



UNIVERSIDADE ESTADUAL DO MARANHÃO

Curso de Engenharia Mecânica

ANDERSON PORTO COSTA

**PROTOTIPAGEM E DESENVOLVIMENTO DE UM APLICATIVO PARA
DIMENSIONAMENTO DE ENGRENAGENS CILÍNDRICAS DE DENTES RETOS**

SÃO LUÍS

2019

ANDERSON PORTO COSTA

**PROTOTIPAGEM E DESENVOLVIMENTO DE UM APLICATIVO PARA
DIMENSIONAMENTO DE ENGRENAGENS CILÍNDRICAS DE DENTES RETOS**

Monografia de graduação apresentada ao Curso de Engenharia Mecânica da Universidade Estadual do Maranhão como parte dos requisitos exigidos para obtenção do título Bacharel em Engenharia Mecânica.

Orientador (a): Prof. Me. Maria Amália Trindade de Castro
Co-orientador: Prof. Me. Carlos Ronyhelton Santana de Oliveira

ESTE EXEMPLAR CORRESPONDE A
VERSÃO FINAL DA MONOGRAFIA
DEFENDIDA POR ANDERSON PORTO
COSTA, E ORIENTADA PELO(A) PROF.
ME. MARIA AMÁLIA TRINDADE DE
CASTRO

.....
ASSINATURA DO(A) ORIENTADOR(A)

SÃO LUÍS

2019

Costa, Anderson Porto.

Prototipagem e desenvolvimento de um aplicativo para dimensionamento de engrenagens cilíndricas de dentes retos / Anderson Porto Costa. – São Luís, 2019.

60 f.

Monografia (Graduação) – Curso de Engenharia Mecânica, Universidade Estadual do Maranhão, 2019.

Orientador: Profa. Ma. Maria Amália Trindade de Castro.

UNIVERSIDADE ESTADUAL DO MARANHÃO
CURSO DE ENGENHARIA MECÂNICA
DEPARTAMENTO DE ENGENHARIA MECÂNICA E PRODUÇÃO

TRABALHO DE CONCLUSÃO DE CURSO

**PROTOTIPAGEM E DESENVOLVIMENTO DE UM APLICATIVO PARA
DIMENSIONAMENTO DE ENGRENAGENS CILÍNDRICAS DE DENTES RETOS**

Autor: Anderson Porto Costa

Orientador (a): Prof. Me. Maria Amália Trindade de Castro

Co-orientador: Prof. Me. Carlos Ronyhelton Santana de Oliveira

A Banca Examinadora composta pelos membros abaixo aprovou esta Monografia:

Prof. Me. Maria Amália Trindade de Castro, Presidente

Instituição: Universidade Estadual do Maranhão – UEMA

Prof. Me. Carlos Ronyhelton Santana de Oliveira

Instituição: Universidade Estadual do Maranhão - UEMA

Prof. Me. Kaio Henrique Ferreira Nogueira de Nogueira

Instituição: Universidade Estadual do Maranhão - UEMA

A Ata da defesa com as respectivas assinaturas dos membros encontra-se no processo de vida acadêmica do aluno.

São Luís/MA, 28 de junho de 2019.

DEDICATÓRIA

Dedico a Deus e aos meus Pais por sempre terem me apoiado e me dado todo o amor e carinho.

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus pelo dom da vida e por Ele ter me conduzido e abençoado em toda essa trajetória.

À minha família, em especial à minha mãe Joselita Porto por todo amor e inspiração todos os dias e pela sua luta para que eu chegasse até aqui, ao meu pai Manoel Martiniano por me inspirar todos os dias com sua força, honestidade e compaixão pelo próximo e pela ajuda durante todos esses anos, à minha mãezinha Ana Maria Porto por todo cuidado e amor em todos esses anos da minha vida, à minha Madrinha Ana Lúcia por todo amor e por sempre acreditar em mim, à minha Tia Lilian, Amélia, Carmelita e Alexsandra por todo cuidado durante todos esses anos e pelo amor demonstrado, ao meu Padrinho José Raimundo por todo apoio e carinho que pude receber durante a minha vida.

Aos meus irmãos Ana Beatriz, Lucas, Karen e Sheila que amo muito e são muito especiais na minha vida. A todos meus primos que crescemos juntos e podemos compartilhar momentos incríveis.

À minha orientadora Professora Me. Amália Trindade Castro, ao meu Coorientador Professor Me. Carlos Ronyehlton por ter aceito esse desafio e ter sido um grande amigo durante esse tempo. Aos meus grandes amigos Alberto Félix, Luan Lemos, Sidney Sá, Marvin Barros, Andrews Anderson que pude conhecer durante a graduação e que levarei para a vida inteira.

Aos meus grandes amigos que o futsal pôde me proporcionar e que levarei para sempre comigo essa amizade. Aos meus amigos que convivem comigo diariamente Iderlan Alves e Matheus Luis por toda ajuda e momentos de alegria durante esses anos dividindo apartamento, onde criamos uma amizade muito grande.

Aos meus amigos Victor Hugo, Tainam, José Antônio, Gabriel Sousa e Adriano Novais por ter proporcionado grandes momentos de alegria durante esses anos de amizade. Ao meu grande amigo Carlão por ser sempre uma pessoa boa e tratar sempre bem durante todos esses anos. Aos meus amigos de Caxias que sempre vou levar comigo em especial Antunes Costa e o José Igo, famoso trio parada dura.

À minha grande e maravilhosa namorada Anne Rithelly por todo o amor, paciência, companheirismo e cuidado

As minhas amigas cobras da biologia Amanda, Laiza, Samantha e Luh e a todos os meus professores, que muito admiro e contribuíram para minha formação profissional e pessoal.

À Universidade Estadual do Maranhão pela formação e a todos aqueles amigos de universidade que contribuíram direta ou indiretamente durante essa caminhada.

Muito obrigado!

*“Deem graças ao Senhor, porque ele é bom.
O seu amor dura para sempre!”*

Salmos 136:1

RESUMO

As ECDR's (engrenagens cilíndricas de dentes retos) possuem um grande número de aplicações nas indústrias e em diversos equipamentos presentes em setores da sociedade que utilizam como princípio a transmissão de potência. O presente estudo buscou realizar o projeto e desenvolvimento de um aplicativo para dimensionamento de engrenagens cilíndricas de dentes retos para dispositivos móveis portadores do sistema operacional Android. O ambiente de desenvolvimento integrado (IDE) utilizado para a criação do aplicativo foi o Android Studio que facilita um desenvolvimento acelerado e auxilia na criação de aplicativos de alta qualidade para todos os dispositivos Android. Após todas as implementações de código do aplicativo foram realizados exercícios de simulação de projetos que se encaixam no contexto do aplicativo, com propósito de validação do mesmo. Assim, pode-se observar a partir dos resultados obtidos a eficiência do aplicativo, uma vez que, ele visa a minimização do tempo utilizado para cálculos de ECDRs.

Palavras-chave: Engrenagens cilíndricas de dentes retos; Aplicativos; Android.

ABSTRACT

The ECDR's have large number of operations in the industries and various equipment presente in the other sectors of society that use as principle of power transmission. The presente study sought to develop an app for sizing of cylindrical gears of straight teeth for mobile devices carriers of the Android operating system. The program used to create the application was Android Studio that has an accelerated development and assists in creation high-quality apps for all the Android devices. After the application development was made examples that fits in the app contexto, to case there are possible crashes in the application and so correct them. Thus, it can be observed from the obtained results the efficiency of the application, since it aims the minimization of the time used to sizing calculations.

Keywords: Cylindrical Gears of Straight Teeth; Application; Android

LISTA DE ILUSTRAÇÕES

Figura 4.1 - engrenagens helicoidais com eixos paralelos.	22
Figura 4.2 - Engrenagens helicoidais com eixos perpendiculares.	22
Figura 4.3 - Engrenagens cônicas de dentes retos.	23
Figura 4.4 - Engrenagens cônicas de dentes helicoidais.	24
Figura 4.5 - Engrenagem Parafuso sem fim.	25
Figura 4.6 – Conjunto de engrenagens cilíndricas de dentes retos.....	26
Figura 4.7- Elementos básicos de uma engrenagem de dentes retos.	27
Figura 4.8 - Imagem do Android Studio	35
Figura 5.1 - Template utilizado no projeto.....	41
Figura 5.2 - Exemplo de utilização de TextView e EditText.....	42
Figura 5.3 - Implementação de Buttons no aplicativo.	42
Figura 5.4 - Exemplo de utilização de TextView e EditText para transmissões.	43
Figura 5.5 - Implementação de Buttons para transmissões.....	44
Figura 6.1 - Exemplo do dimensionamento da engrenagem com módulo e n° de dentes.	52
Figura 6.2 - Dimensionamento da engrenagem com módulo e diâmetro primitivo	53
Figura 6.3 - Dimensionamento da engrenagem com número de dentes e diâmetro primitivo.	54
Figura 6.4 - Resultado da transmissão de engrenagens por meio do d. primitivo e rotação.	55
Figura 6.5 - Resultado da transmissão de engrenagens por meio do d. primitivo e torque.	56
Figura 6.6 - Resultado da transmissão de engrenagens por meio do n° de dentes e rotação.	57
Figura 6.7 - Resultado da transmissão de engrenagens por meio do n° de dentes e torque.....	57

LISTA DE TABELAS

Tabela 4.1 - Elementos Básicos de Engrenagens	28
---	----

LISTA DE ABREVIATURAS E SIGLAS

α	-	Ângulo de pressão [°]
ω	-	Rotação [rad/s]
Db	-	Diâmetro da base [mm]
De	-	Diâmetro externo [mm]
Di	-	Diâmetro interno [mm]
Dp	-	Diâmetro primitivo [mm]
DTD	-	Definition Type Document
DTS	-	Definition Type Scheme
E	-	Espessura da engrenagem [mm]
ECDR	-	Engrenagens cilíndricas de dentes retos
F	-	Folga do dente [mm]
GUI	-	Graphical User Interface
H	-	Altura do dente [mm]
I	-	Relação de transmissão
IDE	-	Integrated Development Environment
JVM	-	Java Virtual Machine
m	-	Módulo [mm]
mm	-	Milímetro
N	-	Newton
p	-	Passo circular [mm]
POO	-	Programação Orientada a Objetos
rad	-	Radianos
s	-	Segundos
S	-	Espessura de dente [mm]
SDK	-	Software Development Kits
SGML	-	Standard General Markup Language
SO	-	Sistema Operacional

- T - Torque [N*mm]
- XML - eXtensible Markup Language
- Z - Número de dientes

SUMÁRIO

FOLHA DE APROVAÇÃO	iii
AGRADECIMENTOS	v
RESUMO	viii
ABSTRACT	ix
LISTA DE ILUSTRAÇÕES	x
LISTA DE TABELAS	xi
LISTA DE ABREVIATURAS E SIGLAS	xii
1 INTRODUÇÃO	17
2 JUSTIFICATIVA	19
3 OBJETIVO	20
3.1 Objetivo Geral:	20
3.2 Objetivos específicos:	20
4 FUNDAMENTAÇÃO TEÓRICA	21
4.1 Engrenagens	21
4.2 Tipos de Engrenagens	21
4.2.1 Engrenagens cilíndricas de dentes helicoidais	22
4.2.2 Engrenagens Cônicas	23
4.2.3 Parafuso sem fim	24
4.2.4 Engrenagens cilíndricas de dentes retos.....	25
4.3 XML	29

4.3.1	Características do XML	30
4.3.2	Regras do XML	31
4.3.3	Benefícios do XML	32
4.4	Linguagem de programação Java	33
4.5	ANDROID STUDIO	35
4.5.1	Outros IDEs para desenvolvimento de aplicativos Android	35
5	METODOLOGIA	37
5.1	Desenvolvimento do aplicativo para dimensionamento das ECDRs.....	40
6	RESULTADOS.....	52
7	CONSIDERAÇÕES FINAIS	58
	REFERÊNCIAS.....	59

1 INTRODUÇÃO

Engrenagens são elementos mecânicos rígidos dotados de dentes, utilizados na transmissão de movimentos rotativos entre eixos, em sistemas que transferem potência de uma unidade motora para uma movida. No princípio de sua aplicação ocorreu durante o berço da civilização. Naquele período, as engrenagens eram confeccionadas com discos de madeira, no qual os dentes eram produzidos separados e depois montados. Elas faziam parte principalmente de rodas d'água para moinhos. Atualmente, elas continuam sendo amplamente utilizadas, porém os materiais usados são de resistência mais elevada e os dentes são integrados ao disco (MAITRA, 2013). As engrenagens cilíndricas de dentes retos (ECCR) têm a geometria mais simplificada dentre todas e são as que ocupam maior espaço no mercado. São basicamente rodas dentadas fabricadas a partir de um cilindro em que o perfil do dente é obtido geralmente por um processo de usinagem.

As ECCR's fazem parte de diversas máquinas e produtos dos mais variados campos de aplicação. Desde a área industrial até em equipamentos eletrodomésticos. A responsabilidade da engrenagem diante o funcionamento de uma determinada máquina ou produto é alta, pois a quebra dela pode acarretar em parada do equipamento, gerando elevadas perdas na produção e conseqüentemente redução do lucro das empresas. Portanto, a análise estrutural desse componente, na fase de projeto, é extremamente necessária. Esses dimensionamentos são baseados em normas específicas para engrenagens, as quais são rigorosas e conservadoras, a fim de garantir que esse elemento de máquina resistirá aos carregamentos aplicados durante um período desejado de funcionamento (BETIM,2015).

Ao longo dos anos, diversas áreas da engenharia têm evoluído bastante. Durante essa evolução, o grande foco tem sido melhorar a eficiência na prática de diversas atividades já conhecida pelo engenheiro há um certo tempo.

A evolução da tecnologia dos aparelhos celulares permitiu oferecer ao usuário recursos que vão muito além da realização de uma chamada ou de envio de uma mensagem de texto. As melhorias de hardware dos aparelhos celulares permitiram o desenvolvimento de sistemas operacionais mais avançados. Com sistema de desenvolvimento mais avançado foi possível desenvolver aplicativos melhores, com cada vez mais recursos e serviço ao usuário.

A utilização de aplicativos vem sendo cada vez mais frequente, os avanços tecnológicos têm trazido novidades que ajudam no desenvolvimento e na praticidade tornando a vida das pessoas mais simples em todos os parâmetros.

Mediante o exposto, é proposto nesse trabalho o desenvolvimento de um aplicativo para dispositivos móveis de fácil manuseio que possibilite ao usuário realizar um prévio dimensionamento de engrenagens cilíndricas de dentes retos, uma vez que, dimensionar engrenagens pode ser um trabalho demorado e no final do processo pode ser que o resultado não seja satisfatório. De acordo com Andrade (2016), a complexidade do dimensionamento não está na aplicação da teoria, mas sim na quantidade de decisões que deve ser feita. Isso faz com que a mesma metodologia de cálculos seja tenha que ser repetida várias vezes até encontrar o resultado desejado.

2 JUSTIFICATIVA

Realizar a etapa de dimensionamento de forma manual é muito trabalhoso, por isso é de extrema importância que esse procedimento seja facilitado, uma forma eficaz de atingir esse objetivo é a utilização de aplicativos.

Foram selecionadas as plataformas de dispositivos móveis, tais como smartphones e tablets como foco para o aplicativo proposto, pois com o crescente aumento da tecnologia, aos poucos o homem pode obter qualquer conhecimento e ter alguns recursos em suas mãos e fica apto a desenvolver qualquer atividade. Segundo a Agência Nacional de Telecomunicações (Anatel), no mês de fevereiro de 2019 o Brasil possuía um total de 229 milhões de celulares. Se compararmos com o número da população brasileira de 209 milhões (IBGE,2019), há 1,1 celulares para cada habitante.

Atualmente esses dispositivos tem aumentado consideravelmente sua capacidade de processamento, tornado possível o desenvolvimento de aplicativos voltados para as diversas áreas das engenharias. O aplicativo tem o intuito de ajudar no desenvolvimento da engrenagem durante o processo, mostrar as consequências de cada escolha feita e ajudar a encontrar a melhor combinação para a mesma.

3 OBJETIVO

3.1 Objetivo Geral:

Desenvolver um aplicativo para dispositivos móveis portadores do sistema operacional Android com ênfase no dimensionamento de engrenagens cilíndricas de dentes retos.

3.2 Objetivos específicos:

- ✓ Implementar uma interface gráfica de usuário (GUI) baseada nos conceitos do *Material Design* do Google;
- ✓ Desenvolver a metodologia de cálculo de engrenagens cilíndricas de dentes retos através da codificação em linguagem Java;
- ✓ Aplicar conceitos de Programação Orientada a Objetos (POO) no desenvolvimento do aplicativo através do ambiente de desenvolvimento integrado (IDE) Android Studio.
- ✓ Desenvolver o aplicativo para que seja intuitivo e de fácil utilização pelo usuário comum e principalmente por profissionais das áreas de engenharia correlatas a prática de dimensionamento de engrenagens.

4 FUNDAMENTAÇÃO TEÓRICA

4.1 Engrenagens

Denomina-se engrenagem a peça de forma cilíndrica (engrenagem cilíndrica), cônica (engrenagem cônica) ou reta (cremalheira), dotada de dentes em sua superfície externa ou interna, cuja finalidade é transmitir movimento sem deslizamento e potência, multiplicando os esforços com finalidade de gerar trabalho (JUVINAL, 2008).

Essas são frequentemente empregadas, tanto para eixos paralelos como para eixos reversos ou concorrentes, servindo para potências, rotações e relações de multiplicação, que variam desde valores mínimos até máximos. Distinguem-se pela transmissão de forças sem deslizamento (relação de multiplicação constante e independente da carga), pela segurança de funcionamento e pela vida útil, pela resistência às sobrecargas e pequena manutenção, pelas suas dimensões reduzidas e devido ao seu alto rendimento. Por outro lado, deve-se levar em conta o seu maior custo bem como os maiores ruídos durante o funcionamento e a transmissão relativamente rígida (eventualmente deve ser previsto um acoplamento elástico para amortecer os choques) (NIEMANN, 1971).

4.2 Tipos de Engrenagens

Dentre os diversos tipos de engrenagens existentes, existem 4 tipos básicos de engrenagens:

- Engrenagem cilíndricas de dentes retos;
- Engrenagem cilíndricas de dentes helicoidais;
- Engrenagens cônicas;
- Parafuso sem fim.

A seguir, terá uma breve análise de cada tipo.

4.2.1 Engrenagens cilíndricas de dentes helicoidais

As engrenagens cilíndricas de dentes helicoidais possuem dentes inclinados com relação ao eixo de rotação e não possuem um perfil reto (figura 4.1). Os dentes inclinados possuem um ângulo de inclinação denominado ângulo de hélice, que variam de 10° a 40° . São utilizadas para transmissões de grandes potências.



Figura 4.1 - Engrenagens helicoidais com eixos paralelos.
Fonte: ANDRADE (2018)

Em alguns casos essas engrenagens não possuem eixos paralelamente entre si (figura 4.2). O movimento se dá pelo escorregamento entre os dentes, possuindo apenas um ponto de contato entre as engrenagens. Com isso reduz sua capacidade de transmissão de torque e potência.



Figura 4.2 - Engrenagens helicoidais com eixos perpendiculares.
Fonte: JUNIOR (2019)

Uma das vantagens de se usar engrenagens helicoidais consiste no fato de os dentes engatam pouco a pouco, isso gera um impacto menor e conseqüentemente são mais silenciosas e causam menos vibrações. Outra vantagem é seu ângulo de hélice, o número de dentes em contato será maior.

Uma desvantagem do uso de dentes helicoidais, pois a carga axial gerada nos dentes, os mancais devem ser projetos para que suporte essas cargas.

As engrenagens cilíndricas de dentes helicoidais são utilizadas em vários segmentos da indústria, porém, sendo comum em indústria automobilística, em caixas de marchas, na indústria siderúrgica, na indústria ferroviária e alimentícia também, tendo sempre uma ampla ligação com grandes cargas e velocidades. São utilizadas em transmissões de rotações elevadas.

4.2.2 Engrenagens Cônicas

Essas engrenagens possuem o formato cônica, podendo ser com dentes retos (figura 4.3) ou helicoidais (figura 4.4).



Figura 4.3 - Engrenagens cônicas de dentes retos.
Fonte: ANDRADE (2018).



Figura 4.4 - Engrenagens cônicas de dentes helicoidais.
Fonte: ANDRADE (2018)

A maior vantagem da engrenagem cônica de dentes retos é que ela pode mudar a direção de um giro e sua força, podendo transmitir a energia entre eixos com diferentes ângulos entre eles. Outro ponto a ser levado em conta, esses tipos de engrenagens não suportam altas velocidades e cargas, pelo fato dos dentes não serem paralelos entre si.

As engrenagens cônicas de dentes helicoidais, sua forma permite que se tenha um grande aproveitamento de energia que oferece ao sistema. Essas engrenagens possui uma maior eficiência e diminui os ruídos.

São produzidas para máquinas com sistemas de cargas maiores e altas velocidades, assim como as cilíndricas.

4.2.3 Parafuso sem fim

Essas engrenagens operam diferentes em relação as demais apresentadas, são utilizadas quando buscam velocidades de transmissão reduzidas. Sua principal vantagem é possibilitar grandes reduções, desde de 1:1 chegando até valores maiores que 1:300.

Outro fato é que engrenagens sem-fim diferentemente das demais citadas, o eixo (rosca sem-fim), gira a engrenagem (podendo ser de dentes retos) facilmente (figura 4.5), mas a engrenagem não consegue girar o eixo. Essa característica é útil para máquinas como

transportadores, nos quais a função de travamento pode agir como um freio para a esteira quando o motor não estiver funcionando.



Figura 4.5 - Engrenagem Parafuso sem fim.
Fonte: ANDRADE (2018).

4.2.4 Engrenagens cilíndricas de dentes retos

Nesse trabalho será abordado engrenagens cilíndricas de dentes retos. Um dos mais comuns são as engrenagens cilíndricas de dentes retos. Nestas engrenagens os dentes são dispostos paralelamente entre si e em relação ao eixo, como pode ser observado na Figura 4.6.



Figura 4.6 – Conjunto de engrenagens cilíndricas de dentes retos.
Fonte: ANDRADE (2018).

Segundo Pezzano e Klein (1957) as engrenagens de dentes retos são usadas em transmissões que necessitam de uma mudança de posição das engrenagens em serviço, pois apresentam uma maior facilidade de “engatar”. Este tipo de engrenagem é mais empregado em transmissões de baixa do que de alta rotação, devido principalmente ao elevado nível de ruído que produz. Essas engrenagens são mais simples e possuem um custo de produção mais barato em relação as demais.

Estes elementos permitem a redução ou aumento de torque e velocidades, com mínimas perdas de energia. A mudança de velocidade e torção é feita na razão dos diâmetros primitivos. Aumentando a rotação, o momento torçor diminui e vice-versa. Assim, num par de engrenagens, a maior delas terá sempre rotação menor e transmitirá momento torçor maior. O movimento dos dentes entre si processa-se de tal modo que no diâmetro primitivo não há deslizamento, havendo apenas rolamento. Nas demais partes do flanco, existem ação de deslizamento e rolamento. A relação de transmissão é um parâmetro de extrema importância em uma transmissão por engrenagens, pois representa se a transmissão será do tipo redutora ou amplificadora (KODA, 2009).

Os elementos básicos das engrenagens são mostrados na Figura 4.7 e a Tabela 4.1 apresenta a simbologia e o significado de cada elemento.

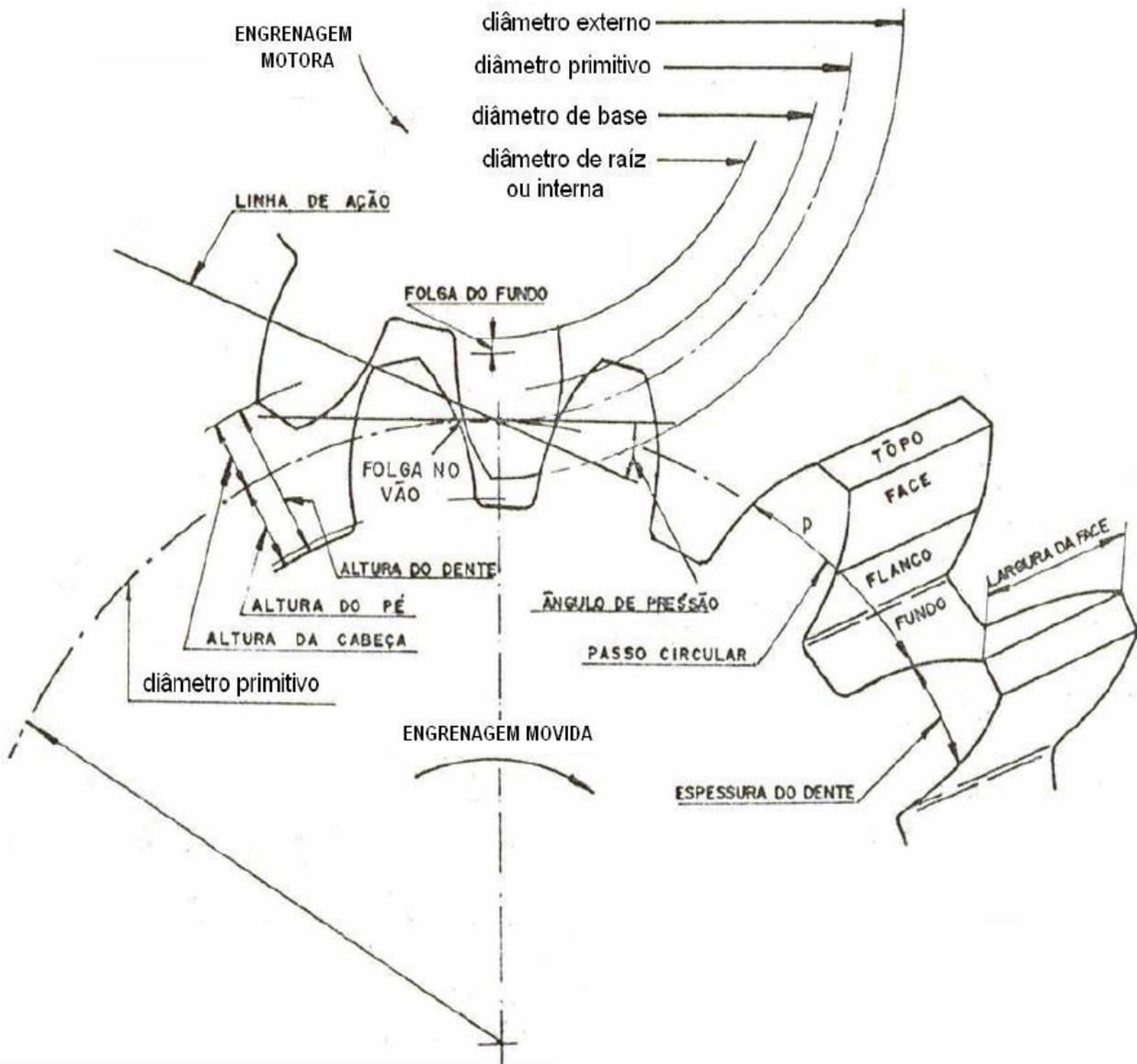


Figura 4.7- Elementos básicos de uma engrenagem de dentes retos.
 Fonte: Pezzano e Klein, 1957).

Tabela 4.1 - Elementos Básicos de Engrenagens

Simbologia	Significado
De	Diâmetro externo
Di	Diâmetro interno
Dp	Diâmetro primitivo
Db	Diâmetro da base
f	Folga do dente
h	Altura do dente
S	Espessura de dente
E	Espessura da engrenagem
p	Passo circular
m	Módulo
α	Ângulo de pressão

Fonte: (Pezzano e Klein, 1957).

O ângulo de pressão (α) de um par de engrenagens é definido como o ângulo entre a linha de ação e a direção da velocidade no ponto primitivo. Os ângulos de pressão são padronizados pelos fabricantes de engrenagens e pelas normas vigentes. Os valores normalmente utilizados são os de 14,5°, 20° e 25°, sendo 20° o mais utilizado e 14,5° praticamente obsoleto atualmente. As engrenagens podem ser fabricadas em qualquer ângulo de pressão, porém, os custos de fabricação são mais altos quando comparados com aquelas fabricadas com ângulos padronizados, pois teriam de ser feitas ferramentas especiais para usinagem. Para que as engrenagens possam trabalhar juntas, elas devem ser usinadas com o mesmo módulo e ângulo nominal de pressão (NORTON, 2013, p. 685).

4.3 XML

O XML (*eXtensible Markup Language*) é uma linguagem de marcação que armazena todo o tipo de dado, utilizando caracteres Unicode (esquema de codificação de caracteres com 16 bits), que armazena qualquer tipo de caractere ou símbolo. Essa disposição de registrar vários tipos de dados é chamada de “Selfdescribe data” ou dados auto descritivos, que faz a representação de informações em qualquer idioma (LIGHT, 1999).

O XML possui uma forma de texto simples, bastante flexível que vem derivado do SGML (*Standard General Markup Language*) ISO 8879, que possui um padrão de formatação de dados, onde todos esses dados fiquem organizados.

Todas as informações armazenadas no XML estão dentro de tags, que são marcações de início e fim onde estão os elementos, as tags são demarcadas pelos símbolos < e > e no final por uma barra \.

Através da linguagem XML podemos criar outras linguagens para especificar cada um dos usos de XML, são as sublinguagens. Sublinguagens são linguagens que servem para determinar outras linguagens, ou seja, são metalinguagens (DIAS, 2006).

Existem duas metalinguagens adquiridas a partir do XML, o DTD (*Definition Type Document*) e o DTS (*Definition Type Scheme*) (FARIA, 2005).

O DTS e o DTD são regras do XML que auxiliam a validar os dados quando a aplicação que os recebe não há internamente uma descrição da informação que está recebendo.

O DTS (*Definition Type Scheme*) estabelece um conjunto de regras que definem sua composição do documento a partir de uma gramática formal. Na sua estrutura detalha quais elementos podem ser usados e onde eles podem ser empregados. O DTD (*Definition Type Document*) indica todas as partes, obrigatórias e opcionais, que o documento XML deve seguir ao ser criado (CRIAR, 2006).

4.3.1 Características do XML

O XML tem como principal característica o conteúdo separado, que consiste em concentrar na estrutura da informação e não na aparência, a partir disso criou as seguintes características: extensível, interpolar, universal, simples, transportável, flexível e portabil (PITTSMOULTIS, 2000).

- **Extensível:** Utilizado em todos os campos de conhecimento. O programador tem total liberdade para criar linguagens de tags que melhor lhe agradar. Existe uma grande escolha na estrutura dos dados, facilitando a troca dos mesmos.
- **Interpolaridade:** Os dados são vistos como documentos XML e não como arquivos em formatos exclusivos. O próprio editor de texto pode tratar de todos os dados de uma determinada estrutura. A interpolaridade assegura que o documento será formal e preciso no ponto de vista dos dados até a maneira como salvá-los. Com o XML é possível padronizar os dados, de forma que eles possam ser recuperados das mais diferentes fontes sem que haja grandes problemas.
- **Universal:** Todos os dados são iguais, tanto para o envio, recebimento e processamento.
- **Simples:** Os dados são escritos de maneira que possa ler e editar facilmente.
- **Transportável:** É possível implantar, programar e aplicar aos diferentes sistemas.
- **Flexível:** É possível representar qualquer tipo de informação.
- **Portabilidade:** Pode ser utilizado em qualquer linguagem de programação e qualquer sistema operacional.

Com essas características obtém-se uma linguagem que:

- Separa conteúdo de formatação;
- É de fácil leitura tanto para humanos como para máquinas;
- Tem ampla liberdade para criação de tags;
- Pode fazer comunicação com diferentes bancos de dados;

4.3.2 Regras do XML

A principal regra do XML é a padronização no seu formato. Outra regra está ligada caso ocorra erro em sua programação, o XML interrompe o processo e exibe a mensagem de erro. Esse modo de tratar erros é conhecido como tratamentos de erros draconianos.

As regras podem variar de acordo com sua classificação: Regras específicas DTS e Sem DTS. Quando o documento contiver validação o processador fará a verificação de acordo com as regras específicas DTS, caso o documento não tenha a validação, o processador fará a verificação sem o DTS, apenas com as regras do XML (DIAS, 2006).

Algumas regras do XML:

- Um documento XML deve conter um ou mais elementos;
- O nome de um elemento deve ser igual na marcação (tags) inicial e final;
- O XML é case-sensitive, há diferenças entre letras maiúsculas e minúsculas;
- O único elemento que não pode fazer parte do conteúdo de outro elemento é o elemento raiz;
- Os elementos que estão dentro da tag inicial e final deve estar alinhados;
- Os nomes dos elementos podem conter letras, dígitos, hífen ou underscores;
- Os nomes dos elementos que começam com xml, XML ou outra combinação dessa string são reservados por padrão;
- Os caracteres permitidos são os mesmo dos nomes dos elementos;
- O valor dado ao atributo deve estar entre apostrofes ‘...’ ou aspas duplas “...”;
- Caracteres “<” e “&” não podem ser usados no texto como são usados nas marcações. Se esses caracteres são necessários utiliza-se “<”, ao invés de “<” e “&” ao invés de “&”;
- É possível botar comentário em qualquer lugar do documento fora de uma marcação;

4.3.3 Benefícios do XML

O objetivo de usar XML é trazer flexibilidade. Dentre os benefícios para desenvolvedores e usuários temos: buscas mais eficientes, integração de dados de fontes diferentes, computação e manipulação local, múltiplas formas de visualizar os dados, atualizações granulares dos documentos, escalabilidade, fácil compreensão (DIAS, 2006).

4.4 Linguagem de programação Java

A linguagem Java possui ambientes distribuídos complexos como a rede Internet. Mas permite seu programador ir além disso, proporcionando uma linguagem de programação de uso geral, possuindo recursos suficientes para a construção de uma variedade de aplicativos que podem ou não depender do uso de conexão (WUTKA,1997).

A decisão na utilização de Java para esta aplicação é principalmente pela neutralidade dessa linguagem em relação à plataforma, pois o uso de outras linguagens, como C/C++ ou Pascal, demandam a recompilação do programa para um formato no qual o computador do usuário entenda o código fonte toda vez que se necessita migrar de plataforma. No caso de Java, o mesmo código já compilado pode executar em mais de uma plataforma sem a obrigação de alterações do código fonte e recompilação (OLIVEIRA, 2018).

Além da linguagem Java ser simples, de fácil aprendizado ou migração, pois possui um reduzido número de construções. A redução das construções mais aptas a erros de programação, tais como ponteiros e gerenciamento de memória via código de programação também faz com que a programação em Java seja mais eficiente. Contém um conjunto de bibliotecas que fornecem grande parte da funcionalidade básica da linguagem, incluindo rotinas de acesso à rede e criação de interface gráfica (INDRUSIAK, 1996).

De acordo com Silva (2007) a linguagem Java tem como características:

- O paradigma de orientação a objetos,
- Ser ao mesmo tempo compilada e interpretada. Os programas fonte em Java (extensão .java) são compilados para um formato binário de código conhecido como *bytecode* (extensão .class) que é independente de plataforma. O *bytecode* será então executado em outra máquina de forma interpretada,
- Ser uma linguagem neutra em relação à arquitetura, o que garante portabilidade entre plataformas,
- Ser multitarefa. Java suporta o conceito de threads permitindo a execução simultânea de diversos segmentos de código,

- Ser dinâmica. A ligação (*linking*) do programa com as bibliotecas de terceiros são executadas dinamicamente, ao contrário de outras linguagens que precisam ser ligadas estaticamente,
- Ser segura. Devido à característica distribuída de Java, ela foi pensada para incorporar diversos recursos de segurança, raramente encontrada em outras linguagens. Podem se especificar quais permissões de acesso o programa em Java terá em relação à rede ou disco rígido,
- Ser simples quando comparada a C++. A linguagem Java apesar de ser derivada da linguagem C++, evitou características problemáticas dessa linguagem que causavam confusão e falta de legibilidade como, por exemplo, herança múltipla,
- Ser compacta. A maioria das máquinas virtuais e programas em Java ocupam pouco espaço de memória. Isso permite que Java possa funcionar em máquinas com poucos recursos como celulares e palmtops,
- Ter alto desempenho. O emprego de compiladores *just in time* (JIT), permite a compilação de partes críticas do código (*hotspots*) e alcançar velocidades de execução próximas a linguagens compiladas como C.

4.5 ANDROID STUDIO

O Android Studio é o IDE (*Integrated Development Environment*) oficial do Android que foi desenvolvido especialmente para o sistema Android, possui um desenvolvimento acelerado e auxilia na criação de aplicativos de alta qualidade para todos os dispositivos Android.

Apresenta ferramentas personalizadas para desenvolvedores do Android, contendo ferramentas avançadas para edição, depuração, testes e geração de perfis de código, como pode-se observar na Figura 4.8.

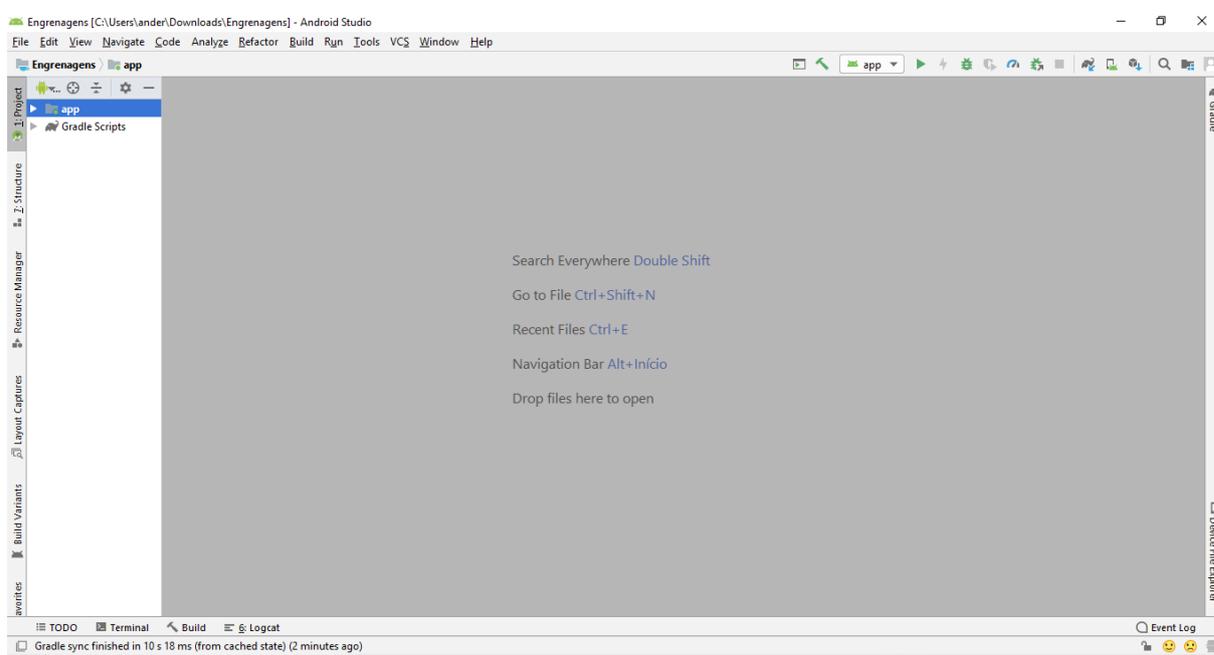


Figura 4.8 - Imagem do Android Studio
Fonte: Autores (2019)

4.5.1 Outros IDEs para desenvolvimento de aplicativos Android

O IntelliJ IDEA é um IDE JVM (*Java Virtual Machine*) que permite, além de outras coisas, o uso do Android SDK (*Software Development Kits*) para que aplicativos desse SO mobile sejam desenvolvidos.

O IntelliJ IDEA foi somente utilizado como IDE base, as configurações para que o IDE Android Studio seja um IDE exclusivo para desenvolvimento Android, essas configurações foram e são desenvolvidas pelos profissionais do Google.

A seguir listo alguns outros IDE's também utilizados para desenvolvimento Android:

- **NetBeans e Plugin Android;**
- **Xamarin:** este é para desenvolvimento de app multiplataforma, porém utilizando a linguagem C#. IDE ainda mantido pela Microsoft;
- **Delphi;**
- **Basic4Android:** developers que desenvolve aplicativos utilizando o IDE Basic;
- **IntellXDK:** tecnologias Web mobile, utilizando o HTML5;
- **Corona Labs:** para criação de games. Linguagem para utilização Lua;
- **Adobe PhoneGap:** este utilizará também tecnologias Web: CSS, HTML e JavaScript.

O Android Studio tem como principal objetivo acelerar o desenvolvimento de aplicativos para o SO mobile do Google e se tornar o principal IDE, em números de uso, para desenvolvimento de aplicativos deste sistema operacional.

Com este IDE temos todas as ferramentas para o desenvolvimento e testes completos dos projetos de aplicativos. É possível desenvolver aplicativos também para Tablet, TV, relógio, Google auto e Google Glass.

Para um desenvolvedor Java, Kotlin ou C/C++ é possível seguir sem problemas com o Android Studio, pois possui linguagens oficiais de desenvolvimento.

5 METODOLOGIA

Na metodologia de cálculos para dimensionamento das ECDRs e para suas conseqüentes relações de transmissão será adotado como padrão o ângulo de pressão (α) igual a 20° como valor usual recomendado nas diversas literaturas de referência da área. Foram adotados três possíveis dados de entrada para cálculo da engrenagem, nos quais esses são: o diâmetro primitivo, o módulo ou o número de dentes da engrenagem. Assim serão usados dois desses dados de entrada para cálculo dos demais dados do dimensionamento da engrenagem.

Segundo Shigley (2005), o círculo primitivo ou de passo é um círculo teórico sobre o qual todos se baseiam para o cálculo; seu diâmetro é o diâmetro primitivo. Esses círculos são tangentes entre si, onde o pinhão é chamado a menor das engrenagens e a coroa é a de maior diâmetro.

Portanto, o diâmetro primitivo (D_p) é igual o diâmetro intermediário entre o diâmetro externo (D_e) e o diâmetro interno (D_i). Seu cálculo é:

$$D_p = D_e - 2M \quad (5.1)$$

Onde, M = Módulo.

O módulo é a razão entre o diâmetro primitivo e o número de dentes (Z). A unidade utilizada de comprimento é o milímetro. Esse número é que caracteriza a engrenagem e se constitui em sua unidade de medida. O módulo é o número que serve de base para calcular a dimensão dos dentes. Seu cálculo é:

$$M = D_p / Z \quad (5.2)$$

Com isso o cálculo do número de dentes (Z) é:

$$Z = M * D_p \quad (5.3)$$

O passo circular (p) é a medida que corresponde à distância entre dois dentes consecutivos, medida à altura do D_p . Seu cálculo é:

$$p = \pi * M \quad (5.4)$$

Onde $\pi = 3,14$.

Para o cálculo do passe de base (P_b):

$$P_b = \pi * M * \cos(20^\circ) \quad (5.5)$$

O diâmetro externo (D_e) é o diâmetro máximo da engrenagem. Seu cálculo é:

$$D_e = M * (Z + 2) \quad (5.6)$$

Já o diâmetro interno (D_i) é o menor diâmetro da engrenagem, também chamado de diâmetro de pé de dente ou diâmetro de dedendo. Seu cálculo é:

$$D_i = D_p - (2,166 * M) \quad (5.7)$$

O diâmetro de base é calculado por meio do D_p e o cosseno do ângulo de pressão, adotado como 20° . Assim:

$$D_b = D_p * \cos(20^\circ) \quad (5.8)$$

A altura total do dente (h) é dada por:

$$h = (1,167 * M) + M \quad (5.9)$$

A espessura do dente (S) é pode ser calculada a partir do passo ou também através do módulo. Seu cálculo é:

$$S = p * 2 \text{ ou} \quad (5.10)$$

$$S = (\pi/2) * M \quad (5.11)$$

Para o cálculo da espessura da engrenagem (E) é igual:

$$E = 8 * M \quad (5.12)$$

O cálculo para folga do dente é igual:

$$f = 0,167 * M \quad (5.13)$$

Para a primeira engrenagem pede-se no aplicativo a entrada de dados de quatro possíveis variáveis, que são: número de dentes, diâmetro primitivo, rotação e torque. Para entrada de dados para segunda engrenagem pede-se duas das quatro possíveis em cada espaço. No resultado será informando o tipo de relação entre as engrenagens (pinhão e coroa). Além da relação de transmissão (i), do módulo (M), do número de dentes (Z) e rotação (ω).

Para o cálculo da relação de transmissão (i) de engrenagens em geral é dada pela Equação 5.14.

$$i = \frac{D_{p2}}{D_{p1}} = \frac{Z_2}{Z_1} = \frac{\omega_1}{\omega_2} = \frac{T_2}{T_1} \quad (5.14)$$

Onde:

D_{p1} = diâmetro primitivo da roda motora.

D_{p2} = diâmetro primitivo da roda movida.

Z_1 = número de dentes da roda motora.

Z_2 = número de dentes da roda movida.

ω_1 = rotação da engrenagem 1.

ω_2 = rotação da engrenagem 2.

T_1 = torque do pinhão 1.

T_2 = torque da coroa 2.

Para a primeira engrenagem tem-se os dados oferecido pelo usuário. Para o cálculo do Torque da segunda engrenagem (T_2), caso não oferecido:

$$T_2 = \frac{\omega_1 * T_1}{\omega_2}, (N*mm) \quad (5.15)$$

O cálculo da rotação (ω_2):

$$\omega_2 = \frac{\omega_1 * T_1}{T_2}, \text{ rad/s} \quad (5.16)$$

5.1 Desenvolvimento do aplicativo para dimensionamento das ECDRs

Apresentando a metodologia aplicada, por meio do programa Android Studio, desenvolvido para Android, foi desenvolvido o aplicativo para dimensionamento de engrenagens de dentes retos, que decidiu nomeá-lo de *Engrenagens*. Optou-se pelo uso desse programa pois além do editor de código e das ferramentas de desenvolvedor avançados, o Android Studio proporciona ainda mais soluções para somar sua produtividade na criação de aplicativos Android.

Para o desenvolvimento do projeto, dentro do programa Android Studio oferece alguns tipos de pré-projetos e templates para iniciar o projeto. Foi escolhido o template Navigation Drawer Activity (figura 5.1), pois tem fácil manuseio e se adequa ao projeto do aplicativo.

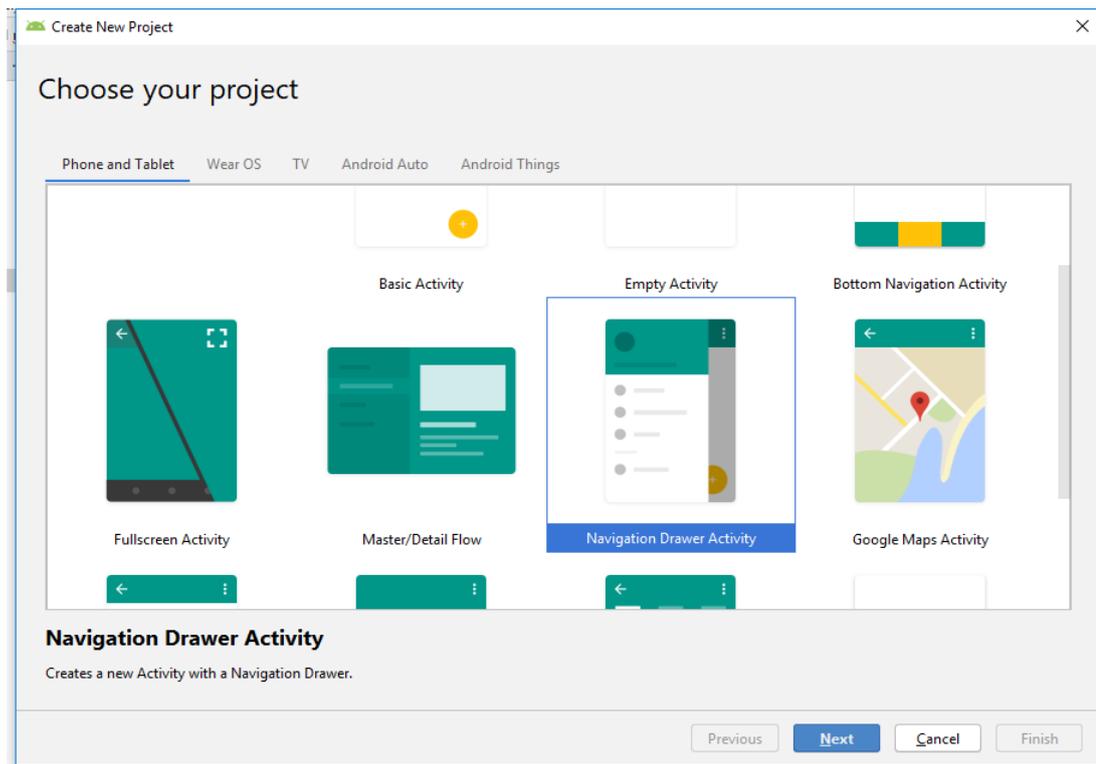


Figura 5.1 - Template utilizado no projeto.
Fonte: Autores (2019).

Para o desenvolvimento do aplicativo, inicialmente foi implementada a sua interface gráfica através do uso da linguagem XML. Primeiramente criou-se de interface para engrenagens, onde através do *LinearLayout* podemos organizar a orientação de todas as *views*, podendo ser verticais ou horizontais. Os *TextView* são os elementos que mostram informações de texto para o usuário, onde esse usuário não consegue editar. O *EditText* é um elemento de interface onde o usuário consegue informar valores de entrada via teclado. Também foram utilizados dois *Spinners*, onde possui a opção de clicar e abrir uma lista de possibilidades de confirmações de variáveis, onde podem ser selecionados o módulo, o número de dentes ou o diâmetro primitivo.

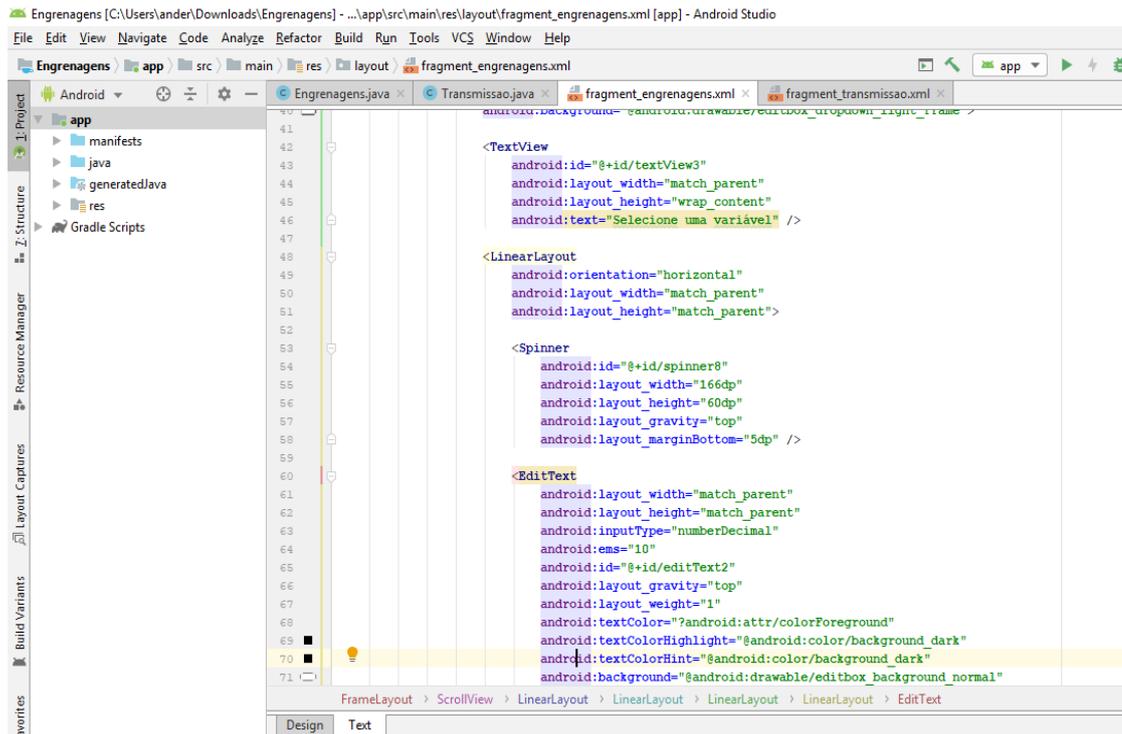


Figura 5.2 - Exemplo de utilização de TextView e EditText.
Fonte: Autores (2019)

Logo após a criação dos *TextViews* e *EditTexts* utilizados para entrada de variáveis do aplicativo, foi criado as opções *Button* para o dimensionamento da engrenagem *Dimensionar* e para limpar todos os dados de entrada *Limpar*.



Figura 5.3- Implementação de Buttons no aplicativo.
Fonte: Autores (2019)

Depois criou os *Resultados*, opção que irá mostrar na interface do aplicativo através da criação de *TextViews* os resultados do dimensionamento da engrenagem cilíndrica de dentes retos. Esses resultados são a partir das operações realizadas no código Java.

Para um segundo XML, foi criado a interface de Transmissões, assim como a interface de Engrenagens foram criados *LinearLayouts* para o desenvolvimento. Criou os *TextViews* e “*EditTexts*” para entrada da primeira engrenagem, no qual possuía as variáveis de número de dentes, diâmetro primitivo, rotação e torque.

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="31dp">

    <TextView
        android:text="N. de dentes:"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/textView7"
        android:layout_weight="1"
        android:gravity="center_vertical"
        android:layout_marginLeft="5dp"
        android:textColor="@android:color/background_dark" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/editText3"
        android:gravity="center_vertical"
        android:layout_weight="1"
        android:background="@android:drawable/editbox_background_normal" />

    <TextView
        android:text="D. primitivo:"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/textView5"
        android:layout_weight="1" />
```

Figura 5.4 - Exemplo de utilização de *TextView* e *EditView* para transmissões.
Fonte: Autores (2019)

Para a entrada de dados da segunda engrenagem foi criado novamente de outros dois *Spinner*, em que o primeiro tem as opções de entrada número de dentes ou diâmetro primitivo e na segunda, as opções de entrada rotação ou torque.

Em seguida foi criado o *Button* calcular, para o cálculo da transmissão e o *Button* limpar, para a opção de limpeza dos dados de entrada.

```

210
211 <LinearLayout
212     android:orientation="horizontal"
213     android:layout_width="match_parent"
214     android:layout_height="38dp"
215     android:layout_marginTop="10dp">
216
217     <Button
218         android:text="Calcular"
219         android:layout_width="match_parent"
220         android:layout_height="wrap_content"
221         android:id="@+id/button4"
222         android:layout_weight="1" />
223
224     <Button
225         android:text="Limpar"
226         android:layout_width="match_parent"
227         android:layout_height="wrap_content"
228         android:id="@+id/button"
229         android:layout_weight="1" />
230 </LinearLayout>
231

```

Figura 5.5 - Implementação de Buttons para transmissões.

Fonte: Autores (2019)

Logo após foi criado os *Resultados*, no qual esse mostra a partir dos *TextViews* para a primeira engrenagem o tipo de transmissão (Pinhão ou Coroa) e para a segunda também o tipo de transmissão (Pinhão ou Coroa) e os resultados obtidos a partir de cálculos no Java.

Depois da criação das interfaces no XML, desenvolveu a implementação das equações em Java a partir de Java Class, onde define o comportamento e atributo da aplicação. Para o desenvolvimento do projeto, dividiu em duas classes, engrenagens e transmissões.

A classe de engrenagens inicializou as variáveis que foram criadas no XML, através do código:

```

public class Engrenagens extends Fragment {

    private String[] variaveis = new String[]{"Selezione", "m (módulo)", "Dp (diâm. Primitivo)", "Z (N. de Dentes)"};

    private Spinner sp, sp2;

    private TextView texto1, texto2, texto3, texto4, texto5, texto6, texto7, texto8, texto9, texto10, texto11, texto12, texto13, texto14;
    private EditText valor1, valor2, valor3;

    private Button btnDimensionar, btnLimpar;

    public Engrenagens() {

    }

}

```

Onde as variáveis *private* são variáveis criadas para desenvolvimento do projeto.

Logo depois foi criado o método *onCreatView* que é fundamental para a inicialização dos componentes da atividade.

```

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                        Bundle savedInstanceState) {
    View v = inflater.inflate(R.layout.fragment_engrenagens, container,
false);

    ArrayAdapter<String> adapter = new
ArrayAdapter<String>(this.getActivity(),
android.R.layout.simple_spinner_dropdown_item, variaveis);

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_it
em);

    sp = (Spinner) v.findViewById(R.id.spinner8);
    sp.setAdapter(adapter);

    texto1 = (TextView) v.findViewById(R.id.textView6);
    texto2 = (TextView) v.findViewById(R.id.textView9);
    texto3 = (TextView) v.findViewById(R.id.textView11);
    texto4 = (TextView) v.findViewById(R.id.textView12);
    texto5 = (TextView) v.findViewById(R.id.textView13);
    texto6 = (TextView) v.findViewById(R.id.textView14);
    texto7 = (TextView) v.findViewById(R.id.textView15);
    texto8 = (TextView) v.findViewById(R.id.textView16);
    texto9 = (TextView) v.findViewById(R.id.textView17);
    texto10 = (TextView) v.findViewById(R.id.textView18);
    texto11 = (TextView) v.findViewById(R.id.textView20);
    texto12 = (TextView) v.findViewById(R.id.textView27);
    texto13 = (TextView) v.findViewById(R.id.textView26);
    texto14 = (TextView) v.findViewById(R.id.textView25);

    valor1 = (EditText) v.findViewById(R.id.editText2);
    valor2 = (EditText) v.findViewById(R.id.editText5);
    valor3 = (EditText) v.findViewById(R.id.editText6);

    btnDimensionar = (Button) v.findViewById(R.id.button2);
    btnLimpar = (Button) v.findViewById(R.id.button3);

```

Para implementação do *Spinner*, responsável pela seleção da opção para entrada de variáveis, usou-se o método de *setOnItemSelectedListener*, no qual exige dois métodos o *onItemSelected()* e *onNoItemSelected()*.

```

sp.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int
position, long id) {
        if(position == 0){
            texto1.setText("Unidade");
        }
        else if(position == 1){
            texto1.setText("mm");
        }
        else if(position == 2){

```

```

        texto1.setText("mm");
    }
    else if(position == 3){
        texto1.setText("dentes");
    }
}
@Override
public void onNothingSelected(AdapterView<?> parent) {
}
});

```

O mesmo ocorrendo para o *Spinner* da segunda variável de entrada do dimensionamento da engrenagem.

Para o *Button* dimensionar e limpar, adotou o método de *onClickListener*, que consiste no método de click no aplicativo.

```

btnDimensionar.setOnClickListener(new Button.OnClickListener() {
    @RequiresApi(api = VERSION_CODES.N)
    @Override
    public void onClick(View v) {
        DimensionarEngrenagens(v);
    }
});

btnLimpar.setOnClickListener(new Button.OnClickListener() {
    @RequiresApi(api = VERSION_CODES.N)
    @Override
    public void onClick(View v) {
        Limpar();
    }
});

```

Durante a utilização do aplicativo caso o usuário click para realizar dimensionamento e não informe algum valor dentre os solicitados, aparece de imediato na tela a mensagem: “Atenção”; “Preencha os campos em branco”.

```

@RequiresApi(api = VERSION_CODES.N)
public void DimensionarEngrenagens(View view) {
    if (valor1.getText().toString().isEmpty() ||
    valor2.getText().toString().isEmpty() ||
    valor3.getText().toString().isEmpty()) {
        new AlertDialog.Builder(this.getActivity())
            .setTitle("Atenção")
            .setMessage("Preencha os campos em branco!")
            .setPositiveButton("Ok!", new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int which)

```

```

        }
        }).show();
    }
    else{
        int posicao1 = sp.getSelectedItemPosition();
        int posicao2 = sp2.getSelectedItemPosition();

        double Z = 0, m = 0, dp = 0;
        double a = Double.parseDouble(String.valueOf(valor3.getText()));
    }
}

```

Caso o usuário não selecione uma das opções de variáveis e informar que os valores são zeros (0), o aplicativo mostrará uma mensagem: “Atenção”; “Selecione uma opção”. Se não ocorrer isso, o aplicativo desenvolverá os cálculos a partir das variáveis de entrada do usuário.

```

double Z = 0, m = 0, dp = 0;
double a = Double.parseDouble(String.valueOf(valor3.getText()));

if (posicao1 == 0 || posicao2 == 0) {
    new AlertDialog.Builder(this.getActivity())
        .setTitle("Atenção")
        .setMessage("Selecione uma opção!")
        .setPositiveButton("Ok!", new DialogInterface.OnClickListener()
        {
            public void onClick(DialogInterface dialog, int which) {
            }
        })
        .show();
} else {

    if (posicao1 == 1 && posicao2 == 2) {
        m = Double.parseDouble(String.valueOf(valor1.getText()));
        dp = Double.parseDouble(String.valueOf(valor2.getText()));
        Z = dp / m;

    } else if (posicao1 == 1 && posicao2 == 3) {
        m = Double.parseDouble(String.valueOf(valor1.getText()));
        Z = Double.parseDouble(String.valueOf(valor2.getText()));
        dp = Z*m;

    } else if (posicao1 == 2 && posicao2 == 1) {
        dp = Double.parseDouble(String.valueOf(valor1.getText()));
        m = Double.parseDouble(String.valueOf(valor2.getText()));
        Z = dp / m;
    }
}

```

```

} else if (posicao1 == 2 && posicao2 == 3) {
    dp = Double.parseDouble(String.valueOf(valor1.getText()));
    Z = Double.parseDouble(String.valueOf(valor2.getText()));
    m = dp / Z;

} else if (posicao1 == 3 && posicao2 == 1) {
    Z = Double.parseDouble(String.valueOf(valor1.getText()));
    m = Double.parseDouble(String.valueOf(valor2.getText()));
    dp = Z*m;

} else if (posicao1 == 3 && posicao2 == 2) {
    Z = Double.parseDouble(String.valueOf(valor1.getText()));
    dp = Double.parseDouble(String.valueOf(valor2.getText()));
    m = dp / Z;
}

```

Para o cálculo de dimensionamento de engrenagem cilíndrica de dentes retos, foi implementado as equações de acordo com a literatura informada, no desenvolvedor Java. Para os resultados com três casas decimais após a virgula.

```
double de, di, db, h, p, pb, f, E, S;
```

```

de = dp+2*m;
di = dp-2*1.167*m;
db = dp*Math.cos(a*3.141);
h = m+m*1.167;
p = 3.14*m;
pb = 3.14*m*Math.cos(a);
S = p/2;
E = 8*m;
f = 0.167 * m;

```

```
DecimalFormat df = new DecimalFormat("###");
```

```

texto3.setText("N. de dentes \nZ: "+df.format(Z));
texto4.setText("D. primitivo \nDp: "+df.format(dp)+" [mm]");
texto5.setText("Módulo \nm: "+df.format(m)+" [mm]");
texto6.setText("D. externo \nDe: "+df.format(de)+" [mm]");
texto7.setText("D. interno \nDi: "+df.format(di)+" [mm]");
texto8.setText("D. de base \nDb: "+df.format(db)+" [mm]");
texto9.setText("Alt. do dente \nh: "+df.format(h)+" [mm]");
texto10.setText("Passo \np: "+df.format(p)+" [mm]");
texto11.setText("P. de base \nPb: "+df.format(pb)+" [mm]");
texto12.setText("Esp. do dente \nS: "+df.format(S)+" [mm]");
texto13.setText("Esp. da engrenagem \nE: "+df.format(E)+" [mm]");
texto14.setText("Folga do dente \nf: "+df.format(f)+" [mm]");

```

Assim como na classe de engrenagens do Java, a classe de transmissões, os campos para variáveis que foram criadas no XML precisam ser inicializados.

```

public class Transmissao extends Fragment {

    private String[] variaveis1 = new String[]{"Selecione", "Dp (diâm.
Primitivo)", "Z (N. de Dentes)"};
    private String[] variaveis2 = new String[]{"Selecione", "ω
(Rotação)", "T (Torque)"};
    private Spinner sp, sp2;

    private TextView texto1, texto2, texto3, texto4, texto5, texto6, texto7,
texto, texto0;
    private EditText valor1, valor2, valor3, valor4, valor5, valor6;

    private Button btnDimensionar, btnLimpar;

    public Transmissao() {
    }
}

```

Assim como no desenvolvimento para engrenagens, foi usado os mesmos métodos *onCreatView*, *onItemSelectedListener*, *onClickListener* para criação da classe de Transmissão.

Na classe Transmissão foram desenvolvidas as equações para os cálculos já apresentados. Caso o usuário não selecione uma das opções de variáveis, e informar que os valores são zeros (0), o aplicativo mostrará uma mensagem: “Atenção”; “Selecione uma opção”. Se não ocorrer isso, o aplicativo desenvolverá os cálculos a partir das variáveis de entrada do usuário.

```

double Z1 = 0, m = 0, dp1 = 0, dp2 = 0, Z2 = 0, w1 = 0, T1 = 0, w2 = 0, T2
= 0, i = 0, grand = 0;
double a = Double.parseDouble(String.valueOf(valor3.getText()));

Z1 = Double.parseDouble(String.valueOf(valor1.getText()));
dp1 = Double.parseDouble(String.valueOf(valor2.getText()));
w1 = Double.parseDouble(String.valueOf(valor3.getText()));
T1 = Double.parseDouble(String.valueOf(valor4.getText()));

if (posicao1 == 0 || posicao2 == 0) {
    new AlertDialog.Builder(this.getActivity())
        .setTitle("Atenção")
        .setMessage("Selecione uma opção!")
        .setPositiveButton("Ok!", new DialogInterface.OnClickListener()
        {
            public void onClick(DialogInterface dialog, int which) {
            }
        })
        .show();
} else {

    if (posicao1 == 1 && posicao2 == 1) {
        dp2 = Double.parseDouble(String.valueOf(valor5.getText()));
        Z2 = (Z1*dp2) / dp1;

        w2 = Double.parseDouble(String.valueOf(valor6.getText()));
        T2 = (w1*T1)/w2;

    } else if (posicao1 == 1 && posicao2 == 2) {
        dp2 = Double.parseDouble(String.valueOf(valor5.getText()));
        Z2 = (Z1*dp2) / dp1;
    }
}

```

```

    T2 = Double.parseDouble(String.valueOf(valor6.getText()));
    w2 = (w1*T1)/T2;

} else if (posicao1 == 2 && posicao2 == 1) {
    Z2 = Double.parseDouble(String.valueOf(valor5.getText()));
    dp2 = (dp1*Z2)/Z1;

    w2 = Double.parseDouble(String.valueOf(valor6.getText()));
    T2 = (w1*T1)/w2;

} else if (posicao1 == 2 && posicao2 == 2) {
    Z2 = Double.parseDouble(String.valueOf(valor5.getText()));
    dp2 = (dp1*Z2)/Z1;

    T2 = Double.parseDouble(String.valueOf(valor6.getText()));
    w2 = (w1*T1)/T2;

}

```

Para o resultado,

```

DecimalFormat df = new DecimalFormat("###");

if(dp1<=dp2) {
    texto1.setText("Tipo: Pinhão");
    texto2.setText("Coroa");
}
else {
    texto1.setText("Tipo: Coroa");
    texto2.setText("Pinhão");
}

m = dp2/Z2;
texto3.setText(df.format(m));

if(posicao1 == 1){
    texto4.setText("N. de dentes: "+df.format(Z2));}
else{
    texto4.setText("D. primitivo: "+df.format(dp2)+" [mm]");
}

if(posicao2 == 2){
    texto5.setText("Rotação (ω): "+df.format(w2)+" [rad/s]");}
else{
    texto5.setText("Torque (T): "+df.format(T2)+" [Nmm]");
}

i = w1/w2;
texto6.setText("Relação (i): "+df.format(i));

if(i>1){
    texto7.setText("Classificação: i>1 - redutor de velocidades");
} else if(i == 1){
    texto7.setText("Classificação: i=1 - sem variação de velocidades");
}
else{
    texto7.setText("Classificação: i<1 - ampliador de velocidades");
}

```

Assim foi implementado as equações no código Java e desenvolvido o aplicativo através dos métodos utilizados no Android Studio.

6 RESULTADOS

O aplicativo *Engrenagens* foi submetido a simulações com propósito de validação do mesmo.

Primeiro adotou-se dados para o dimensionamento de engrenagens, com possíveis modos de entrada como módulo, número de dentes e diâmetro primitivo, assim utilizou-se os três para teste.

Os dados do exemplo são:

- Módulo (m): 2 mm
- Número de dentes (Z): 20 dentes
- Diâmetro primitivo (Dp): 40,0 mm

As primeiras variáveis foi a escolha do módulo e do número de dentes (figura 6.1)

The screenshot shows the 'Engrenagens' application interface. At the top, there is a blue header with a hamburger menu icon, the title 'Engrenagens', and a vertical ellipsis icon. Below the header, there are two tabs: 'ENGRENAGENS' (selected) and 'TRANSMISSÃO'. The main content area is titled 'Engrenagem cilíndrica de dentes retos'. It contains three input fields: 'Selecione uma variável' with a dropdown menu set to 'm (módulo)' and a text input field containing '2' followed by 'mm'; another 'Selecione uma variável' with a dropdown menu set to 'Z (N. de De..)' and a text input field containing '20' followed by 'dentes'; and a third input field for 'Ângulo de Pressão:' containing '20'. Below these fields are two buttons: 'DIMENSIONAR' and 'LIMPAR'. At the bottom, there is a green header for 'Resultados' followed by a table of calculated values.

Resultados		
N. de dentes Z: 20,0	D. primitivo Dp: 40,0 [mm]	Módulo m: 2,0 [mm]
D. externo De: 44,0 [mm]	D. interno Di: 35,332 [mm]	D. de base Db: 39,997 [mm]
Alt. do dente h: 4,334 [mm]	Passo p: 6,28 [mm]	P. de base Pb: 2,563 [mm]
Esp. do dente S: 3,14 [mm]	Esp. da engrenagem E: 16,0 [mm]	Folga do dente f: ,334 [mm]

Figura 6.1 - Exemplo do dimensionamento da engrenagem com módulo e n° de dentes.

Fonte: Autores (2019)

Os resultados obtidos no exemplo são:

- Número de dentes (Z): 20 dentes
- D. primitivo (Dp): 40,0 mm
- Módulo (m): 2,0 mm
- D. externo (De): 44,0 mm
- D. interno (Di): 35,332 mm
- D. base (Db): 39,997 mm
- Altura do dente (h): 4,334 mm
- Passo (p): 6,28 mm
- P. da base (Pb): 2,563 mm
- Esp. do dente (S): 3,14 mm
- Esp. da engrenagem (E): 16,0 mm
- Folga do dente (f): 0,334 mm

Utilizando agora o módulo (m) e o diâmetro primitivo (Dp), figura 6.2, observou-se que tem os mesmos resultados encontrado anteriormente.

Resultados		
N. de dentes Z: 20,0	D. primitivo Dp: 40,0 [mm]	Módulo m: 2,0 [mm]
D. externo De: 44,0 [mm]	D. interno Di: 35,332 [mm]	D. de base Db: 39,997 [mm]
Alt. do dente h: 4,334 [mm]	Passo p: 6,28 [mm]	P. de base Pb: 2,563 [mm]
Esp. do dente S: 3,14 [mm]	Esp. da engrenagem E: 16,0 [mm]	Folga do dente f: ,334 [mm]

Figura 6.2 - Dimensionamento da engrenagem com módulo e diâmetro primitivo
Fonte: Autores (2019)

O mesmo acontece quando entramos com os dados de número de dentes e diâmetro primitivo, figura 6.3.

Resultados		
N. de dentes Z: 20,0	D. primitivo Dp: 40,0 [mm]	Módulo m: 2,0 [mm]
D. externo De: 44,0 [mm]	D. interno Di: 35,332 [mm]	D. de base Db: 39,997 [mm]
Alt. do dente h: 4,334 [mm]	Passo p: 6,28 [mm]	P. de base Pb: 2,563 [mm]
Esp. do dente S: 3,14 [mm]	Esp. da engrenagem E: 16,0 [mm]	Folga do dente f: ,334 [mm]

Figura **Erro! Nenhum texto com o estilo especificado foi encontrado no documento..3-** Dimensionamento da engrenagem com número de dentes e diâmetro primitivo.
Fonte: Autores (2019)

Para o cálculo da transmissão entre duas engrenagens, no exemplo proposto informa-se quatro dados da primeira engrenagem, como: número de dentes (Z), o diâmetro primitivo (D_p), rotação (ω) e torque (T).

Na segunda engrenagem para a primeira variável pode-se informar duas opções, o número de dentes ou diâmetro primitivo. A segunda variável também possui duas opções, a rotação ou torque da engrenagem.

Os dados da primeira engrenagem:

- Número de dentes (Z): 20 dentes
- D. primitivo (D_p): 40 mm
- Rotação (ω): 100 rad/s

- Torque (T): 50 N*mm

Para a segunda engrenagem, os dados iniciais serão do diâmetro primitivo e rotação, onde:

- D. primitivo (Dp): 20 mm
- Rotação (ω): 200 rad/s

Os dados obtidos são, conforme a figura 6.4:

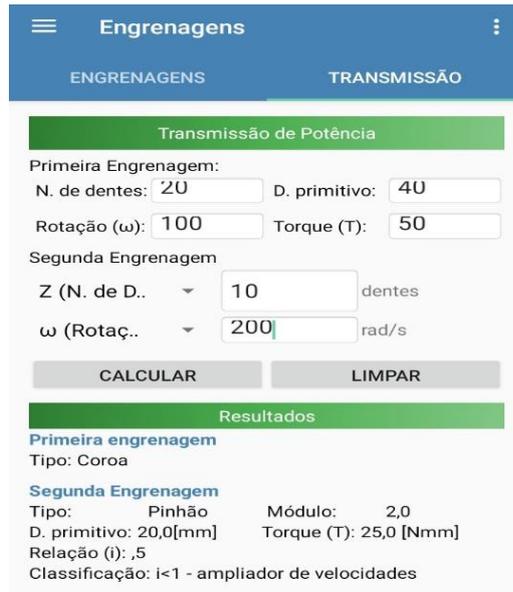
- Módulo (M): 2,0
- Número de dentes: 10 dentes
- Torque (T): 25,0 N*mm
- Relação (i): 0,5
- Tipo: Pinhão

Figura 6.3 - Resultado da transmissão de engrenagens por meio do d. primitivo e rotação.
Fonte: Autores (2019)

Continuando com a primeira variável diâmetro primitivo e mudando apenas a segunda para torque com valor de 25,0 N*mm, podemos observar que ainda continua os mesmos valores dos primeiros cálculos. Como podemos observar na figura 6.5

Figura 6.4 - Resultado da transmissão de engrenagens por meio do d. primitivo e torque.
 Fonte: Autores (2019)

Utilizando agora na primeira variável o número de dentes e na segunda variável a rotação, figura 6.5, observa-se que contém os mesmos valores encontrados no primeiro exemplo. O mesmo acontece quando continuamos com a primeira variável e trocamos a segunda variável para torque, figura 6.6.



Engrenagens

ENGRENAGENS TRANSMISSÃO

Transmissão de Potência

Primeira Engrenagem:
 N. de dentes: D. primitivo:
 Rotação (ω): Torque (T):

Segunda Engrenagem
 Z (N. de D.): dentes
 ω (Rotaç..): rad/s

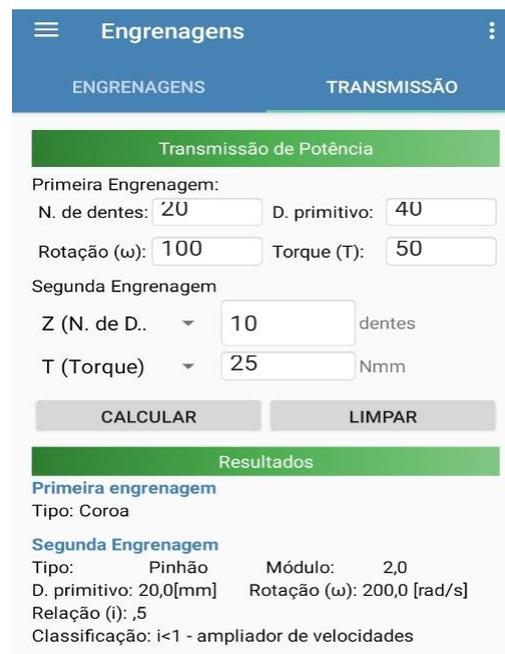
CALCULAR LIMPAR

Resultados

Primeira engrenagem
 Tipo: Coroa

Segunda Engrenagem
 Tipo: Pinhão Módulo: 2,0
 D. primitivo: 20,0[mm] Torque (T): 25,0 [Nmm]
 Relação (i): ,5
 Classificação: $i < 1$ - ampliador de velocidades

Figura 6.5 - Resultado da transmissão de engrenagens por meio do n° de dentes e rotação.
 Fonte: Autores (2019)



Engrenagens

ENGRENAGENS TRANSMISSÃO

Transmissão de Potência

Primeira Engrenagem:
 N. de dentes: D. primitivo:
 Rotação (ω): Torque (T):

Segunda Engrenagem
 Z (N. de D.): dentes
 T (Torque): Nmm

CALCULAR LIMPAR

Resultados

Primeira engrenagem
 Tipo: Coroa

Segunda Engrenagem
 Tipo: Pinhão Módulo: 2,0
 D. primitivo: 20,0[mm] Rotação (ω): 200,0 [rad/s]
 Relação (i): ,5
 Classificação: $i < 1$ - ampliador de velocidades

Figura 6.6 - Resultado da transmissão de engrenagens por meio do n° de dentes e torque.
 Fonte: Autores (2019)

Os resultados obtidos pela simulação mostraram que são compatíveis com as recomendações literárias e também com valores de saídas (para os mesmos valores de entrada) de outros softwares, como por exemplo o Ansys.

7 CONSIDERAÇÕES FINAIS

O presente estudo conseguiu alcançar o seu principal objetivo. O desenvolvimento de um aplicativo intuitivo e de fácil utilização para dimensionamento de engrenagens cilíndricas de dentes retos através de dispositivos móveis portadores do S.O. Android.

Neste trabalho foi desenvolvido um aplicativo com auxílio da IDE Android Studio para o dimensionamento de engrenagens cilíndricas de dentes retos que viabilizará aos usuários a minimização do tempo na elaboração de cálculos de dimensionamento. Onde foi possível a implementação dos conceitos de POO no desenvolvimento da aplicação.

O aplicativo denominado *Engrenagens* pode ter bastante relevância para profissionais da área de engenharia ou mesmo para estudantes do curso de Engenharia Mecânica especificamente na disciplina de Elementos de Máquinas a fim de facilitar a utilização da metodologia de cálculos.

REFERÊNCIAS

ANDRADE, Leonardo. A. **Desenvolvimento de um programa para dimensionar engrenagens de dentes retos**. 2016. 69p. Trabalho de Conclusão de Curso – Pontifícia Universidade Católica, Rio de Janeiro, 2016.

ANDRADE, Thales. **9 Tipos de Engrenagens e Suas Aplicações**. 2018. Disponível em: <https://fluxoconsultoria.poli.ufrj.br/blog/projetos-mecanicos/9-tipos-de-engrenagens>. Acessado em 11 de abr. de 2019.

BETIM, Victor. T. **Análise da relação entre tensões e variações geométricas em engrenagens cilíndricas de dentes retos utilizando método analítico e numérico**. 2015. 154p. Trabalho de conclusão de curso - Universidade Tecnológica Federal do Paraná, Curitiba.

CRIAR WEB. “**Manuais e recursos para desenvolvimento web**”. [INTERNET: <http://www.criarweb.com>. Acessado em 11/04/2019.

DIAS, Tatianne D. F. B. **Estudo da Linguagem do XML e sua integração com Plataforma Java**. 2006. 43 p. Monografia (Bacharelado em Ciência da Computação) – Curso de Ciência da Computação da Faculdade de Jaguariúna, Jaguariúna.

DUDLEY, D.W., **Disposição de engrenagens**. “Manual de engrenagens”. Edit. D. W. Dudley. México: Comp. Edit. Continental S.A., 1973.

ESTATÍSTICA DE CELULARES NO BRASIL. **TELECO: Tecnologia em Telecomunicações**, 2019. Disponível em: <http://www.teleco.com.br/ncel.asp>. Acessado em 11 de abr. de 2019.

FARIA, Rogério Amorim de. **Treinamento Avançado em XML**. 1. ed. São Paulo, Digerati Books, 2005.

FERNANDES, O. C., **Apostila de elementos de máquinas: introdução ao projeto de engrenagens cilíndricas**. Departamento de Engenharia Mecânica, EECS. USP, 1997.

INDRUSIAK, L. S., **Linguagem Java**. Rio Grande do Sul: Universidade Federal do Rio Grande do Sul. Grupo JavaRs, 1996. 15 p.

JUNIOR, Antônio P. **Engrenages**. 2019. Disponível em: <https://www.emaze.com/@AIQTCOOQ>. Acessado em 11 de abr. de 2019.

JUVINAL, R. C., MARSHEK, K. M., **Projeto de Componentes de Máquinas**. 4ª Ed. Rio de Janeiro: Editora LTC Livros Técnicos e Científicos Editora S.A., 2008.

KODA, Fábio., **ESTUDO DA FADIGA DE CONTATO EM ENGRENAGENS CILÍNDRICAS DE DENTES RETOS**, 2009. 185p. Dissertação (Mestrado em Engenharia) - Programa de Pós-graduação em Engenharia Mecânica e de Materiais, Universidade Tecnológica Federal do Paraná, Curitiba.

LIGHT, Richard; MORAES, Neilande. **Iniciando em XML**. 1. ed. São Paulo, Makron Books, 1999.

MAITRA, G. M. Handbook of gear design. 2. ed. São Paulo: **Blucher**, 2013.

NIEMAN, G., **Elementos de Máquinas**. 1. ed. São Paulo: Edgard Blucher, Ltda. 1971.

NORTON, Robert L. **Projeto de Máquinas: Uma abordagem integrada**. 4. ed. Porto Alegre: Bookman, 2013.

OLIVEIRA, Carlos R.S., **PROJETO E DESENVOLVIMENTO DE UMA PLATAFORMA EM LINGUAGEM JAVA PARA SIMULAÇÃO DE VOO DE FOGUETES**, 2018. 112 p. Dissertação (Mestrado Profissional em Sistemas Aeroespaciais) - Programa de Pós-Graduação em Engenharia de Computação e Sistemas, Universidade Estadual do Maranhão, São Luís.

PEZZANO, P.; KLEIN; A. **Engranajes y poleas**. 4º ed., Buenos Aires: El Ateneo, 287 p., 1957.

PITTS-MOULIS, Natanya.; KIRL, Cheryl. **XML Black Book**. 1° ed. São Paulo: Books, 2000.

POPULAÇÃO DO BRASIL. **IBGE: Instituto Brasileiro de Geografia e Estatística**,2019. Disponível em: https://www.ibge.gov.br/apps/populacao/projecao/box_popclock.php. Acessado em 11 de abr. de 2019.

Silva, J. F. V. da. **JFloat: Uma biblioteca de ponto flutuante para a linguagem Java com suporte a arredondamento direcionado**. Dissertação de Mestrado. Universidade Federal do Rio Grande do Norte. Natal/RN. 2007.

SHIGLEY, Joseph E.; Mischke, Charles R.; Budynas, Richard G. **Projeto de Engenharia Mecânica**, 7. ed. Porto Alegre: Bookman, 2005.

WUTKA, M. **Java: Técnicas Profissionais**. 1° ed. Editora: Berkeley, 1997.