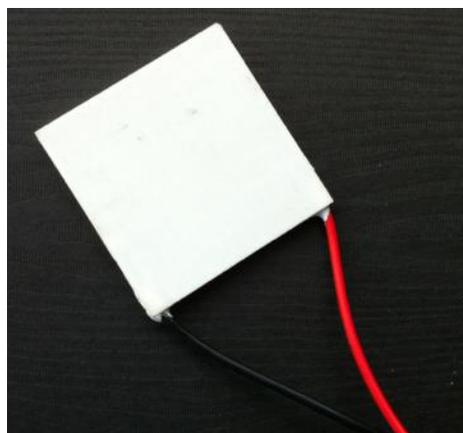


Figura 3.8 - Célula de Peltier (Autor: Ayrton Gissoni)

### 3.2.6 Fonte

Para esse projeto foram utilizadas duas fontes de alimentação chaveada ATX, para poder fornecer energia elétrica para as três células de Peltier. As fontes são da marca Coletex, modelo DSE 200P que possui as seguintes especificações:

- Entrada de 115V/230V com corrente 5/3A e frequência de 60/50Hz;
- Saída DC com tensão de +3,3V, +5V e +12V com corrente máxima de 7A, 14A e 10A respectivamente;
- Potência máxima de 200W.

### 3.2.7 Cooler com Dissipador

Para esse projeto foram utilizados seis *coolers* com dissipador, para poder dissipar o calor da célula de Peltier, sendo dois por célula, um em contato com a face quente e outro com a fria. Os *coolers* são da marca Multilaser, os dissipadores da face fria são da marca Multilaser e o dissipador da face quente é importado e foi comprado no site oficial de Pelier (<http://www.peltier.com.br>).

O *cooler* com dissipador da marca Multilaser possui as seguintes especificações:

- Material do dissipador é o alumínio 6063;
- Velocidade de rotação é de 2200 RPM;
- Voltagem DC é de 12V;
- Consumo de energia 2,4W;
- Fluxo de ar máximo de 40,9CFM;
- Vida útil é de 30mil horas para uma temperatura de 25 graus Celsius;
- Ruído de 25dB;
- Peso de 260g;
- Dimensões (unitário – ventilador): 95 X 95 X 25mm;
- Dimensões (unitário): 95 X 95 X 56,5mm.

Na Figura 3.9 é mostrado o *cooler* com dissipador da marca Multilaser.

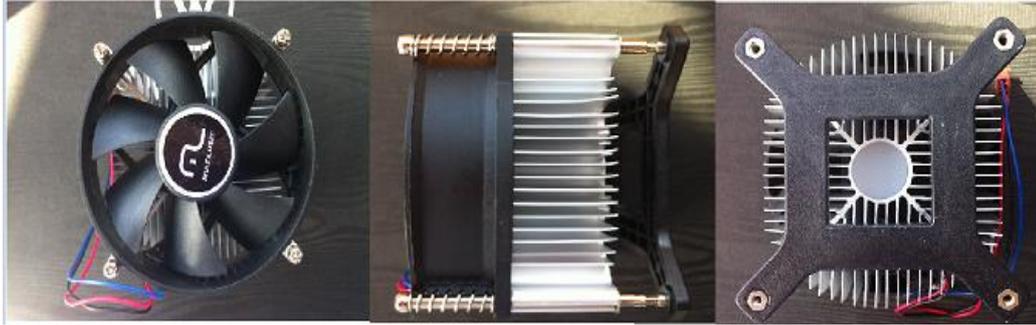


Figura 3.9 - Cooler com Dissipador (Autor: Ayrton Gissoni)

O dissipador da face quente possui as seguintes dimensões:

- Dimensões (unitário): 120 X 100 X 33mm.

Na Figura 3.10 é mostrado o dissipador da face quente.

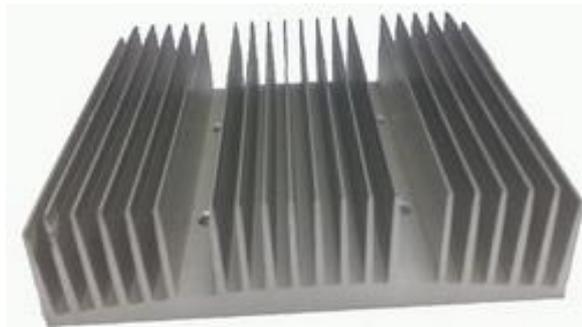


Figura 3.10 - Dissipador da Face Quente (Fonte: <http://www.peltier.com.br/produtos.htm>)

### 3.3 Montagem do *Hardware*

Para a montagem do *hardware*, utilizou-se dos materiais descritos na seção 3.2. A Figura 3.11 mostra um esquemático da montagem do *hardware*.

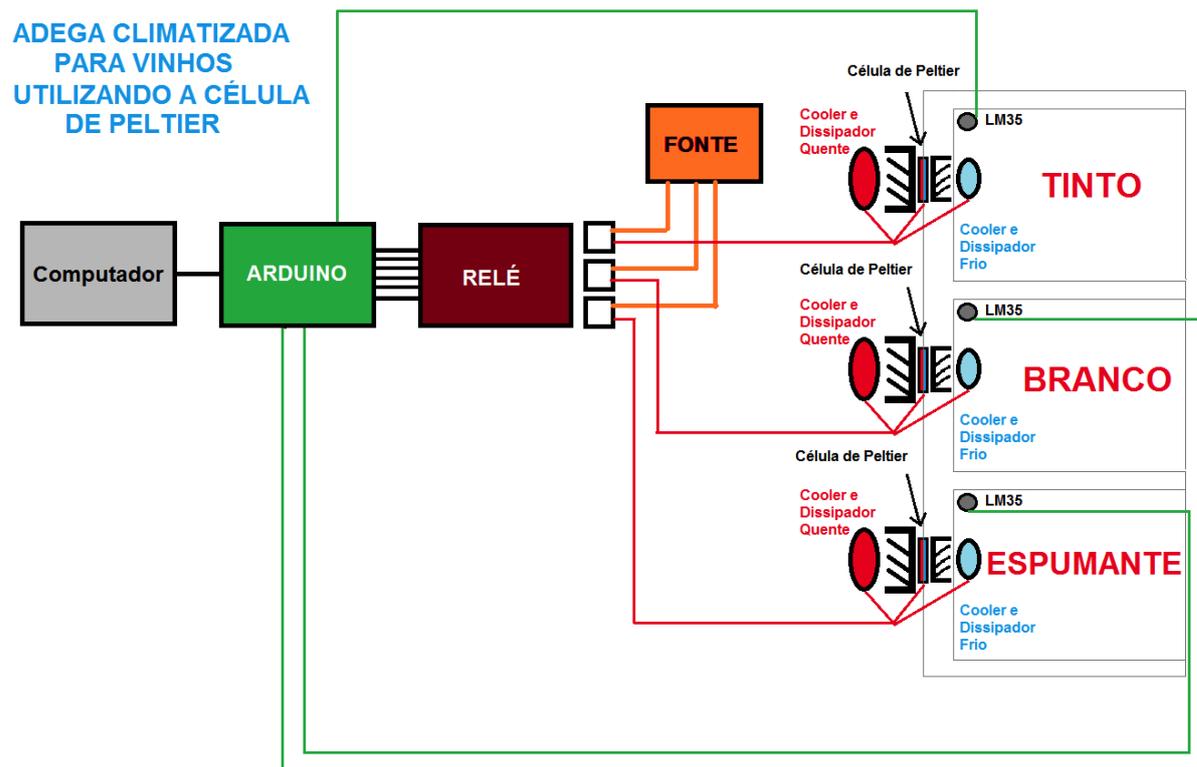


Figura 3.11 - Esquema de Montagem do Hardware (Autor: Ayrton Gissoni)

Foi montado um sistema de refrigeração utilizando como base para refrigeração de cada compartimento a célula de Peltier, utilizando dissipadores e *coolers* para o lado quente, para que não ocorresse o superaquecimento da célula e para o lado frio, para forçar a convecção de ar e o resfriamento do ar quente de dentro de cada um dos compartimentos. Utilizou-se o esquema “sanduíche”, que coloca a pastilha térmica no meio dos dissipadores frio e quente. Aplicou-se a pasta térmica na Célula de Peltier para que aumentasse a transferência de calor.

Os *Coolers* foram colocados em modo exaustor (ventando para fora), pois após vários testes observou-se que a célula de Peltier ficava com uma eficiência melhor (do que ventando para dentro) para resfriar o compartimento interno, pois quanto mais retirou-se o calor do lado quente da célula (o calor flui conforme o fluxo do vento), melhor a célula conseguia atingir menores temperaturas e possuir uma melhor eficiência.

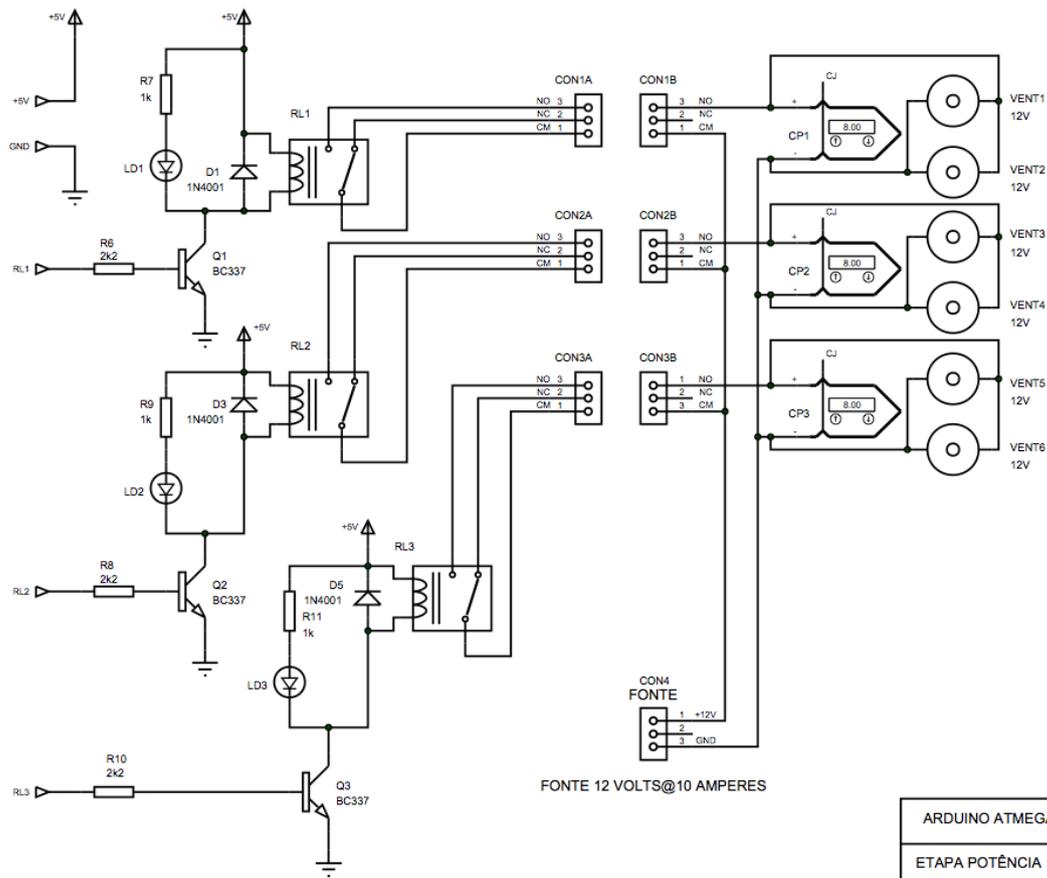
O *cooler* e o dissipador do lado quente ficaram fora do compartimento, a célula de Peltier e o dissipador frio ficaram no meio do isopor (buraco feito sob

medida na parte posterior da caixa de isopor) e o *cooler* frio ficou dentro da caixa de isopor.

Para o protótipo, foram utilizadas duas fontes de 12V, cada uma com capacidade de 10A, um total de 20A alimentando todo o sistema de climatização onde cada célula de Peltier conseguia “puxar” uma corrente de 4A, apesar da célula de Peltier ser dimensionada utilizando teoricamente uma corrente de 6A, essa corrente não foi obtida durante a realização dos testes do protótipo.

Cada fonte foi ligada em paralelo alimentando a célula de Peltier e os seus *coolers* (frio e quente). Utilizou-se do módulo de relés para que fosse feito o acionamento de cada módulo de Peltier, de cada compartimento, de acordo com a temperatura apropriada.

O módulo relé faz o acionamento do sistema de refrigeração (funcionando como “chave”) de acordo com os comandos enviados pelo Arduino. O módulo relé é alimentado com 5V pelo Arduino, onde utilizamos a alimentação VCC e o GND (terra). Para o envio de dados para o módulo relé, utilizou-se as portas digitais 22, 24 e 26 do Arduino (uma para cada compartimento) ligando-as na chave 1, 2 e 3 do módulo relé respectivamente. Na saída do módulo relé, colocou-se no NO (*Normally Open* – Normalmente Aberto) o fio positivo (fio vermelho) da célula de peltier e dos *coolers*, colocou-se no CM (Comum) o fio positivo da fonte (amarelo) e o NC (*Normally Closed* – Normalmente Fechado) não foi utilizado. Na Figura 3.12 é mostrado o desenho esquemático do circuito elétrico.



**Figura 3.12 - Esquema Elétrico de Climatização (Autor: Ayrton Gissoni)**

O Arduino recebe do computador, via interface serial, a temperatura desejada para cada compartimento e faz a leitura do sensor de temperatura LM35 que está localizado em cada um dos compartimentos, utilizando as portas analógicas 2, 4 e 6. Essa leitura feita pelo Arduino é repassada ao computador pela interface serial. A temperatura lida é comparada com a temperatura desejada, verificando se é maior ou menor, ligando ou desligando o sistema de refrigeração. Também é medida a temperatura ambiente por um sensor localizado fora da adega, utilizando a porta analógica 0. Na Figura 3.13 é mostrado o desenho esquemático do circuito elétrico.

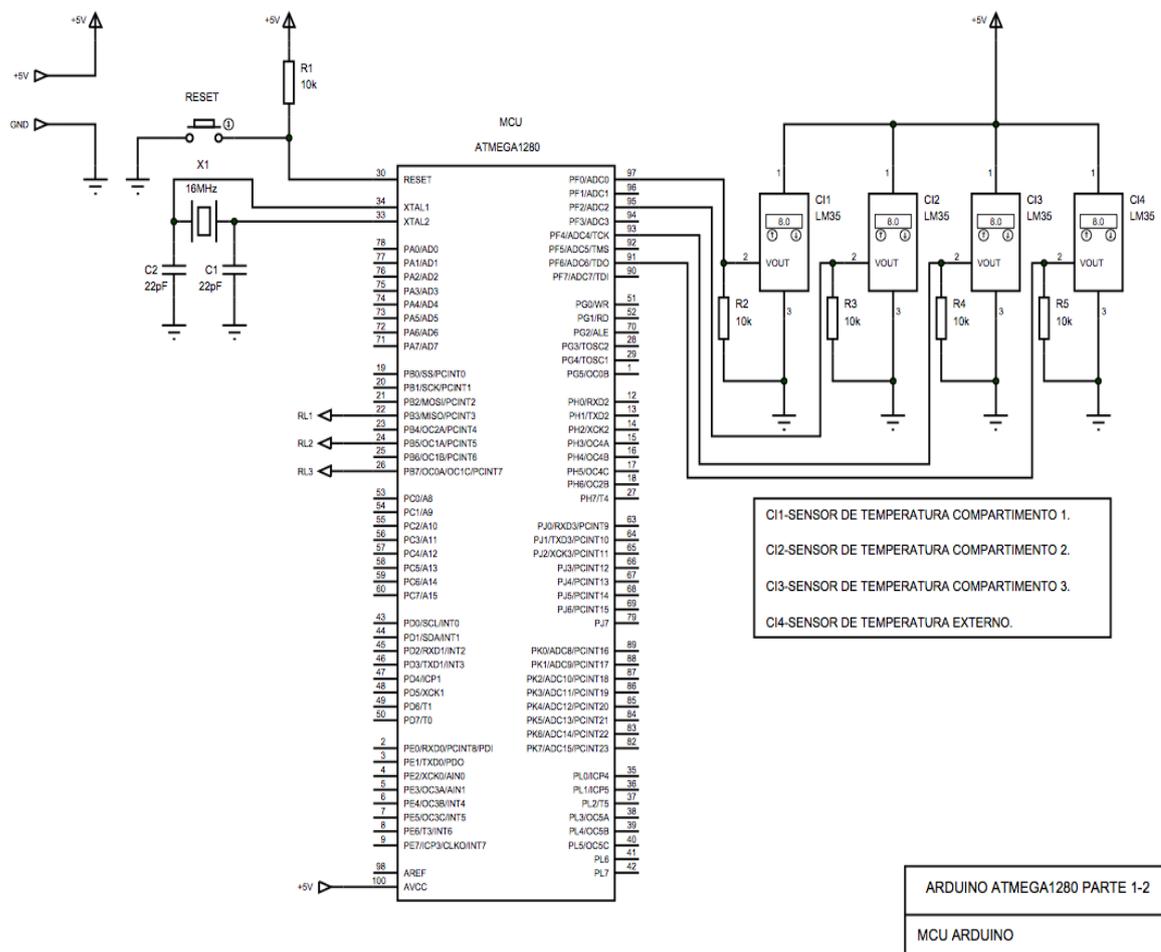


Figura 3.13 - Esquema Elétrico de Controle de Temperatura (Autor: Ayrton Gissoni)

O sensor de temperatura (LM35) foi colocado em cada compartimento, utilizando um fio blindado para que fosse amenizado o máximo possível os ruídos externos que atrapalham na leitura da temperatura fazendo com que o sensor a leia de forma errada. O LM35 foi alimentado pelo Arduino através do VCC ligando no +Vs do LM35 e GND no GND. Para transmissão da leitura da temperatura foi utilizado a saída Vout do sensor ligada em uma porta analógica do Arduino. Para evitar ruído externo quando o sensor de temperatura estivesse “desligado”, foi colocado um resistor de baixa impedância (10K Ohms) entre Vout do LM35 e o GND do Arduino conforme mostrado na Figura 3.14.

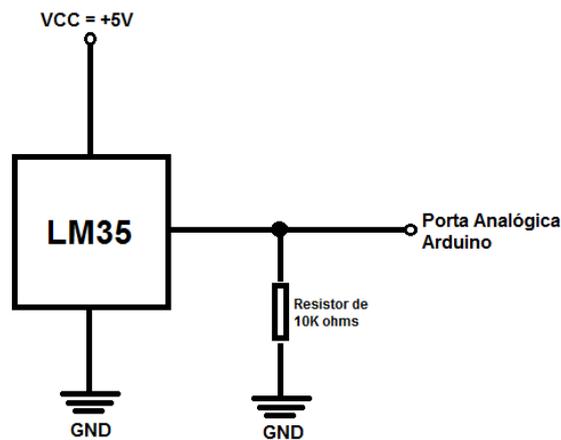


Figura 3.14 - Ligação do Sensor LM35 (Autor: Ayrton Gissoni)

### 3.4 Software de Controle de Temperatura - Arduino

Com as informações referentes a temperatura de cada um dos compartimentos feita pelo sensor de temperatura LM35 e transmitida ao Arduino para que ele possa controlar a temperatura dos compartimentos ligando e desligando o circuito através do módulo relé, foi desenvolvido através da IDE para a programação em Arduino, um programa (descrito no Apêndice A) que faz esse controle de temperatura.

O Arduino possui a função *setup*, a qual é feita a configuração dos pinos e portas utilizados. Nessa função foi iniciada as portas digitais (22, 24 e 26) referentes aos compartimentos, os quais possuem um sistema de climatização que serão ligados e desligados pelo módulo relé. Também é na função *setup* que é inicializada a porta serial.

Na função *loop*, que é uma função que fica executando como um loop o programa do Arduino, foi feita a comunicação com o *software* de cadastramento de vinhos (descrito no Apêndice C) para que o usuário possa selecionar a temperatura desejada. É nessa função que é lida a temperatura de cada compartimento por meio do comando “*analogRead (sensortemperatura)*”, onde é passado qual é a porta analógica que será lido o sinal enviado pelo sensor de temperatura (portas 0, 2, 4 e 6).

Na função loop, também é calculado uma média e enviada ao *software* de cadastramento de vinhos utilizando o comando “Serial.println(mediaTemperatura)”, o qual também manda a temperatura desejada pelo usuário que é utilizada como parâmetro de referência para que o Arduino ligue e desligue o sistema de climatização.

A temperatura desejada pelo usuário é enviada pelo Java e recebida pelo Arduino através do comando “Serial.read()” que lê a porta serial. É através de uma classe de comunicação no *software* de cadastro de vinhos, que é enviado pela porta serial a temperatura desejada pelo usuário. Essa classe é a *ComunicacaoArduino.java* que possui a bibliotecas *RXTXcomm.jar* a qual reconhece a porta serial do Arduino para o envio das informações.

Por causa de ruídos e alterações na leitura do LM35, foi feita uma média de temperatura que é comparada com a temperatura de referência dada pelo usuário (temperatura desejada), para que o Arduino envie o comando, passando o compartimento, ao módulo relé mandando desligar e ligar o sistema de climatização, através do comando “digitalWrite (compartimento, HIGH)” e “digitalWrite (compartimento, LOW)” respectivamente.

## CAPÍTULO 4 DESENVOLVIMENTO DO SOFTWARE DE CADASTRAMENTO DOS VINHOS

Este capítulo apresenta a descrição do desenvolvimento do *software* para cadastro de vinhos da adega climatizada, que utiliza como base a linguagem Java e o banco de dados MySQL. O código do *software* é descrito no Apêndice C.

### 4.1 Descrição da Arquitetura do Software

Este projeto foi feito com base na arquitetura MVC (*Model* – Modelo, *View* – Visão e *Controller* – Controle) que tem como objetivo principal separar os dados (Modelo) da interface do usuário (Visão) e da regra de negócio da aplicação (controle).

O MVC é um padrão de desenvolvimento e *design* que separa uma aplicação em três camadas distintas. Uma camada de Modelo que se relaciona ao estado dos dados atuais que a aplicação administra, outra camada que é a Visão, a qual se exibem os dados ou informações da aplicação, e a terceira camada é o Controle que coordena as duas camadas anteriores, exibindo a interface correta ou executando algum trabalho relacionado aos dados que a aplicação precisa. As principais características do MVC são:

- Possui re-usabilidade, pode-se criar bibliotecas e adicionar novas interfaces para a aplicação facilmente;
- A possibilidade de se desenvolver em paralelo, podendo começar o projeto por qualquer uma das camadas;
- Divisão de responsabilidades, onde os programadores trabalham na programação e *Web designers* na construção visual do *software*;
- Redução do esforço na manutenção do *software*, pois as alterações são efetuadas separadamente (em cada uma das camadas) não afetando as outras camadas do sistema.

Na Figura 4.1 é mostrada a arquitetura MVC utilizada para construção do *software* de cadastramento de vinhos. O código foi feito na IDE NetBeans, foi utilizado o banco de dados MySQL, para comunicação entre a linguagem Java e o

Banco de Dados foi utilizado o drive JDBC, para apresentar a aplicação para o navegador foi utilizado o servidor de aplicação TomCat.

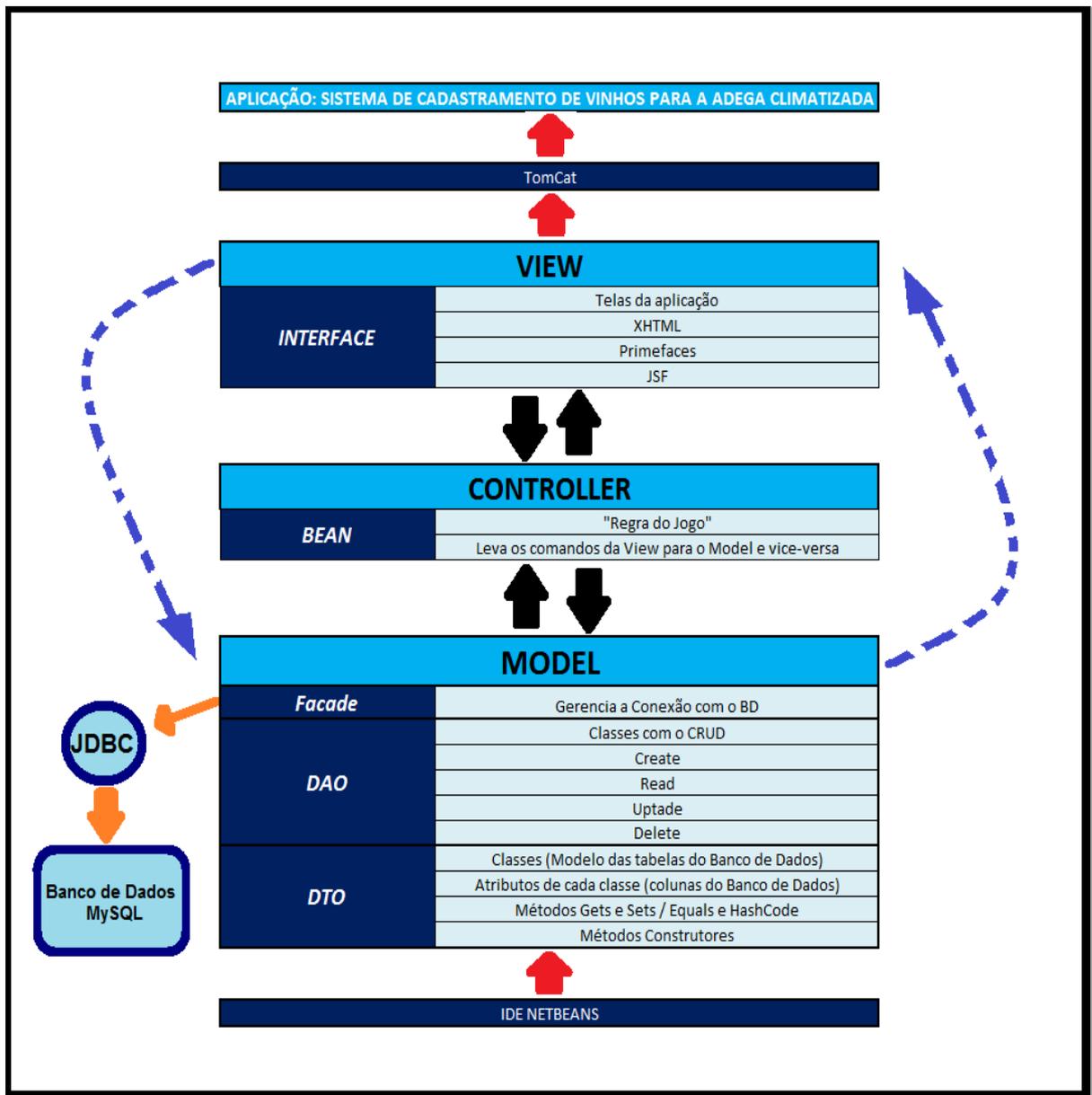


Figura 4.1 - Arquitetura MVC (Autor: Ayrton Gissoni)

#### 4.1.1 Modelo

O modelo é a camada responsável pela persistência dos dados na aplicação contendo todas as funcionalidades como criar, visualizar, atualizar e excluir os dados (CRUD), pode também possuir outras responsabilidades com o

controle de chaves, ordenação entre outras. Essa camada representa um objeto real como a garrafa de vinho.

O modelo representa os dados do cadastro e utiliza a linguagem SQL para manipular dados do banco de dados (MySQL). O modelo somente acessa a base de dados e deixa os dados prontos para o controlador que os encaminha para a camada de visão, pois o modelo não precisa saber onde será apresentado seus dados, basta disponibilizá-los corretamente e garantir a sua persistência.

Para facilitar o desenvolvimento do modelo, este foi dividido em mais três camadas, as quais se comunicam umas com as outras, elas são: DTO, DAO e Facade.

#### **4.1.1.1 DTO (*Data Transfer Object*)**

O DTO é uma subcamada do modelo, é chamado de *Data Transfer Object*, que significa objeto de transferência de dados. O DTO faz uma abstração de cada uma das tabelas, onde cada tabela se torna uma classe e cada coluna um atributo desta classe. No projeto, o elemento principal é a garrafa de vinho, que no banco de dados (BD) é uma tabela (tblGarrafa) e no DTO é uma classe Garrafa que possui os atributos (informações sobre o vinho) de id (que é a chave primária do meu BD), nome, tipo, ano, amadurecimento, potencial, teor alcoólico, uva e vinícola.

O DTO possui além dos atributos da classe, métodos que são utilizados como padrões de boas práticas pelos programadores da linguagem Java, são os métodos construtores da classe que criam a garrafa de vinho, os métodos gets e sets que servem para obter ou atribuir um valor a um determinado atributo da garrafa de vinho, o método hashCode que serve para retornar um código hash de um objeto usado normalmente para agilizar a busca em collections, o método equals que serve para comparar se um objeto é igual a outro e o método toString que lista os atributos do objeto garrafa. Na Figura 4.2 é mostrado parte do código da principal classe do DTO do projeto.

```

public class Garrafa implements Serializable {}

//Atributos da Classe
private Integer id;
private String nome;
private String tipo;
private Integer ano;
private String amadurecimento;
private Integer potencial;
private Double teor;
private Uva uva;
private Vinicola vinicola;

//Métodos Construtores
public Garrafa() {...}

public Garrafa(String nome, String tipo, Integer ano, String amadurecimento,
                Integer potencial, Double teor, Uva uva, Vinicola vinicola) {
    this.nome = nome;
    this.tipo = tipo;
    this.ano = ano;
    this.amadurecimento = amadurecimento;
    this.potencial = potencial;
    this.teor = teor;
    this.uva = uva;
    this.vinicola = vinicola;
}

```

Figura 4.2 - Código JAVA DTO (Autor: Ayrton Gissoni)

#### 4.1.1.2 DAO (*Data Access Object*)

O DAO é uma subcamada do modelo, é chamado de *Data Access Object*, que significa objeto de acesso a dados. O DAO é responsável pelo acesso a dados do banco de dados da adega. O DAO utiliza a linguagem SQL para criar, ler, atualizar e deletar (CRUD) informações relacionadas aos vinhos da adega.

No projeto foi criado a classe GarrafaDao, que possui os métodos inserir que é utilizado para inserir uma nova garrafa de vinho no meu banco de dados, o método buscar todos retorna uma lista contendo todas as garrafas de vinho do BD, o método buscar por id que procura no BD por um único vinho através da chave primária, o método alterar que permite alterar os dados da garrafa de vinho e o método excluir que deleta uma garrafa de vinho do BD. Na Figura 4.3 é mostrado a classe GarrafaDao e seus métodos.

```

public class GarrafaDao implements IDao<Garrafa>{

    private Conexao conexao;
    private PreparedStatement ps;
    private ResultSet rs;

    public GarrafaDao(Conexao conexao) {...}

    @Override
    public void inserir(Garrafa garrafa) throws SQLException {...}

    @Override
    public List<Garrafa> buscarTodos() throws SQLException {...}

    @Override
    public Garrafa buscarPorID(Integer id) throws SQLException {...}

    @Override
    public void alterar(Garrafa garrafa) throws SQLException {...}

    @Override
    public void excluir(Garrafa garrafa) throws SQLException {...}

}

```

Figura 4.3 - Código JAVA DAO (Autor: Ayrton Gissoni)

A classe GarrafaDao utiliza o objeto “ps” da classe PreparedStatement que é responsável por executar no BD as seguintes SQL’s para cada um dos métodos:

- "INSERT INTO tblgarrafa (nomeGarrafa, anoGarrafa, tipoGarrafa, amadurecimentoGarrafa, potencialGarrafa, teorGarrafa, idUva, idVinicola) VALUES(?,?,?,?,?,?,?,?)";
- "SELECT \* FROM tblgarrafa";
- "SELECT \* FROM tblgarrafa WHERE idGarrafa = ?";
- "UPDATE tblgarrafa SET nomeGarrafa = ?, tipoGarrafa = ?, anoGarrafa = ?, amadurecimentoGarrafa = ?, potencialGarrafa = ?, teorGarrafa = ?, idUva = ?, idVinicola = ? WHERE idGarrafa = ?";
- "DELETE FROM tblgarrafa WHERE idGarrafa = ?";

O resultado de cada uma das *queries*, por exemplo no método buscarTodos, é armazenado no objeto “rs” da classe ResultSet e posteriormente

passado ao objeto garrafa da classe Garrafa que é guardado numa lista de garrafas, onde no final do método buscarTodos é retornado o objeto garrafas do tipo List. Na Figura 4.4 é mostrado o código do método buscarTodos.

```

public List<Garrafa> buscarTodos() throws SQLException {
    List<Garrafa> garrafas = null;
    Garrafa garrafa = null;

    ps = conexao.getConnection().prepareStatement("SELECT * FROM tblgarrafa");
    rs = ps.executeQuery();

    if(rs.next()) {
        garrafas = new ArrayList<>();
        do {
            garrafa = new Garrafa();
            garrafa.setId(rs.getInt("idGarrafa"));
            garrafa.setNome(rs.getString("nomeGarrafa"));
            garrafa.setTipo(rs.getString("tipoGarrafa"));
            garrafa.setAno(rs.getInt("anoGarrafa"));
            garrafa.setAmadurecimento(rs.getString("amadurecimentoGarrafa"));
            garrafa.setPotencial(rs.getInt("potencialGarrafa"));
            garrafa.setTeor(rs.getDouble("teorGarrafa"));
            garrafa.setUva(new UvaDao(conexao).buscarPorID(rs.getInt("idUva")));
            garrafa.setVinicola(new VinicolaDao(conexao).buscarPorID(rs.getInt("idVinicola")));
            garrafas.add(garrafa);
        } while(rs.next());
    }

    return garrafas;
}

```

Figura 4.4 - Código JAVA Método buscarTodos (Autor: Ayrton Gissoni)

### 4.1.1.3 FACADE

O Facade é a “Fachada” que é utilizada como boa prática em padrões de projetos e serve para gerenciar as conexões com o banco de dados e invocar o DAO para executar as *queries*.

No projeto, para gerenciar as conexões e invocar o DAO, foi desenvolvido a classe GarrafaFacade, que possui os mesmos métodos do DAO, porém agora o Facade abre e fecha as conexões com o BD. Na Figura 4.5 é mostrado o código da classe GarrafaFacade.

```

public class GarrafaFacade {

    private Conexao conexao = new Conexao();
    private GarrafaDao dao;
    private List<Garrafa> garrafas;

    public List<Garrafa> buscarTodos()
    {
        garrafas = null;
        try {
            conexao.abrir();
            dao = new GarrafaDao(conexao);
            garrafas = dao.buscarTodos();
        } catch (SQLException ex) {
            Logger.getLogger(GarrafaFacade.class.getName()).log(Level.SEVERE, null, ex);
        } finally {
            conexao.fechar();
        }
        return garrafas;
    }

    public void inserir(Garrafa garrafa)
    {...}

    public void excluir(Garrafa garrafa)
    {...}

    public void alterar(Garrafa garrafa)
    {...}

}

```

Figura 4.5 - Código JAVA FACADE (Autor: Ayrton Gissoni)

### 4.1.2 Controle

A camada de controle serve para controlar e gerenciar a comunicação entre as camadas de modelo e visão, controlando o fluxo de dados, regras de negócios e ações dos usuários. O controle faz a interpretação das ações, solicitações e interações da interface (visão) do usuário e se comunica com o modelo (base de dados da aplicação). Então o modelo executa a solicitação e atualiza a visão com os resultados da solicitação. Sendo assim o comportamento da aplicação sempre passa pelo controle, o qual possui as regras de negócios da aplicação.

### 4.1.2.1 BEAN

É a própria camada de controle, que tem a função de gerenciar as informações entre as camadas de visão e de modelo. O Bean é a parte mais “inteligente” do *software*, pois é ele quem aplica as regras de negócio (a lógica do programa) e toma as decisões para acionar o modelo e atualizar a visão. É no Bean que são feitas as validações e conferência dos dados.

O cadastro da garrafa de vinho é feito utilizando o Bean cuja a classe é chamada de *GarrafaInserirBean* que tem a função de cadastrar uma nova garrafa (*novaGarrafa*) no BD. Para isso o *GarrafaInserirBean* recebe as informações cedidas pelo usuário na camada de visão, a qual repassa essas informações ao *GarrafaInserirBean* que executa o método “inserir”, este método chama o método *GarrafaFacade* (da camada modelo-Facade) que por sua vez chama e executa o método “inserir” do *GarrafaDao* (camada modelo-DAO). Na Figura 4.6 é mostrado o código do método *inserir*.

```
public String inserir() {
    try {
        for(Uva uva : listaDeUvas) {
            if(uva.getNome().equals(novaGarrafa.getUva().getNome())) {
                novaGarrafa.setUva(uva);
            }
        }

        for(Vinicola vinicola : listaDeVinicolas) {
            if(vinicola.getId() == novaGarrafa.getVinicola().getId()) {
                novaGarrafa.setVinicola(vinicola);
            }
        }

        GarrafaFacade garrafaFacade = new GarrafaFacade();
        garrafaFacade.inserir(novaGarrafa);

        FacesContext.getCurrentInstance().addMessage(null, new FacesMessage(FacesMessage.SEVERITY_INFO,
            "Inserir", "A garrafa foi inserida com sucesso"));
    } catch(Exception ex) {
        Logger.getLogger(GarrafaInserirBean.class.getName()).log(Level.SEVERE, null, ex);
        FacesContext.getCurrentInstance().addMessage(null, new FacesMessage(FacesMessage.SEVERITY_ERROR, "Inserir",
            "Ocorreu um erro ao inserir uma garrafa. Por favor, tente novamente."));
    }

    return "garrafa.xhtml?faces-redirect=true";
}
```

Figura 4.6 - Código JAVA Método *inserir* (Autor: Ayrton Gissoni)

O Bean relacionado a excluir as garrafas de vinho, se chama *GarrafaBean*, esta classe possui um método chamado “excluir”, que faz a chamada

a classe GarrafaFacade e excuta o método “excluir” a garrafa selecionada. Na Figura 4.7 é mostrado parte do código da classe GarrafaBean.

```
public class GarrafaBean implements Serializable {

    private List<Garrafa> listaFiltro;
    private List<Garrafa> listaDeGarrafas;
    private List<Uva> listaDeUvas;
    private List<Pais> listaDePaises;
    private List<Regiao> listaDeRegioes;
    private List<Vinicola> listaDeVinicolas;
    private Garrafa garrafaSelecionada;
    private Garrafa novaGarrafa;

    public GarrafaBean() {
        listaFiltro = new ArrayList<Garrafa>();
        listaDeGarrafas = new GarrafaFacade().buscarTodos();
        listaDeUvas = new UvaFacade().buscarTodos();
        listaDePaises = new PaisFacade().buscarTodos();
        listaDeRegioes = new ArrayList<Regiao>();
        listaDeVinicolas = new ArrayList<Vinicola>();

        inicializarGarrafaSelecionada();
        inicializarNovaGarrafa();
    }

    Getters & Setters

    public void excluir(ActionEvent event) {
        GarrafaFacade garrafaFacade = new GarrafaFacade();
        garrafaFacade.excluir(garrafaSelecionada);

        inicializarGarrafaSelecionada();
    }
}
```

**Figura 4.7 - Código JAVA Classe GarrafaBean (Autor: Ayrton Gissoni)**

Para fazer alterações no cadastro da garrafa de vinho, foi utilizado o Bean GarrafaAlterarBean que recebe da visão os dados alterados pelo usuário e os manda para o modelo que atualiza o cadastro. Na Figura 4.8 é mostrado parte do código da classe GarrafaAlterarBean.

```

public class GarrafaAlterarBean implements Serializable {

    private List<Garrafa> listaDeGarrafas;
    private List<Uva> listaDeUvas;
    private List<Pais> listaDePaises;
    private List<Regiao> listaDeRegioes;
    private List<Vinicola> listaDeVinicolas;

    private Garrafa garrafaSelecionada;

    public GarrafaAlterarBean() {
        listaDeGarrafas = new GarrafaFacade().buscarTodos();
        listaDeUvas = new UvaFacade().buscarTodos();
        listaDePaises = new PaisFacade().buscarTodos();
        listaDeRegioes = new ArrayList<Regiao>();
        listaDeVinicolas = new ArrayList<Vinicola>();
        garrafaSelecionada = new Garrafa();
        garrafaSelecionada.setUva(new Uva());
        garrafaSelecionada.setVinicola(new Vinicola());
    }

    Getters & Setters

    public String alterar() {
        try {
            for(Uva uva : listaDeUvas) {
                if(uva.getNome().equals(garrafaSelecionada.getUva().getNome())) {
                    garrafaSelecionada.setUva(uva);
                }
            }
            for(Vinicola vinicola : listaDeVinicolas) {
                if(vinicola.getId() == garrafaSelecionada.getVinicola().getId()) {
                    garrafaSelecionada.setVinicola(vinicola);
                }
            }
            GarrafaFacade garrafaFacade = new GarrafaFacade();
            garrafaFacade.alterar(garrafaSelecionada);
            FacesContext.getCurrentInstance().addMessage(null, new FacesMessage(FacesMessage.SEVERITY_INFO,
                "Alterar", "A garrafa foi alterada com sucesso"));
        } catch(Exception ex) {
            Logger.getLogger(GarrafaAlterarBean.class.getName()).log(Level.SEVERE, null, ex);
            FacesContext.getCurrentInstance().addMessage(null, new FacesMessage(FacesMessage.SEVERITY_ERROR,
                "Alterar", "Ocorreu um erro ao alterar uma garrafa. Por favor, tente novamente."));
        }
        return "garrafa.xhtml?faces-redirect=true";
    }
}

```

Figura 4.8 - Código JAVA da Classe GarrafaAlterarBean (Autor: Ayrton Gissoni)

### 4.1.3 Visão

A camada de visão é a interface de representação da parte visual da apresentação do modelo para o usuário, é através dela que a entrada de dados e a visualização de respostas são geradas. Ela contém os formulários, as tabelas, os menus e os botões para a entrada e saída dos dados. O papel da visão é transformar os dados vindos do modelo em informações, mascarando, manipulando,

alinhando, concatenando, ajustando, separando e substituindo os códigos por descrições que melhor representem o objeto real, no caso, a garrafa de vinho.

A visão tem o dever de garantir que a apresentação reflita o estado do modelo, de forma que quando os dados do modelo mudam, a visão é notificada pelo controlador e se atualiza baseada em mudanças de estado do modelo e quando a visão necessita incluir, mudar ou deletar um determinado dado, a camada de controle notifica o modelo, o qual se atualiza. Isso acontece porque a camada de visão não necessita saber onde e nem como seus dados são armazenados.

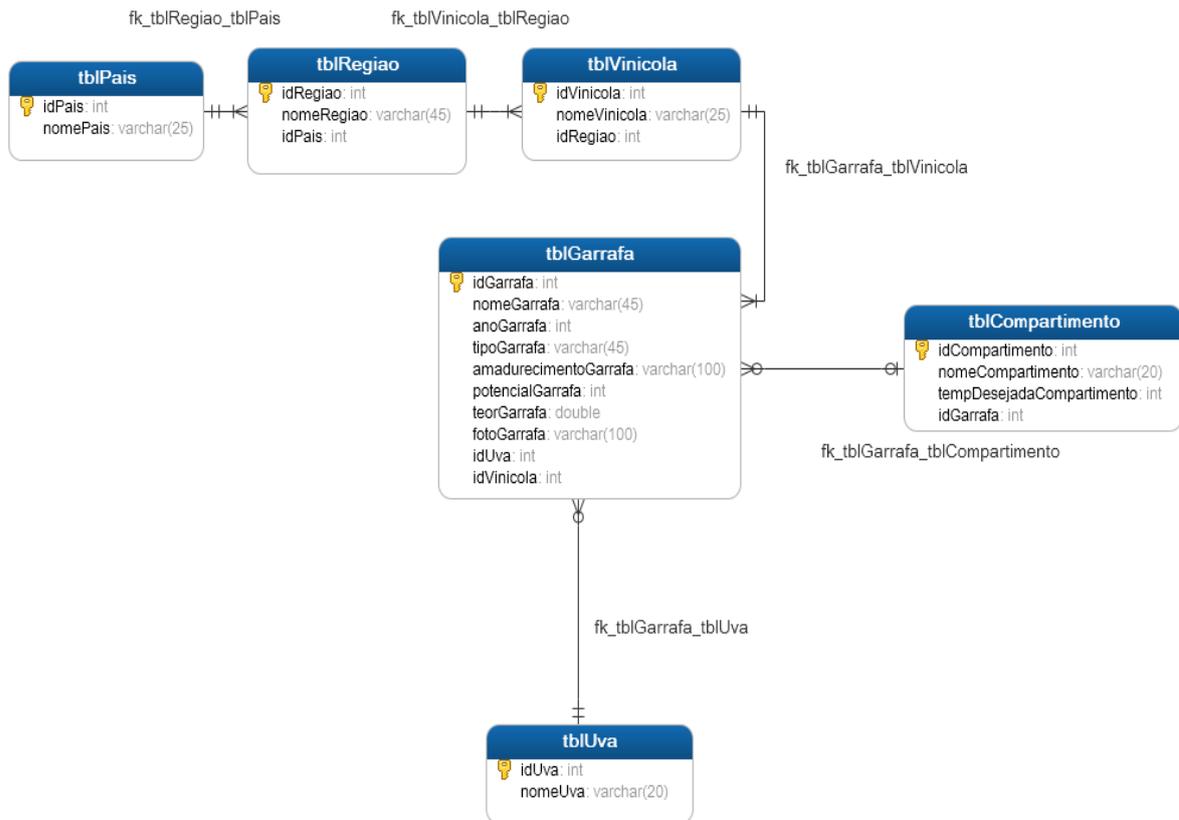
## **4.2 Banco de Dados**

Para este projeto foi utilizado o banco de dados MySQL que é um sistema de gerenciamento de banco de dados relacional (SGBD) de código aberto que utiliza a linguagem SQL (*Structure Query Language* – Linguagem de Consulta Estruturada) que é uma linguagem para inserir, acessar e gerenciar o conteúdo armazenado num banco de dados. A escolha do MySQL foi devido o conhecimento já adquirido sobre essa ferramenta ao longo do curso de engenharia e também por causa que o MySQL é usado na maioria das aplicações gratuitas para gerir bases de dados.

### **4.2.1 Modelo de Entidade de Relacionamento**

O modelo de entidade de relacionamento foi utilizado neste projeto para descrever de maneira conceitual e lógica a representação das entidades, que são as tabelas do banco de dados responsáveis por guardarem os dados referentes ao vinho a ser cadastrado, e seu relacionamento de umas com as outras.

Na Figura 4.9 é mostrado o MER (Modelo de Entidade de Relacionamento) entre as tabelas do banco de dados, onde a entidade central (mais importante) é a garrafa de vinho, que possui informações como o nome, ano, tipo, método de amadurecimento da garrafa, potencial de guarda, teor alcoólico do vinho, tipo de uva e vinícola que fez o vinho.



**Figura 4.9 - MER (Autor: Ayrton Gissoni)**

No MER da adega climatizada as entidades (tabelas) se relacionam umas com as outras, por exemplo o relacionamento entre tabelas 1 para N:

- A Tabela País se relaciona com a Tabela Região, pois um determinado país pode ter várias regiões produtoras de vinho;
- A Tabela Região se relaciona com a Tabela Vinícola, pois uma região pode ter várias vinícolas;
- A Tabela Vinícola se relaciona com a Tabela Garrafa, pois uma vinícola produz várias garrafas de vinho;
- A Tabela Uva se relaciona com a Tabela Garrafa, pois a partir de um tipo de uva pode-se fazer diversas garrafas de vinho.

Já o relacionamento entre a Tabela Compartimento e a Tabela Garrafa é de apenas um tipo de compartimento possui certas garrafas de vinho, ou seja, uma garrafa de vinho só pode estar ligada exclusivamente a um determinado compartimento.

### 4.2.2 JDBC e Classe de Conexão

O sistema de cadastramento de vinhos é desenvolvido em Java, o qual abstrai os métodos através do qual é possível se fazer uma conexão, pois as conexões são feitas através de uma ponte que implementa todas as funcionalidades que um banco de dados padrão deve nos fornecer, como por exemplo, toda conexão deve permitir executar código de pesquisa, atualização entre outras.

A linguagem Java manipula seus dados através do driver JDBC (*Java Database Connectivity*) que é uma API (conjunto de classes, métodos e interfaces) capaz de fazer o envio das instruções em linguagem SQL para o banco de dados, no caso foi utilizado o banco de dados MySQL.

O serviço de encontrar uma ponte, ou seja, um *driver* certo é delegado para um controlador de drivers (*DriverManager*) que gerencia os *drivers*. O *DriverManager* chama um método `getConnection` passando os parâmetros de endereço (URL), usuário (USR) e senha (PSW) para abrir a conexão.

Na Figura 4.10 é mostrado o código utilizado para fazer a conexão com o banco de dados MySQL, utilizando o drive JDBC. A classe *Conexão* possui os métodos `abrir` e `fechar`, responsáveis por abrir e fechar a conexão com o banco de dados.

```

public class Conexao {

    private Connection connection;
    private static final String USR = "root";
    private static final String PSW = "talita";
    private static final String URL = "jdbc:mysql://localhost/adeaga";

    public PreparedStatement ps;
    public ResultSet rs;

    public Connection getConnection() {
        return connection;
    }

    public void abrir() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            connection = DriverManager.getConnection(URL, USR, PSW);
            System.out.println("CONEXAO: conexão aberta com sucesso");
        } catch (SQLException ex) {
            System.out.println("CONEXAO: falha ao abrir conexão");
            Logger.getLogger(Conexao.class.getName()).log(Level.SEVERE, null, ex);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(Conexao.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    public void fechar() {
        try {
            if(connection != null || !connection.isClosed()) {
                connection.close();
                System.out.println("CONEXAO: conexão fechada com sucesso!");
            }
        } catch (SQLException ex) {
            System.out.println("CONEXAO: falha ao fechar conexão");
            Logger.getLogger(Conexao.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

```

Figura 4.10 - Código JAVA Classe Conexão (Autor: Ayrton Gissoni)

### 4.3 Interface da Aplicação

A interface do *software* de cadastro de vinhos é feita em linguagem Xhtml, foi utilizado um *framework* chamado PrimeFaces, o qual possui componentes JSF, botões, tabelas e formulários dinâmicos e que são atualizados pelo Ajax, evitando que se recarregue toda a tela.

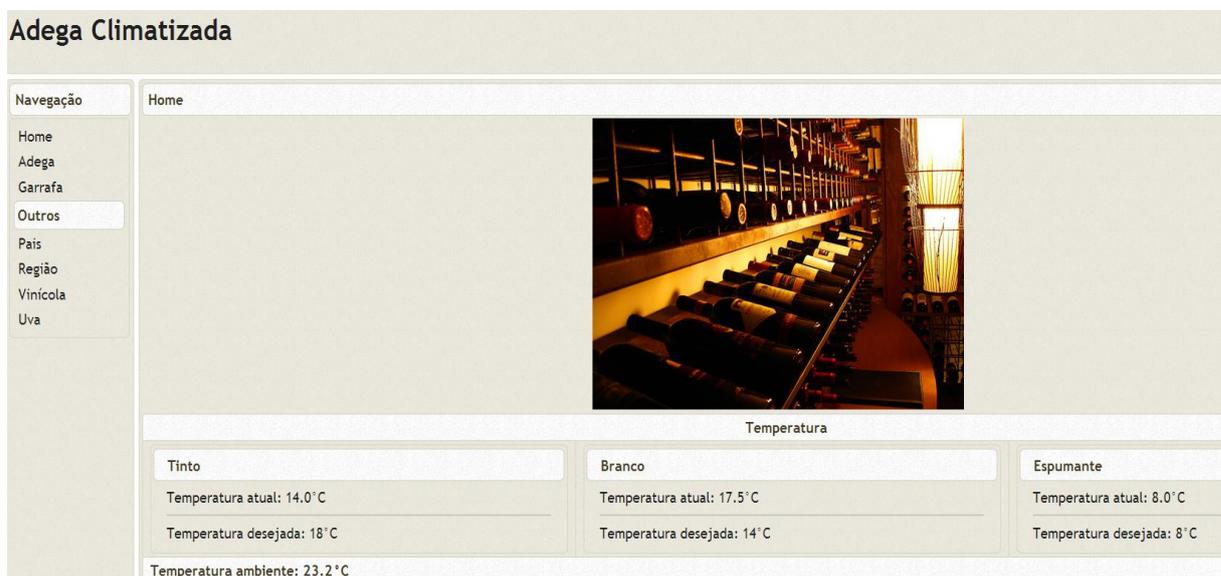
O Xhtml se comunica com o Java utilizando uma linguagem de extensão, onde a camada de visão aciona a camada de controle para enviar e receber dados do modelo. Foi utilizado a camada de interface, a qual faz requisições a camada Bean.

O servidor de aplicação utilizado nesse projeto é o Tomcat Apache que é um *container Web* de código fonte aberto, é baseado em Java e foi criado para executar aplicações *Web*.

### 4.3.1 Descrição das Telas do Sistema

A aplicação faz, de forma simples, o cadastramento e localização dos vinhos, bem como o controle de temperatura de cada um dos três compartimentos do protótipo.

Na interface é possível cadastrar (criar/incluir) um vinho, pesquisar (ler) os vinhos existentes na adega, atualizar os dados cadastrais do vinho escolhido e excluir um vinho específico. Na Figura 4.11 é mostrada a tela inicial da aplicação de cadastramento e controle de temperatura.



**Figura 4.11 - Tela Inicial (Autor: Ayrton Gissoni)**

A tela inicial possui um menu “Navegação”, onde o usuário poderá entrar na tela “Garrafa” e cadastrar, no menu “Inserir”, ou procurar um vinho através dos filtros disponíveis na tela Garrafa. Ao clicar na “ADEGA” ou na foto da adega, o usuário visualizará um esquema de como estão guardados os vinhos em sua adega.

Na Figura 4.12 é mostrado a tela de cadastramento de vinhos, onde o usuário poderá cadastrar o vinho que vai ser colocado em sua adega.



A imagem mostra uma interface web para o cadastro de uma garrafa de vinho. O formulário é intitulado "Inserir Garrafa" e contém os seguintes campos:

- Nome: Texto "Bueno Paralelo"
- Safra: Texto "2010"
- Tipo: Menu suspenso com "Tinto" selecionado
- Uva: Menu suspenso com "Cabernet Sauvignon" selecionado
- Teor Alcoólico: Campo numérico com "13.5 %" e setas de ajuste
- Potencial de Guarda: Campo numérico com "5 ano(s)" e setas de ajuste
- Amadurecimento: Área de texto com "12 meses em barricas de carvalho franceses"
- País: Menu suspenso com "Brasil" selecionado
- Região: Menu suspenso com "Campanha" selecionado
- Vinícola: Menu suspenso com "Selecione a vinícola" selecionado

Na base do formulário, há dois botões: "Inserir" e "Limpar".

**Figura 4.12 - Tela Inserir Garrafa (Autor: Ayrton Gissoni)**

A tela de cadastramento permitirá o usuário cadastrar o vinho desejado, preenchendo as informações básicas sobre esse vinho, como: nome, ano, selecionando o tipo, o país, a região, selecionado a vinícola e a uva, preenchendo amadurecimento, potencial de guarda e teor alcoólico.

Na Figura 4.13 é mostrado a tela de pesquisa de vinhos, onde o usuário poderá procurar o seu vinho desejado.

Nome	Tipo	País	Ano	Uva	
Cave Geisse	Branco	Brasil	2009	Pinot Noir	[Lupa] [Excluir]
Miolo Reserva	Branco	Brasil	2012	Sauvignon Blanc	[Lupa] [Excluir]
Miolo Reserva	Tinto	Brasil	2011	Tempranillo	[Lupa] [Excluir]
Bueno Paralelo	Tinto	Brasil	2010	Cabernet Sauvignon	[Lupa] [Excluir]
Santa Rita 120	Branco	Chile	2011	Chardonnay	[Lupa] [Excluir]
Ventisquero Clássico	Branco	Chile	2012	Sauvignon Blanc	[Lupa] [Excluir]
Canepa Reserva Privata	Tinto	Chile	2010	Cabernet Sauvignon	[Lupa] [Excluir]
Casa Rivas	Tinto	Chile	2011	Carménère	[Lupa] [Excluir]
Sainte Croix du Mont	Branco	França	2007	Sauvignon Blanc	[Lupa] [Excluir]
Domaine Amiot Guy Bourgogne Rouge	Tinto	França	2010	Pinot Noir	[Lupa] [Excluir]

**Figura 4.13 - Tela Pesquisar Garrafa (Autor: Ayrton Gissoni)**

A tela de pesquisa permitirá o usuário encontrar o seu vinho desejado, para isso o usuário digita nos filtros os parâmetros relacionados ao vinho, até que seja encontrado e exibido na lista o vinho procurado.

Após encontrar o vinho desejado, o usuário poderá visualizá-lo apenas clicando em um botão em forma de lupa. Ao clicar, o vinho será visualizado conforme mostrado na Figura 4.14.

### Bueno Paralelo

Safra: 2010  
Tipo: Tinto

#### Elaboração

Uva: Cabernet Sauvignon  
Teor: 13.5  
Amadurecimento: 12 meses em barricas de carvalho francês

#### Terroir

País: Brasil  
Região: Campanha  
Vinícola: Miolo

Ok

**Figura 4.14 - Tela Visualizar Garrafa (Autor: Ayrton Gissoni)**

Se o usuário decidir por excluir o vinho de seu sistema de cadastro, basta clicar no botão em forma de lixeira, aparecerá uma mensagem de confirmação da exclusão, conforme a Figura 4.15 onde o usuário poderá escolher uma das opções: sim e não.



**Figura 4.15 - Tela de Exclusão (Autor: Ayrton Gissoni)**

Para atualizar o cadastro do vinho, basta o usuário clicar no menu "Alterar" na tela Garrafa e então aparecerá uma tela conforme a Figura 4.16 , onde usuário poderá alterar as informações sobre o vinho selecionado e cadastrado apenas clicando no botão "Visualizar" e reescrevendo-a novamente. Para confirmar a alteração, basta clicar no botão "ALTERAR".

**Alterar Garrafa**

Detalhe

Nome:  
Bueno Paralelo

Safra:  
2010

Tipo:  
Tinto

Uva:  
Cabernet Sauvignon

Teor Alcoólico:  
13.5 %

Potencial de Guarda:  
5 ano(s)

Amadurecimento:  
12 meses em barricas de carvalho francês

País:  
Brasil

**Figura 4.16 - Tela Alterar Garrafa (Autor: Ayrton Gissoni)**

A tela adega mostra ao usuário a disposição de seus vinhos em cada compartimento e permite também o ajuste e controle de temperatura de cada um. Para o controle de temperatura, o usuário poderá visualizar a temperatura de cada compartimento e alterá-la conforme desejado apenas pressionando o botão “+” e “-”. Na Figura 4.17 é mostrada a tela da adega.

Adega		
Tinto	Branco	Espumante
<p>Cadastrado</p> <p>Nome: Bueno Paralelo</p> <p>Safra: 2010</p> <p> </p>	<p>Cadastrado</p> <p>Nome: Sainte Croix du Mont</p> <p>Safra: 2007</p> <p> </p>	<p>Cadastrar</p> <p>Nome: _____</p> <p>Safra: _____</p> <p></p>
<p>Cadastrar</p> <p>Nome: _____</p> <p>Safra: _____</p> <p></p>	<p>Cadastrado</p> <p>Nome: Miolo Seleção Rosé</p> <p>Safra: 2010</p> <p> </p>	<p>Cadastrar</p> <p>Nome: _____</p> <p>Safra: _____</p> <p></p>
<p>Temperatura atual: 17°C  </p> <p>Temperatura desejada: 18°C</p>	<p>Temperatura atual: 13°C  </p> <p>Temperatura desejada: 12°C</p>	<p>Temperatura atual: 9°C  </p> <p>Temperatura desejada: 8°C</p>

**Figura 4.17 - Tela Adega Garrafa (Autor: Ayrton Gissoni)**

## CAPÍTULO 5 TESTES E RESULTADOS

Este capítulo apresenta uma descrição dos testes realizados no protótipo da adega climatizada, no *software* de controle de temperatura e no sistema de cadastramento dos vinhos, visando avaliar o desempenho e verificar os resultados obtidos com o desenvolvimento desse protótipo descrito no capítulo 3 e 4.

### 5.1 Testes Realizados para o Protótipo

Os testes realizados (os cenários de testes) são descritos no Apêndice B, os quais mostram a verificação da melhor maneira para se conseguir fazer a climatização por meio de da célula de Peltier que foi utilizando o dissipador maior para o lado quente, *coolers* em modo exaustor, a fim de dissipar com mais eficiência o calor da face quente da célula, fazendo com que ela conseguisse atingir menores temperaturas.

A maioria dos testes utilizou a célula de Peltier do modelo TEC1-12706, *coolers* e dissipadores fazendo um “sanduíche”, colocando a célula no meio de dois dissipadores com *cooler*. O circuito (célula mais *coolers*) foi ligado em paralelo a uma fonte de tensão de 12V e 10A e verificou-se que a célula consumia cerca de 4A de corrente.

O melhor resultado obtido foi no cenário 9, a uma temperatura de 16°C, onde foi testado um dos compartimentos do protótipo da adega climatizada, colocando nele duas garrafas de vinho. O teste foi realizado durante 10 horas e a temperatura de dentro do isopor diminuiu e atingiu o equilíbrio em seus 9°C evidenciando que calor estava sendo dissipado de forma eficiente, a caixa de isopor estava fazendo a correta isolamento do ambiente e as garrafas de vinho atingiram a temperatura adequada (9°C) dentro de uma variabilidade máxima de 3°C.

### 5.2 Teste do *Software* de Cadastramento dos vinhos

O teste foi realizado com vários vinhos, entre eles o vinho Bueno Paralelo, que foi inserido no sistema de cadastro através da interface da aplicação,

posteriormente foi verificado que este vinho foi inserido devidamente no banco de dados, conforme mostrado na Figura 5.1.

idGarrafa	nomeGarrafa	anoGarrafa	tipoGarrafa	amadurecimentoG	potencialGarrafa	teorGarrafa	idUva	idVinicola
1	Cave Geisse	2009	Branco	Método tradiciona	2	12.5	11	22
2	Miolo Reserva	2012	Branco	Em tanques de aç	3	13.5	13	23
3	Miolo Reserva	2011	Tinto	Em barricas de cai	3	13.5	12	23
4	Bueno Paralelo	2010	Tinto	12 meses em barr	5	13.5	7	23
5	Santa Rita 120	2011	Branco	Em tanques de aç	3	14	14	24
6	Ventisquero Clássico	2012	Branco	Em tanques de aç	2	12	13	25
7	Canepa Reserva Privata	2010	Tinto	10 meses em barr	5	13.5	7	26
8	Casa Rivas	2011	Tinto	50% em barricas d	3	13.6	8	27

Figura 5.1 - Inserir Vinho no Banco (Autor: Ayrton Gissoni)

Na tela “Garrafas”, foi feita a pesquisa preenchendo primeiramente o filtro “Ano”, então a pesquisa retornou diversos vinhos com o ano 2010, depois preencheu-se o filtro “Tipo” com o tipo tinto, novamente a pesquisa retorno vinhos tintos do ano 2010, e por último, preencheu o filtro “Nome” com o nome “bueno”. Na Figura 5.2 é mostrado o resultado da pesquisa.

### Adega Climatizada

**Menu**

- Principal
- Home
- Adega
- Cadastrar
- Pesquisar
- Informações Complementares

**Pesquisar**

Nome	Tipo	Ano	Região	
<input type="text" value="bueno"/>	<input type="text" value="tinto"/>	<input type="text" value="2010"/>	<input type="text"/>	
Bueno Paralelo	Tinto	2010	Campanha	<input type="button" value="↻"/> <input type="button" value="✖"/> <input type="button" value="🗑"/>

Figura 5.2 - Pesquisa do Vinho (Autor: Ayrton Gissoni)

Foi realizada uma alteração no vinho, mudando o ano do vinho para 2011. Na Figura 5.3 é mostrado a atualização feita no banco de dados.

idGarrafa	nomeGarrafa	anoGarrafa	tipoGarrafa	amadurecimentoGarrafa	potencialGarrafa	teorGarrafa	idUva	idVinicola
1	Cave Geisse	2009	Branco	Método tradicional 28 mese		2 12.5	11	22
2	Miolo Reserva	2012	Branco	Em tanques de aço inox		3 13.5	13	23
3	Miolo Reserva	2011	Tinto	Em barricas de carvalho fran		3 13.5	12	23
4	Bueno Paralelo	2011	Tinto	12 meses em barricas de car		5 13.5	7	23
5	Santa Rita 120	2011	Branco	Em tanques de aço inoxidáv		3 14	14	24

Figura 5.3 - Alteração do Vinho no BD (Autor: Ayrton Gissoni)

Na tela Adega o vinho foi incluído clicando no botão “+” no compartimento dos tintos, conforme mostrado na Figura 5.4.

Figura 5.4 - Vinho Inserido na Adega (Autor: Ayrton Gissoni)

Posteriormente, na tela “Garrafas”, foi excluído o vinho clicando no botão em forma de lixeira e confirmando a exclusão. Na Figura 5.5 é mostrado que o cadastro do vinho foi excluído do banco de dados.

idGarrafa	nomeGarrafa	anoGarrafa	tipoGarrafa	amadurecimentoGarrafa	potencialGarrafa	teorGarrafa	idUva	idVinicola
1	Cave Geisse	2009	Branco	Método tradicional 28 mese	2	12.5	11	22
2	Miolo Reserva	2012	Branco	Em tanques de aço inox	3	13.5	13	23
3	Miolo Reserva	2011	Tinto	Em barricas de carvalho fran	3	13.5	12	23
5	Santa Rita 120	2011	Branco	Em tanques de aço inoxidáv	3	14	14	24

Figura 5.5 - Exclusão do Vinho no BD (Autor: Ayrton Gissoni)

### **5.3 Dificuldades Encontradas / Soluções**

Esta seção retrata as principais dificuldades e as soluções encontradas ao longo do desenvolvimento do protótipo de adega climatizada e do *software* de cadastramento de vinhos.

#### **5.3.1 Corrente elétrica**

Uma limitação existente no projeto é a corrente elétrica, pois se a corrente total da adega climatizada for muito “grande”, impacta no local de instalação desta adega, que deve fornecer a corrente elétrica necessária para alimentar a célula de Peltier, a qual necessita de uma corrente de 4A (sendo a corrente máxima de 6A).

Para que os compartimentos possam ter temperaturas diferentes, eles devem ser ligados de forma isolada uns dos outros. Para o projeto, cada compartimento foi ligado em paralelo com a fonte, necessitando uma corrente de 12A (sendo a máxima da 18A).

Como solução, foram colocadas duas fontes, cada uma de 10A e juntas fornecem cerca de 20A. A primeira fonte é responsável pelos compartimentos 1 e 2 e a segunda fonte pelo compartimento 3.

#### **5.3.2 Eficiência da Célula de Peltier**

Uma das maiores dificuldades deste projeto foi conseguir uma boa eficiência para a climatização de cada compartimento utilizando a célula de Peltier, pois esta pastilha termoelétrica aquece com muita facilidade o seu lado quente e se não dissipado o calor de forma correta, o calor acaba esquentando o lado frio da pastilha e perdendo toda a sua eficiência, fazendo com que ao invés de esfriar, esquente.

Foram feitos vários testes para verificar a melhor eficiência da célula de Peltier e a solução foi colocar um dissipador maior e mais robusto para o lado quente, pois desta forma o calor era dissipado de forma correta. Além disso, para

melhorar a eficiência, foram colocados dois *coolers* em modo exaustor, pois dessa forma o calor flui na direção do vento.

### 5.3.3 Ruídos no Sensor LM35

Um dos problemas encontrados no projeto foram os ruídos externos recebidos pelo sensor LM35, pois esses ruídos interferem muito na leitura da temperatura de cada compartimento e faz com que o Arduino tome decisões erradas no acionamento do sistema de refrigeração.

Uma das soluções para amenizar o ruído do sensor, foi o uso de cabo blindado manga ANTEX 4X26 AWG, o qual faz o transporte de dados entre o Arduino e o sensor LM35.

Um outro problema é que o sensor marca temperaturas negativas quando os três compartimentos estão ativados. Para evitar esse problema, foi colocado um resistor de 10K Ohms entre a saída Vout do LM35 e o GND do Arduino.

### 5.3.4 Quantidade de Beans

Um dos problemas encontrados no desenvolvimento do sistema de cadastro dos vinhos foi a relação entre a quantidade de Beans da camada de controle em função da quantidade de páginas da camada de visão, pois na utilização de apenas um Bean para todas as páginas, foi constatado um aumento na complexidade de manutenção do código e com o passar do tempo a página ficava cada vez mais pesada, uma vez que a quantidade de manipuladores de evento (ações/requisições que o usuário faz a interface – visão) controlados por um único Bean, aumenta proporcionalmente ao número de componentes (botões, formulários, tabelas, filtros e etc...) das páginas mantidas por ela.

A solução foi criar um Bean para cada página *Web*, o que resultou em um código mais limpo e fácil de manter em caso de alterações na página ou até mesmo na resolução de erros relacionados a esta, como também ao carregar uma determinada página, só era carregado o controlador desta página.

### 5.3.5 Escopo dos Beans

Outro assunto que interferiu bastante no desenvolvimento da aplicação foi na definição do escopo dos Beans. O escopo pode ser visto como a forma pela qual os dados na aplicação ficam disponíveis e visíveis para o usuário.

A escolha de um escopo para um Bean pode impactar profundamente no desempenho e até mesmo em seu funcionamento. Portanto um estudo aprofundado e uma atenção maior foram empreendidos para que os escopos utilizados na aplicação fossem escolhidos corretamente, que no caso foi utilizado o ViewScoped, onde o tempo de vida do Bean, de cada página, permanece ativo apenas enquanto a página estiver aberta.

### 5.3.6 Componentes do *Framework*

Um dos problemas que ocorreram durante o desenvolvimento da aplicação foi relacionado à implementação de alguns componentes do *Framework* PrimeFaces.

O componente que apresentou mais dificuldade e problemas em termos de implementação foi o Dialog, componente que mostra uma painel suspenso na página mostrando mensagens ao usuário, semelhantemente ao Alert do JavaScript.

Vários erros na página “Garrafas” foram ocasionados por parte deste componente, como por exemplo, ao alterar uma garrafa um Dialog era chamado para mostrar o objeto de uma garrafa escolhida, mas os dados que apareciam eram de uma garrafa que fora anteriormente selecionada, entre outros erros, principalmente relacionados às operações de inserir e alterar.

Como solução a estes problemas o CRUD foi dividido em mais duas páginas, uma para a inserção de uma nova garrafa, e outra para alteração da garrafa.

### 5.3.7 Comunicação Java e Arduino

Uma das dificuldades encontradas foi a comunicação entre o *software* feito em Java e o microcontrolador Arduino, pois o Java envia ao Arduino a temperatura de cada compartimento selecionada pelo usuário (temperatura desejada) e recebe do Arduino a temperatura de cada compartimento (temperatura atual) que é monitorada pelo sensor LM35.

A dificuldade está em utilizar o mesmo canal (interface Serial) para transportar essas informações. Como solução para este problema foi utilizado a interface serial, com a biblioteca RXTXcomm.jar a qual reconhece a porta serial do Arduino para o envio das informações, e foi desenvolvido um projeto separado, contendo a classe em Java que faz a comunicação com o Arduino que implementa todo o protocolo de informações.

Para se fazer a diferenciação entre os dados enviado pelo Java e os dados enviados pelo Arduino, foi desenvolvido um *parser* que interpreta a temperatura lida pelo sensor e enviada do Arduino para o Java, onde juntamente com o envio da temperatura, é enviado uma espécie de “código” como por exemplo: “a: valor da temperatura” para temperatura ambiente, “t: valor da temperatura” para vinhos tintos do compartimento 1, “b: valor da temperatura” para vinhos brancos do compartimento 2, “e: valor da temperatura” para vinhos espumantes do compartimento 3.

## 5.4 Resultados Obtidos

### 5.4.1 Temperatura dos Compartimentos

Para avaliar os resultados alcançados, foi feito um teste com duração de 9 horas, onde se coletava a cada 10 minutos as medidas das temperaturas de cada um dos compartimentos. Para a coleta foi utilizado um termômetro em graus Celsius, pois o termômetro não leva interferência de ruídos externos e possui uma maior precisão que o sensor de temperatura LM35.

O teste foi iniciado no dia 12 de junho de 2013, às 12 horas e se estendeu até às 21 horas. A temperatura ambiente inicial do teste foi de 25°C marcada às 12 horas e a temperatura ambiente final do teste foi de 21°C, marcada ao final do teste às 21 horas. No Arduino foi carregado um programa que acionava cada um dos compartimentos até que se atingisse a temperatura mínima de armazenamento para cada tipo de vinho.

Para o primeiro compartimento, o sistema de climatização iniciou-se em modo ligado, pois a temperatura de 25°C era maior que a temperatura mínima para vinhos tintos (14°C). Ao decorrer de 9 Horas, foram atingidos os 14°C. Para a geração do gráfico, foi incluído um outro teste, que mostra que quando a temperatura do compartimento 1 é atingida, o Arduino manda um comando para o módulo relé, o qual desliga o sistema de climatização. Ao decorrer do tempo, o compartimento aquece, aumentando sua temperatura até se chegar em 18°C (temperatura máxima para vinhos tintos), então o Arduino manda um outro comando para o módulo relé, o qual liga o sistema de climatização, fazendo a temperatura baixar novamente para os 14°C em cerca de 1 hora. Na Figura 5.6 é mostrado o comportamento da temperatura para o compartimento 1.

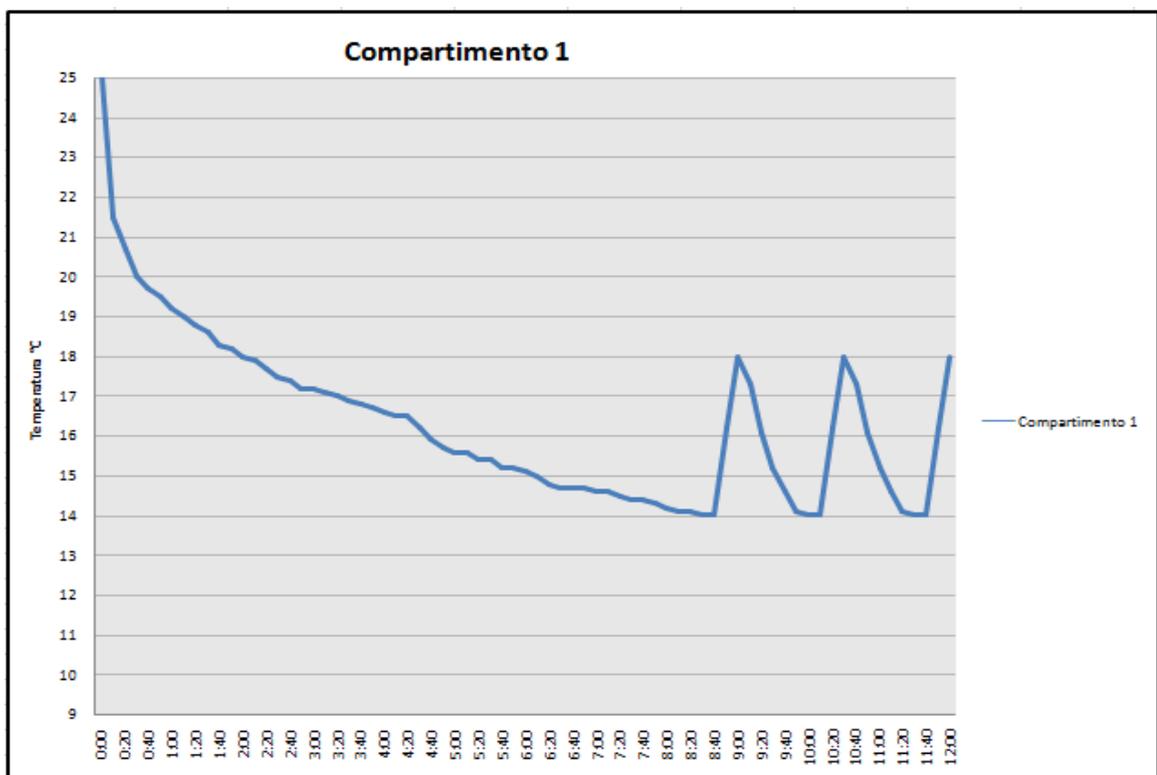
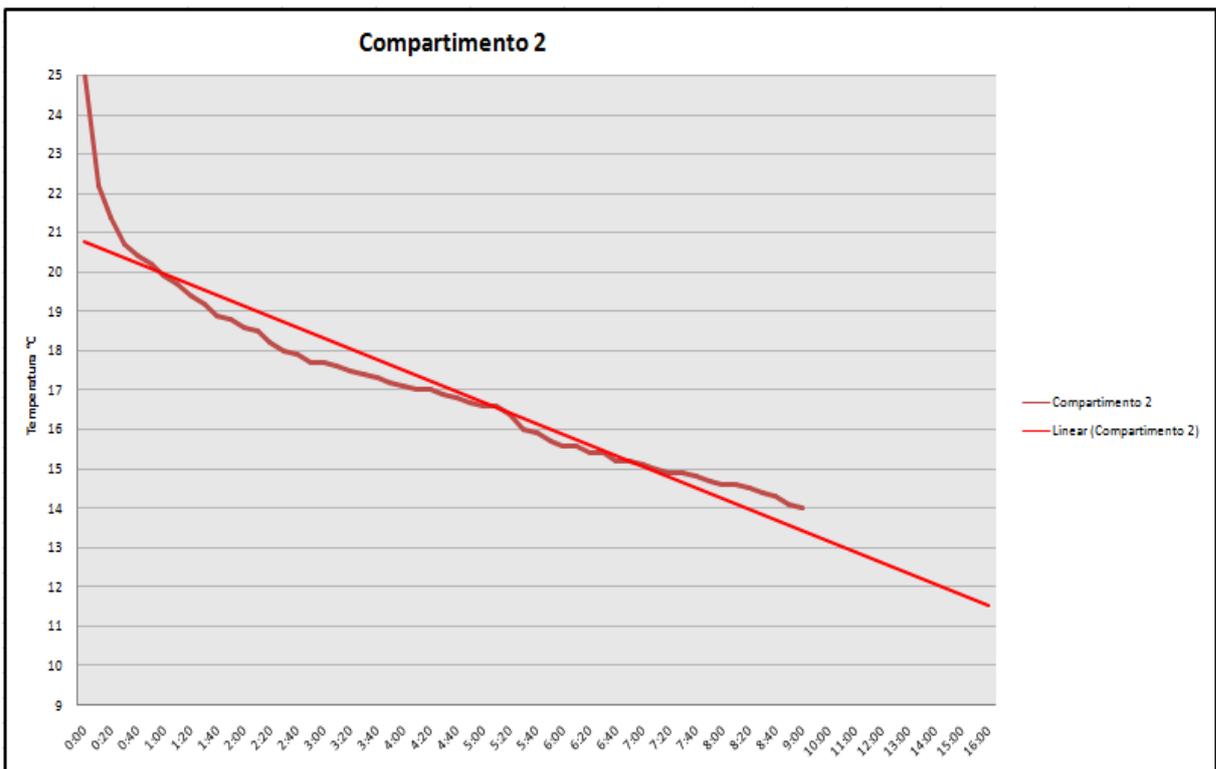


Figura 5.6 - Temperatura do Compartimento 1 (Autor: Ayrton Gissoni)

Para o segundo compartimento, o sistema de climatização iniciou-se em modo ligado, pois a temperatura de 25°C era maior que a temperatura mínima para vinhos brancos (8°C). Ao decorrer de 9 Horas, foram atingidos os 14°C. Para a geração do gráfico foi feito uma linha de tendência linear (utilizando o Excel) que mostra o comportamento da temperatura do compartimento 2, tendo como base os dados coletados, e informa uma provável tendência de temperatura. Na Figura 5.7 é mostrado o comportamento da temperatura para o compartimento 2.



**Figura 5.7 - Temperatura do Compartimento 2 (Autor: Ayrton Gissoni)**

Para o terceiro compartimento, o sistema de climatização iniciou-se em modo ligado, pois a temperatura de 25°C era maior que a temperatura mínima para vinhos espumantes (6°C). Ao decorrer de 9 Horas, foi atingido os 14°C. Para a geração do gráfico foi feito uma linha de tendência polinomial (utilizando o Excel) que mostra o comportamento da temperatura do compartimento 3, tendo como base os dados coletados, e informa uma provável tendência de temperatura. Na Figura 5.8 é mostrado o comportamento da temperatura para o compartimento 3.

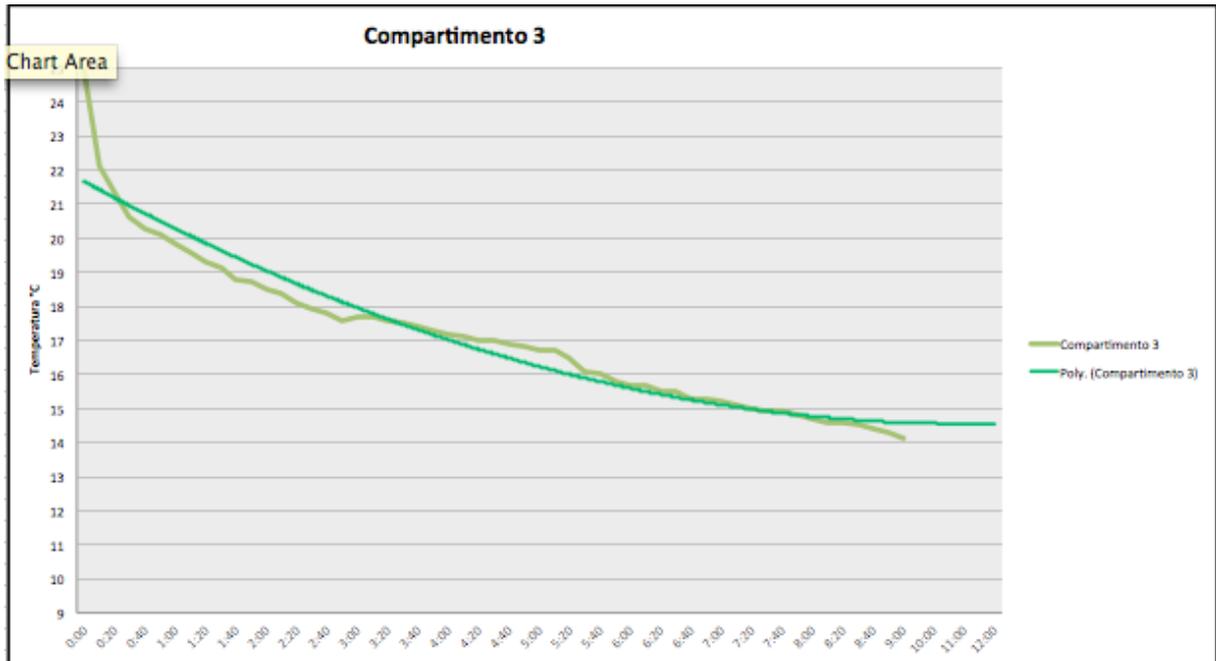


Figura 5.8 - Temperatura do Compartimento 3 (Autor: Ayrton Gissoni)

#### 5.4.2 Protótipo

Com o desenvolvimento deste trabalho foi obtido um protótipo de adega climatizada de três compartimentos e que possui um controle de temperatura automatizada que é feito utilizando o Arduino, módulo relé e sensores de temperatura. Na Figura 5.9 é mostrado a parte posterior do protótipo, onde se tem todo o circuito elétrico da adega climatizada (Fonte, Arduino, Relé, Protoboard) e o sistema de climatização (*coolers* e dissipadores do lado quente).

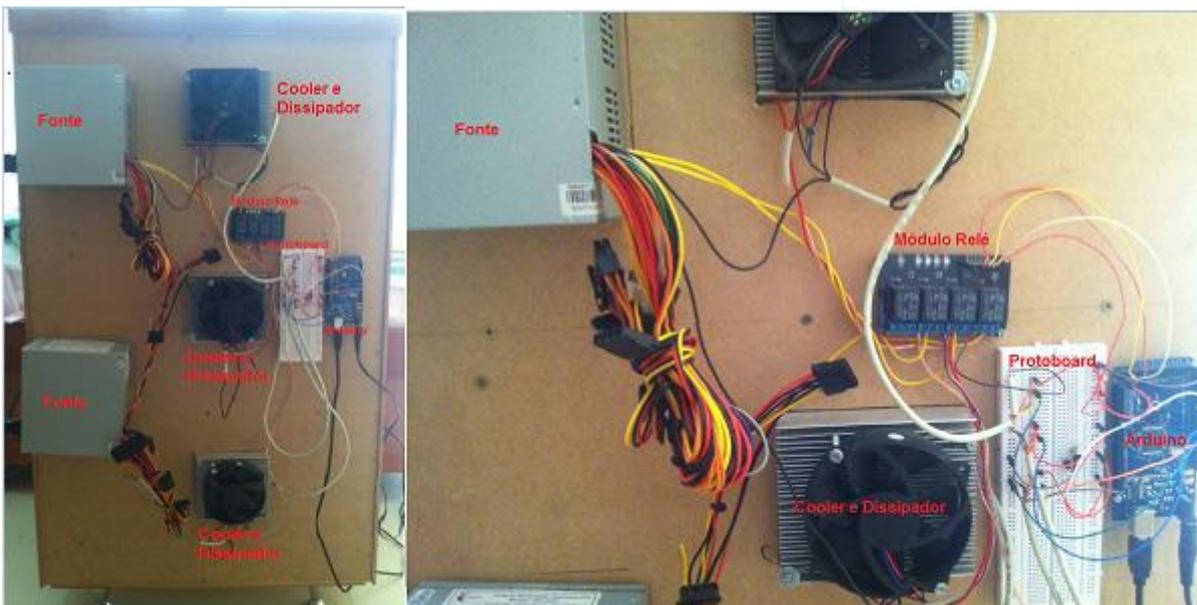


Figura 5.9 - Vista Posterior do Protótipo (Autor: Ayrton Gissoni)

Na Figura 5.10 é mostrado o protótipo visto de frente, onde temos os três compartimentos. Dentro de cada compartimento temos duas garrafas de vinho, um sensor de temperatura e um *cooler* com dissipador do lado quente.



Figura 5.10 - Vista Frontal do Protótipo (Autor: Ayrton Gissoni)

## 5.5 Produto Gerado

O produto gerado com este projeto foi uma adega climatizada para uso pessoal que utiliza como tecnologia para a climatização a célula de Peltier. A adega possui 3 compartimentos cada um podendo guardar 2 garrafas, sendo o primeiro compartimento destinado aos tintos, o segundo aos brancos e o terceiro aos espumantes.

A temperatura mínima atingida na adega é de 9°C após um tempo de aproximadamente de 10 horas, com uma temperatura externa de 16°C. A adega climatizada necessita de uma corrente de 12A, funciona com entrada de 220V. A Tabela 5 informa os principais materiais necessários para montagem do protótipo e seus respectivos custos.

**Tabela 5 - Materiais/Custos**

Materiais	Quantidade	Custo R\$	Total
Célula de Peltier	3	30	90
Dissipador Quente	3	60	180
Cooler com dissipador Frio	6	35	210
Arduíno	1	130	130
Módulo Relé	1	30	30
Sensor de Temperatura	4	4	16
Fonte	2	50	100
Caixa de Isopor 21L	3	26	78
Folha de Isopor 40mm	3	9	27
Móvel da adega	1	500	500
<b>TOTAL (R\$)</b>			<b>1361</b>

**Autor: Ayrton Gissoni**

## CAPÍTULO 6 CONSIDERAÇÕES FINAIS

Este capítulo trata das considerações finais a respeito deste do protótipo desenvolvido ao longo deste projeto, bem como apresenta propostas para futuros trabalhos nessa área.

### 6.1 Conclusão

Nesse trabalho foi desenvolvido uma adega climatizada que possui três compartimentos para os diferentes tipos de vinhos e que utiliza como base para a climatização a célula de Peltier. Também foi desenvolvido um *software* para cadastro dos vinhos e controle de temperatura.

Com os testes realizados na adega climatizada para verificar a temperatura atingida em cada compartimento, conclui-se que o objetivo geral foi atingido em grande parte, pois a temperatura é controlada de forma automatizada, permitindo que o usuário possa inserir a temperatura desejada para os principais tipos de vinhos. A temperatura mínima atingida pela adega (tendo como temperatura externa 16°C) foi de 9°C após 10 horas de funcionamento, estando 3°C acima da temperatura mínima do compartimento 3 e implicando que a temperatura foi atingida para uma variabilidade máxima de mais ou menos 3°C. Porém, para uma temperatura externa superior a 21°C, a temperatura mínima do compartimento 2 e do compartimento 3 não foi atingida para um tempo de 9 horas de teste.

O objetivo específico de projetar um *hardware* capaz de controlar a temperatura de cada compartimento da adega via célula de Peltier, foi parcialmente atingido, visto que para temperaturas externas superiores a 21°C, a célula de Peltier não funcionou tão bem, quanto para climatizar o compartimento 3 a uma temperatura de 6°C.

O *hardware* utilizando o microcontrolador Arduino como sendo o responsável por enviar comandos e controlar a temperatura e os demais dispositivos, atingiu totalmente o objetivo para o qual foi designado, pois o Arduino além de ser de fácil instalação, atendeu completamente as funções do protótipo e ainda possui um grande número de portas, as quais podem ser adicionadas outros

componentes para projetos futuros. O módulo relé, a fonte, e os demais componentes do *hardware* (com exceção do sensor LM35 e da célula de Peltier), funcionaram conforme o esperado.

O sensor LM35, mesmo após várias modificações e adaptações, como a utilização de cabo blindado, a colocação de resistores para diminuir o ruído externo e o cálculo das médias de leitura, que foram modificações que melhoraram a leitura do sensor. O sensor de temperatura LM35 não funcionou como esperado, pois o sensor sofre algumas oscilações na leitura da temperatura, fazendo com que não possua uma boa precisão para este projeto.

A tecnologia da célula de Peltier não possui uma boa eficiência para climatizar três compartimentos, pois a corrente necessária para cada módulo é alta, tornando a corrente total da adega climatizada mais alta ainda, atingindo cerca de 12A, limitando a sua instalação somente para os ambientes que forneçam esta corrente necessária.

Para que houvesse uma melhora na climatização de cada compartimento, possibilitando que fosse atingida temperaturas mais baixas e em um menor tempo, teria que ser colocado mais módulos de Peltier (células de Peltier com *coolers* e dissipadores), porém esta solução é inviabilizada pela limitação da corrente elétrica e seu custo, pois no caso da corrente elétrica, se passássemos a dobrar a quantidade de módulos de 3 para 6, ter-se-ia uma corrente elétrica de 24A, a qual não é disponível em todos os ambientes.

Verificou-se que a teoria divergiu da prática em dois pontos neste projeto. O primeiro foi o dimensionamento da célula de Peltier, o qual foi dimensionado teoricamente para climatizar a adega em 1 hora, e na prática mostrou que mesmo obedecendo a todos os cuidados exigidos pelo fabricante na sua instalação, a célula de Peltier não obteve os mesmos resultados esperados que na teoria. O segundo ponto foi o do sensor LM35, que teoricamente possui uma boa precisão, o que, na prática, não foi constatado.

A adega climatizada de três compartimentos utilizando a célula de Peltier não é viável, pois consegue armazenar poucas garrafas (duas garrafas por compartimento) e o custo-benefício de se armazenar vinhos diferentes, se torna

muito caro e ainda se tem a limitação de não se atingir a temperatura adequada para os vinhos brancos e espumantes, dado uma temperatura externa de 21°C.

Em Brasília, para onde o projeto foi dimensionado, de dia a temperatura ambiente varia entre 21°C a 25°C e somente de noite a temperatura fica abaixo dos 20°C, chegando muitas vezes aos 16°C, então durante o dia a temperatura da adega ficaria entre os seus 14°C e de noite chegaria a 9°C.

A tecnologia da célula de Peltier é viável para um único compartimento, com um único tipo de vinho, como os tintos que são os mais comuns e sua faixa de temperatura mais elevada que a dos outros vinhos, podendo utilizar dois módulos de Peltier para climatizar cerca de 8 garrafas na adega, atingindo uma temperatura máxima de 14°C.

Apesar de o *software* de cadastro de vinhos não ser o foco e nem o ponto chave deste projeto, sendo que esse sistema de cadastro foi feito visando unicamente aumentar as funcionalidades da adega climatizada. Este *software* após vários testes e implementações, funcionou conforme o esperado e possibilitou o cadastramento dos vinhos.

A linguagem Java se comunicou perfeitamente com o banco de dados MySQL por meio do *drive* JDBC, possibilitando a camada de modelo fazer a persistência dos dados referentes ao cadastro dos vinhos.

A camada de controle se comunicou e garantiu a comunicação entre as atualizações do modelo e visão, mesmo tendo-se dificuldades na construção do Bean que gerencia cada uma das visões e administra o modelo.

A camada de visão utilizando da linguagem Xhtml e o *Framework* PrimeFaces facilitou a construção da interface *Web*, pois o *Framework* já possui diversos componentes e temas que facilitam na construção do *layout* da aplicação.

## 6.2 Proposta para Futuros Projetos

Utilização de outro sensor de temperatura diferente do LM35 e que possua uma maior precisão, como por exemplo, o sensor SHT11, que é um sensor capaz de medir a temperatura e umidade relativa do ar, é extremamente compacto e

já vem calibrado. O SHT11 é um dispositivo de dimensões mínimas, alta precisão e baixo consumo. Se comunica com outros sistemas digitais sendo implementado na maioria dos microcontroladores.

Uma boa modificação no projeto seria desenvolver a aplicação de cadastro de vinho já existente, melhorando suas funcionalidades, para dispositivos móveis como os celulares e *tablets*, fazendo por exemplo, um aplicativo para *Iphone* ou *Android*.

Seria uma boa opção a inclusão da comunicação via rede sem fio para que o aplicativo para dispositivos móveis possa se conectar com o Arduino, para que o controle de temperatura possa ser feito de qualquer lugar da sua residência.

Outra ideia interessante é a de se aprimorar o sistema de cadastro de vinhos, para que esse possa rodar na *Web* tendo múltiplos usuários, cada um com sua adega virtual, criando um sistema de autenticação de usuários com senhas.

## REFERÊNCIAS

ARDUINO. Disponível em < Fonte: [www.Arduino.cc/](http://www.Arduino.cc/)>. Acesso em: abril de 2013

BENNETT.C.O; MYERS.J.E. **Fenômenos de Transporte: Quantidade de Movimento, Calor e Massa**. 2 ed. São Paulo: McgrawHil 2008.

COSTA, Vânia Aparecida da. **Estudo das propriedades termoelétricas de fios quânticos de telureto de chumbo**. Dissertação de Mestrado, Pós-Graduação em Engenharia e Tecnologia Espaciais/Ciência e Tecnologia de Materiais e Sensores, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, 2010. Orientada por: SILVA, Erasmo Assumpção de Andrade e.

HB ELETRONIC COMPONENTS. Disponível em: <<http://www.hebeiltd.com.cn/?p=z.peltier.pricelist>>. Acesso em: abril de 2013

LEPRÉ, Georges. **Grande Larousse do Vinho**. 1 ed. São Paulo: Lafonte, 2012.

MARQUES, Marco Antônio da Silva. **Mini refrigerador baseado em Efeito Peltier acionado por controle remoto infravermelho**. Monografia, Faculdade de Tecnologia e Ciências Aplicadas, Centro Universitário de Brasília, Brasília, DF, 2011. Orientado por: FARIAS, Maria Marony.

MOURA, José Américo de Sousa. **Filmes Nanométricos de FeN e AlN crescidos por sputtering e aplicações do Efeito Peltier** .Dissertação de Doutorado, Departamento de Física Teórica e Experimental, Universidade Federal do Rio Grande do Norte, Natal, RN, 2010. Orientado por: FEITOSA, Carlos Chesman de Araújo.

RAMALHO, NICOLAU, TOLEDO. **Os Fundamentos da Física 2**. 9 ed. São Paulo: Moderna 2007

SANTOS, Suzamara. **Pequeno livro do vinho: Guia para toda hora**. 1 ed. São Paulo: Verus, 2006.

SOUZA, Daniel de Oliveira. **Condicionador de ar para cabine de caminhão, utilizando Células de Peltier**. Monografia, Faculdade de Tecnologia e Ciências Aplicadas, Centro Universitário de Brasília, Brasília, DF, 2011. Orientado por: FARIAS, Maria Marony.

STRAZZA, Fabricio. RIBERI, Rodrigo Midusauskas. **Projeto de uma micro-adeega climatizada de baixo custo**. Trabalho de conclusão de curso, Escola Politécnica, Universidade de São Paulo, São Paulo, SP, 2004. Orientados por: FIORELLI, Flávio Augusto Sanzovo.

TAPIA, Patricio. **Vinho sem segredos**. 1 ed. São Paulo: Planeta, 2006.

TECNOLOGIA em Peltier. Disponível em: <<http://www.peltier.com.br>>. Acesso em: abril de 2013

TIPLER, Paul. **Física para Cientistas e Engenheiros: Mecânica, Oscilações, Ondas e Termodinâmica - volume I**. 5 ed. LOCAL: LTC, 2009

## APÊNDICE - A

Neste Apêndice A é mostrado o código utilizado no Arduino.

```
const int sensortemperaturaA = 0; // definir pino  
const int sensortemperatura1 = 2; // definir pino  
const int sensortemperatura2 = 4; // definir pino  
const int sensortemperatura3 = 6; // definir pino
```

```
const int compartimento1 = 22; // definir pino  
const int compartimento2 = 24; // definir pino  
const int compartimento3 = 26; // definir pino
```

```
int temperaturaA = 0; // iniciar variavel em = 0  
int temperatura1 = 0; // iniciar variavel em = 0  
int temperatura2 = 0; // iniciar variavel em = 0  
int temperatura3 = 0; // iniciar variavel em = 0
```

```
int mediatemperaturaA = 0; // iniciar variavel em = 0  
int mediatemperatura1 = 0; // iniciar variavel em = 0  
int mediatemperatura2 = 0; // iniciar variavel em = 0  
int mediatemperatura3 = 0; // iniciar variavel em = 0
```

```
int somatemperaturaA = 0; // iniciar variavel em = 0  
int somatemperatura1 = 0; // iniciar variavel em = 0  
int somatemperatura2 = 0; // iniciar variavel em = 0  
int somatemperatura3 = 0; // iniciar variavel em = 0
```

```
int contadorA = 0;// iniciar variavel em = 0
int contador1 = 0;// iniciar variavel em = 0
int contador2 = 0;// iniciar variavel em = 0
int contador3 = 0;// iniciar variavel em = 0

int somarTemperatura(int temperatura1, int temperatura2) //Soma temperatura e
coloca no acumulador "somatemperatura"
{
    return (temperatura1 + temperatura2);
}

int converterTemperatura(int temperatura)//Converte a temperatura
{
    return (500 * temperatura) / 1023;
}

int calcularMedia(int temperatura, int contador) //Calcula a media
{
    return (temperatura/contador);
}

void setup()
{
    Serial.begin(9600); //taxa bauds que e a velocidade da serial
    pinMode(compartimento1, OUTPUT); // configura a porta digital 22
    pinMode(compartimento2, OUTPUT); // configura a porta digital 24
    pinMode(compartimento3, OUTPUT); // configura a porta digital 26
```

```
}
```

```
//Inicializa as temperaturas desejadas
```

```
int temperaturaDesejada1 = 18;
```

```
int temperaturaDesejada2 = 12;
```

```
int temperaturaDesejada3 = 8;
```

```
void loop()
```

```
{
```

```
////////////////////// COMUNICACAO ARDUINO ////////////////////////
```

```
    if (Serial.available() >= 3) { // Avalia e retorna a quantidade de caracteres
disponiveis vindos da serial
```

```
        temperaturaDesejada1 = Serial.read();
```

```
        Serial.println(temperaturaDesejada1);
```

```
        temperaturaDesejada2 = Serial.read();
```

```
        Serial.println(temperaturaDesejada2);
```

```
        temperaturaDesejada3 = Serial.read();
```

```
        Serial.println(temperaturaDesejada3);
```

```
    }
```

```
    // mandar para o arduino a temperatura atual de cada compartimento e
receber do arduino a temperatura desejada pela serial
```

```
////////////////////// AMBIENTE ////////////////////////
```

```
    temperaturaA = converterTemperatura(analogRead(sensorTemperaturaA)); //
converter temperatura
```

```

    if (temperaturaA > 0){ //Verificar e tratar temperatura
        contadorA++;
        somatemperaturaA = somarTemperatura(somatemperaturaA,
temperaturaA);
        mediatemperaturaA = calcularMedia(somatemperaturaA, contadorA);
    }

    //erial.print("Temperatura Ambiente: "); //Exibir Temperatura
Serial.print("a:");
Serial.println(mediatemperaturaA);

//Serial.println("");
delay(500);

////////////////////// COMPARTIMENTO 1 ////////////////////////
temperatura1 = converterTemperatura(analogRead(sensortemperatura1));

if (temperatura1 > 0){ //Verificar e tratar temperatura
    if (contador1 >= 5){
        contador1 = 0;
        somatemperatura1 = 0;
    }
    contador1++;
    somatemperatura1 = somarTemperatura(somatemperatura1,
temperatura1);
    mediatemperatura1 = calcularMedia(somatemperatura1, contador1);
}

```

```

//Serial.print("Temperatura do Compartimento 1: "); //Exibir Temperatura
Serial.print("t:");

Serial.println(mediatemperatura1);

Serial.flush(); //Forca o envio pela serial, limpando o buffer

//Serial.println("");

//Desativa e ativa o rele - sistema de climatizacao
if (mediatemperatura1 <= temperaturaDesejada1){
    digitalWrite(compartimento1,LOW);
}
if (mediatemperatura1 > temperaturaDesejada1){
    digitalWrite(compartimento1,HIGH);
}

delay(500);

//////////////////// COMPARTIMENTO 2 //////////////////////
temperatura2 = converterTemperatura(analogRead(sensortemperatura2));

if (temperatura2 > 0){ //Verificar e tratar temperatura
    if (contador2 >= 5){
        contador2 = 0;
        somatemperatura2 = 0;
    }
    contador2++;
    somatemperatura2 = somarTemperatura(somatemperatura2,
temperatura2);
    mediatemperatura2 = calcularMedia(somatemperatura2, contador2);

```

```

}

//Serial.print("Temperatura do Compartimento 2: "); //Exibir Temperatura
Serial.print("b:");
  Serial.println(mediatemperatura2);
Serial.flush();//Forca o envio pela serial, limpando o buffer
  //Serial.println("");

//Desativa e ativa o rele - sistema de climatizacao
  if (mediatemperatura2 <= temperaturaDesejada2){
    digitalWrite(compartimento2,LOW);
  }
  if (mediatemperatura2 > temperaturaDesejada2){
    digitalWrite(compartimento2,HIGH);
  }
  delay(500);

//////////////////// COMPARTIMENTO 3 //////////////////////
temperatura3 = converterTemperatura(analogRead(sensortemperatura3));

if (temperatura3 > 0){ //Verificar e tratar temperatura
  if (contador3 >= 5){
    contador3 = 0;
    somatemperatura3 = 0;
  }
  contador3++;
}

```

```
        somatemperatura3 = somarTemperatura(somatemperatura3,
temperatura3);

        mediatemperatura3 = calcularMedia(somatemperatura3, contador3);
    }

    //Serial.print("Temperatura do Compartimento 3: "); //Exibir Temperatura
Serial.print("e:");

    Serial.println(mediatemperatura3);

Serial.flush();//Forca o envio pela serial, limpando o buffer

    //Serial.println("");

//Desativa e ativa o rele - sistema de climatizacao
    if (mediatemperatura3 <= temperaturaDesejada3){
        digitalWrite(compartimento3,LOW);
    }

    if (mediatemperatura3 > temperaturaDesejada3){
        digitalWrite(compartimento3,HIGH);
    }

    delay(500);

    //Serial.println("");

}
```

## APÊNDICE - B

Neste Apêndice B são descritos os cenários de testes, realizados no decorrer do projeto, para a verificação da melhor maneira para se conseguir fazer a climatização por meio de da célula de Peltier

### Cenário 1

Neste cenário, a uma temperatura de 21°C, foi feito o teste da célula de Peltier do modelo TEC1-12706 ligada sozinha a uma fonte de tensão de 12V, a célula necessitou de uma corrente de 2,83A. Logo nos primeiros segundos a célula esquentou rapidamente de um lado e esfriou de outro. Poucos segundos depois foi notado que a parte fria começou a esquentar e logo estava em temperatura elevada, ocasionando em pouco tempo, cerca de menos de 1 minuto a abertura do circuito e a queima da célula, pois o calor da face quente não foi dissipado corretamente e entrando em equilíbrio térmico passou o calor para a face fria. Na Figura 1 é mostrado o cenário do teste 1:

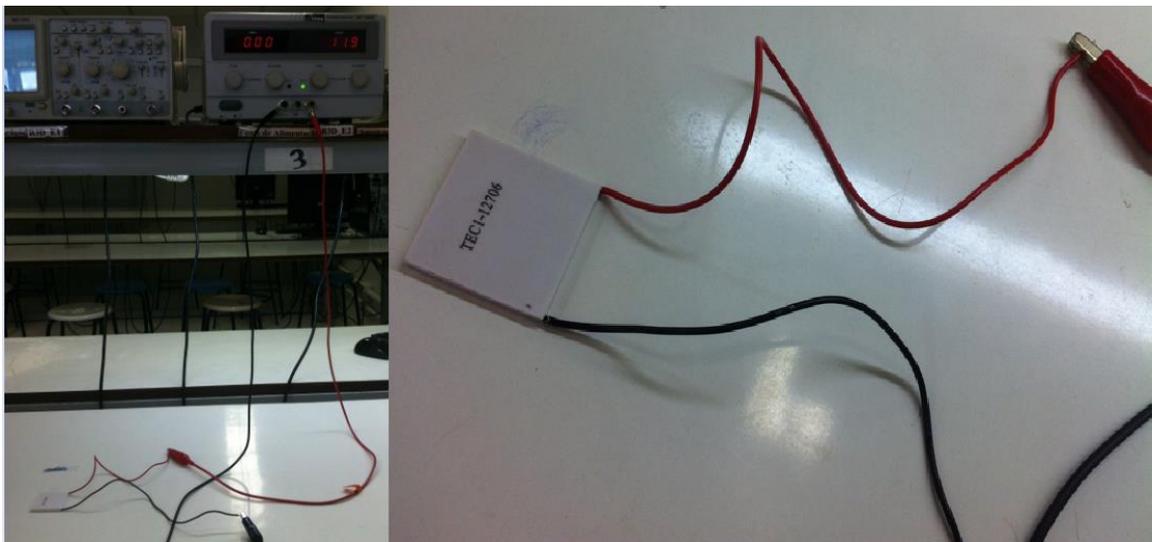


Figura 1 Apêndice B - Cenário 1 (Autor: Ayrton Gissoni)

## Cenário 2

Neste cenário, a uma temperatura de 21°C, foi feito o teste da célula de Peltier do modelo TEC1-12706, colocando a face quente em contato com o dissipador com *cooler*, para que o calor da face quente fosse dissipado e não ocorre-se a queima da célula de Peltier como no cenário 1.

O circuito (célula mais *cooler*) foi ligado em paralelo a uma fonte de tensão de 12V e 3A. Logo nos primeiros segundos verificou que a face fria esfriava, porém com o passar do tempo (Cerca de dois minutos) a temperatura da face fria foi subindo, evidenciando que o dissipador da marca Multilaser não estava conseguindo dissipar o calor.

## Cenário 3

Neste cenário, a uma temperatura de 21°C, foi feito o teste da célula de Peltier do modelo TEC1-12706, colocando a face quente em contato com o dissipador com *cooler* da marca Multilaser, dessa vez assoprando o ar para fora (funcionando como um exaustor) para que o calor da face quente fosse dissipado e não ocorre-se a queima da célula de Peltier como no cenário 1 e nem o aumento da temperatura da face fria como no cenário 2.

O circuito (célula mais *cooler*) foi ligado em paralelo a uma fonte de tensão de 12V e 3A. Logo nos primeiros segundos verificou que a face fria esfriava e com o passar do tempo temperatura da face fria se estabilizou em 16,1°C evidenciando o calor não estava sendo dissipado de forma eficiente.

## Cenário 4

Neste cenário, a uma temperatura de 21°C, foi feito o teste da célula de Peltier do modelo TEC1-12706, fazendo um “sanduíche”, colocando a célula no meio de dois dissipadores com *cooler* da marca Multilaser. O teste foi feito com os *coolers* soprando em cima da célula. Foi colocado o módulo (*cooler*, dissipador e célula) em

um ambiente termicamente isolado (caixa de isopor de 3L) para verificar melhor a temperatura.

Foi feito o corte no isopor (do tamanho do dissipador frio) de modo que a célula de Peltier ficasse no meio do isopor, o *cooler* da face quente fora do isopor e o *cooler* da face fria dentro do isopor.

O circuito (célula mais *coolers*) foi ligado em paralelo a uma fonte de tensão de 12V e 3A. Logo nos primeiros não se verificou alteração de temperatura. O teste foi realizado durante 30 minutos e a temperatura de dentro do isopor aumentou, ficando em 32,22°C evidenciando o calor não estava sendo dissipado de forma eficiente e ainda estava entrando na caixa de isopor.

## **Cenário 5**

Neste cenário, a uma temperatura de 21°C, foi feito o teste da célula de Peltier do modelo TEC1-12706, fazendo um “sanduíche”, colocando a célula no meio de dois dissipadores com *cooler* da marca Multilaser. O teste foi feito com os *coolers* soprando para fora no modo exaustor. Foi colocado o módulo (*cooler*, dissipador e célula) em um ambiente termicamente isolado (caixa de isopor de 3L) para verificar melhor a temperatura igual no cenário 5.

O circuito (célula mais *coolers*) foi ligado em paralelo a uma fonte de tensão de 12V e 3A. Logo nos primeiros não se verificou alteração de temperatura. O teste foi realizado durante 30 minutos e a temperatura de dentro do isopor aumentou, ficando em 29,44°C evidenciando o calor não estava sendo dissipado de forma eficiente e ainda estava entrando na caixa de isopor.

## **Cenário 6**

Neste cenário, a uma temperatura de 28,9°C, em ambiente aberto foi feito o teste da célula de Peltier do modelo TEC1-12706, colocando a face quente em contato com um dissipador maior (dissipador da face quente) e de outro modelo,

colocando o *cooler* em modo exaustor e em contato com esse dissipador, para que o calor da face quente fosse dissipado de forma eficiente.

O circuito (célula mais *cooler*) foi ligado em paralelo a uma fonte de tensão de 12V e que alimentou a célula com uma corrente pouco acima de 4A. Logo nos primeiros segundos verificou que a face fria esfriava e com o passar do tempo (Cerca de cinco minutos) a temperatura da face fria atingiu 1,7°C evidenciando que o dissipador com o *cooler* em modo exaustor estava conseguindo dissipar bem o calor.

## **Cenário 7**

Neste cenário, a uma temperatura de 21°C, em ambiente isolado termicamente em uma caixa de isopor de 3L, foi feito o teste da célula de Peltier do modelo TEC1-12706, fazendo um “sanduíche”, colocando a célula no meio de dois dissipadores com *cooler*. O teste foi feito com os *coolers* soprando para fora no modo exaustor. O dissipador externo (do lado quente) foi o dissipador maior. O módulo foi colocado no fundo da caixa de isopor (em baixo da caixa) e foi colocado em contato com a mesa, pois ainda não havia parafuso correto para prender um dissipador no outro.

O circuito (célula mais *coolers*) foi ligado em paralelo a uma fonte de tensão de 12V e que alimentou a célula com uma corrente pouco acima de 4A. Logo nos primeiros segundos verificou diminuição pequena de temperatura. O teste foi realizado durante 30 minutos e a temperatura de dentro do isopor diminuiu e atingiu os 9°C evidenciando que calor estava sendo dissipado de forma eficiente e a caixa de isopor estava fazendo a correta isolação do ambiente.

Na Figura 2 é mostrado o cenário do teste 8, o termômetro do multímetro marca 9°C e o termômetro em Fahrenheit marca 42°F que equivale a 5,6°C.

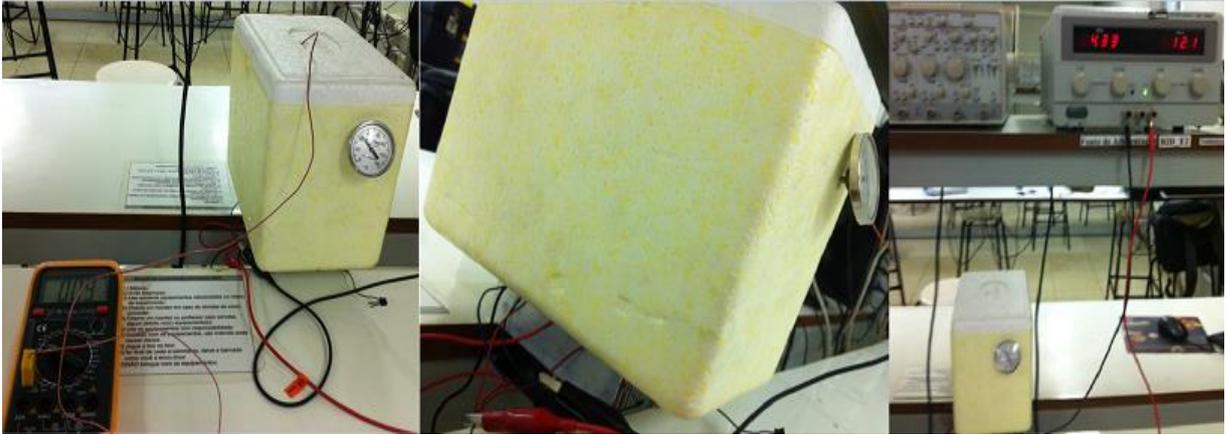


Figura 2 Apêndice B - Cenário 7(Autor: Ayrton Gissoni)

### Cenário 8

Neste cenário, a uma temperatura de 30°C, em ambiente isolado termicamente em uma caixa de isopor de 21L, foi feito o teste da célula de Peltier do modelo TEC1-12706, fazendo um “sanduíche”, colocando a célula no meio de dois dissipadores com *cooler*. O teste foi feito com os *coolers* soprando para fora no modo exaustor. O dissipador externo (do lado quente) foi o dissipador maior. O módulo foi preso (por parafuso) no fundo da caixa de parafuso.

O circuito (célula mais *coolers*) foi ligado em paralelo a uma fonte de tensão de 12V e que alimentou a célula com uma corrente de 4A. O teste foi realizado durante 1 hora e a temperatura de dentro do isopor diminuiu e atingiu os 16,5°C evidenciando que calor estava sendo dissipado de forma eficiente, porém, a caixa de isopor possuía um volume muito grande (21L), fazendo com que a temperatura de 6°C não fosse atingida.

### Cenário 9

Neste cenário, a uma temperatura de 16°C, foi testado um dos compartimentos do protótipo da adega climatizada, colocando nele duas garrafas de vinho. O circuito (célula mais *coolers*) foi ligado em paralelo a uma fonte de tensão de 12V e que alimentou a célula com uma corrente de 4A. O teste foi realizado durante 10 horas e a temperatura de dentro do isopor diminuiu e atingiu o equilíbrio em seus 9°C evidenciando que calor estava sendo dissipado de forma eficiente, a

caixa de isopor estava fazendo a correta isolação do ambiente e as garrafas de vinho atingiram a temperatura adequada (9°C) dentro de uma variabilidade máxima de 3°C.

## **APÊNDICE – C**

Este Apêndice C possui o código do software de cadastramento de vinhos e a classe utilizada para comunicação entre o Java e o Arduino.

## ANEXO – A

**Tabela 1 Anexo A - Modelos de Células de Peltier do Fabricante Hebeiltd**

Número	Descrição
TEC1-01708	15x15x3.7mm 8.5A/2.0V 9.5W 17 couples
TEC1-01708S	Sealed 15x15x3.7mm 8.5A/2.0V 9.5W 17 couples
TEC1-03108	20x20x3.3mm 8.5A/3.7V 17.6W 31 couples
TEC1-04908	25x25x3.7mm 8.5A/5.9V 27.4W 49 couples
TEC1-07103	30x30x4.9mm 3.3A/8.5V 18W 71 couples
TEC1-07103HTS	High temp. Seal. 30x30x4.9mm 3.3A/8.5V 18W 71 couples
TEC1-07108	30x30x3.8mm 8.5A/8.5V 40W 71 couples
TEC1-07108S	Sealed 30x30x3.8mm 8.5A/8.5V 40W 71 couples
TEC1-07108HT	High temp. 30x30x3.8mm 8.5A/8.5V 40W 71 couples
TES1-12703	30x30x3.7mm 3.2A/15.4V 30W 127 couples
TES1-12703S	Sealed 30x30x3.7mm 3.2A/15.4V 30W 127 couples
TES1-12704	30x30x3.3mm 4.3A/15.4V 40W 127 couples
TES1-12704S	Sealed 30x30x3.3mm 4.3A/15.4V 40W 127 couples
TEC1-12703	40x40x4.9mm 3.3A/15.4V 38W 127 couples
TEC1-12703S	Sealed 40x40x4.9mm 3.3A/15.4V 38W 127 couples
TEC1-12703HT	High temp. 40x40x4.9mm 3.3A/15.4V 38W 127 couples
TEC1-12703HTS	High temp. Seal. 40x40x4.9mm 3.3A/15.4V 38W 127 couples
TEC1-12705	40x40x4.2mm 5.3A/15.4V 57W 127 couples
TEC1-12705S	Sealed 40x40x4.2mm 5.3A/15.4V 57W 127 couples
TEC1-12705HT	High temp. 40x40x4.2mm 5.3A/15.4V 57W 127 couples
TEC1-12706	40x40x3.8mm 6.4A/15.4V 63W 127 couples
TEC1-12706S	Sealed 40x40x3.8mm 6.4A/15.4V 63W 127 couples
TEC1-12706HTS	High temp. Seal. 40x40x3.8mm 6.4A/15.4V 63W 127 couples
TEC1-12707	40x40x3.7mm 7.4A/15.4V 75W 127 couples
TEC1-12707HT	High temp. 40x40x3.7mm 7.4A/15.4V 75W 127 couples
TEC1-12707HTS	High temp. Seal. 40x40x3.7mm 7.4A/15.4V 75W 127 couples
TEC1-12708	40x40x3.5mm 8.5A/15.4V 85W 127 couples
TEC1-12708S	Sealed 40x40x3.5mm 8.5A/15.4V 85W 127 couples
TEC1-12708HTS	High temp. Seal. 40x40x3.5mm 8.5A/15.4V 85W 127 couples
TEC1-12710	40x40x3.3mm 10.5A/15.4V 100W 127 couples
TEC1-12710S	Sealed 40x40x3.3mm 10.5A/15.4V 100W 127 couples
TEC1-12710HT	High temp. 40x40x3.3mm 10.5A/15.4V 100W 127 couples
TEC1-12710HTS	High temp. Seal. 40x40x3.3mm 10.5A/15.4V 100W 127 couples
TEC1-12715	50x50x4.3mm 15.6A/15.4V 150W 127 couples
TEC1-12715S	Sealed 50x50x4.3mm 15.6A/15.4V 150W 127 couples
TEC1-12715HT	High temp. 50x50x4.3mm 15.6A/15.4V 150W 127 couples

TEC1-12715HTS	High temp. Seal. 50x50x4.3mm 15.6A/15.4V 150W 127 couples
TEC1-12730	62x62x4.8mm 30.7A/15.4V 350W 127 couples
TEC1-12730S	Sealed 62x62x4.8mm 30.7A/15.4V 350W 127 couples
TEC1-12730HT	High temp. 62x62x4.8mm 30.7A/15.4V 350W 127 couples
TEC1-12730HTS	High temp. Seal. 62x62x4.8mm 30.7A/15.4V 350W 127 couples
TEC1-26316	High temp. 50x50x3.1mm 16A/31.5V 300W 263 couples

<http://www.hebeiltd.com.cn/?p=z.peltier.pricelist>

## ANEXO – B



# Technical Specification

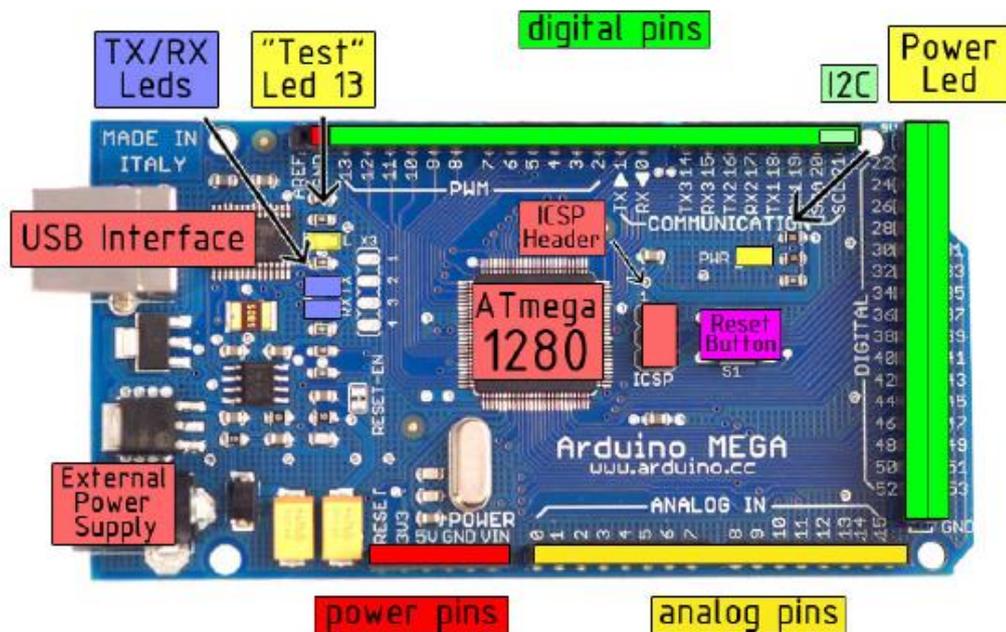


EAGLE files: [arduino-mega-reference-design.zip](#) Schematic: [arduino-mega-schematic.pdf](#)

## Summary

ATmega1280	ATmega1280
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	128 KB of which 4 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

## the board



radiospares

RADIONICS



## Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board FTDI chip. Maximum current draw is 50 mA.
- **GND.** Ground pins.

## Memory

The ATmega1280 has 128 KB of flash memory for storing code (of which 4 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

## Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.
- **External Interrupts:** 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM:** 0 to 13. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI:** 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Decimila.
- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I<sup>2</sup>C:** 20 (SDA) and 21 (SCL). Support I<sup>2</sup>C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I<sup>2</sup>C pins on the Duemilanove or Decimila.

The Mega has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

## Communication

The Arduino Mega has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega1280 provides four hardware UARTs for TTL (5V) serial communication. An FTDI FT232RL on the board channels one of these over USB and the [FTDI drivers](#) (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega's digital pins.

The ATmega1280 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. To use the SPI communication, please see the ATmega1280 datasheet.

## Programming

The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega1280 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

## Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega1280 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

## USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

## Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. Please note that I<sup>2</sup>C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).



## How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

### Linux Install

### Windows Install

### Mac Install

Once you have downloaded/unzipped the arduino IDE, you'll need to install the FTDI Drivers to let your PC talk to the board. First **Plug the Arduino to your PC via USB cable**.

### Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>  
Arduino-0017>Examples>  
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select MEGA

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.

```

Blink | Arduino 0017
File Edit Sketch Tools Help
[Icons]
Blink.g
int ledPin = 13; // LED connected to digital pin 13
// The setup() method runs once, when the sketch starts
void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power
void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000); // wait for a second
}
  
```



Done compiling

Press Compile button  
(to check for errors)



Upload



TX RX Flashing



Blinking Led!



## ANEXO - C



November 2000

## LM35 Precision Centigrade Temperature Sensors

### General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of  $\pm 1/4^\circ\text{C}$  at room temperature and  $\pm 3/4^\circ\text{C}$  over a full  $-55$  to  $+150^\circ\text{C}$  temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only  $60\ \mu\text{A}$  from its supply, it has very low self-heating, less than  $0.1^\circ\text{C}$  in still air. The LM35 is rated to operate over a  $-55$  to  $+150^\circ\text{C}$  temperature range, while the LM35C is rated for a  $-40$  to  $+110^\circ\text{C}$  range ( $-10$  with improved accuracy). The LM35 series is available pack-

aged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

### Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear  $+10.0\ \text{mV}/^\circ\text{C}$  scale factor
- $0.5^\circ\text{C}$  accuracy guaranteeable (at  $+25^\circ\text{C}$ )
- Rated for full  $-55$  to  $+150^\circ\text{C}$  range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than  $60\ \mu\text{A}$  current drain
- Low self-heating,  $0.08^\circ\text{C}$  in still air
- Nonlinearity only  $\pm 1/4^\circ\text{C}$  typical
- Low impedance output,  $0.1\ \Omega$  for  $1\ \text{mA}$  load

### Typical Applications

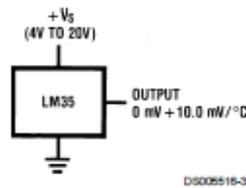
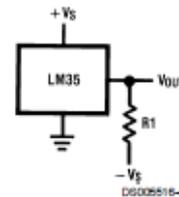


FIGURE 1. Basic Centigrade Temperature Sensor  
( $+2^\circ\text{C}$  to  $+150^\circ\text{C}$ )

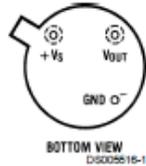


Choose  $R_1 = -V_S/50\ \mu\text{A}$   
 $V_{\text{OUT}} = +1,500\ \text{mV}$  at  $+150^\circ\text{C}$   
 $= +250\ \text{mV}$  at  $+25^\circ\text{C}$   
 $= -550\ \text{mV}$  at  $-55^\circ\text{C}$

FIGURE 2. Full-Range Centigrade Temperature Sensor

## Connection Diagrams

**TO-46  
Metal Can Package\***



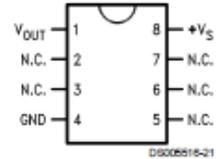
**BOTTOM VIEW**  
D5008816-1

\*Case is connected to negative pin (GND)

Order Number LM35H, LM35AH, LM35CH, LM35CAH or LM35DH

See NS Package Number H03H

**SO-8  
Small Outline Molded Package**



D5008816-21

N.C. = No Connection

**Top View**  
Order Number LM35DM  
See NS Package Number M08A

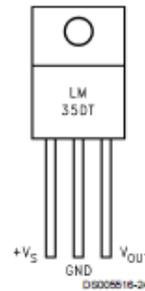
**TO-92  
Plastic Package**



**BOTTOM VIEW**  
D5008816-2

Order Number LM35CZ,  
LM35CAZ or LM35DZ  
See NS Package Number Z03A

**TO-220  
Plastic Package\***



D5008816-24

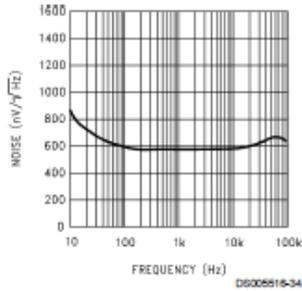
\*Tab is connected to the negative pin (GND).

Note: The LM35DT pinout is different than the discontinued LM35DP.

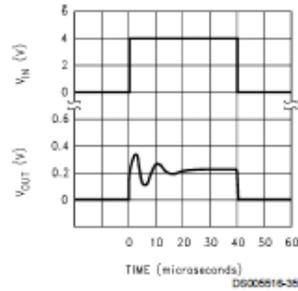
Order Number LM35DT  
See NS Package Number TA03F

Typical Performance Characteristics (Continued)

Noise Voltage



Start-Up Response



Applications

The LM35 can be applied easily in the same way as other integrated-circuit temperature sensors. It can be glued or cemented to a surface and its temperature will be within about 0.01°C of the surface temperature.

This presumes that the ambient air temperature is almost the same as the surface temperature; if the air temperature were much higher or lower than the surface temperature, the actual temperature of the LM35 die would be at an intermediate temperature between the surface temperature and the air temperature. This is especially true for the TO-92 plastic package, where the copper leads are the principal thermal path to carry heat into the device, so its temperature might be closer to the air temperature than to the surface temperature.

To minimize this problem, be sure that the wiring to the LM35, as it leaves the device, is held at the same temperature as the surface of interest. The easiest way to do this is to cover up these wires with a bead of epoxy which will insure that the leads and wires are all at the same temperature as the surface, and that the LM35 die's temperature will not be affected by the air temperature.

The TO-46 metal package can also be soldered to a metal surface or pipe without damage. Of course, in that case the V- terminal of the circuit will be grounded to that metal. Alternatively, the LM35 can be mounted inside a sealed-end metal tube, and can then be dipped into a bath or screwed into a threaded hole in a tank. As with any IC, the LM35 and accompanying wiring and circuits must be kept insulated and dry, to avoid leakage and corrosion. This is especially true if the circuit may operate at cold temperatures where condensation can occur. Printed-circuit coatings and varnishes such as Humiseal and epoxy paints or dips are often used to insure that moisture cannot corrode the LM35 or its connections.

These devices are sometimes soldered to a small light-weight heat fin, to decrease the thermal time constant and speed up the response in slowly-moving air. On the other hand, a small thermal mass may be added to the sensor, to give the steadiest reading despite small deviations in the air temperature.

Temperature Rise of LM35 Due To Self-heating (Thermal Resistance,  $\theta_{JA}$ )

	TO-46, no heat sink	TO-46*, small heat fin	TO-92, no heat sink	TO-92*, small heat fin	SO-8 no heat sink	SO-8** small heat fin	TO-220 no heat sink
Still air	400°C/W	100°C/W	180°C/W	140°C/W	220°C/W	110°C/W	90°C/W
Moving air	100°C/W	40°C/W	90°C/W	70°C/W	105°C/W	90°C/W	25°C/W
Still oil	100°C/W	40°C/W	90°C/W	70°C/W			
Stirred oil	50°C/W	30°C/W	45°C/W	40°C/W			
(Clamped to metal, infinite heat sink)		(24°C/W)				(55°C/W)	

\*Wakefield type 201, or 1" disc of 0.020" sheet brass, soldered to case, or similar.

\*\*TO-92 and SO-8 packages glued and leads soldered to 1" square of 1/16" printed circuit board with 2 oz. foil or similar.

**ANEXO - D**

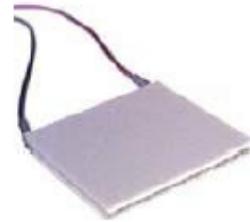


**Thermoelectric  
Cooler**

**TEC1-12706**

**Performance Specifications**

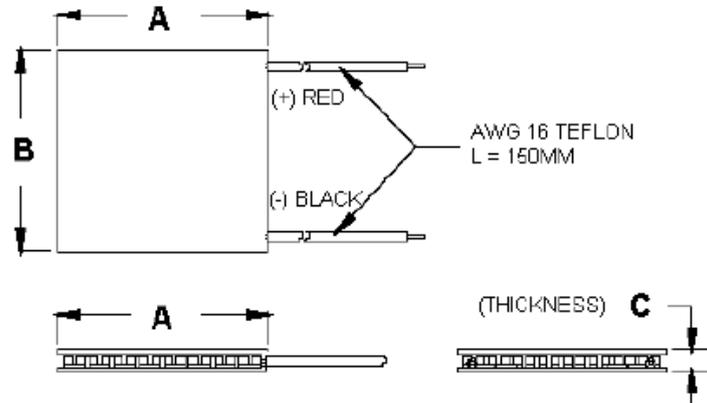
Hot Side Temperature (° C)	25° C	50° C
Qmax (Watts)	50	57
Delta Tmax (° C)	66	75
I <sub>max</sub> (Amps)	6.4	6.4
V <sub>max</sub> (Volts)	14.4	16.4
Module Resistance (Ohms)	1.98	2.30



Performance curves on page 2



**TEC1-12706**



Ceramic Material: Alumina (Al<sub>2</sub>O<sub>3</sub>)  
 Solder Construction: 138°C, Bismuth Tin (BiSn)

**Size table:**

A	B	C			
40	40	3.8			

**Operating Tips**

- Max. Operating Temperature: 138°C
- Do not exceed I<sub>max</sub> or V<sub>max</sub> when operating module.
- Life expectancy: 200,000 hours
- 
- Please consult HB for moisture protection options (sealing).
- Failure rate based on long time testings: 0.2%.