

Amilcar Joel Simm

***Proposta de Ferramenta Web para o Suporte ao
Ensino***

São José – SC

agosto / 2017

Amilcar Joel Simm

***Proposta de Ferramenta Web para o Suporte ao
Ensino***

Monografia apresentada à Coordenação do
Curso Superior de Tecnologia em Sistemas
de Telecomunicações do Instituto Federal de
Santa Catarina para a obtenção do diploma de
Tecnólogo em Sistemas de Telecomunicações.

Orientador:

Prof. Roberto Wanderley da Nóbrega, Dr.

CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS DE TELECOMUNICAÇÕES
INSTITUTO FEDERAL DE SANTA CATARINA

São José – SC

agosto / 2017

Monografia sob o título “*Proposta de Ferramenta Web para o Suporte ao Ensino*”, defendida por Amilcar Joel Simm e aprovada em 18 de agosto de 2017, em São José, Santa Catarina, pela banca examinadora assim constituída:

Prof. Roberto Wanderley da Nóbrega, Dr.
Orientador

Prof. Deise Monquelate Arndt, Me.
IFSC

Prof. Ederson Torresini, Me.
IFSC

Feliz aquele que transfere o que sabe e aprende o que ensina.

Cora Coralina

Agradecimentos

Dedico meus sinceros agradecimentos para todos que me ajudaram a concluir este trabalho.

Difícil a todos nomear, mas no tempo da graduação, devo citar meus professores, por estarem partilhando o que sabem, e meus colegas, por estarem presentes e dispostos a dividir as suas questões e a receberem a cota das minhas.

Em especial, a Professora Deise Monquellate Arndt, que desde as etapas de escolha do tema para este trabalho incentivou o processo, ao Professor Ederson Torresini, que gentilmente auxiliou na etapa de hospedagem da ferramenta no servidor *web*, e também, a meu orientador, Professor Roberto Wanderley da Nóbrega, pelo apoio, paciência e persistência em todo o período que finalizou minha graduação.

Não posso deixar de lembrar, sem nomear, sob pena de injustamente esquecer alguém, a todos os meus amigos, presentes de Deus nesta encarnação.

Muito obrigado!

Resumo

Durante o tempo de graduação é frequente a referência, por parte dos discentes, à complexidade dos conteúdos a aprender. A necessidade de ferramental de apoio ao processo de aprendizagem é discutida e trabalhada, seja na forma de tutoria com alunos mais experientes, passando por uso de simuladores diversos através de *software* proprietários ou livres, e de cursos extras promovidos pelos próprios docentes, não só em período letivo como em horários alternativos, conforme necessidades verificadas ao longo dos semestres.

A proposta deste Trabalho de Conclusão de Curso é implementar um *software* educacional, que será ferramenta *web*, para dar suporte a aulas presenciais, que seguem paralelamente. Pretende-se utilizar planos de estudo com roteiros organizados em módulos, com objetivos gerais e específicos, incluindo subsídios textuais e ferramentas de simulação para possíveis exercícios de fixação de conteúdo.

Como ferramenta *web*, deverá dispor de material visualmente atraente, que possibilite um acesso amplo e unificado, não se descuidando de critérios de usabilidade. Com isso, o conteúdo independeria efetivamente de *software* proprietários e necessidade de alto processamento daqueles que acessarem o material, por se utilizar um modelo cliente-servidor.

Abstract

During the time spent in undergraduate courses, it is usual to hear from students that subjects are quite difficult to learn. The need for means to support the learning process is frequently discussed and provided either in the form of mentoring with more experienced students, the use of various simulators written with proprietary or free software, or extra courses organized by teachers themselves, not only during school time, but also in alternative hours according to demands identified over the semester.

In this coursework, we propose an implementation of an educational software covering topics in digital communications, in the form of a web tool supporting traditional classroom lessons occurring in parallel. We aim to use study plans with scripts organized into modules, with general and specific objectives, including textual subsidies and simulation tools which may help the solution of quizzes.

The implemented web tool should allow broad and unified access while also being visually attractive and careful with usability issues. In addition, the tool should be independent of proprietary software and should not need high-end processing power of those accessing the material, since it will employ a client–server model.

Sumário

Lista de Figuras

Lista de Tabelas

1	Introdução	p. 13
1.1	Objetivo do trabalho	p. 14
1.2	Motivação	p. 15
1.3	Organização do texto	p. 15
2	Software educacionais	p. 16
2.1	O processo de ensino-aprendizagem a partir da óptica dos sistemas complexos	p. 16
2.1.1	Sistemas complexos	p. 18
2.1.2	A sala de aula	p. 19
2.2	Metodologias	p. 20
2.3	Elaboração de conteúdo	p. 21
2.4	Usabilidade	p. 22
2.4.1	Navegação	p. 23
2.4.2	Simulador	p. 29
3	Desenvolvimento	p. 37
3.1	Tecnologias utilizadas no processo de desenvolvimento	p. 37
3.1.1	Breve descrição das tecnologias	p. 37
3.2	<i>Framework</i> Django	p. 39

3.2.1	Passos iniciais com o <i>framework</i> Django	p. 40
3.3	O simulador	p. 43
3.3.1	Estrutura do arquivo <i>simulador.py</i>	p. 43
3.4	Ambiente de produção	p. 47
4	Conclusões	p. 48
	Apêndice A – Passo a passo de instalação do sigsis	p. 50
	Apêndice B – Roteiros de Amostragem e Quantização	p. 53
B.1	Roteiro 0: Um modelo de Amostragem e Quantização	p. 55
B.1.1	Objetivo geral do roteiro	p. 55
B.1.2	Objetivos específicos do roteiro	p. 55
B.1.3	Subsídios do Roteiro	p. 55
B.1.4	Bibliografia do roteiro	p. 57
B.1.5	Exercícios do roteiro	p. 57
B.2	Roteiro 1: Amostragem	p. 58
B.2.1	Objetivo geral do roteiro	p. 58
B.2.2	Objetivos específicos do roteiro	p. 58
B.2.3	Subsídios do Roteiro	p. 58
B.2.4	Bibliografia do roteiro	p. 60
B.2.5	Exercícios do roteiro	p. 60
B.3	Roteiro 2: Teorema da amostragem	p. 61
B.3.1	Objetivo geral do roteiro	p. 61
B.3.2	Objetivos específicos do roteiro	p. 61
B.3.3	Subsídios do Roteiro	p. 61
B.3.4	Bibliografia do roteiro	p. 63
B.3.5	Exercícios do roteiro	p. 63

B.4	Roteiro 3: Quantização e Codificação	p. 64
B.4.1	Objetivo geral do roteiro	p. 64
B.4.2	Objetivos específicos do roteiro	p. 64
B.4.3	Subsídios do Roteiro	p. 64
B.4.4	Bibliografia do roteiro	p. 67
B.4.5	Exercícios do roteiro	p. 67
B.5	Roteiro 4: Quantização Uniforme	p. 69
B.5.1	Objetivo geral do roteiro	p. 69
B.5.2	Objetivos específicos do roteiro	p. 69
B.5.3	Subsídios do Roteiro	p. 69
B.5.4	Bibliografia do roteiro	p. 74
B.5.5	Exercícios do roteiro	p. 74
B.6	Roteiro 5: Quantização Não-Uniforme	p. 75
B.6.1	Objetivo geral do roteiro	p. 75
B.6.2	Objetivos específicos do roteiro	p. 75
B.6.3	Subsídios do Roteiro	p. 75
B.6.4	Bibliografia do roteiro	p. 77
B.6.5	Exercícios do roteiro	p. 77
B.7	Roteiro 6: Modulação por codificação de pulso – PCM	p. 78
B.7.1	Objetivo geral do roteiro	p. 78
B.7.2	Objetivos específicos do roteiro	p. 78
B.7.3	Subsídios do Roteiro	p. 78
B.7.4	Bibliografia do roteiro	p. 79
B.7.5	Exercícios do roteiro	p. 80

Referências Bibliográficas

Lista de Figuras

2.1	Modelo do cabeçalho e menus laterais da página <i>index</i> da ferramenta.	p. 24
2.2	Modelo do rodapé da página de roteiros da ferramenta.	p. 24
2.3	Interface de administração do usuário <i>admin</i>	p. 25
2.4	Interface de administração do usuário <i>user</i>	p. 25
2.5	Mensagens de erro na página de parametrização de módulo.	p. 28
2.6	Mensagens de erro na página de parametrização de roteiro duplicado.	p. 29
2.7	Renderização do simulador de Sinais e Transformada de Fourier.	p. 31
2.8	Renderização do simulador de Sinais, Transformada de Fourier e Amostras.	p. 31
2.9	Renderização do simulador do Quantizador Genérico.	p. 32
2.10	Renderização do simulador do Quantizador Uniforme.	p. 33
2.11	Ampliação da figura do simulador ao se posicionar o <i>mouse</i> sobre a mesma.	p. 34
2.12	Mensagem de erro de parametrização do início e fim do eixo X.	p. 35
2.13	Tratamento de erros de não plotagem (a) e de vetores demasiado grandes (b).	p. 36
2.14	Tratamentos de erros do campo <i>Níveis de Quantização</i>	p. 36
2.15	Mensagem de dado não aceito pelo campo.	p. 36
3.1	Visão da estrutura de diretórios e arquivos após a criação do projeto.	p. 40
3.2	Visão da estrutura de diretórios e arquivos após a criação do <i>app core</i>	p. 41
B.1	Modelo de Comunicação Digital.	p. 53
B.2	Processo de conversão analógico-digital e digital-analógico.	p. 56
B.3	Esquema gráfico do processo de amostragem e quantização.	p. 56
B.4	Esquema do processo de amostragem e reconstrução, associado a obtenção de amostras mediante um chaveamento.	p. 58

B.5	Representação de onda triangular e amostras.	p. 59
B.6	Sinal $M(f)$	p. 62
B.7	Esboço do espectro do sinal $m(t) = \text{sinc}(200t)$	p. 63
B.8	Modelo com níveis de quantização e seus limiares.	p. 65
B.9	Esquema do processo de quantização e codificação.	p. 66
B.10	Níveis, limiares e passo de quantização <i>mid-tread</i> e <i>mid-rise</i>	p. 69
B.11	Gráfico de onda da função $f(t) = \cos(2\pi 10t)$ e as possibilidades de quantificação uniforme.	p. 70
B.12	Esboço da curva entrada-saída do quantizador.	p. 71
B.13	Amostras de $t = -500$ ms a $t = 500$ ms, da onda $m(t) = 6\sin(2\pi t)$, com $f_m = 1$ Hz $\Rightarrow T_m = 1$ s, $f_s = 8$ Hz $\Rightarrow T_s = 125$ ms.	p. 72
B.14	Modelo com quatro níveis de quantização e seus respectivos passos e valores de pico do sinal.	p. 73
B.15	Modelo de quantizador não-uniforme.	p. 75
B.16	Distribuição estatística da fala de um único locutor nas magnitudes de sinais de voz.	p. 76
B.17	Modelo PCM.	p. 78
B.18	Sinal $x(t) = 4\cos(2\pi ft)$ e respectivos valores de amostra, amostras quantizadas, código e sequência digital PCM.	p. 79

Lista de Tabelas

B.1	Tabela de roteiros e opções de simulação para o Módulo “Amostragem e Quantização”.	p. 54
B.2	Tabela de pares transformados.	p. 62
B.3	Tabela de equivalências dos níveis de quantização	p. 66
B.4	Tabela de níveis de quantização e <i>bits</i> para o quantizador.	p. 71
B.5	Tabela de saídas do quantizador para um ciclo completo de entrada.	p. 72

1 *Introdução*

O relato de dificuldades de aprendizagem e a procura de recursos extra-classe para compreensão mais adequada dos conceitos recebidos no período da graduação são frequentes.

Alves e Mantovani (2016) afirmam que um ensino médio e fundamental pouco exigentes e especialmente deficientes na área de exatas estão como causa recorrente das dificuldades de aprendizagem nas faculdades de Engenharia, chegando até a dificultar a permanência dos alunos no curso. Entretanto os autores ressaltam:

“As deficiências cognitivas, principalmente em Matemática e Interpretação de Textos, observadas nos cursos de Engenharia, costumam acompanhar os estudantes ao longo do curso de graduação. No entanto, isso não pode ser utilizado como alibi para o baixo rendimento dos alunos durante o curso. É necessário que as instituições de ensino criem meios para suprir a carência dos estudantes e eles devem participar e se empenhar para reduzir seus próprios défices.”(ALVES; MANTOVANI, 2016)

Propõe-se, neste trabalho, um enfrentamento dessas questões oferecendo complementos as aulas presenciais em momentos fora do espaço de sala de aula na forma de uma ferramenta *web*. *Software* compondo um ambiente virtual para a aprendizagem tem sido utilizado como ferramenta educacional. O acesso promovido pela *internet* promove um melhor gerenciamento pelo aluno do tempo dedicado aos estudos, além de um amplo acesso de qualquer lugar passível de conexão.

A proposta de interação entre o ensino presencial e o a distância (EaD) tem sido avaliada no IFSC:

Entre as instituições de educação federal, o Instituto Federal de Santa Catarina (IF-SC) - Campus Florianópolis - passa por um momento em que alguns docentes vivenciam as duas realidades: o ensino presencial e a EaD. O cenário indica que as duas culturas educacionais - o ensino presencial e a EaD - estão caminhando rumo a

uma convergência de tecnologias e práticas educacionais. Novas tecnologias estão surgindo e aumentando a capacidade de interação entre aluno e professor, mesmo a distância. Assim, é interessante verificar se e como os cursos técnicos presenciais do IF-SC estão intensificando o uso das TICs [Tecnologias da Comunicação e Informação]. (ANDRADE; PEREIRA, 2012, p. 4)

1.1 Objetivo do trabalho

O objetivo do trabalho é a implementação de um *software* educacional, através de plataforma *web*, que possibilite atividades complementares às aulas presenciais. Tem-se módulos de ensino, compostos de roteiros, que são inseridos nesta plataforma.

Nesta versão, a proposta deste software educacional pode ter como casos de uso:

- A consulta de conteúdos já organizados previamente na ferramenta como módulos de estudo;
- A organização de módulos de estudo por alunos orientados pelo professor, para publicação na web, que estimulariam a revisão bibliográfica e a criação de roteiros sobre temas específicos;
- A criação de simulações por alunos que já tenham tido fundamentos de programação orientada a objetos, complementando os roteiros propostos no item anterior;
- A integração de disciplinas diversas com disciplinas de programação, estabelecendo momentos integradores no curso;
- O envolvimento de toda turma com um processo de estudo do código-fonte, disponível para *download*, e seu melhoramento.

Para demonstração da ferramenta propõe-se um módulo de “Amostragem e quantização”, que pode ser apontado como base tecnológica de um Curso de Comunicações Digitais e que compreenda o estudo dos seguintes tópicos:

- Teorema da amostragem;
- Quantização uniforme e não-uniforme;
- Modulação por codificação de pulso (PCM).

Pretende-se o desenvolvimento destes itens em sete roteiros, escritos de forma adequada à plataforma *web*.

Para o desenvolvimento do **módulo proposto** neste TCC, alguns conteúdos foram pré-requisitos: programação orientada a objetos, conhecimentos de princípios de telecomunicações, como unidades de medida em telecomunicações dB, dBm, ...), conhecimentos básicos de sinais e sistemas, e cálculo. Os tópicos citados podem ser entendidos como aqueles necessários ao desenvolvimento de competências e habilidades gerais no tema a ser trabalhado.

1.2 Motivação

Especificamente nesta implementação foi realizada uma proposta para alguns tópicos de Comunicações Digitais, porque houve ao longo da graduação um interesse gradativo nesta área. Ressalta-se que o tema proposto não implica uma limitação da ferramenta para esta área de conhecimento, mas apenas uma escolha para demonstração do software neste trabalho.

A possibilidade de estudar a linguagem de programação *Python* e o *framework* Django foram outros atrativos para a construção desta ferramenta.

Ao ler-se o artigo do professor Cunha (2015) optou-se pelo referencial da complexidade como marco para as propostas educacionais que foram realizadas no Capítulo 2 e estabeleceu-se uma identificação com esse pensar.

1.3 Organização do texto

O texto está organizado da seguinte forma: No Capítulo 2 são apresentadas as indagações que levaram a elaboração do projeto, sua justificativa de execução, além de questões pertinentes a metodologia utilizada e a elaboração de conteúdo a ser inserido na forma de hipermídia. No Capítulo 3 é apresentado o desenvolvimento da ferramenta e as tecnologias envolvidas. Por fim, no Capítulo 4 são apresentadas as conclusões sobre este trabalho.

Estão inclusos também dois apêndices: o primeiro contendo um passo a passo da instalação das ferramentas e do próprio *software*, e o segundo com o texto dos roteiros de estudo do módulo de “Amostragem e quantização”.

2 *Software educacionais*

O avanço tecnológico na atualidade é inegável. A compreensão das mudanças propiciadas por este movimento ainda estão em processo de assimilação, já que este progresso continua. Todas as atividades estão permeadas por tecnologia de alguma maneira. Andrade e Pereira (2012) apontam que:

Muito longe de ser uma exceção, a educação formalizada nas escolas, institutos de educação e universidades, configura-se em mais uma dessas atividades afetadas pelo avanço das TICs. No universo daquilo que é chamado de educação formal, a educação a distância (EaD) é a representação do uso intenso de TICs no processo de ensino e de aprendizagem. (ANDRADE; PEREIRA, 2012, p. 2)

Procurar novas formas que tornem efetiva a aprendizagem, ou seja, que o aluno atinja os objetivos de um plano de ensino, tem sido objeto de discussão em qualquer reunião de planejamento didático. Ainda Andrade e Pereira (2012) ressaltam que:

Pesquisas recentes demonstram que as duas modalidades de educação estão se aproximando de um modelo híbrido que integre o que há de bom do ensino presencial com as inovações da EaD, tese que é defendida por Tori(2010), Luzzi(2007) e outros pesquisadores da educação. (ANDRADE; PEREIRA, 2012, p. 3)

Mas antes disso, cabe a verificação, ainda que brevemente, desse processo.

2.1 O processo de ensino-aprendizagem a partir da óptica dos sistemas complexos

Segundo Cunha (2015), ao realizar-se o planejamento e à operacionalização do ensino, alguns aspectos do processo ensino-aprendizagem relacionados mais especificamente ao ambiente da sala de aula, à escolha de metodologias de ensino, à abordagem dos conteúdos, ao

desenvolvimento de competências e à estruturação do currículo, podem ser ampliadas e aplicadas a partir do referencial da complexidade.

Cunha (2015) ressalta que, essencialmente, a questão maior é alterar o foco do processo de ensino, do professor para o estudante. Essa alteração implica trabalhar a aprendizagem ativa em detrimento da aprendizagem passiva. Exemplos desta última, seriam as tradicionais formas utilizadas para ensinar, tais como a aula expositiva centralizada no professor e as atividades de laboratório com procedimentos padronizados e direcionados para a comprovação de teorias. Já a aprendizagem ativa implica em trabalhar o próprio aprender.

Conforme Andrade e Pereira (2012), há o professor mediador do processo de ensino-aprendizagem e o professor repassador de conteúdo. Delimitar estas duas situações pode não ser tão óbvio:

“No ensino presencial as fronteiras entre ser um professor mediador e ser um professor repassador de conteúdo ainda não estão muito claras. O que é ser mediador quando se dá aula com livro didático, lousa e giz? No ensino presencial é muito comum conteúdos em slides projetados em telão para discutir com os alunos; isso é ser mediador, mas é também ser repassador de conteúdo.”(ANDRADE; PEREIRA, 2012, p. 157)

Ainda que se proponha uma integração de formas de ensinar e aprender, utilizadas tanto nos sistemas de educação à distância, quanto em sistemas de educação presencial, para se chegar no “melhor de dois mundos”, e que o foco passe a ser o estudante, é inegável que ao professor cabe papel de extrema relevância e grande trabalho. Conforme Andrade e Pereira (2012):

“No centro dessa convergência está o professor que é desafiado a dominar novas tecnologias, dialogar com profissionais de outras áreas, adaptar materiais didáticos a linguagem multimidiática, ter versatilidade diante das mudanças e desconstruir conceitos relacionados à cultura do ensino presencial.”(ANDRADE; PEREIRA, 2012, p. 154)

Estratégias relacionadas ao aprender a aprender são temas de trabalhos e teses (Cunha (2015, p. 4)). Pensa-se que é desafiador compreender a proposta de inovação, e para isto a teoria dos sistemas complexos pode ajudar. A teoria da complexidade pode ser aplicada quando se pensar os diversos elementos de um sistema interligados e interagindo entre si. Esses elementos podem ser entendidos como aqueles que se inter-relacionam na sala de aula.

2.1.1 Sistemas complexos

Cunha (2015), considera “que determinados processos apresentam relações imbricadas de tal forma que é impossível abordá-los sem considerar a teia das relações que se estabelecem entre eles.” Isso, a noção de complexidade, tem em oposição a noção de simplicidade, que não necessariamente é excluída na complexidade. A proposta é a de que se use o pensamento complexo, onde o simples não dê conta.

Conforme Cunha (2015), pode-se entender como três, os princípios que norteiam a noção de simplicidade:

1. Ordenação: os processos são ordenáveis e assim, determinísticos. O acaso (se acontecer) é por ignorância de algum conhecimento, e portanto, situação a resolver;
2. Análise: separa-se os fatos para estudo detalhado dos fenômenos. A realidade é desvendada com precisão e exatidão;
3. Exclusão ou independência: a exclusão ou afastamento se refere ao observador, que não interfere no processo.

Já nas proposições complexas, pensa-se que a desordem e a incerteza estão integradas ao processo de conhecer. O que se avalia na experiência enquanto aluno, que apesar de bastante delimitado o conteúdo a ser desenvolvido durante a graduação, o processo de aprendizagem é permeado pela incerteza, na visão dos alunos, e desordenado quanto a indisciplinaridade, nem tão evidente para os discentes. Morin apud Cunha (2015) tratando da incerteza do conhecimento: “O conhecimento é, pois, uma aventura incerta que comporta em si mesma, permanentemente, o risco de ilusão e de erro.” (Morin (2000, p. 86)).

Nussenzveig apud Cunha (2015) elenca as características de um sistema complexo:

“[...] é dinâmico e está em constante evolução; é aberto, interagindo com o meio ambiente; cada unidade produz uma resposta aos sinais que recebe das outras, sendo que essa resposta não guarda uma simples relação de proporcionalidade ao estímulo recebido; o sistema responde aleatoriamente aos impulsos recebidos; o sistema se auto organiza de forma espontânea, criando ordem a partir de um estado desordenado; podem surgir propriedades coletivas emergentes, com novas interações diante de estados críticos; o comportamento do sistema depende da história anterior; o sistema é adaptativo, isto é, envolve aprendizado em função da interação com o ambiente.”(CUNHA, 2015)

No âmbito do ensino de TICs, o dinamismo da criação e revisão de tecnologias gera novos saberes constantemente, implicando em mais um motivo na visão de complexidade destes sistemas.

2.1.2 A sala de aula

O sistema complexo que pretende-se trabalhar é o da sala de aula, onde interagem os alunos e os professores. Esses atores são pessoas, com visões de mundo, histórias de vida, objetivos diversos, mas com um propósito de elaboração em conjunto: a aprendizagem. Por isso, Cunha (2015) define assim: “A sala de aula é o lugar do encontro.” Com estas definições, avalia-se a sala de aula como espaço de convivência, então pode-se propor com esta perspectiva metodologias que possam facilitar esta convivência e promover uma certa homogeneidade de conhecimentos acerca do que está sendo aprendido.

Muito mais que construção de conhecimento, o espaço é de desconstrução. Deve-se privilegiar antes a dúvida, que a certeza. Assim, não necessariamente a finalização do semestre ou do curso, é o determinado pelo professor ou pela direção. Ainda Cunha (2015): “Não é, portanto, um conhecimento simplesmente transmitido, mas elaborado, testado, internalizado pelos sujeitos.” Aí avalia-se a necessidade de um ferramental de ensino que transcenda o espaço e o instante da aula propriamente dita e propicie entendimento e elaboração de teorias, estimulando o processo criativo.

Promover autonomia é também muito desejável a todos os alunos de TICs. Frequentemente o discurso da necessidade de “correr atrás” de soluções para os problemas que surgiam no processo educativo, é anunciado. Mas como se pensa autonomia?

“Assim, deve-se pensar ‘autonomia’ não enquanto condição para a EaD e mas como uma qualidade a ser promovida por ela. O ensino presencial também exige, sob certa medida, a autogestão por parte do aluno, mas não se pode esperar que ele venha para a escola pronto, dotado em plenitude de autonomia. É ali, na escola, que vai desenvolver essas qualidades” (ANDRADE; PEREIRA, 2012, p. 155).

Então reavalia-se o entendimento que temos de teoria, conforme Morin (2005):

“Uma teoria não é o conhecimento; ela permite o conhecimento. Uma teoria não é uma chegada; é a possibilidade de uma partida. Uma teoria não é uma solução; é a possibilidade de tratar um problema. Em outras palavras, uma teoria só realiza seu papel cognitivo, só ganha vida com o pleno emprego da atividade mental do sujeito.

É essa intervenção do sujeito que dá ao termo método seu papel indispensável.” (MORIN, 2005, p. 335).

2.2 Metodologias

Objetiva-se uma proposta na qual o aluno trabalhe o próprio problema além de sua solução, com foco maior no processo que no produto da aprendizagem; promovendo a autonomia e o comprometimento do aluno neste processo de aprendizagem, e não com a visão de responsabilidades para apenas um dos elementos do sistema, seja o aluno, o professor ou a escola, na representação de seus gestores.

O *software* educacional pode ser entendido como problema/projeto, por possibilitar a consulta e simulação de exercícios, a inserção de novos conteúdos, o desenvolvimento de novos simuladores e a geração de material didático (apostilas, figuras, referências).

A possibilidade deste *software* ser utilizado no estudo extra-classe implica alguns benefícios já conhecidos: maior liberdade de estudo e gerenciamento do tempo. Além disso, pode-se ver o *software* como objeto de aprendizagem, ou seja, um material facilmente acessível a compreensão e navegação.

Os *software* educacionais são uma das possibilidades de complementar o momento com o professor em sala de aula. Então definir estas ferramentas e a melhor forma de implementá-las torna-se fundamental.

Botti (2015), et al., compilam conceitos que podem apontar diretrizes na implementação dos *software* em si:

“Os *software* educativos são programas de computador desenvolvidos especificamente visando a aprendizagem de determinado conteúdo, competência ou habilidade. Sua utilização facilita a aprendizagem através da interação, motivação e descoberta (PRIETO et al., 2005). Sabe-se que interfaces educativas servem como meio à negociação de significado e construção de conhecimentos específico em contextos específicos (GOMES; WANDERLEY, 2003). Assim, o *software* educativo pode promover aprendizagem, demanda cognitiva para a aquisição do conhecimento e construção de relações e conceitos (BASSANI et al., 2006).” (BOTTI, 2015, p. 24)

Opta-se por uma ferramenta *web*, para que se tenha um acesso mais amplo e também por

verificar-se ser comum o uso da internet por estudantes, o que não implicaria treinamento no uso do aplicativo.

As discussões propostas até agora antecedem o desenvolvimento da ferramenta, podendo ser revistas ao longo de seu uso.

2.3 Elaboração de conteúdo

Conforme as necessidades evidenciadas na citação anterior, comporão a fase inicial do processo de desenvolvimento:

1. A elaboração de planos de ensino estruturados em roteiros, com objetivos específicos, subsídio em texto a ser disponibilizado em hipertexto, com referência bibliográfica e exercícios;
2. O desenvolvimento de mídias (figuras, vídeo, áudio, etc.) a serem integrados ao hipertexto;
3. Simulador com funcionalidades específicas para o roteiro.

Admite-se esta fase como uma base de definição tanto administrativa, quanto didática, para elaboração de conteúdo.

Cada roteiro comporá um tópico do grande tema a ser estudado. Vários roteiros comporão um módulo, que possuirá um objetivo geral, estendido a todos os roteiros. Os vários módulos comporão o conteúdo geral de Comunicações Digitais, definido na Seção 1.1.

Para este trabalho, elaborar-se-á um módulo de Amostragem e Quantização, com sete roteiros de estudo, que se encontram no Apêndice B. Cita-se os roteiros:

- Roteiro 0: Um modelo de Comunicação Digital;
- Roteiro 1: Amostragem;
- Roteiro 2: Teorema da Amostragem;
- Roteiro 3: Quantização e Codificação;
- Roteiro 4: Quantização Uniforme;
- Roteiro 5: Quantização Não-Uniforme;
- Roteiro 6: Modulação por codificação de pulso - PCM.

2.4 Usabilidade

Entende-se usabilidade como a capacidade de um sistema ser utilizado com facilidade e com eficiência pelo usuário. Assim procurou-se uma série de padrões já consolidados para serem aplicados na ferramenta.

Procura-se atender aos critérios de ergonomia e usabilidade, já consagrados em seu uso e propostos inicialmente pelos pesquisadores franceses, Dominique Scapin e Christian Bastien, ligados ao INRIA (Instituto Nacional de Pesquisa em Automação e Informática da França). Conforme Cybis, Betiol e Faust (2010):

“Eles propuseram, em 1993, um conjunto de oito critérios ergonômicos principais que se subdividem em 18 subcritérios e critérios elementares. O objetivo de tal sistema é o de minimizar a ambiguidade na identificação e classificação das qualidades e problemas ergonômicos do *software* interativo.

Os critérios, subcritérios e critérios elementares são:

- condução
 - convite
 - agrupamento e distinção entre itens
 - * agrupamento e distinção por localização
 - * agrupamento e distinção por formato
 - legibilidade
 - feedback imediato
- carga de trabalho
 - brevidade
 - * concisão
 - * ações mínimas
 - densidade informacional
- controle explícito
 - ações explícitas
 - controle do usuário
- adaptabilidade
 - flexibilidade

- consideração da experiência do usuário
- gestão de erros
 - proteção contra os erros
 - qualidade das mensagens de erros
 - correção dos erros
- homogeneidade/consistência
- significado de códigos e denominações
- compatibilidade”

(CYBIS; BETIOL; FAUST, 2010, p. 25-26)

2.4.1 Navegação

O sistema possui duas interfaces: a de usuário e a de administrador. A de usuário destina-se aos alunos que estiverem consultando a ferramenta para o aprendizado do conteúdo de um módulo e de seus roteiros. A de administrador destina-se aos professores ou alunos que estiverem realizando a inserção de conteúdos ou administrando os usuários da ferramenta.

Interface de usuário

O *site* apresentará possibilidades de navegação no cabeçalho, barra lateral esquerda e rodapé. Assim procura-se atender os critérios de *condução*, *carga de trabalho* e *controle explícito*, conforme a Seção 2.4.

O módulo estará como *link* na barra superior e os roteiros comporão um menu flutuante oculto, acessível ao se posicionar o ponteiro do *mouse* sobre o *link* do módulo. O menu lateral à esquerda terá também *links* para funcionalidades, quais sejam, acesso a material geral do IFSC-SJ, como página *wiki*, página da biblioteca, e em evidência o *link* para a página de um **simulador** com todas as possíveis funcionalidades, independente de roteiros específicos. No alto à direita, há o *link* para a área restrita, a interface administrativa. Esses *links* mudarão a cor da fonte ou da borda, ao se posicionar o *mouse* sobre eles, facilitando a visualização e o situar-se na página. A proposta pode ser visualizada na Figura 2.1. Quando for acessado um roteiro, próximo do rodapé haverá indicativo de qual roteiro está sendo visualizado e as possibilidades de outros roteiros naquele módulo. A proposta pode ser visualizada na Figura 2.2, sendo que a localização das informações sobre o módulo e roteiro, bem como os *links* estão circundados em vermelho (apenas na figura).

Haverá a possibilidade de inserção de um simulador de sinais, amostragem e quantizações nas páginas de roteiro de cada módulo. As questões de usabilidade deste simulador serão descritas na Subseção 2.4.2.



Figura 2.1: Modelo do cabeçalho e menus laterais da página *index* da ferramenta.

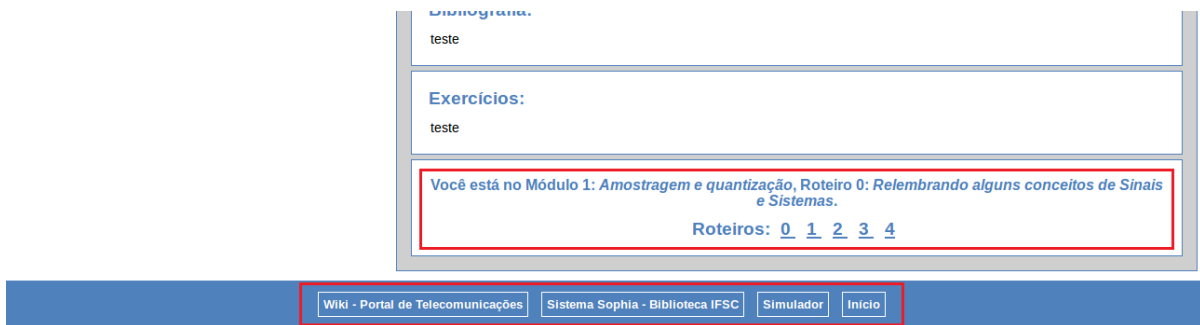


Figura 2.2: Modelo do rodapé da página de roteiros da ferramenta.

Interface de administrador

A interface de administração da ferramenta inicia por uma página de autenticação, onde são solicitados usuário e senha. As possibilidades são distintas conforme o perfil de quem acessa. Criou-se dois grupos: um grupo *admin* e outro grupo *user*.

O usuário *admin* possui em sua tela, acesso a administração de Grupos e Usuários, e da página Index, Módulos e Roteiros. Outras ações como acesso ao *site*, alteração de senha e encerramento de sessão, são possibilitados por um menu no topo à direita. Há um quadro com

feedback de ações realizadas. Esta interface é dada pelo *framework* Django — a ser descrito no Capítulo 3— mas necessita de customização. A tela pode ser visualizada na Figura 2.3.

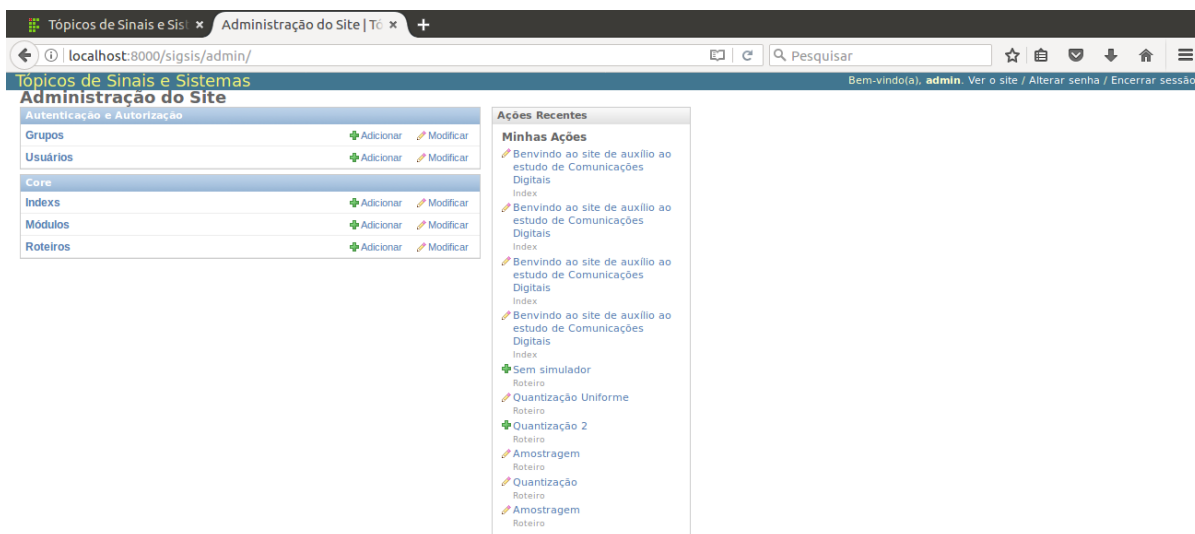


Figura 2.3: Interface de administração do usuário *admin*.

Já o usuário *user* possui em sua tela, acesso restrito, possibilitando a administração da página *Index*, *Módulos* e *Roteiros*. Observa-se que a página *Index*, está setada apenas para modificação, não podendo ser excluída ou nova página ser criada, por este usuário, o que se comprova pela ausência do botão “+ Adicionar”, como pode ser visualizado na Figura 2.4. Outras ações como acesso ao *site*, alteração de senha e encerramento de sessão, permanecem pelo menu no topo à direita. Também se mantém o quadro com *feedback* de ações realizadas.

Demonstra-se assim, uma possibilidade de restrição, customizada pelo usuário *admin*, nas guias *Grupos* e *Usuários*. O que pretende-se evidenciar com a criação deste grupo de usuários (*user*), é que, conforme acordos pré-estabelecidos, poderia-se por exemplo, deixar ao encargo de um professor a criação dos módulos e seus roteiros, e permitir a um aluno vinculado a ele, somente a inserção de conteúdo. Isto através da criação de um novo grupo com estas permissões específicas, por exemplo, um grupo *editor*.

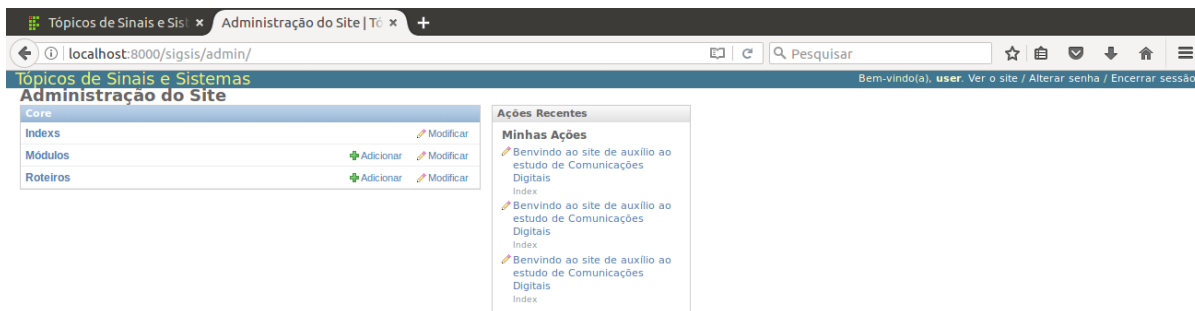


Figura 2.4: Interface de administração do usuário *user*.

Tanto a guia *Grupos*, quanto a guia *Usuários* lista os grupos/usuários criados e possibilita a criação de um novo grupo/usuário, através de um botão “Adicionar grupo/usuário +”. Ao lado de cada nome de grupo/usuário, há um *checkbox* que, ao ser marcado, possibilita no *input* “Ação”, a sua exclusão. Na guia *Usuário* há um “Filtro”, onde é possível filtrar usuários por grupo, entre outras ações, de acordo com as parametrizações realizadas individualmente, em cada grupo e usuário.

Ao se criar um novo grupo ou acessar um grupo criado, há um *input* com o nome do grupo e outro campo com as permissões.

Já para o usuário há mais campos a definir. Inicia-se com Nome e Senha. Depois há *fieldsets* específicos e seus campos a parametrizar:

- *Informações pessoais*

- Primeiro nome;
- Último nome;
- Endereço de email;

- *Permissões*

- Ativo - possui texto de ajuda: “Indica que o usuário será tratado como ativo. Ao invés de excluir contas de usuário, desmarque isso.”
- Membro da equipe - possui texto de ajuda: “Indica que usuário consegue acessar este *site* de administração.”
- Status de superusuário - possui texto de ajuda: “Indica que este usuário tem todas as permissões sem atribuí-las explicitamente.”
- Grupos - possui texto de ajuda: “Os grupos que este usuário pertence. Um usuário terá todas as permissões concedidas a cada um dos seus grupos. Mantenha pressionado o “Control”, ou “Command” no Mac, para selecionar mais de uma opção.”
- Permissões do usuário - possui texto de ajuda: “Permissões específicas para este usuário. Mantenha pressionado o “Control”, ou “Command” no Mac, para selecionar mais de uma opção.”

Importante: *Portanto há a possibilidade de permissionamentos específicos para cada usuário, através do campo Permissões do usuário.*

- *Datas importantes* - Este *fieldset* é povoado automaticamente e não necessita ser preenchido.

Ressalta-se que essa interface é propiciada pelo *framework* Django — a ser descrito no Capítulo 3, e não foi alterada.

A guia **Index** possui os seguintes campos a serem preenchidos: *Título do Texto* e *Texto da página Home/Index*.

A guia **Módulo** possui os seguintes campos a serem preenchidos: *Título do Módulo*, *Número de sequência do Módulo* e *Objetivo Geral do Módulo*.

Já a guia **Roteiro** possui um número maior de campos a serem preenchidos:

1. *Título do Roteiro*;
2. *Número de sequência do Roteiro*;
3. *Objetivo Geral do Roteiro*;
4. *Objetivos Específicos do Roteiro*;
5. *Módulo* - Este campo será povoado com os módulos já cadastrados;
6. *Simulador do Roteiro* - Este campo será povoado com as seguintes possibilidades: Sem Simulador, Sinal, Amostras, Quantizador Genérico e Quantizador Uniforme;
7. *Subsídios do Roteiro*;
8. *Bibliografia do Roteiro*;
9. *Exercícios do Roteiro*.

Todos os campos tem indicação de “:”, para evidenciar a necessidade de preenchimento. A descrição do *input* de dados, compõe um *label* que ao ser clicado, leva à área de preenchimento ou assinala o *checkbox* correspondente; assim não é imprescindível que se clique dentro da área de preenchimento para acessá-la.

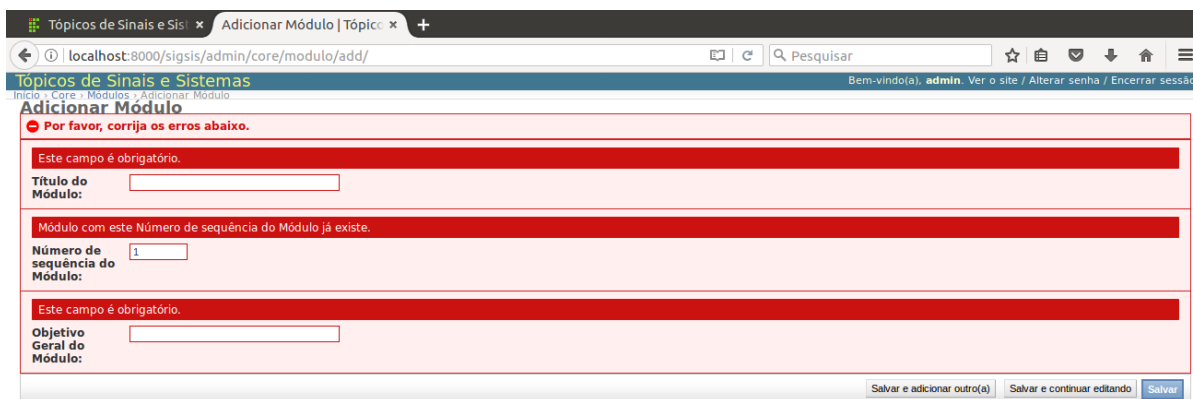
Assim procura-se atender os critérios de *condução*, *carga de trabalho*, *gestão de erros* e, em especial, o subcritério de *agrupamento e distinção entre itens*, conforme a Seção 2.4.

Estes conjuntos de interfaces (Index, Módulos e Roteiros) são gerados pelo *framework* Django a partir de parametrizações em código-fonte a serem descritas no Capítulo 3.

Tratamento de erros da interface de administração

Os avisos de erro são padrão do *framework* Django e apresentam-se em caixas de texto de fundo vermelho, mas as condições para que ocorram devem ser desenvolvidas. Os campos indicativos de números de sequência somente aceitam caracteres numéricos inteiros. Não é possível salvar página em index, módulos ou roteiros com campos vazios.

Módulo: Parametrizou-se a inserção de dados para um novo *Módulo*, de forma a não ser possível campos vazios e também que cada novo módulo tenha um único número de sequência. Isto será necessário para que se faça a correta listagem de *links* no menu do cabeçalho, lateral e do rodapé, na interface de usuário. Pode-se verificar as mensagens de erro na Figura 2.5.



A captura de tela mostra uma página web de administração com o seguinte conteúdo:

- Barra de navegação superior: "Tópicos de Sinais e Sis" e "Adicionar Módulo | Tópicos".
- URL: localhost:8000/sigsis/admin/core/modulo/add/
- Barra de usuário: Bem-vind(a), admin. Ver o site / Alterar senha / Encerrar sessão
- Título da página: Adicionar Módulo
- Mensagem de erro geral: Por favor, corrija os erros abaixo.
- Formulário com três seções de erro destacadas em vermelho:
 - Seção 1: "Este campo é obrigatório." seguido por um campo vazio para "Título do Módulo".
 - Seção 2: "Módulo com este Número de sequência do Módulo já existe." seguido por um campo com o valor "1" para "Número de sequência do Módulo".
 - Seção 3: "Este campo é obrigatório." seguido por um campo vazio para "Objetivo Geral do Módulo".
- Botões de ação: "Salvar e adicionar outro(a)", "Salvar e continuar editando" e "Salvar".

Figura 2.5: Mensagens de erro na página de parametrização de módulo.

Roteiro: Há uma verificação para que não seja inserido dois roteiros com mesmo número de sequência, em um mesmo módulo. Pode-se verificar a mensagem de erro na Figura 2.6.

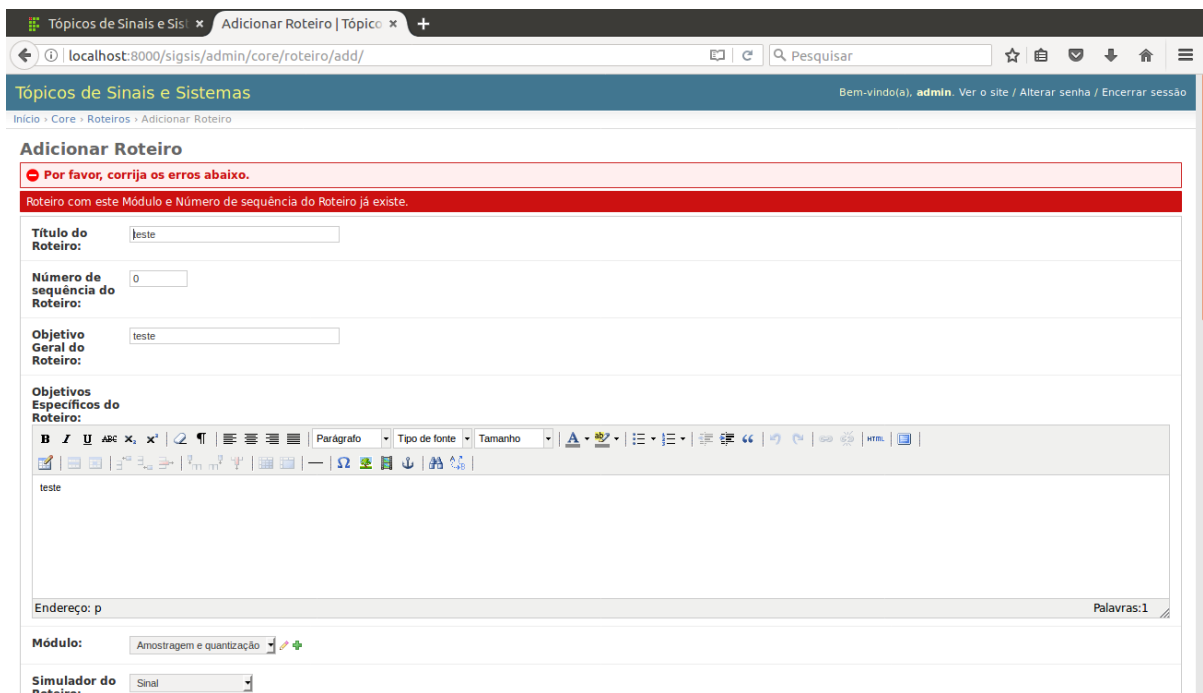


Figura 2.6: Mensagens de erro na página de parametrização de roteiro duplicado.

2.4.2 Simulador

O simulador construirá gráficos de sinais contínuos e discretos, com a parametrização de suas características pelo usuário. Pode ou não estar na página, conforme o desejo daquele que publicar um roteiro. Ao ser enviado os dados do formulário, retornará com o processamento na mesma página, no início da área do simulador, para melhor visualização do gráfico solicitado.

Áreas de preenchimento do simulador

Organizou-se o simulador num grande campo de dados ou *fieldset* com outros *fieldsets* internamente delimitando áreas de preenchimento, numa proposta de organização de raciocínio para o usuário. Todos os campos tem indicação de “:”, para evidenciar a necessidade de preenchimento. Houve o cuidado de parametrizar os campos de dados para que fosse possível avançar adequadamente, utilizando a tecla de tabulação. A descrição do *input* de dados, compõe um *label* que ao ser clicado, leva à área de preenchimento ou assinala o *checkbox* correspondente; assim não é imprescindível que se clique dentro da área de preenchimento, para acessá-la.

Assim procura-se atender os critérios de *condução*, *carga de trabalho*, *gestão de erros* e, em especial, o subcritério de *agrupamento e distinção entre itens*, conforme a Seção 2.4.

As áreas internas e seus respectivos campos são:

1. Gráfico/Plano - compreende o título do gráfico e a extensão do eixo X. Os *inputs* já vem com preenchimento inicial, como forma de sugestão de dados. O texto entre colchetes sugere a unidade adotada ou identificadores da plotagem. Inclui os campos:
 - (a) Título do gráfico;
 - (b) Início do eixo [s];
 - (c) Fim do eixo [s].

2. Sinal - compreende as especificações do sinal a ser representado. Inclui os campos:
 - (a) Componente de tensão contínua DC [V];
 - (b) Amplitude A [V];
 - (c) Forma de Onda - este campo permitirá a escolha dos seguintes sinais: cosseno, seno, sinc, sinc², serra, quadrada e triangular;
 - (d) Frequência Fundamental f [Hz];
 - (e) Deslocamento no tempo [s].

3. Simulação - compreende as especificações das simulações possíveis. Mais de uma simulação poderá ser vista no mesmo gráfico. Existem campos que ficarão visíveis conforme a simulação pretendida e parametrizada por aquele que publicar um roteiro. Inclui os campos:
 - (a) Sinais e Transformada de Fourier - aparecem em todas as simulações. Incluem os campos:
 - i. Sinal [Vermelho] - que *plotará* apenas o sinal;
 - ii. Transformada de Fourier [Azul] que *plotará* apenas a Transformada de Fourier do sinal parametrizado;

Quando selecionada a simulação *Sinal*, estes *inputs* passam a ser exibidos. Caso sejam selecionados os dois campos, a imagem terá as duas plotagens em áreas diferentes da mesma figura, já que se tratam de domínios diferentes em cada solicitação. Na Figura 2.7 pode-se visualizar a renderização com os dois campos solicitados.

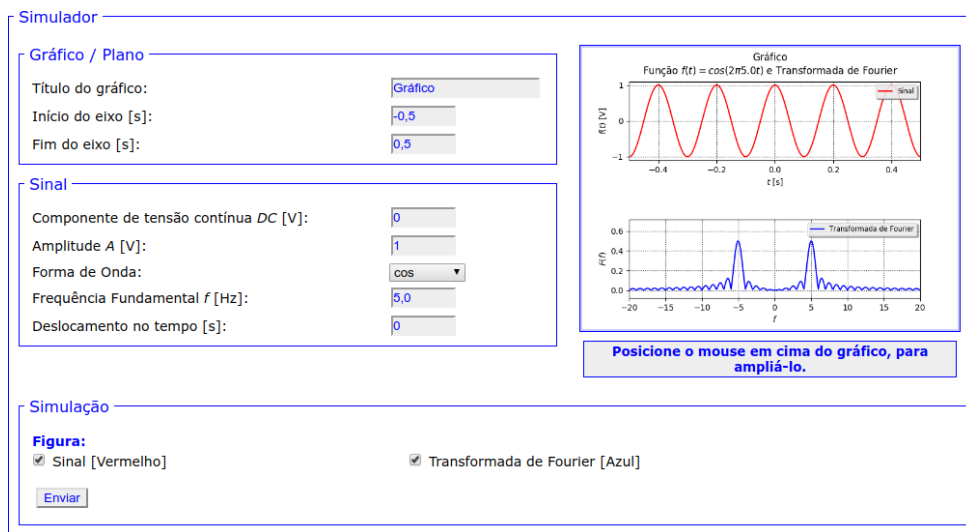


Figura 2.7: Renderização do simulador de Sinais e Transformada de Fourier.

(b) Sinais e amostragem. Incluem os campos:

- i. Taxa de amostragem f_s [Hz];
- ii. Sinal [Vermelho] - que *plotará* apenas o sinal;
- iii. Transformada de Fourier [Azul] que *plotará* apenas a Transformada de Fourier do sinal parametrizado;
- iv. Amostras [Verde - Tracejado] que *plotará* apenas as amostras;

Quando selecionada a simulação *Amostragem*, estes *inputs* passam a ser exibidos. Na Figura 2.8 pode-se visualizar essa renderização.

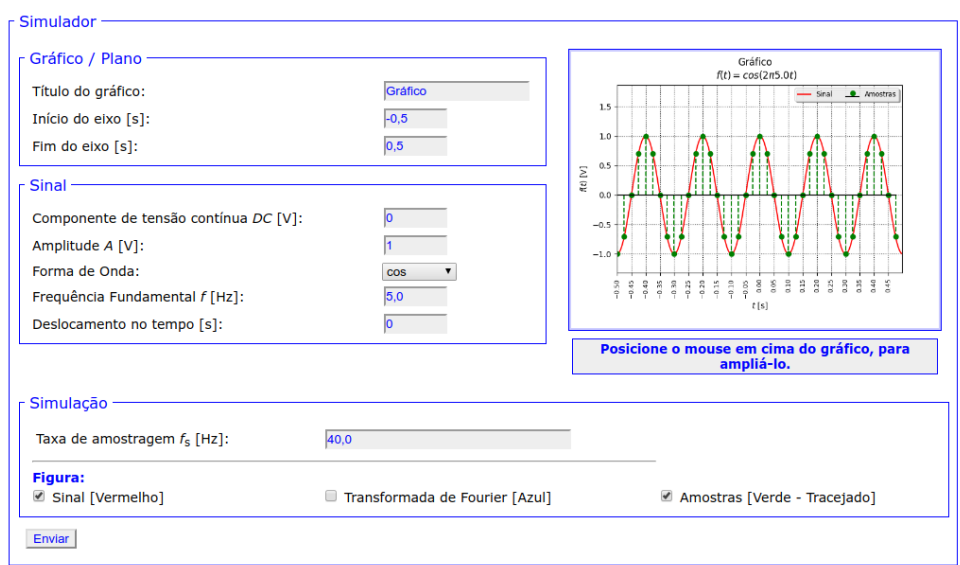


Figura 2.8: Renderização do simulador de Sinais, Transformada de Fourier e Amostras.

(c) Quantização Genérica - somente aparecerá quando assim for parametrizado por

aquele que publicar um roteiro. Inclui os campos:

- i. Níveis de Quantização - há um elemento de ajuda, sinalizado por uma figura com um ponto de interrogação, que ao se posicionar o ponteiro do *mouse* sobre a mesma, apresentará o texto: “Para o Quantizador Genérico. Em ordem crescente e separados por espaço em branco.”;
- ii. Limiar inferior do nível [0-1] - da mesma forma que o anterior, há um elemento de ajuda, com o seguinte texto: “Para o Quantizador Genérico. O número entre 0 e 1, refere-se a fração ou percentual para o limiar inferior do nível. *Default*: metade ou 0,5.”;
- iii. Sinal Quantizado - Genérico [Ciano] - que *plotará* apenas este sinal quantizado;
- iv. Erro de Quantização Genérico [Cinza] - que *plotará* este erro de quantização.

Quando selecionada a simulação *Quantização Genérica*, este *fieldset* passa a ser exibido. Na Figura 2.9 pode-se visualizar essa renderização.

Simulador

Gráfico / Plano

Título do gráfico:

Início do eixo [s]:

Fim do eixo [s]:

Sinal

Componente de tensão contínua DC [V]:

Amplitude A [V]:

Forma de Onda:

Frequência Fundamental f [Hz]:

Deslocamento no tempo [s]:

Posicione o mouse em cima do gráfico, para ampliá-lo.

Simulação

Taxa de amostragem f_s [Hz]:

Figura:

Sinal [Vermelho] Transformada de Fourier [Azul] Amostras [Verde - Tracejado]

Quantização Genérica

Níveis de Quantização:

Limiar inferior do nível [0-1]:

Figura:

Sinal Quantizado - Genérico [Ciano] Erro de Quantização Genérico [Cinza]

Figura 2.9: Renderização do simulador do Quantizador Genérico.

- (d) Quantização Uniforme - somente aparecerá quando assim for parametrizado por aquele que publicar um roteiro. Inclui os campos:

- i. N^o de Níveis de Quantização [2-256] - da mesma forma que em elementos anteriores, há um elemento de ajuda, com o seguinte texto: “O valor deverá estar entre 2 e 256, pensando em potências de 2 (2^1 a 2^8), teremos valores que variarão entre 1 *bit* a 8 *bits*.”;
- ii. Sinal Quantizado *Mid Tread* [Azul] - que *plotará* este sinal quantizado;
- iii. Erro de Quantização *Mid Tread* [Cinza] - que *plotará* este erro de quantização;
- iv. Sinal Quantizado *Mid Rise* [Magenta] - que *plotará* este sinal quantizado;
- v. Erro de Quantização *Mid Rise* [Preto] - que *plotará* este erro de quantização.

Quando selecionada a simulação *Quantização Uniforme*, este *fieldset* passa a ser exibido. Na Figura 2.10 pode-se visualizar essa renderização.

Simulador

Gráfico / Plano

Título do gráfico:

Início do eixo [s]:

Fim do eixo [s]:

Sinal

Componente de tensão contínua DC [V]:

Amplitude A [V]:

Forma de Onda:

Frequência Fundamental f [Hz]:

Deslocamento no tempo [s]:

Posicione o mouse em cima do gráfico, para ampliá-lo.

Simulação

Taxa de amostragem f_s [Hz]:

Figura:

Sinal [Vermelho] Transformada de Fourier [Azul] Amostras [Verde - Tracejado]

Quantização Uniforme

N^o de Níveis de Quantização [2-256]:

Figura:

Sinal Quantizado *Mid-tread* [Azul] Erro de Quantização *Mid-tread* [Cinza]

Sinal Quantizado *Mid-rise* [Magenta] Erro de Quantização *Mid-rise* [Preto]

Figura 2.10: Renderização do simulador do Quantizador Uniforme.

A figura com o gráfico estará em modo minimizado, à direita no topo da área do simulador, mas se ampliará ao posicionar-se o ponteiro do *mouse* sobre ela. A posição da ampliação, sempre ocupará o ponto de visualização do usuário. Na Figura 2.11 pode-se visualizar essa renderização. Observa-se na imagem, que estará representada a função abaixo do título do

gráfico. Também na área do gráfico, apresenta-se legenda para identificar as ondas ou sinais gerados.

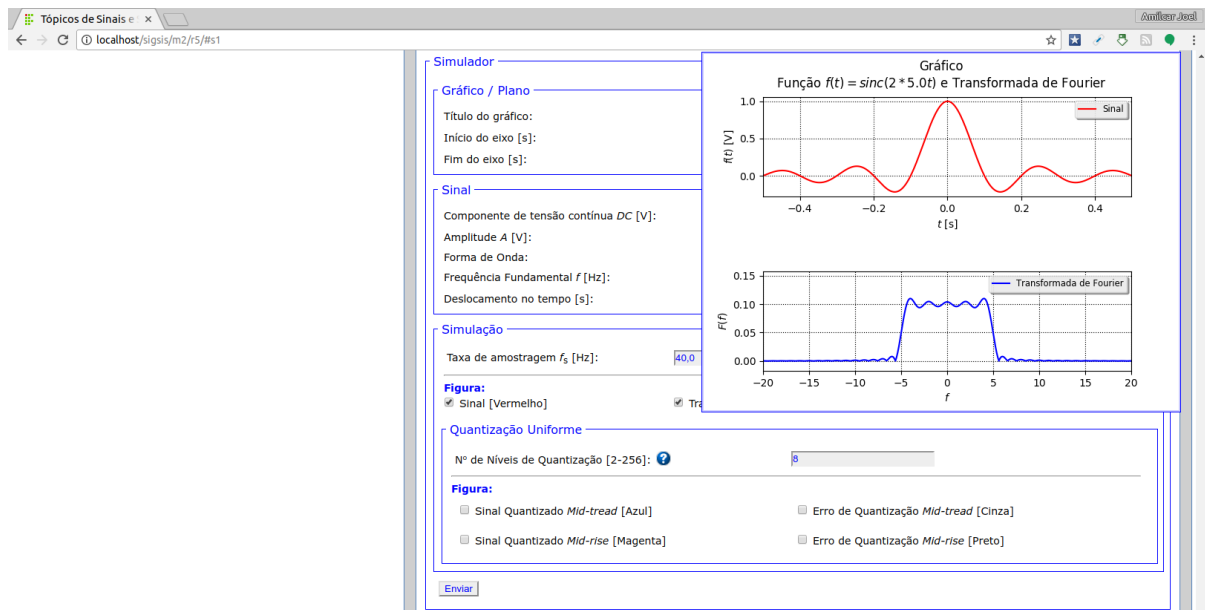


Figura 2.11: Ampliação da figura do simulador ao se posicionar o *mouse* sobre a mesma.

Tratamento de erros do simulador

O simulador apresenta vários tratamentos de erros e abordar-se-á por campos de dados ou *fieldsets*. O tratamento de erros procurou atender os critérios de *condução*, *carga de trabalho*, *gestão de erros* já expostos na Seção 2.4.

Fieldset Gráfico/Plano: apresenta campos de texto e números. Todos os campos são de preenchimento obrigatório, mas pode-se deixar os valores já inseridos.

Os campos *Início do eixo* e *Fim do eixo* possuem verificação, que não permite a inserção de letras ou outros caracteres que não números, sejam eles inteiros ou não. Ao tentar se inserir uma letra por exemplo, o formulário exibirá uma mensagem de erro, e não permite o envio. Estes campos aceitam tanto o separador decimal como o ponto, quanto como a vírgula; também é possível usar a notação ponto e número, para frações de um. Verifica-se também, se o início do eixo é menor que o fim do eixo; caso contrário, surgirá mensagem de erro e os campos em questão receberão borda avermelhada. Na Figura 2.12 pode-se visualizar essa renderização.

Fieldset Sinal: apresenta campos de números e um *combobox* para a forma de onda. Da mesma forma que o *fieldset* anterior, verifica-se caso haja valores não numéricos, sinalizando ao usuário e impedindo o envio do formulário. Todos os campos são de preenchimento obrigatório, mas pode-se deixar os valores já inseridos.

Simulador

Gráfico / Plano

Título do gráfico:

Início do eixo [s]:

Fim do eixo [s]:

Sinal

Componente de tensão contínua DC [V]:

Amplitude A [V]:

Forma de Onda:

Frequência Fundamental f [Hz]:

Deslocamento no tempo [s]:

Simulação

Figura:

Sinal [Vermelho] Transformada de Fourier [Azul]

Não processado! O início do eixo não pode ser maior que o fim do eixo. Revise os valores.



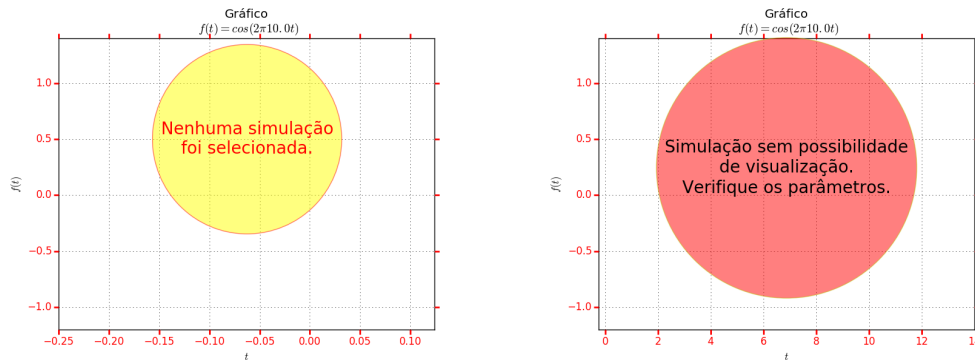
Figura 2.12: Mensagem de erro de parametrização do início e fim do eixo X.

Fieldset Simulação: Este *Fieldset* se apresenta como uma composição de *inputs* de dados e até mais outros dois campos de dados: *Fieldset* Quantização Genérica e *Fieldset* Quantização Uniforme. Esta composição plena só é visível na página do Simulador. Inicialmente há um *input*, para a *Taxa de amostragem*, devidamente tratado para receber somente número.

Ainda compõe duas possibilidades iniciais de *checkbox* para simulação de *Sinal* e outro para a *Amostragem*. Com estes campos duas verificações ocorrem: a primeira é a de nenhuma simulação estar setada, a construção da figura ocorrerá, mas com um círculo amarelo, indicando a necessidade de setar alguma simulação, como pode ser verificado na Figura 2.13(a). A segunda é uma verificação da simulação pretendida: caso os valores parametrizados produzam vetores por demais extensos, o que comprometeria a visualização, tornando inútil a simulação, além de ocupar o servidor, travando a aplicação. Neste caso, a construção da figura ocorrerá, mas com um círculo vermelho, indicando a necessidade de revisão dos parâmetros. Na Figura 2.13(b) pode-se visualizar essa renderização.

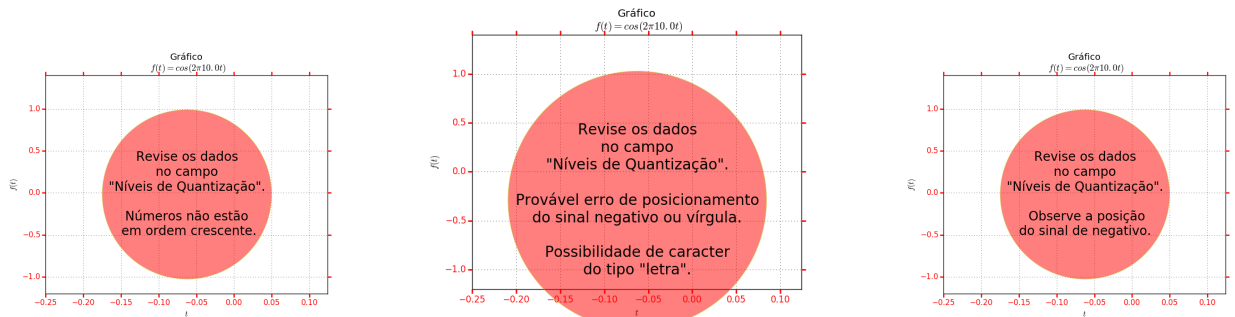
Fieldset Quantização Genérica: Dois campos e dois *checkbox* compõe este *fieldset*. O campo *Níveis de Quantização* verificará se os dados digitados são numéricos, separados por espaço em branco e em ordem crescente. Caso assim não for, a construção da figura ocorrerá, mas com um círculo vermelho e com mensagens de erro, como podem ser vistas na Figura 2.14.

O campo *Limiar inferior do nível* somente aceitará valores entre 0 e 1, com aviso ao usuário de incorreção e não permitindo o envio do formulário com dados incorretos.



(a) Mensagem de ausência de simulação. (b) Mensagem de parâmetros inadequados à simulação.

Figura 2.13: Tratamento de erros de não plotagem (a) e de vetores demasiado grandes (b).



(a) Mensagem de dados não ordenados de forma crescente. (b) Mensagem de dados não numéricos ou erros de posicionamento de vírgula ou sinal negativo. (c) Mensagem de um ou mais erros de posicionamento do sinal de negativo.

Figura 2.14: Tratamentos de erros do campo *Níveis de Quantização*.

Fieldset Quantização Uniforme: Um campo e quatro *checkbox* compõe este *fieldset*. O campo *Nº de Níveis de Quantização* somente aceitará valores inteiros entre 2 e 256, com aviso ao usuário de incorreção e não permitindo o envio do formulário com dados incorretos. Há mensagens de erro que surgem ao se inserir um valor inválido; apresentam-se como caixas de texto, com uma exclamação laranja. Um exemplo dessas mensagens está na Figura 2.15.

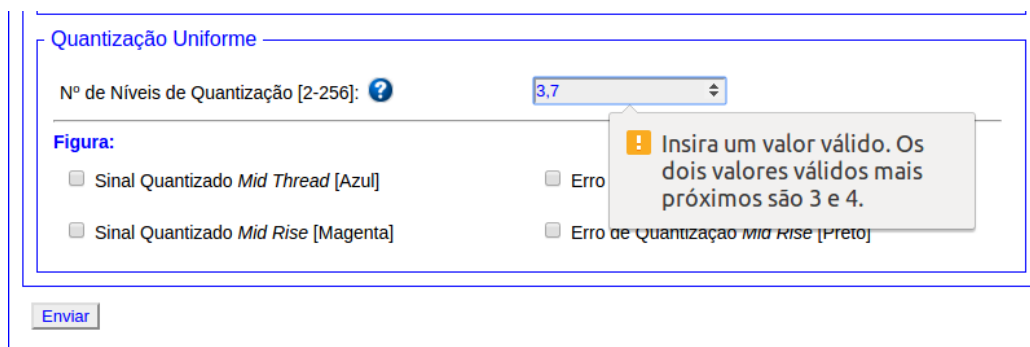


Figura 2.15: Mensagem de dado não aceito pelo campo.

3 *Desenvolvimento*

O código-fonte do *software* desenvolvido está disponível em <https://github.com/amilcarsimm/sigsis-tcc>.

Importa destacar que o momento anterior ao desenvolvimento, de elaboração de diretrizes, formato dos conteúdos e metodologias a serem utilizados, exposto no Capítulo 2, foi fundamental para estabelecer a etapa descrita neste capítulo.

3.1 **Tecnologias utilizadas no processo de desenvolvimento**

Foram utilizadas as seguintes tecnologias:

1. Linguagem *Python* e suas bibliotecas;
2. Linguagens de marcação (*HyperText Markup Language - HTML*), estilos (*Cascading Style Sheets - CSS*) e um mecanismo para uso de notações matemáticas (*MathJax*), utilizando \LaTeX ;
3. *Framework Django*;
4. Sistema Gerenciador de Banco de Dados (SGBD) *MySQL*;
5. Servidor *HTTP Nginx*.

3.1.1 **Breve descrição das tecnologias**

Linguagem *Python* e suas bibliotecas

Python combina uma grande robustez com uma sintaxe muito clara. Além disso, todos os lançamentos *Python* são *open source*, portanto não há problemas com licenciamento, além de termos vasta documentação disponível no *site* do desenvolvedor em <https://www.python.org/>. Trabalha-se neste projeto com a versão *Python 3*.

Através de suas bibliotecas de computação científica, há a possibilidade de desenvolver aplicações seguindo a lógica e sintaxe similares as que se trabalham em *software* proprietários como o MATLAB.

As diversas bibliotecas utilizadas possuem páginas específicas de seus desenvolvedores, na internet, as quais foram amplamente consultadas para a correta utilização. Destacamos as páginas de bibliotecas de computação científica:

1. **NumPy:** Pacote para computação científica com *Python*. *NumPy* adiciona uma ferramenta rápida e bastante sofisticada para manipulação de matrizes n-dimensionais com a linguagem *Python*. Site do desenvolvedor: <http://www.numpy.org/>.
2. **SciPy:** Ferramentas de computação científica para *Python* (*Scientific tools for Python*). Site do desenvolvedor: <https://www.scipy.org/>.
3. **Matplotlib:** Biblioteca para criação de gráficos e figuras de alta qualidade em uma variedade de formatos. Site do desenvolvedor: <http://matplotlib.org/>.

Linguagens de marcação (*HyperText Markup Language - HTML*) e estilos (*Cascading Style Sheets - CSS*)

A importância de utilizar os recursos do *HTML* e do *CSS* para produzir um conteúdo que atenda os critérios de usabilidade, conforme a Seção 2.4, foram dimensionados no início do projeto. As dúvidas sobre sintaxe dessas linguagens foram sanadas em <https://www.w3schools.com/>.

A ferramenta educacional se apresenta como um *site*, então o cuidado com a correta marcação e estilização dos elementos de hipertexto das páginas é de suma importância para a permanência do usuário. No tratamento de erros do simulador, descrito no tópico de mesmo nome, à página 34, os elementos do formulário corretamente parametrizados, possibilitaram boa parte das validações, ainda no lado cliente.

MathJax para as notações matemáticas

Faz-se necessário a notação matemática nas páginas da ferramenta. Para tanto optou-se por uma linguagem que convertesse a sintaxe \LaTeX para exibição em hipertexto. Após alguns testes optou-se pela tecnologia **MathJax**.

MathJax é um mecanismo de exibição *JavaScript*, *open source*, para \LaTeX , *MathML* e notação *AsciiMath*, que funciona em todos os atuais navegadores e não exige configuração por

parte do usuário (sem *plugins* para baixar ou *software* para instalar). Um simples comando *include* para *MathJax* na página *HTML* e a notação matemática são suficientes para uma boa visualização no navegador.

3.2 Framework Django

A utilização de *frameworks* é prática comum. Um *framework* é um conjunto de classes de uma determinada linguagem, que trabalham colaborativamente, possibilitando a utilização de várias aplicações diferenciadas. Com isso, a economia de trabalho é flagrante, permitindo uma concentração maior no projeto lógico, e não em aspectos de desenvolvimento básico de sistemas do tipo **CRUD** (acrônimo para as funções essenciais de todas as aplicações: *create*, *read*, *update* e *delete*).

O Django é um *framework Python*. Procurou-se instalar uma versão LTS (*Long Term Support*), e a escolhida foi a 1.8.11, com suporte até abril de 2018. Possui uma arquitetura dita **MTV**: *Model*, *Template* e *View*. É uma estrutura hierárquica onde os *templates* através de requisições do usuário, acessam a *view*, que por sua vez acessam a *model* que estabelece a ligação com o banco de dados.

O Django possui interface nativa para administração, que vem a ser muito positiva para a continuidade do projeto, com a inserção de novos módulos, roteiros e correções da página *index*, além de gerenciamento de usuários da ferramenta.

Sistema Gerenciador de Banco de Dados MySQL

Considerado o SGDB gratuito mais utilizado no mundo, o MySQL, por sua facilidade de gerenciamento, é o mais indicado para compor a ferramenta. O serviço utiliza a linguagem SQL (*Structure Query Language*) para inserir, acessar e gerenciar o conteúdo armazenado num banco de dados.

Servidor Nginx

Para o ambiente de produção, conforme descrição na Seção 3.4, o servidor de eleição é o Nginx, que tem sido apontado como rápido e leve.

Conforme a Wikipédia (2017):

“Segundo a pesquisa sobre servidores web (Web Server Survey), conduzido pela

Netcraft em outubro de 2015, o servidor Nginx se mostrou o segundo servidor web mais utilizado entre os sites ativos na web pesquisados, sendo utilizado por 15,33% destes. A pesquisa também indicou que o servidor é o mais utilizado entre os sites mais acessados da rede, com 23,66% dos sites utilizando a plataforma como base. De acordo com W3Techs, o servidor é utilizado por 29,7% dos sites no top 1 milhão de sites mais acessados da web, 39,5% dos sites no top 100 mil e por 47,6% dos sites no top 10 mil.

Em fevereiro de 2017, a adoção de Nginx foi:

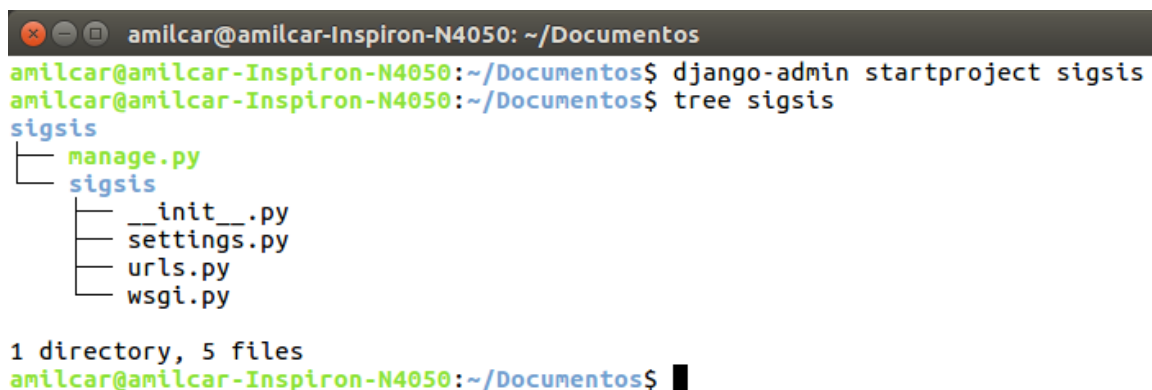
Brasil: 17,41% de todos os domínios.

Portugal: 17,62% de todos os domínios.”(WIKIPÉDIA, 2017)

3.2.1 Passos iniciais com o *framework* Django

Inicialmente, em ambiente de desenvolvimento, ou seja com o processo de desenvolvimento de *software* em uma computador local, instala-se uma série de ferramentas necessárias para este processo. No apêndice A temos os comandos necessários para tal, com a ligeira modificação que não se fará o *download* e sim se criará o projeto com as etapas dos próximos parágrafos. Como sistema operacional (SO), optou-se pelo Ubuntu 16.04, por ser uma versão LTS, com suporte até abril de 2024 (por oito anos), que já era utilizado pelo aluno.

Seguindo o tutorial do *site* do *Django Project* (<https://docs.djangoproject.com/en/1.8/intro/tutorial01/>), na pasta destinada ao projeto, que foi nomeado como *sigsis*, criou-se o mesmo através do comando `django-admin startproject sigsis`. A estrutura criada pode ser verificada conforme a saída do comando `tree` no terminal, exibida na Figura 3.1:



```
amilcar@amilcar-Inspiron-N4050: ~/Documentos
amilcar@amilcar-Inspiron-N4050:~/Documentos$ django-admin startproject sigsis
amilcar@amilcar-Inspiron-N4050:~/Documentos$ tree sigsis
sigsis
├── manage.py
└── sigsis
    ├── __init__.py
    ├── settings.py
    ├── urls.py
    └── wsgi.py

1 directory, 5 files
amilcar@amilcar-Inspiron-N4050:~/Documentos$
```

Figura 3.1: Visão da estrutura de diretórios e arquivos após a criação do projeto.

Neste estágio, é possível verificar uma estrutura já acessível através do navegador. Inicialmente na pasta do projeto, para esta versão do Django, executa-se o comando `python3`

`manage.py syncdb` para criação de um banco de dados provisório. Nesta etapa é criado um usuário administrativo, uma email para este usuário, e uma senha; optou-se por *admin*, *admin@admin.com* e *admin*, respectivamente. Inicia-se o servidor de testes, através do comando `python3 manage.py runserver` e então a interface com a mensagem “*It worked! Congratulations on your first Django-powered page.*” passa a ser exibida em `http://127.0.0.1:8000/` e também pode-se acessar a interface administrativa em `http://127.0.0.1:8000/admin`.

Cria-se a aplicação *web* ou o *app*, que após customizada possibilitará a visão do *site*, acessando a pasta onde está o arquivo *manage.py*, com o comando `python3 manage.py startapp core`. Isto criará um diretório *core* (entendido aqui como núcleo) e o projeto passa ter a seguinte estrutura de diretórios e arquivos, conforme a Figura 3.2:

```

amilcar@amilcar-Inspiron-N4050: ~/Documentos
amilcar@amilcar-Inspiron-N4050:~/Documentos$ cd sigsis/
amilcar@amilcar-Inspiron-N4050:~/Documentos/sigsis$ python3 manage.py startapp core
amilcar@amilcar-Inspiron-N4050:~/Documentos/sigsis$ cd ..
amilcar@amilcar-Inspiron-N4050:~/Documentos$ tree sigsis
sigsis
├── core
│   ├── admin.py
│   ├── __init__.py
│   ├── migrations
│   │   └── __init__.py
│   ├── models.py
│   ├── tests.py
│   └── views.py
├── manage.py
└── sigsis
    ├── __init__.py
    ├── __pycache__
    │   ├── __init__.cpython-35.pyc
    │   └── settings.cpython-35.pyc
    ├── settings.py
    ├── urls.py
    └── wsgi.py

4 directories, 13 files
amilcar@amilcar-Inspiron-N4050:~/Documentos$

```

Figura 3.2: Visão da estrutura de diretórios e arquivos após a criação do *app core*.

Assim o projeto básico provido pelo *framework* está criado, devendo-se desenvolver e customizar o restante. O projeto será uma coleção de *apps* e configurações que resultarão num *site* único.

O arquivo *models.py*

No arquivo *models.py* tem-se o código que gerará as tabelas do banco de dados. Assim seguir-se-á a estrutura já proposta na Seção 2.3, com módulos, que podem ter vários roteiros, além de uma página *index*. O simulador não gerará registro em banco de dados, por isso não necessitará de um *model*.

Os campos que essas tabelas terão já estão descritos no tópico *Interface de administrador*, à página 24: serão atributos de uma classe do tipo *models.Model*. Neste arquivo, através da

parametrização dos atributos e metadados da classe, também serão gerados atributos das *tags HTML*, a serem renderizados nas páginas de interface.

O arquivo *settings.py*

Com os *models* criados, deve-se incluir o *app core* nos *INSTALLED_APPS* e com o comando `python3 manage.py makemigrations core` as tabelas serão criadas.

Neste arquivo estarão outras configurações para o correto funcionamento da aplicação. Podem-se encontrar parametrizadas diversas ações, entre elas:

- Incluiu-se outros *apps* necessários à edição adequada dos textos a serem inseridos nos campos dos roteiros e página *index*. Também trabalhou-se com um *app* específico para customizar elementos do formulário do simulador.
- Definiu-se o MySQL como SGDB do projeto.
- Parametrizou-se o *app* para edição de texto nos *inputs* da interface administrativa, assim definindo quais seriam as suas opções.
- Definiu-se os *templates*, arquivos com códigos *HTML* exibidos aos usuários.

O arquivo *admin.py*

Através de modificações neste arquivo, serão customizadas as telas de administração do Django. As classes criadas através dos *models* serão aqui devidamente parametrizadas em suas exibições, permitindo buscas, rótulos de campos e demais itens que facilitem e tornem intuitiva a navegação.

Os arquivos *views.py* e *urls.py*

Nestes arquivos define-se os objetos que serão passados como dicionários nas *urls* e assim possibilitar o acesso a dados gravados no SGDB. Assim para requisitar uma *view*, esta terá que estar mapeada para uma *url*.

Nesta etapa, foram criadas as estruturas de exibição conhecidas como *templates*: arquivos *HTML*, com parâmetros para renderização adequada do que se pretende. Como já exposto no tópico *Interface de administrador*, à página 24, não criou-se os *templates* para esta interface, mas foi necessária sua customização.

Já para interface de usuário, tudo foi criado. Tem-se uma página *index*, uma para os roteiros, e outra para o simulador em todas as suas possibilidades. A estrutura para esta exibição é a seguinte:

1. Arquivo *base.html* com o código para o cabeçalho, barra lateral esquerda e rodapé, presentes em todas as páginas da ferramenta, já que estará como conteúdo estendido nos demais arquivos.
2. Arquivo *index.html* com as informações iniciais e boas vindas.
3. Arquivo *detalhe.html* com o código a ser inserido para a visualização dos elementos do roteiro (dados do módulo a que o roteiro faz parte, objetivos, subsídios, exercício e bibliografia).
4. Arquivo *simulador.html* com o *fieldset* ou campo de dados do simulador. Será incluído no arquivo *detalhe.html* caso o identificador do parâmetro *simulador* do roteiro seja diferente de zero, conforme decisão do criador do roteiro.
5. Arquivo *simulador-somente.html* que apresenta somente o simulador, mas com todos os campos.

3.3 O simulador

O simulador está implementado no arquivo *simulador.py*. Inicialmente cria-se uma classe do tipo *forms.Form*, na qual os campos do formulário, já descritos na Subseção 2.4.2, serão atributos desta classe. Neste arquivo, através da parametrização dos atributos e metadados, também serão gerados atributos das *tags HTML*, a serem renderizados nas páginas de interface.

Nos templates, nos arquivos *simulador.html* e *simulador-somente.html*, há uma combinação de estruturas condicionais para a exibição de elementos de simulação, conforme o que for definido pelo editor do roteiro. Foram implementados trechos com instruções *Python* encapsulados pelas *tags* “`{{ [código Python] }}`”, para transcrição de variáveis, ou “`{% [código Python] %}`”, para inserção de estruturas condicionais ou de repetição.

3.3.1 Estrutura do arquivo *simulador.py*

Além da criação da classe do tipo *forms.Form* com os atributos deste formulário, neste mesmo arquivo estão algumas funções para implementar as funcionalidades previstas pelo simulador.

A primeira função, denominada *get_simulador*, processa os dados do simulador, que ao serem enviados devem inicialmente ser validados. Este processo usualmente já será realizado previamente através do *HTML*, no lado cliente, mas também haverá a verificação no servidor. Após haverá a verificação do envio, para que ocorra sua *normalização* para um formato consistente. Isso significa dizer, por exemplo, que o dado enviado de um campo do tipo *FloatField*, será ajustado para um dado *Python* do tipo *float*. O método que realiza esta *normalização*, chama-se *cleaned_data*.

Nesta primeira função, é verificado se o valor de início do eixo X é maior que seu fim. Se for, o processamento já retorna à página para indicação de erro, conforme pode ser verificado na Figura 2.12.

A segunda função, denominada *constroi_onda_amostras*, constrói os vetores para as diversas possibilidades de ondas e também para suas amostras.

A terceira função, denominada *verifica_niveis*, trabalha o campo *Níveis de Quantização*, que por ser um campo de texto, necessita de uma conversão para uma lista de valores do tipo *float*. Esta função se ocupa especialmente dos possíveis erros de preenchimento, já abordados no tópico *Tratamento de erros do simulador*, à página 34.

A quarta função, denominada *quantizacao_niveis*, construirá os vetores com os limiares dos níveis de quantização a serem utilizados na função seguinte.

A quinta função, denominada *quantizacao_vetor*, construirá os vetores para as possibilidades de quantização e de seus erros. Seis vetores possuem ali as diretrizes para sua construção.

A sexta função, denominada *titula_grafico*, ocupa-se de determinar os títulos que serão usados no gráfico.

E a sétima função, denominada *showimage*, cria a imagem com o gráfico que será exibido nas páginas de roteiros.

Implementação do simulador

O simulador foi implementado de modo a atender as necessidades do módulo sobre Amostragem e Quantização.

Procurou-se inicialmente os elementos para a formação básica de um sinal, a ser representado no tempo, onde o eixo X, ou das abcissas, representaria o tempo decorrido, e o Y, ou das ordenadas, representaria a amplitude do sinal: o típico gráfico de forma de onda. A função pretendida tem como forma geral:

$$f(t) = A + Bs(\omega(t + D)) \quad (3.1)$$

onde:

- A é a componente de tensão contínua (*DC - direct current*);
- B é a amplitude do sinal;
- s é a forma de onda pretendida (cosseno, seno, etc.);
- ω é a velocidade angular que é dada por $\omega = 2\pi f$, sendo f a frequência fundamental;
- D é o deslocamento no tempo.

Assim para o **sinal** tem-se a necessidade de parametrizar esses cinco componentes.

Atenta-se que para a construção da função em *Python* foi necessária adaptação da Função (3.1) para contemplar a criação do vetor do sinal e das suas amostras. No caso de uma função cosseno, por exemplo, ficaria assim:

$$s = \text{float}(dc) + \text{float}(ampl) * \cos(2\pi \text{float}(freq) * (t + \text{float}(desl))) \quad (3.2)$$

onde:

- $\text{float}(dc)$ é a componente de tensão contínua (*DC - direct current*);
- $\text{float}(ampl)$ é a amplitude do sinal;
- \cos é a forma de onda pretendida (no caso, cosseno);
- $\text{float}(freq)$ é a frequência fundamental;
- t é um vetor com início e fim do eixo X, e um valor para um espaçamento uniforme dos diversos valores compreendidos entre esses limites;
- $\text{float}(desl)$ é o deslocamento no tempo.

A variável t é dada pela função *NumPy arange*. A função *arange* cria um vetor de valores em um intervalo com início e fim dos dados, espaçados de maneira uniforme. Este espaçamento, é dado em função do período da onda, ou seja, o tempo para uma oscilação, que corresponde ao inverso da frequência fundamental. Para o sinal utiliza-se a variável t , e para as amostras,

$t2$. Em t multiplicou-se o período por $1/100$, já que será *plotado* o sinal e pretende-se uma visualização de continuidade entre os pontos. Já para $t2$ utilizou-se o período como sendo o inverso da taxa de amostragem, por se utilizar a opção *stem*, para a visualização dos impulsos gerados ao longo da amostra da onda. A sintaxe para as funções, ficou assim:

Para t :

$$\begin{aligned} t &= 1/(100 * \text{float}(\text{freq})) \\ t &= \text{arange}(t_start_c, t_stop_c, t_step) \end{aligned} \quad (3.3)$$

onde:

- t_step é o *passo*, o intervalo, entre um elemento do vetor e o seguinte;
- $\text{float}(\text{freq})$ é a frequência fundamental;
- t é a variável que será populada com o vetor criado;
- t_start_c é o início do eixo e o primeiro elemento do vetor;
- t_stop_c é o fim do eixo e o último elemento do vetor.

Para $t2$:

$$t2 = \text{arange}(t_start_c, t_stop_c, 1/fs_c) \quad (3.4)$$

similar à sintaxe (3.3), onde fs_c é taxa ou frequência de amostragem.

No âmbito da **área de plotagem** deste sinal, teremos a definição do eixo das abcissas, com a determinação da fração de tempo a ser visualizada, com seu início e fim. É boa prática que se titule o gráfico, incluindo a função a ser desenhada. Também que se identifique os dois eixos.

Para as **amostras**, há que se estabelecer a taxa ou frequência de amostragem. É o que basta além das parametrizações anteriores.

No caso das quantizações, optou-se por quantizar as amostras recolhidas do sinal, e a partir daí, as informações necessárias seriam a dos níveis de quantização e seus limiares, que podem variar, conforme as formas usuais de quantização. Os dois novos parâmetros estabelecerão um novo vetor com valores no eixo Y.

Na **quantização uniforme**, estabelece-se que o limiar de quantização seja em 50%, ou seja, em cada nível haverá uma nova divisão, no ponto médio entre os dois valores limites. Portanto é suficiente que se estabeleça quantos níveis terá o eixo das ordenadas. Resta ainda decidir pelas duas formas possíveis: *mid tread* e *mid rise*. A diferença está em que na primeira, um dos níveis

cruza a região do zero e na segunda, isso não acontece, podendo os níveis adjacentes estarem a distância de um passo/2.

Por fim, na **quantização genérica**, os níveis deverão ser discriminados pelo usuário. O limiar de quantização também, ou seja, avaliando a partir do limite inferior do nível, o usuário fará a escolha de uma fração de 1, que indicará um percentual para o limite inferior e o restante comporá a fração superior. Portanto será necessário dois *inputs* para esses parâmetros. Da mesma forma que na quantização uniforme, nivela-se para baixo as amostras que estão entre um nível e o limiar determinado pelo usuário, e para cima, aquelas que estiverem entre a outra porção e o nível seguinte; caso a amostra esteja exatamente no limiar, é nivelada para baixo.

3.4 Ambiente de produção

O ambiente de produção é onde a ferramenta será executada (produzida) após a fase de desenvolvimento. A prática no IFSC-SJ para ferramentas *web* consiste na criação de uma máquina virtual (*Virtual Machine - VM*), para hospedagem da aplicação desenvolvida, com as seguintes configurações: 1 processador de 2 núcleos, 2 GB de RAM e 32 GB de disco, com uma única partição / com *swap* em arquivo. Foram liberadas as portas HTTP, HTTPS e SSH.

Na VM foram instalados os pacotes necessários para a execução do projeto, conforme o Apêndice A.

Atente-se que foi utilizado a tecnologia de *virtualenv*, ou seja, um ambiente isolado para este projeto, com as especificações de versão do *Python* e bibliotecas.

Essencialmente, para que tenha-se o Nginx atendendo requisições HTTP e fazendo *proxy* reverso com a aplicação *Python*, foi necessário que se configurasse o “Serviço Django” (*Serviço* é entendido como uma ferramenta que executa em segundo plano e providenciará funcionalidades para a rede interna e/ou externa) combinado em dois arquivos: */etc/init/django.conf*, que chama o *script run.sh*, na pasta do arquivo do projeto sigsis. Para *rodar* o Serviço Django usa-se o comando `sudo start django` e para encerrá-lo, `sudo stop django`.

4 Conclusões

O presente trabalho visou a construção de uma ferramenta educacional centrada no aluno. Definiu-se a elaboração de uma plataforma web, mas com um diferencial dos atuais ambientes virtuais de aprendizagem.

Então, pensou-se em uma estrutura de construção modular, composta de roteiros, com objetivos gerais e específicos claramente definidos, promovendo a avaliação do conteúdo trabalhado. Também ofertou-se a possibilidade de incluir um simulador ou não.

Para o módulo construído, a simulação produz um gráfico de forma de onda, suas amostras e as possíveis quantizações e erros, num processo de modulação por código de pulso. Também há a possibilidade de se *plotar* a Transformada de Fourier do sinal. A indicação de exercícios e bibliografia complementam a proposta. Novos módulos com novos temas e novas simulações são passíveis de ser desenvolvidos, em qualquer área, dando a possibilidade de ampliação e revisão modular permanente.

Para este desenvolvimento, a prévia reflexão sobre *software* educacional, motivou a pesquisa de diversas visões sobre o assunto. Constatou-se uma preocupação pela forma que se tem conduzido a formação de professores da área de Engenharia e o reflexo em seus alunos, através dos artigos da Revista de Ensino e Engenharia, citados neste TCC (Cunha (2015) e Alves e Mantovani (2016)). As avaliações partem dos próprios discentes e de especialistas em educação; na bibliografia consultada, também se verificou que há avaliações do próprio IFSC (Andrade e Pereira (2012)) e de outras instituições de ensino brasileiras (Alves e Mantovani (2016)), o que as tornam relevantes num cenário local. A aplicação dos paradigmas da complexidade destacam-se como proposta de revisão do processo ensino–aprendizagem.

A utilização dos conceitos de usabilidade permitiu um desenvolvimento voltado para a experiência do usuário, de modo a promover sua permanência no *site*–ferramenta.

A etapa de desenvolvimento possibilitou uma visão do *framework* Django e suas possibilidades, elaboração de materiais (sejam eles textos ou figuras), não só para uma monografia, mas também para serem disponibilizados na *web*. Ao final, tem-se um produto hospedado no

Servidor *HTTP* do IFSC-SJ, além deste texto.

Em relação a trabalhos futuros, há a possibilidade de inclusão de outros temas de Comunicação Digital e quais mais forem de intenção dos utilizadores. A inserção de conteúdos implicaria em desenvolvimento de novos simuladores e as adequações nos *templates* já existentes.

Considerando as boas práticas gerais, sugere-se:

- Implementar a verificação dos campos, de forma similar ao que já foi feito no lado servidor, também no lado cliente, utilizando *JavaScript*.
- Adequar a ferramenta para que se torne responsiva, ou seja, adapte-se automaticamente à largura de tela do dispositivo que a acessa.
- Revisar as questões que possam surgir a partir do uso da ferramenta e adequar seu *layout*, para atender as boas práticas de usabilidade.
- Otimizar o código, para que se torne mais elegante e claro, para aqueles que por ele se interessarem.
- Produzir mais conteúdo multimídia para tornar a ferramenta mais atrativa e garantir a permanência do usuário no *site*.
- Implementar uma forma de impressão das páginas dos roteiros, com *layout* específico, para uma possível composição de apostila.

APÊNDICE A – Passo a passo de instalação do sigsis

Lembra-se que estes passos são para o SO Ubuntu 16.04 LTS. Inicialmente faz-se o *download* do projeto sigsis em <https://github.com/amilcarsimm/sigsis-tcc>. Descompacta-se no diretório de preferência; sugere-se a pasta *home*. Uma série de comandos serão executados no terminal:

1. Instalar pacotes do sistema com os seguintes comandos:

```
sudo apt install mysql-server mysql-client
```

(Neste momento será criado o usuário *root* do MySQL. Com a ferramenta *mysql-client* cria-se um usuário *admin* com senha *admin*. Autenticando-se como este usuário, cria-se um banco de dados com o nome ‘sigsis’. Na pasta sigsis executa-se o comando `python3 manage.py syncdb`, e na criação do usuário, utilizar o login ‘admin’, senha ‘admin’ e email ‘admin@sigsis.com’, respectivamente.)

```
sudo apt install python-virtualenv
```

```
sudo apt install libfreetype6-dev libpng-dev libxft-dev
```

```
sudo apt install libmysqlclient-dev
```

```
sudo apt install python3-pip
```

2. Cria-se o *virtualenv* e ativa-se. No primeiro diretório do projeto sigsis executa-se o comando:

```
virtualenv -p /usr/bin/python3 venv
```

```
source ../venv/bin/activate
```

3. Atualiza-se a ferramenta pip através do comando:

```
python3 -m pip install --upgrade pip
```

(Conforme <https://stackoverflow.com/questions/26575587/>)

4. Instala-se os pacotes *Python 3* assinalados no arquivo *requirements.txt*, executando o comando:

```
sudo pip3 install -r requirements.txt
```

(Conforme https://pip.pypa.io/en/stable/user_guide/#requirements-files)

5. Em *sigsis/sigsis/settings.py*, editar a linha 22, colocando o caminho onde estará a pasta do sistema na máquina local.

6. Para executar o Django:

- Com o servidor de testes:

```
python3 manage.py runserver
```

- Com o Nginx. Instala-se o servidor com o comando:

```
sudo apt-get install nginx
```

Edita-se o arquivo */etc/nginx/sites-enabled/default*, que passa a ter o seguinte conteúdo:

```
server {
    listen 80 default_server;
    listen [::]:80 default_server ipv6only=on;

    root /usr/share/nginx/html;
    index index.html index.htm;

    server_name _;

    location / {
        proxy_pass http://127.0.0.1:8000/;
    }
}
```

Cria-se o arquivo */etc/init/django.conf*, com o seguinte conteúdo (corrigindo o nome do usuário e o *path* para o arquivo *run.sh*):

```
# Django
description "Django"
```

```
start on runlevel [2345]
```

```
stop on runlevel [!2345]
```

```
respawn
```

```
respawn limit 10 5
```

```
umask 022
```

```
exec su - <usuário> -c "/path/para/o arquivo/run.sh"
```

No servidor com Ubuntu 14.04, inicia-se o “Serviço Django” com o comando `start django`, e o encerramos com `stop django`, Em uma máquina com Ubuntu 16.04, o serviço deverá ser inicializado com o comando `exec su - <usuário> -c "/path/para/o arquivo/run.sh"`.

7. Acessar através de um navegador, <http://localhost:8000/>.

APÊNDICE B – Roteiros de Amostragem e Quantização

Neste apêndice constarão todos os roteiros de estudo que também estão disponíveis na ferramenta.

Ressalta-se que um módulo anterior de revisão de Sinais e Sistemas é recomendado. As bibliografias sobre Amostragem e Quantização incluem estas revisões em capítulos anteriores. Haykin (2004), por exemplo, trabalha em apêndices, como “material suplementar”, teoria das probabilidades, representação de sinais e sistemas e outros temas necessários ao estudo de Comunicações Digitais. Já Lathi (2007) lista diversos temas pertinentes a este módulo em um momento denominado *Background*, antes dos capítulos propriamente ditos.

Estuda-se um modelo de Comunicação Digital, conforme a Figura B.1 baseada em Gallager (2008):

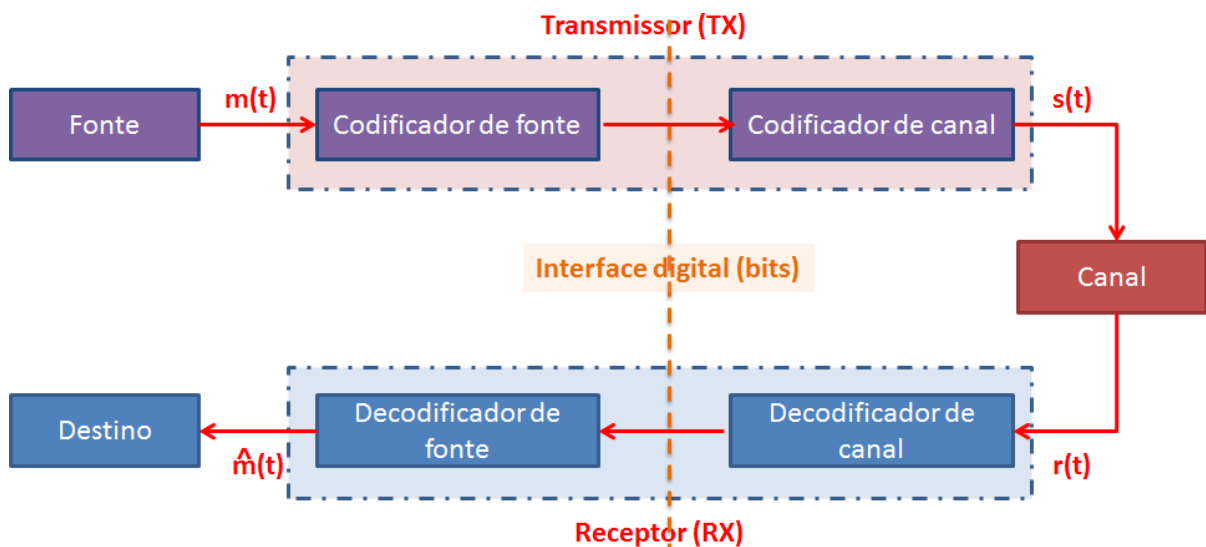


Figura B.1: Modelo de Comunicação Digital.

Neste modelo, o tema “Amostragem e quantização” está inserido na etapa do Codificador e Decodificador de fonte. Propõe-se que compreenda o estudo dos seguintes tópicos:

- Teorema da amostragem;
- Quantização uniforme e não-uniforme;
- Modulação por codificação de pulso (PCM).

Os sete roteiros a seguir desenvolvem os tópicos já citados. Demonstrar-se-á as possibilidades do simulador, acrescentando a cada roteiro uma das opções possíveis do campo “Simulador do roteiro”, na interface de administração do site, na opção “Core” > “Roteiro”, conforme a Tabela B.1:

Nº do roteiro	Título	Opção de simulação
0	Um modelo de Comunicação Digital	Sinal
1	Amostragem	Amostras
2	Teorema da amostragem	Amostras
3	Quantização e Codificação	Quantizador Genérico
4	Quantização Uniforme	Quantizador Uniforme
5	Quantização Não-Uniforme	Quantização Genérico
6	Modulação por codificação de pulso – PCM	Sem Simulador

Tabela B.1: Tabela de roteiros e opções de simulação para o Módulo “Amostragem e Quantização”.

B.1 Roteiro 0: Um modelo de Amostragem e Quantização

B.1.1 Objetivo geral do roteiro

Introduzir o módulo de Amostragem e Quantização.

B.1.2 Objetivos específicos do roteiro

1. Delimitar a abrangência do módulo de Amostragem e Quantização.
2. Enumerar as etapas do modelo de Amostragem e Quantização.
3. Introduzir a ferramenta de simulação.

B.1.3 Subsídios do Roteiro

Pode-se perguntar o porquê das comunicações digitais estarem tão comuns na atualidade. Pode-se apontar algumas características que tornam o digital, objeto de eleição nas comunicações:

1. Maior imunidade ao ruído;
2. Maiores distâncias a serem cobertas através de repetidores regenerativos;
3. Maior longevidade pelo armazenamento digital;
4. Interconexão de sistemas por interface digital;
5. Processamento digital (*hardware* e *software*);
6. Permite novas técnicas de multiplexação.

Algumas dessas características serão melhores compreendidas ao longo dos roteiros e módulos de estudo.

Um modelo de Comunicação Digital envolve a transmissão de sinais. Um sinal analógico para ser transmitido, deverá ser convertido em um sinal digital, e ao chegar a seu destino deverá ser convertido novamente para um sinal analógico.

Para isso o sinal analógico deverá passar por alguns processos: o primeiro será a **amostragem**, após será feita a **quantização** e a seguir teremos a **codificação**. Será transmitido por

um canal, e então sofrerá a **decodificação**, seguida de sua **reconstrução**, chegando a seu destino. Portanto, tem-se uma fonte que emite sinais analógicos, e é necessária uma conversão analógico-digital (AD) no transmissor e uma conversão digital-analógico (DA) no receptor, conforme demonstrado na Figura B.2.

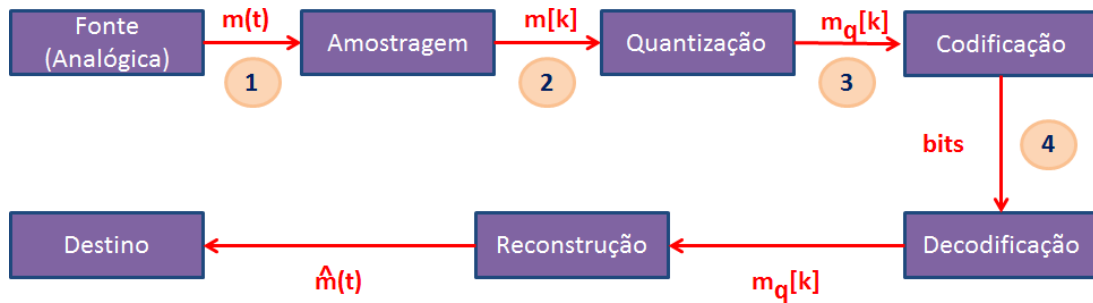


Figura B.2: Processo de conversão analógico-digital e digital-analógico.

As numerações na Figura B.2 possuem o seguinte significado:

1. Sinal real: tempo contínuo / amplitude contínua;
2. Sequência de números reais: tempo discreto / amplitude contínua;
3. Sequência de números reais: tempo discreto / amplitude quantizada;
4. Sequência de *bits*.

Ou de outra forma, conforme demonstrado na Figura B.3:

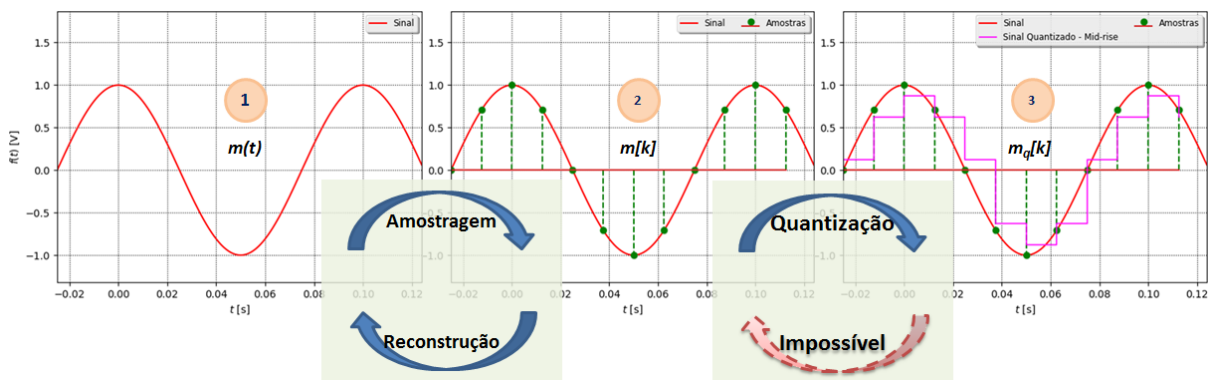


Figura B.3: Esquema gráfico do processo de amostragem e quantização.

Observa-se que a etapa de amostragem permite um retorno, e de ter-se a reconstrução exata do sinal original. Já o mesmo não será possível após a quantização. Cada uma destas etapas será desenvolvida nos roteiros seguintes.

B.1.4 Bibliografia do roteiro

1. HAYKIN, S. Sistemas de comunicação. Bookman, 2004. Cap. Fundamentos e Visão Geral. p. 38-39.

B.1.5 Exercícios do roteiro

1. Sugere-se revisão de conceitos como: sinais discretos e contínuos, frequência fundamental de ondas, período e deslocamento no tempo.
2. Exercite a construção de gráficos de tipos de onda no simulador e procure identificar o período dos sinais nas figuras.

B.2 Roteiro 1: Amostragem

B.2.1 Objetivo geral do roteiro

Introduzir o conceito de amostragem de sinais contínuos no tempo.

B.2.2 Objetivos específicos do roteiro

1. Conceituar amostragem.
2. Determinar o processo de obtenção de amostras.
3. Definir intervalo e taxa de amostragem.
4. Estimular o uso da ferramenta de simulação.

B.2.3 Subsídios do Roteiro

A ideia de amostragem preconiza que um sinal em tempo contínuo pode ser reconstruído a partir de suas amostras em tempo discreto. Para isso se associa um chaveamento, e a partir daí obtêm-se amostras a um determinado intervalo de tempo. Essas amostras devem ser de tal forma suficientes que permitam se reconstruir o sinal sem erro, ou com uma tolerância a ser determinada.

O esquema da Figura B.4 demonstra o processo de amostragem e reconstrução, associado a obtenção de amostras mediante um chaveamento:

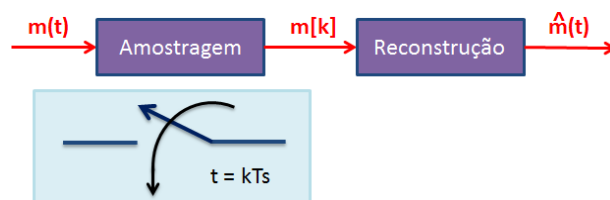


Figura B.4: Esquema do processo de amostragem e reconstrução, associado a obtenção de amostras mediante um chaveamento.

Admita-se a seguinte notação:

- $m(t)$: sinal a ser amostrado;
- $m[k] = m(kT_s)$: sequência de amostras;

- $\widehat{m}(t)$: sinal reconstruído a partir das amostras.

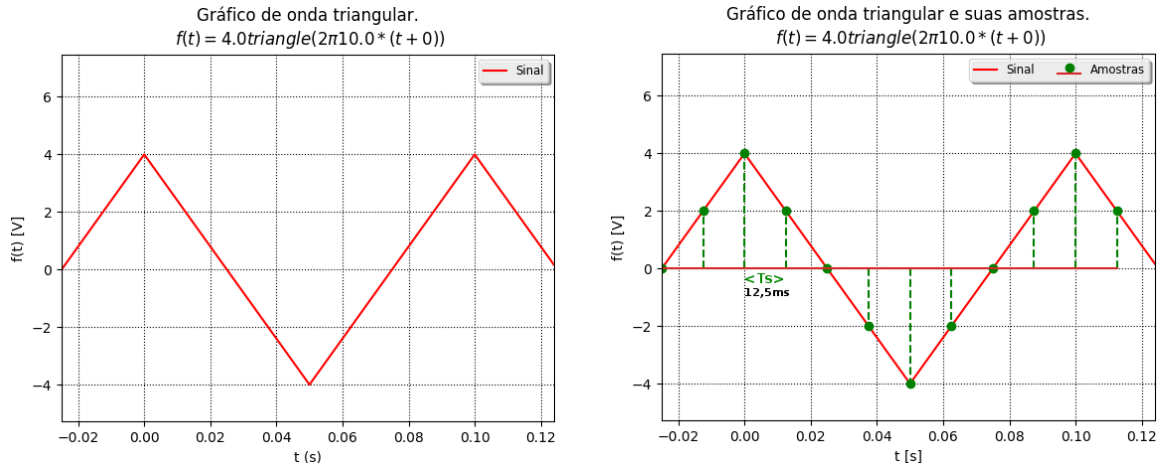
Então a cada fração de tempo, dada como T_s , uma amostra será obtida. Com a definição desse intervalo de amostragem (T_s), obtém-se a taxa ou frequência de amostragem (f_s), pelo seu inverso:

- T_s : intervalo de amostragem (s);
- $f_s = \frac{1}{T_s}$: taxa de amostragem (amostras/s \equiv Hz).

As unidades de medida serão segundo para T_s e amostras por segundo, que equivale a Hertz (Hz), para a taxa de amostragem (f_s).

Exercício resolvido: Seja $m(t)$ uma onda triangular de amplitude 4 V e frequência fundamental de 10 Hz. Determine a sequência de amostras, sabendo que a taxa de amostragem é de $f_s = 80$ Hz e que a primeira amostra ocorre em $t = 0$.

Conforme a Figura B.5, é proposta a seguinte **solução**:



(a) Onda triangular de amplitude 4 V e frequência fundamental de 10 Hz.

(b) Onda triangular e suas amostras.

Figura B.5: Representação de onda triangular e amostras.

$$f = 10 \text{ Hz} \Rightarrow T = 100 \text{ ms} \quad (\text{período fundamental de } m(t))$$

$$f_s = 80 \text{ Hz} \Rightarrow T_s = 12,5 \text{ ms}$$

Resposta:

$$\text{Sequência de amostras: } m[k] = \left[\binom{k=0}{4}, \binom{k=1}{2}, \binom{k=2}{0}, \binom{k=3}{-2}, \binom{k=4}{-4}, \binom{k=5}{-2}, \binom{k=6}{0}, \dots \right]$$

B.2.4 Bibliografia do roteiro

1. HAYKIN, S. Sistemas de comunicação. Bookman, 2004. Cap. 03, p. 206-207.
2. LATHI, B. P. Sinais e Sistemas Lineares, [S.l.]: Bookman, 2007. Cap. 08, p. 678.

B.2.5 Exercícios do roteiro

1. Utilize o simulador para parametrizar um gráfico de onda para o sinal do exemplo acima. Parametrize também as amostras.
2. Verifique o que acontece ao diminuir para a metade a taxa de amostragem. Qual o período de amostragem? Qual seria a sequência de amostras?
3. Verifique as outras possibilidades de formas de onda, para estes mesmos parâmetros.

B.3 Roteiro 2: Teorema da amostragem

B.3.1 Objetivo geral do roteiro

Apresentar o Teorema da Amostragem.

B.3.2 Objetivos específicos do roteiro

1. Enunciar o Teorema da Amostragem.
2. Verificar a fórmula de interpolação de Whittaker-Shannon.
3. Exercitar o cálculo de largura de banda e o esboço de espectro de um sinal.

B.3.3 Subsídios do Roteiro

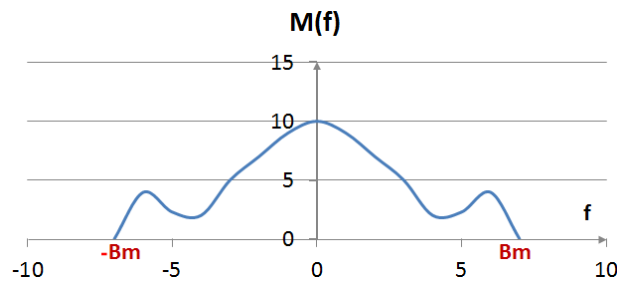
$$\left(\begin{matrix} \textit{Wittaker} \\ 1915 \end{matrix} \right) - \left(\begin{matrix} \textit{Nyquist} \\ 1928 \end{matrix} \right) - \left(\begin{matrix} \textit{Kotelnikov} \\ 1933 \end{matrix} \right) - \left(\begin{matrix} \textit{Shannon} \\ 1949 \end{matrix} \right)$$

O Teorema da Amostragem foi gradualmente desenvolvido por mais de um cientista, conforme a cronologia acima, e é também conhecido como Teorema de Nyquist-Shannon, ou Teorema de Nyquist.

Conforme Lathi (2007, p. 678), “Um sinal em tempo contínuo pode ser processado, a partir de suas amostras, por um sistema que opere em tempo discreto.” É enfatizada a necessidade desta taxa de amostragem permitir a reconstrução sem erro do sinal original. Para definir-se qual é esse quantitativo, estabelece-se o **Teorema da Amostragem**.

A menor frequência de amostragem (f_s) é dada como o dobro da largura de faixa absoluta do sinal (B_m), ou seja, $f_s = 2B_m$ [Hz]. Este Teorema se aplica a sinais passa-baixa. Em sinais passa-banda o esquema pode ser um pouco mais complexo. Essa taxa de amostragem é dita Taxa de Nyquist e é admitida como a menor taxa de amostragem para preservar a informação do sinal original.

Assim, seja $m(t)$ um sinal e $M(f) = \mathcal{F}\{m(t)\}$ sua transformada de Fourier. Suponha que $M(f) = 0$ para todo $|f| > B_m$, conforme a Figura B.6:

Figura B.6: Sinal $M(f)$.

Então, o sinal $m(t)$ pode ser recuperado perfeitamente a partir de suas amostras $m[k] = m(kT_s)$ se e somente se

$$f_s > 2B_m \quad (\text{B.1})$$

onde $f_s = \frac{1}{T_s}$. Nesse caso:

$$m(t) = \sum_{k=-\infty}^{\infty} m[k] \operatorname{sinc}\left(\frac{t}{T_s} - k\right) \quad (\text{B.2})$$

O termo $f_{Nyq} \triangleq 2B_m$ é conhecido como **taxa de Nyquist** e a fórmula (B.2) é conhecida como **fórmula de interpolação de Whittaker-Shannon**. Importa ressaltar que utiliza-se a definição de Haykin (2004, p. 805): $\operatorname{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$.

Para aplicação do teorema, sugere-se uma revisão de conhecimentos sobre Transformada de Fourier. Apresenta-se alguns pares de transformadas na Tabela B.2. O símbolo \Leftrightarrow indica o par, a coluna $m(t)$ indica a função no tempo e a coluna $M(f)$, a função na frequência correspondente.

Nº	$m(t)$	\Leftrightarrow	$M(f)$
1	$\delta(t)$	\Leftrightarrow	1
2	1	\Leftrightarrow	$\delta(f)$
3	$\delta(t - t_o)$	\Leftrightarrow	$e^{-j2\pi f t_o}$
4	$e^{j2\pi f t_o}$	\Leftrightarrow	$\delta(f - f_o)$
5	$\cos(2\pi f_o t)$	\Leftrightarrow	$\frac{1}{2}\delta(f + f_o) + \frac{1}{2}\delta(f - f_o)$
6	$\sin(2\pi f_o t)$	\Leftrightarrow	$-\frac{1}{2j}\delta(f + f_o) + \frac{1}{2j}\delta(f - f_o)$
7	$\operatorname{rect}\left(\frac{t}{\tau}\right)$	\Leftrightarrow	$\tau \operatorname{sinc}(\tau f)$
8	$\operatorname{sinc}\left(\frac{t}{\tau}\right)$	\Leftrightarrow	$\tau \operatorname{rect}(\tau f)$
9	$\operatorname{tri}\left(\frac{t}{\tau}\right)$	\Leftrightarrow	$\tau \operatorname{sinc}^2(\tau f)$
10	$\operatorname{sinc}^2\left(\frac{t}{\tau}\right)$	\Leftrightarrow	$\tau \operatorname{tri}(\tau f)$

Tabela B.2: Tabela de pares transformados.

Exercício resolvido:

Esboçar o espectro e determinar a largura de banda do sinal $m(t) = \text{sinc}(200t)$. Em seguida, determine a mínima taxa de amostragem de modo a ser possível recuperar perfeitamente o sinal a partir de suas amostras.

Inicialmente verifica-se que $m(t) = \text{sinc}(200t) = \text{sinc}\left(\frac{t}{\frac{1}{200}}\right)$

Verificando a Tabela B.2, nº 10, tem-se que:

$$M(f) = \frac{1}{200} \text{rect}\left(\frac{1}{200}f\right) = \frac{1}{200} \text{rect}\left(\frac{f}{200}\right)$$

O esboço do espectro da função $M(f) = \frac{1}{200} \text{rect}\left(\frac{f}{200}\right)$ pode ser visto na Figura B.7. Considera-se para determinação da banda somente os valores positivos, assim $B_m = 100$ Hz; $f_s = 2B_m = 200$ amostras/s.

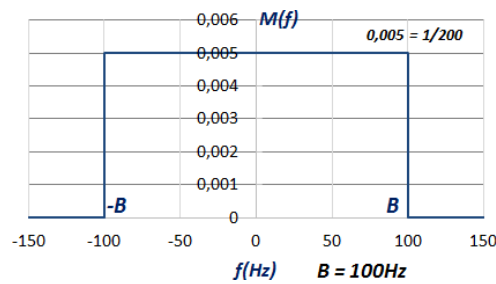


Figura B.7: Esboço do espectro do sinal $m(t) = \text{sinc}(200t)$.

B.3.4 Bibliografia do roteiro

1. HAYKIN, S. Sistemas de comunicação. Bookman, 2004. Cap. 03, p. 206-209 e p. 805.
2. LATHI, B. P. Sinais e Sistemas Lineares, [S.l.]: Bookman, 2007. Cap. 08, p. 678-685.

B.3.5 Exercícios do roteiro

Esboce o espectro e determine a largura de banda e a taxa de Nyquist dos seguintes sinais:

1. $m(t) = \text{sinc}^2(200t)$ (tente simular com o início do eixo em -0,002 s e final em 0,002 s)
2. $m(t) = \text{sinc}(200t) + \text{sinc}^2(200t)$
3. $m(t) = 2 + 600\text{sinc}(200t) + 8\cos(2\pi 50t)$

O simulador está com valores pré fixados. Verifique se a taxa de amostragem de 40 Hz, para um sinal $m(t) = \cos(2\pi 5t)$, atende a taxa de Nyquist. Verifique também para um sinal com uma onda triangular, com estes mesmos parâmetros.

B.4 Roteiro 3: Quantização e Codificação

B.4.1 Objetivo geral do roteiro

Verificar o processo de quantização.

B.4.2 Objetivos específicos do roteiro

1. Definir o processo de quantização.
2. Entender a necessidade de quantizar as amostras.
3. Estabelecer os parâmetros necessários para um quantizador.
4. Definir erro de quantização.

B.4.3 Subsídios do Roteiro

Para a conversão analógico para digital, além da amostragem, é necessária a **quantização**. Isto se dá pelo fato de que um sinal analógico pode assumir infinitos valores de amplitude, mas um sinal digital não. Assim é necessário “arredondar” as amostras pra que se enquadrem num número finito de valores. Este arredondamento é o processo de quantização. Os valores para os quais as amostras serão arredondadas são denominados **níveis de quantização**. Haykin (2004) ressalta que:

“Qualquer sentido humano (a audição ou a visão) como receptor final pode detectar somente diferenças de intensidades finitas. Isso significa que o sinal contínuo original pode ser *aproximado* por um sinal construído de amplitudes discretas selecionadas de um conjunto disponível, baseadas no mínimo erro.”(HAYKIN, 2004, p. 215)

Assim o sinal quantizado poderá ser praticamente indistinguível do sinal original, dependendo da aplicação.

Questiona-se quantos níveis (pulsos) distintos são a melhor opção? Segundo Lathi (2007):

“Do ponto de vista prático, trabalhar com uma quantidade muito grande de pulsos distintos é muito difícil. Prefere-se utilizar a menor quantidade de pulsos distintos, sendo o menor número possível igual a dois. Um sinal digital utilizando apenas

dois símbolos ou valores é o sinal binário. Um sinal digital binário (um sinal que só pode assumir dois valores) é muito desejável, devido à sua simplicidade, economia e facilidade de trabalho. Podemos converter um sinal *L-ário* em um sinal binário utilizando codificação de pulso.” (LATHI, 2007, p. 697)

Assim, para L níveis de quantização, serão necessários um mínimo de b dígitos do código binário, tal que $2^b = L$ ou $b = \log_2 L$.

Parâmetros de um quantizador:

- L : número de níveis de amplitude usados no quantizador;
- $v_0, v_1, v_2, \dots, v_{L-1}$: níveis de quantização;
- $\lambda_0, \lambda_1, \lambda_2, \dots, \lambda_L$: limiares de quantização ou decisão.

Num modelo com níveis de quantização e seus limiares estão representados os níveis como v_l (pela letra grega *nu*) e seus limiares como λ_l (pela letra grega *lambda*), conforme a Figura B.8. Já o espaçamento entre dois níveis adjacentes é chamado *quantum* ou *tamanho do degrau*.

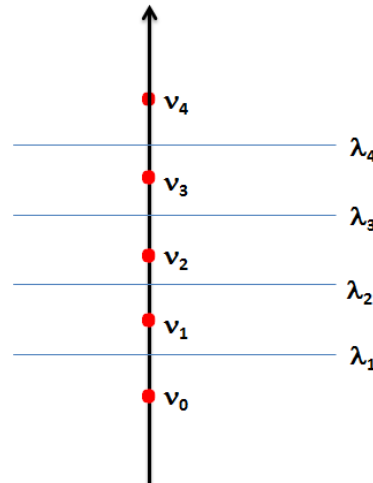


Figura B.8: Modelo com níveis de quantização e seus limiares.

Admitindo-se a seguinte notação:

- $m(t)$: sinal de tempo contínuo e amplitude contínua;
- $m[k]$: sinal de tempo discreto e amplitude contínua (amostras);
- $m_q[k]$: amostras quantizadas.

pode-se estabelecer uma **Regra de quantização**: $m_q[k] = v_l$, se $m[k] \in [\lambda_l, \lambda_{l+1}]$.

O esquema para o processo que contemple amostragem, quantização e codificação, pode ser visualizado na Figura B.9:

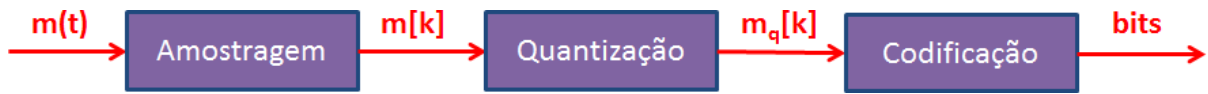


Figura B.9: Esquema do processo de quantização e codificação.

Relembramos que, tipicamente, escolhe-se $L = 2^b$, isto é, uma potência de 2. Assim, $b = \log_2 L$ é o número de *bits* necessários para representar cada amostra quantizada.

Para 16 pulsos distintos ($L = 16$), teremos 16 níveis de quantização. Assim, se $L = 2^b$, a cada um dos 16 níveis teremos associado uma palavra código, variando de 0000 a 1111.

Para a transmissão ou processamento digital, precisamos associar um pulso elétrico diferente a cada estado binário, que poderia ser um pulso positivo para o binário 1 e um pulso negativo para o binário 0. É um código de linha e outros existem, mas serão estudados em outro Módulo. Lathi (2007, p. 699), demonstra esta equivalência em uma tabela, muito similar a Tabela B.3, onde as setas para cima, representam o pulso positivo, e as setas para baixo, o pulso negativo:

Dígito	Equivalente binário	Representação da forma de onda			
0	0000	↓	↓	↓	↓
1	0001	↓	↓	↓	↑
2	0010	↓	↓	↑	↓
3	0011	↓	↓	↑	↑
4	0100	↓	↑	↓	↓
5	0101	↓	↑	↓	↑
6	0110	↓	↑	↑	↓
7	0111	↓	↑	↑	↑
8	1000	↑	↓	↓	↓
9	1001	↑	↓	↓	↑
10	1010	↑	↓	↑	↓
11	1011	↑	↓	↑	↑
12	1100	↑	↑	↓	↓
13	1101	↑	↑	↓	↑
14	1110	↑	↑	↑	↓
15	1111	↑	↑	↑	↑

Tabela B.3: Tabela de equivalências dos níveis de quantização

O sinal binário obtido é o sinal digital resultante da conversão do analógico-digital do sinal analógico original. Este sinal binário é chamado de Sinal Modulado por Código de Pulso (PCM). PCM será desenvolvido de forma mais ampla no Subseção B.7.3 (roteiro 6).

Tipos de quantizador: Os quantizadores podem ser uniformes ou não-uniformes. Essa uniformidade é dada pelo espaçamento entre os níveis: se é constante, tem-se um quantizador uniforme, se não, um quantizador não-uniforme ou genérico.

Erro (ou ruído) de quantização

O processo de codificação do Sinal Modulado em Amplitude de Pulso (PAM), num sinal PAM quantizado, envolve descartar algumas das informações analógicas originais. Esta distorção, introduzida pela necessidade de aproximar a forma de onda analógica com as amostras quantificados, é referenciado como **ruído de quantização**. A presença de tal ruído diminui a medida que aumenta o número de níveis empregados no processo de quantização.

Erro ou ruído de quantização é definido por:

$$e[k] = m_q[k] - m[k], \quad (\text{B.3})$$

de modo que $m_q[k] = m[k] + e[k]$.

Figuras de mérito:

- Erro máximo relativo: $\frac{\max e[k]}{\max m[k]}$, onde $\max m[k] = m_p$ é o valor de pico do sinal.
- Razão sinal-ruído de quantização: $\text{SQNR} = \frac{P_m}{P_e}$, onde P_m é a potência do sinal e P_e é a potência do ruído de quantização.

Assume-se que $-m_p \leq m[k] \leq m_p$.

B.4.4 Bibliografia do roteiro

1. HAYKIN, S. Sistemas de comunicação. Bookman, 2004. Cap. 03, p. 223-225.
2. LATHI, B. P. Sinais e Sistemas Lineares, [S.l.]: Bookman, 2007. Cap. 08, p. 697-700.

B.4.5 Exercícios do roteiro

1. Utilize o simulador, com os dados pré-existent, para *plotar* um sinal, suas amostras, a quantização genérica e o erro de quantização.
2. Verifique o que acontece ao acrescentar um nível fora do domínio da onda, como por exemplo 1,25.

3. Em que amostras quantizadas há um maior erro de quantização?

B.5 Roteiro 4: Quantização Uniforme

B.5.1 Objetivo geral do roteiro

Verificar o processo de quantização uniforme.

B.5.2 Objetivos específicos do roteiro

1. Definir o processo de quantização uniforme.
2. Exercitar o esboço de curva entrada \times saída do quantizador e a conversão para códigos binários.

B.5.3 Subsídios do Roteiro

A quantização uniforme tem passos uniformes de quantização por todas as magnitudes do sinal. Assim é indicada para sinais que apresentam uma frequência média de ocorrências em todas as amplitudes. A distância entre os níveis é constante e pode ser representado por Δ .

Os quantizadores podem ser de dois tipos: *mid-tread* ou *mid-rise*. *Mid-tread* terá um dos níveis em zero, enquanto que *mid-rise* terá o zero em um ponto mediano entre dois níveis, correspondendo a $\frac{\Delta}{2}$. Observe a Figura B.10 com uma verificação de 4 níveis ($L = 4$) para as duas formas de quantificação uniforme.

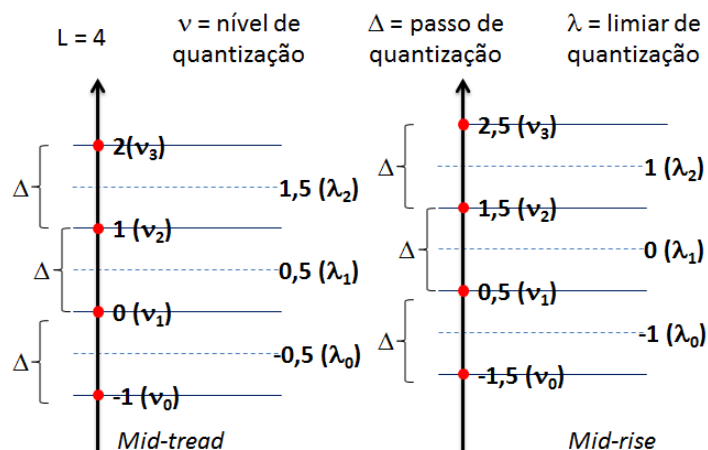
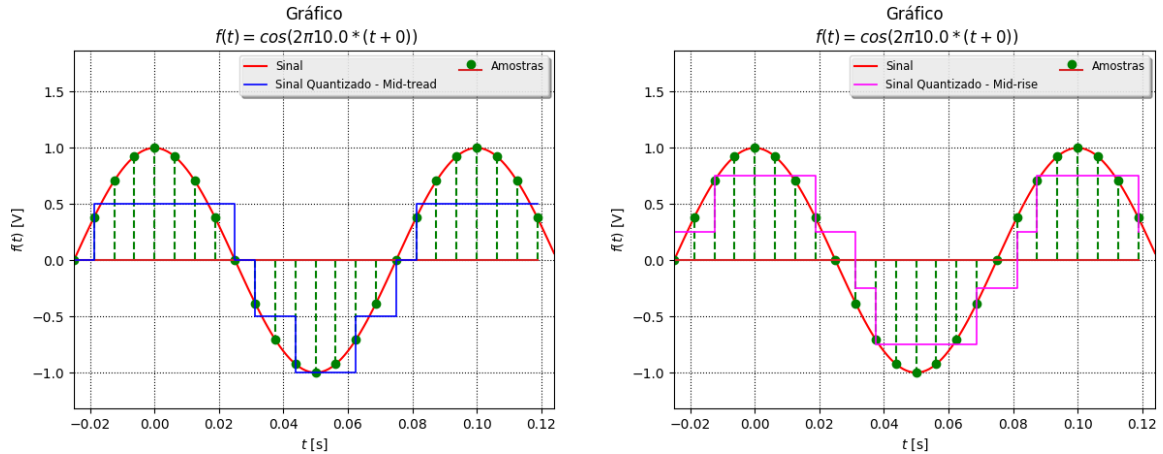


Figura B.10: Níveis, limiares e passo de quantização *mid-tread* e *mid-rise*.

Como o espaçamento é uniforme entre os níveis, necessariamente os limiares estarão na metade do nível. Assim, para os níveis sendo representados por v_l , se um valor amostrado

estiver entre v_l e $v_l + \frac{\Delta}{2}$, será arredondado para v_l ; já se for maior que $v_l + \frac{\Delta}{2}$, será arredondado para v_{l+1} .

Na Figura B.11, pode-se verificar os dois modelos de quantização uniforme.



(a) Gráfico de onda da função $f(t) = \cos(2\pi 10t)$ e quantização *mid-tread*.

(b) Gráfico de onda da função $f(t) = \cos(2\pi 10t)$ e quantização *mid-rise*.

Figura B.11: Gráfico de onda da função $f(t) = \cos(2\pi 10t)$ e as possibilidades de quantização uniforme.

Exercício resolvido:

A onda sinusoidal $m(t) = 6 \sin(2\pi t)$ [V] é amostrada a uma taxa de 8 Hz e quantizada com um quantizador *mid-rise* de 4 bits com passo de 1 V.

- Esboce a curva entrada-saída do quantizador.
- Construa (arbitrariamente) uma tabela de níveis de quantização e bits para o quantizador.
- Determine as saídas do quantizador para um ciclo completo da entrada, assumindo que a primeira amostra é tomada em $t = -500$ ms.

Solução:

(a) Conforme a Figura B.12: Número de *bits*: 4; $N_b = 4 \Rightarrow L = 16$ níveis.

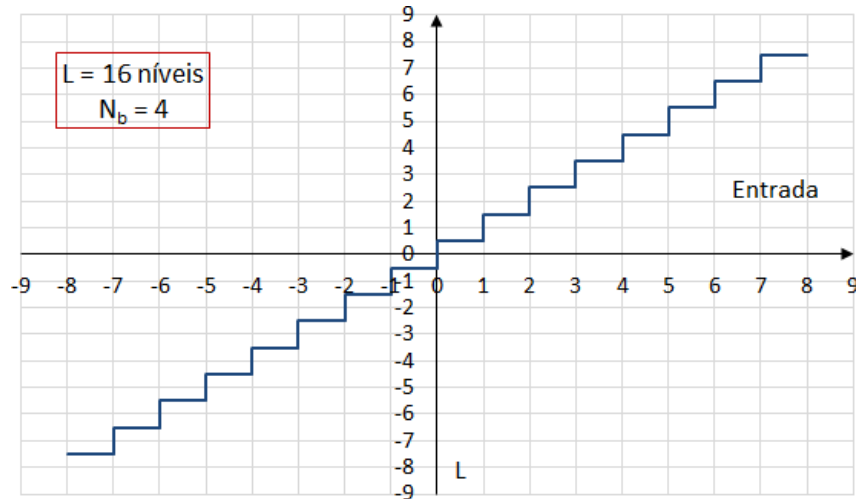


Figura B.12: Esboço da curva entrada-saída do quantizador.

(b) Na Tabela B.4 tem-se uma proposta de níveis de quantização e *bits* para o quantizador.

Nº	Nível de quantização	Representação binária
1	-7,5	0000
2	-6,5	0001
3	-5,5	0010
4	-4,5	0011
5	-3,5	0100
6	-2,5	0101
7	-1,5	0110
8	-0,5	0111
9	0,5	1000
10	1,5	1001
11	2,5	1010
12	3,5	1011
13	4,5	1100
14	5,5	1101
15	6,5	1110
16	7,5	1111

Tabela B.4: Tabela de níveis de quantização e *bits* para o quantizador.

(c) Na Figura B.13 pode se verificar as amostras que se estendem de $t = -500\text{ms}$ a $t = 500\text{ms}$.

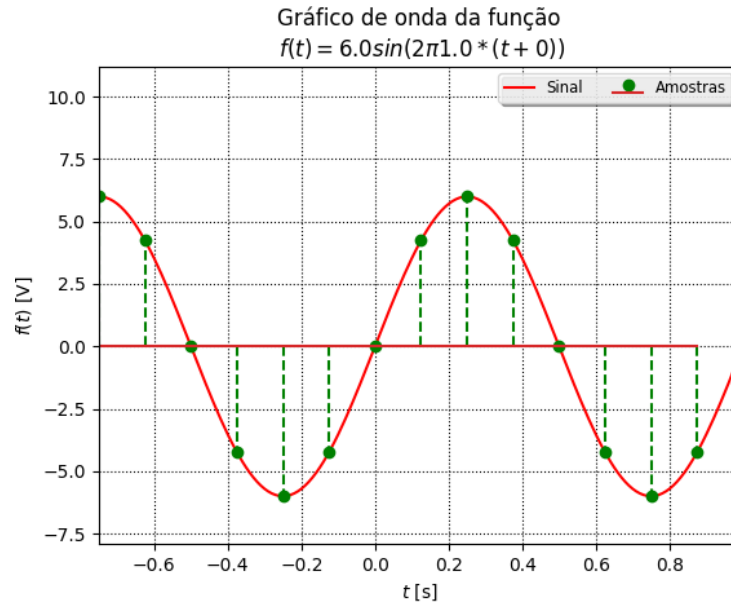


Figura B.13: Amostras de $t = -500\text{ ms}$ a $t = 500\text{ ms}$, da onda $m(t) = 6\sin(2\pi t)$, com $f_m = 1\text{ Hz} \Rightarrow T_m = 1\text{ s}$, $f_s = 8\text{ Hz} \Rightarrow T_s = 125\text{ ms}$.

Na Tabela B.5 verifica-se as saídas do quantizador para um ciclo completo de entrada.

t [ms]	$m(t)$ [V]	$m_q(t)$ [V]	Representação binária
-500	0	-0,5	0111
-375	-4,24	-4,5	0011
-250	-6	-6,5	0001
-125	-4,24	-4,5	0011
0	0	-0,5	0111
125	4,24	4,5	1100
250	6	5,5	1101
375	4,24	4,5	1100
500	0	-0,5	0111

Tabela B.5: Tabela de saídas do quantizador para um ciclo completo de entrada.

Saída do quantizador: 011100110001001101111100110111000111

Análise do quantizador uniforme:

Deve-se enfatizar que a quantização do sinal é a quantização das amostras do sinal, portanto tem-se Δ como o passo de quantização, m_p como o valor de pico do sinal, que corresponde a amostra de maior amplitude ($\max m[k] = m_p$) e L como o número de níveis, então pode-se concluir que:

$$\Delta = \frac{2m_p}{L} \quad (\text{B.4})$$

então, conforme evidenciado na Figura B.13 a possível faixa de valores para erro é $[-\frac{\Delta}{2}, \frac{\Delta}{2}]$.

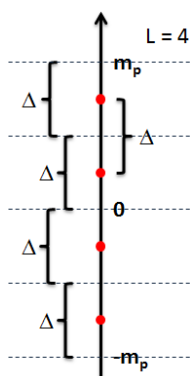


Figura B.14: Modelo com quatro níveis de quantização e seus respectivos passos e valores de pico do sinal.

Portanto:

- Erro máximo absoluto: $\frac{\Delta}{2}$
- Erro máximo relativo ao valor de pico: $\frac{\frac{\Delta}{2}}{m_p} = \frac{1}{L}$
- Razão sinal-ruído de quantização (SNQR): $3 \frac{P_m}{m_p^2} L^2$, onde P_m é a potência média do sinal.

Hipótese: o ruído de quantização ($e[k]$) assume valores no intervalo $[-\frac{\Delta}{2}, \frac{\Delta}{2}]$ com igual frequência. Ou seja: $e[k] \sim \text{unif}(-\frac{\Delta}{2}, \frac{\Delta}{2})$.

Considerando que se $e[k]$ é um processo aleatório de média zero, então a potência de $e[k]$ é dada por:

$$P_e = E[(e[k])^2] = \text{var}[e[k]] \quad (\text{B.5})$$

E a variância de $e[k]$ é dada por:

$$\text{var}[e[k]] = \frac{(\frac{\Delta}{2} - (-\frac{\Delta}{2}))^2}{12} = \frac{\Delta^2}{12} \quad (\text{B.6})$$

Observação: sugere-se revisão de sinais de energia e sinais de potência, além de variáveis aleatórias com os temas função de densidade de probabilidade e variância.

Segue que:

$$\text{SQNR} = \frac{P_m}{P_e} = \frac{P_m}{\frac{\Delta^2}{12}} = \frac{P_m}{\left(\frac{4m_p^2}{L^2}\right) \cdot \frac{1}{12}} \quad (\text{B.7})$$

Convertendo para dB:

$$\begin{aligned} [\text{SQNR}]_{dB} &= 10 \log_{10} 3 \frac{P_m}{m_p^2} L^2 \\ &= 10 \log_{10} 3 \frac{P_m}{m_p^2} + 10 \log_{10} L^2 \\ &= 10 \log_{10} 3 \frac{P_m}{m_p^2} + 20 \log_{10} L \\ &= 10 \log_{10} 3 \frac{P_m}{m_p^2} + 20 \log_{10} 2^{N_b}, \text{ sendo } L = 2^{N_b} \\ &= 10 \log_{10} 3 \frac{P_m}{m_p^2} + (20 \log_{10} 2) N_b \end{aligned} \quad (\text{B.8})$$

Conclusão: $20 \log_{10} 2 \approx 6$ dB, então há uma melhora na *SNQR* de 6 dB por *bit*.

B.5.4 Bibliografia do roteiro

1. HAYKIN, S. Sistemas de comunicação. Bookman, 2004. Cap. 03, p. 223-225.
2. LATHI, B. P. Sinais e Sistemas Lineares, [S.l.]: Bookman, 2007. Cap. 08, p. 697-700.

B.5.5 Exercícios do roteiro

1. Estabeleça nos dois sinais assinalados na Figura B.11, os níveis de quantização *mid-tread* e *mid-rise*.
2. Parametrize o número de níveis de quantização para 8, 16, 32 e 64, para uma mesma onda e verifique as alterações no gráfico.
3. Verifique em cada uma das parametrizações anteriores, os erros de quantização.
4. Refaça o exercício resolvido deste roteiro, mas para um quantizador *mid-tread*.

B.6 Roteiro 5: Quantização Não-Uniforme

B.6.1 Objetivo geral do roteiro

Verificar o processo de quantização não-uniforme.

B.6.2 Objetivos específicos do roteiro

1. Definir o processo de quantização não-uniforme.
2. Verificar a aplicação da quantização não-uniforme em sinais de voz.

B.6.3 Subsídios do Roteiro

O objetivo da quantização não-uniforme é melhorar a relação sinal-ruído (SQNR) e também diminuir o número de *bits* por amostra. A ideia é a de favorecer as amplitudes menores em detrimento das maiores, em sinais de grande faixa dinâmica.

Quantizadores não-uniformes e sinais de voz

Segundo Haykin (2004, p. 223), um sinal de voz tem um intervalo de tensão na ordem de 1 para 1.000, candidatando-o para a quantização não-uniforme, sendo que as ocorrências em magnitudes mais altas são raras. O quantizador a ser utilizado possui a seguinte arquitetura: o sinal sofre uma compressão inicial para então ser quantizado uniformemente. No receptor ele é novamente expandido. O modelo deste quantizador pode ser definido conforme a Figura B.15:

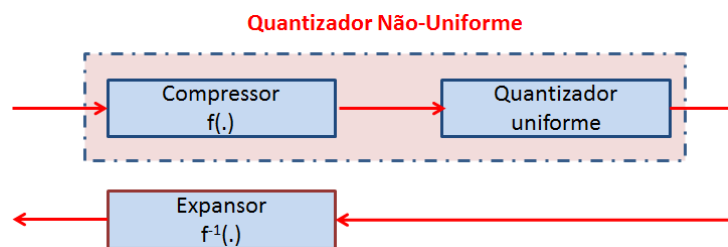


Figura B.15: Modelo de quantizador não-uniforme.

(SKLAR, 2002, p. 83) explica o modelo, afirmando que a compressão altera a distribuição de magnitudes do sinal de entrada de modo que não haverá uma preponderância de sinais de baixa amplitude na saída do compressor. Após a compressão, o sinal distorcido passará por um quantizador uniforme. No receptor será feita a expansão ou compressão inversa do sinal recebido. O par de processamento (compressão e expansão) é normalmente referido como *companding*. Um sistema com um quantizador uniforme seria um desperdício para sinais de

voz, já que muitos dos passos de quantização seriam raramente utilizados. Num sistema que utiliza os níveis de quantização espaçados igualmente, o ruído de quantização é o mesmo para todas as magnitudes do sinal. Portanto, com uma quantização uniforme, a relação sinal-ruído (SNR) se torna pior para níveis de entrada mais baixos do que para os níveis de entrada mais altos. Assim, com o quantizador não-uniforme, a ideia é favorecer a amplitudes menores em detrimento das maiores.

Reproduz-se o gráfico presente em Sklar (2002, p. 81:figura 2.17), que evidencia a baixa ocorrência de amplitudes maiores num sinal de voz, na Figura B.16:

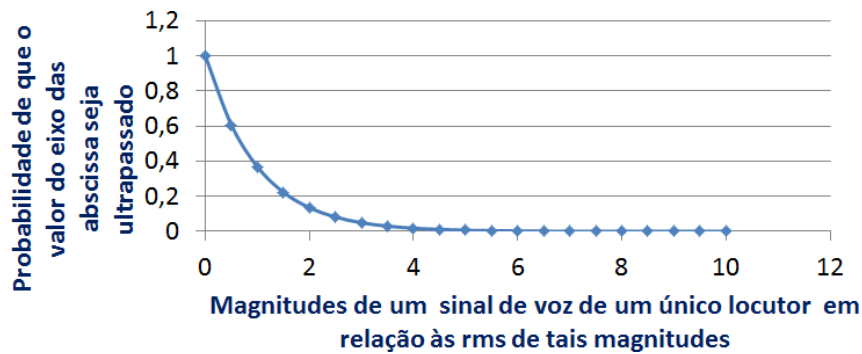


Figura B.16: Distribuição estatística da fala de um único locutor nas magnitudes de sinais de voz.

Em telefonia (sinais de voz) dois padrões são utilizados:

Lei μ (nos Estados Unidos da América e no Japão) definida por:

$$f(x) = \frac{\ln(1 + \mu x)}{\ln(1 + \mu)}, \quad 0 \leq x \leq 1, \quad x = \frac{m(t)}{m_p}$$

$$f^{-1}(y) = \frac{(1 + \mu)^y - 1}{\mu}, \quad 0 \leq y \leq 1 \quad (\text{B.9})$$

$$\mu = 100 \text{ (7 bits)} \quad \text{ou}$$

$$\mu = 255 \text{ (8 bits)}$$

Lei A (no Brasil, Europa, etc.) definida por:

$$f(x) = \frac{1}{1 + \ln A} \cdot \begin{cases} Ax & , 0 \leq x \leq A^{-1} \\ 1 + \ln(Ax) & , A^{-1} \leq x \leq 1 \end{cases} \quad x = \frac{m(t)}{m_p}$$

$$f^{-1}(y) = \frac{1}{A} \cdot \begin{cases} (1 + \ln A)y & , 0 \leq y \leq (1 + \ln A)^{-1} \\ e^{(1 + \ln A)y - 1} & , (1 + \ln A)^{-1} \leq y \leq 1 \end{cases} \quad (\text{B.10})$$

$$A = 87,6 \text{ ou } 87,7 \text{ (8 bits)}$$

Pode-se mostrar que a SQNR pode ser aproximada por:

$$\begin{aligned} \text{SQNR} &\approx \frac{3L^2}{[1 + \ln A]^2} \quad , \text{ se } A \gg \frac{m_p}{\sqrt{P_m}} \quad (\text{Lei } A) \\ \text{SQNR} &\approx \frac{3L^2}{[\ln(1 + \mu)]^2} \quad , \text{ se } \mu \gg \frac{m_p}{\sqrt{P_m}} \quad (\text{Lei } \mu) \end{aligned} \quad (\text{B.11})$$

Então, conforme visto no desenvolvimento das equações do roteiro anterior, temos novamente uma melhora de 6dB por *bit*.

B.6.4 Bibliografia do roteiro

1. HAYKIN, S. Sistemas de comunicação. Bookman, 2004. Cap. 03, p. 223-225.
2. LATHI, B. P. Sinais e Sistemas Lineares, [S.l.]: Bookman, 2007. Cap. 08, p. 697-700.
3. SKLAR, B. Digital Communications: Fundamentals and Applications. [S.l.]: BEIJING: Publishing House of Electronics Industry, 2002. Cap. 02, p. 81-83.

B.6.5 Exercícios do roteiro

1. Converta para dB a aproximação da SQNR, para a lei μ , com os valores de $\mu = 100$ e $\mu = 255$.
2. Converta para dB a aproximação da SQNR, para a lei A, com os valores de $A = 87,6$ e $A = 87,7$.
3. Estabeleça um quadro comparativo entre os dois padrões e seus valores.

B.7 Roteiro 6: Modulação por codificação de pulso – PCM

B.7.1 Objetivo geral do roteiro

Verificar o processo de modulação por codificação de pulso (PCM).

B.7.2 Objetivos específicos do roteiro

1. Integrar as etapas descritas nos roteiros anteriores.
2. Apresentar o modelo PCM.
3. Exercitar a verificação das etapas PCM.

B.7.3 Subsídios do Roteiro

O processo que compreende amostragem, quantização (uniforme ou não) e codificação é chamado de Modulação por Codificação de Pulso ou PCM. É um método para representar digitalmente amostras de sinais analógicos. Segundo Sklar (2002, p. 79), PCM é o nome dado à classe de sinais em banda base obtidos a partir da codificação de cada amostra quantizada em uma palavra digital.

Banda base é o termo usado para transmissão de sinais sem deslocamento no espectro das frequências, ou seja, sem portadora. Para transmissão em banda base, os *bits* são enviados de acordo com um padrão digital ou codificação. Importa ressaltar que a transmissão ocupará uma alta largura de banda, que deve estar disponível.

Apresenta-se o seguinte modelo PCM, conforme a Figura B.17:

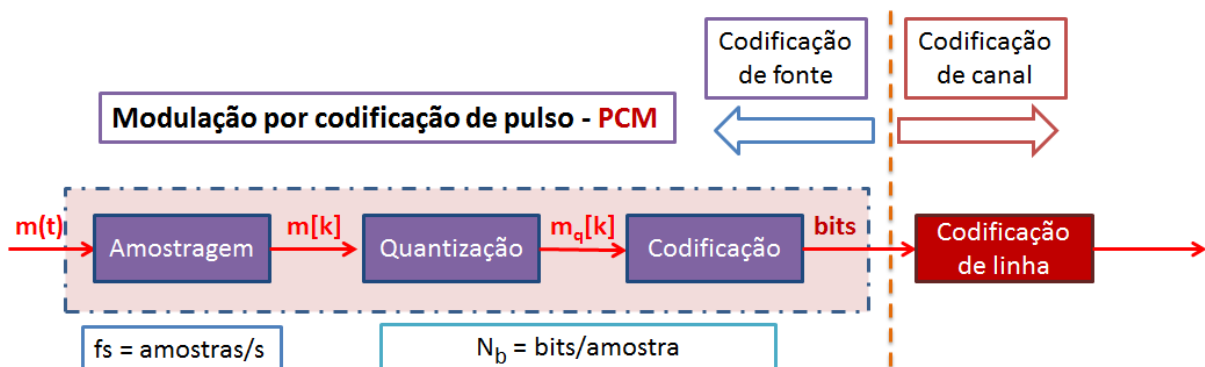


Figura B.17: Modelo PCM.

Como se evidencia na Figura B.17, para que o processo seja eficiente, ou seja, que haja a reprodução adequada da mensagem enviada de um ponto a outro ponto, duas variáveis são fundamentais: a taxa de amostragem (f_s) e o número de *bits* (N_b), que determinaram a taxa de *bits* ou *bit rate* (R_b), que é dada por:

$$R_b = f_s N_b \text{ (bits/s)} \tag{B.12}$$

Tanto Sklar (2002, p. 79–80), quanto Haykin (2004, p. 225–226) afirmam que o processo envolverá a transmissão de formas de onda de pulsos equivalentes a uma palavra digital. Exemplifica-se com um sinal analógico $x(t)$ limitado entre -4 a $+4$ V. Define-se o passo entre os níveis de quantização (L) em 1 V, obtendo-se oito níveis ($L = 8$), sendo eles $[-3,5; -2,5; -1,5; -0,5; 0,5; 1,5; 2,5; 3,5]$. Cada nível será representado por um número de código, iniciado com 0 para $-3,5$ V, 1 para $-2,5$ V e assim por diante. Cada número de código terá sua representação em aritmética binária, variando de 000 para o código 0, a 111 para o código 7. Na Figura B.18 verifica-se as etapas de PCM de um sinal $x(t) = 4 \cos(2\pi ft)$, com as especificações de quantização e codificação já descritas.

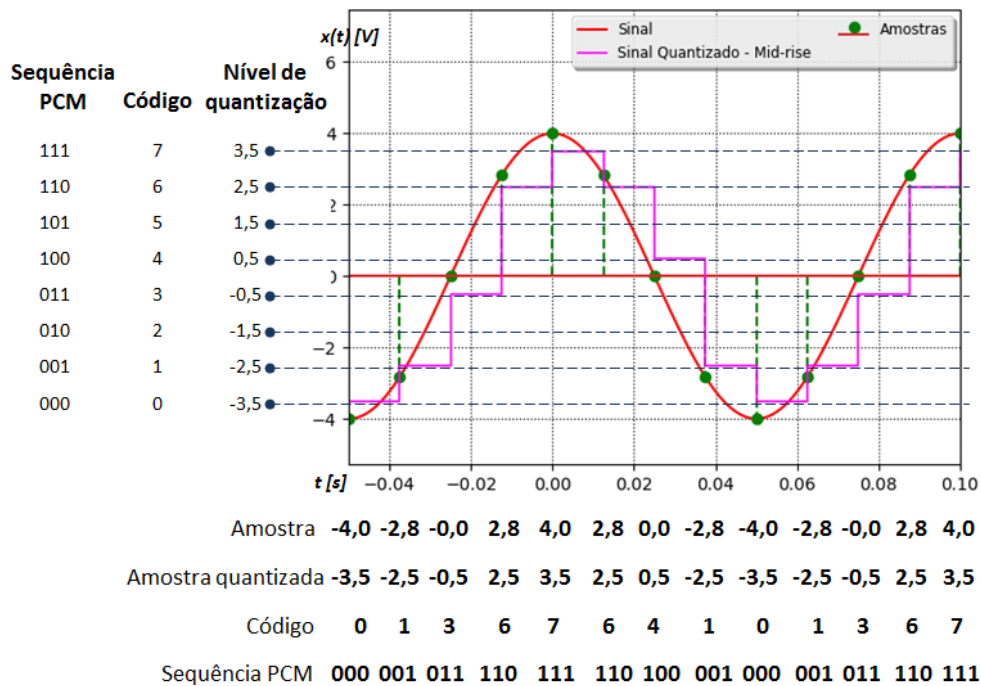


Figura B.18: Sinal $x(t) = 4 \cos(2\pi ft)$ e respectivos valores de amostra, amostras quantizadas, código e sequência digital PCM.

B.7.4 Bibliografia do roteiro

1. HAYKIN, S. Sistemas de comunicação. Bookman, 2004. Cap. 03, p. 223-225.

2. LATHI, B. P. Sinais e Sistemas Lineares, [S.l.]: Bookman, 2007. Cap. 08, p. 695, 697-698.
3. SKLAR, B. Digital Communications: Fundamentals and Applications. [S.l.]: BEIJING: Publishing House of Electronics Industry, 2002. Cap. 02, p. 79-81.

B.7.5 Exercícios do roteiro

1. Siga até a página do simulador, parametrize o sinal mostrado na Figura B.18, só que com 16 níveis de quantização. Construa um mapa, conforme a exemplo dado e identifique quantos bits serão necessários para esta modulação.

Referências Bibliográficas

- ALVES, M. F. S.; MANTOVANI, K. L. Identificação do perfil dos acadêmicos de engenharia como uma medida de combate à evasão. *Revista de Ensino de Engenharia*, Belo Horizonte - MG, v. 35, n. 2, p. 26–36, 2016. Disponível em: <<http://198.136.59-.239/abengeorg/revista/index.php/abenge/article/download/464/751>>. Acesso em: 05 jul. 2017.
- ANDRADE, L. A. d. R.; PEREIRA, E. M. d. A. Educação a distância e ensino presencial: convergência de tecnologias e práticas educacionais. In: _____. *Trajetórias de Educação Profissional e Tecnológica*. 1. ed. [S.l.]: Florianópolis: IFSC, 2012. p. 143–163. ISBN 978-85-64426-40-5.
- BOTTI, N. C. L. e. a. Tecnologia educacional: Uma estratégia para o ensino de saúde mental. *Cadernos Brasileiros de Saúde Mental*, Florianópolis - SC, v. 7, n. 15, p. 24–31, 2015. ISSN 1984-2147. Disponível em: <<http://incubadora.periodicos.ufsc.br/index.php/cbsm/article/view/2329/4434>>. Acesso em: 18 abr. 2017.
- CUNHA, F. M. Ensino de engenharia: Abordagem pela complexidade. *Revista de Ensino de Engenharia*, Belo Horizonte - MG, v. 34, n. 1, p. 3–16, 2015. ISBN 0101-5001. Disponível em: <<http://dx.doi.org/10.15552/2236-0158/abenge.v34n1p3-16>>. Acesso em: 18 abr. 2017.
- CYBIS, W. de A.; BETIOL, A. H.; FAUST, R. *Ergonomia e usabilidade: conhecimentos, métodos e aplicações*. [S.l.]: Novatec, 2010. ISBN 9788575222324.
- GALLAGER, R. *Principles of Digital Communication*. [S.l.]: Cambridge University Press, 2008. ISBN 9781139468602.
- HAYKIN, S. *Sistemas de comunicação*. [S.l.]: Bookman, 2004. ISBN 9788573079364.
- LATHI, B. P. *Sinais e Sistemas Lineares*. [S.l.]: Bookman, 2007.
- MORIN, E. *Os sete saberes necessários à educação do futuro*. [S.l.]: São Paulo: Cortez; Brasília: UNESCO, 2000, 2000. ISBN 85-249-0741-X.
- MORIN, E. *Ciência com consciência*. [S.l.]: Rio de Janeiro: Bertrand Brasil, 2005. ISBN 85-286-0579-5.
- SKLAR, B. *Digital Communications: Fundamentals and Applications*. [S.l.]: BEIJING: Publishing House of Electronics Industry, 2002.
- WIKIPÉDIA, a. e. l. *Nginx*. 2017. Disponível em: <<https://pt.wikipedia.org/wiki/Nginx>>.