Caracterização de Métricas de Usabilidade para Expertise de Usuários de Ferramentas de Modelagem 3D

Arnaldo Correia de Andrade filho
Instituto Federal Da Bahia
Rua Emídio dos Santos, S/N, Barbalho
Salvador-Ba, Brasil
arnaldofilho@ifba.edu.br

Antonio Carlos Souza
Instituto Federal Da Bahia
Rua Emídio dos Santos, S/N, Barbalho
Salvador-Ba, Brasil
antoniocarlos@ifba.edu.br

RESUMO

O presente trabalho propõe uma caracterização de métricas de usabilidade voltada para avaliar a *expertise* de usuários de *softwares* de modelagem tridimensional. Para tanto, um processo que conta com três etapas foi criado. Inicialmente, foi utilizado o algoritmo *k-means* para agrupar os usuários de acordo com sua *expertise*, medida em termos das métricas propostas. Em seguida, foi feito um treinamento com a base de dados contendo as métricas avaliadas sobre usuários para que os mesmos fossem classificados em diferentes níveis de *expertise*. Três algoritmos de classificação foram submetidos à base de dados: Bayes, árvores de decisão e florestas aleatórias. A última etapa deste trabalho consistiu em um teste comparativo entre os resultados fornecidos pelos três algoritmos classificadores. Desse modo, a partir do método de validação cruzada, *k-fold*, foi possível determinar que o algoritmo de florestas aleatórias foi o mais preciso com 83% de acurácia. Considerando-se os resultados obtidos, concluiu-se que os métodos e variáveis propostos são válidos para inferir a *expertise* de usuários de *softwares* de modelagem 3D.

Palavras Chave

Modelagem 3D, Expertise, Clusterização , Classificação

1. INTRODUÇÃO

Modelagem tridimensional é um processo presente em indústrias altamente rentáveis como a cinematográfica, a de jogos, a de projetos paisagísticos, a de impressão de objetos em 3D, entre outras. Para possibilitar o aumento da produtividade – ou para que se tenha uma curva de aprendizado menor – no uso das ferramentas de modelagem 3D, diversos estudos como [**bowman**, 1, 14] argumentam que é necessário melhorar a relação entre a interface e o usuário.

Diante do fato exposto, este trabalho visa caracterizar métricas que possam inferir a *expertise* do usuário ao modelar algum objeto em um *software* de modelagem 3D, de forma que pesquisadores possam identificar se alterações no *software* facilitam, ou não, o seu manuseio por parte dos usuários. A ferramenta desenvolvida para realização de tal tarefa permite a análise automática de uma série de dados coletados durante o uso de um *software* de modelagem 3D. Tal medida faz parte do conjunto de processos proposto e defendido por Ivory e Hearst [14], que recomendam que a coleta e interpretação dos dados no processo de avaliação de métricas de usabilidade sejam feitas de forma automatizada.

A automatização de uma determinada tarefa possibilita a análise de um massiva base de dados de forma escalável. Automatizar a inferência de *expertise* de um usuário, seja ele de um software de modelagem 3D ou não, possibilita a avaliação do desempenho individual e grupal na realização de suas respectivas tarefas. Isso permite que haja remanejamentos ou redução de uma equipe de análise de interfaces gráficas, diminuindo assim o custo com os recursos humanos.

Diante dessa perspectiva, este trabalho visa a caracterização de métricas que possam estimar a *expertise* de um usuário ao modelar objetos em um software de modelagem 3D. Contudo é necessária a criação de mecanismos que viabilizem essa estimativa. A partir da captura de dados relacionados às variáveis propostas, este trabalho permite que os usuários sejam divididos em grupos de acordo com a sua *expertise*.

O *software* de modelagem 3D escolhido para validação deste projeto foi o *Blender*. Ele é um *software* de código aberto, gratuito, e que permite sua extensão através da linguagem *Python* [12]. Além disso, todos os objetos modelados neste trabalho são classificados como modelos *low poly*, caracterizados por possuírem poucos polígonos para tornar o objeto mais leve para renderização.

O presente trabalho está estruturado da seguinte forma: a Seção 2 apresenta a fundamentação teórica. Nela são mostrados os conceitos e utilização da aprendizagem de máquina bem como a importação dos métodos de classificação e clusterização de dados. Por conseguinte, o algoritmo k-means é introduzido como método de clusterização. O método k-fold é exposto de modo a se destacar como é importante no processo de validação de classificadores. Em seguida o tema de interfaces 3D e modelagem 3D é abordado, de modo a elucidar os tipos de modelagem 3D. Fazem parte ainda dessa seção, os sistemas especialistas com suas estruturas e vantagens. São apresentados a avaliação de usabilidade, os métodos de avaliação e as métricas para interfaces. A Seção 3 trata dos trabalhos correlatos. Na Seção 4 a caracterização é feita. Nela são mostrados a avaliação da expertise dos usuários e são listados os requisitos funcionais e não funcionais. Em seguida, na Seção 5 os métodos utilizados nesse trabalho são exibidos. A Seção 6 trata dos resultados obtidos. Por

fim a Seção 7 expões as conclusões e trabalhos futuros nessa ordem.

2. FUNDAMENTAÇÃO TEÓRICA

A seguir, são discutidos com maiores detalhes as técnicas e os conceitos bases utilizados para a realização do trabalho proposto.

2.1 Aprendizagem De Máquina

O processo de avaliação de interfaces quando feito de forma manual carece de uma maior escalabilidade. Muitos pesquisadores, como Ivory e Hearst [14], defendem a automatização do processo de avaliação de interfaces melhorando assim sua escalabilidade. Outros pesquisadores como Ortega et al. [1] têm defendido que além da automatização desse processo de avaliação é necessário também aplicar algoritmos de aprendizagem de máquina para avaliar interfaces de usuário.

Segundo James et al. [15], a aprendizagem de máquina é uma área que possui um conjunto de ferramentas usadas para encontrar padrões em uma determinada base de dados, e que permite a previsão de uma resposta que tem como tipo uma variável qualitativa. Essa área pode ser dividida em aprendizagem supervisionada e aprendizagem não supervisionada.

Aprendizagem supervisionada consiste em prever ou estimar uma saída baseada em um ou mais dados de entrada. Um exemplo de aprendizagem supervisionada é a regressão linear. Através do método dos mínimos quadrados, ela faz a busca em uma base de dados e tenta estimar uma equação de primeiro grau para prever um valor no eixo y com base em valores que o usuário forneceu ao eixo x. Já a aprendizagem não supervisionada consiste em entender um determinado conjunto de dados sem precisar de treinamento como na aprendizagem supervisionada [15]. Um exemplo de método não supervisionado é a clusterização, método que consiste em segmentar os dados sem que se tenha nenhum conhecimento anterior da base [15].

A seguir, são apresentadas com mais detalhes as diferentes abordagens utilizadas para aprendizagem de máquina supervisionada e não-supervisionada.

2.1.1 Classificação

Uma das maiores dificuldades na área de aprendizagem de máquina é fazer com que a máquina aprenda novos padrões e conceitos de acordo com uma base de dados. Nesse sentido, o objetivo da área é fazer com que a máquina consiga associar um determinado elemento desconhecido a um conjunto de classes previamente conhecidas. Essa área da aprendizagem de máquina responsável em criar algoritmos para resolver esses problemas é a classificação. Como exemplos de sistemas de classificação temos um *software* de análise de crédito que diz se o correntista está apto a tomar um empréstimo. Nesse caso o programa analisa como entrada os dados históricos que o banco possui sobre o cliente e então o *software* apresenta como saída se o cliente está apto ou não a tomar o empréstimo. Baseado nesse valor de saída o gerente do banco pode ou não emprestar o dinheiro ao correntista do banco.

Classificação é um método que pertence ao grupo da aprendizagem supervisionada, pois para o seu funcionamento correto é necessário que haja uma saída de dados de supervisão. De acordo com Hosmer Jr et al. [7], a diferença entre classificação e regressão consiste na saída, que no caso da classificação é binária e na regressão é numérica. Ou seja, ao invés de ter uma resposta numérica, haverá uma previsão dos grupos ao qual pertencem aqueles dados. Já de acordo com Gan [11], classificação consiste na manipulação ou aprendizado sobre um conjunto de dados nomeados ou rotulados previamente. Com essa classificação prévia, pode-se inferir para um determinado registro individual em qual grupo pertence aquele registro.

Conforme escreve Rish [23]; Garreta e Moncecchi [22], um tipo de classificador especial é o classificador ingênuo de Bayes. Ele usa o teorema de Bayes para calcular as probabilidades de uma variável pertencer a um determinado grupo, ou classe.

Outro tipo de classificador comum para aprendizagem supervisionada é a árvore de decisão. Segundo Nong Ye [28], árvores de decisão permitem classificar um atributo alvo com base em um vetor de atributos. Durante a fase de treinamento do classificador, é montada uma árvore com base nos atributos mais importantes extraídos sobre a base de dados existente. Para classificar um conjunto de dados desconhecidos, é feito o cálculo do vetor de atributos esperado como entrada para as árvores de decisão, e esse vetor é submetido às árvores, de forma que, de acordo com os valores desses atributos, é identificado a qual grupo aquele determinado conjunto de dados pertence, assim como pode ser visto na Figura 1.

Uma variante da abordagem tradicional de uso das árvores de decisão consiste em criar várias árvores de decisão onde cada uma delas classifica uma parte da amostra dos dados [2]. Após cada árvore analisar os dados e tomar a decisão sobre a classificação, a classificação individual que tiver mais votos é atribuída a um determinado indivíduo que não possui classificação conhecida.

Um dos problemas das árvores de decisão é que elas se ajustam de forma precisa ao conjunto de dados observado na fase treinamento, contudo não são genéricas o suficiente para generalizar os atributos aprendidos na fase de treinamento, não tendo bom desempenho ao operar sobre um conjunto de dados novos.Essa característica é chamada de *overfitting*. Para tanto, as florestas aleatórias (do termo em inglês *random forest*) [4] foram criadas com o objetivo de mitigar esse problema de *overfitting*. Diante disso, quando se cria um conjunto de árvores aleatórias para classificação de amostras desses dados, evita-se esse problema e melhora-se a precisão da classificação.

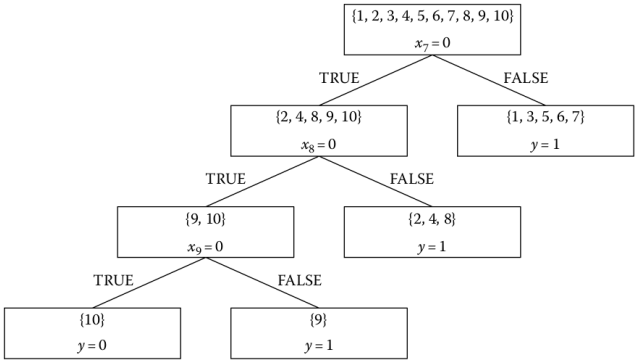
2.1.2 Clusterização

Existem problemas relacionados a aprendizagem de máquina em que é necessário conhecer padrões de dados sem que se faça o treinamento dos algoritmos. Pode-se tomar como exemplo segmentar um conjunto de consumidores de acordo com a renda [15].

O processo de clusterização segundo Gan [11] envolve os seguintes passos:

- Representação do padrão: Escolha das variáveis que serão utilizadas no processo de análise;
- Definição de medida de similaridade: Medida que será utilizada para medir a distância entre indivíduos. Geralmente é utilizada a distância euclidiana;
- Clusterização: Aplicação dos algoritmos de clusterização;

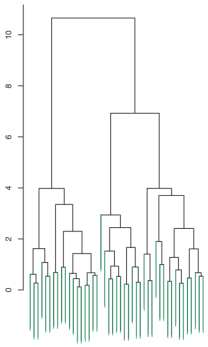
Figura 1: Exemplo de árvore de decisões [28]



- Abstração de Dados: Interpretação dos dados para que os *clusters* sejam melhor compreendidos;
- Avaliação da saída: Verificação da validade do modelo proposto;

De acordo com James et al. [15], a clusterização hierárquica não precisa dizer o número de subgrupos existentes dentro dos dados. A outra vantagem dessa abordagem é que os dados são exibidos em forma de árvore do tipo dendrograma, conforme mostrado na Figura 2.

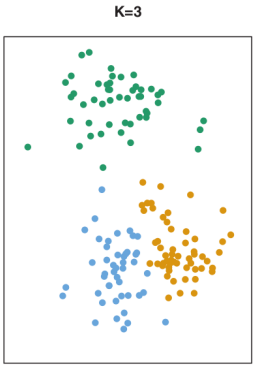
Figura 2: Exemplo de dendrograma [15]



Na clusterização não hierárquica, conforme diz James et al. [15], os subgrupos de dados são buscados de acordo com uma quantidade de subgrupos definida. Esses subgrupos são divididos de acordo com a distância existente entre as suas características. Essas características são representadas por vetores em um espaço multidimensional. A Figura 3 mostra um exemplo de clusterização não hierárquica, em que, ao contrário da clusterização hierárquica, é necessário determinar o número de *clusters*. Primeiro determina-se a quantidade de *clusters* que se deseja, que no caso da Figura 3 é k=3. Logo após essa escolha o algoritmo vai dividir os dados em 3 subgrupos, conforme mostrado na Figura 3.

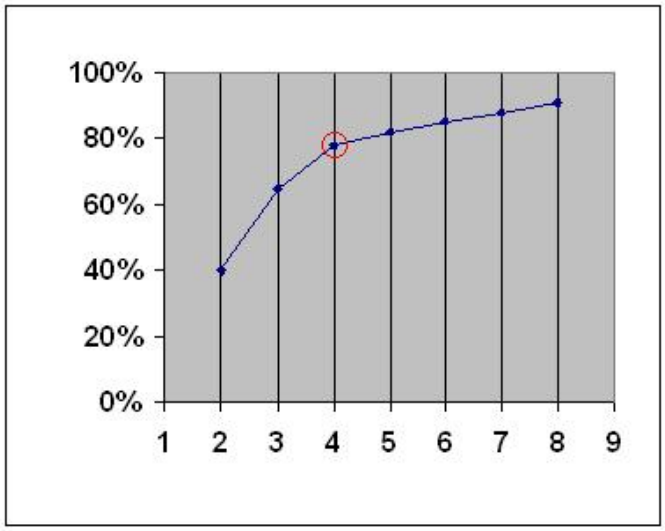
Um questionamento surge a partir da utilização do modelo de clusterização não hierárquica. Como definir a quantidade de *clusters* de um determinado conjunto dados? A resposta da pergunta anterior é: Existe um método chamado curva *elbow* que consiste em uma curva que fornece uma evidência

Figura 3: Exemplo de Clusterização Não Hierárquica [15]



relacionada à quantidade de números de subgrupos existente em um determinado conjunto de dados. Segundo Park [21], o método *elbow* é um método que analisa a porcentagem de variância em função do número de *clusters* existentes em um conjunto de dados. Um exemplo de curva gerada pelo método *elbow* pode ser visto na Figura 4.

Figura 4: Exemplo de curva *Elbow* [26]

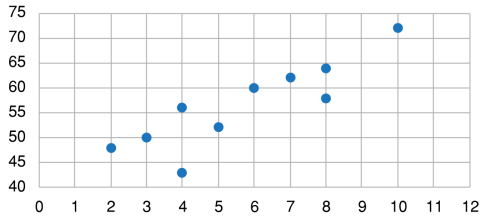


Um outro método utilizado no trabalho para ajudar em sua validação é o gráfico de dispersão. Com o gráfico de dispersão é possível mostrar a relação entre duas variáveis [29]. Visualizando como duas variáveis se comportam é possível verificar como estão distribuídos os pontos. Logo após entender como estão distribuídos os pontos é possível verificar a formação dos *clusters*. A Figura 5 mostra um exemplo de um gráfico de dispersão.

2.1.3 Algoritmo K-Means

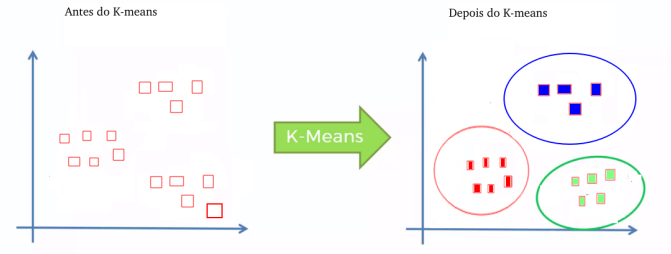
Um algoritmo de clusterização muito popular é o algoritmo *k-means*, que é utilizado para achar sub-grupos em um determinado conjunto de dados. De acordo MacQueen [17], no procedimento *k-means* se escolhem aleatoriamente k pontos dentro do conjunto de dados e em seguida se adiciona

Figura 5: Exemplo de gráfico de dispersão [27]



cada outro ponto presente no conjunto de dados à um grupo associado ao ponto aleatório escolhido que lhe for mais próximo. Depois de um ponto ser adicionado a um grupo, a média desse grupo é ajustada de modo a levar em conta o novo ponto. Assim, em cada estágio, os *k-means* são, de fato, os meios dos grupos que eles representam. Diante desse fato esse algoritmo é ideal para saber quais os grupos de usuários existem nos dados capturados. Na Figura 6, observa-se uma ilustração criada pelo próprio autor demonstrando o funcionamento do algoritmo *k-means*.

Figura 6: Exemplo do funcionamento do k-means.

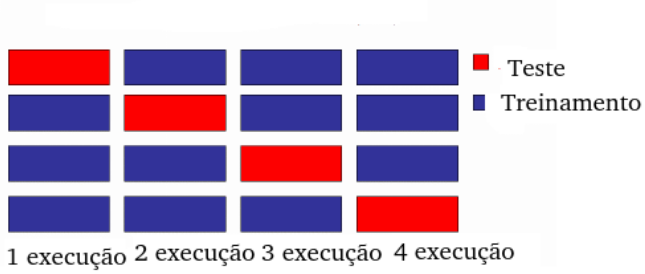


Com os subgrupos encontrados, é mais fácil identificar quais as diferenças entre os usuários individualmente ou em grupo, pois aqueles usuários pertencentes ao mesmo grupo terão características similares, enquanto os usuários pertencentes a grupos diferentes terão características distintas. Com essa análise pronta, fica simples classificar cada usuário individualmente para posteriormente treinar um algoritmo de classificação. Sem a noção de como os dados estão agrupados, este agrupamento teria que ser feito manualmente, e não se teria uma maneira objetiva de agrupar esses dados.

2.1.4 Validação Cruzada

Outro questionamento que surge com os algoritmos relacionados à aprendizagem de máquinas é qual algoritmo utilizar ou mesmo se o algoritmo é eficiente e preciso. Uma vez que, para deixar decisões de negócio sob a responsabilidade de uma máquina, esses algoritmos têm que ser confiáveis, é preciso saber se aquele algoritmo está prevendo os resultados de maneira correta. Caso contrário, isso pode prejudicar o negócio. Para verificar o quão um algoritmo é preciso, os pesquisadores utilizam uma técnica chamada validação cruzada [18]. Essa validação consiste em dividir os dados em dois conjuntos: um para teste e outro para treinamento. Durante o processo calcula-se a diferença entre os valores reais coletados pelo pesquisador e os valores previstos pelo modelo proposto pelo pesquisador. Essa validação é utilizada para verificar a precisão do valor previsto ao aplicar o algoritmo de aprendizagem de máquina sobre o conjunto de dados coletado. De acordo como James et al. [15], validação cruzada *K-fold* envolve dividir aleatoriamente um conjunto de dados em k grupos. Uma parte para teste e a parte k-1 para o treino. Após o término da execução da validação do conjunto de treino e teste, é necessário selecionar outro subconjunto dos dados e fazer a permutação com a parte k-1 para treino. O processo termina quando não houver mais possibilidades de permutar os subconjuntos entre a parte de treino e a parte de teste. A seguir a Figura 7 mostra o funcionamento da validação *k-fold*.

Figura 7: Exemplo do funcionamento do k-fold.



2.2 Interfaces 3D

Muitos pesquisadores estão utilizando métodos relacionados à aprendizagem de máquina na área de interfaces de usuário [1]. A existência de assistentes que ajudam o usuário a aprender melhor a interface dando dicas de uso não é um fenômeno recente. O assistente do *Microsoft Office* [25] é um exemplo do uso da aprendizagem de máquina para melhorar a interação do usuário com a interface. De acordo com Ortega et al. [1], a Interface de Usuário é o meio de comunicação entre o usuário e o sistema computacional. A interface de usuário recebe os dados de entrada do usuário e tem como saída os dados formatados de uma maneira que o usuário possa entender. Então as interfaces 3D são os meios de comunicação entre os usuários e os objetos tridimensionais. Segundo Ortega et al. [1], é importante estudar interfaces 3D pelos seguintes motivos:

- Tarefas reais são feitas em 3D;
- As interfaces 3D estão se desenvolvendo cada vez mais;
- As interfaces 3D são mais complicadas de se implementar;
- Interfaces 3D necessitam de um estudo maior para melhorar a usabilidade.

Um dos desafios no desenvolvimento de interfaces 3D é usar a inteligência artificial para melhorar as interfaces. Pode-se citar como exemplo fazer uma operação sobre o objeto 3D e prever o local onde o objeto será colocado.

Com a aplicação das ferramentas de inteligência artificial, busca-se melhorar a relação entre a interface 3D com seus respectivos usuários, além de aumentar a produtividade na atividade de modelagem 3D.

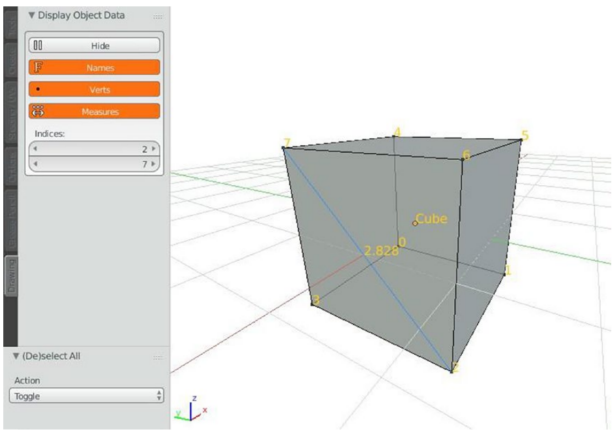
2.2.1 *Modelagem 3D*

De acordo com Conlan [3], modelagem 3D é a arte de manipular dados para criar representações de objetos e ambientes 3D. De acordo com Conlan [6], existem dois tipos de modelagem.

- Manual: A modelagem manual é feita através de uma ferramenta voltada a criação de modelos 3D. Exemplos de ferramentas são *Blender*, *Maya* e *3ds max*. Os modelos são criados nessas ferramentas através de primitivas como cubos, cilindros, dentre outras. Ao aplicar uma série de operações nessas primitivas, constrói-se o modelo desejado.
- Automática: A modelagem automática é feita por algoritmos computacionais.

Este trabalho visa coletar dados automaticamente a partir da criação de modelos 3D feitos de maneira manual. O objetivo é coletar esses dados durante a criação do modelo por um determinado usuário. Em seguida, analisar esses dados para verificar a *expertise* de determinados grupos de usuário. Após conhecer como é distribuído o grupo de usuários, este trabalho almeja com base nesses dados identificar em qual grupo se encontra um usuário individual. Na Figura 8, segue um exemplo de interface voltada a modelagem de objetos 3D.

Figura 8: Exemplo De Modelagem e a interface do *Software Blender* [6]



2.3 Sistemas especialistas

O objetivo desse trabalho é caracterizar métricas de produtividade de usuário através da quantidade de operações feitas por ele, bem como pelo tempo que ele leva para produzir um determinado modelo e a forma geométrica do objeto. Segundo Mendes [19], sistemas especialistas são *softwares* que são baseados em conhecimento, construídos principalmente com base em regras que reproduzem o conhecimento do perito e que são utilizados para resolução de determinados problemas em domínios específicos.

Já segundo Peter e Linda et al[16], sistemas especialistas são sistemas capazes de oferecer soluções em determinado domínio ou são habilitados a dar conselhos no mesmo nível que especialistas em seus respectivos campos de trabalho ou estudo.

Para Varanda Paiva [16], um sistema especialista é um método de solução vinculado à inteligência artificial. Esse sistema usa uma base de conhecimento fornecida por um conjunto de especialistas no assunto. Além disso, ele utiliza um modo de inferir este conhecimento por meio da simulação do raciocínio usada pelos especialistas para resolver determinados tipos de problemas.

De acordo com Varanda Paiva [20], a diferença entre sistemas especialistas e sistemas normais é o fato de que eles usam um modelo de algoritmos, enquanto sistemas especialistas usam modelos baseados em heurísticas na resolução dos problemas.

Segundo Dewendra et al. [9], os sistemas especialistas possuem diversas características. Entre elas, pode-se citar:

- É projetado para um problema específico;
- Reage diante da incerteza de dados;
- Possui uma curva de aprendizado menor;
- Faz recomendações para o usuário;
- É projetado para crescer de maneira incremental;
- O conhecimento é adquirido e codificado durante entrevistas com os especialistas.

Segundo Varanda Paiva [20], os sistemas especialistas são formados por:

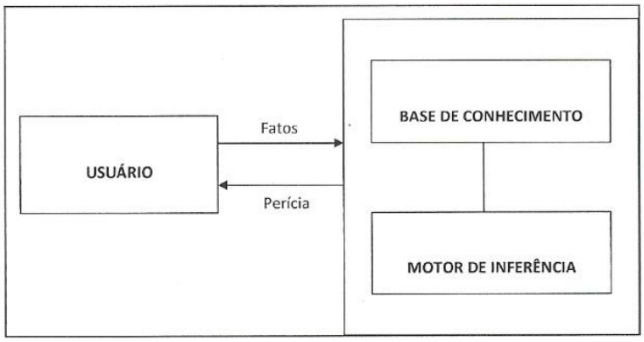
- Interface: Responsável por interagir com os usuários através de comandos vindos de dispositivos como teclados ou *mouses*.
- Base de conhecimento: Conhecimento do problema a ser resolvido.
- Motor de inferência: Regras implementadas no *software* para resolver problemas de um domínio específico.

A Figura 9 mostra a estrutura com todos os componentes de um sistema especialista.

Existem diversas vantagens apontadas ao adotar um sistema especialista. De acordo com Mendes [19], pode-se citar:

- Um sistema especialista esconde a complexidade do problema e então pode ser aprendido mais facilmente por uma maior quantidade de pessoas. Com isso, o conhecimento do especialista é distribuído para várias pessoas.
- Um sistema especialista abstrai o conhecimento e a experiência do especialista, fazendo com que o usuário possa fazer tarefas complexas sem ter que conhecer profundamente como de fato funcionam as heurísticas e as regras implementadas no *software*.
- Sistemas especialistas procuram reduzir a dependência entre as organizações e os seus respectivos especialistas.

Figura 9: Estrutura de um sistema especialista [20]



- Sistemas especialistas são ideais para o treinamento de pessoas em determinado domínios, pois eles implementam funções que ajudarão as pessoas a aprenderem a fazer determinadas atividades de maneira mais simples, já que eles abstraem toda a complexidade do problema que foi resolvido pelo *software*.

2.4 Usabilidade

De acordo com Ortega et al. [1], usabilidade é quando existe a aceitação do usuário de um sistema, ou seja, quando ele satisfaz as suas necessidades. Assim, pode-se concluir que quando o usuário consegue realizar suas atividades em um determinado sistema, aquele determinado sistema tem uma boa usabilidade. Outra definição de acordo com Ortega et al. [1], diz que a usabilidade é processo da engenharia que busca entender a comunicação entre a interface e o usuário que a usa. Abaixo alguns atributos relacionados à usabilidade:

- Aprendizagem: É atributo do sistema que está relacionado a sua facilidade de uso e do usuário rapidamente aprendê-lo;
- Eficiência: Diz respeito a produtividade do usuário ao realizar alguma tarefa no sistema;
- Memorabilidade: É um atributo que mensura o quão o sistema consegue ser lembrado pelos usuários após passar um certo tempo sem utilizá-lo;
- Erro: É a quantidade de erros do usuário ao realizar determinada tarefa;
- Satisfação: É o atributo relacionado à agradabilidade do sistema na perspectiva do usuário.

2.4.1 Avaliação de usabilidade

A avaliação de usabilidade consiste em medir o quanto aquele determinado sistema ou funcionalidade atende as necessidades do usuário [14]. O processo de medir usabilidade de acordo com Ivory e Hearst [14], consiste na divisão dos seguintes passos.

- Captura de dados: Consiste em colecionar dados que possam ser utilizados na avaliação de usabilidade. Pode-se citar como exemplo erros cometidos pelo usuário bem como o tempo que ele gasta para realizar uma determinada tarefa;
- Análise de dados: Consiste em interpretar os dados coletados na fase anterior;
- Crítica: Busca corrigir os erros ou propor uma nova interface com base na interpretação dos dados coletados na fase anterior;

A produção de software em massa de uma determinada empresa não deveria influenciar no aumento da equipe para avaliar esses *softwares*. Tal medida deixaria o negócio inviável. Logo, Ivory e Hearst [14] propõem automatizar alguns passos ou todos os passos supracitados.

2.5 Métodos de Avaliação de interfaces

Avaliação de interface, segundo Bowman et al. [3], é simplesmente a apreciação, análise e teste de uma interface, visando entender a interação do usuário com a interface do *software*. Existem vários métodos de avaliação de interfaces. Dentre os métodos de avaliação pode-se destacar:

- Avaliação baseada em heurísticas
- Avaliação formativa
- Caminhos cognitivos
- Questionários
- Entrevistas
- Avaliação do desempenho de um usuário ao realizar alguma tarefa
- Avaliação da performance da interface de um determinado *software*.

Segundo Bowman et al. [3], **avaliação baseada em heurística** é feita por um conjunto de profissionais especializados em desenvolvimento de interfaces. Nesse tipo de avaliação, busca-se verificar a qualidade das interfaces de usuário através da avaliação feita por profissionais que entendem de desenvolvimento e construção de interfaces de usuário.

De acordo com Bowman et al. [3], **caminho cognitivo** é uma abordagem que busca entender a relação do usuário e a interface através dos caminhos a serem seguidos pelo usuário na realização de uma determinada tarefa. Por exemplo, ao digitar uma palavra errada, o primeiro pensamento do usuário é apagá-la e reescrevê-la corretamente. Se o usuário já tem uma experiência com outro *software* que realiza a atividade anterior da mesma forma, ele não terá dificuldade alguma em realizar essa tarefa, pois os passos que ele faz para a realização de uma determinada tarefa já estão gravados em sua memória. Esse caminho que ele percorre para realizar essa tarefa é chamado de caminho cognitivo.

Avaliação formativa é o processo de entender a relação do usuário com a interface do *software* através da realização de uma determinada tarefa [3]. Na realização de uma determinado tarefa pode-se capturar por exemplo o tempo de realização da atividade bem como outras variáveis que sejam importantes para verificar o desempenho de um grupo de usuários ao realizar determinado projeto. Muitos *softwares* capturam dados quando utilizados, possibilitando a realização de algumas análises desses dados que podem levar o desenvolvedor à conclusão de que é preciso fazer modificações na interface.

Conforme Bowman et al. [3], a **avaliação comparativa** faz um estudo comparativo entre técnicas de design, componentes e interface bem como sua usabilidade. Uma maneira possível de fazer essa avaliação seria pedir ao usuário para realizar uma determinada tarefa em *softwares* com interfaces diferentes. Em seguida, coletar dados durante a realização dessa determinada tarefa. Após a captura de dados realiza-se a análise dos mesmos. Em detrimento dessa análise, decide-se ou não alterar a interface do usuário.

Segundo Bowman et al. [3], os **questionários** são utilizados antes e depois do processo de desenvolvimento da interface, a fim de obter as opiniões dos usuários a respeito da interface desenvolvida. É importante conhecer as opiniões dos usuários, pois são eles que utilizarão o *software* para a realização das atividades de uma determinada organização. Caso eles tenham uma maior facilidade no manuseio daquela determinada ferramenta, haverá uma maior produtividade por parte daquele determinado conjunto de usuários.

De acordo com Bowman et al. [3], **entrevista** é uma técnica utilizada para capturar informações falando diretamente com o usuário. Nessa conversa pode existir uma série de questões predefinidas. No entanto existe a possibilidade de que o entrevistador faça perguntas não programadas ao usuário. Isso faz com que a entrevista seja um método de captura de informações mais informal.

2.6 Métricas para interfaces

Conforme Bowman et al. [3], as métricas de interface são as medidas utilizadas para mensurar as características da interface a ser avaliada. As métricas de *performance* de sistema dizem respeito ao processamento de dados para o *software* pelo usuário e pela espera do termino desse processamento. Se não houver um bom tempo de resposta do *software*, o usuário ficará impaciente e sua produtividade poderá ser comprometida [3]. A métrica baseada em tarefas mede a qualidade de uma determinada tarefa feita pelo usuário. Um exemplo disso é o cálculo do tempo de um usuário ao digitar um texto com 100 palavras. Outro exemplo é a contagem da quantidade de erros que o usuário cometeu durante a realização dessa digitação [3].

Respostas subjetivas são as impressões que o usuário tem da interface: facilidade de uso, curva de aprendizagem baixa e satisfação ao utilizar o software. Essas respostas subjetivas são capturadas através de questionários ou entrevistas. Os dados que podem ser coletados são qualitativos ou quantitativos [3].

3. TRABALHOS CORRELATOS

De acordo com Fang e Zhai [10], a recuperação de especialistas é uma área da computação que se preocupa em localizar especialistas em um tópico especializado de forma automática. O trabalho de Fang e Zhai [10] propõe um *framework* geral baseado em probabilidades para localização de especialistas. Os modelos que são experimentados no trabalho são a localização de especialista com base em perfis e o outro modelo é o de recuperação de informação com base em documentos. Segundo os autores, resolver o problema de localizar os especialistas mais relevantes consiste em fazer um *ranking* dos especialistas mais relevantes dado um tópico de interesse. Então este trabalho busca calcular as probabilidades de um determinado especialista ser o indivíduo mais relevante dado uma busca. Convencionou-se no trabalho que o individuo que tivesse maior probabilidade seria o que teria mais relevância. Diante disso é possível que se possa criar um *ranking* dos dez melhores dado um determinado tópico.

O trabalho de Campbell et al. [5] utiliza dados gerados pela comunicação por *emails* para localização de *expertise*. No entanto, além de utilizar os dados dos textos dos *emails*, a proposta do trabalho é também utilizar os padrões de comunicação para localização dos especialistas, ou seja, os autores propõem um modelo em redes para a recuperação de especialistas. Basicamente o processo é feito em três partes. A primeira parte consiste em colecionar todos os *emails* relacionados a um determinado tópico. A segunda etapa consiste em analisar os *emails* enviados por cada par. A última etapa consiste em analisar o grafo de *expertise* de cada usuário.

O trabalho de Patrick [13], tem como meta modelar um conjunto de ações de interface como se fosse uma árvore. Como exemplo, o clique do botão poderia ser um nó da árvore e o pressionamento de uma tecla seria outro nó. Esta árvore é comparada a outras árvores geradas. Daí as tarefas semelhantes são detectadas. Por fim as árvores com tarefas semelhantes são unidas. Outra possibilidade é a detecção de sequências de ações que o usuário faz. Com isso, localiza-se algum dado que possa ajudar na compreensão no uso da interface, para que se possa utilizar esse método na automatização de avaliação de interfaces.

O trabalho de Tao [24] busca usar da programação orientada a aspectos para capturar os dados do usuário. A programação orientadas a aspectos é o modelo de programação que possibilita a devida separação de responsabilidades. No artigo o autor procura criar aspectos que são responsáveis por capturar os dados de eventos do usuário para posterior análise. Tao [24] somente cumpre as etapas de captura de dados e a parte de análise de dados e crítica conforme a metodologia de Ivory e Hearst [14]. Conclui-se que usar como método a programação orientada a aspectos é um sucesso, já que ela é um método não intrusivo. O que permite a retirada dos aspectos quando não for mais necessário. Dado que a execução dos aspectos podem comprometer a performance no sistema. Logo conclui-se que a programação a aspectos é um bom método de captura de dados de usuário para fins de avaliação de interfaces.

Já no trabalho de Ivory e Hearst [14], demonstra-se as vantagens ao automatizar os processos de avaliação de interfaces. Adicionalmente, propõe-se uma taxonomia de automatização de avaliação de interfaces de usuário. As vantagens obtidas durante a automatização de tarefas de avaliação de interfaces de usuário são de acordo com Ivory e Hearst [14]:

- Reduzir o custo de avaliação de interface de usuário. Métodos que capturem, analisem e critiquem uma interface de usuário podem diminuir o tempo de avaliação de interfaces de usuário e consequentemente, os custos.
- Aumentar as descobertas de erros causados pelo usuário. Assim, criam-se modelos de finalização de tarefas, para que se possam detectar o quanto se desviou desses modelos de tarefas.
- Prever custos de tempo e erro em um design inteiro da interface. Como não é sempre possível avaliar todos os aspectos de uma interface usando avaliação não automatizada, modelos analíticos permitem ampliar a cobertura de recursos avaliados.
- Reduzir a quantidade de pessoas que avaliam a inter-

face. A automatização de tarefas como crítica reduz a quantidade de pessoas que fazem a avaliação da interface bem como a necessidade de contratar pessoas que não tenham tanto conhecimento em avaliação de interfaces.

- Aumentar a quantidade de características que podem ser avaliadas. Como o tempo de avaliação de interfaces é menor, logo uma maior quantidade de características da interface pode ser avaliada.
- Permitir comparação entre diferentes designs de interface. A criação de um modelo de completude de tarefas permite verificar em qual interface o usuário tem a maior produtividade.
- Incorporar o processo de design de interface no desenvolvimento, ao invés de ser testado depois da fase de desenvolvimento.

Embora haja uma série de vantagens, a automatização da avaliação de interfaces segundo Ivory e Hearst [14] não é um método substitutivo de outros métodos como heurísticas ou respostas subjetivas, pois existem métodos que podem ser utilizados antes do desenvolvimento da interface começar. A taxonomia proposta por Ivory e Hearst [14] é:

1. Método de Classe: descreve o método de avaliação em alto nível e os tipos são:
   - Teste: Um avaliador observa usuários interagindo com a interface e determina problemas de usabilidade.
   - Inspeção: Um avaliador usa uma série de heurísticas para encontrar os problemas de usabilidade.
   - Consulta: Usuário provê *feedback* via entrevistas, questionários e outros.
   - Modelos analíticos: Um avaliador emprega modelos de usuário e interface para gerar previsões de usabilidade.
   - Simulação: Um avaliador emprega modelos de interface para imitar um usuário interagindo com uma interface e criar um relatório dos resultados dessa interação.
2. Tipo de método: descreve como a avaliação é conduzida dentro de uma classe de método. Pensando o protocolo como classe de método teste ou método de simulação.
3. Tipo de Automatização: descreve a parte do processo que será automatizada.
   - Nenhuma Automatização: Nenhuma parte do processo de avaliação é automatizado.
   - Captura: Os dados relacionados à usabilidade são capturados de forma automática.
   - Análise: O *software* identifica automaticamente os problemas de usabilidade.
   - Crítica: O *software* mostra os relatórios dos problemas e as possíveis soluções para a resolução dos mesmo.
4. Nível de Esforço: descreve o esforço necessário para fazer a avaliação da interface.
   - Esforço Mínimo: não requer uso ou modelagem de interface.
   - Modelo de Desenvolvimento: requer que o avaliador desenvolva um modelo de interface de usuário e / ou um modelo de usuário, a fim de empregar o método.
   - Uso Informal: Deixa o usuário fazer tarefas livremente.
   - Uso Formal: O usuário fará tarefas que o avaliador escolher.

4. CARACTERIZAÇÃO DAS MÉTRICAS

A seguir serão mostradas as descrições dos métodos que a ferramenta proposta neste trabalho utiliza para a captura e a análise dos dados. Os métodos empregados seguem a taxonomia de Ivory e Hearst [14]. Os métodos são: Método de Classe, Tipo de método, Tipo de Automatização e Nível de Esforço. Para a análise dos dados capturados são utilizadas algumas técnicas relacionadas à aprendizagem de máquina como a classificação e clusterização. As métrica foram propostas neste trabalho, porque ao revisar a literatura não se encontrou nenhum trabalho semelhante relacionado à recuperação de especialistas voltados a ferramentas 3D. Como as operações e a geometria dos objetos são fundamentais na criação dos objetos 3D, foram selecionadas as métricas citadas abaixo com adição do tempo de projeto. A função do tempo é medir a eficiência da qual um objeto geométrico é modelado, a partir da quantidade de operações feitas para concluí-lo.

4.1 Avaliação da expertise do usuário

1. Representação do Padrão (variáveis a serem escolhidas).
   - Tempo de projeto: Mede o quão o usuário é eficiente ao lidar com a ferramenta. Daí a importância de capturar o tempo que o usuário leva para fazer aquela determinada atividade.
   - Número de faces do objeto: Mostra quão completo está o objeto do ponto de vista geométrico. Por exemplo se um usuário esquecer uma face ou apagá-la por acidente, ao se comparar o modelo do usuário com o modelo completo, será possível verificar o quão esses objetos são diferentes.
   - Número de arestas: Mostra o quão o modelo é completo geometricamente em relação a outro modelo construído por outro usuário.
   - Número de vértices: Mostra quão determinado modelo é completo geometricamente em relação a outro modelo.
   - Quantidade de translações: Para a criação de um determinado modelo deve existir uma quantidade mínima de operações a serem feitas. Logo o usuário que possui o menor número de operações tem uma quantidade de erro menor.
   - Quantidade de extrusão: Assim como na translação é importante saber o quanto o usuário se desviou da quantidade mínima de operações que é necessário para fazer aquele determinado modelo.

- Quantidade de redimensionamento: Assim como na translação e extrusão é importante saber o quanto o usuário se desviou da quantidade mínima de operações que é necessárias para fazer aquele determinado modelo.
- Quantidade de rotação: Contar o número de operações é importante, porque para a criação de um determinado modelo deve existir uma quantidade mínima de operações a serem feitas. Logo o usuário que tiver o menor número de operações terá uma quantidade de erro menor.
- Quantidade do operador de adicionar primitiva: Contabilizar a quantidade de operações de adição de primitivas, objetos que ao sofrer determinadas operações criam objetos mais complexos. Exemplos de primitivas são cubo, cilindro, esfera e outros). Isso é importante, para saber o quanto o usuário se desviou da quantidade mínima de operações que é preciso para fazer aquele determinado modelo.
- Quantidade de operações de deleção: As operações de deleção e de adição de primitivas são utilizadas, porque são operações opostas. Uma quantidade de operações de adicionar seguidas de muitas quantidades de operações de deleção podem representar erros do usuário na criação dos modelos.

2. Método de classe.
   - O método de classe escolhido é o modelo analítico. Nesse modelo, o avaliador emprega modelos de usuário e interface para gerar previsões de usabilidade [14]. Foi feita essa escolha, pois a proposta do trabalho consistiu em propor um modelo de medição de *expertise* de usuário voltado a ferramentas de modelagem de objetos 3D.
3. Nível de esforço Formal.
   - O nível de esforço utilizado no trabalho foi o formal. Esforço formal consiste na escolha de uma tarefa que seja escolhida pelo avaliador [14]. Pois é preciso que haja uma tarefa em comum para todos os usuários, para então, calcular-se com maior precisão as diferenças dos dados coletados durante a realização da atividade pelo usuário.
4. Captura de Dados.
   - Os dados de usuário são capturados de forma automática no momento em que o usuário desenvolve uma atividade predefinida pelo avaliador. Os dados foram capturados através de *scripts* feitos em *python*. O *Blender* possui uma API que permite a manipulação da geometria dos objetos. Além disso possibilita a captura de dados necessários para a realização deste trabalho. Parte dos dados capturados foram providos pelo sistema de *log* do próprio *Blender*. Outros dados foram capturados analisando-se a cena da interface do *Blender* e o tempo coletado durante a realização da atividade desde que o *Blender* é executado. O tempo de realização da atividade é o tempo que o usuário salva o modelo no arquivo subtraído do tempo de início da realização da atividade no *Blender*. Os dados foram capturados em um curso ministrado pelo autor do trabalho. Esse curso é dividido em duas partes. A primeira parte introduziu ao usuário como utilizar os principais operadores providos pelo *Blender*. No segundo momento uma tarefa é passada para o usuário e nesse momento a captura de dados é realizada. A atividade a ser praticada é a mesma para todos os usuários. Isso garante um grau de dificuldade igual para todos os participantes do estudo, podendo assim avaliar os dados de maneira mais precisa, e podendo verificar de maneira fidedigna quais os usuários que têm mais *expertise*.
5. Definição da medida de Dissimilaridade.
   - A medida de dissimilaridade selecionada é a distância euclidiana. Essa medida é escolhida com base no domínio do problema. Como os dados em sua maioria são naturais e contínuos não existe necessidade de buscar outras formas de medir a dissimilaridade.
6. Análise e transformação dos dados.
   - Após a captura dos dados são transformados. Nem sempre os dados que são armazenados estão em um formato adequado para a análise. Diante disso, é preciso que os dados sejam transformados para a posterior análise. O *Blender* salva os arquivos de *log* em um formato de uma lista da linguagem *python*. Além disso salva operações que não são utilizadas nesse trabalho. Diante disso foi necessário criar um *script* que filtrasse somente os dados relevantes e contabilizasse o número de operações relevantes para o trabalho. Após executar o *script* os dados filtrados foram salvos em um arquivo (*CSV Comma-separated values*). Após deixar os dados em um formato ideal, foi o momento de se aplicar os algoritmos de clusterização e classificação. O algoritmo de clusterização é aplicado para entender se existem subgrupos dentro dos dados coletados. Já o algoritmo de classificação serve para apontar a qual grupo pertence aquele dado não classificado ou não nomeado.

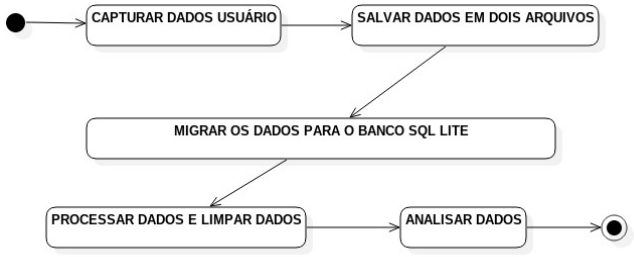
   Na Figura 10 é mostrado o diagrama de atividades da ferramenta implementada.

4.2 Requisitos funcionais

A seguir serão mostrados os requisitos funcionais do presente trabalho.

- Captura de dados: É a parte da ferramenta onde os dados são coletados durante a modelagem do objeto 3D. Essa parte da ferramenta está implementada em *Python*. Após a captura de todos os dados eles são salvos em dois arquivos distintos. Um arquivo armazena os seguintes dados:
  - Tempo de projeto.
  - Número de faces.
  - Número de arestas.

Figura 10: Diagrama atividade da ferramenta proposta pelo autor.



  - Número de vértices.

  Já o outro arquivo guarda as seguintes variáveis:

  - Quantidade de translações.
  - Quantidade de rotações.
  - Quantidade de redimensionamentos.
  - Quantidade de operadores de adicionar primitivas.
  - Quantidades de operações de *deleção*.
  - Quantidade de operações de extrusão.

  O formato usado para salvar o primeiro arquivo é o formato separado por ponto e virgula, conhecido popularmente como (*CSV Comma-separated values*). Já o segundo arquivo é salvo pelo formato que o *Blender* escolhe para registrar as operações de registro de *log*.

- Limpeza e estruturação dos dados: Essa etapa tem como objetivo a filtragem dos dados relevantes para o projeto. Outro objetivo é a transformação desses dados em um formato passível de análise apropriada. No caso do projeto, é preciso que na ferramenta exista uma parte que possa filtrar os dados de registro de *log* do *Blender*. Adicionalmente são necessárias a migração e junção desses dados em uma fonte de dados única. Os dados ao serem unidos são migrados para uma base de dados *SQL Lite*. Foi escolhido o banco de dados *SQL Lite*, porque ele é um banco de dados em arquivo e passível de manipulação via SQL (*Structured Query Language*).

- Pré-Processamento dos dados: O propósito dessa etapa é fazer a normalização e padronização dos dados. Essa fase é necessária, pois ao analisar os dados sem fazer esse pré-processamento, é possível que alguns algoritmos sejam sensíveis a valores extremos entre duas ou mais variáveis. Tal viés comprometeria a integridade da análise dos dados bem como as previsões que serão feitas posteriormente.

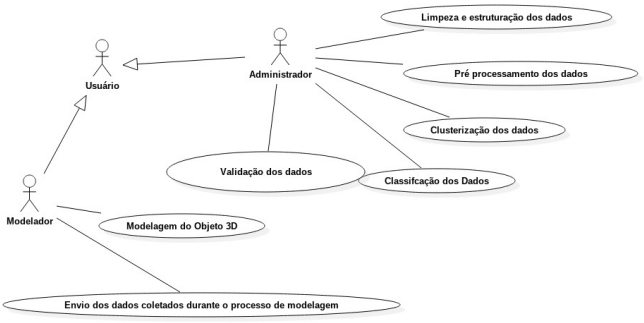
- Clusterização dos dados: Nessa etapa ocorre a identificação dos sub-grupos dentro do conjunto de dados. A clusterização é relevante, pois mostra o comportamento dos dados e como os usuários estão agrupados de acordo com a *expertise*.

- Classificação dos dados: Nessa fase há a identificação individual sobre qual grupo cada usuário pertence.

- Validação: Tem como objetivo calcular a porcentagem de acertos ao classificar o usuário individualmente. Para isso é utilizada a validação cruzada.

Na Figura 11, o autor mostra o diagrama de caso de uso da ferramenta proposta no trabalho.

Figura 11: Diagrama de caso de uso da ferramenta proposta pelo autor.



4.3 Requisitos não funcionais

Interoperabilidade é a propriedade de um sistema de se comunicar ou ser executado em ambientes computacionais heterogêneos. O *Blender* é um *software* interoperável que pode ser executado em diversos sistemas operacionais. Essa propriedade do *Blender* e do *Python* possibilitou a criação de um *script* que coletasse dados e pudesse ser executada em diversos sistemas operacionais. Todo o processo de desenvolvimento da ferramenta preocupou-se que a caracterização fosse executada nos sistemas operacionais mais comuns como Linux e Windows.

5. MÉTODO

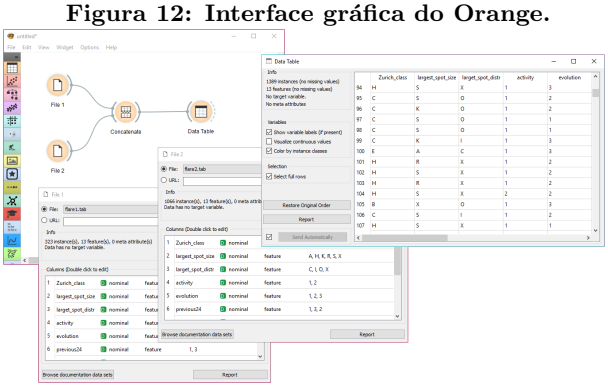
No início da pesquisa, procurou-se dados relacionados ao problema que este trabalho pretende resolver. No entanto esses dados não foram encontrados. Portanto, foi necessário encontrar uma maneira de ter esses dados disponíveis. Para esse trabalho, foi criado um *script* capaz de capturar os dados durante a modelagem de um objeto 3D. Para a captura dos dados foi necessário lecionar um curso de *Blender* de onde se pôde coletar dados dos usuários. O curso foi dividido em duas etapas. A primeira etapa do treinamento teve como foco dar uma introdução ao estudo de modelagem 3D. Já na segunda foi definida uma tarefa comum a todos e nessa parte os dados foram coletados. Essa atividade foi realizada da seguinte forma. Os usuários foram divididos em dois grupos: o primeiro recebeu um treinamento dos principais comandos utilizados para fazer a modelagem do objeto. Após o treinamento a tarefa foi designada a esses usuários. Já para o segundo grupo de usuários a atividade foi proposta antes do treinamento. Desse modo, foi possível determinar se os algoritmos foram capazes de distinguir os grupos de usuários treinados dos que não receberam instrução. Durante a realização das tarefas foram armazenados os dados utilizados como entrada para os algoritmos de clusterização. Estes foram salvos em dois arquivos diferentes. Em seguida esses dados foram migrados para uma base de dados em *SQL Lite*. Tal passo é necessário para permitir a clusterização e

a classificação dos mesmos. Por fim, para determinar a precisão de cada algoritmo de classificação utilizou-se o método *k-fold*.

Outros métodos que foram utilizados para verificar a utilização do *k-means* foi o dendrogramas que consiste em um gráfico em aŕvore que mostra os *clusters* divididos pelo *kmeans*. Um outro método que foi utilizado para a avaliar a aplicação de *kmeans* foi o método *Método elbow* que consiste em saber a quantidade de *clusters* ideial para determinada quantidade de dados. Para finalizar a avaliação relacionada à clusterização de dados foi criado um gráfico de dispersão mostrando como os dados estão divididos. Além disso, foram avaliados de forma qualitativa dois objetos criados pelos usuários. Os objetos avaliados foram tirados de grupos distintos. Aquele grupo que recebeu o treinamento comparando com o objeto de um usuário que não recebeu o treinamento antes de fazer a atividade.

Para caracterizar as métricas de usabilidade a partir de algoritmos da área de aprendizagem de máquina, foi utilizado o *Orange*, que é um ambiente voltado a programação visual que possui diversos componentes implementados. Um dos componentes implementados é o de teste e *score* assim como os componentes dos algoritmos de classificação que foram utilizados neste trabalho. Por esses motivos que o *Orange* foi escolhido. Conforme defende Janez Dem [8], o *Orange* é uma suíte voltada a mineração de dados e aprendizagem de máquina voltada a linguagem *python* e para a programação visual. Pela ferramenta já ter implementado diversos algoritmos de aprendizagem de máquina, além de possuir ferramentas voltadas para validar os algoritmos de classificação, é que foi escolhido o *Orange* para execução e validação da abordagem proposta. A Figura 12 mostra uma imagem que representa a interface gráfica do *Orange*.

Figura 12: Interface gráfica do Orange.



6. RESULTADOS

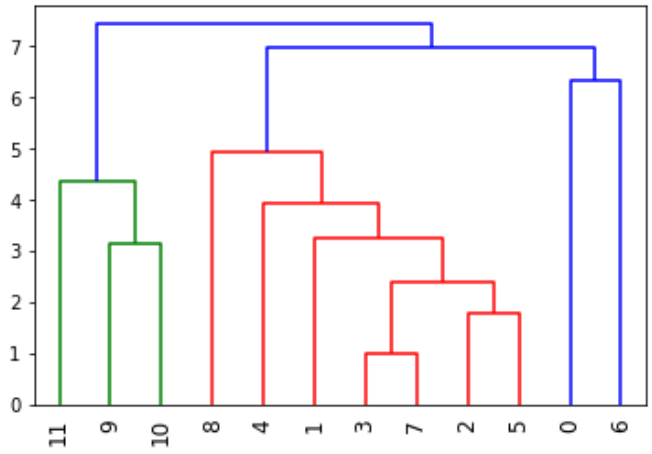
A seguir são demonstrados os resultados da análise dos dados que foram coletados.

6.1 Análise de Dados

Após a coleta de dados foi necessário excluir alguns usuários, uma vez que alguns dados estavam errados e outros não satisfizeram o critério de ter poucos polígonos. Possivelmente o usuário utilizou o operador *smooth* existente no *Blender* fazendo com que o número de polígonos crescesse muito ao salvar parte de sua geometria. Após a escolha dos dados aplicou-se uma série de análise de dados explicadas e exibidas a seguir.

6.2 Dendrograma

Figura 13: Dendrograma



Nesta seção encontram-se respostas para algumas questões levantadas por esse trabalho. Uma delas é: é possível com as variáveis propostas agrupar os usuários de acordo com sua *expertise*? Outro questionamento a ser respondido é se o método *k-means* pode ser utilizado para fazer esse agrupamento tendo como entrada as variáveis propostas nesse trabalho.

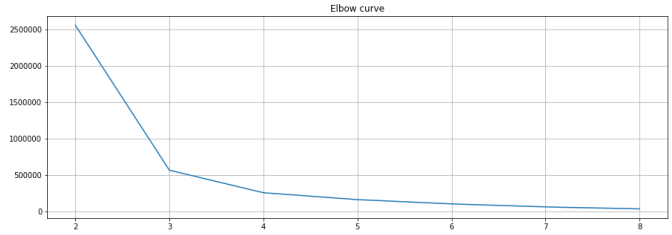
Para a validação dessa parte do trabalho algumas premissas são estipuladas na hora da coleta de dados e na escolha da atividade a ser realizada. No momento da captura de dados, os usuários estão divididos em dois grupos. O primeiro grupo é composto por aqueles usuários que fizeram a atividade sem nenhum treinamento, representado no dendrograma pelos indivíduos [9,10,11] mostrados na Figura 13. Já o outro grupo realizou a tarefa após a realização do treinamento que são representados pelos indivíduos de [0 a 8] na Figura 13. Outra premissa estabelecida é que todos os indivíduos realizam a mesma tarefa. Dentro do grupo dos que realizam o treinamento existem dois subgrupos que são representados respectivamente por [1,2,3,4,5,7] representados pelos usuários que tiveram um melhor desempenho no geral enquanto que os usuários [0,6] obtiveram um desempenho intermediário de acordo com as variáveis propostas pelo trabalho.

Observa-se no dendrograma da Figura 13 que os indivíduos que fizeram o treinamento antes de realizarem a atividade estão em um grupo diferente daqueles que fizeram a atividade sem o treinamento. E percebe-se também que mesmo dentre os usuários treinados nota-se que o algoritmo os dividiu. Indicando assim que pode-se classificar os usuários em iniciante, intermediário e avançado. Diante desse fatos é possível concluir que o *kmeans* e as variáveis propostas são válidas para agrupar os usuários de acordo com sua *expertise*.

6.3 Método elbow

Um outro método utilizado para validar a divisão dos usuários de acordo com sua *expertise* é saber qual é a quantidade desses subgrupos para uma determinado conjunto de

Figura 14: Método elbow

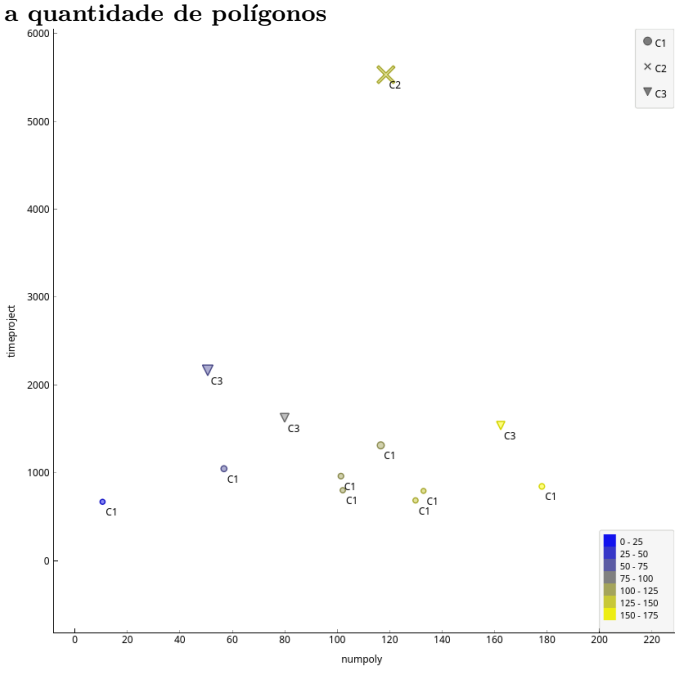


dados. Para determinar a quantidade de subgrupos de usuários é utilizado um método que quantifica *clusters* de um determinado conjunto de dados. Para tanto, o método *elbow* consiste em testar a variância dos dados em relação a quantidade de subgrupos formados. Assim que o benefício deixa de ser relevante (um salto entre uma quantidade de subgrupos e a próxima quantidade) ele entra em um estágio plano no gráfico representado na figura 14. Nesse momento nota-se que diferença da distância é quase insignificante. Entende-se que o algoritmo é relevante com aquela quantidade de K. Nesse caso 3 subgrupos de usuários. Após a visualização da figura 14, conclui-se que o número de subgrupos é três. Portanto quando aumenta-se o número de subgrupos não existe mais diferença na variância.

Ao confrontar os três subgrupos apontados pelo gráfico *elbow* com a divisão geranda no dendrograma da Figura 13. Confirma-se os três grupos de usuários distintos pela *expertise*.

6.4 Gráfico de dispersão

Figura 15: Gráfico de tempo de projeto em relação a quantidade de polígonos



Outro método para verificar a validade das hipóteses propostas nesse trabalho consiste em usar um gráfico de dispersão. Os gráficos de dispersão utilizam coordenadas cartesianas para exibir os valores dos dados de um determinado conjunto. Para a criação do gráfico mostrado na figura 15 utilizam-se as variáveis tempo e número de polígonos. Com base na dispersão dos dados e na distâncias entre os pontos observa-se três grupos bem definidos no gráfico da figura 15. Esse é um indício da validade dos métodos propostos no presente trabalho. Fazendo gráficos com outras variáveis é possível ter uma ideia de qual usuário escolher como especialista. Essa escolha vai depender dos requisitos necessários que o contratante do especialista deseja.

6.5 Comparação dos Algoritmos de classificação .

Tabela 1: Tabela mostrando a acurácia dos algoritmos testados nesse trabalho

Classe alvo	Média sobre as classes
Tipo de amostragem	validação cruzada de 3 folds
Método	Acurácia
Tree	0.750
Random Forest	0.833
Naive Bayes	0.417

Uma outra questão ser respondida pelo trabalho é se algum algoritmo de classificação pode ser usado para classificar usuários que ainda não tem seu perfil conhecido. Para responder esse questionamento software *Orange* que testa uma série de algoritmos de classificação e verifica as suas respectivas acurácias. Para validar essa parte do trabalho a metodologia de validação cruzada *k-fold* é utilizada. Após a medição dos escores dos algoritmos chegou-se a tabela 1 e que o mais apropriado para fazer a classificação como base na acurácia é o algoritmo de florestas aleatórias (*random forest*).

6.6 Comparação dos objetos feitos por dois usuários.

Figura 16: Atividade feita pelo usuário 1

A tarefa designada para os usuários foi a de criar um polvo a partir de um cilindro. Logo o método utilizado para validar as hipóteses é uma análise qualitativa dos objetos cons-

Figura 17: Atividade feita pelo usuário 9

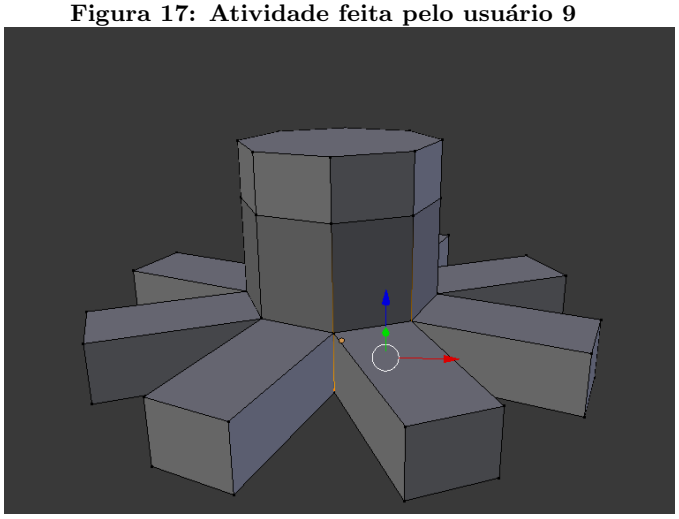
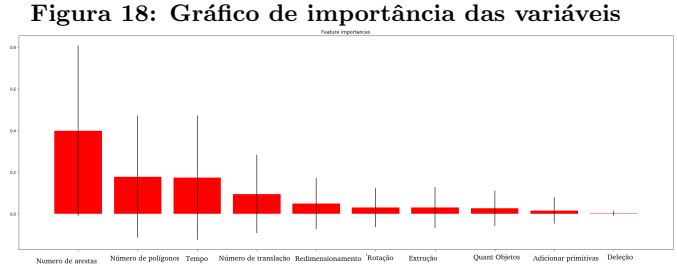


Figura 18: Gráfico de importância das variáveis



truídos pelos mesmos. Essa análise compara o tempo e a quantidade de passos utilizados para a construção do polvo feito por usuário com e sem treinamento de modelagem 3D. Os usuários comparados são os usuário 9 e 1 do dendrograma. Através da figura 17 cujo o usuário é o de número nove que sua geometria apresenta uma incompletude na geometria do modelo 3D, mesmo tendo gasto um tempo de aproximadamente 2236 segundos para fazê-lo. Verifica-se o a completude geométrica do objeto criado pelo usuário um bem representado pela figura 16 como um tempo menor de aproximadamente 937 segundos. Diante dos dados apontados pode-se concluir que o *kmeans* junto com as variáveis propostas são apropriadas para fazer o agrupamento de usuário com base em sua *expertise*.

6.7 Árvore de decisão e a importância das variáveis

A Figura 18 apresenta um gráfico com a importância de cada variável a partir de uma árvore de decisão. Nele as operações primitivas não evidenciam muita relevância com referência ao grupo pertencente de uma determinada tupla de dados. Portanto essa técnica pode ser utilizada para em conjunto de dados diferentes se as mesmas variáveis tem o mesmo nível de importância. Atividades diferentes podem ter níveis de variáveis diferentes. No presente trabalho após a geração da árvore de decisão, pode- se verificar se pode chegar aos mesmos resultados utilizando apenas as três variáveis. Caso essa premissa seja verdadeira pode-se substituir as nove variáveis por apenas três no processo de classificação. A importância de cada variável é um indício de que essas variáveis podem ser dispensadas no processo de análise de dados. Geralmente essas técnicas são utilizadas para diminuir a quantidade de dimensões existentes nos dados.

7. CONCLUSÕES

Diante das análises feitas tanto quantitativas quanto qualitativas é possível comprovar que as variáveis propostas bem como os métodos propostos funcionam bem. Percebeu-se que o algoritmo *kmeans* agrupa os usuários de acordo com a expertise de usuário. Já os métodos de classificação se conseguiu uma acurácia de oitenta por cento. Uma boa acurácia para a classificação dos usuários. Após a validação, as métricas podem ser utilizadas por outros pesquisadores, medir a expertise de usuários ao utilizar uma nova versão de interface e compará-la com a expertise ao utilizar a antiga antiga interface. Assim propor ou não uma mudança definitiva da mesma. Pesquisadores também podem propor novas formas de medição mais eficientes e mais precisas. Outro ponto a ser considerado é utilizar essas métricas desse trabalho em processos seletivos com o objetivo de contratar os usuários mais aptos.

8. Referências

[1] Francisco R. Ortega et al. *Interaction Design for 3D User Interfaces The World of Modern Input Devices for Research,Applications, and Game Development*. New York: CRC Press, 2016.

[2] V. Ayyadevara. *Pro Machine Learning Algorithms*. 1st. Hyderabad, Andhra Pradesh, India: Apress., 2018. ISBN: 978-1-4842-3563-8, 978-1-4842-3564-5.

[3] Doug A. Bowman et al. *3D User Interfaces: Theory and Practice*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2017. ISBN: 978-0-13-403432-4.

[4] Leo Breiman. "Random Forests". Em: *Machine Learning* 45.1 (2001), pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324.

[5] Christopher S. Campbell et al. "Expertise Identification Using Email Communications". Em: *Proceedings of the Twelfth International Conference on Information and Knowledge Management*. CIKM '03. New Orleans, LA, USA: ACM, 2003, pp. 528–531. ISBN: 1-58113-723-0. DOI: 10.1145/956863.956965. URL: http://doi.acm.org/10.1145/956863.956965.

[6] Chris Conlan. *The Blender Python API Precision 3D Modeling and Add-on Development*. Bethesda, Maryland: Apress, 2017.

[7] Rodney X. Sturdivant David W. Hosmer Jr. Stanley Lemeshow. *Applied Logistic Regression*. New Jersey: John Wiley, 2000.

[8] Janez Demšar et al. "Orange: Data Mining Toolbox in Python". Em: *Journal of Machine Learning Research* 14 (2013), pp. 2349–2353. URL: http://jmlr.org/papers/v14/demsar13a.html.

[9] Bhupendra K Singh Dewendra K Singh e Dr Yogendra P Dubey. "Expert Systems and their Application in Library and Information Systems". Em: *DESIDOC Bulletin of Information Technology* (1996).

[10] Hui Fang e ChengXiang Zhai. "Probabilistic Models for Expert Finding". Em: *Advances in Information Retrieval*. Ed. por Giambattista Amati, Claudio Carpineto e Giovanni Romano. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 418–430. ISBN: 978-3-540-71496-5.

[11] Guojun Gan. *Data Clustering in C++: An Object-Oriented Approach*. 1st. Chapman & Hall/CRC, 2011. ISBN: 1439862230, 9781439862230.

[12] Jason van Gumster. *Blender For Dummies*. 3rd. For Dummies, 2015. ISBN: 1119039533, 9781119039532.

[13] Patrick Harms. "Automated Field Usability Evaluation Using Generated Task Trees". Tese de doutorado. University of Göttingen, 2016. URL: http://nbn-resolving.de/urn:nbn:de:gbv:7-11858/00-1735-0000-0028-8684-1-1.

[14] Melody Y. Ivory e Marti A Hearst. "The State of the Art in Automating Usability Evaluation of User Interfaces". Em: *ACM Comput. Surv.* 33.4 (dez. de 2001), pp. 470–516. ISSN: 0360-0300. DOI: 10.1145/503112.503114. URL: http://doi.acm.org/10.1145/503112.503114.

[15] Gareth James et al. *An Introduction to Statistical Learning – with Applications in R*. Vol. 103. Springer Texts in Statistics. New York: Springer, 2013. ISBN: 978-1-4614-7137-0. DOI: 10.1007/DOI.

[16] Linda C. van der Gaag tePer J.F. Lucas. *Principles of Expert Systems*. Addison-Wesley, 1991.

[17] J. MacQueen. "Some methods for classification and analysis of multivariate observations". Em: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif.: University of California Press, 1967, pp. 281–297. URL: https://projecteuclid.org/euclid.bsmsp/1200512992.

[18] Timothy Masters. *Assessing and Improving Prediction and Classification: Theory and Algorithms in C++*. 1st. Berkely, CA, USA: Apress, 2017. ISBN: 1484233352, 9781484233351.

[19] Raquel Dias Mendes. "Inteligência artificial: sistemas especialistas no gerenciamento da informação". Em: *Ibict* (1997).

[20] Gustavo Varanda Paiva. *"Aplicação de um Sistema Especialista para o Diagnóstico em Tempo Real das Condições Limite de Operação em Usinas"NUCLEARES*. Rio de Janeiro: UFRJ/ESCOLA POLITÉCNICA, 2015.

[21] Geon Yong Park. "EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN". Em: 2014.

[22] Guillermo Moncecchi Raúl Garreta. *Learning scikit-learn: Machine Learning in Python*. Birmingham B3 2PB, UK: Packt Publishing, 2013.

[23] I. Rish. *An empirical study of the naive bayes classifier*. Rel. téc. 2001.

[24] Yonglei Tao. "Automated Data Collection for Usability Evaluation in Early Stages of Application Development". Em: *ACACOS 08* (2008).

[25] a enciclopédia livre. Wikipédia. *Assistente do Microsoft Office*. 704/2019. URL: https://pt.wikipedia.org/wiki/Assistente_do_Microsoft_Office.

[26] A enciclopédia livre. Wikipédia. *Elbow method (clustering)*. 2019. URL: https://en.wikipedia.org/wiki/Elbow_method_(clustering).

[27] A enciclopédia livre. Wikipédia. *Gráfico de dispersão*. 2019. URL: https://pt.wikipedia.org/wiki/Gr%C3%A1fico_de_dispers%C3%A3o.

[28] Nong Ye. *Data Mining: Theories, Algorithms, and Examples*. 1st. Boca Raton, FL, USA: CRC Press, Inc., 2013. ISBN: 1439808384, 9781439808382.

[29] Allen Yu, Claire Chung e Aldrin Yim. *Matplotlib 2.X By Example: Multi-dimensional Charts, Graphs, and Plots in Python*. Packt Publishing, 2017. ISBN: 1788295269, 9781788295260.