

**BERNARDO MOREDO ROCCO  
VINÍCIUS ARAUJO DA COSTA**

**DESENVOLVIMENTO DE PÓS PROCESSADOR  
PARA FABRICAÇÃO DE MÓVEIS**

São Paulo  
2022

**BERNARDO MOREDO ROCCO  
VINÍCIUS ARAUJO DA COSTA**

**DESENVOLVIMENTO DE PÓS PROCESSADOR  
PARA FABRICAÇÃO DE MÓVEIS**

Trabalho apresentado à Escola Politécnica  
da Universidade de São Paulo para ob-  
tenção do Título de Engenheiro Mecatrônico.

São Paulo  
2022

**BERNARDO MOREDO ROCCO  
VINÍCIUS ARAUJO DA COSTA**

**DESENVOLVIMENTO DE PÓS PROCESSADOR  
PARA FABRICAÇÃO DE MÓVEIS**

Trabalho apresentado à Escola Politécnica  
da Universidade de São Paulo para ob-  
tenção do Título de Engenheiro Mecatrônico.

Orientador:

Marcos de Sales Guerra Tsuzuki

São Paulo  
2022

Dedicamos este trabalho a nossas famílias e amigos, que representaram os alicerces e suportes necessários durante essa jornada. Em especial nossos pais, que incentivaram e garantiram que pudessemos aproveitar ao máximo as nossas oportunidades.

# AGRADECIMENTOS

Agradecemos, primeiramente, à nossos pais e irmãs, que nos acompanharam e incentivaram durante a graduação e a elaboração deste trabalho. Também agradecemos ao Professor Doutor Marcos de Sales Guerra Tsuzuki, por seu acompanhamento e suporte por todo o projeto.

Agradecemos especialmente à Modú, que confiou em nosso trabalho e financiou este projeto.

Não podemos deixar de agradecer nossos amigos, que nos ajudaram na revisão do texto e nos ouviram ao longo do desenvolvimento.

*Hey, we do what we do best. We improvise, all right?*

– Brian O’Conner

# RESUMO

Dentro da indústria de móveis há um distanciamento para com os avanços tecnológicos. Medições e desenhos de móveis ainda são processos manuais e dependentes de pessoas qualificadas. Com o objetivo de facilitar o processo de construção, precificação e logística de se produzir um móvel, este trabalho desenvolve um back-end e pós processador de móveis. Em parceria com a Modú, empresa do ramo mobiliário, foi construída uma base de dados relacional com informações referentes aos materiais, regras construtivas e dimensionais para alimentar um programa que, ao receber dados de um móvel a ser construído, calcula as peças necessárias e suas dimensões. Assim, foi possível obter, partindo das informações geométricas tridimensionais de um móvel, as chapas de corte otimizadas para o empacotamento de peças e sua lista de ferragens.

**Palavras-Chave** – móveis; otimização; customização em massa; design modular;

# ABSTRACT

Within the furniture industry there is a distancing towards technological advances. Measurements and technical drawings are still manual processes dependent on qualified people. With the goal of assisting the building, pricing and logistics processes of producing furniture, this work develops a back-end and post processor. In partnership with Modú, a furniture company, a relational database was built with information regarding materials, constructive and dimensional rules to feed a program that, when receiving data from a piece of furniture to be built, calculates the necessary boards and their dimensions. Therefore, it was possible to obtain, based on the three-dimensional geometric information of a piece of furniture, the optimized cutting plates for the packaging of boards and their list of hardware.

**Keywords** – furniture; optimization; mass customization; modular design;



# LISTA DE FIGURAS

1	Mapeamento da interface entre cliente e produção para a customização de produtos. . . . .	15
2	Paradigmas de produção manufaturada. . . . .	16
3	Diferenças entre design e produção. . . . .	17
4	Framework de features. . . . .	18
5	Diferentes abordagens de KBx. . . . .	19
6	Interface do Configurador durante o projeto de um móvel. . . . .	20
7	Móvel representado acima no configurador, após sua construção. . . . .	21
8	Estrutura do pós processador. . . . .	22
9	Exemplo da Arquitetura MVC (LEFF; RAYFIELD, 2001). . . . .	23
10	Exemplo de estrutura de uma base de dados relacional. . . . .	24
11	Base de dados construída. . . . .	28
12	Gerenciamento dos materiais pela página de Admin. . . . .	30
13	Exemplo passo-a-passo da interpretação das regras de negócio. . . . .	34
14	Esquema de separação do método Guilhotina (JYLÄNKI, 2010). . . . .	36
15	Exemplo de chapa otimizada de um móvel. . . . .	37
16	Diagrama de resumo do desenvolvimento do pós configurador. . . . .	39
17	Planilha com as especificações do móvel de teste. . . . .	40
18	Lista de Peças gerada para o móvel de teste. . . . .	41
19	Resumo do Pedido com as imagens dos planos de corte. . . . .	41

## LISTA DE TABELAS

1	Objetos carregados por padrão na base de dados. . . . .	29
2	Quantidade de ferragens para cada tipo de placa. . . . .	38

# SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>11</b>
1.1	Motivação . . . . .	11
1.2	Objetivos . . . . .	12
1.3	Estrutura do Texto . . . . .	13
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>14</b>
2.1	Customização em Massa . . . . .	14
2.2	Automação de Processos Produtivos . . . . .	15
2.3	Modú . . . . .	20
<b>3</b>	<b>Ferramental Utilizado</b>	<b>22</b>
3.1	Django . . . . .	22
3.2	Base de Dados Relacional . . . . .	23
3.3	REST API . . . . .	24
<b>4</b>	<b>Desenvolvimento</b>	<b>25</b>
4.1	Premissas, Parâmetros e Requisitos do Projeto . . . . .	26
4.2	Base de Dados . . . . .	27
4.3	Base de Dados Inicial . . . . .	28
4.4	Página de Administrador . . . . .	30
4.5	Integração com o Configurador . . . . .	30
4.6	Processamento dos Dados . . . . .	31
4.7	Regras de Negócio . . . . .	33
4.8	Lista de Peças . . . . .	35
4.9	Plano de Corte . . . . .	35

4.10	Cálculo de Ferragens . . . . .	38
4.11	Resumo . . . . .	38
<b>5</b>	<b>Resultados</b>	<b>40</b>
<b>6</b>	<b>Conclusão</b>	<b>42</b>
6.1	Trabalhos Futuros . . . . .	42
	<b>Referências</b>	<b>44</b>
	<b>Anexo A – JSON para Comunicação com o Configurador</b>	<b>45</b>

# 1 INTRODUÇÃO

Móveis são essenciais para a vida moderna. Desde armários na cozinha até móveis de cabeceira, é necessário possuir algum local para se guardar pertences de forma eficiente e segura. Porém, o processo de se comprar um móvel é muito pouco flexível - ou se recorre a lojas de móveis prontos, ou se paga caro para um arquiteto e/ou marceneiro projetar e fabricar um móvel personalizado à sua casa e necessidade. Nesse contexto surgiu a Modú, uma startup que visa facilitar a compra e fabricação de mobílias sob medida. Ela atende tanto clientes diretos quanto fabricantes artesanais utilizando sua ferramenta digital de configuração de móveis baseado nas dimensões disponíveis - o configurador. Ainda assim, na Modú, a esteira produtiva entre o móvel do configurador e a fabricação em si é um processo manual e pouco otimizado, baseado em planilhas do Excel e procedimentos de copia e cola.

Já é possível encontrar ferramentas no mercado que realizam certos tipos de automação e otimização no processo da construção de móveis mas, por conta das tecnologias e técnicas empregadas na Modú, não há soluções que possam ser integradas com a plataforma existente. Assim, torna-se necessária uma solução própria que atenda os requisitos impostos, além da redução de custos da operação com softwares de terceiros. O projeto desenvolvido nesse trabalho realiza a tradução das informações do configurador 3D da Modú para peças bidimensionais de um móvel prontas para serem produzidas, além de organizar essa informação em um arquivo em formato compatível com o programa CorteCloud, no qual se realiza o pedido das peças atualmente na empresa. Para isso, foi criado um banco de dados para armazenar as informações necessárias e um aplicativo complementar na infraestrutura em Django, já utilizada na Modú.

## 1.1 Motivação

A população brasileira vem se digitalizando a passos largos, saindo de 67% da população em 2018 e atingindo 83% em 2021 segundo o Núcleo de Informação e Coordenação do

Ponto BR. Somado a isso, as restrições de locomoção impostas em 2020 e 2021, fizeram o mercado de e-commerce brasileiro crescer - 27% - atingindo uma marca histórica de R\$168 bilhões em 2021, reportado pelo relatório Webshoppers 45 do NielsenIQ Ebit.

Já o mercado de Móveis movimentou, no Brasil, cerca de R\$52 bilhões em 2020 segundo o Euromonitor, com previsão de chegar a R\$76 bilhões até 2025. Além disso, o mercado é fragmentado, com muitas empresas pequenas. O maior representante dentre os fabricantes - a Coteminas SA - tem apenas 5,7% do mercado. A venda online de móveis, por sua vez, tem uma penetração de apenas 7,1%, enquanto em países como EUA e Reino Unido, os valores se aproximam de 12% e 14%. Todo esse cenário mostra como há um enorme potencial de crescimento para empresas nesse mercado, consolidando ou substituindo os fabricantes de pequeno porte.

Assim, a Modú enxerga extrema necessidade em otimizar seu processo produtivo e ganhar escala em sua operação. Tarefas que hoje são feitas em planilhas e demoram dias, podem ser automatizadas via programação e passarão a ser executadas em minutos.

## 1.2 Objetivos

Este trabalho tem como objetivo desenvolver um pós processador comercial para móveis em madeira. O projeto faz parte de uma plataforma integrada de produção de móveis planejados. Para realizar o desenvolvimento do pós processador em conformidade com o mercado, foi realizada parceria com a empresa Modú, startup do ramo em questão e grande interessada em modernizar o processo de fabricação de móveis.

A empresa possui um configurador 3D de móveis, desenvolvido em parceria com o Laboratório de Geometria Computacional da Universidade de São Paulo, construído por Mizael S. Falheiro. O configurador consiste em um ambiente digital onde os clientes podem montar o seu móvel de acordo com sua necessidade, adequando-o e personalizando-o em tempo real com visualização de sua montagem. O pós processador receberá deste Configurador os parâmetros geométricos do móvel criado e, a partir disso, realizará consultas em uma base de dados unificada carregada com as informações produtivas utilizadas na empresa, com o objetivo de produzir os arquivos necessários para a produção.

Dessa forma, o pós processador posiciona-se como o Knowledge Base System da produção, centralizando as regras de negócios produtivas que realizam a tradução entre o mundo geométrico tridimensional, representado no configurador, e o mundo real bidimensional, expresso na produção.

É esperado que o pós processador prepare arquivos para utilização em softwares comerciais (como o CorteCloud) já utilizados na esteira produtiva da empresa, além de produzir os arquivos de corte das placas de madeira - com a respectiva otimização de material - e a lista de ferragens necessárias para a montagem. Por fim, espera-se agilizar o processo de produção e realizar a integração entre o cliente, personalizando seu móvel no configurador, e o produto final, podendo ser produzido por diversos fornecedores da região.

No processo serão criados:

1. Banco de dados para guardar as informações dos móveis criados pelos clientes para consultas futuras.
2. Método de interpretação das regras de negócio da companhia e traduzir as informações do configurador de móveis nas peças necessárias para montagem, de acordo com a parametrização desejada.
3. Resumo das informações processadas, com fácil acesso para a empresa, contendo a lista de peças, chapas de corte otimizadas e ferragens necessárias.

## 1.3 Estrutura do Texto

Primeiramente no capítulo 2 será feita uma revisão bibliográfica sobre os conceitos que envolvem o projeto, estabelecendo o estado da arte atual. Em seguida, no capítulo 3 serão abordadas as ferramentas utilizadas no desenvolvimento do projeto, seguido do processo de desenvolvimento em si que será explorado no capítulo 4. No capítulo 5 temos os resultados que foram obtidos ao final do projeto e, por fim, as conclusões e próximos desenvolvimentos no capítulo 6.

## 2 REVISÃO BIBLIOGRÁFICA

Antes de iniciar o estudo do pós processador, é importante compreender sobre a literatura na área de móveis customizados e o que já foi (e não foi) desenvolvido. Por ser um assunto específico, não há praticamente literatura técnica de engenharia com relação a móveis, mas é possível encontrar desenvolvimentos na área de customização em massa e de automação de processos produtivos, os princípios teóricos do pós processador que foi desenvolvido.

### 2.1 Customização em Massa

Avanços tecnológicos permitiram uma maior popularidade de softwares de CAD e também uma cadeia produtiva integrada, com isso nasce no mercado uma tendência ao que é chamado de *mass customization*, a customização em massa. Nesse cenário, de acordo com (JOST; SÜSSER, 2020), torna-se crescente a interface entre empresa e consumidor, tornando o papel do segundo crucial na elaboração do produto.

Neste mesmo estudo destaca-se ainda a importância de entender o processo produtivo em duas etapas diferentes: A primeira, onde a empresa constrói uma plataforma padrão com os componentes do produto e a segunda onde o consumidor interage com a plataforma para definir suas preferências de produto.

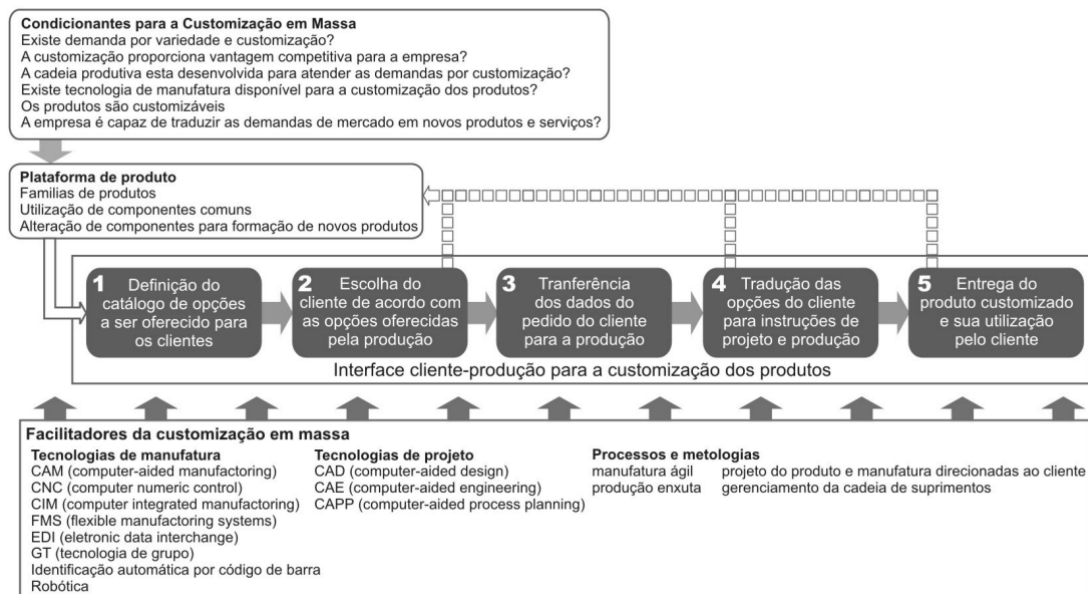
A partir de uma abordagem matemática, (JOST; SÜSSER, 2020) exploram modelos de precificação e demanda quanto ao teor de envolvimento dos consumidores na cadeia produtiva. Este tipo de simulação foge do escopo deste projeto, porém é interessante notar nas suas conclusões em que o melhor cenário possível é aquele onde o produtor oferece o produto personalizado, mas mantém em seu catálogo ofertas padronizadas.

Dentro deste cenário de customização em massa, nasce o *Design for Mass Customization (DFMC)*, onde objetiva-se encontrar economias na produção, complexidade e desenvolvimento do produto. O trabalho desenvolvido em (FETTERMAN; ECHE-



VESTE, 2011) explora o conceito de DFMC dentro do mercado de móveis, explorando as possibilidades pouco exploradas no campo. De início, é importante compreender a estrutura da interface entre cliente e produção demonstrada na figura 1. Nesta interface temos uma plataforma de produtos logo na etapa inicial, que é precedida pelos condicionantes para a customização em massa ser vantajosa. Em seguida temos as etapas seguidas pela interface cliente-produção, partindo da definição das opções a serem oferecidas ao cliente e pela escolha do cliente em si. Na sequência essas informações são transportadas do pedido para a produção, que traduz essas opções em instruções e, por fim, entrega o produto customizado ao cliente.

**Figura 1:** Mapeamento da interface entre cliente e produção para a customização de produtos.



Fonte: (FETTERMANN; ECHEVESTE, 2011)

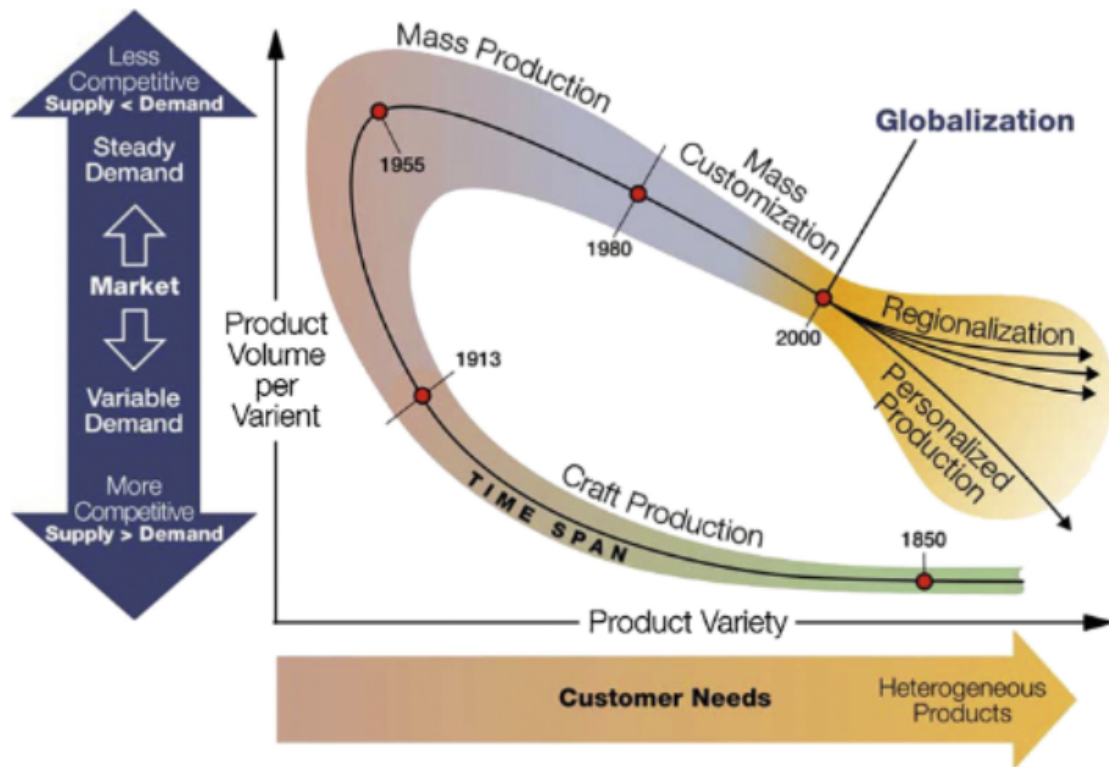
É possível observar que os "Facilitadores da customização em massa" (CAD, CAM, CAE, CNC e etc.) são parte ativa de toda a cadeia da interface cliente-produção. Entretanto, como concluído pelos autores, destaca-se a pouca utilização dos mesmos pela indústria em geral, especialmente na etapa 4, que será o objeto de pesquisa deste trabalho.

## 2.2 Automação de Processos Produtivos

Com a discussão sobre Customização em Massa surge, naturalmente, a necessidade de se automatizar a produção destes novos produtos. Desta forma, destaca-se a origem do termo *Smart Manufacturing* que, como explorado por (LU; XU; WANG, 2020), são operações integradas e colaborativas que respondem a mudanças de demanda e condições

de fábrica usando entendimento dos dados, planejamento e execução de todos aspectos da manufatura, sendo facilitado por técnicas de modelagem e simulação. Dentro do mercado global, destaca-se o Smart Manufacturing como uma tendência da hiper personalização e regionalização de produtos, como na Figura 2.

**Figura 2:** Paradigmas de produção manufaturada.

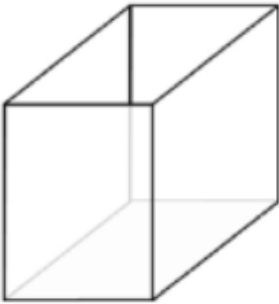
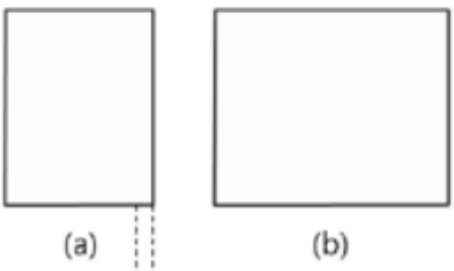
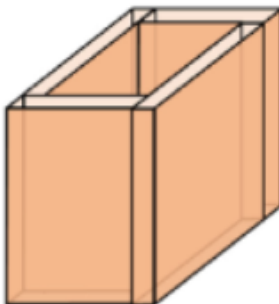
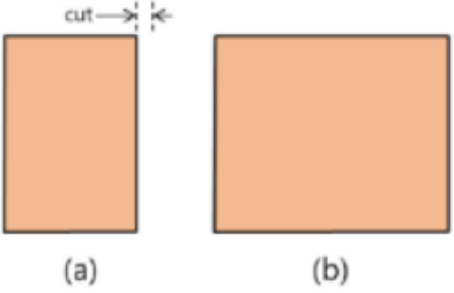


Fonte: (LU; XU; WANG, 2020)

Dentro do contexto explorado pelos autores, destaca-se a importância para o conceito de *Personalized-product-based manufacturing process automation*, onde os autores exploram uma integração e automação total em todos processos, desde o design do produto até sua inspeção. A criação de um Digital Twin do produto é necessária nesse momento, uma vez que há a necessidade de acompanhar todas etapas e informações da produção.

Explorando um problema próximo ao de móveis, o trabalho de (SON et al., 2020) consiste em automatizar a produção de cascos de navios, uma vez que a tradução dos modelos CAD/CAM existentes não é suficiente para a manufatura. Dentro do problema náutico explorado pelos autores, há o desafio de evitar que partes de material se sobreponham, como mostrado na Figura 3, é observada semelhante complexidade quando discutida a modelagem de móveis.

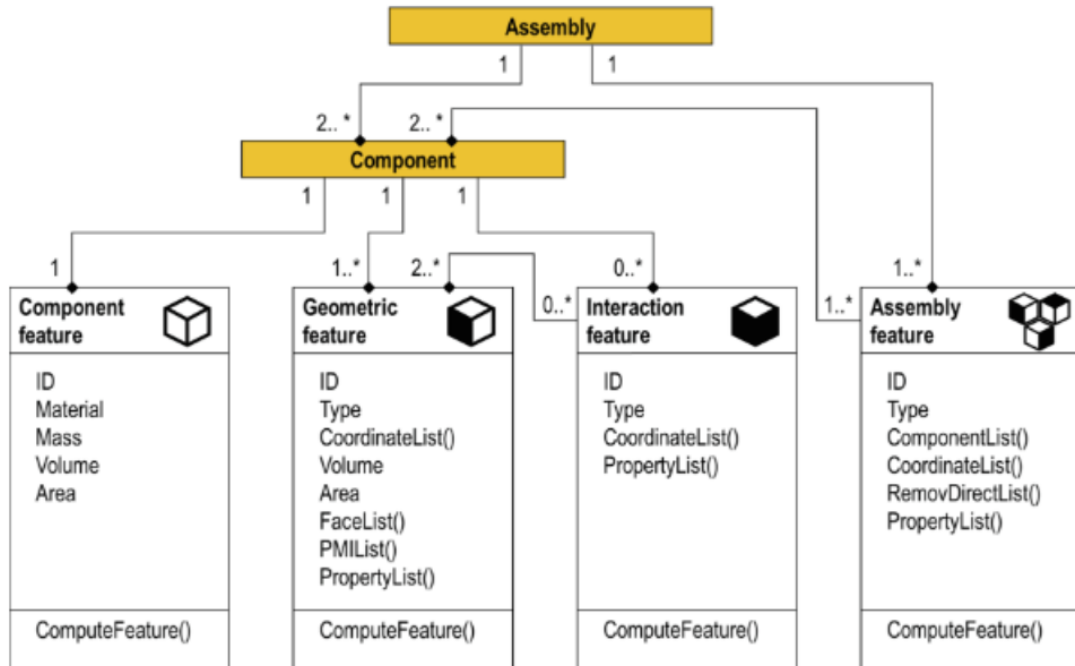
**Figura 3:** Diferenças entre design e produção.

	Assembled Square Pillar	Pieces of Square Pillar
Design Information		 (a) (b)
Manufacturing information		 (a) (b)

Fonte: (SON et al., 2020)

Além disso, a formalização proposta por (FAVI et al., 2022) traz a tona uma interessante discussão quanto ao processo de desenvolvimento proposto. A formalização propõe uma separação da parte fabricada em features, como mostrado na Figura 4:

**Figura 4:** Framework de features.



Fonte: (FAVI et al., 2022)

Na concepção dos autores, uma montagem (Assembly) consiste de diversos componentes (partes mecânicas, como um parafuso ou pistão), que por si tem características geométricas (como furos e ranhuras) e características de interação (como a distância entre furos). Diversos componentes se relacionam através das features de montagem (que relacionam um pistão com o virabrequim). Dessa forma, uma forma complexa pode ser modularizada em diversas partes, possibilitando um entendimento computacional mais acurado da peça a ser fabricada.

Na concepção do mercado de móveis, vale notar que há o conceito de modularização usado pela Modú. Neste teor, os móveis construídos pelo configurador (FALHEIRO et al., 2022) constituem uma estrutura integrada, que será explorada ao longo deste trabalho.

O projeto de (ORTNER-PICHLER; LANDSCHÜTZER, 2022) explora a integração via web de modelagem (CAD/CAM) através da metodologia *Knowledge Based Engineering (KBE)*. É interessante notar as diferentes abordagens *Knowledge Based* na Figura 5:

**Figura 5:** Diferentes abordagens de KBx.

KBx			
Knowledge-based engineering approaches at different detail design levels			
	<b>KBE</b> Knowledge Based Engineering	<b>KBSD</b> Knowledge Based System Design	<b>KBL</b> Knowledge Based Layouting
<b>scope of automated engineering</b>	components, parts, machines	machines and systems	systems
<b>functions</b>	full automated (detail) design of parts and subassemblies	full automated master and layout design of assemblies and systems, specification of machinery	full automated layouting of systems, specification of systems
<b>use for</b>	- customizing machinery - tailored products - product families	- dimensioning motors - defining interfaces - CAD top-down design - CAE models (structural, dyn.)	- space requirements - early cost estimation (bidding) - drafting bill of material
<b>CAD domain</b>	detail geometry models	reduced geometry for CAE	shrink wrap geometry for layout
<b>data, information and knowledge sources</b> (beside employee know how and workflows)	- standards, best practice - production facilities - manufacturer data <b>engineering theory</b>	- standards, best practice - supplier and engine data <b>eng. and mechanics theory</b>	- standards, best practice - manufacturer database - customer rel. management <b>logistics theory</b>
<b>2.0 knowledge sources</b>	web services, online databases, knowledge management systems		
<b>2.0 technologies</b>	wikis, semantics		
<b>practical applic.</b>	tagged standards and technical docs, relation matrices		

Fonte: (ORTNER-PICHLER; LANDSCHÜTZER, 2022)

Apesar da proposta dos autores ser integrar um ambiente web com uma plataforma de CAD (diferente deste trabalho, que busca usar a web para juntar o CAD a produção), são utilizados 5 conceitos chave a partir do KBE:

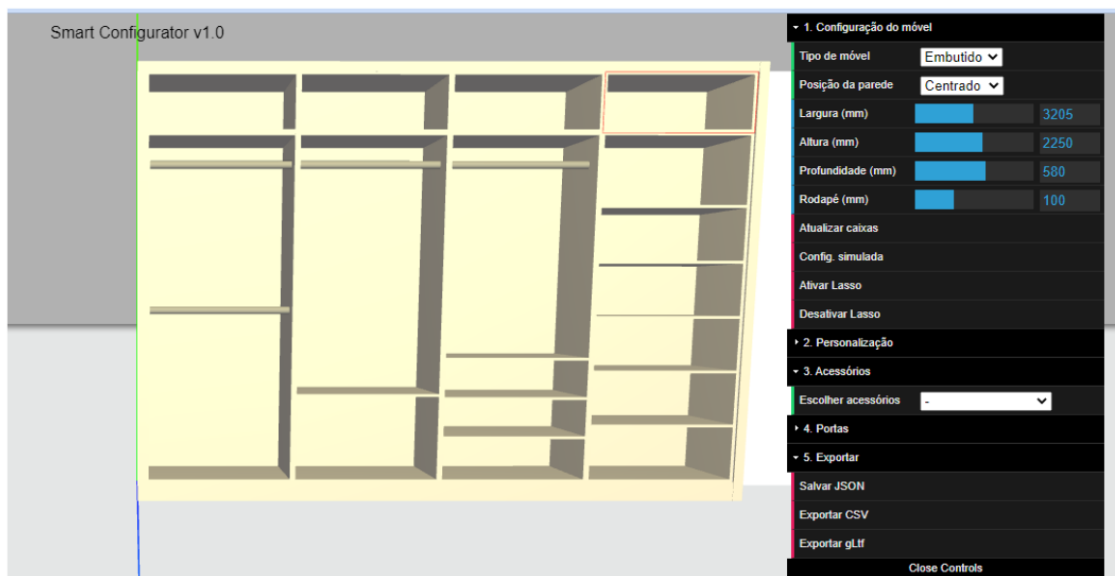
1. **Export e re-import:** Facilitar o tráfego de informações entre os sistemas.
2. **Encapsulamento de caixa preta:** Modularização do objeto construído, de maneira a permitir que cada parte seja editável por si.
3. **Conexão de interface:** Dar a possibilidade do usuário editar os parâmetros das caixas pretas.
4. **Estrutura orientada a objeto:** Há uma hierarquia em todos objetos referentes às caixas pretas.
5. **Agrupamento:** É possível criar e agrupar caixas pretas.

Tomando algumas liberdades sobre os conceitos, pode-se observar facilmente um paralelo com os móveis criados pelo configurador. O conceito de caixa preta utilizado pelos autores aproxima-se do conceito de caixa utilizado na modularização dos móveis da Modú, que será explorado mais a frente. Dessa forma, observando o sucesso dos autores, espera-se que o pós processador siga um caminho semelhante.

## 2.3 Modú

A Modú é uma startup criada em 2020 por Enrico Martins, ex-aluno da Escola Politécnica da USP. A ideia central por trás da empresa é democratizar o design e construção de móveis personalizados, de forma prática e sem exigir conhecimentos técnicos. A principal ferramenta, desenvolvida por eles, que os possibilita cumprir esse objetivo é seu configurador. Ele foi desenvolvido inteiramente pela empresa e torna um fácil um trabalho que é muitas vezes realizado manualmente ou em programas de CAD complexos e inacessíveis para o público geral. Além disso, a Modú também oferece seu programa para marceneiros e arquitetos que queiram introduzi-lo no seu cotidiano com os clientes como um grande diferencial no atendimento. O configurador, é um ambiente online hospedado em um site onde o cliente pode acompanhar e realizar a construção do seu próprio móvel. O móvel é apresentado em 3D e o programa tem uma interface intuitiva que possibilita a customização dos móveis. Na figura 6 pode-se observar o configurador durante a criação de um móvel e na figura 7 o móvel resultante dessa criação.

**Figura 6:** Interface do Configurador durante o projeto de um móvel.



Fonte: Modú

**Figura 7:** Móvel representado acima no configurador, após sua construção.



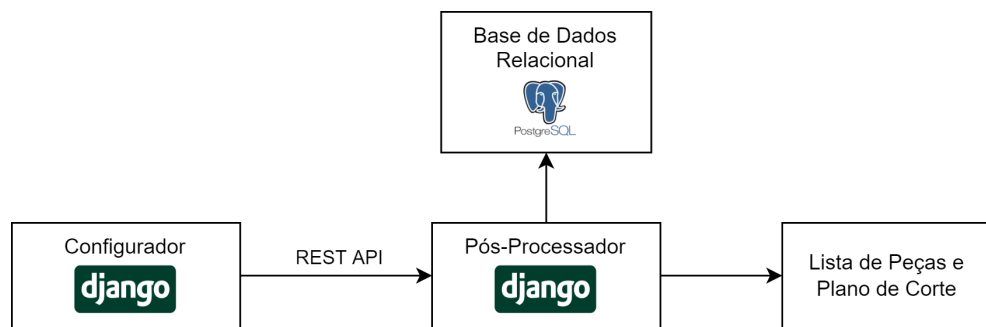
Fonte: Modú

Dentro da Modú se utiliza uma nomenclatura específica para cada tipo de peça utilizada nos móveis que são construídos. Para total compreensão desse texto, é importante que o leitor esteja familiarizado com os conceitos da área de trabalho da empresa. A Modú trabalha com móveis modulares, onde cada módulo tem sua função específica e a junção desses módulos, com diferentes tamanhos e utilidades, formam o móvel final. Na mesma hierarquia que as caixas temos algumas peças de acabamento como as máscaras - peças que envolvem as caixas e dão o aspecto uniforme ao móvel e arremates - acabamentos para preencher o espaço entre o móvel e o chão/teto, além das portas do móvel.

As caixas e todas as outras peças não metálicas são compostas, por sua vez, de placas de madeira. Cada placa tem sua especificação de tamanho, tipo de material, e de fita de borda - fita vinílica que é colada na espessura da madeira para esconder o miolo do material e retirar uma aparência crua do móvel.

### 3 FERRAMENTAL UTILIZADO

Ao se trabalhar com uma empresa parceira no desenvolvimento de um projeto, é necessário se adaptar a realidade da infraestrutura onde o projeto será instalado e construído. Na Modú, empresa parceira para o desenvolvimento do pós processador, já havia uma estrutura baseada em Django para o configurador onde o pós processador se conecta. Dessa forma, optou-se por utilizar a estrutura representada no diagrama da figura 8 no projeto, simplificando a comunicação com as soluções já existentes:



**Figura 8:** Estrutura do pós processador.

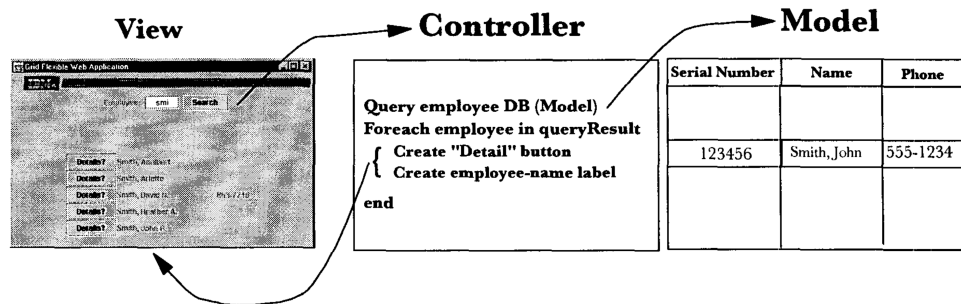
Nela, temos uma interface em REST API que realiza a comunicação com o configurador e que passa as informações para a aplicação em Django. Essa aplicação se comunica com uma base de dados relacional para consultar as regras de negócio e salvar as informações de cada móvel e na sequência processa e exporta a lista de peças em Excel e o plano de corte.

#### 3.1 Django

Django é uma biblioteca para Python com foco em desenvolvimento web que utiliza uma arquitetura *model-view-controller* (MVC). Nesse modelo o usuário interage com uma interface (*view*), visualizando e editando informações. Esta se comunica com o controlador (*controller*), que é responsável por fazer a ponte com o a base de dados (*model*). O



controlador realiza as *queries* necessárias para ler e editar as informações na base, trata essas informações e aplica a lógica do programa, e, por fim, gera a nova visualização para o usuário(LEFF; RAYFIELD, 2001). A figura 9 ilustra o funcionamento da arquitetura MVC:

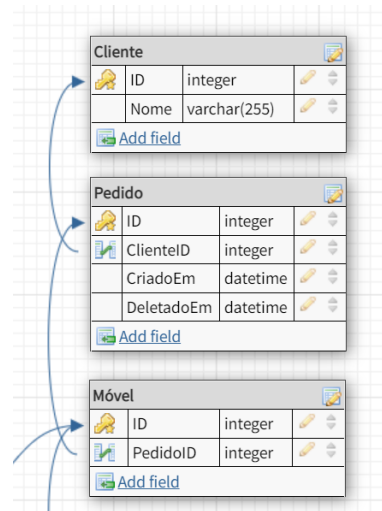


**Figura 9:** Exemplo da Arquitetura MVC (LEFF; RAYFIELD, 2001).

O Django também conta com uma estrutura de aplicativos que podem ser criados e utilizados de forma independente ao projeto principal.

## 3.2 Base de Dados Relacional

O modelo relacional de base de dados é uma das estruturas de dados mais utilizadas e isso se dá por conta de sua facilidade de uso e compreensão. Um modelo relacional organiza as informações não em matrizes com posições numeradas e com endereços definidos, mas sim em tabelas, como as demonstradas na figura 10. Todas as tabelas no modelo relacional são nomeadas e tem suas colunas também com nomes próprios. Para acessarmos alguma informação, informa-se qual a relação que será acessada, qual a chave primária nessa relação e qual o atributo desejado. Além disso, o modelo relacional é forte pelas relações que se constroem entre as tabelas, com chaves relacionando as entradas e garantindo uma base consistente. Esse modelo de base de dados garante dados íntegros, identificáveis e de fácil acesso(CODD, 1989).



**Figura 10:** Exemplo de estrutura de uma base de dados relacional.

### 3.3 REST API

Dentre as estruturas de APIs existentes, uma das mais comuns é a REST. Criada na tese de doutorado de R. Fielding e R. Taylor (FIELDING; TAYLOR, 2000), REST significa *Representational State Transfer* (Transferência Representacional de Estado), e é uma estrutura voltada a transmissão de conteúdos na web. Seu principal benefício é sua completude, onde cada interação contém toda a informação necessária para seu processamento. Isso facilita a programação e diminui os recursos necessários para uma comunicação eficiente.

## 4 DESENVOLVIMENTO

O desenvolvimento do projeto se iniciou pela definição das premissas e requisitos do programa em conjunto da empresa, seguido pela estruturação da base de dados - etapa essencial para o desenvolvimento das demais etapas - e pela criação de uma base de dados inicial. Na sequência, foram implementados o interpretador de regras de negócio e processamento de dados, além das interfaces de administração e de integração com o configurador. Por fim, foram criadas as funções que geram a lista de peças dos móveis, os planos de corte e o cálculo de ferragens.

Dessa forma, foram realizadas releases semanais, onde foram testados móveis no programa com complexidade crescente, adicionando e ajustando os pontos necessários. Este tempo entre releases permitiu que a empresa traduzisse de maneira programática e lógica os processos realizados, compreendendo ao todo a extensão de atuação do pós processador.

### 1. Primeira Release

Foram carregados 3 móveis simples, formados de caixas completas com prateleiras, portas e máscaras. Alternavam-se materiais e dimensões, com o objetivo de validar as regras de negócios e as definições de cada placa.

### 2. Segunda Release

Foi carregado um móvel formado de caixas de teto reduzido, para validar as novas regras de negócios para um Tipo de Caixa diferente.

### 3. Terceira Release

O novo móvel contava com uma caixa sem fundo e novos acabamentos, do tipo Arremate. Aqui, validou-se a possibilidade de adicionar novos acabamentos nos móveis e mais um tipo de caixa.

### 4. Quarta Release

Aqui o desafio foi adicionar um novo tipo de caixa, com cálculos diferentes das demais, o Pé Esquadro, além do acabamento do Rodapé T.

### 5. Quinta Release

Foi adicionado ao móvel Gavetas Externas, que agem como caixas dentro da caixa. Foi necessário um novo desenvolvimento para suportar essa nova arquitetura

### 6. Sexta Release

Implementaram-se Gavetas Internas, que se comportam como gavetas externas a uma caixa auxiliar interna do móvel.

### 7. Sétima Release

Criada uma nova *endpoint* para realizar o cálculo e otimização das placas para corte, resumo do pedido e contagem de ferragens.

Após de cada release todos os pontos de desenvolvimento foram revisitados, garantindo pleno funcionamento do programa e adequação aos requisitos.

## 4.1 Premissas, Parâmetros e Requisitos do Projeto

É importante notar que, tratando-se de uma parceria empresa-universidade, os requisitos do projeto foram se modificando ao longo do projeto, de forma que se adequavam às necessidades de desenvolvimento. Assim, apesar de ter sido a primeira etapa realizada, os requisitos foram constantemente revisitados e revisados pelos autores e pela empresa, adequando-se ao funcionamento do programa.

O projeto foi implementado em um projeto já existente, e com isso integra-se facilmente às soluções já adotadas na Modú. Além disso, assume-se que todas as informações necessárias para o funcionamento serão geradas previamente pelo configurador ou inseridas pela empresa. A implementação mais comum de uma API em REST utiliza o protocolo HTTP para realizar as consultas e alterações na base de dados, com um endereço para cada informação. Com as premissas bem definidas, partiu-se para a definição dos seguintes parâmetros e requisitos:

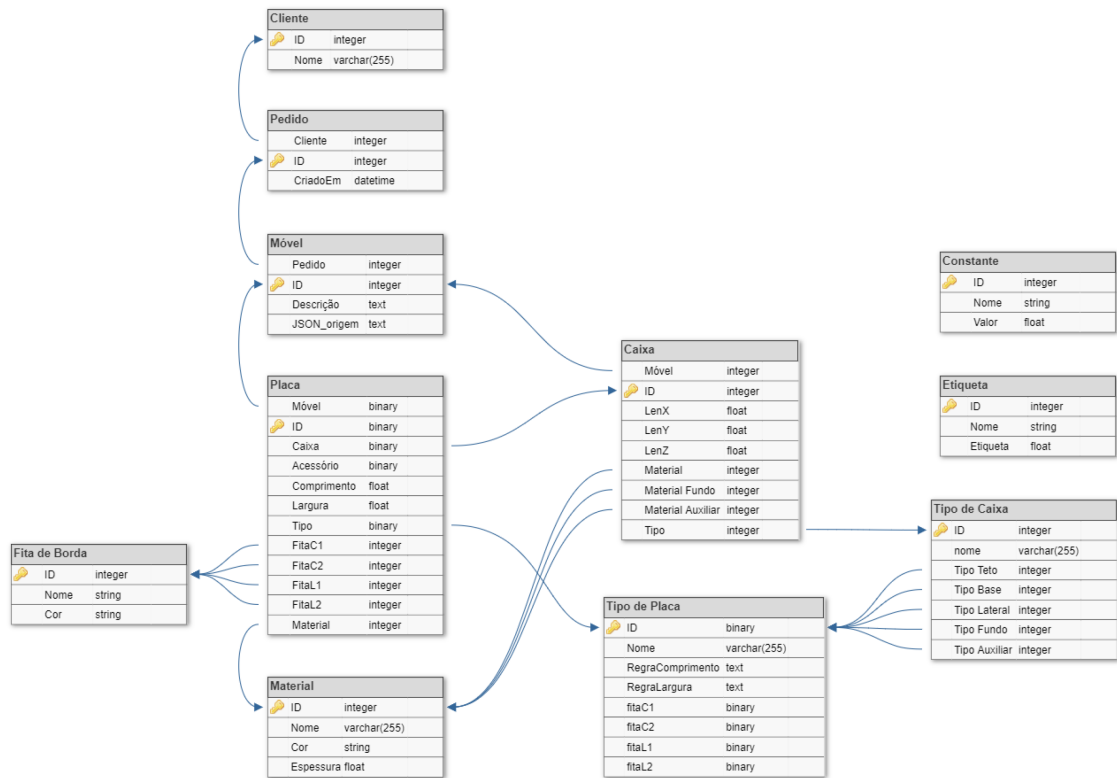
- Se comunicar com o configurador da Modú;
- Ser escalável;
- Ter fácil manutenção sem a necessidade de conhecimentos de programação;
- Gerar as placas para qualquer configuração possível;
- Gerar chapas de corte otimizadas para um móvel;

- Identificar as placas de acordo com sua posição no móvel;
- Facilidade de inserir novos tipo de móveis.

É possível observar que se optou por não estabelecer grandes restrições de abordagem para solução do problema para justamente estimular a busca de caminhos inexplorados para o resultado desejado. Além disso, foi necessário, ao longo do desenvolvimento, revisitar os requisitos e parâmetros estabelecidos, adequando-os às necessidades funcionais e produtivas implementadas.

## 4.2 Base de Dados

Iniciar o desenvolvimento pela base de dados foi uma decisão acertada para o projeto. O processo de construção dessa base foi extremamente importante para a compreensão das regras de negócio da Modú e o funcionamento geral da empresa e do fluxo que seria construído nas próximas etapas. O ambiente de desenvolvimento escolhido para a base foi o POSTGRESql, dada a completude da ferramenta e também por ser o programa utilizado onde foi hospedada a ferramenta. A partir das reuniões semanais que ocorreram para compreensão da estrutura necessária para atender as necessidade dos clientes, pode-se observar na figura 11 a estrutura final da base de dados.



**Figura 11:** Base de dados construída.

É possível observar que a estrutura da base é em sua maioria linear, devido à hierarquia das informações com as quais estão sendo trabalhadas. Na estrutura construída os dados começam a partir das informações do usuário. Este, por sua vez, pode ter  $N$  móveis, que são compostos por  $N$  caixas. Estas, por fim, são formadas de  $N$  placas. Adicionalmente, temos tabelas auxiliares que armazenam as opções existentes no configurador como material da placa, tipo da fita de borda, tipo da caixa. Além disso, para a implementação das regras de negócio, temos também uma tabela contendo os tipos das placas, onde são armazenadas as regras de negócio que definem as dimensões da placa.

### 4.3 Base de Dados Inicial

Visando facilitar os testes e também a configuração da base de dados em caso de perda, foi programado uma função que carrega as opções essenciais na base de dados do pós processador. Esta função é importante para garantir um padrão do sistema e demonstrar o seu funcionamento. Assim, é realizada a criação dos objetos apresentados na tabela 1 na base de dados, seguindo instruções da Modú:

Modelo	Dados carregados	Modelo	Dados carregados
Tipo de caixa	Caixa Completa Caixa Teto Reduzido Caixa sem fundo Pé Esquadro Gaveta Externa Caixa Gaveteiro Tipo 1 Caixa Gaveteiro Tipo 2 Caixa Gaveteiro Tipo 3	Material	Branco tx 25 Branco tx 18 Branco tx 15 Branco tx 6 Preto tx 25 Preto tx 18 Preto tx 15 Preto tx 6 Freijó tx 18 Freijó tx 25 Cinza sagrado essencial 18
	Porta Máscara Vertical Máscara Horizontal Teto Caixa Completa Base Caixa Completa Lateral Caixa Completa Fundo Caixa Completa Prateleira Caixa Completa Prateleira Estendida Teto Caixa Teto Reduzido Arremate Direito Arremate Esquerdo Arremate Superior Rodapé T Frente Pé Esquadro Lateral Pé Esquadro Frente/Traseira Gaveta Frontão Gaveta Externa Lateral Gaveta Externa Fundo Gaveta Externa Teto Base Gaveteiro 1 Teto Base Gaveteiro 2 Teto Base Gaveteiro 3 Lateral Gaveteiro Arremate Interno		prat_folga_fr ajuste_prof_cx arred_tam_int esp_fundo_cx rebaixo_fundo recuo_fundo prof_teto_r largura_arrmt recuo_fr_pe recuo_tras_pe largura_rodape dif_lat_gav folga_fro_cx recuo_gav_cx esp_corred_tel frente_arrmt_int recuo_cx_cx_gaveteiro largura_arrmt_int
Fitas de Borda	Branco tx Preto tx Freijó tx Cinza sagrado essencial	Etiquetas	Caixa Completa Máscara Vertical Máscara Horizontal Teto Caixa Teto Reduzido Lateral Prateleira Arremate Direito Arremate Esquerdo Arremate Superior
⋮	⋮		

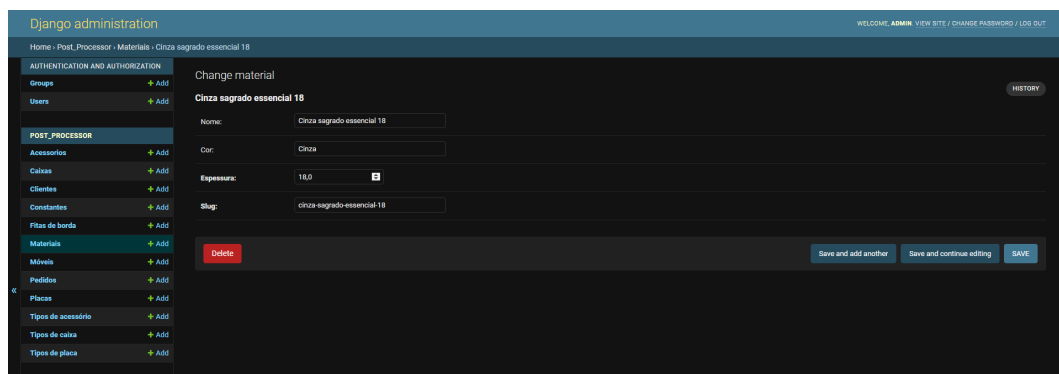
**Tabela 1:** Objetos carregados por padrão na base de dados.

Os dados a serem carregados foram construídos em conjunto com a Modú à medida

que o pós processador foi evoluindo.

## 4.4 Página de Administrador

Para realizar o gerenciamento da base de dados, foi utilizado o pacote `Django-admin`, que cria uma página web onde é possível visualizar todas as tabelas e realizar todas as operações CRUD (Create, Read, Update e Delete) na base de dados. A figura 12 apresenta a tela padrão de edição de um material através da página de administrador. Todas as tabelas tem telas semelhantes para edição de suas entradas de forma acessível a pessoas sem conhecimento de programação e SQL.



**Figura 12:** Gerenciamento dos materiais pela página de Admin.

## 4.5 Integração com o Configurador

A integração do projeto com o configurador em sua versão de produção ocorreu após o desenvolvimento de todas as funções descritas acima e de uma bateria de testes que serão explorados no próximo capítulo. Essa demanda de manter separado o desenvolvimento do pós processador do resto do projeto se deu para preservar a funcionalidade do configurador em caso de erros inesperados, mas também pelo fato do próprio configurador estar em desenvolvimento pela companhia. Com isso, foi realizado um trabalho em paralelo para adaptar o configurador ao novo pós processador e, quando ambas as partes foram concluídas, foi feito o trabalho de integração.

Para realizar essa comunicação entre módulos, optou-se por utilizar um JSON padronizado - exposto em no anexo A, que segue a estrutura abaixo:

- **Variáveis** - Chave contendo os valores a serem considerados para as constantes construtivas, caso sejam diferentes do padrão;



- **Cliente** - Chave com as informações do cliente;
- **Movel** - Chave contendo as informações do móvel a ser processado;
- **Portas** - Lista com as portas presentes no móvel e suas características;
- **Acabamentos** - Lista com os acabamentos presentes no móvel (arremates, rodapés e etc.) e suas características;
- **Caixas** - Lista de todas as caixas que compõe o móvel, com suas características, nome de referência a ser utilizado e número de prateleiras, caso tenha alguma. Caixas também tem duas chaves dentro delas:
  - **Caixa gaveteiro** - Chave com as informações de tipo e medida específicas para caixas gaveteiro;
  - **Gavetas** - Chave com as características das gavetas presentes na caixa, se houver alguma.

As características de portas, acabamentos, caixas e gavetas seguem a estrutura padrão abaixo:

- **Tipo** - Representa o tipo do objeto, dentro de sua categoria, seguindo as opções disponíveis na base de dados;
- **Len (X, Y e Z)** - Dimensões do objeto nas coordenadas do plano cartesiano padrão de cada móvel;
- **Material** - Material da estrutura, seguindo o carregado em base, com diferenciação em espessura e cor;
- **Quantidade** - Quantidade de objetos iguais, se aplicável;
- **Fita de borda** - Cor da fita de borda a ser aplicada no material, nas faces onde é possível aplicá-las e seguindo a mesma padronização do banco de dados.

## 4.6 Processamento dos Dados

Com a base de dados previamente carregada com os materiais, fitas de borda, tipos de placa, tipos de caixa e constantes é esperado que o programa receba dados do Front com as informações sobre o móvel a ser produzido e, a partir de ambos dados, processe as

informações construtivas deste móvel. Para a passagem de dados é recebido um objeto do tipo `JSON`, onde há os dados separados nas seguintes categorias: Constantes, Cliente, Móvel, Portas, Acabamentos, Caixas. Constantes contém informações sobre personalizações únicas para o projeto em questão, Cliente e Móvel contém informações sobre a identificação do cliente, pedido e móvel, já os demais contém as informações sobre o material, dimensões e tipos das peças que compõe o móvel identificado.

Uma vez que as informações são recebidas, o programa realiza as seguintes etapas com o objetivo de criar os objetos relativos na base de dados e calcular suas respectivas dimensões:

1. Ler as seções do `JSON`.
2. Carregar as constantes, tipos de placa, tipos de caixa, materiais e fitas de borda configuradas.
3. Identificar o cliente que realizou o pedido.
4. Identificar o pedido ao qual o móvel faz parte - e criá-lo, se não existir.
5. Identificar o móvel - e criá-lo, se não existir.
6. Iterar pelos acabamentos e pelas portas, criando eles associados ao móvel e seguindo os tipos recebidos.
7. Iterar pelas caixas, criando eles associados ao móvel e seguindo os tipos recebidos.
  - (a) Criar as duas laterais da caixa, seguindo o tipo de lateral do tipo de caixa informada.
  - (b) Criar o teto e base da caixa, seguindo o mesmo formato.
  - (c) Criar o fundo da caixa, se houver.
  - (d) Criar as prateleiras, se houver.
  - (e) Criar as gavetas, se houver - para cada gaveta é criada uma nova caixa e estes itens se repetem.

Ao final destes passos estão carregados em base Caixas e Placas, com dimensões adequadas calculadas de acordo com as regras de negócios e prontas para serem manipuladas.

## 4.7 Regras de Negócio

Para a implementação das regras de negócio no pós processador, a maior dificuldade a ser superada era desenvolver uma solução que possibilitasse a manutenção sem a necessidade de se alterar o código do programa, dado que será realizada por um time que não necessariamente terá experiência de programação. Para isso, foi criada uma solução baseada na função `eval` do `Python`. Assim, as regras de negócio (de comprimento e de largura) para cada tipo de placa é salva na base de dados e pode ser alterada facilmente pela página de administrador da ferramenta e posteriormente interpretada pelo programa. Seus cálculos literais são realizados usando variáveis que serão recebidas do configurador e constantes construtivas carregadas no programa de forma dinâmica.

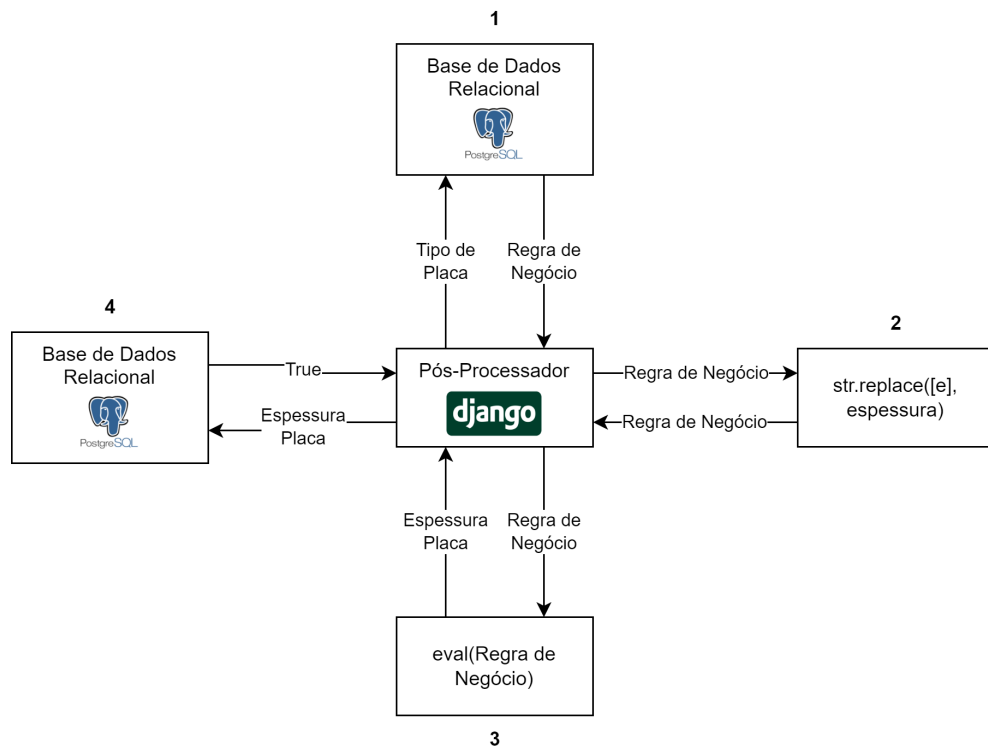
Assim, para a leitura das regras de negócios, foram definidas as seguintes variáveis:

- `[l]` é a largura do objeto pai;
- `[a]` é a altura do objeto pai;
- `[p]` é a profundidade do objeto pai;
- `[e]` é a espessura do material do objeto pai;
- `[ef]` é a espessura do fundo da caixa, se houver;
- `[ec]` é a espessura da caixa, para gavetas.

A partir dessas regras, que foram montadas de forma genérica, o código foi configurado pra ler as strings de acordo com cada tipo de placa, substituir os caracteres que indicam alguma medida dos outros objetos pelo valor correspondente e depois interpretar essa *string* utilizando o `eval`, obtendo a medida correta. A figura 13 mostra as etapas do processamento dessas regras junto com o fluxo de informações existente. Em resumo, as etapas de interpretação são as seguintes:

1. O pós processador consulta a base de dados para resgatar, de acordo com o tipo de placa, qual a regra de negócio que será utilizada.
2. É feita a interpretação da regra substituindo as variáveis definidas em conjunto com a Modú pelos valores respectivos do objeto pai. Nesse caso estaremos substituindo uma espessura, representada por `[e]`.

3. Aplicamos a operação `eval` do Python para interpretar a string como instruções do Python, e realizar a conta desejada.
4. Após o processo se repetir para todas as dimensões da placa, o pós processador salva as informações da mesma na base de dados relacional.



**Figura 13:** Exemplo passo-a-passo da interpretação das regras de negócio.

A partir desses passo-a-passo é possível observar como é flexível a estrutura proposta e também a posição central da base de dados relacional no funcionamento do pós processador. De maneira ilustrativa, o processo para o cálculo do comprimento de uma placa do tipo Base Caixa Completa em uma caixa de dimensões 200x200x190mm e material de espessura 15mm segue os seguintes cálculos:

1. Regra de Comprimento de Placa do tipo Base Caixa Completa, recuperada da base `[1] - [e] * 2 - arred_tam_int`
2. Substituindo `[1]` e `[e]` pelas medidas da caixa recebida no JSON `200 - 15 * 2 - arred_tam_int`
3. Substituindo `arred_tam_int` pelo seu valor correspondente (constante configurada

na base)

$200-15*2-1$

4. Realizando o `eval()` da sentença

169

## 4.8 Lista de Peças

Uma vez com o móvel completamente separado em suas placas, é necessário gerar a lista de peças que o compõe. Para isso, utilizando a estrutura de base de dados com chaves externas para validação e links diretos entre placas, caixas e móveis, a tarefa de se gerar a lista se torna trivial. Abaixo segue a lógica do algoritmo gerador da lista:

1. Selecionar todas as placas do móvel desejado.
2. Exportar as linhas da base de dados para um `DataFrame`.
3. Substituir os IDs das fita de borda, móvel, material e função pelos nomes correspondentes.
4. Adequar os nomes dos itens as funções utilizando as Etiquetas.
5. Exportar o `DataFrame` ajustado para um arquivo em Excel.

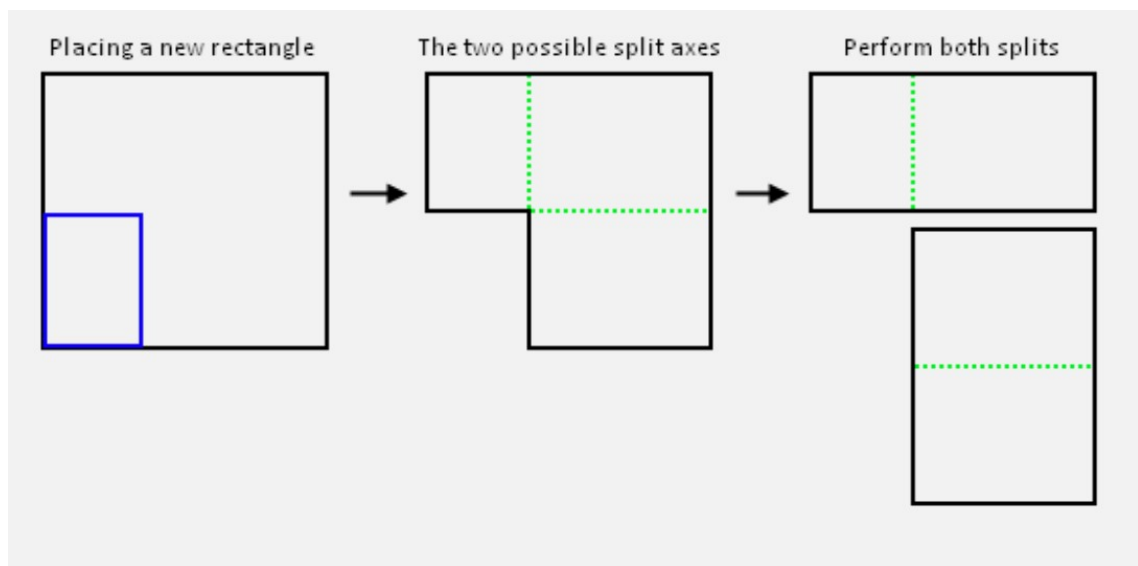
## 4.9 Plano de Corte

Para realizar a diagramação do plano de corte foi realizada a escolha de uma biblioteca em *Python* que cumprisse com os requisitos do problema. Neste caso, não foi considerada a necessidade de desenvolver o código frente a grande gama de soluções já disponíveis. Dessa forma, foi escolhida para trabalho a biblioteca `rectpack` (Disponível no GitHub). Com ela é possível realizar o empacotamento 2D de diversas placas dentro das chapas de madeira disponíveis comercialmente, realizando a otimização das áreas de material e da quantidade de cortes.

Foi escolhido o método Guilhotina (`GuillotineBlsfMinas`) para realizar a solução do problema, com o objetivo de minimizar os cortes realizados na madeira. O método é explicado por (JYLäNKI, 2010) mas, de maneira simplificada, segue o seguinte procedimento - exemplificado também na figura 14:

1. Posicionar um retângulo livre no canto da chapa.
2. Separar o "L" restante de acordo com a regra definida (horizontal ou vertical).
3. Posicionar um retângulo na nova chapa.

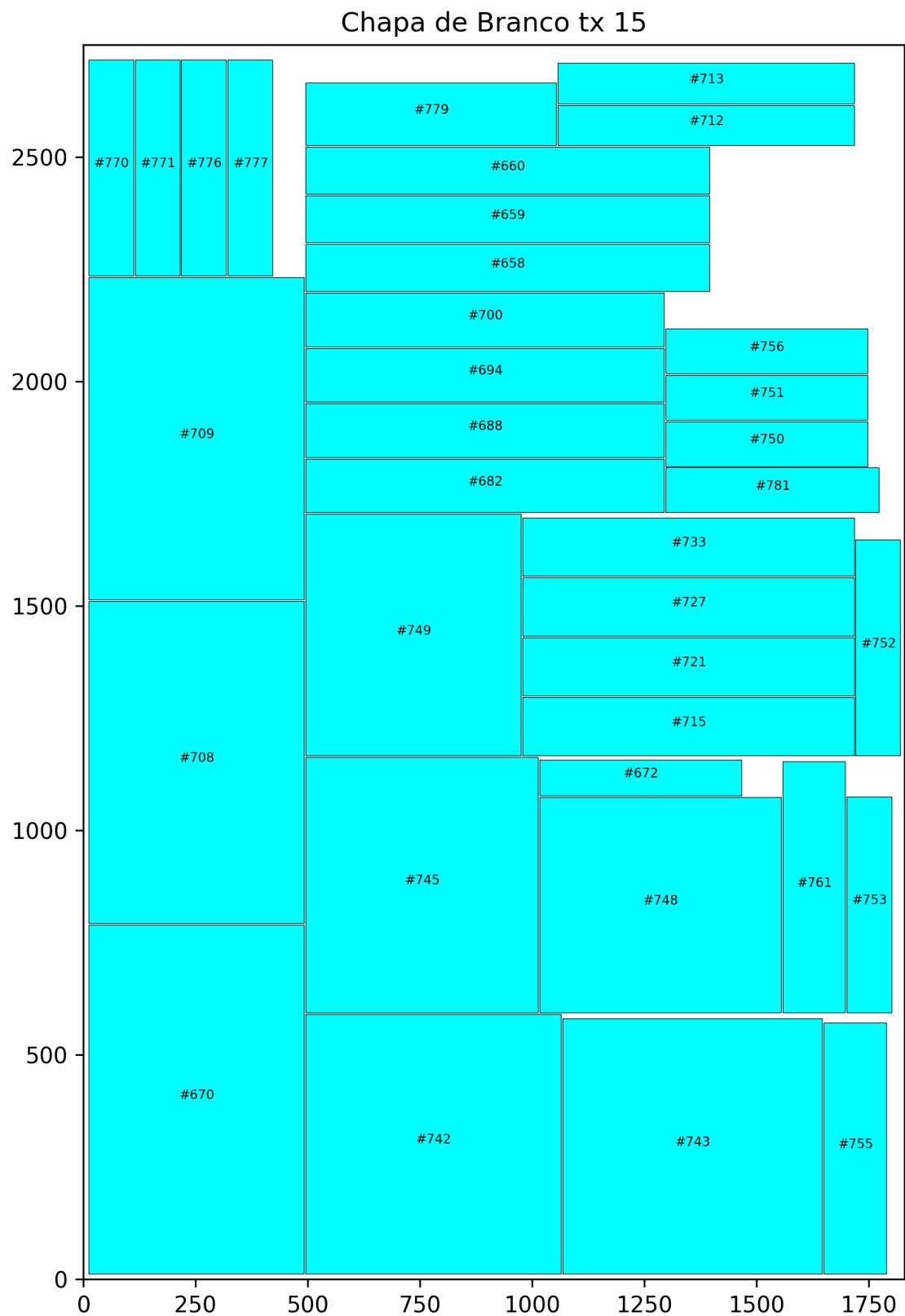
Aqui é importante notar duas regras modificadoras aplicadas, *Best Long Side Fit* (BLSF), que minimiza a sobra do maior lado do retângulo e *Minimal Area Split* (MINAS), que junta as áreas do "L" com objetivo de formar o maior retângulo possível.



**Figura 14:** Esquema de separação do método Guilhotina (JYLÄNKI, 2010).

É importante notar que, em adição ao método realizado pela biblioteca, foi necessário outros dois cálculos para realizar o diagrama: calcular a espessura da ferramenta de corte e calcular uma "sobra" nas placas comerciais (que geralmente possuem falhas e imperfeições em suas extremidades). Assim, utilizando duas constantes de ambiente e descontando da otimização as devidas espessuras, pode-se realizar o empacotamento.

A partir disso, para cada tipo de material, calcula-se a área total das placas do móvel e dividimos esse valor pela área das placas disponíveis comercialmente. Este número, arredondado para cima, é o número de placas que pode-se usar para o empacotamento. É importante notar que, por não haver atualmente uma integração com sistemas de estoque, são utilizadas somente chapas inteiras, assumindo que não haverá reutilização das sobras de um pedido em outro. Assim, pode-se processar o algoritmo, obtendo-se os diagramas de corte assim como na figura 15



**Figura 15:** Exemplo de chapa otimizada de um móvel.

## 4.10 Cálculo de Ferragens

Para facilitar no momento de construir o móvel é ideal que o pós processador realize o cálculo das ferragens necessárias após o processamento das placas. Assim, seguiu-se as regras da tabela 2, que descrevem a quantidade de ferragens para cada placa do móvel.

Ferragem	Tipo de Placa	Regra
Suportes de prateleira	Prateleiras	$4 * n^o$ de placas
Corrediças	Laterais de gaveta	$n^o$ de laterais/2
Dobradiças	Portas	2, se altura < 900 3, se $900 < \text{altura} < 1500$ 4, se $1500 < \text{altura} < 2000$ 5, se $2000 < \text{altura} < 2700$

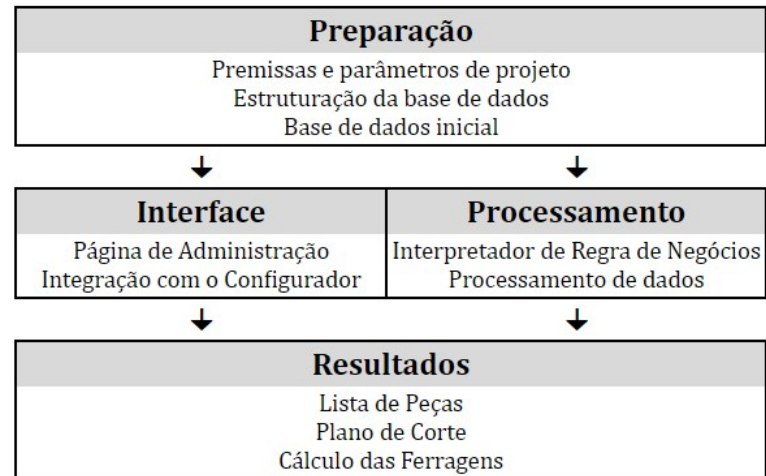
**Tabela 2:** Quantidade de ferragens para cada tipo de placa.

A partir desses dados, pode-se buscar as placas de um certo móvel que estejam de acordo com a tipagem e calcular a quantidade de ferragens de cada tipo que são necessárias para a construção deste.

## 4.11 Resumo

Para resumir os desenvolvimentos realizados é possível entender quatro seções diferentes: Preparação, Interface, Processamento e Resultados; na estrutura hierárquica da imagem 16. Assim, a frente de preparação foi a primeira a se desenvolver, com a definição das premissas e requisitas do projeto, desenvolvimento da base de dados e estruturação de uma base de dados inicial. Posteriormente, as frentes de Interface e processamento desenvolvem-se em paralelo: Interface compreendendo a página de administrador e a integração com o configurador e Processamento com o interpretador de regras de negócio e processamento de dados. Por fim, desenvolve-se os Resultados, onde é realizada a lista de peças, plano de corte e cálculo de ferragens.





**Figura 16:** Diagrama de resumo do desenvolvimento do pós configurador.

## 5 RESULTADOS

Como citado anteriormente, o desenvolvimento foi baseado em releases semanais com acompanhamento da empresa e crescente complexidade. Ao fim de cada release foi gerada a Lista de Peças de cada Móvel e esta foi comparada com o output desejado, gerado manualmente. Em todos os casos o sistema mostrou-se bem sucedido, levando a uma grande economia de tempo da empresa e possibilitando maiores automações e escalabilidade no futuro devido a sua arquitetura baseada em tecnologia REST.

Na figura 17 se encontra o input de um dos móveis utilizados para teste. Ele foi recebido em formato de planilha e convertido manualmente para o JSON a ser alimentado no pós processador - importante notar que este processo será realizado em produção pelo configurador. É possível observar na figura 18 a Lista de Peças gerada para esse móvel, já no formato final e aprovado pela Modú, além do Resumo do Pedido com as imagens dos planos de corte otimizados na figura 19.

Móvel	Tipo	Definition Name	Material	Quantity	LenX	LenY	LenZ	thick_box	thick_back_box	thick_foot
Móvel 6	Caixa completa	Caixa#1	Branco tx	1	200	200	280	15	6	
Móvel 6	Frontão	Frontão#1	Freijó	2		94	280			
Móvel 6	Porta de giro	Porta#1	Preto tx	1	590	190	18			
Móvel 6	Caixa sem fundo	Caixa#2	Branco tx	1	400	200	280	15		
Móvel 6	Frontão	Frontão#2	Preto tx	2		94	280			
Móvel 6	Arremate direito	Arremate direito#1	Freijó	1	50	350	105			
Móvel 6	Arremate esquerdo	Arremate esquerdo#1	Freijó	1	50	350	105			
Móvel 6	Arremate superior	Arremate superior#1	Freijó	1	600	50	105			
Móvel 6	Pé esquadro	Pé esquadro#1	Branco tx	1	600	100	300			
Móvel 6	Rodapé T	Rodapé T #1	Freijó	1	600	100				15

**Figura 17:** Planilha com as especificações do móvel de teste.

Cliente	Móvel	id	comprimento	largura	Tipo Caixa	Fita C1	Fita C2	Fita L1	Fita L2	Material	Função	Complemento
Pedro	Móvel #6	12	190	590		Preto tx	Preto tx	Preto tx	Preto tx	Preto tx 18	Porta	Móvel #7
Pedro	Móvel #6	13	350	210						Freijó tx 15	ArremD	Móvel #7
Pedro	Móvel #6	14	350	210						Freijó tx 15	ArremE	Móvel #7
Pedro	Móvel #6	15	600	210						Freijó tx 15	ArremS	Móvel #7
Pedro	Móvel #6	16	600	205		Freijó tx				Freijó tx 15	Rodapé T	Móvel #7
Pedro	Móvel #6	17	200	279	Caixa Completa	Branco tx				Branco tx 15	Lat	Móvel #7 Caixa #1
Pedro	Móvel #6	18	200	279	Caixa Completa	Branco tx				Branco tx 15	Lat	Móvel #7 Caixa #1
Pedro	Móvel #6	19	169	279	Caixa Completa	Branco tx				Branco tx 15	Teto	Móvel #7 Caixa #1
Pedro	Móvel #6	20	169	279	Caixa Completa	Branco tx				Branco tx 15	Base	Móvel #7 Caixa #1
Pedro	Móvel #6	21	183	183	Caixa Completa					Branco tx 6	Fundo	Móvel #7 Caixa #1
Pedro	Móvel #6	22	250	54	Gaveta Externa	Branco tx	Branco tx			Branco tx 15	Lat Gaveta Externa	Móvel #7 Caixa #2
Pedro	Móvel #6	23	250	54	Gaveta Externa	Branco tx	Branco tx			Branco tx 15	Lat Gaveta Externa	Móvel #7 Caixa #2
Pedro	Móvel #6	24	112	54	Gaveta Externa	Branco tx	Branco tx			Branco tx 15	Frente/Traseira Gaveta	Móvel #7 Caixa #2
Pedro	Móvel #6	25	112	54	Gaveta Externa	Branco tx	Branco tx			Branco tx 15	Frente/Traseira Gaveta	Móvel #7 Caixa #2
Pedro	Móvel #6	26	233	126	Gaveta Externa					Branco tx 6	Fundo Gaveta Externa	Móvel #7 Caixa #2
Pedro	Móvel #6	27	94	190	Gaveta Externa	Freijó tx	Freijó tx	Freijó tx	Freijó tx	Freijó tx 18	Frontão Gaveta Externa	Móvel #7 Caixa #2
Pedro	Móvel #6	28	250	54	Gaveta Externa	Branco tx	Branco tx			Branco tx 15	Lat Gaveta Externa	Móvel #7 Caixa #3
Pedro	Móvel #6	29	250	54	Gaveta Externa	Branco tx	Branco tx			Branco tx 15	Lat Gaveta Externa	Móvel #7 Caixa #3
Pedro	Móvel #6	30	112	54	Gaveta Externa	Branco tx	Branco tx			Branco tx 15	Frente/Traseira Gaveta	Móvel #7 Caixa #3
Pedro	Móvel #6	31	112	54	Gaveta Externa	Branco tx	Branco tx			Branco tx 15	Frente/Traseira Gaveta	Móvel #7 Caixa #3
Pedro	Móvel #6	32	233	126	Gaveta Externa					Branco tx 6	Fundo Gaveta Externa	Móvel #7 Caixa #3
Pedro	Móvel #6	33	94	190	Gaveta Externa	Freijó tx	Freijó tx	Freijó tx	Freijó tx	Freijó tx 18	Frontão Gaveta Externa	Móvel #7 Caixa #3
Pedro	Móvel #6	34	200	279	Caixa Sem Fundo	Branco tx				Branco tx 15	Lat	Móvel #7 Caixa #4
Pedro	Móvel #6	35	200	279	Caixa Sem Fundo	Branco tx				Branco tx 15	Lat	Móvel #7 Caixa #4
Pedro	Móvel #6	36	369	279	Caixa Sem Fundo	Branco tx				Branco tx 15	Teto	Móvel #7 Caixa #4
Pedro	Móvel #6	37	369	279	Caixa Sem Fundo	Branco tx				Branco tx 15	Base	Móvel #7 Caixa #4
Pedro	Móvel #6	38	250	54	Gaveta Externa	Branco tx	Branco tx			Branco tx 15	Lat Gaveta Externa	Móvel #7 Caixa #5
Pedro	Móvel #6	39	250	54	Gaveta Externa	Branco tx	Branco tx			Branco tx 15	Lat Gaveta Externa	Móvel #7 Caixa #5
Pedro	Móvel #6	40	312	54	Gaveta Externa	Branco tx	Branco tx			Branco tx 15	Frente/Traseira Gaveta	Móvel #7 Caixa #5
Pedro	Móvel #6	41	312	54	Gaveta Externa	Branco tx	Branco tx			Branco tx 15	Frente/Traseira Gaveta	Móvel #7 Caixa #5
Pedro	Móvel #6	42	233	326	Gaveta Externa					Branco tx 6	Fundo Gaveta Externa	Móvel #7 Caixa #5
Pedro	Móvel #6	43	94	390	Gaveta Externa	Freijó tx	Freijó tx	Freijó tx	Freijó tx	Preto tx 18	Frontão Gaveta Externa	Móvel #7 Caixa #5
Pedro	Móvel #6	44	250	54	Gaveta Externa	Branco tx	Branco tx			Branco tx 15	Lat Gaveta Externa	Móvel #7 Caixa #6
Pedro	Móvel #6	45	250	54	Gaveta Externa	Branco tx	Branco tx			Branco tx 15	Lat Gaveta Externa	Móvel #7 Caixa #6
Pedro	Móvel #6	46	312	54	Gaveta Externa	Branco tx	Branco tx			Branco tx 15	Frente/Traseira Gaveta	Móvel #7 Caixa #6
Pedro	Móvel #6	47	312	54	Gaveta Externa	Branco tx	Branco tx			Branco tx 15	Frente/Traseira Gaveta	Móvel #7 Caixa #6
Pedro	Móvel #6	48	233	326	Gaveta Externa					Branco tx 6	Fundo Gaveta Externa	Móvel #7 Caixa #6
Pedro	Móvel #6	49	94	390	Gaveta Externa	Freijó tx	Freijó tx	Freijó tx	Freijó tx	Preto tx 18	Frontão Gaveta Externa	Móvel #7 Caixa #6
Pedro	Móvel #6	50	175	100	Pé Esquadro					Branco tx 15	LatPé	Móvel #7 Caixa #7
Pedro	Móvel #6	51	175	100	Pé Esquadro					Branco tx 15	LatPé	Móvel #7 Caixa #7
Pedro	Móvel #6	52	600	100	Pé Esquadro					Branco tx 15	FrentePé	Móvel #7 Caixa #7
Pedro	Móvel #6	53	600	100	Pé Esquadro					Branco tx 15	FrentePé	Móvel #7 Caixa #7

Figura 18: Lista de Peças gerada para o móvel de teste.

## Resumo do Móvel ID-7 | Pedido-7 | Pedro

Descrição: Móvel #8

### Chapas

Material	Quantidade
Preto tx 18	1
Freijó tx 18	1
Freijó tx 15	1
Branco tx 15	4
Branco tx 6	2

### Ferragens

Tipo	Quantidade
Suporte de prateleiras	12
Dobradiças	12
Corrediças	450mm - 15 pares

### Chapas Otimizadas

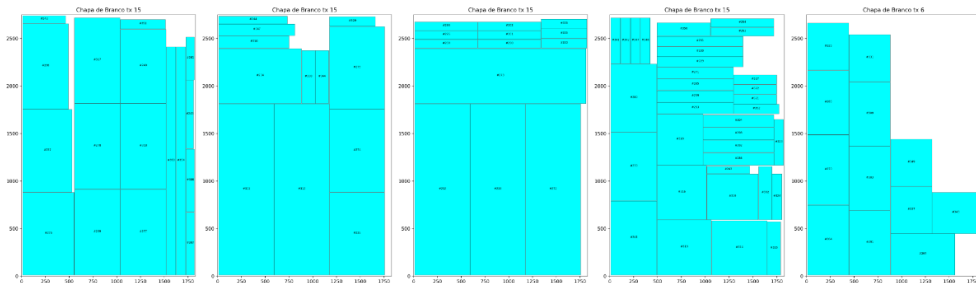


Figura 19: Resumo do Pedido com as imagens dos planos de corte.

## 6 CONCLUSÃO

A construção do pós processador apresentado nesse trabalho se deu pela necessidade de adicionar agilidade, consistência e facilidade em um processo que antes era feito de forma semi-automática dentro da empresa. As informações dos móveis eram alimentadas de forma manual para uma série de planilhas, que realizavam as contas necessárias, mas que, com o crescimento da empresa, o método não seria mais sustentável.

A partir de uma série de conversas com os membros da companhia foi possível desenhar a estrutura da solução a ser construída e começar o seu desenvolvimento. No primeiro momento foi criada uma base de dados capaz de representar no meio digital toda a estrutura e hierarquia presentes na construção de um móvel modular. Em seguida construiu-se um método de interpretação flexível e escalável para as regras de negócio da companhia. Uma vez pronta essa lógica, o programa processa os cálculos necessários e se utiliza de uma biblioteca já existente no Python para traduzir as placas digitais para o mundo físico novamente, por meio da geração da lista de peças e do plano de corte otimizado. Por fim, foi feita a conexão do pós processador com a estrutura atual do configurador e da companhia por meio de um JSON padronizado que garante a comunicação simples e eficaz entre as partes.

O resultado final se encontra atualmente entregue à Modú, que irá realizar os passos finais de integração com o configurador. Ademais, já foi possível notar pelos testes realizados o ganho de eficiência e repetibilidade ao se utilizar um algoritmo próprio e estruturado para os processos da empresa.

### 6.1 Trabalhos Futuros

A medida em que a Modú se desenvolver e se tornar capaz de montar móveis cada vez mais complexos, será necessário implementar novas features e realizar acompanhamentos constantes da base de dados para garantir que nela estão as informações mais atualizadas. Além disso, há diversos trabalhos complementares que trazem benefícios para a empresa e

poderiam ser explorados no futuro, como um precificador automático dos móveis e um CRM mais estruturado, sempre pensando na integração com o conjunto agora formado por configurados e pós configurador.

## REFERÊNCIAS

- CODD, E. *Relational Database: A Practical Foundation for Productivity*. [S.l.]: Elsevier, 1989. 60-68 p.
- FALHEIRO, M. S. et al. Smart configurator to integrate customized furniture design and fabrication. *IFAC-PapersOnLine*, v. 55, n. 2, p. 205–210, 2022. ISSN 2405-8963. 14th IFAC Workshop on Intelligent Manufacturing Systems IMS 2022. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2405896322001951>>.
- FAVI, C. et al. Engineering knowledge formalization and proposition for informatics development towards a cad-integrated dfx system for product design. *Advanced Engineering Informatics*, v. 51, 2022. ISSN 14740346.
- FETTERMANN, D. de C.; ECHEVESTE, M. E. S. Product development for mass customisation: alternatives to the modular furniture industry. *Revista Espacios*, v. 32, 2011. ISSN 07981015.
- FIELDING, R. T.; TAYLOR, R. N. Architectural styles and the design of network-based software architectures. University of California, Irvine, 2000. AAI9980887.
- JOST, P. J.; SÜSSER, T. Company-customer interaction in mass customization. *International Journal of Production Economics*, v. 220, 2020. ISSN 09255273.
- JYLÄNKI, J. A thousand ways to pack the bin - a practical approach to two-dimensional rectangle bin packing. 2010.
- LEFF, A.; RAYFIELD, J. Web-application development using the model/view/controller design pattern. In: *Proceedings Fifth IEEE International Enterprise Distributed Object Computing Conference*. [S.l.: s.n.], 2001. p. 118–127.
- LU, Y.; XU, X.; WANG, L. *Smart manufacturing process and system automation – A critical review of the standards and envisioned scenarios*. 2020.
- ORTNER-PICHLER, A.; LANDSCHÜTZER, C. Integration of parametric modelling in web-based knowledge-based engineering applications. *Advanced Engineering Informatics*, v. 51, 2022. ISSN 14740346.
- SON, S. et al. Production automation system for three-dimensional template pieces used to evaluate shell plate completeness. *International Journal of Naval Architecture and Ocean Engineering*, v. 12, 2020. ISSN 20926790.

## ANEXO A – JSON PARA COMUNICAÇÃO COM O CONFIGURADOR

---

```
1 {
2     "variaveis":{
3         "variable_name":float
4     },
5     "cliente":{
6         "nome":string,
7         "pedido_id":float
8     },
9     "movel":{
10        "movel_id":float,
11        "descricao":string
12    },
13    "portas":[
14        {
15            "tipo":slug,
16            "lenX":float,
17            "lenY":float,
18            "lenZ":float,
19            "material":slug,
20            "quantidade":int,
21            "fita_de_borda":slug
22        }
23    ],
24    "acabamentos":[
25        {
26            "tipo":slug,
27            "lenX":float,
28            "lenY":float,
```

```

29         "lenZ":float ,
30         "material":slug ,
31         "quantidade":int ,
32         "fita_de_borda":slug
33     },
34 ],
35 "caixas":[
36     {
37         "referencia":string ,
38         "tipo":slug ,
39         "lenX":float ,
40         "lenY":float ,
41         "lenZ":float ,
42         "prateleiras":int ,
43         "caixa_gaveteiro":{
44             "tipo":slug ,
45             "lenY":float
46         },
47         "gavetas":[
48             {
49                 "tipo":slug ,
50                 "lenY":float ,
51                 "quantidade":int ,
52                 "material_frontao":slug ,
53                 "material":slug ,
54                 "material_fundo":slug ,
55                 "fita_de_borda":slug ,
56                 "fita_de_borda_frontao":slug
57             }
58         ],
59         "material":slug ,
60         "material_fundo":slug ,
61         "material_prateleira":slug ,
62         "fita_de_borda":slug
63     },
64 ]
65 }

```

---