

# Goal-Oriented Requirements Engineering

Beatriz Braga Batista de Souza



# Como requisitos são elicitados para sistemas IA?

Vamos analisar esse requisito para um sistema com tradução automática:

[RF01] O sistema deve fornecer traduções precisas e confiáveis do texto de origem para o texto de destino

# Engenharia de Requisitos

A principal medida do sucesso de um sistema de software é o grau em que ele atende aos seu propósito. Portanto, identificar esse propósito deve ser uma das principais atividades do desenvolvimento de sistemas de software.

Assim, a engenharia de requisitos deve abordar as razões pelas quais um sistema de software é necessário, as funcionalidades que ele deve ter para atingir seu propósito e as restrições sobre como o software deve ser projetado e implementado.



# Processo natural de engenharia de requisitos

## 1º Passo

O contexto atual em questão é analisado em sua forma organizacional, operacional e técnica

## 2º Passo

Os problemas são apontados e identificados

## 3º Passo

São então identificados e aperfeiçoados **objetivos** de alto nível para resolver esses problemas

## 4º Passo

Os requisitos são então elaborados para atingir esses **objetivos**

# Exemplo

**Objetivo:** Desenvolver um aplicativo móvel para gerenciamento de finanças pessoais.

[RF01] O aplicativo deve permitir aos usuários registrar suas transações financeiras diárias, incluindo despesas, receitas e transferências entre contas.

[RF02] O aplicativo deve gerar relatórios mensais de despesas e receitas, apresentando gráficos e resumos categorizados para facilitar a visualização e análise das finanças.

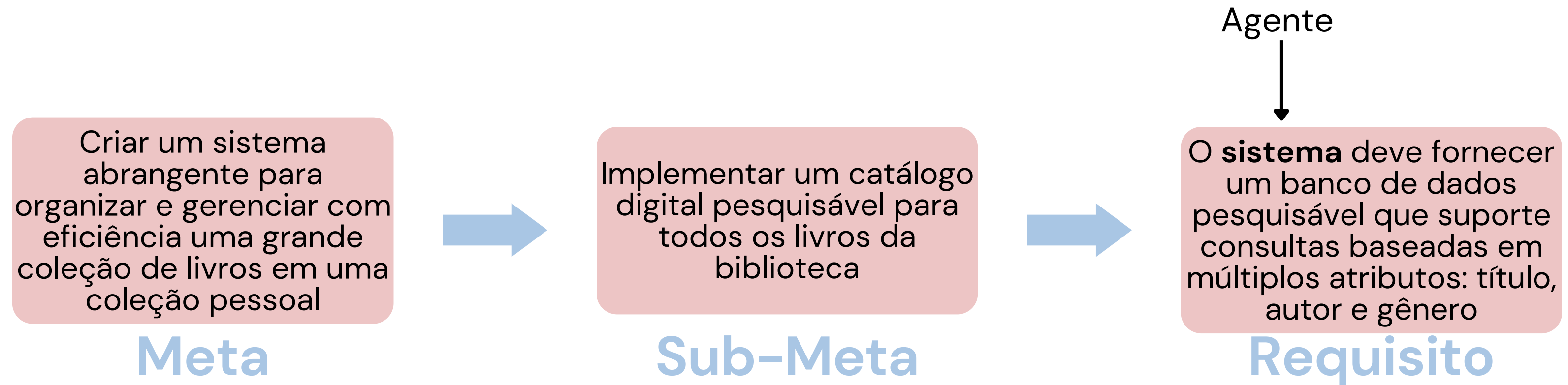
[RF03] O aplicativo deve integrar-se com contas bancárias dos principais bancos, permitindo a importação automática de transações para facilitar o acompanhamento das finanças.

# Introduzindo GORE

- O GORE é uma abordagem para a engenharia de requisitos através de metas.
- Uma **meta** é um objetivo que o sistema em consideração deve atingir.
- O GORE envolve a elicitac  o, o refinamento, a decomposi  o e a operacionaliza  o de metas para garantir o alinhamento com os objetivos dos stakeholders.
- Os requisitos no GORE funcionam como **implementa  es das metas**.
- No GORE, os agentes s  o respons  veis por atingir as metas.

# Exemplo

A partir de uma meta, é possível ir quebrando ela em sub-metas até ela chegar num nível baixo o suficiente para ser considerado um requisito.



**Pergunta:** Quais outras sub-metas e requisitos podem ser extraídos dessa meta?

# Agentes e requisitos



O **sensor de temperatura** deve disparar um alerta se a temperatura no local de armazenamento exceder 25°C.



O **aplicativo** deve gerar automaticamente relatórios mensais resumindo as estatísticas da atividade do usuário no aplicativo.



# ER Tradicional vs. GORE

## Tradicional

**Ponto de partida:** Declarações de requisitos iniciais fornecidas pelo cliente, centradas em "o que" o sistema deve fazer.

**Objetivo:** Produzir um documento de requisitos abrangente que possa ser entregue aos programadores.

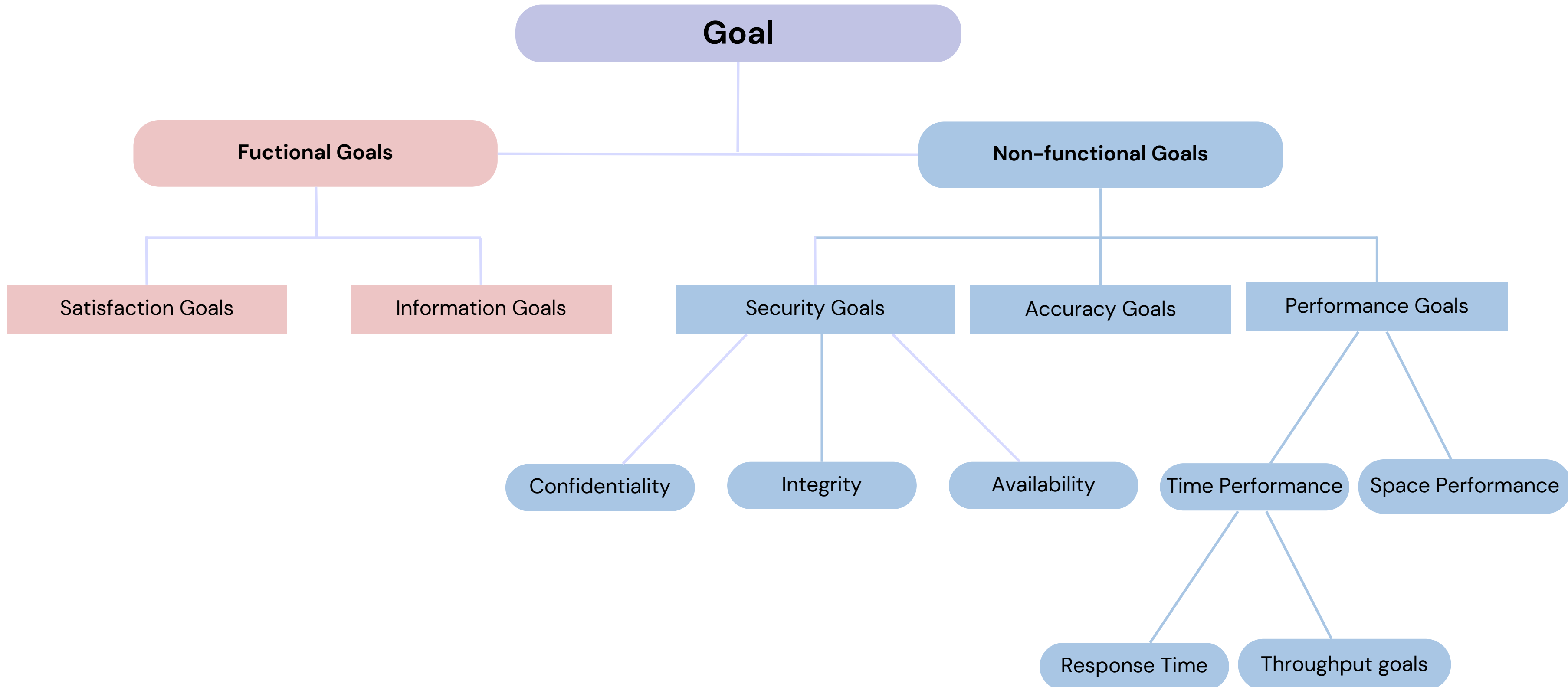
## GORE

**Ponto de partida:** Identificar e compreender os objetivos que os stakeholders têm para o sistema, abordando o "porque" o sistema deve fazer o que faz.

**Objetivo:** Alinhar os requisitos do sistema com as metas e as intenções dos stakeholders, assegurando que o sistema resultante cumpra estas metas de nível superior.

# Taxonomia no GORE

Produzir um documento de requisitos abrangente que possa ser entregue aos programadores.



# Tipos de Metas

## Soft Goals

Estes não podem ser estabelecidos com critérios de satisfação claros. São utilizados para comparar aperfeiçoamentos alternativos.

**Exemplo:** “Melhorar a satisfação dos usuários com o aplicativo.”

## Hard Goals

Estas podem ser verificadas e a sua satisfação pode ser claramente estabelecida.

**Exemplo:** “Garantir que o aplicativo processe as transações em 2 segundos.”

# Frameworks de Aplicação do GORE

## NFR

O framework NFR (Non-Functional Requirements) centra-se na modelação e análise sistemáticas dos requisitos não funcionais, assegurando que estes estão na mente dos programadores.

## KAOS

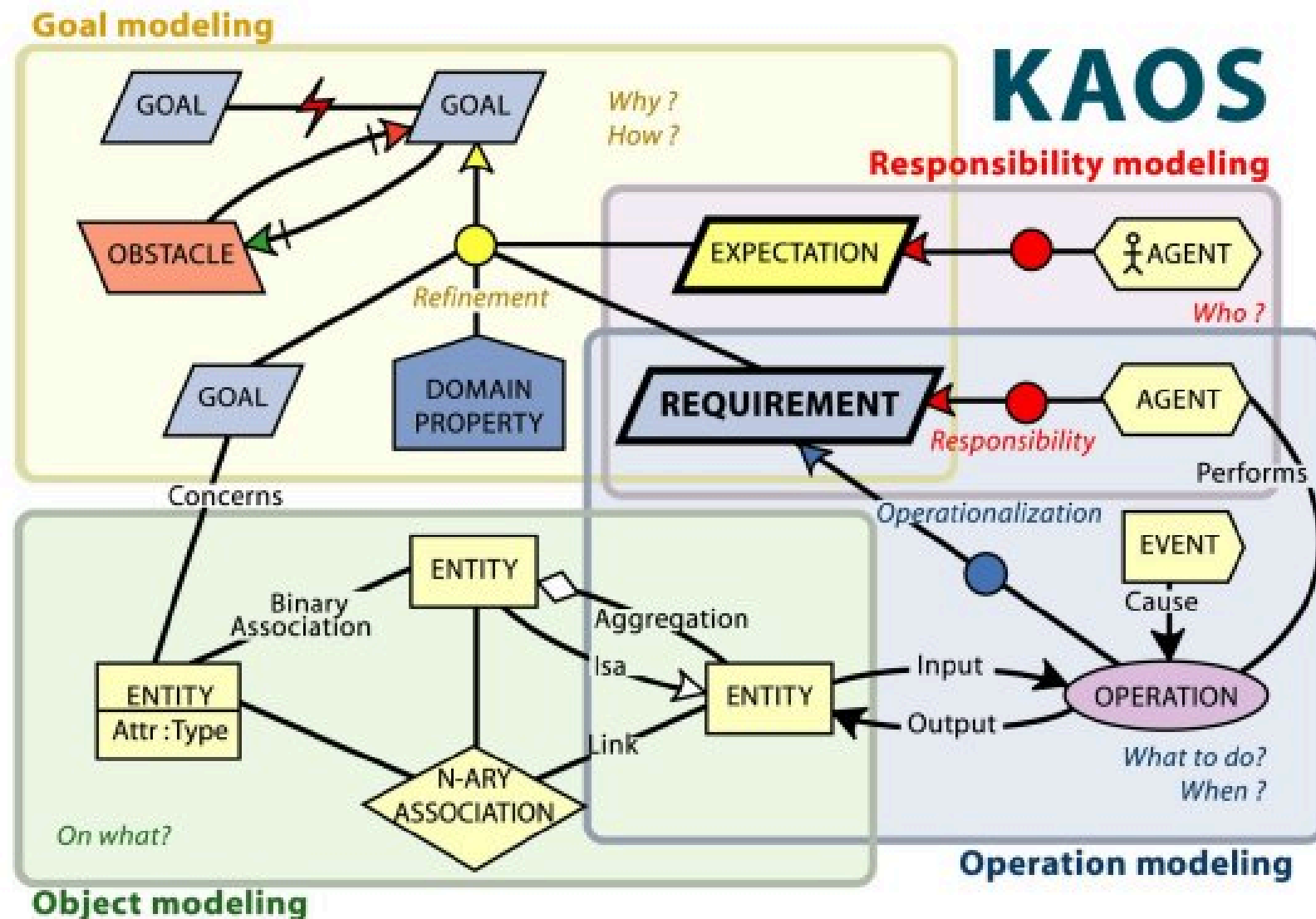
KAOS é uma metodologia de engenharia de requisitos orientada para objetivos que integra raciocínio semi-formal, qualitativo e formal.

## i\*

O framework i\* é uma metodologia de modelação orientada para os agentes utilizada na engenharia de requisitos, na reengenharia de processos empresariais e na análise do impacto organizacional.

# Metodologia KAOS

Knowledge Acquisition in autOmated Specification or Keep All Objects Satisfied

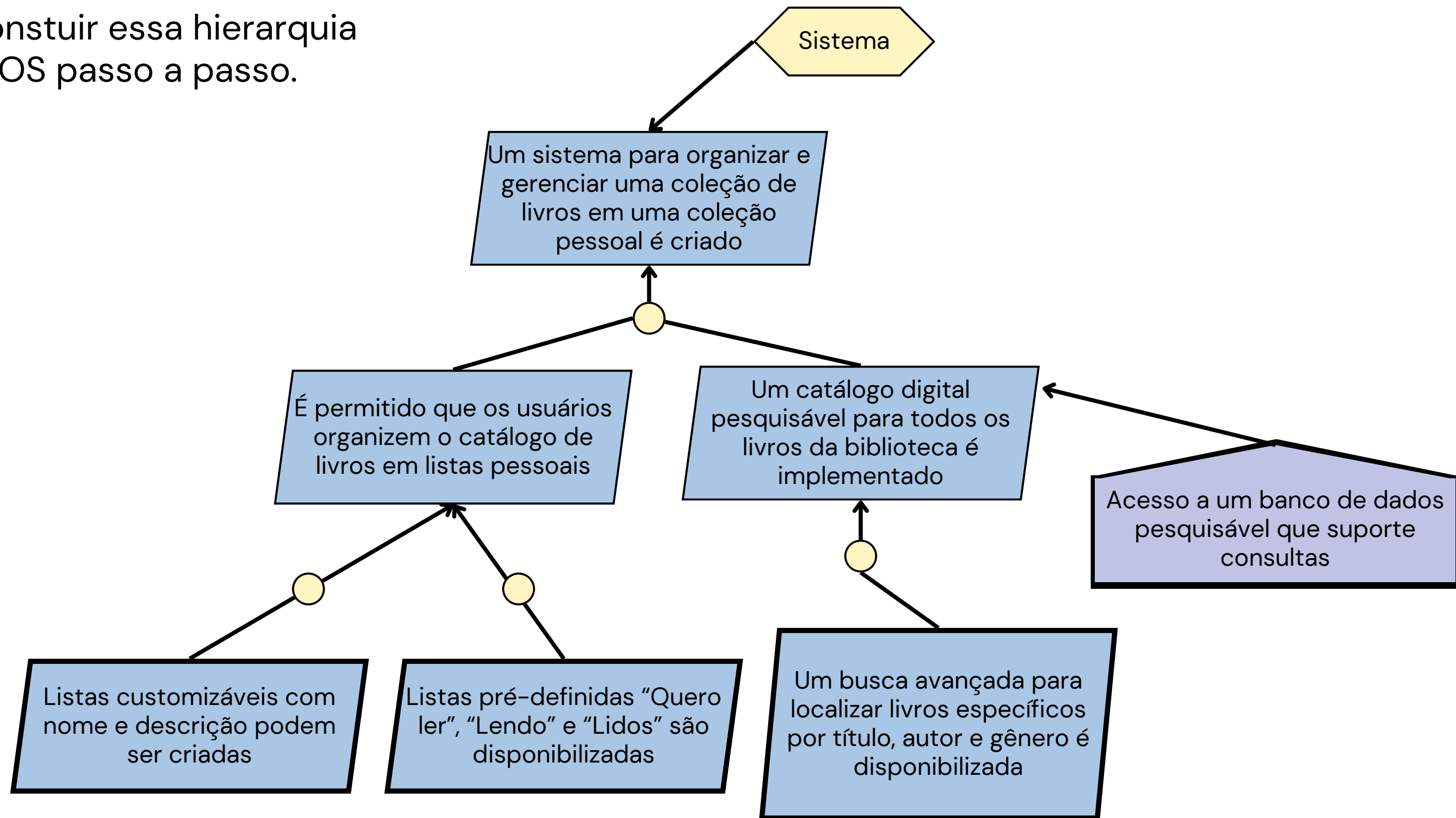


A KAOS é descrita como um framework multiparadigma que permite combinar diferentes níveis de expressão e raciocínio: semi-formal para modelagem e estruturação de objetivos, qualitativo para seleção entre as alternativas e formal, quando necessário, para um raciocínio mais preciso.

A KAOS tem origem numa cooperação entre a **Universidade de Oregon** e a **Universidade de Lovaina** (Bélgica) em **1990**. A pesquisa, as extensões e as melhorias da metodologia continuam a ser efetuados regularmente na Universidade de Lovaina.

# Exemplo

Vamos construir essa hierarquia com o KAOS passo a passo.

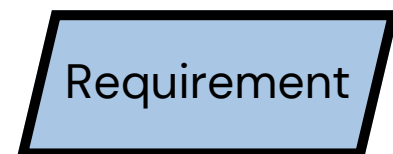


# Elementos KAOS

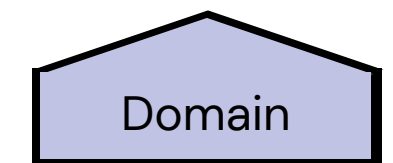
A modelagem de objetivos KAOS é o conjunto de diagramas de objetivos inter-relacionados que foram elaborados para resolver um determinado problema



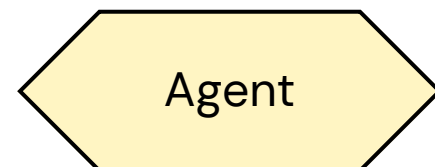
São propriedades desejadas do sistema.



Os requisitos são metas atribuídas a um só agente de software.

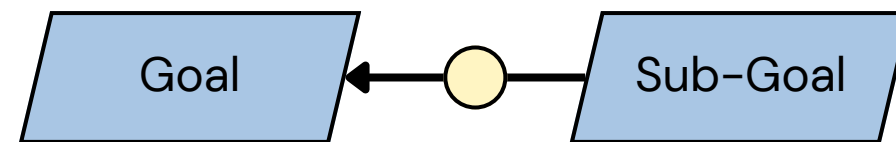


São descrições sobre o contexto ou ambiente que estão propensos a mudanças (regras de negócio, leis da física e etc.)

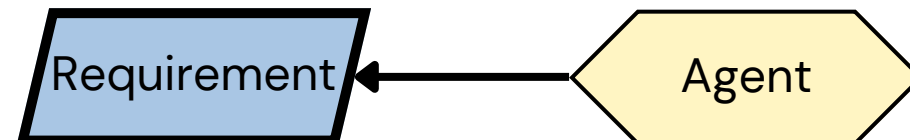


Os agentes são seres humanos ou componentes automatizados responsáveis pela concretização dos requisitos.

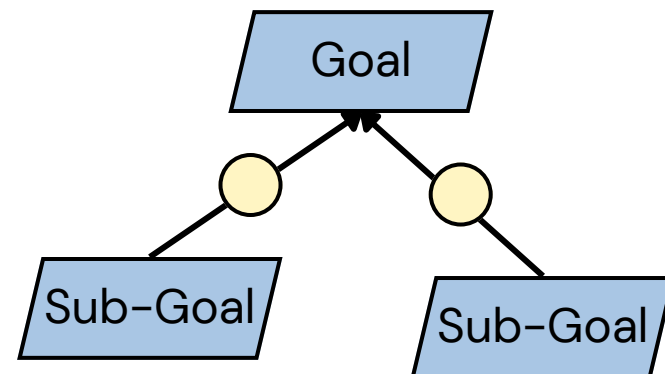
# Links KAOS



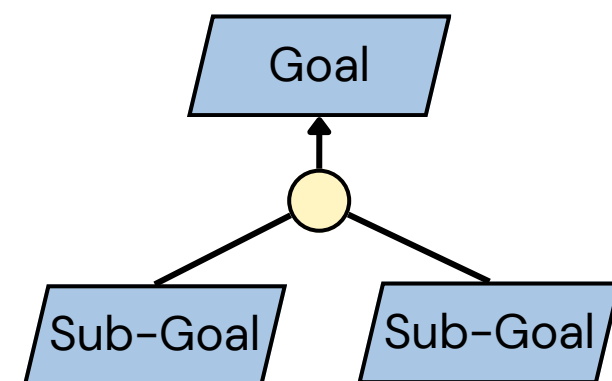
As **ligações entre metas** destinam-se a captar situações em que as metas apoiam positivamente ou negativamente outras metas



São introduzidas **ligações de responsabilidade** para relacionar os requisitos e seu agente.



As **ligações de refinamento OR** significam que a satisfação de **uma das submetas** é suficiente para satisfazer a meta principal.



As **ligações de refinamento AND** relacionam uma meta com um conjunto de submetas. Isto significa que a satisfação de **todas as submetas** no refinamento é suficiente para satisfazer a meta principal.

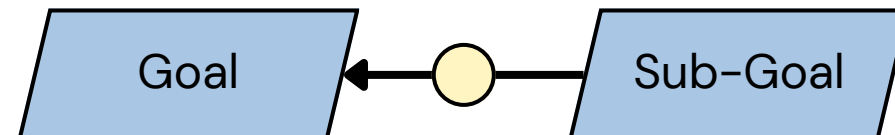


# Exemplo

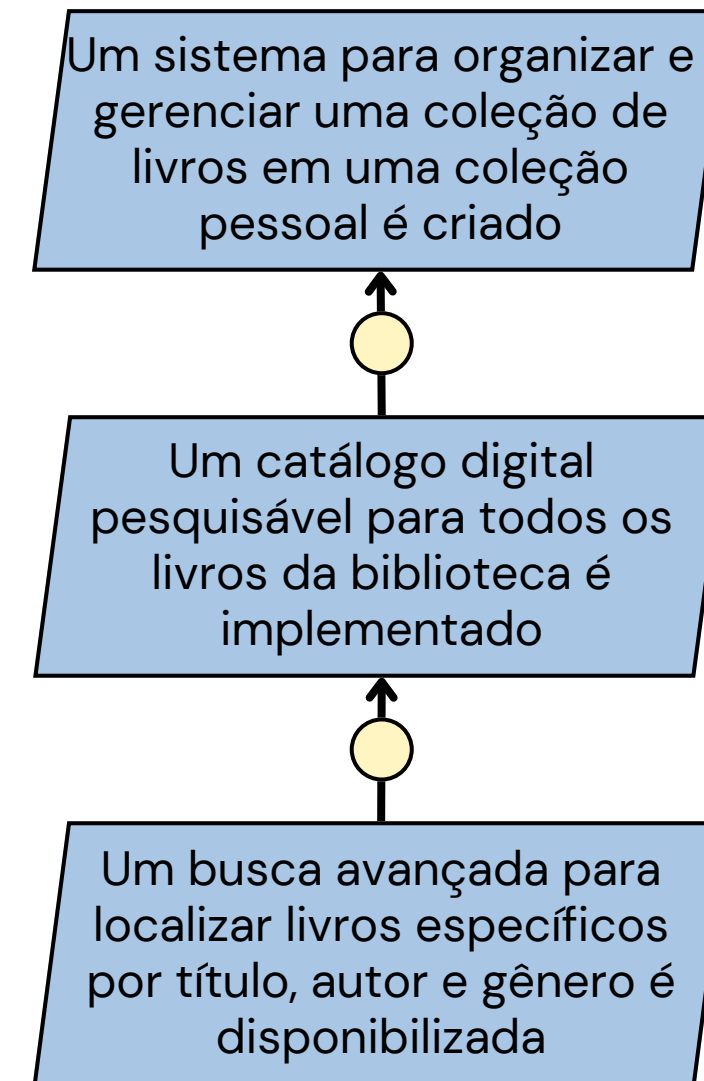
Vamos criar uma hierarquia básica refinando um goal.



São propriedades desejadas do sistema.

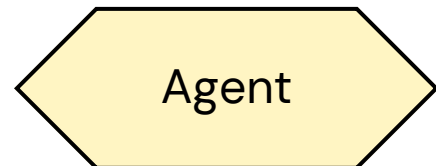


As ligações entre metas destinam-se a captar situações em que as metas apoiam positivamente ou negativamente outras metas

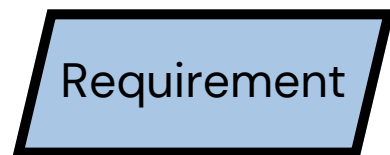


# Exemplo

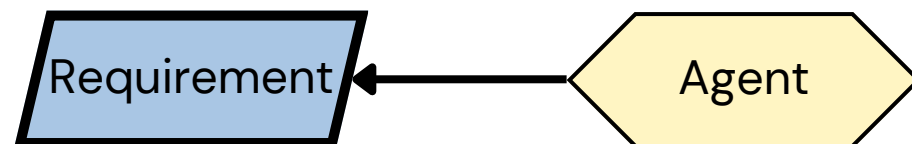
Ao terminar de refinar um sub-goal, você pode ligar ela a um agente, dessa forma, temos um requisito.



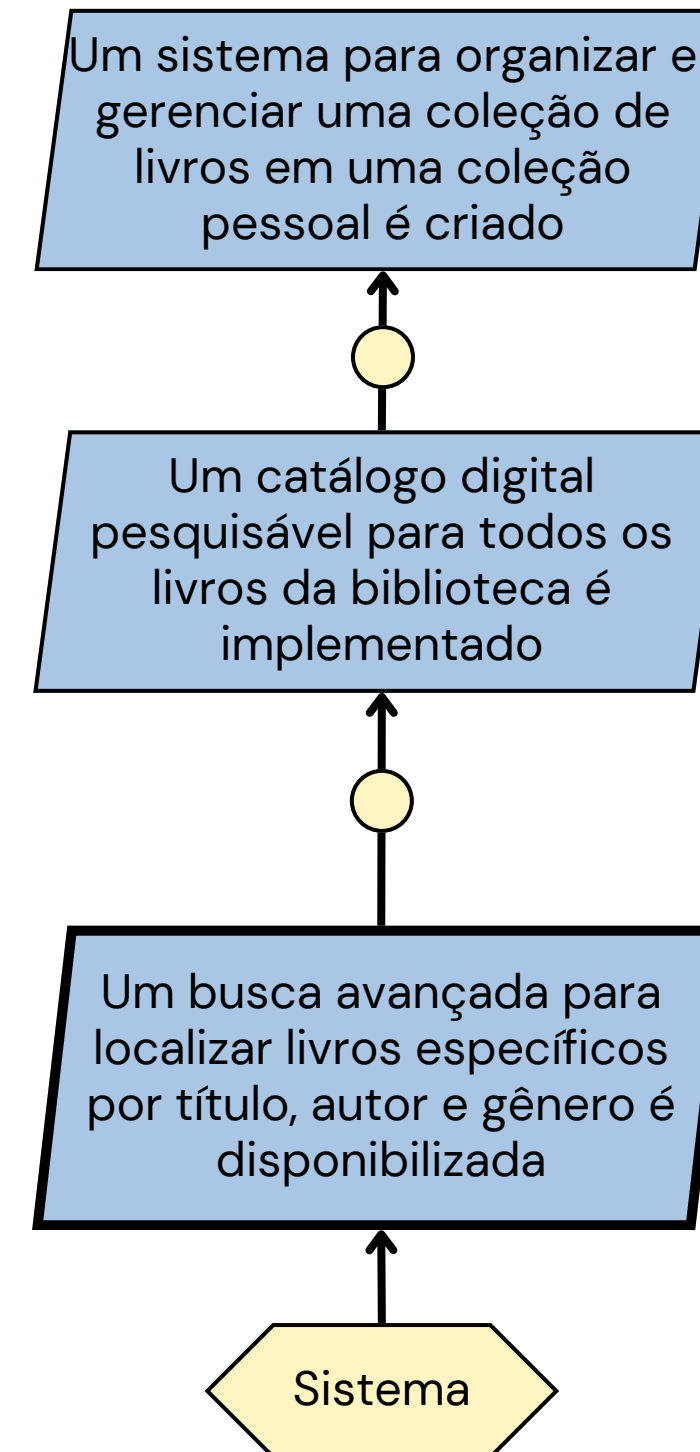
Os agentes são seres humanos ou componentes automatizados responsáveis pela concretização dos requisitos.



Os requisitos são metas atribuídas a um só agente de software.

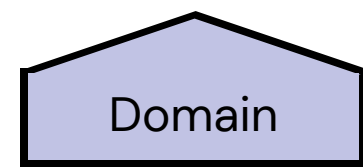


São introduzidas ligações de responsabilidade para relacionar os requisitos e seu agente.

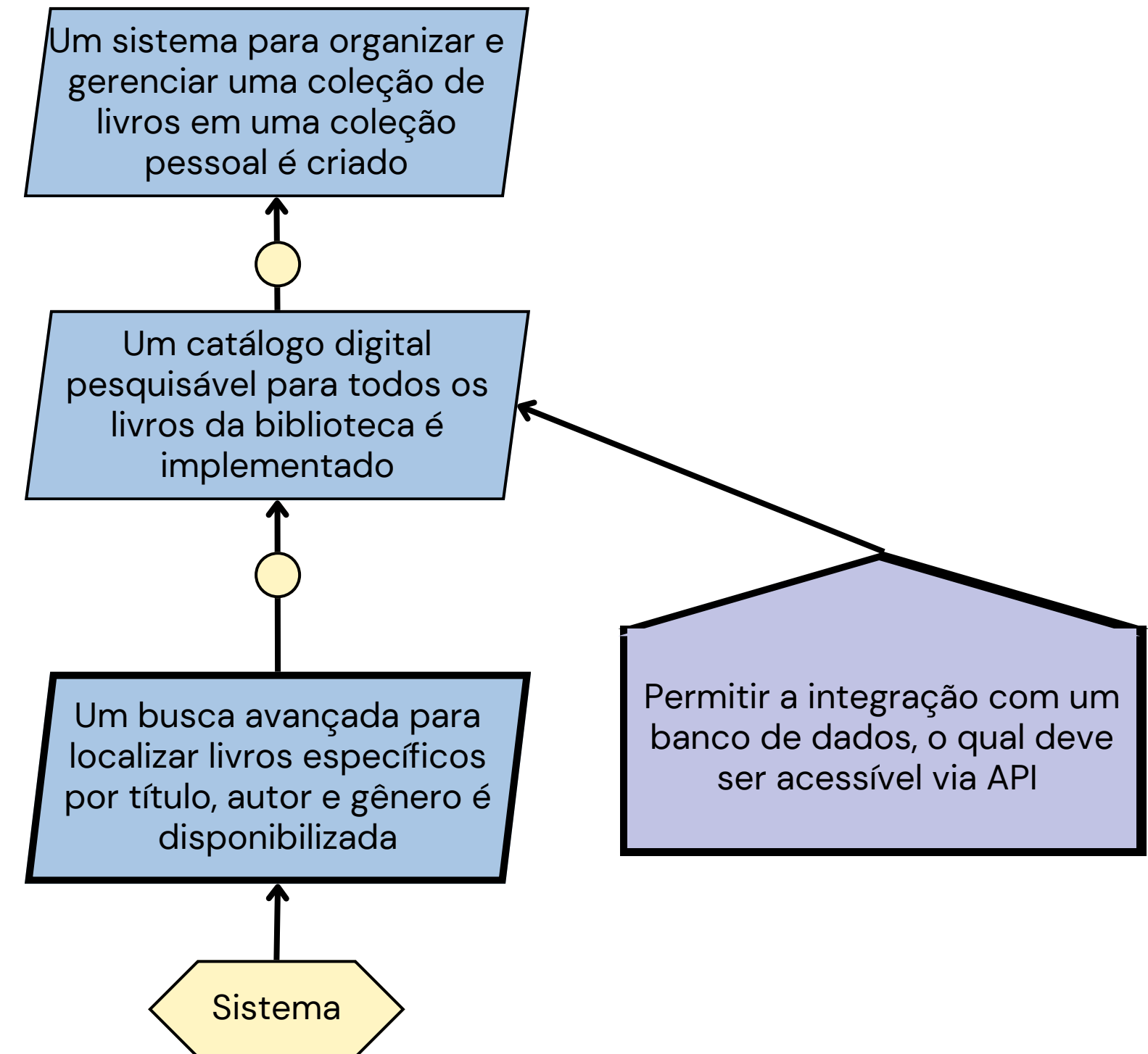


# Exemplo

Deve-se garantir um acesso a um banco de dados para que o objetivo se cumpra, vamos indicar esse fato no diagrama.

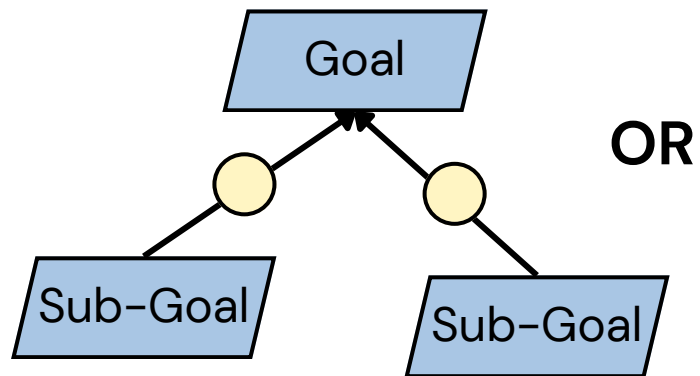


São descrições sobre o contexto ou ambiente que estão propensos a mudanças (regras de negócio, leis da física e etc.)

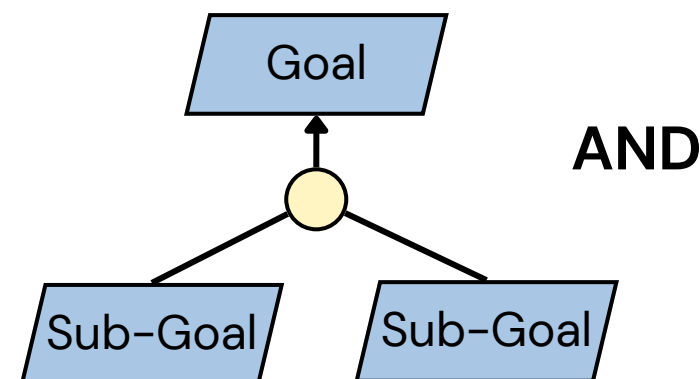


# Exemplo

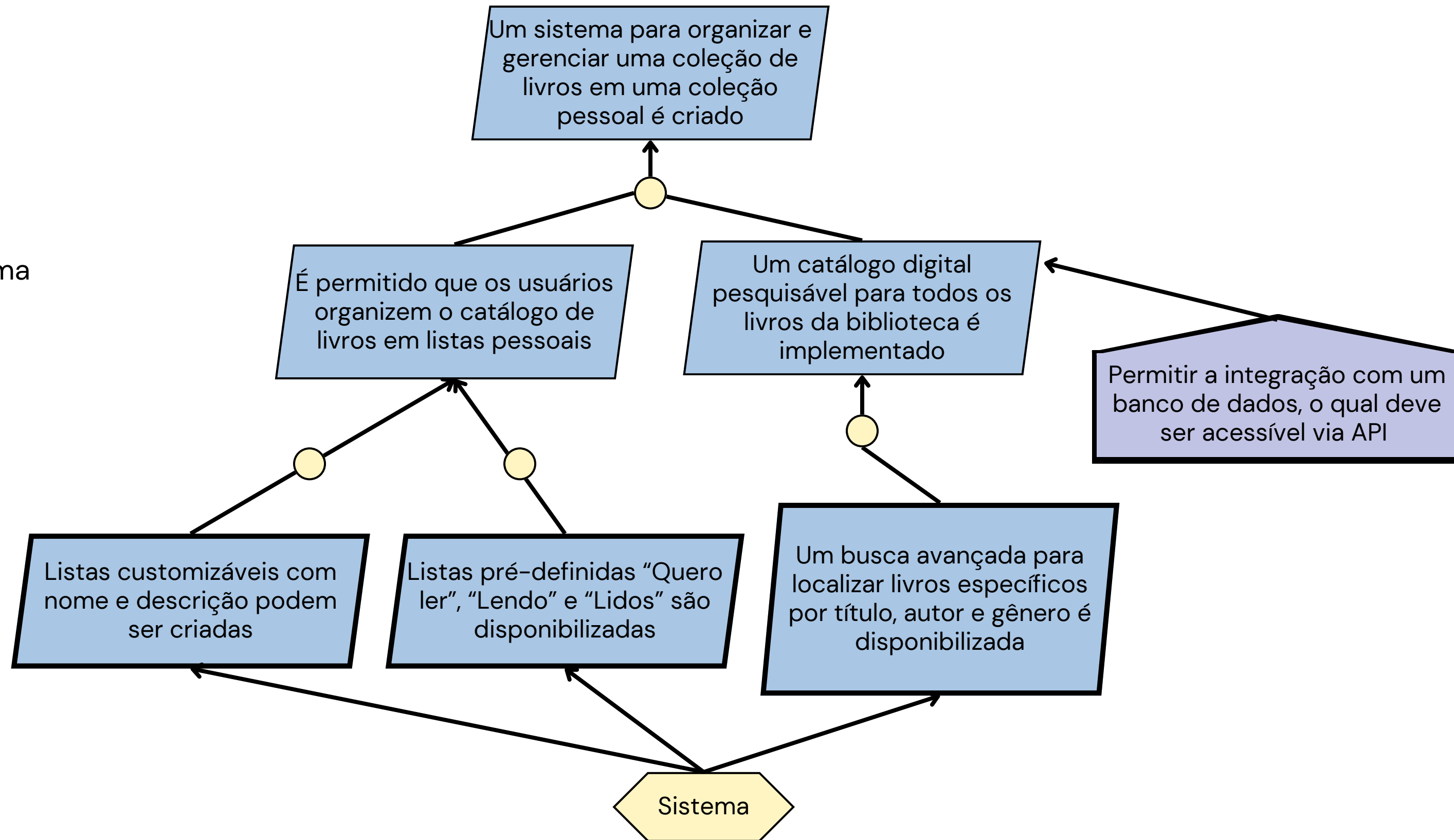
Vamos elicitar mais requisitos.



As ligações de refinamento OR significam que a satisfação de uma das submetas é suficiente para satisfazer a meta principal.



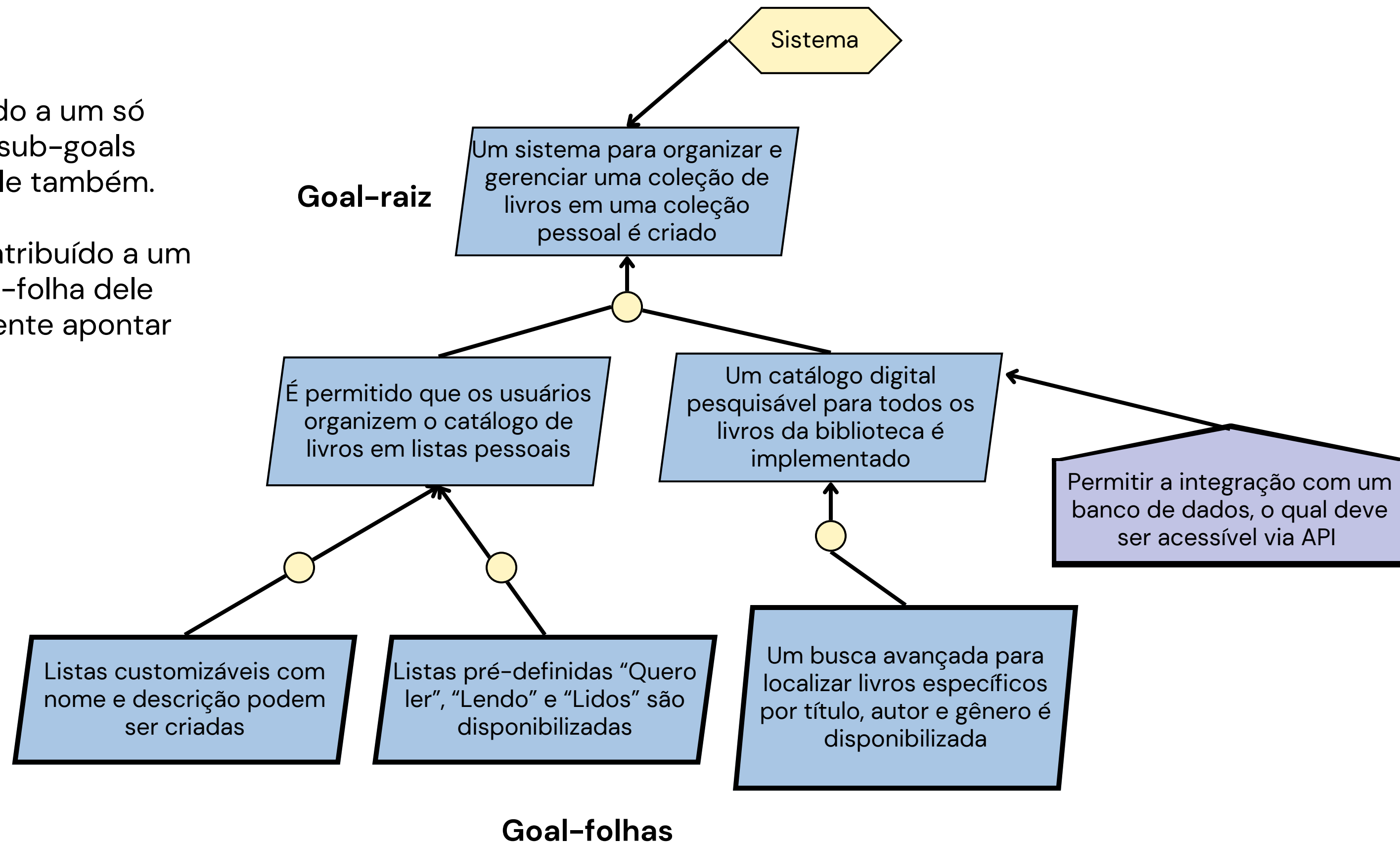
As ligações de refinamento AND relacionam uma meta com um conjunto de submetas. Isto significa que a satisfação de todas as submetas no refinamento é suficiente para satisfazer a meta principal.



# Exemplo

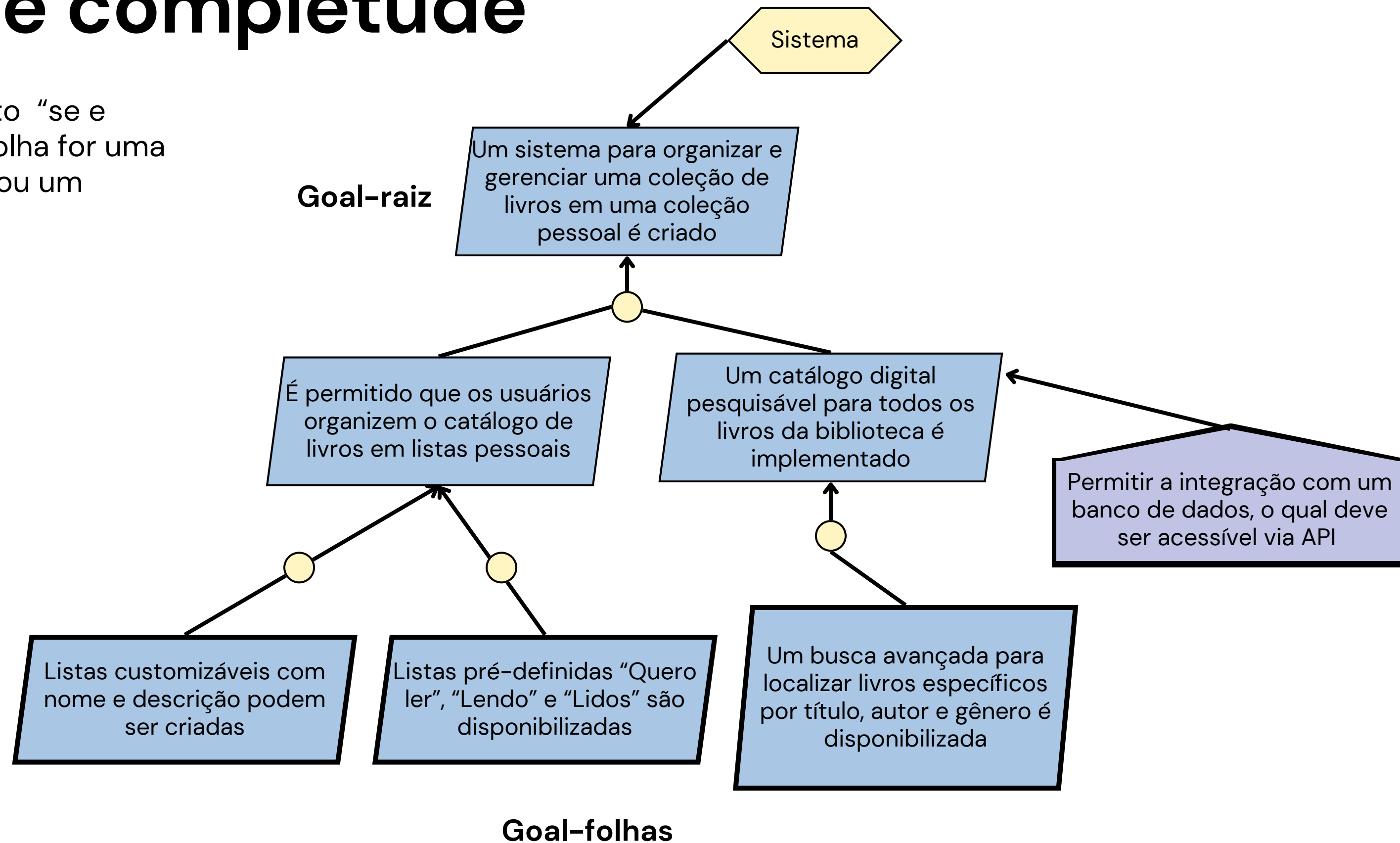
Um goal pode ser atribuído a um só agente se todos os seus sub-goals podem ser atribuídos a ele também.

Caso um goal possa ser atribuído a um só agente, então os goals-folha dele não precisam explicitamente apontar para um agente.



# Critério de completude

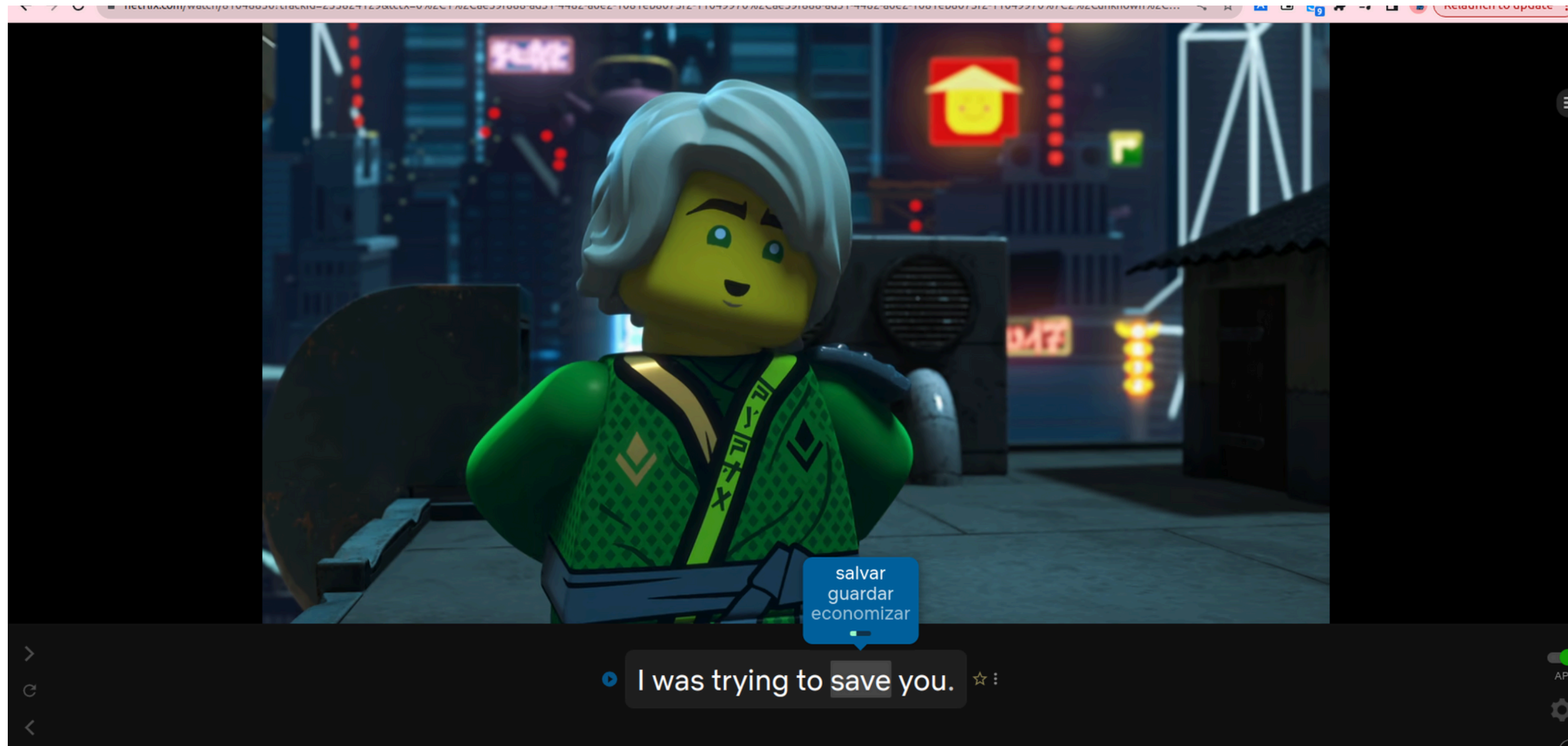
O diagrama está completo “se e somente se” cada goal-folha for uma propriedade de domínio ou um requisito.





# Exemplo

O **Language Reactor** é uma extensão para a Netflix para ajudar no aprendizado de linguas estrangeiras. Na qual, caso habilitada, o usuário pode passar o mouse por cima de uma palavra estrangeira desconhecida na legenda e ele mostra em um **pop up** o **significado no dicionário** traduzido.

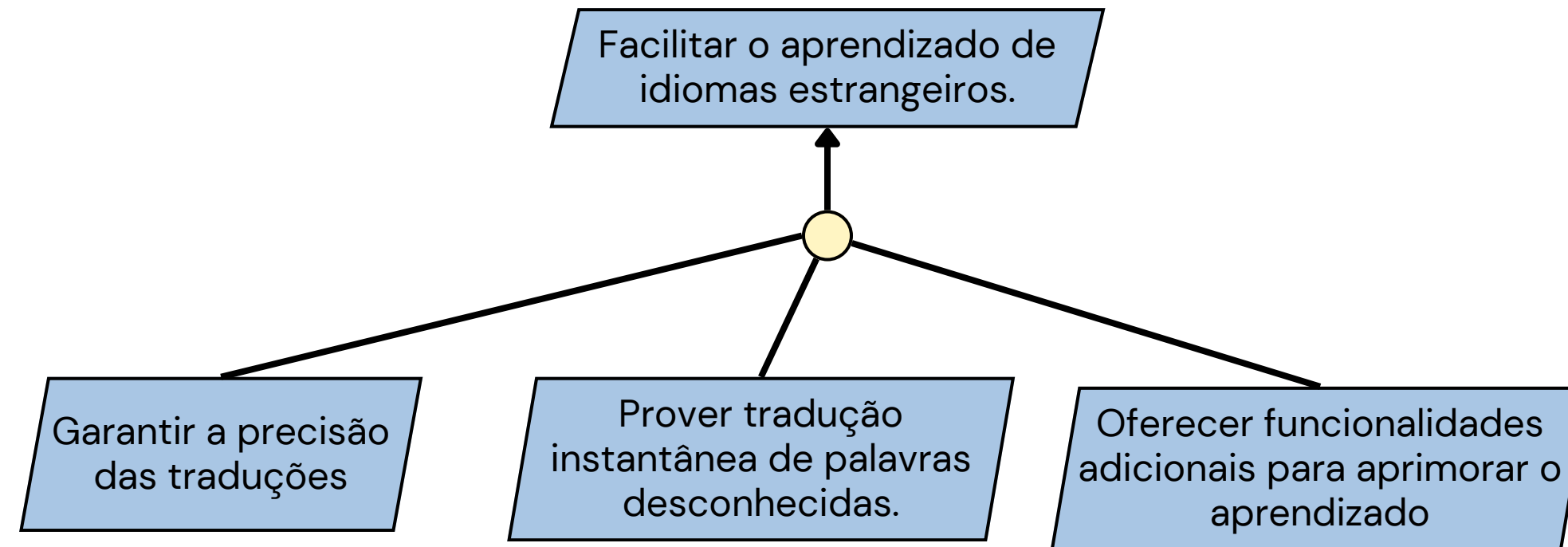


# Exemplo

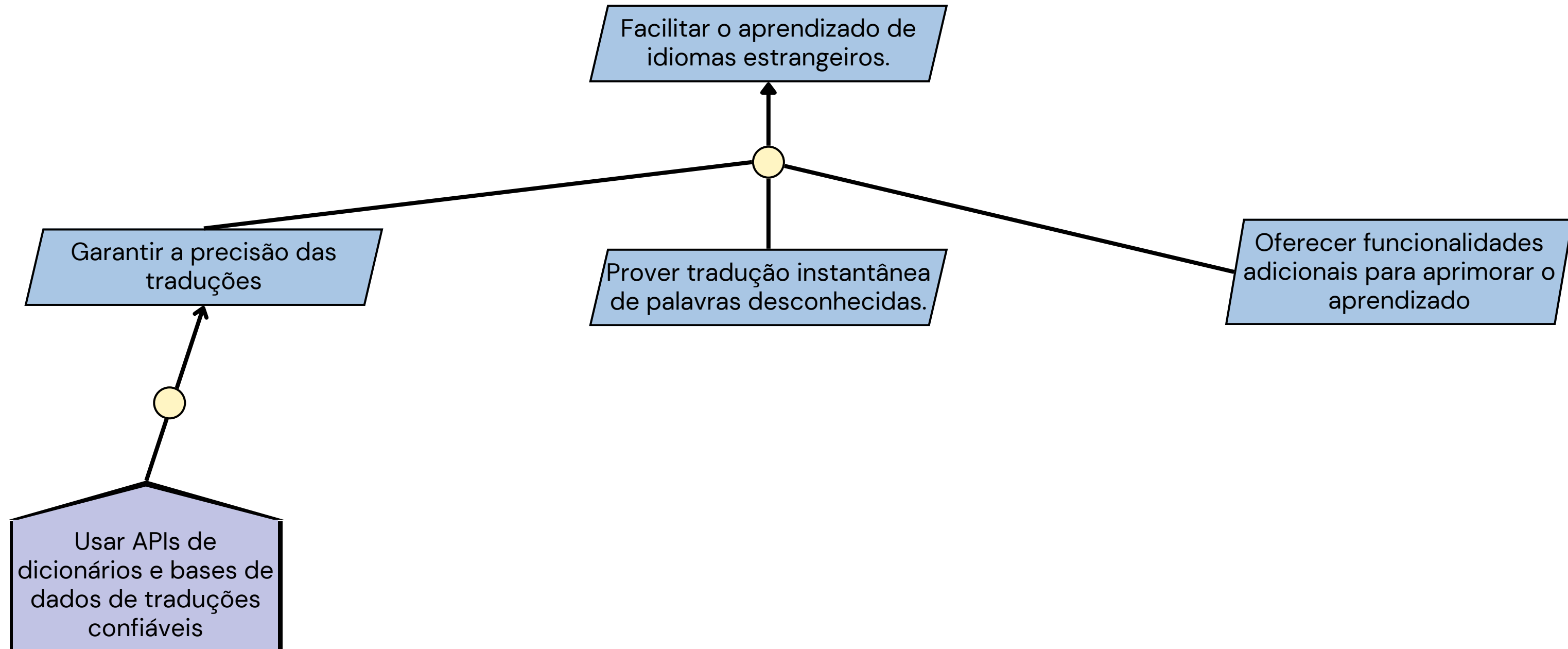
Facilitar o aprendizado de idiomas estrangeiros.



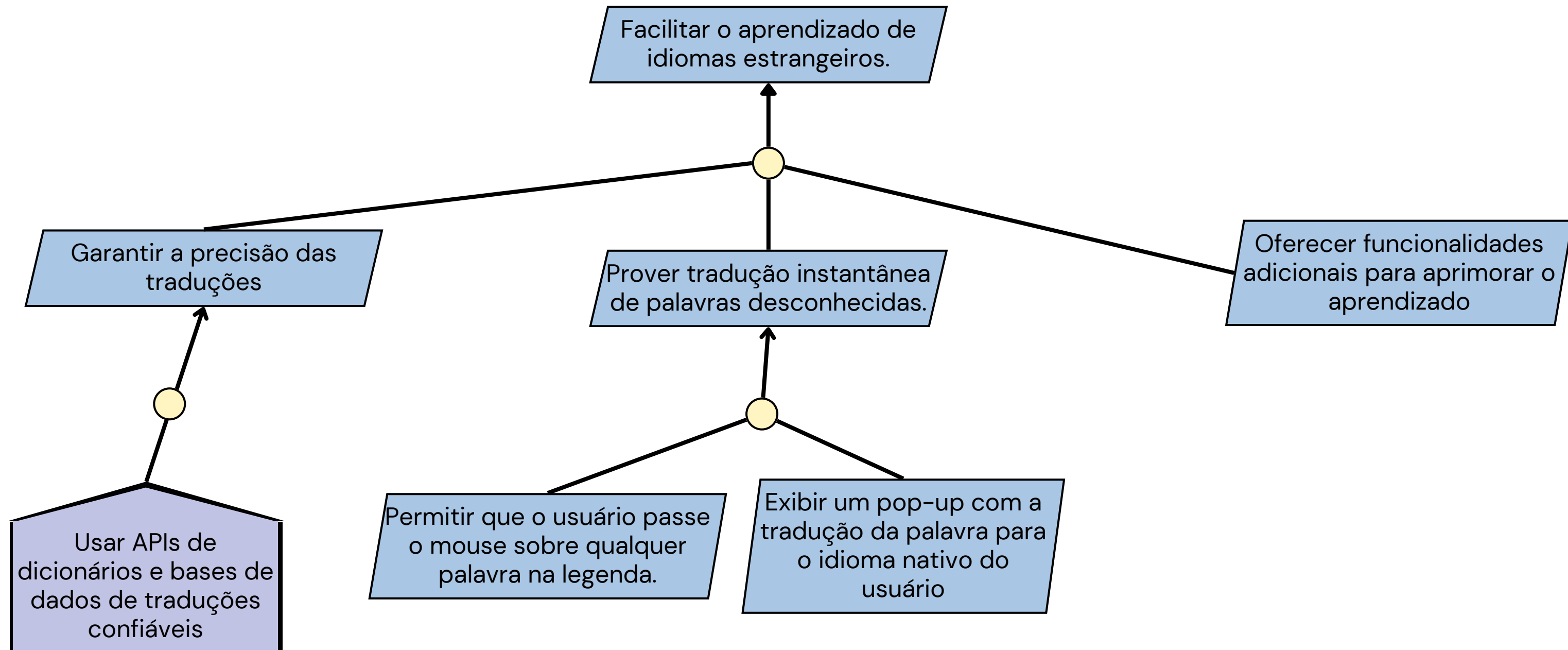
# Exemplo



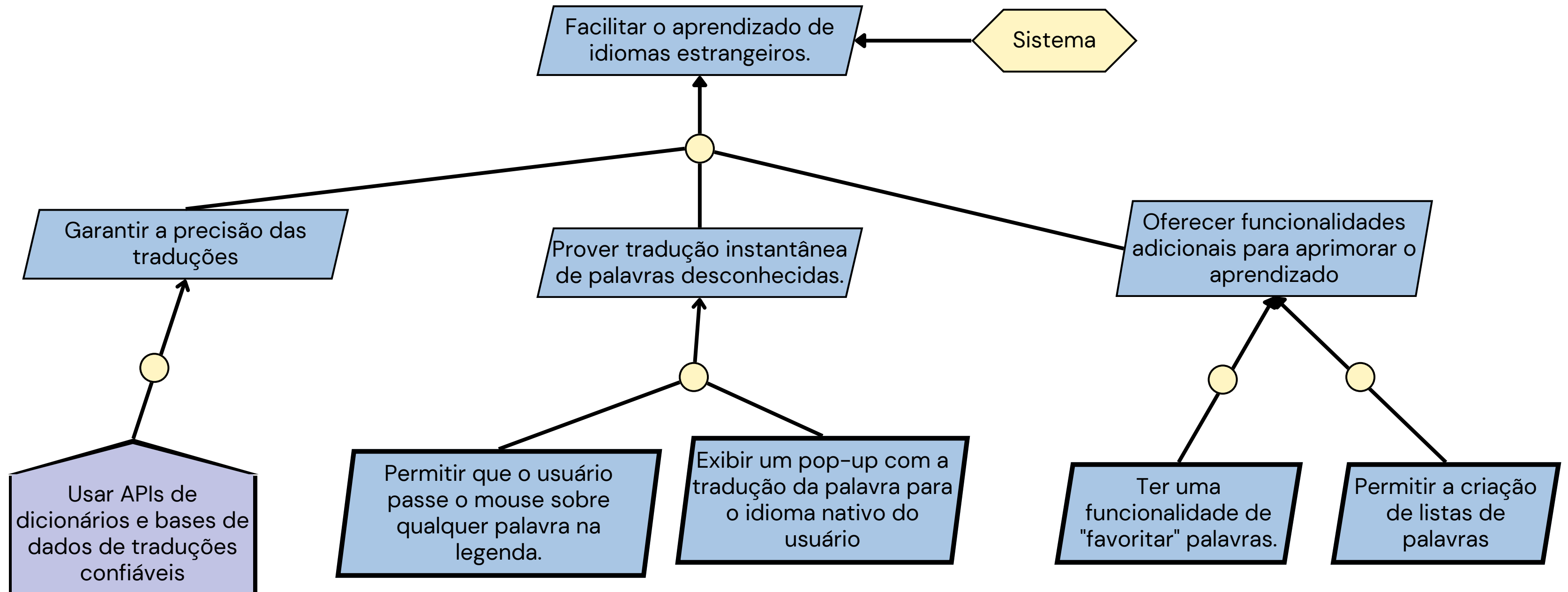
# Exemplo



# Exemplo

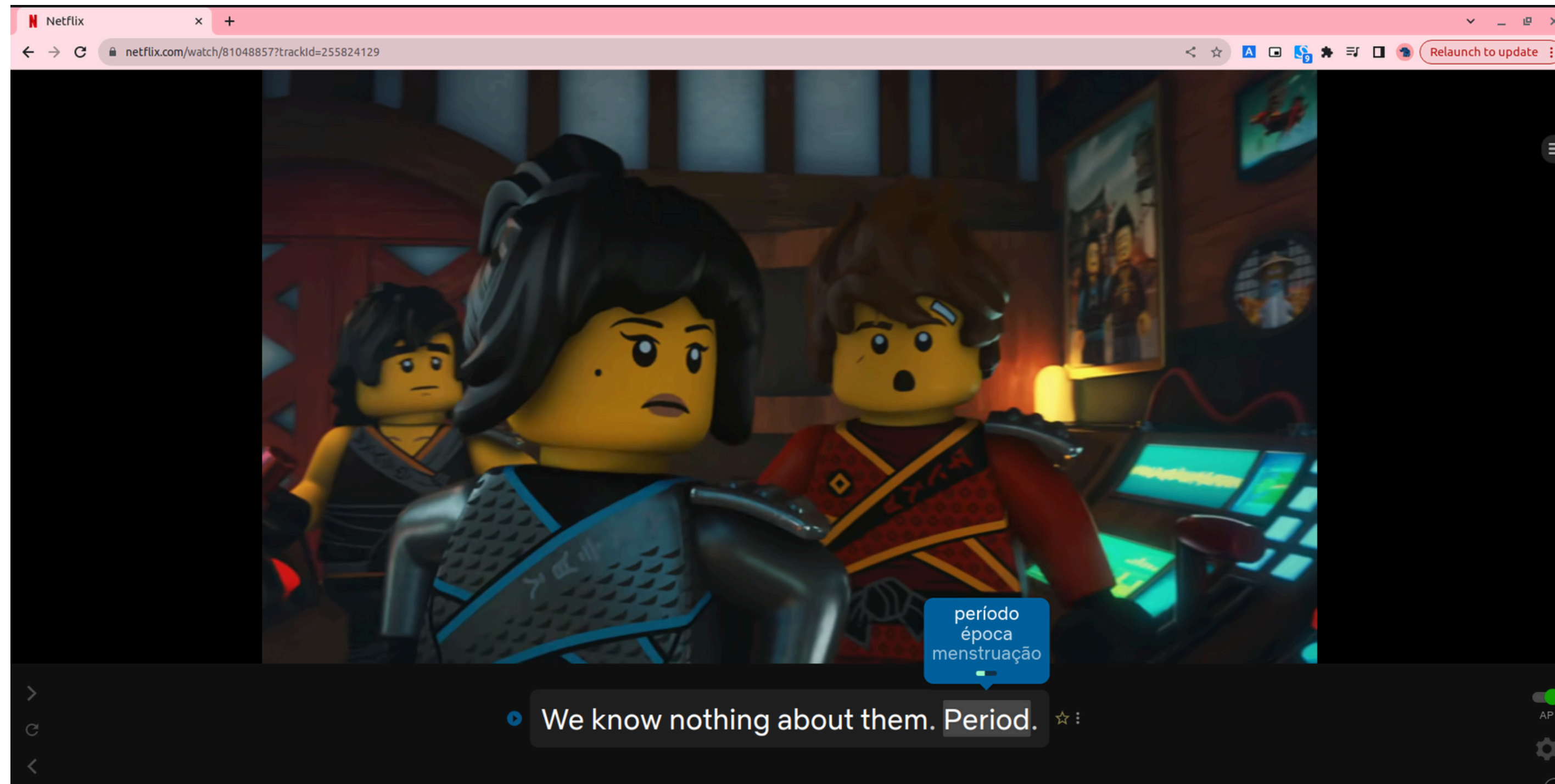


# Exemplo



# Exemplo com IA

Nem sempre ao usar o dicionário, a tradução virá precisamente. Logo, ela precisa ser contextualizada.



# Exemplo com IA

**ReadLang** é uma aplicação onde você pode ler textos em línguas estrangeiras. Ao clicar em uma palavra ele oferece a tradução e um texto gerado por IA a explicando no contexto proposto.


composta da mia mamma, mio papà e i miei tre fratelli . Alle 22:30 circa vado a letto a leggere alcuni libri prima di dormire.

brothers

**fratelli** - brothers

💡 The Italian word "fratelli" means "brothers" in English. In its usual meaning, it refers to male siblings or brothers in a general sense. In the given context, it specifically refers to the speaker's three brothers, indicating that the speaker has three male siblings.

[Switch explanations to Italian](#)

 Edit word

# Áreas-chave para o desenvolvimento de sistemas de IA

## User Needs

Esta área se concentra na identificação e análise dos requisitos dos usuários para sistemas de IA, incluindo as capacidades do sistema, as interações dos usuários e se o sistema deve assistir ou automatizar tarefas. Ele enfatiza a escolha de métricas de avaliação apropriadas, considerando trade-offs.

## Model Needs

Esta área aborda a seleção e o treinamento do modelo de IA com base nas necessidades das partes interessadas, como explicabilidade vs precisão. Envolve documentar os dados usados no ajuste do modelo para mitigar preconceitos e decidir entre treinamento estático versus dinâmico.

## Data Needs

Esta área cobre a identificação de fontes de dados, tipos, qualidade, rótulos, recursos, precisão e exatidão. Enfatiza a garantia da justiça e inclusão dos dados e do equilíbrio entre o custo de aquisição de mais dados e os benefícios que isso proporciona.



# Áreas-chave para o desenvolvimento de sistemas de IA

## Feedback and User Control

Esta área envolve a obtenção de feedback explícito e implícito dos usuários e permitir-lhes controlar e ajustar o sistema. Inclui o planejamento de mecanismos de feedback e a garantia de que os usuários possam calibrar as configurações iniciais.

## Explainability and Trust

Esta área se concentra em fornecer explicações para decisões do sistema de IA para construir a confiança do usuário. Envolve explicar resultados e previsões, exibir níveis de confiança e adaptar as explicações ao contexto para evitar excesso de confiança ou mal-entendidos.

## Errors and Failure

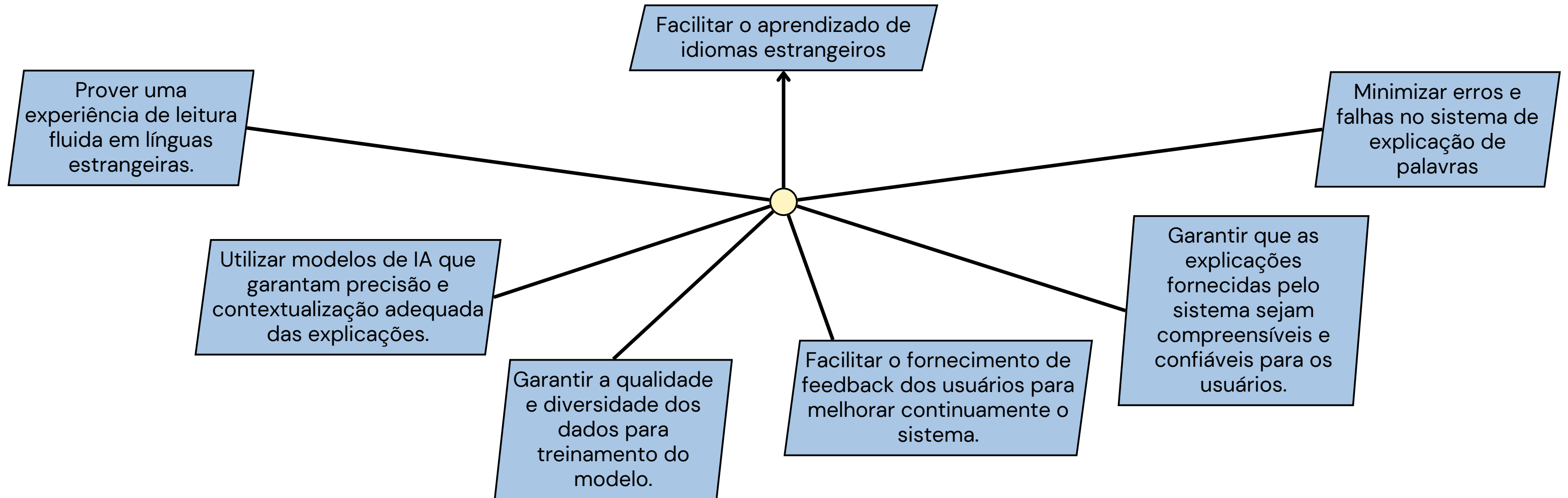
Esta área aborda a identificação e definição de erros, suas fontes e o impacto nos usuários. Inclui planejamento para correção de erros, mitigação dos efeitos dos erros e cautela com suposições baseadas em dados confidenciais para manter a confiança do usuário.



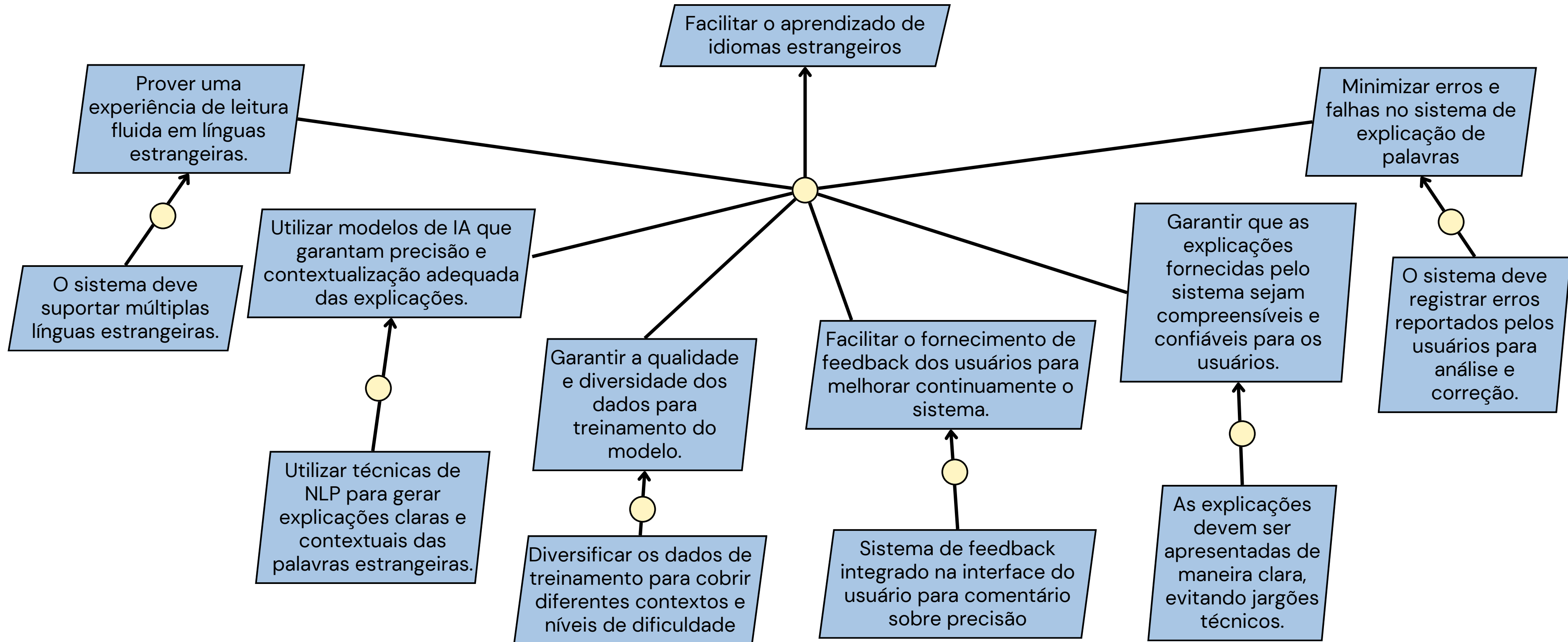
# Exemplo com IA

Facilitar o aprendizado de idiomas estrangeiros

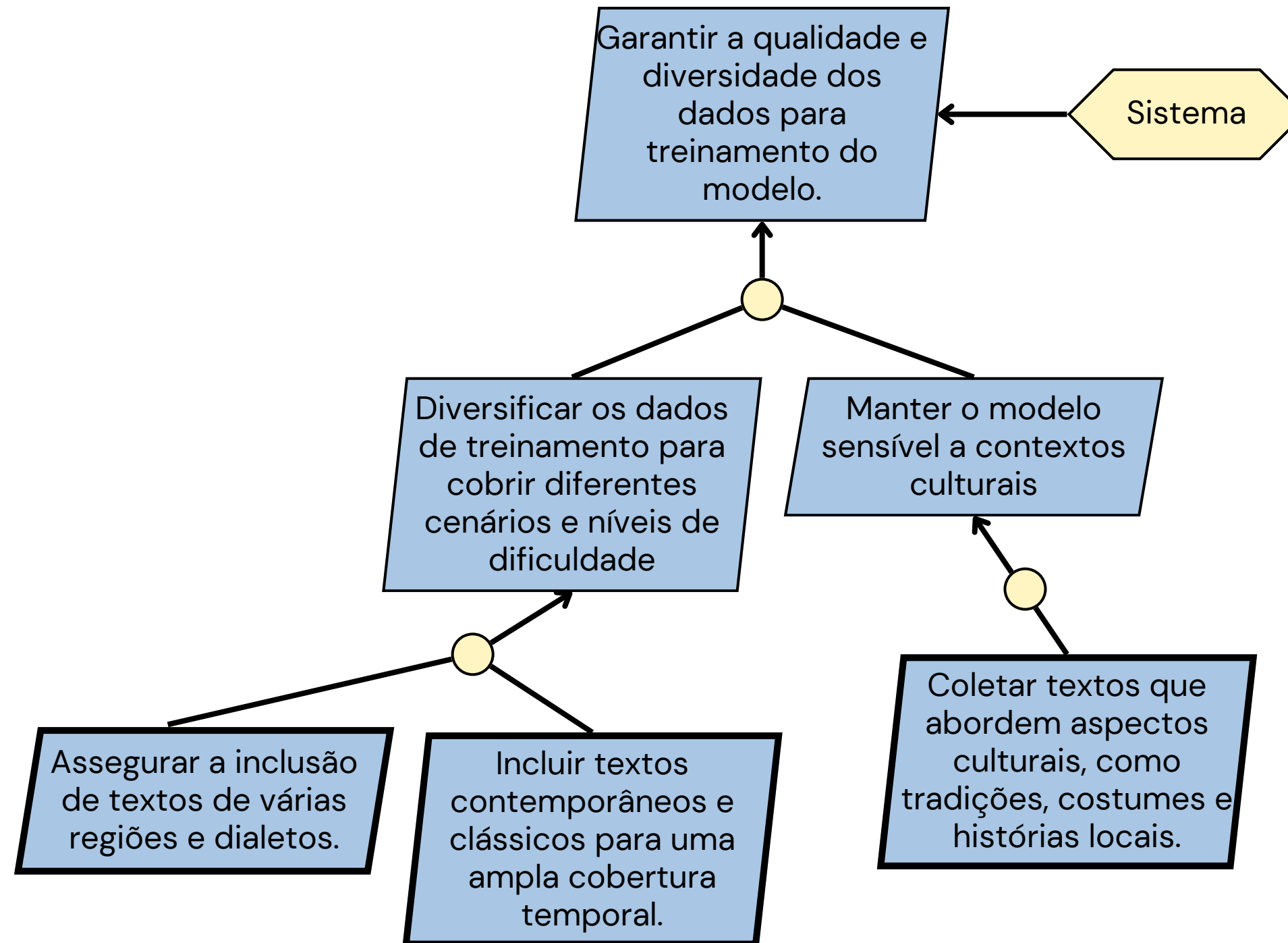
# Exemplo com IA



# Exemplo com IA



# Exemplo com IA



# Benefícios do GORE

**1**

**Rastreabilidade e  
refinamento**

**2**

**Persistencia dos  
requisitos**

**3**

**Comunicação com  
Stakeholder**

**4**

**Estabilidade da  
informação**

# Bibliografia

- Aljahdali, Sultan, Jameela, Bano, Nisar, Hundewale. "Goal Oriented Requirements Engineering –A Review". Proceedings of the ISCA 24th International Conference on Computer Applications in Industry and Engineering, CAINE 2011. (2011).
- Van Lamsweerde, Axel. "Goal-oriented requirements engineering: A guided tour." Proceedings fifth ieee international symposium on requirements engineering. IEEE, 2001.
- Lapouchnian, Alexei. "Goal-oriented requirements engineering: An overview of the current research." University of Toronto 32 (2005).
- Uszok, Jeffrey Bradshaw Andrzej, and Jeffrey Bradshaw. "Kaos tutorial." URL [http://ontology.ihmc.us/KAoS/KAoS\\_Tutorial\\_files/frame.html](http://ontology.ihmc.us/KAoS/KAoS_Tutorial_files/frame.html).