**Conceitos SE:**

Objetivo de simplificar a busca da resposta requerida. Ter habilidade para aprender com a experiência e explicar o que estão fazendo e porque o fazem.

A **base de conhecimento** é o componente que armazena o conhecimento especializado sobre o domínio em questão. Ela contém **fatos** e **regras**, que são usados pelo sistema para processar informações e chegar a conclusões.

A **máquina de inferência** é responsável por processar os dados da base de conhecimento para tirar conclusões lógicas. Ela utiliza as regras e fatos armazenados para fazer deduções, resolver problemas e tomar decisões.

**Fluxo de Trabalho Simplificado:**

1. **Entrada:** O usuário fornece dados ou consultas ao sistema.
2. **Inferência:** A máquina de inferência processa esses dados, consultando a base de conhecimento para aplicar as regras e deduzir novos fatos.
3. **Saída:** O sistema apresenta a conclusão ou a solução baseada nas inferências realizadas.

Um **shell para sistema especialista** é um ambiente de desenvolvimento pré-construído que fornece a estrutura básica para criar sistemas especialistas sem a necessidade de programar toda a infraestrutura do zero. Ele já inclui componentes essenciais, como:

1. **Base de Conhecimento** – Onde os fatos e regras do domínio são armazenados.
2. **Máquina de Inferência** – O mecanismo que processa as regras e realiza deduções.
3. **Interface de Usuário** – Um meio para entrada de dados e exibição de respostas.

**Fases de Desenvolvimento:**

**Identificação:**

1. Identificar o problema que o sistema especialista pretende resolver.
2. Compreender os problemas enfrentados pelos especialistas humanos e como um sistema automatizado pode ajudar, objetivos que devem ser atingidos, o que é solução?
3. Identificar o campo específico em que o sistema será aplicado.
4. Identificar participantes do projeto (Dono do sistema, Engenheiro de Conhecimento, Especialista, Programadores).
5. Fontes de conhecimento, quais os dados.
6. Classes/Características desses problemas que devem ser resolvidos.
7. O que impede a solução e como vai afetar o SE?

A black and white screen with white text

Description automatically generated

**Conceituação:**

1. Planejar como o sistema especialista será estruturado.
2. Tipo de base de conhecimento, técnicas de inferência e a interface com o usuário, formalizar a base conceitual, grau de refinamento sobre a representação dos dados.
3. Definição de escopo, estabelecer claramente o que o sistema fará e o que não fará.
4. Seleção de Tecnologias.
5. Arquitetura do sistema, planejar como os componentes do SE (base de conhecimento, máquina de inferência, interface etc.) interagirão, definir relações, mecanismos de controle
6. O que é fato e o que é inferido, hipóteses, hierarquias das relações, processos até a solução, fluxo da informação e restrições sobre os processos.

**Formalização**:

1. Capturar o conhecimento especializado do domínio de forma estruturada e formalizada, coleta de conhecimento.
2. Representação do Conhecimento.
3. Traduzir o conhecimento informal dos especialistas em uma forma que o sistema possa usar (por exemplo, regras "SE... ENTÃO..." ou lógica de primeira ordem), como as soluções são geradas.
4. Criação da Base de Conhecimento, organizar o conhecimento formalizado em uma base de dados estruturada para ser utilizada pela máquina de inferência.
5. Características dos dados (tipo, precisão, consistência, aquisição, volume).

**Implementação:**

1. **Desenvolvimento de Regras e Fatos:** Programar as regras e os fatos na base de conhecimento.
2. **Desenvolvimento da Máquina de Inferência:** Implementar o mecanismo de inferência (como raciocínio dedutivo ou indutivo).
3. **Construção da Interface:** Criar uma interface de usuário intuitiva para facilitar a interação com o sistema.

**A screenshot of a computer program

Description automatically generated**

**Teste e Avaliação:**

1. Desempenho e utilidade.
2. Explicação, interação homem máquina, elegância.
3. Eficiência.
4. Credibilidade, qualidade das respostas.
5. Comparar o conhecimento do SE com o do especialista, confrontar com um humano.

**Revisão:**

1. Melhorias.

A diagram of a system

Description automatically generated**Arquitetura Genérica de um SE:**

**Base de Conhecimento (BC) – Representação do Conhecimento:**

Define fatos e regras que servirão como base do SE. Escolhemos como representar esse conhecimento. A **base de conhecimento** conterá os tipos de contratos e cláusulas associadas. A **representação do conhecimento** será representada por regras de produção (se tipo = aluguel então incluir cláusula X).

**Motor de Inferência – Estratégia de Inferência e Tipo de Raciocínio:**

Manipular as regras e fatos definidos na base de conhecimento para chegar a conclusões sobre quais cláusulas são relevantes para cada contrato com base nas respostas do usuário.

Decidimos qual estratégia de inferência utilizar:

**Encadeamento para frente (Forward Chaining):** quando partimos dos fatos conhecidos até uma conclusão.

Definimos o tipo de raciocínio aplicado:

**Dedutivo:** se baseia em regras gerais para chegar a uma conclusão específica.

**Testes e Validação:**

Executamos o sistema e verificamos se ele está chegando às respostas corretas com base na inferência e na representação do conhecimento utilizadas.

**Justificativas para o tema escolhido:**

**Existe necessidade de uma solução que justifique o custo e o esforço para construir o SE?**

Sim, a necessidade de implementar o sistema é válida uma vez que não encontramos esse tipo de ferramenta de consulta em escritórios de advocacia por exemplo. O sistema tem a principal vantagem de agilizar o processo de documentação por parte dos advogados uma vez que automação traria maior agilidade no preparo dos contratos reduzindo também possíveis redundâncias na criação e edição de contratos com cláusulas idênticas e/ou similares.

**A perícia humana não está sempre disponível ou consistente quando necessária?**

Sim, Advogados e especialistas jurídicos nem sempre estão disponíveis imediatamente para analisar cada caso e sugerir cláusulas contratuais adequadas. Além disso, diferentes profissionais podem ter interpretações distintas sobre quais cláusulas são mais apropriadas para um determinado contexto. O sistema especialista ajudaria a padronizar esse processo, garantindo que os contratos gerados sigam boas práticas e normas jurídicas consistentes.

**Esse problema pode ser resolvido utilizando raciocínio simbólico?**

Sim, pois envolve a manipulação de regras e conceitos jurídicos que podem ser representados de forma lógica e estruturada. O sistema pode usar regras de produção do tipo **"Se... então..."** para determinar quais cláusulas devem ser incluídas em um contrato com base nas respostas do usuário.

**INFORMACOES PARA O RELATÓRIO TÉCNICO:**

**A close-up of a text

Description automatically generated**

**a.** O tema escolhido consiste em um Sistema Especialista para Geração de Contratos Imobiliários com Clausulas Relevantes ao Contexto do Cliente. O domínio de conhecimento compreende a área do direito, especificamente o direito imobiliário por conta dos tipos dos contratos que são trabalhados.

**b.** A área de atuação segue na automatização na criação da documentação de contratos imobiliários. Esse tema tem aplicação prática no Direito Imobiliário, podendo ser utilizado por advogados, imobiliárias e corretores para agilizar a criação de documentos contratuais de forma mais eficiente e precisa.

**c.** O sistema tem como objetivo automatizar a geração de contratos imobiliários.

Com base nas informações fornecidas pelo usuário, o SE modela o contrato imobiliário selecionado (aluguel, compra e venda, permuta, cessão de direitos, prestação de serviços e comodato) e fornece um template HTML, incluindo, além das cláusulas, também os seus respectivos itens considerados mais relevantes de acordo com o contexto coletado pela interface de usuário. O modelo HTML fornecido permite o usuário a baixar o contrato em formado DOCX e PDF.

O sistema garante que o documento gerado esteja alinhado com as melhores práticas jurídicas, reduzindo o tempo necessário para a elaboração de contratos e agilizando o processo de documentação.

**d.** Fernanda Garzon de Oliveira, Advogada empresarial no escritório de advocacia Taube Toretta de Cascavel/PR. Bacharel em Direito pela Faculdade UNIVEL – Cascavel-PR (2018) e Pós-graduanda em Direito Penal e Processo Penal. Registro OAB/PR 118.849. O [CURRÍCULO](Base%20de%20Conhecimento/CV_Fernanda_Garzon_Advogada%20.pdf) para consulta completa das credenciais da especialista pode ser encontrado na pasta Base de Conhecimento.

**A close-up of a document

Description automatically generated**

**A.**

Foram reunidos 6 contratos imobiliários contendo 10 clausulas cada. Essa etapa se deu por meio de reuniões semanais de forma presencial onde foi elaborado junto ao especialista quais dados seriam necessários para desenvolver o sistema especialista. O especialista, após a definição do problema e contextualização do objeto a ser desenvolvido, ficou responsável pela pesquisa e levantamento das informações necessárias para compor a base de conhecimento do sistema.

**Tipos de Contratos Imobiliários:**

1. Contrato de Compra e Venda de Imóvel
2. Contrato de Locação de Imóvel
3. Contrato de Comodato de Imóvel
4. Contrato de Permuta de Imóvel
5. Contrato de Sessão de Imóvel
6. Contrato de Prestação de Serviço

[Os contratos em PDF e DOCX podem ser encontrados na pasta BASE DE CONHECIMENTO -> CONTRATOS.](Base%20de%20Conhecimento/Contratos)

[Cada contrato tem sua base legal (jurídica) de uso descrita no arquivo JURISDICAO CONTRATOS na pasta BASE DE CONHECIMENTO -> CONTRATOS.](Base%20de%20Conhecimento/Contratos/JURISDICAO%20CONTRATOS.docx)

As partes grifadas e em negrito em cada contrato representam onde o usuário responsável pela criação do contrato pode ajustar conforme o contexto da situação.

[Cada contrato foi convertido em um modelo HTML onde o usuário consegue exportá-lo como um documento PDF ou DOCX. Os modelos podem ser encontrados na pasta BASE DE CONHECIMENTO -> MODELOS HTML.](Base%20de%20Conhecimento/Modelos%20HTML)

[A explicação sobre o código dos modelos de contrato HTML se encontra na pasta BASE DE CONHECIMENTO -> MODELOS HMTL no arquivo MODELO-CONTRATO.](Base%20de%20Conhecimento/Modelos%20HTML/modelo-contrato.html)

[**O documento CLAUSULAS, presente em BASE DE CONHECIMENTO -> CLAUSULA**](Base%20de%20Conhecimento/Clausulas/CLAUSULAS.docx), contém, grifadas em diferentes cores a fim de diferenciar cada uma, clausulas comuns entre os contratos, ou seja, que aparecem na maioria dos contratos, e que podem ser generalizadas entre os modelos de contratos. O documento ainda contém uma área OUTRAS CLAUSULAS UNIVERSAIS contendo clausulas que não estão presentes nos contratos, porém são clausulas reais que podem ser generalizadas para uso de acordo com o contexto do contrato. Essas novas clausulas foram selecionadas a fim de aumentar o volume de dados na base de conhecimento para favorecer a efetividade no uso do sistema desenvolvido. Por fim o documento contém o nome de cada contrato e todas as suas respectivas clausulas a fim de deixar tudo reunido em um único local para facilitar a consulta sobre os dados. **As cláusulas que não estão destacadas são consideradas clausulas especificas de seu respectivo contrato.**

Ao todo temos 6 contratos, cada um com 10 clausulas, sendo 7 clausulas consideradas universais entre eles (nem todas essas 7 clausulas estão presentes em todos os contratos), e 10 clausulas extras consideradas universais para os 6 tipos de contratos.

Todo o processo de coleta das cláusulas e seus respectivos contratos foi executado e fornecido pela advoga especialista além da generalização das cláusulas consideradas universais entre os contratos, a classificação das cláusulas e a preparação das regras de utilização.

Vale destacar que a escolha das cláusulas que foram consideradas comuns entre mais de um dos 6 contratos, a forma como as cláusulas foram generalizadas, e a definição sobre o uso de cada cláusula foram etapas estabelecidas por meio do olhar da especialista sendo esta responsável por assegurar que o objeto em desenvolvimento segue as diretrizes jurídicas validas e engloba o escopo da proposta deste trabalho.

Sendo assim o especialista foi responsável por entregar ao final da pesquisa os 6 modelos de contrato com 10 clausulas cada um, além das 10 clausulas extras. O especialista também entregou o documento com a jurisprudência de cada contrato como forma de apoio legal sobre a validade do uso de cada modelo de contrato.

**B.**

[O documento CLAUSULAS, presente em BASE DE CONHECIMENTO -> CLAUSULA](Base%20de%20Conhecimento/Clausulas/CLAUSULAS.docx) contémestabelecido as regras de utilização de cada cláusula entre os contratos considerando os seguintes contextos: quando utilizar a cláusula? como? em quais contratos a cláusula poderia ser utilizada? como generalizar a cláusula considerando que temos 6 contratos que abordam diferentes contextos? Qual o contexto e quais variáveis são relevantes na escolha da clausula pelo advogado de acordo com cada contexto contratual? Além disso o documento aborda o contexto da aplicação de cada cláusula avaliada para maior esclarecimento sobre a função da clausula.

Abaixo temos as variáveis que foram identificadas para a seleção das clausulas mais relevantes de acordo com o tipo de contrato. O advogado deve responder as informações que o sistema necessita para definir quais clausulas e itens dessas clausulas estarão presentes no respectivo contrato de acordo com os valores das variáveis identificadas.

**Variáveis do Sistema Especialista:**

**tipo\_contrato:** qual o tipo de contrato sendo criado? (tipo categórica: os 6 tipos de contratos modelados nesse trabalho).

**objetivo\_especifico:** Há um bem (imóvel, serviço ou direito) específico sendo negociado? O usuário descreveu o bem/serviço/direito? (Tipo Booleano).

**partes\_definidas:** As partes (cedente/cessionário, comprador/vendedor etc.) foram identificadas? (Tipo Booleano).

**tempo\_duracao:** O contexto do contrato necessita a determinação de condições sobre prazos? (Tipo Booleano).

**existe\_pagamento:** Existe alguma contraprestação financeira? (Tipo booleano).

**penalidade\_prevista:** Há previsão de multa ou penalidade pela infração e Descumprimento dos direitos aplicáveis negociados entre as partes?(Tipo Booleano).

**aviso\_previo:** É necessário a notificação sobre avisos prévios para situações específicas?(Tipo Booleano).

**localizacao\_partes:** Existe a necessidade sobre a identificação sobre a residência das partes e a região responsável por resoluções jurídicas e de eventuais disputas relacionadas ao contrato? (Tipo Booleano).

**observacoes\_finais:** Existe a necessidade da inclusão de observações sobre a validade do contrato entre outras questões gerais? (Tipo Booleano).

**ajustes\_futuros:** As partes desejam incluir uma previsão de possibilidade de ajustes futuros ao contrato? (Tipo Booleano).

**confirmacao\_legislacao:** Existe a necessidade de observações sobre ainterpretação, prevalência de termos e legislação ou regulamentação específica aplicável ao contrato? (Tipo Booleano).

**exige\_garantia:** O contrato exige alguma garantia formal em relação a riscos relevantes de inadimplência? (Tipo Booleano).

**tipo\_garantia:** O contrato deve possuir algum tipo de garantia? (caução, fiador, seguro fiança etc.) (Tipo Booleano).

**interesse\_preferencia:** Existe a possibilidade do interesse de uma das partes em ter maior prioridade caso o bem seja negociado? (Tipo Booleano).

**meio-comunicacao:** Existe a necessidade de estabelecer um meio de comunicação que será utilizado para notificações (e-mail, carta registrada, ambos)? (Tipo Booleano).

**natureza\_objeto\_especializada:** A obrigação contratual exige mão de obra ou serviços de terceiros especializados? (Tipo Booleano).

**permite\_subcontratacao:** Existe a possibilidade da aceitação que o contratado utilize terceiros para executar suas obrigações? (Tipo Booleano).

**complexidade\_execucao:** A execução das obrigações é complexa ou envolve múltiplas etapas? (Tipo Booleano).

**riscos\_externos:** Existe a possibilidade que o objeto do contrato seja prejudicado devido a riscos de fatores externos (greves, crises, pandemias)? (Tipo Booleano).

**duracao\_contrato:** O contrato possui tempo de duração relevante envolvendo a execução de obrigações contínuas que são passiveis de alta formalidade e proteção jurídica? (Tipo Booleano).

**coleta\_informacao:** O contrato necessita aborda a existência da coleta de informações relevantes como dados cadastrais ou documentais fornecidas pelas partes? (Tipo Booleano).

**veracidade\_informacao:** O contrato exige formalização jurídica robusta para validação de informações sobre todos os dados informados e suas dependências? (Tipo Booleano).

**condicoes\_uso:** O contrato deve prever a definição sobre a utilização sobre o objeto negociado além de associações e exigências especificas que devem ser respeitadas relacionadas ao uso do bem?(Tipo Booleano).

**restricoes\_objeto:** O contrato necessita determinar restrições e exigências sobre questões de utilização, ambientalismo, zoneamento e demais advertências especificas a respeito de continências sobre o objeto?(Tipo Booleano).

**compromissos\_danos:** O contrato deve estabelecer as diretrizes sobre a responsabilidade das partes por eventuais danos sobre o objeto? (Tipo Booleano).

**risco\_danos:** Existe risco potencial de danos ao bem, imóvel ou serviço? (Tipo Booleano).

**peridiocidade:** O contrato necessita prever condições de prazo, execução contínua ou periódica e termos especiais para renovação? (Tipo Booleano).

**transferencia\_sucessao:** Existe a necessidade de definir obrigações e contenções sobre transferências de direitos e sucessão do objeto sobre as partes envolvidas? (Tipo Booleano).

**obrigacoes\_tributarias:** O contrato necessita definições formais sobre responsabilidades tributais das partes envolvidas relacionadas a natureza do objeto?(Tipo Booleano).

**natureza\_juridica:** A natureza do contrato exige o englobamento de questões tributarias especificas a respeito de questões fiscais jurídicas? (Tipo Booleano).

As regras de produção seguem o padrão IF-ELSE onde as variáveis são avaliadas para identificar qual clausula deve ser inserida e seus respectivos itens. Cada pergunta feita ao usuário representa uma **variável** que influencia a seleção das cláusulas e seus respectivos itens, e por meio disso podemos representar os fatos que identificam o que o SE sabe sobre o respectivo contrato e, assim, as regras analisam os fatos onde a decisão se uma cláusula deve ser incluída é avaliada.

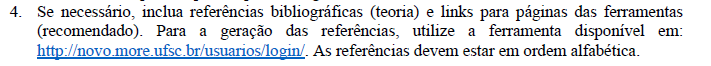
A representação em alto nível da regra para uma melhor abstração pode ser encontrado no arquivo [O documento CLAUSULAS, presente em BASE DE CONHECIMENTO -> CLAUSULA](file:///C:\Users\Eduar\OneDrive\Documents\Programacao\GitHub%20Repositories\Expert-System\Base%20de%20Conhecimento\Clausulas\CLAUSULAS.docx) ao final de cada clausula. A representação demonstra a logica que será seguida para cada regra na implementação da base de conhecimento na shell CLIPS.

**C.**

1. **Estratégia de Inferência:**
   1. A estratégia de inferência escolhida é a foward chaining (encadeamento para frente) uma vez que o sistema determina as cláusulas relevantes ao contrato de acordo com os dados inseridos pelo usuário.
2. **Tipo de Raciocínio:**
   1. O tipo de raciocínio desenvolvido será o dedutivo pois o modo de inferência é baseado a partir dos dados informados pelo usuário sem a necessidade de aprendizado prévio. O sistema se utiliza dos fatos e regras pré-definidas para decidir quais clausulas incluir.
3. **Representação de Conhecimento:**
   1. O conhecimento será representado a partir de regras de produção (IF-THEN) por ser uma estrutura rápida e eficiente que se alinha com a forma de inferência e raciocínio definidos.
4. **Shell:**
   1. A princípio a shell escolhida será o CLIPS.

**A close up of black text

Description automatically generated**

****

[**CLIPS Rule Based Programming Language download | SourceForge.net**](https://sourceforge.net/projects/clipsrules/)

**Checklist das Etapas do Fluxo de Trabalho:**

1. **Identificação do Problema:** Definir claramente o problema a ser resolvido.
2. **Concepção do Sistema:** Planejar a arquitetura e as tecnologias a serem usadas.
3. **Formalização do Conhecimento:** Coletar e estruturar o conhecimento especializado.
4. **Implementação do Sistema:** Programar o sistema com base no conhecimento formalizado.

**D.**

**ETAPA 1 DA IMPLEMENTACAO: BASE DE CONHECIMENTO**

O trabalho implementa o SE utilizando o CLIPS – Baseado em regras, amplamente usado para sistemas especialistas simbólicos. Fácil estruturação das cláusulas contratuais em regras "se-então".’

A base de conhecimento contendo os fatos e a regras de produção podem ser encontradas na pasta [BASE DE CONHECIMENTO -> BK-CLIPS no arquivo base-conhecimento.clp.](Base%20de%20Conhecimento/BK-CLIPS/base-conhecimento.clp)

A base de conhecimento armazena apenas o identificador da clausula e gerar o texto na fase de geração do modelo HMTL. O CLIPS apenas determina quais cláusulas devem ser incluídas, mas o texto delas é armazenado separadamente em um JSON. O CLIPS cuida da inteligência e lógica de seleção.

**A base de conhecimento será composta por:**

* **Facts (fatos):** Informações fornecidas pelo usuário (ex: tipo-contrato, partes-juridicas etc).
* **Rules (regras):** Regras de produção if-then, como as que foram criadas para cada cláusula no arquivo CLAUSULAS.
* **Cláusulas Generalizadas:** Apenas os identificadores das cláusulas vão para a base de conhecimento (não o texto completo).

Cada pergunta feita ao usuário representa uma variável (ou atributo) que influencia a seleção das cláusulas. Quando o usuário responde, a resposta é convertida em fatos no CLIPS.

**A screenshot of a computer

AI-generated content may be incorrect.**

Para cada variável (fatos) definimos todos os valores validos evitando que o sistema receba valores inválidos. A maioria das variáveis são booleanas, ou seja, as respostas do usuário são limitadas a SIM e NÃO.

**Exemplo de Fato no CLIPS:**

**(objetivo-especifico sim)**

**(partes-definidas sim)**

**Exemplo de Regra no CLIPS:**

**(defrule nome-da-regra**

**(Fato 1)**

**(Fato 2)**

**(Fato 3)**

**...**

**=>**

**(assert (Novo Fato Incluído))**

**)**

Utilizamos **deftemplate** para definir o "formato" do contrato. Aqui definimos o contrato em si onde ele é representando como uma estrutura de múltiplos fatos que nada mais são do que as variáveis identificadas do sistema para a contextualização para a identificação das cláusulas e seus itens. Templates são como estruturas ou registros. Eles permitem definir fatos com múltiplos campos (slots), ou seja, cada fato pode ter vários atributos.

**A screen shot of a computer program

AI-generated content may be incorrect.**

**Abaixo temos um exemplo de regra implementada no CLIPS:**

**defrule** são as regras que verificam fatos e disparam para incluir cláusulas.

Usamos **assert** para incluir os novos fatos caso a regra seja considerada válida.

A seleção das cláusulas e seus itens segue o modelo "cláusula + itens separados" onde o primeiro assert indica a cláusula que vai ser incluída e o segundo assert indica quais itens da clausula serão incluídos.

Os nomes das regras seguem o mesmo padrão incluir-clausula-nome-da-clausula- (completo ou parcial n)

A regra é avaliada considerando os fatos presentes e os não presentes que são relevantes para a cláusula e a partir disso ocorre a inferência para a seleção da clausula.

No CLIPS, E é o comportamento padrão. Ou seja, se você escrever múltiplas condições dentro de uma regra, ele já interpreta como "todas essas condições devem ser verdadeiras".

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Com a base de conhecimento pronta agora podemos inferir os fatos no próprio motor de inferência do CLIPS. O tópico Teste e Avaliação explica como testar a base manualmente na CLIPSIDE.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**ETAPA 2: IMPLEMENTANDO O BACKEND**

Com a base de conhecimento pronta agora precisamos implementar um backend responsável por enviar as respostas do usuário (respondidas em um formulário frontend) para o motor de inferência do CLIPS, e depois pegar a resposta do CLIPS (clausulas e seus respectivos itens) e inserir dentro de um documento HTML modelando o contrato especificado e apresentá-lo no frontend.

**APP.PY:**

O backend foi implementado utilizando uma API Flask em Python. Esse servidor local é responsável pelo recebimento das informações do frontend em formato JSON. Esse JSON é enviado para o motor do CLIPS que executa o processo de inferência sobre a base de conhecimento e devolve as clausulas e itens selecionados, estes por sua vez, são novamente convertidos em JSON e enviador para o gerador do contrato HTML, com o contrato modelado a API retorna esse HTML para o frontend. [A explicação sobre o código da API pode ser encontrada no arquivo app.py em Implementacao -> backend -> app.py](Implementacao/backend/app.py)

**MOTOR\_CLIPS:**

O backend além da API Flask contém o código responsável pelo motor de inferência do CLIPS. Esse código utiliza a biblioteca clipspy possibilitado rodar o motor de inferência CLIPS sem ter a necessidade de ter o CLIPS instalado na máquina ou acessá-lo como um subprocesso. [O código motor\_clips.py pode ser encontrado em Implementacao -> backend -> clips\_engine -> motor\_clips.py contendo toda a explicação sobre seu funcionamento.](Implementacao/backend/clips_engine/motor_clips.py)

**Observação Importante:** Neste trabalho testamos todas as formas possíveis de integrar o CLIPS ao python sendo elas: utilizar o CLIPS como subprocesso (não eficiente e muitos problemas com compatibilidade e ainda existe a necessidade de ter o CLIPS instalado na máquina), biblioteca pyclips (desatualizado e descontinuado no python 2) e biblioteca clipspy. Estamos utilizando o clipspy pois foi a única opção que funcionou para integrar o CLIPS ao python, esta biblioteca não está atualizada para versões do python 3.9+ e ainda no ambiente Windows alguns problemas de compatibilidade podem ocorrer ao tentar usar esta biblioteca.

O clipspy se demonstrou 100% funcional utilizando python 3.8.18 usando Linux ou no ambiente WSL dentro do Windows. Isso não necessariamente é um problema considerando que em uma hospedagem real do sistema ele teria seu próprio ambiente virtual com as dependências necessárias instaladas.

**O motor\_clips recebe da API as respostas do frontend em formato JSON:**

**A screen shot of a computer

AI-generated content may be incorrect.**

Esses dados do JSON representam os fatos dentro da base de conhecimento. O motor\_clips é responsável por receber esse JSON e para cada item do JSON converte para o formato adequado do fato de acordo com a base de conhecimento. Com os fatos montados no estilo esperado:

**A screen shot of a computer screen

AI-generated content may be incorrect.**

É executado a inferência pelo motor do CLIPS. O CLIPS retorna os fatos inferidos contendo as clausulas e quais itens serão inseridos, esses fatos são novamente convertidos para um formato JSON no seguinte estilo:

**A screen shot of a computer

AI-generated content may be incorrect.**

**GERADOR\_HTML:**

Após ser retornado para a API o JSON com os resultados da inferência pelo motor-clips o código responsável por modelar o contrato HTML pode ser encontrado em [Implementacao -> backend -> gerador\_html.py](Implementacao/backend/gerador_html.py) contendo toda sua explicação.

Este código recebe o JSON com as respostas da inferência e a partir disso modela o contrato HTML. Como dito anteriormente o modelos dos contratos e o modelo base de contrato podem ser encontrados em [Cada contrato foi convertido em um modelo HTML onde o usuário consegue exportá-lo como um documento PDF ou DOCX. Os modelos podem ser encontrados na pasta BASE DE CONHECIMENTO -> MODELOS HTML.](file:///C:\Users\Eduar\OneDrive\Documents\Programacao\GitHub%20Repositories\Expert-System\Base%20de%20Conhecimento\Modelos%20HTML)

Então para cada tipo de contrato é inserido seus respectivos cabeçalhos, rodapés e suas clausulas obrigatórias e depois os dados do JSON são extraídos para que se possa identificar as clausulas e seus itens respectivos que devem estar dentro do contrato.

O JSON retornado pelo motor\_clips contem apenas os identificadores da clausula e seus itens e o tipo de contrato, o backend contem outro arquivo JSON clausulas.json encontrado [em Implementacao -> backend -> clausulas.json](Implementacao/backend/clausulas.json), este arquivo contem de fato toda a descrição textual das cláusulas e também as tags html embutidas nos textos para questões de estilos visuais.

Então o código gerador\_html relaciona os identificadores retornados pelo motor\_clips com as cláusulas que estão JSON clausulas.json e adiciona estes elementos dentro do html que esta sendo montado.

Concatenando todas as informações necessárias para formar o html o código retorna para a API o HTML completo do respectivo contrato. Além disso o código gera dentro da pasta backend o documento HTML do contrato gerado para que possamos conferir se tudo está correto.

**A screenshot of a computer program

AI-generated content may be incorrect.**

Por fim o contrato é retornado pela API onde quem vai receber é o frontend.

**ETAPA 3: IMPLEMENTANDO O FRONTEND**

Por fim, com o backend completo, implementamos o frontend utilizando JS, HTML e CSS. O front end consiste em um formulário onde o usuário responde a todas as perguntas, cada pergunta equivale a uma das variáveis da base de conhecimento que são avaliadas nas regras.

O formulário então dispara uma requisição para a API enviando as respostas no formato JSON esperado por ela e a partir disso toda a logica do backend ocorre. Por fim a API retorna para o frontend o documento HTML que é embutido dentro html da página por meio de um iframe por questões de preservação das tags de estilo e logica JS que é embutida dentro do modelo de contrato. O usuário pode ajustar suas respostar e gerar de forma dinâmica novos contratos que podem ser baixados tanto em PDF quando em Docx.

**Abaixo o fluxo resumido de todo o processo:**

**A black and white text

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

[**Na pasta Expert-Sytem pode-se encontrar o manual do usuário explicando como configurar o ambiente para roda a API e como utilizar a aplicação.**](Manual%20do%20Usuario.docx)

1. **Teste e Avaliação:** Testar e avaliar o sistema para garantir que atenda aos requisitos.

**TESTANDO A BASE DE CONHECIMENTO NO CLIPSIDE:**

Inicialmente testamos a base de conhecimento diretamente no CLIPS sem nenhuma interface implementada onde fazemos manualmente assert os fatos e rodamos o motor de inferência. Realizamos todos os testes para garantir que todas as regras estejam funcionando corretamente e que todas as cláusulas estejam sendo incluídas corretamente de acordo com todos os contextos possíveis identificados pelas regras.

Na base de conhecimento utilizamos deffacts que inicializa o sistema com um exemplo de contrato (para teste).

**A screen shot of a computer

AI-generated content may be incorrect.**

**Usando CLIPSIDE nos carregamos o arquivo da base de conhecimento:**

Em Environmnet -> Clear

Em Environmnet -> Load Constructs -> Selecionar a base de conhecimento

No terminal digitamos: (reset) (run) e por fim (facts)

**A screenshot of a computer

AI-generated content may be incorrect.**

**TESTANDO O BACKEND MANUALMENTE POR MEIO DE REQUISICOES NO TERMINAL:**

Com a API rodando podemos realizar uma solicitao manualmente pelo terminal, aqui estamos utilizando linux ou WSL para simular o request para API. Esse teste simula o fluxo que sera executado desde o frontend inicial até a resposta final.

**Subindo o servidor Flask local:**

**A screen shot of a computer

AI-generated content may be incorrect.**

**Realizando Requisicao:**

**curl -X POST http://localhost:5000/gerar-contrato -H "Content-Type: application/json" -d '{**

**"tipo-contrato": "comodato",**

**"objetivo-especifico": "sim",**

**"partes-definidas": "sim",**

**"coleta-informacao": "sim",**

**"veracidade-informacao": "sim",**

**"aviso-previo": "nao",**

**"confirmacao-legislacao": "sim",**

**"penalidade-prevista": "sim",**

**"exige-garantia": "sim",**

**"tipo-garantia": "sim",**

**"interesse-preferencia": "sim",**

**"meio-comunicacao": "sim"**

**}'**

**A black screen with white text

AI-generated content may be incorrect.**

O código motor\_clips contêm diversos prints para que possamos acompanhar todas as etapas do processo de inferência. Abaixo podemos ver os principais resultados destacados:

**Fatos retornados pelo CLIPS após a inferência:**

**A computer screen with white text

AI-generated content may be incorrect.**

**Fatos CLIPS modelas em formato JSON e o JSON que é retornado para a API:**

**A black screen with white text

AI-generated content may be incorrect.**

**Documento HTML retornado pela API flask:**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

Na própria pasta do backend é gerado o contrato HTML onde podemos confirmar com clareza se tudo funcionou como o esperado ao final da requisição.

1. **Revisão e Ajustes:** Melhorar o sistema com base nos testes e no feedback recebido.