

# INSTRUÇÕES PARA O ARTIGO DO TRABALHO DE CONCLUSÃO DE CURSO – BCC – SIS

João Pedro da Silva, José Carlos Pereira – Orientador

**Resumo:** O resumo é uma apresentação concisa dos pontos relevantes de um texto. Informa suficientemente ao leitor, para que este possa decidir sobre a conveniência da leitura do texto inteiro. Deve conter OBRIGATORIAMENTE o OBJETIVO, METODOLOGIA, RESULTADOS e CONCLUSÕES. O resumo não deve ultrapassar 10 linhas e deve ser composto de uma sequência corrente de frases concisas e não de uma enumeração de tópicos. O resumo deve ser escrito em um único texto corrido (sem parágrafos). Deve-se usar a terceira pessoa do singular. As palavras-chave, a seguir, são separadas por ponto, com a primeira letra maiúscula. Caso uma palavra-chave seja composta por mais de uma palavra, somente a primeira deve ser escrita com letra maiúscula, sendo que as demais iniciam com letra minúscula, desde que não sejam nomes próprios.]

**Palavras-chave:** Ciência da computação. Sistemas de informação. Monografia. Resumo. Formato.

## 1 INTRODUÇÃO

O presente *template* deve ser seguido para a confecção dos trabalhos finais de conclusão dos cursos de Ciência da Computação e Sistemas de Informação da FURB. Ele é fortemente (mas não totalmente) baseado no modelo da Sociedade Brasileira de Computação de modo a facilitar posteriores adaptações para publicações das produções dos TCCs em eventos científicos. O *template* proposto segue as normas da ABNT para citações e referências bibliográficas, bem como para referências à figuras, quadros e tabelas.

Para o TCC dos cursos, **o presente artigo está limitado a 20 páginas, incluindo as referências bibliográficas e excluindo os anexos**, alguns dos quais são obrigatórios. Os principais elementos de formatação estão explicados no anexo no fim deste *template*. O formato pode ser diferente caso o artigo seja submetido a um evento científico da área. Neste caso o estudante pode manter o artigo no formato do evento, mas deve anexar o comprovante de submissão. Dúvidas ou problemas devem ser sanados com a coordenação do TCC do curso.

A introdução deve despertar no leitor o interesse pelo texto, apresentando os assuntos que serão tratados e o enfoque que será dado ao tema central. Deve iniciar com uma **contextualização** do estudo a ser realizado, explicando claramente sua origem/motivação. A visão geral do tema deve então ser afunilada até se chegar ao problema a ser pesquisado. Após o problema ter sido identificado, deve-se delimitar que aspectos ou elementos serão tratados. Deve-se deixar bem claro o problema que se quer resolver, com o desenvolvimento do trabalho e **justificar** porque o assunto merece ser estudado.

Deve esclarecer a **formulação do problema** a ser investigado e deve ser finalizado com os **objetivos do trabalho** que podem ser subdivididos **em geral (obrigatório)** e **específicos (opcional)**. O objetivo principal deve ser descrito em uma frase única, usando o verbo no infinitivo. Os objetivos específicos detalham o objetivo principal ou definem subprodutos do trabalho. Também se relacionam a formas de validação ou avaliação do trabalho realizado. Os objetivos devem ser mensuráveis quanto a se e como foram ou não atingidos. Os objetivos específicos devem ser enumerados, usando verbos no infinitivo.

## 2 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica pode ser subdividida em subseções de acordo com o que for mais conveniente. A seguir são propostas formas de organização sendo que a de correlatos é obrigatória. **Recomenda-se fortemente que esta seção não ultrapasse quatro páginas.**

### 2.1 CONCEITOS, TÉCNICAS E/OU FERRAMENTAS

Na primeira parte deve-se abordar os conceitos, técnicas e/ou ferramentas mais relevantes envolvidos com o tema, devendo ser omitidas metodologias de especificação e ferramentas de implementação que já são conhecidas.

### 2.2 VERSÃO ANTERIOR DO SISTEMA/FERRAMENTA/BIBLIOTECA/Framework (OPCIONAL)

Quando o projeto propõe uma continuação ou extensão de um Trabalho de Conclusão de Curso (TCC) anterior, deve-se descrevê-lo em uma seção específica.

### 2.3 TRABALHOS CORRELATOS

Na segunda parte da fundamentação devem ser descritos os trabalhos correlatos. Devem ser incluídos preferencialmente trabalhos acadêmicos com características e funcionalidades semelhantes ao que está sendo produzido.

A descrição deve ser feita em quadros conforme apresentado no Quadro 1. Os itens apresentados no quadro são obrigatórios.

Quadro 1 – Trabalho Correlato 1

Referência	
Objetivos	
Principais funcionalidades	
Ferramentas de desenvolvimento	
Resultados e conclusões	

Fonte: elaborado pelo autor.

Os quadros devem ser finalizados com uma breve análise da relação com o trabalho proposto.

### 3 DESCRIÇÃO DO APLICATIVO

Este capítulo pretende apresentar os detalhes de especificação e implementação do aplicativo. São apresentados quatro subcapítulos. O primeiro subcapítulo apresenta a visão geral do aplicativo, de forma a ambientar o leitor quanto ao funcionamento do processo de composição e as ações de alterações de propriedades musicais. O segundo subcapítulo apresenta os detalhes da carga de dados do aplicativo, detalhando a estrutura de composição e o processo de leitura de arquivos MIDI. O terceiro...

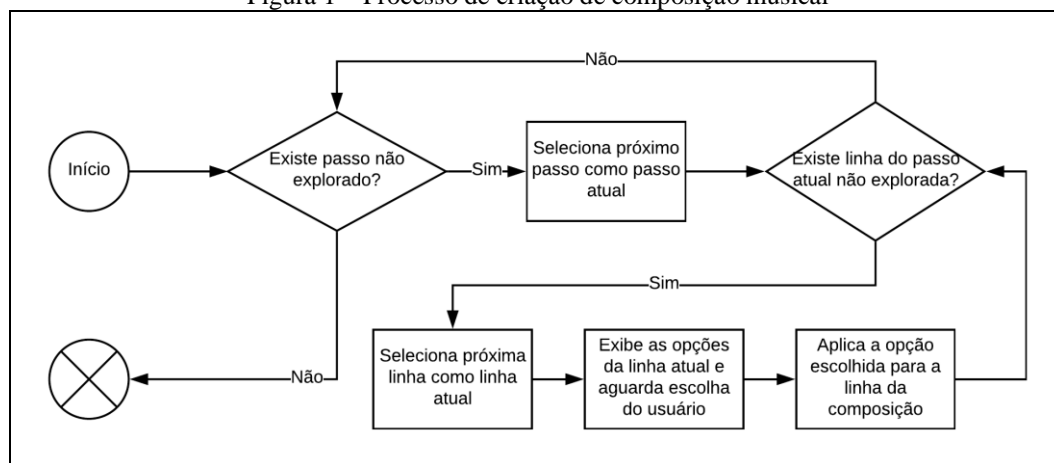
#### 3.1 VISÃO GERAL DO APLICATIVO

O aplicativo disponibiliza para o usuário uma forma de combinar fragmentos musicais para gerar uma composição musical personalizada. O processo de criação da composição musical personalizada é organizado em etapas. Em cada etapa o usuário deve escolher um fragmento musical dentre os possíveis fragmentos exibidos pelo aplicativo. Quando um fragmento é escolhido ele é adicionado à composição, os fragmentos não escolhidos são descartados e uma nova etapa de escolha se inicia. Para possibilitar esta dinâmica, os fragmentos musicais são agrupados em dois níveis. O primeiro nível existe para que seja possível segregar os fragmentos de acordo com o posicionamento que eles podem assumir dentro da composição. Já o segundo nível pretende segregar os fragmentos em razão da sua função musical. Para estruturar os dados dos fragmentos musicais e organizar a dinâmica de composição musical o aplicativo utiliza quatro elementos, são eles:

- a) opção: é o elemento que representa um fragmento musical;
- b) linha: é o elemento que agrupa as opções;
- c) passo: é o elemento que agrupa linhas;
- d) fonte de composição: é o elemento que agrupa passos.

O aplicativo utiliza uma fonte de composição para configurar o processo de criação de uma nova composição musical personalizada. Quando o processo de criação da composição é iniciado, com base nos passos existentes na fonte de composição, o aplicativo divide a composição em macro etapas, gerando uma macro etapa de composição por passo. Dentro de cada passo, para cada linha existente no passo, o aplicativo gera uma etapa de composição, que contém as opções disponíveis ao usuário para aquela combinação de agrupamento. Com base nesses dados, o aplicativo exibe as opções, de etapa em etapa, para que o usuário realize a escolha de um fragmento musical por etapa. O processo de composição se inicia com a exibição das opções da primeira linha do primeiro passo. Assim que o usuário realiza a escolha de uma opção o aplicativo adiciona sua escolha à composição e passa a exibir as opções da segunda linha do primeiro passo, se existir mais de uma linha. Quando todas as linhas do primeiro passo forem exploradas, o aplicativo passa a exibir os dados do segundo passo, se existir mais de um passo. Esse processo se repete até que todos os passos tiverem sido explorados. Quando todos os passos tiverem sido explorados a composição musical está completa. Todos os passos de uma composição musical devem possuir as mesmas linhas em seu agrupamento. A Figura 1 exibe o fluxograma que representa o processo de criação de composição musical.

Figura 1 – Processo de criação de composição musical



Fonte: elaborado pelo autor.

O agrupamento dos fragmentos musicais em grupos de passos e linhas é realizado da forma como foi apresentada para possibilitar restrições em relação à posicionamento e função musical de cada fragmento musical. O agrupamento de linha existe para que, a cada passo de composição, o usuário tenha que escolher um elemento que tem uma função musical para a composição, como, por exemplo, melodia ou ritmo. Já o agrupamento de passo tem relação com a segregação de posicionamento. Para que a composição criada com o aplicativo funcione de acordo com as regras

de teoria musical e harmonia, todos os elementos dentro de um passo devem ser compatíveis uns com os outros, independentemente da linha em que os fragmentos musicais se encontram. Devido a estes dois agrupamentos, caso os dados da fonte de composição estejam consistentes entre si, a composição resultado será consistente.

Além de possibilitar a escolha de fragmentos musicais para a geração de uma composição musical personalizada, o aplicativo também permite a edição de propriedades sonoras dos fragmentos musicais. Quatro são as propriedades que podem ser alteradas e essas alterações podem surtir efeito a nível de composição, de linha ou de opção. Quando uma propriedade sonora de opção é alterada, o novo valor escolhido somente altera a opção editada. Quando uma propriedade sonora de linha é alterada, todas as opções que já foram escolhidas para a linha terão a propriedade alterada editada. Quando uma propriedade sonora da composição for alterada a nova propriedade é aplicada em todos os fragmentos da fonte de composição. O Quadro 2 demonstra a relação das alterações de propriedades sonoras com os elementos de composição e relaciona os elementos que o aplicativo permite alterar com as propriedades sonoras.

Quadro 2 – Relação entre propriedades sonoras editáveis e propriedades da composição

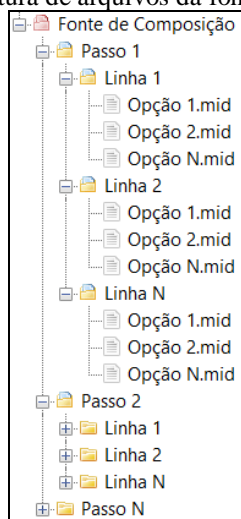
Propriedade sonora	Elemento de composição alterado	Funcionamento
Timbre	opção	O timbre é alterado quando a propriedade de instrumento musical da opção é alterada.
Volume	linha	O volume é alterado quando a propriedade de volume da linha é alterada.
Duração	fonte de composição	A duração é alterada quando a propriedade de velocidade da composição é alterada.
Frequência	fonte de composição	A frequência é alterada quando a propriedade de tonalidade da composição é alterada.

Fonte: elaborado pelo autor.

### 3.2 CARGA DE DADOS DE COMPOSIÇÃO

Para que os fragmentos musicais sejam disponibilizados para o aplicativo é utilizado o sistema de arquivos do dispositivo móvel. Com base em pastas e arquivos, a estrutura dos elementos de composição é representada. Uma pasta representa a fonte de composição. Dentro desta pasta, outras pastas representam os passos. Cada pasta dentro da pasta de fonte de composição representa um passo. Dentro de cada pasta de passo, outras pastas representam as linhas. Cada pasta dentro de uma pasta de passo representa uma linha. Dentro de cada pasta de linha, arquivos MIDI representam as opções. Cada arquivo pode conter dados de uma ou várias opções. A Figura 2 demonstra a organização da estrutura em questão.

Figura 2 – Estrutura de arquivos da fonte de composição



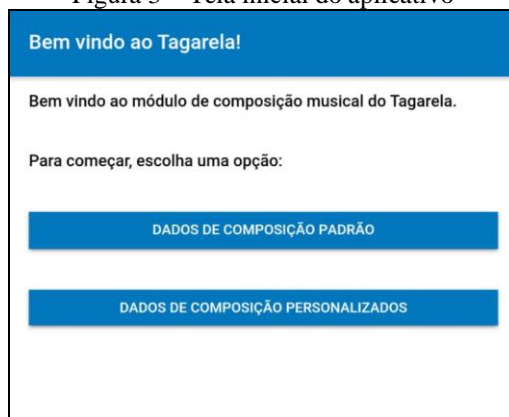
Fonte: elaborado pelo autor.

Por padrão, o aplicativo contém quatro fontes de composição. Estas fontes de composição são disponibilizadas com a instalação do aplicativo. Porém, para possibilitar que o usuário crie fontes de composição personalizadas, é possível criar esta estrutura em um diretório externo à aplicação. O caminho relativo do diretório monitorado pelo aplicativo, Tagarela/Musicoterapia/Composicoes, é o mesmo para dispositivos Android ou IOS. Porém, para dispositivos Android o diretório deve ser criado na raiz do armazenamento interno do dispositivo e para IOS no diretório de documentos. Estas pastas devem ser criadas manualmente pelo usuário se for necessário utilizar uma fonte de composição personalizada. Isto pois, em alguns dispositivos, as pastas criadas pelo aplicativo

não ficam visíveis ao usuário. Todas as pastas existentes na pasta `Composicoes` são consideradas fontes de composição pelo aplicativo.

Para que o aplicativo consiga fazer distinção entre os dados padrão e os dados personalizados, a tela inicial da aplicação dispõe de dois botões. O primeiro botão realiza a leitura de dados de fontes de composição padrão. Já o segundo realiza a leitura de dados do diretório de fontes de composição personalizadas. A Figura 3 demonstra a tela inicial do aplicativo. Quando qualquer um dos botões for pressionado, uma lista contendo o nome dos diretórios de fonte de composição é exibida. Se nenhuma pasta de fonte de composição for encontrada, uma mensagem é exibida ao usuário informando sobre a ausência de fontes de composição. Quando uma das opções listadas é selecionada, o processo de carga de dados de composição é iniciado.

Figura 3 – Tela inicial do aplicativo



Fonte: elaborado pelo autor.

O processo de carga de dados consiste em carregar os dados da fonte de composição para a memória do aplicativo, bem como definir os valores de configuração padrão para a composição. Também é nesta etapa que os dados de configuração anteriormente salvos são recarregados, caso existam. No processo de carga de dados as classes `MusicalCompositionConfigControl` e `MusicalCompositionSourceControl` são instanciadas. Estas classes são responsáveis por instanciar e carregar os dados para as classes `MusicalCompositionConfig` e `MusicalCompositionSource` respectivamente.

O primeiro método executado pelo processo de carga de dados é o `loadConfigs` da classe `MusicalCompositionConfigControl`. Este método é responsável por instanciar a classe de configuração `MusicalCompositionConfig` e definir os valores dos seus atributos. Esta classe de configuração contém dados sobre a localização dos elementos de composição no sistema de arquivos do dispositivo. Ou seja, os caminhos relativos de cada passo, linha e opção existentes na fonte de composição escolhida são atributos desta classe. Além disto, esta classe também armazena dados de parâmetros da composição. O próximo subcapítulo abordará as questões que dizem respeito à parametrização do processo de composição.

Após a criação da classe modelo de configuração, a classe modelo das fontes de composição é instanciada. Este processo ocorre com a execução do método `loadSources` da classe `MusicalCompositionSourceControl` que recebe como parâmetro a instância da classe `MusicalCompositionConfig`, criada na execução da função anterior. Este método lê os arquivos MIDI de acordo com os caminhos determinados na classe de configuração e carrega para a memória do dispositivo os dados MIDI que são pertinentes à composição. Para isto, para cada opção mapeada no objeto de configuração, a função `setupMidiFromFile` da classe `MidiFileControl` é executada. A classe `MidiFileControl` é atributo da classe `MusicalCompositionSourceControl`.

A função `setupMidiFromFile` recebe uma string binária como parâmetro e retorna um array de objetos `Midi`. Cada objeto do array retornado representa uma opção para o grupo linha/ passo que está sendo processado. A string binária é obtida através da leitura do arquivo MIDI pelo plugin `File` do Ionic. A função `setupMidiFromFile` realiza a leitura, caractere a caractere, da string binária determinando quais dados são importados para um objeto `Midi` e se a configuração do arquivo está de acordo com o esperado pelo aplicativo. Primeiramente a função em questão realiza as verificações dos dados existentes no cabeçalho do arquivo MIDI. Somente são carregados os arquivos MIDI tipo 0 e 1. Os arquivos MIDI tipo 2 geram uma exceção na função em questão, gerando uma mensagem específica para o usuário e interrompendo o processo de carga de dados. Outra validação é o tipo de `division` do arquivo MIDI. Somente é suportado pelo aplicativo o tipo 0. Arquivos MIDI com tipos diferentes deste também geram exceção, mensagem ao usuário e interrompem o processo de carga de dados.

Após realizar a validação do cabeçalho MIDI, para cada `track` do arquivo MIDI, é instanciada uma classe `Midi` de tipo 0. O retorno da função `setupMidiFromFile` é um array de instâncias `Midi` contendo os objetos

criados neste processamento. Para cada evento MIDI de cada uma das *tracks* existentes no arquivo MIDI, a função calcula o *delta time* do evento e verifica se o evento MIDI deve ser importado. Caso o evento seja de um tipo que não é importado, o *delta time* é somado ao próximo evento que será importado. Isto para manter a proporção de tempo entre os eventos importados consistente. O Quadro 3 apresenta os eventos MIDI que são importados pelo aplicativo e o motivo pelo qual eles são considerados.

Quadro 3 – Eventos MIDI carregados pelo aplicativo

Evento MIDI	Função do evento para o aplicativo
Note On	Evento que representa o início da execução de uma nota musical. Este evento possui associação com propriedades sonoras de duração e frequência.
Note Off	Evento que representa o fim da execução de uma nota musical. Este evento possui associação com propriedades sonoras de duração e frequência.
Key Signature	Evento que representa a tonalidade. Este evento possui associação com a propriedade sonora de frequência.
Time Signature	Evento que representa as propriedades de fórmula de compasso. Este evento possui associação com a propriedade sonora de velocidade.
End Of Track	Evento que representa o fim de uma <i>track</i> .

Fonte: elaborado pelo autor.

Para cada *track* processada a função em questão realiza a validação de alguns dados para garantir a consistência dos objetos *Midi* criados. Para este processamento o segundo parâmetro da função é considerado. Este parâmetro é um objeto *Midi* que foi importado anteriormente. Este parâmetro é utilizado para comparar alguns eventos e garantir a compatibilidade entre as opções da fonte de composição. O Quadro 4 demonstra essas validações. Se qualquer uma destas validações falharem o processamento é encerrado e uma mensagem de erro, específica para cada situação, é exibida ao usuário.

Quadro 4 – Relação entre propriedades sonoras editáveis e propriedades da composição

Validação	Descrição
Tamanho de <i>delta time</i>	O tamanho da indicação de <i>delta time</i> não pode ser superior ao valor máximo permitido pelo protocolo.
Evento MIDI não identificado	Cada evento MIDI deve possuir os caracteres de identificação mapeados como constantes da classe <i>MidiFileControl</i> .
Mais de um <i>channel</i> por <i>track</i>	O aplicativo não permite que uma <i>track</i> possua notas sendo executadas em mais de um <i>channel</i> .
Mais de um evento <i>Time Signature</i> com valores diferentes entre eles	O aplicativo não suporta alterações de <i>Time Signature</i> em um objeto <i>Midi</i> . Caso mais de um evento deste tipo seja encontrado e os valores de ambos os eventos sejam iguais, somente o primeiro evento é carregado.
Evento <i>Time Signature</i> com valor diferente do evento de mesmo tipo do objeto de comparação obtido por parâmetro	Para que os dados de composição sejam consistentes entre si o aplicativo exige que os eventos <i>Time Signature</i> de todos os objetos <i>Midi</i> de uma fonte de composição possuam os mesmos valores.
Mais de um evento <i>Key Signature</i> com valores diferentes entre eles	O aplicativo não suporta alterações de <i>Key Signature</i> em um objeto <i>Midi</i> . Caso mais de um evento deste tipo seja encontrado e os valores de ambos os eventos sejam iguais, somente o primeiro evento é carregado.
Evento <i>Time Signature</i> com atributo <i>mode</i> diferente do evento de mesmo tipo do objeto de comparação obtido por parâmetro	Para que os dados de composição sejam consistentes entre si o aplicativo exige que os eventos <i>Key Signature</i> de todos os objetos <i>Midi</i> de uma fonte de composição possuam os mesmos valores para o atributo <i>mode</i> .
Mais de um evento <i>End Of Track</i>	Cada <i>track</i> deve possuir somente um evento de finalização.
Evento <i>End Of Track</i> antes do fim da <i>track</i>	O evento de finalização de <i>track</i> deve ocorrer somente no fim da <i>track</i> .
Ausência de evento <i>End Of Track</i>	O aplicativo exige que um evento <i>End Of Track</i> seja encontrado no arquivo MIDI.
Ausência de evento <i>Time Signature</i>	O aplicativo exige que um evento <i>Time Signature</i> seja encontrado no arquivo MIDI.
Ausência de evento <i>Key Signature</i>	O aplicativo exige que um evento <i>Key Signature</i> seja encontrado no arquivo MIDI.
Ausência de evento <i>Set Tempo</i>	O aplicativo exige que um evento <i>Set Tempo</i> seja encontrado no arquivo MIDI.

Fonte: elaborado pelo autor.



A terceira etapa do processo de carga de dados é a execução do método `loadSavedConfigs` da classe `MusicalCompositionConfigControl`. Este método é responsável por buscar o arquivo contendo os dados de parametrização da fonte de composição e carregar seus valores. O arquivo de configuração é um arquivo JSON nomeado `config.json` que possui como atributos os mesmos atributos da classe `MusicalCompositionConfig`. Este arquivo é salvo após a etapa de parametrização da fonte de composição. A localização do arquivo de configuração varia dependendo do tipo da fonte de composição. Se for uma fonte de composição padrão, o arquivo é encontrado no mesmo caminho relativo em que se encontra a composição, mas, como não é possível gravar dados no diretório do aplicativo, no diretório de dados do aplicativo. Se for uma fonte de composição personalizada, o arquivo se encontra na pasta que representa a fonte de composição. Se a função `loadSavedConfigs` encontrar o arquivo de configuração, é realizada uma verificação de coerência da estrutura do arquivo com a estrutura da fonte de composição que está sendo carregada. Caso a verificação falhe, o arquivo de configuração é apagado e uma mensagem é exibida ao usuário informando sobre a inconsistência. Neste caso o processo de carga de dados é interrompido. Caso o arquivo seja consistente com a estrutura da fonte de composição, todos os valores dos atributos do arquivo de configuração são aplicados aos atributos do objeto `MusicalCompositionConfig`.

A execução do método `determinateMidiChannelsAttributesValues` da classe `MusicalCompositionConfigControl` é a quarta etapa do processo de carga de dados. Este método é responsável por determinar os valores de instrumentos musicais disponíveis e o instrumento musical padrão para cada uma das opções disponíveis na fonte de composição. Sendo assim, o método altera atributos do objeto `MusicalCompositionConfig`. Caso a opção possua um objeto `Midi` que utiliza o canal reservado para os instrumentos de percussão, o método em questão atribui a lista de instrumentos de percussão para o atributo de instrumentos musicais disponíveis para a opção. Neste caso, o valor do atributo de instrumento padrão passa a ser o primeiro elemento da lista em questão. Caso contrário, os valores dos dois atributos são determinados pelos valores do arquivo de configuração. Caso o arquivo não tenha sido carregado, o valor utilizado é a lista de instrumentos melódicos para o atributo de instrumentos musicais disponíveis. O valor do atributo de instrumento musical padrão passa a ser o primeiro elemento da lista de instrumentos musicais disponíveis. As listas de instrumentos musicais de percussão e melódicos são constantes da classe `MusicalCompositionConfigControl`. Os elementos destas listas são expostos no anexo XXX (acho que fica legal a listagem dos instrumentos como anexo).

A quinta etapa do processo de carga de dados é a execução do método `setTempoAndKeySignatureValues` da classe `MusicalCompositionConfigControl`. O método em questão é responsável somente por buscar os valores de fórmula de compasso e modo de tonalidade que serão utilizados na composição. Como todos os objetos `Midi` de uma fonte de composição devem possuir a mesma fórmula de compasso e o mesmo modo de tonalidade, os valores da primeira opção da fonte de composição são adotados para determinar os valores para os atributos `numerator`, `denominator` e `mode` do objeto `MusicalCompositionConfig`.

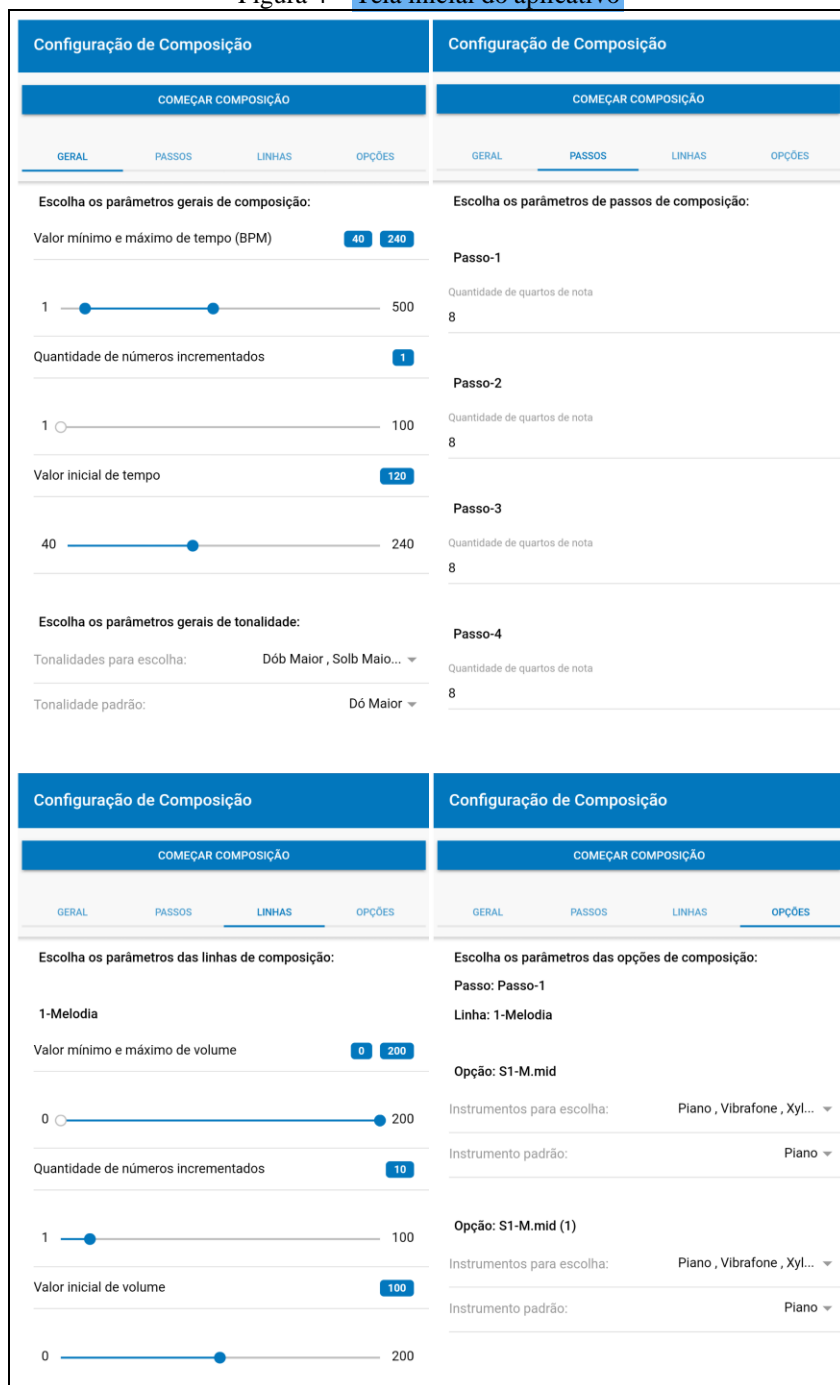
A sexta e última etapa do processo de carga de dados é a execução do método `normalizeTimeDivision` da classe `MusicalCompositionConfigControl`. Este método percorre todos os objetos `Midi` da fonte de composição para determinar o maior valor do atributo `timeDivision.metric` desses objetos. Após o método `ajustMidiTimeDivision` da classe `MidiControl` é executado para cada um dos objetos `Midi` existentes na fonte de composição utilizando como parâmetro o maior valor de `timeDivision.metric`. A classe `MidiControl` é um atributo da classe `MusicalCompositionConfigControl`. O método `ajustMidiTimeDivision` realiza a adequação do valor do atributo `timeDivision.metric` de um objeto `Midi`. Ele recebe como parâmetro o valor que deve ser aplicado ao atributo `timeDivision.metric`, além do próprio objeto `Midi`. Com base no valor atual de `timeDivision.metric` do objeto `Midi` e o valor do parâmetro é gerado um valor de proporção que servirá como fator de multiplicação para os valores de `deltaTime` de todos os eventos do objeto `Midi`. Por exemplo, se o valor atual de `timeDivision.metric` do objeto `Midi` é 100 e o novo valor é 200 o fator de proporção é 2, pois  $2 \times 100 = 200$ . Portanto, para converter o valor de `timeDivision.metric` do objeto `Midi` para 200 é necessário multiplicar os valores de `deltaTime` de todos os eventos de todas as `tracks` do objeto `Midi` por 2. Esta normalização se faz necessária para que, no momento de junções de objetos `Midi` durante a fase de composição, os eventos MIDI se mantenham consistentes em relação à duração dos eventos de cada composição.

### 3.3 CONFIGURAÇÃO DE COMPOSIÇÃO

Uma vez que os dados do objeto `MusicalCompositionConfig` foram carregados pelo processo de carga de dados, a etapa de parametrização do processo de composição se inicia. Nesta etapa, o usuário pode determinar valores para parâmetros de opções, linhas, passos ou fonte de composição. Esses parâmetros serão utilizados durante o processo de composição para determinar quais valores o usuário pode aplicar às propriedades sonoras da composição. Os parâmetros também podem determinar o tamanho mínimo das opções em cada passo ou determinar se dados serão exibidos ou não durante o processo de composição. O funcionamento dos componentes configurados pelos parâmetros

descritos neste subcapítulo será abordado com mais detalhes no próximo subcapítulo. A Figura xxx demonstra a tela de configuração do aplicativo e seus valores para os quatro grupos de parâmetros. Todos os parâmetros alterados nesta tela alteram os atributos do objeto MusicalCompositionConfig.

Figura 4 – Tela inicial do aplicativo



The figure displays two screenshots of the application's configuration screen, showing the 'Configuração de Composição' (Composition Configuration) interface.

**Left Screenshot (GERAL tab):**

- Configuração de Composição** (Composition Configuration)
- COMEÇAR COMPOSIÇÃO** (Start Composition)
- GERAL** (General) tab selected.
- Escolha os parâmetros gerais de composição:** (Choose general composition parameters):
  - Valor mínimo e máximo de tempo (BPM): 40 to 240 (range 40 to 240)
  - Quantidade de números incrementados: 1 (range 1 to 100)
  - Valor inicial de tempo: 120 (range 40 to 240)
- Escolha os parâmetros gerais de tonalidade:** (Choose general tonality parameters):
  - Tonalidades para escolha: Dób Maior, Solb Maio... (range Dób Maior to Solb Maio...)
  - Tonalidade padrão: Dó Maior (range Dó Maior to Dó Maior)

**Right Screenshot (PASSOS tab):**

- Configuração de Composição** (Composition Configuration)
- COMEÇAR COMPOSIÇÃO** (Start Composition)
- PASSOS** (Steps) tab selected.
- Escolha os parâmetros de passos de composição:** (Choose composition steps parameters):
  - Passo-1** (Step-1):
    - Quantidade de quartos de nota: 8 (range 8 to 8)
  - Passo-2** (Step-2):
    - Quantidade de quartos de nota: 8 (range 8 to 8)
  - Passo-3** (Step-3):
    - Quantidade de quartos de nota: 8 (range 8 to 8)
  - Passo-4** (Step-4):
    - Quantidade de quartos de nota: 8 (range 8 to 8)

Fonte: elaborado pelo autor.

Os parâmetros gerais da composição são os que definem os valores que podem ser aplicados para atributos de propriedades sonoras que afetam toda a fonte de composição. Além disso, este grupo de parâmetros possui um parâmetro de controle de exibição do card de informações da composição. Os parâmetros deste grupo afetam o comportamento de três componentes na tela de composição. O primeiro componente é o range do Ionic responsável por determinar a propriedade de tempo da composição. O segundo componente é o componente personalizado list-popover que determina o valor para a tonalidade da composição. o Último componente é o card do Ionic em que são exibidas informações sobre a composição. Os parâmetros gerenciados por este grupo de parâmetros são:

- valor mínimo de tempo: determina o valor mínimo que o usuário pode escolher para a propriedade de tempo da composição. O input do usuário é realizado através de um componente range do Ionic. Os



valores mínimo e máximo para este parâmetro são definidos como constantes na classe `Midi`. O valor definido pelo usuário será utilizado como o valor mínimo do componente `range` que representa o tempo em uma composição;

- b) valor máximo de tempo: determina o valor máximo que o usuário pode escolher para a propriedade de tempo da composição. Os valores mínimo e máximo para este parâmetro são definidos como constantes na classe `Midi`. O `input` do usuário é realizado através do mesmo componente `range` do parâmetro de valor mínimo. O valor definido pelo usuário será utilizado como o valor máximo do componente `range` que representa o tempo em uma composição;
- c) quantidade de número de incrementos: determina o fator de incremento e decremento utilizado pelo componente `range` do Ionic que o usuário irá utilizar na tela de composição para determinar o valor de tempo. O valor deste parâmetro deve estar entre 1 e metade da diferença entre os valores máximo e mínimo determinados nos dois primeiros parâmetros;
- d) valor inicial de tempo: determina o valor que o parâmetro de tempo terá no início da composição. Este valor deve estar entre os valores dos dois primeiros parâmetros;
- e) tonalidades para escolha: determina quais tonalidades podem ser selecionadas pelo usuário na tela de composição. O `input` do usuário é realizado através de um componente `select` do Ionic configurado para aceitar seleção de múltiplos valores. Este parâmetro é uma lista de tonalidades que é utilizada como parâmetro para o componente `list-popover`, que é utilizado na seleção da tonalidade. Os valores desta lista parâmetro são definidos como constantes na classe `Midi`.
- f) tonalidade padrão: determina o valor inicial de tonalidade. O `input` do usuário é realizado através de um componente `select` do Ionic configurado para não aceitar seleção de múltiplos valores. Os valores da lista de tonalidade são definidos como constantes na classe `Midi`.
- g) exibir dados de composição: determina se o `card` de informações da composição será exibido na tela de composição. O `input` do usuário é realizado através de um componente `checkbox` do Ionic

Os parâmetros de `passo` não têm efeito sobre as propriedades sonoras da composição musical. Existe somente um parâmetro por `passo`. Cada parâmetro define o tamanho mínimo dos fragmentos musicais do `passo`. A unidade de medida desse valor é o quarto de nota. Portanto, o nome do parâmetro é quantidade de quartos de nota. Este parâmetro será utilizado na configuração da composição musical garantir que, no mínimo, cada fragmento do `passo` possua a quantidade parametrizada de quartos de nota. Este parâmetro se faz necessário para casos em que o fragmento musical é finalizado com pausas ao invés de notas. Neste caso o arquivo `Midi` pode ser finalizado sem manter os dados de pausas. O parâmetro em questão permite que o aplicativo insira estes dados de pausas se eles não existirem. O `input` destes parâmetros é realizado por componentes de `input` que aceitam somente valores numéricos.

Os parâmetros de `linhas` controlam valores referentes ao parâmetro de volume de cada `linha` da composição. Os parâmetros de `linha` afetam o comportamento de um componente na tela de composição para cada `linha`. Os componentes são `ranges` do Ionic, responsáveis por determinar a propriedade de volume da `linha`. Os parâmetros gerenciados por este grupo de parâmetros, para cada `linha` de composição, são:

- a) valor mínimo de volume: determina o valor mínimo que o usuário pode escolher para a propriedade de volume da `linha`. O `input` do usuário é realizado através de um componente `range` do Ionic. Os valores mínimo e máximo para este parâmetro são definidos como constantes na classe `Midi`. O valor definido pelo usuário será utilizado como o valor mínimo do componente `range` que representa o volume em uma `linha`;
- b) valor máximo de volume: determina o valor máximo que o usuário pode escolher para a propriedade de volume da `linha`. Os valores mínimo e máximo para este parâmetro são definidos como constantes na classe `Midi`. O `input` do usuário é realizado através do mesmo componente `range` do parâmetro de valor mínimo. O valor definido pelo usuário será utilizado como o valor máximo do componente `range` que representa o volume em uma `linha`;
- c) quantidade de número de incrementos: determina o fator de incremento e decremento utilizado pelo componente `range` do Ionic que o usuário irá utilizar na tela de composição para determinar o valor de volume de `linha`. O valor deste parâmetro deve estar entre 1 e metade da diferença entre os valores máximo e mínimo determinados nos dois primeiros parâmetros;
- d) valor inicial de volume: determina o valor que o parâmetro de volume terá no início da composição para a `linha`. Este valor deve estar entre os valores dos dois primeiros parâmetros;

Os parâmetros de `opção` da composição são os que definem os valores que podem ser aplicados para atributos de timbre das opções. Os parâmetros de `opção` afetam o comportamento de um componente na tela de composição para cada `opção`. Os componentes são `buttons` do Ionic, responsáveis por determinar a propriedade de timbre da `opção`. Os parâmetros gerenciados por este grupo de parâmetros, para cada `opção` da composição, são:

- a) instrumentos para escolha: determina quais instrumentos musicais podem ser selecionados pelo usuário na tela de composição. O input do usuário é realizado através de um componente `select` do Ionic configurado para aceitar seleção de múltiplos valores. Este parâmetro é uma lista de instrumentos que é utilizada como parâmetro para o componente `list-popover`, que é utilizado na seleção do instrumento musical. Os valores desta lista parâmetro são definidos no processo de carga de dados de composição.
- b) instrumento padrão: determina o valor inicial de instrumento musical. O input do usuário é realizado através de um componente `select` do Ionic configurado para não aceitar seleção de múltiplos valores. Os valores desta lista parâmetro são definidos no processo de carga de dados de composição.

O processo de parametrização da composição se encerra quando o botão `começar composição` é pressionado. Nesse momento o arquivo de configuração é persistido, de acordo com o tipo da fonte de composição e o objeto `MusicalCompositionControl` é instanciado. Esta classe de controle recebe o objeto `MusicalCompositionConfig` como parâmetro e realiza a configuração do objeto `MusicalComposition`. O processo de instância desse objeto importa todos os valores de configuração para a classe de composição. Nesta etapa também é criada uma estrutura de `Map` de opções e o espectro de todas as opções é gerado.

### 3.4 PROCESSO DE COMPOSIÇÃO

Apresentação dos detalhes da tela de composição

### 3.5 DETALHES DA IMPLEMENTAÇÃO DO APLICATIVO

Apresentação dos detalhes da tela de composição

Nesta seção devem ser descritos os **aspectos mais relevantes de especificação e implementação** para a compreensão sobre o trabalho desenvolvido. O título “DESCRIÇÃO” pode ser complementado com “DO SOFTWARE”, “DA FERRAMENTA” ou “DO PROTÓTIPO” ou aquilo que melhor representar o que foi desenvolvido. A organização desta seção é livre e deve ser seguida uma metodologia própria para cada tipo de trabalho. Reitera-se que, em função da limitação do número de páginas, a descrição deve contemplar o que é mais significativo para a compreensão do que foi desenvolvido.

Destaca-se que os diagramas desenvolvidos bem como outros aspectos de especificação deverão obrigatoriamente constar nos anexos. Opcionalmente poderá constar os códigos mais relevantes da implementação, bem como as telas do trabalho desenvolvido.

## 4 RESULTADOS

De modo a ampliar o seu caráter científico, todos os TCCs devem apresentar e discutir resultados não limitados à comparação com os trabalhos correlatos. Devem ser apresentados os casos de testes do software, destacando objetivo do teste, como foi realizada a coleta de dados e a apresentação dos resultados obtidos, preferencialmente em forma de gráficos ou tabelas, fazendo comentários sobre os mesmos. Também é sugerida a comparação com os trabalhos correlatos apresentados na fundamentação teórica.

## 5 CONCLUSÕES

As conclusões devem refletir os principais resultados alcançados, realizando uma avaliação em relação aos objetivos previamente formulados. Deve-se deixar claro se os objetivos foram atendidos, se as ferramentas utilizadas foram adequadas e quais as principais contribuições do trabalho sociais ou práticas para o seu grupo de usuários bem como para o desenvolvimento científico e ou tecnológico da área.

Deve-se incluir também as limitações e as possíveis extensões do TCC.

## REFERÊNCIAS

As referências devem ser apresentadas em ordem alfabética. Só podem ser inseridas nas referências os documentos citados ao longo da monografia. Todos os documentos citados obrigatoriamente têm que estar inseridos nas referências. A seguir são apresentados alguns exemplos de referências bibliográficas. Destaca-se que deve ser seguida a norma da ABNT.

[parte de um documento:]

AMADO, Gilles. Coesão organizacional e ilusão coletiva. In: MOTTA, Fernando C. P.; FREITAS, Maria E. (Org.). **Vida psíquica e organização**. Rio de Janeiro: FGV, 2000. p. 103-115.

[trabalho acadêmico ou monografia (TCC/Estágio, especialização, dissertação, tese):]

AMBONI, Narcisa F. **Estratégias organizacionais**: um estudo de multicasos em sistemas universitários federais das capitais da região sul do país. 1995. 143 f. Dissertação (Mestrado em Administração) - Curso de Pós-Graduação em Administração, Universidade Federal de Santa Catarina, Florianópolis.

[norma técnica:]

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023**: informação e documentação: referências - elaboração. Rio de Janeiro, 2002a. 24 p.

\_\_\_\_\_. **NBR 10520**: informação e documentação: citações em documentos: apresentação. Rio de Janeiro, 2002b. 7 p.

\_\_\_\_\_. **NBR 14724**: informação e documentação: trabalhos acadêmicos: apresentação. Rio de Janeiro, 2011. 11 p.

[livro:]

BASTOS, Lília R.; PAIXÃO, Lyra; FERNANDES, Lúcia M. **Manual para a elaboração de projetos e relatórios de pesquisa, teses e dissertações**. Rio de Janeiro: Zahar, 1979.

[trabalho acadêmico ou monografia (TCC/Estágio, especialização, dissertação, tese):]

BRUXEL, Jorge L. **Definição de um interpretador para a linguagem Portugal, utilizando gramática de atributos**. 1996. 77 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

[verbete de enciclopédia em meio eletrônico:]

EDITORES gráficos. In: WIKIPEDIA, a enciclopédia livre. [S.l.]: Wikimedia Foundation, 2006. Disponível em: <[http://pt.wikipedia.org/wiki/Editores\\_graficos](http://pt.wikipedia.org/wiki/Editores_graficos)>. Acesso em: 13 maio 2006.

[artigo em evento:]

FRALEIGH, Arnold. The Algerian of independence. In: ANNUAL MEETING OF THE AMERICAN SOCIETY OF INTERNATIONAL LAW, 61, 1967, Washington. **Proceedings...** Washington: Society of International Law, 1967. p. 6-12.

[norma técnica:]

IBGE. **Normas para apresentação tabular**. 3. ed. Rio de Janeiro, 1993. 61 p. Disponível em: <<http://biblioteca.ibge.gov.br/visualizacao/monografias/GEBIS%20-%20RJ/normastabular.pdf>>. Acesso em: 27 ago. 2013.

[artigo em periódico:]

KNUTH, Donald E. Semantic of context-free languages. **Mathematical Systems Theory**, New York, v. 2, n. 2, p. 33-50, jan./mar. 1968.

[trabalho acadêmico ou monografia (TCC/Estágio, especialização, dissertação, tese):]

SCHUBERT, Lucas A. **Aplicativo para controle de ferrovia utilizando processamento em tempo real e redes de Petri**. 2003. 76 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

[página da internet com autor]

SCHULER, João P. S. **Tutorial de Delphi**. Porto Alegre, [2002]. Disponível em: <<http://www.schulers.com/jpss/pascal/dtut/>>. Acesso em: 27 ago. 2013.

[página da internet sem autor]

SCHRATCH. **Program, imagine, share**. [S.l.], [2013?]. Disponível em: <<https://scratch.mit.edu/>>. Acesso em: 27 maio 2013.

[relatório de pesquisa:]

VARGAS, Douglas N. **Editor dirigido por sintaxe**. 1992. Relatório de pesquisa n. 240 arquivado na Pró-Reitoria de Pesquisa, Universidade Regional de Blumenau, Blumenau.

## APÊNDICE A – DIAGRAMAS DE ESPECIFICAÇÃO

**Este apêndice é obrigatório.** Nele devem constar todos os diagramas de especificação desenvolvidos no trabalho. Os diagramas devem conter legendas numeradas na sequência do artigo.

## APÊNDICE B – NOME

Podem ser inseridos outros apêndices no artigo tais como códigos de implementação, telas de interface, instrumentos de coleta de dados, entre outros. **Apêndices são textos elaborados pelo autor** a fim de complementar sua argumentação. Os apêndices são identificados por letras maiúsculas consecutivas, seguidas de um travessão e pelos respectivos títulos. Deve haver no mínimo uma referência no texto anterior para cada apêndice. Colocar sempre um preâmbulo no apêndice. Caso existam tabelas ou ilustrações, identifique-as através da legenda, seguindo a numeração normal das legendas do artigo.

## ANEXO A – DESCRIÇÃO

Elemento opcional, **anexos são documentos não elaborados pelo autor**, que servem de fundamentação, comprovação ou ilustração, como mapas, leis, estatutos, entre outros. Os anexos são identificados por letras maiúsculas consecutivas, seguidas de um travessão e pelos respectivos títulos. Deve haver no mínimo uma referência no texto anterior para cada anexo. Colocar sempre um preâmbulo no anexo. Caso existam tabelas ou ilustrações, identifique-as através da legenda, seguindo a numeração normal das legendas do artigo.

## 6 DESCRIÇÃO DA FORMATAÇÃO

A seguir são apresentadas observações gerais sobre o texto do artigo do Trabalho de Conclusão de Curso (TCC). **Observa-se que esta descrição deve ser retirada do texto final.**

Na confecção do texto deve-se:

- a) usar frases curtas. Segundo Teodorowitsch (2003, p. 3), “Frases com mais de duas linhas aumentam o risco de o leitor não compreender a ideia ou de entendê-la de forma equivocada.”;
- b) usar linguagem impessoal (usar a terceira pessoa do singular) e verbo na voz ativa (a ação é praticada pelo sujeito), com conexão entre os parágrafos;
- c) não usar palavras coloquiais;
- d) não usar palavras repetidas em demasia;
- e) usar verbos no presente quando for referir-se a partes do trabalho que já se encontram disponíveis no texto;
- f) destacar palavras em língua estrangeira em itálico, conforme descrito abaixo:
  - nome de software, ferramenta, aplicativo, linguagem de programação, plataforma, empresa: não deve ser escrito em itálico (exemplos: Delphi 7, Pascal, Object Pascal, Java, JavaScript, Java 2 Micro Edition, Basic, Microsoft Visual C++, C, Windows, Linux, MySQL, Oracle, Eclipse 3.0, Enterprise Architect, Rational Rose, Microsoft, Sun Microsystems),
  - citações: o sobrenome do autor ou o nome da instituição responsável pela autoria do documento citado não deve ser escrito em itálico (exemplo: Segundo Sun Microsystems (2004), ...),
  - palavras em língua estrangeira encontradas nos dicionários nacionais: não devem ser grafadas em itálico (exemplos: software, hardware, web, Internet),
  - demais palavras em língua estrangeira: devem ser escritas em itálico (exemplos: *palmtop*, *classpath*, *play*, etc.). No entanto, Teodorowitsch (2003, p. 7), sugere que alguns termos em língua inglesa devem ser substituídos por termos em português (exemplos: núcleo em vez de *kernel*, aprendizagem de máquina em vez de *machine learning*, etc.);
- g) observar as seguintes regras quanto ao uso de siglas:
  - colocar as siglas entre parênteses precedidas pela forma completa do nome, quando aparecem pela primeira vez no texto (exemplos: Associação Brasileira de Normas Técnicas (ABNT), Trabalho de Conclusão de Curso (TCC)). Caso exista uma lista de siglas na parte pré-textual do volume final, pode-se usar somente a sigla, quando aparecer pela primeira vez no texto,
  - usar apenas a sigla nas demais ocorrências no texto,
  - escrever as siglas em letras maiúsculas e não usar itálico,
  - escrever o plural das siglas sem apóstrofo (exemplos: PCs, APIs, PDAs) e determinar o gênero da sigla conforme o gênero do primeiro substantivo do seu nome (exemplo: o TCC – o Trabalho de Conclusão de Curso).

### 6.1 FORMATAÇÃO

A formatação geral para apresentação do documento, descrita na NBR 14724 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2011), é a seguinte:

- a) o texto divide-se em capítulos, seções e subseções (até cinco divisões);
- b) a apresentação de citações em documentos deve seguir a NBR 10520 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2002b);
- c) a descrição das referências bibliográficas deve estar de acordo com a NBR 6023 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2002a).

Observa-se ainda que todo capítulo, seção ou subseção deve ter no mínimo um texto relacionado.

A monografia deve ser digitada usando as fontes e formatação de parágrafos deste modelo, indicadas no Quadro 5.

Quadro 5– Estilos do modelo

USO	FORMATO
título de capítulo ou seção primária (1)	TF-TÍTULO 1 (Times New Roman, 10pt, negrito, maiúsculas)
título de seção secundária (1.1)	TF-TÍTULO 2 (Times New Roman, 10pt, maiúsculas)
título de seção terciária (1.1.1)	TF-Título 3 (Times New Roman, 10pt, minúsculas, exceto a 1ª letra da 1ª palavra do título e de nomes próprios)
título de seção quaternária (1.1.1.1)	TF-Título 4 (mesma formatação seção ternária)
título de seção quinária (1.1.1.1.1)	TF-Título 5 (mesma formatação seção ternária)
texto	TF-TEXTO (Times New Roman, 10pt)
citação direta com mais de três linhas	TF-CITAÇÃO (Times New Roman, 9pt, com recuo de 4 cm)
itens (alíneas)	ver descrição abaixo (Times New Roman, 10pt)
referência bibliográfica	TF-REFERÊNCIA ITEM (Times New Roman, 10pt, alinhada à margem esquerda)
fonte, legenda, texto de quadro/tabela e figura	TF-FONTE (Times New Roman, 9pt, alinhada à margem esquerda do quadro/tabela) TF-LEGENDA, (Times New Roman, 10pt, centralizada) TF-TEXTO- QUADRO (Times New Roman, 10pt) TF-FIGURA (Times New Roman, 10pt, centralizada)

Fonte: elaborado pelo autor.

O espaçamento, também definido no modelo, deve ser conforme indicado no Quadro 6.

Quadro 6 - Espaçamento

USO	ESPAÇAMENTO
título de capítulo ou seção primária (1)	espaço simples, com 12pt antes do parágrafo
título de seção secundária (1.1)	
título de seção terciária (1.1.1)	
título de seção quaternária (1.1.1.1)	
título da seção quinária (1.1.1.1.1)	
texto	espaço simples, com 6 pt antes do parágrafo
citação direta com mais de três linhas	espaço simples com 6pt antes e depois do parágrafo
itens (alíneas)	espaço simples, com 6 pt antes do parágrafo
referência bibliográfica	espaço simples, com 6 pt antes do parágrafo
legenda e texto de ilustração/tabela	espaço simples, com 6 pt antes do parágrafo
fonte	espaço simples, com 0pt antes do parágrafo

Fonte: elaborado pelo autor.

Na disposição gráfica de itens (alíneas) devem ser observados os seguintes quesitos:

- o texto que antecede os itens termina com dois pontos;
- cada item deve iniciar com uma letra minúscula seguida de fecho parênteses e terminar com um ponto e vírgula, sendo que o último item termina com ponto (FORMATO: TF-ALÍNEA);
- o texto de cada item inicia com letra minúscula, exceto nomes próprios;
- quando contiver subitens, os mesmos devem iniciar com hífen colocado sob a primeira letra do texto do item correspondente (FORMATO: TF-SUBALÍNEA nível 1 ou TF-SUBALÍNEA nível 2, conforme o caso). Nesse caso, cada subitem deve terminar com uma vírgula, exceto o último que termina com ponto ou com ponto e vírgula.

Segue um exemplo:

- cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula (FORMATO: TF-ALÍNEA);
- cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula (FORMATO: TF-ALÍNEA):
  - cada subitem (nível 1) inicia com letra minúscula, cada subitem (nível 1) inicia com letra minúscula (FORMATO: TF-SUBALÍNEA nível 1):
    - cada subitem (nível 2) inicia com letra minúscula, cada subitem (nível 2) inicia com letra minúscula (FORMATO: TF-SUBALÍNEA nível 2),
    - cada subitem (nível 2) inicia com letra minúscula, cada subitem (nível 2) inicia com letra minúscula (FORMATO: TF-SUBALÍNEA nível 2),
  - cada subitem (nível 1) inicia com letra minúscula, cada subitem (nível 1) inicia com letra minúscula

(FORMATO: TF-SUBALÍNEA nível 1);

- c) cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula (FORMATO: TF-ALÍNEA).

#### 6.1.1.1 Exemplo de título de seção quaternária [FORMATO: TF-TÍTULO 4]

Formato: TF-TEXTO.

##### 6.1.1.1.1 Exemplo de título de seção quinária [FORMATO: TF-TÍTULO 5]

Formato: TF-TEXTO.

#### 6.1.2 Formatação de quadros, figuras e tabelas

Um quadro contém apenas informações textuais, que podem ser agrupadas em colunas. Uma figura contém, além das informações textuais, pelo menos um elemento gráfico. Uma tabela é uma apresentação tabular de informações **numéricas** relacionadas.

Os quadros, figuras e tabelas são identificados na parte superior por uma legenda (a qual deve estar centralizada) composta pela palavra designativa (Figura, Quadro ou Tabela, conforme o caso), seguida de seu número em algarismo arábico (usar numeração progressiva, uma sequência para os quadros, outra para as figuras e outra para as tabelas), de hífen e do título. As ilustrações devem:

- aparecer centralizadas no texto;
- estar delimitadas por uma moldura simples;
- aparecer numa única página (quando o tamanho não exceder o da página), inclusive a legenda;
- serem inseridas o mais próximo possível do trecho a que se referem pela primeira vez.

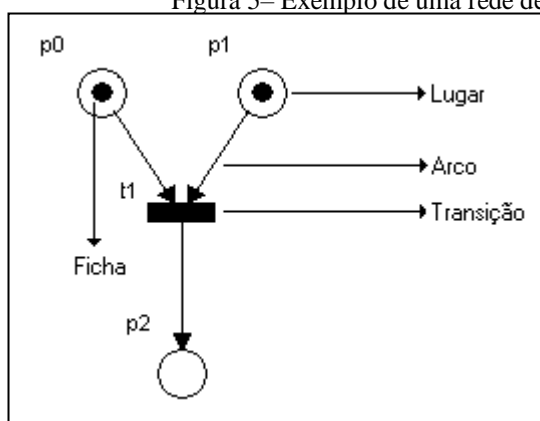
Toda ilustração deve ter fonte, alinhada à margem esquerda da ilustração. Quando foi o próprio autor que fez a ilustração, deve inserir o texto: “Fonte: elaborado pelo autor”.

Observa-se que quando um código fonte for descrito dentro de um quadro, deve-se utilizar letra do tipo `courier new 9pt.` (TF-CÓDIGO-FONTE)

Exemplos de como se deve referenciar uma figura e um quadro e como descrevê-los são mostrados a seguir.]

Um exemplo de uma rede de Petri pode ser visto na Figura 5.

Figura 5– Exemplo de uma rede de Petri



Fonte: Schubert (2003, p. 18).

Um exemplo de código fonte gerado a partir de uma especificação pode ser visto no Quadro 7.

Quadro 7 – Funções que verificam se as transições estão sensibilizadas

```
Function TEstruturaMalha.T1Sensibilizada: boolean;
begin
  result := (Fp2 and Fp4);
end;

function TEstruturaMalha.T2Sensibilizada: boolean;
begin
  result := (Fp1 and Fp3);
end;

function TEstruturaMalha.T3Sensibilizada: boolean;
begin
  result := (Fp2 and Fp4);
end;
```

Fonte: Schubert (2003, p. 63).

Não devem ser usadas bordas (linhas) verticais para fechar as tabelas. Exemplo de como se deve referenciar uma tabela e como descrevê-la é mostrado a seguir.

A quantidade de trabalhos finais realizados no Curso de Ciência da Computação (de 2010 até 2014) é apresentada na Tabela 1.

Tabela 1 – Trabalhos finais realizados no Curso de Ciência da Computação

Ano	Estágios	TCC's	Totais
2010/1	0	16	16
2010/2	0	21	21
2011/1	0	25	25
2011/2	0	23	23
2012/1	0	23	23
2012/2	0	22	22
2013/1	0	25	25
2013/2	0	16	16
2014/1	0	18	18
2014/2	0	13	13
	<b>0</b>	<b>202</b>	<b>202</b>

Fonte: elaborado pelo autor.

### 6.1.3 Exemplos de citações retiradas de documentos ou de nomes constituintes de uma entidade

A apresentação de citações em documentos deve seguir a NBR 10520 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2002b). O sistema a ser usado é o alfabético. Exemplos de citações são: “Numa publicação recente (SEBESTA, 2000) é exposto ...” e “Segundo Silva et al. (1987), execução controlada de programas é ...”.

Quando a citação referir-se a uma parte específica do documento consultado, especificar no texto da monografia a(s) página(s). Esta(s) deverá(ão) seguir a data, separada(s) por vírgula(s) e precedida(s) pelo designativo que a(s) caracteriza(m). Como exemplo, mostra-se: “(SCHIMT, 1999, p. 50)” ou “... visto que Schimt (1999, p. 50) implementou ...”.

As citações diretas (transcrição textual de parte da obra do autor consultado), no texto, com mais de três linhas, devem ser destacadas com recuo de 4 cm da margem esquerda, com letra menor que a do texto utilizado e sem aspas (FORMATO: TF-CITAÇÃO), conforme o exemplo a seguir.

A Associação Brasileira de Normas Técnicas (ABNT) é o Fórum Nacional de Normalização. As Normas Brasileiras, cujo conteúdo é de responsabilidade dos Comitês Brasileiros (ABNT/CB) e dos Organismos de Normalização Setorial (ABNT/ONS), são elaboradas por Comissões de Estudo (CE), formadas por representantes dos setores envolvidos, delas fazendo parte: produtores, consumidores e neutros (universidades, laboratórios e outros). (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2002b, p. 1).

Quando da citação de um nome (identificador) constituinte de uma entidade em um texto, deve-se utilizar o tipo de letra *courier new*, com tamanho dez (10). Para facilitar a formatação, existe o estilo de palavra denominado TF-COURIER10. Como exemplo cita-se nome de classe, atributo ou método. A seguir é apresentado um exemplo.

As classes `TTabelaTransicao` e `TExpressaoRegular` são classes de interface, porém estão sendo consideradas como classes de domínio da aplicação.