

## 5 Re-arquitetura do software C&L

Neste capítulo apresentamos as tecnologias utilizadas na construção do novo C&L e a sua nova arquitetura. Também mostraremos, passo a passo, a aplicação do método proposto na sua construção.

### 5.1. Arquitetura do novo C&L e tecnologias utilizadas

Escolhemos a linguagem Lua [Ierusalimschy03] para implementação da ferramenta. Esta escolha foi motivada por três motivos principais. O primeiro deles é que Lua provê um excelente suporte a programação funcional e acreditamos que o uso de funções se adequa mais a descrição por cenários proposta pelo nosso método. O Segundo motivo é que Lua é uma linguagem cada vez mais utilizada, e acreditamos que suas principais características (livre, rápida, extensível e pequena, dentre outras) sejam bastante úteis quando aplicadas ao desenvolvimento de sistemas Web. E por último, o C&L faz muitas comparações baseadas em expressões regulares para determinar o relacionamento entre elementos. Lua oferece um excelente suporte ao uso de expressões regulares e comparações entre *strings*.

Lua não foi projetada para desenvolvimento de sistema Web, por isso utilizamos a plataforma Kepler [Kepler09]. Esta plataforma disponibiliza uma série de módulos que facilitam o desenvolvimento de código Lua para Web. Dentre os módulos podemos destacar o CGI Lua que permite a criação de páginas dinâmicas para Web e a manipulação de dados provenientes de formulário, o LuaSQL que fornece uma interface simples para interação de Lua com alguns dos principais sistemas gerenciadores de banco de dados e, por último, o Xavante, que é um poderoso servidor Web HTTP 1.1, que usa uma arquitetura modular baseada em tratadores URI mapeados.

Para a construção da nova arquitetura adotamos o framework MVC, pois acreditamos que a divisão em camadas proporcionada por ele favorece a organização e entendimento do código fonte do software, proporcionando uma maior transparência a nível de código. Em nossa arquitetura existe a divisão física

do sistema em camadas e a divisão lógica em módulos. Cada um dos módulos está distribuído pelas camadas de visão, controle e modelo, como mostra a Figura 38.

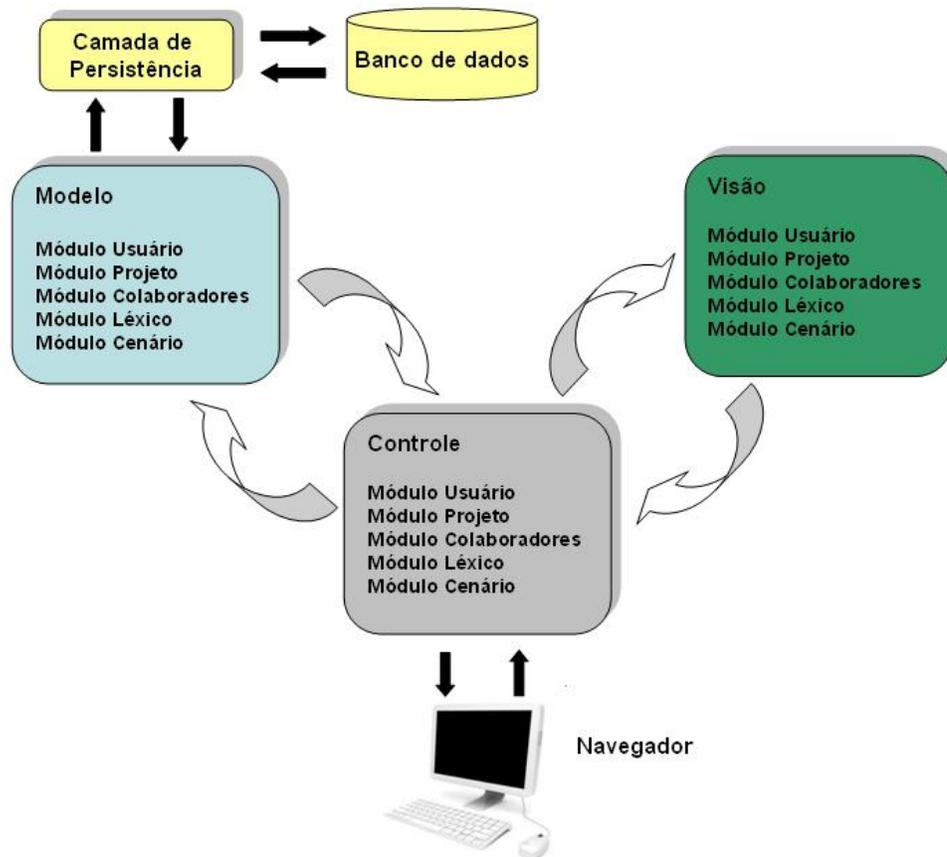


Figura 38 – Arquitetura do novo C&L.

Como ferramenta de apoio à aplicação do método proposto utilizamos o próprio C&L. No princípio, utilizamos a versão antiga para construir os cenários iniciais e o léxico ampliado da linguagem. Conforme o desenvolvimento do novo C&L foi avançando, pudemos migrar os cenários para a nova ferramenta. A partir daí, utilizamos o novo C&L para evoluir os cenários iniciais e os símbolos do léxico.

## 5.2.

### Aplicação do método na construção do novo C&L

Detalharemos a aplicação do método proposto, passo a passo, nas subseções a seguir.

### 5.2.1.

#### Descrição das situações do sistema através de cenários

A seguir apresentamos a descrição das situações do software C&L com o uso do modelo de cenários apresentado anteriormente. Os termos sublinhados ou são referências a outros cenários ou a símbolos do léxico.

**Título:** CADASTRAR USUÁRIO

**Objetivo:** Cadastrar usuário no sistema através da inserção de seus dados no banco de dados.

**Contexto:** Usuário não possui cadastro no sistema.

**Recursos:** dados do usuário (nome, sobrenome, e-mail, instituição, login e senha), banco de dados.

**Atores:** camada de controle

**Episódios:**

- 1- Receber os dados do usuário repassados pela camada de controle.
- 2- Conectar ao banco de dados. **Restrição:** Banco de dados deve estar disponível.
- 3- Inserir dados do usuário no banco de dados.
- 4- Desconectar do banco de dados.
- 5- Informar a camada de controle que os dados foram inseridos com sucesso no banco de dados.

**Exceção:**

O login informado pelo usuário já se encontra no banco de dados do sistema. Informar a camada de controle que os dados não foram inseridos porque o login já existe.

**Título:** ALTERAR DADOS DO USUÁRIO

**Objetivo:** Permitir que o usuário altere seus dados armazenados no sistema, através da edição de um formulário contendo as informações previamente cadastradas.

**Contexto:** Usuário deve ter sido previamente cadastrado no sistema.

**AUTENTICAR USUÁRIO NO SISTEMA.**

**Recursos:** novos dados do usuário.

**Atores:** usuário e sistema.

**Episódios:**

- 1- Sistema exibe um formulário com os seguintes campos já preenchidos, permitindo sua edição: nome, sobrenome, e-mail, instituição.
- 2- Usuário submete o formulário preenchido com os novos dados.  
**Restrição:** O formulário não pode conter campos em branco.
- 3- Sistema atualiza os dados do usuário no banco de dados.
- 4- Sistema avisa que os dados do usuário foram alterados com sucesso e o redireciona para a página principal.

**Título:** ACESSAR O SISTEMA

**Objetivo:** Permitir que o usuário utilize o sistema se identificando através do preenchimento de um pequeno formulário com seu login e senha.

**Contexto:** Usuário deve ter sido previamente cadastrado no sistema.

**Recursos:** login, senha.

**Atores:** usuário, sistema.

**Episódios:**

- 1- Sistema exibe para o usuário um formulário contendo dois campos: login e senha.
- 2- Usuário submete o formulário com os campos preenchidos. **Restrição:** Nenhum dos campos pode ser deixado em branco.
- 3- O sistema carrega os projetos que o usuário participa e o redireciona para a página principal.

**Exceção:**

O login e senha informada pelo usuário não estão corretos. O sistema permite que o usuário tente novamente por mais duas vezes. Se o usuário errar seu login e senha por mais duas vezes ele terá que esperar 20 minutos para tentar novamente.

**Título:** RECUPERAR SENHA DO USUÁRIO

**Objetivo:** Permitir que o usuário recupere sua senha. Ao solicitar recuperação de sua senha o usuário preencherá um formulário com seu login e e-mail. Após a verificação dos dados o sistema irá substituir a senha antiga por uma nova senha randômica e enviá-la para o e-mail do usuário.

**Contexto:** Usuário deve ter sido previamente cadastrado no sistema

**Recursos:** login e e-mail do usuário.

**Atores:** usuário e sistema.

**Episódios:**

- 1- O sistema exibe um formulário contendo os campos e-mail e login para o usuário preencher.
- 2- O usuário submete o formulário preenchido. **Restrição:** Todos os campos devem ser preenchidos.
- 3- O sistema verifica se o login e e-mail informado são iguais aos do cadastro do usuário.
- 4- Sistema gera uma nova senha randômica e a envia para o e-mail do usuário.
- 5- Sistema exibe um aviso informando que a senha foi alterada e que o usuário deve verificar o e-mail informado para obter a nova senha.

**Exceção:**

O login e e-mail informado pelo usuário não estão corretos. O sistema informará ao usuário que o login e o e-mail não estão corretos e permitirá que ele tente mais duas vezes. Caso erre mais duas vezes o usuário terá que esperar por 20 minutos para tentar novamente.

**Título:** SAIR DO SISTEMA

**Objetivo:** Permitir que o usuário saia do sistema, mantendo a integridade das mudanças realizadas.

**Contexto:** Usuário deve ter acessado o sistema.

**Recursos:** dados da sessão.

**Atores:** usuário, sistema.

**Episódios:**

- 1- Usuário seleciona a opção “sair do sistema”.
- 2- O sistema encerra a sessão atual do usuário, mantendo a integridade das mudanças realizadas.
- 3- O sistema redireciona o usuário para a página principal.

**Título:** CADASTRAR PROJETO.

**Objetivo:** Permitir que o usuário cadastre um novo projeto no sistema através da inclusão dos seguintes dados do projeto no banco de dados: nome, descrição e data da inclusão. Os campos nome e descrição devem ser informados pelo usuário através do preenchimento de um formulário. O sistema deve preencher automaticamente o campo data da inclusão com a data atual.

**Contexto:** Usuário deve estar autenticado no sistema.

**Recursos:** dados do projeto (nome, descrição e data da inclusão).

**Atores:** usuário, sistema.

**Episódios:**

- 1- Sistema exibe um formulário de cadastro de projeto, contendo os campos nome e descrição.
- 2- Usuário submete o formulário preenchido. **Restrição:** todos os campos devem estar preenchidos.
- 3- Sistema armazena a data atual, como data da inclusão do projeto.
- 4- Sistema insere os seguintes dados no banco de dados: nome, descrição e data da inclusão.
- 5- Sistema avisa ao usuário que o projeto foi inserido com sucesso e o redireciona para página principal.

**Exceção:**

O usuário já possui um projeto com o mesmo nome informado. O usuário é avisado que deve escolher um novo nome para o projeto.

**Título:** ALTERAR PROJETO.

**Objetivo:** Permitir que o usuário altere a descrição de um projeto já cadastrado, através da edição de um formulário contendo a descrição previamente cadastrada.

**Contexto:** Projeto deve ter sido cadastrado pelo usuário (usuário deve ser administrador).

**Recursos:** descrição do projeto.

**Atores:** usuário, sistema.

**Episódios:**

- 1- Sistema exibe para o usuário um formulário contendo os campos nome, data de inclusão e descrição do projeto, já preenchidos e permite a edição apenas do campo descrição.
- 2- Usuário submete o formulário com a nova descrição do projeto.  
**Restrição:** O campo descrição não pode estar vazio e nem conter apenas espaços em branco.
- 3- Sistema atualiza a descrição do projeto no banco de dados.
- 4- Sistema comunica o usuário que os dados do projeto foram atualizados com sucesso e o redirecionam para a página principal do sistema.

**Título:** REMOVER PROJETO.

**Objetivo:** Permitir que o usuário remova um projeto, seus cenários, símbolos do léxico e arquivos XML vinculados.

**Contexto:** Projeto cadastrado anteriormente pelo usuário (usuário deve ser administrador).

**Recursos:** dados do projeto.

**Atores:** sistema, usuário.

**Episódios:**

- 1- O sistema exibe os dados do projeto e solicita uma confirmação de exclusão para o usuário.
- 2- Usuário confirma exclusão.
- 3- Sistema exclui o projeto, os símbolos do léxico, cenários e arquivos XML vinculados ao projeto.
- 4- Sistema comunica ao usuário que a exclusão foi efetuada com sucesso.

**Título:** SELECIONAR PROJETO.

**Objetivo:** Permitir que o usuário trabalhe com um projeto específico, previamente cadastrado no sistema. O Sistema irá carregar os dados referentes ao projeto e exibirá no menu lateral todos os símbolos do léxico e cenários já cadastrados no projeto.

**Contexto:** Usuário deve possuir um projeto cadastrado ou ser colaborador de um.

**Recursos:** projetos criados pelo usuário e projetos em que o usuário é colaborador.

**Atores:** usuário, sistema.

**Episódios:**

- 1- O sistema exibe uma lista dos projetos acessíveis para o usuário, destacando o seu nível de acesso em cada um deles.
- 2- Usuário seleciona um dos projetos da lista.
- 3- Os cenários e símbolos do léxico previamente cadastrados são exibidos em um menu lateral.
- 4- As seguintes opções de tornam disponíveis para o usuário: cadastrar léxico, cadastrar cenário e gerar grafo do projeto.
- 5- Caso o usuário possua nível de acesso de administrador ou gerente as seguintes opções surgiriam, além das anteriores: GERAR XML DO PROJETO, gerenciar colaboradores do projeto e verificar alterações propostas.

**Título:** GERAR XML DO PROJETO.

**Objetivo:** Permitir que o usuário crie um arquivo XML contendo todas as informações referentes ao projeto, seus símbolos do léxico e cenários e os relacionamentos entre eles.

**Contexto:** Usuário deve selecionar um projeto do menu de projetos.

**Recursos:** dados do projeto, cenários e símbolos do léxico pertencentes ao projeto.

**Atores:** usuário, sistema.

**Episódios:**

- 1- Usuário seleciona a opção “Gerar XML do projeto”
- 2- Sistema atribui automaticamente a data de criação e número de versão para o arquivo XML que será gerado.
- 3- Usuário escolhe se deseja visualizar o arquivo XML formatado através de um arquivo XSL ou o arquivo XML sem formatação.
- 4- Sistema gera o arquivo XML, com todas as informações referentes ao projeto.
- 5- Sistema armazena o arquivo XML gerado no banco de dados.
- 6- Sistema exibe o arquivo XML gerado para o usuário.

**Título:** GERAR GRAFO DO PROJETO.

**Objetivo:** Gerar um grafo do projeto a partir de um dos arquivos XML do projeto, onde os cenários e símbolos do léxico serão representados como nós e seus relacionamentos como arestas.

**Contexto:** Um arquivo XML do projeto deve ter sido gerado.

**Recursos:** arquivo XML.

**Atores:** usuário, sistema, ferramenta de construção de grafos.

**Episódios:**

- 1- Usuário seleciona a opção “Gerar grafo do projeto”.
- 2- Sistema mostra uma lista com as versões de arquivo XML geradas para o projeto selecionado.
- 3- Usuário seleciona uma das versões.
- 4- Sistema chama a ferramenta de construção de grafos e passa como parâmetro o arquivo XML escolhido. A ferramenta de construção de grafos constrói o grafo do projeto e o exibe para o usuário.

**Título:** INCLUIR COLABORADOR NO PROJETO.

**Objetivo:** Permitir a inclusão de colaboradores para ajudar na construção e evolução do projeto. A inclusão é feita através de uma lista contendo os usuários

cadastrados no sistema. Os colaboradores podem possuir os seguintes níveis de acesso: gerente, usuário participativo e usuário passivo. O gerente possui todos os privilégios do administrador do projeto, exceto adicionar outros usuários colaboradores como nível de acesso de gerente. Os usuários participativos podem sugerir alterações, que podem ser aceitas ou não pelo gerente ou administrador. O usuário passivo só pode visualizar os itens do projeto.

**Contexto:** O usuário deve estar autenticado como administrador do projeto.

**Recursos:** login dos usuários cadastrados no sistema, dados do projeto selecionado.

**Atores:** usuário, sistema.

**Episódios:**

- 1- Usuário seleciona a opção “Adicionar colaborador ao projeto”.
- 2- Sistema exibe uma tela com login de todos usuários cadastrados no sistema e com os níveis de acesso possíveis.
- 3- Usuário seleciona um usuário da lista e o inclui na lista de colaboradores do projeto. **Restrição:** O usuário deve selecionar um nível de acesso para o usuário escolhido.
- 4- Sistema inclui usuário como colaborador e permite que ele acesse o projeto.
- 5- Sistema emite um aviso informando o sucesso da operação.

**Título:** REMOVER COLABORADOR DO PROJETO.

**Objetivo:** Permitir a remoção de um colaborador do projeto.

**Contexto:** Usuário deve estar autenticado como gerente ou administrador do projeto.

**Recursos:** lista de colaboradores do projeto, dados do projeto.

**Atores:** sistema, usuário.

**Episódios:**

- 1- Sistema seleciona a opção “Remover colaborador do projeto”.
- 2- Sistema exibe uma lista com os colaboradores do projeto, permitindo que eles sejam removidos.
- 3- Usuário remove os colaboradores que desejar e clica em “confirmar atualizações”.
- 4- Sistema atualiza a lista de colaboradores.
- 5- Sistema emite um aviso informando o sucesso da operação.

**Título:** VERIFICAR ALTERAÇÕES SUGERIDAS PELOS COLABORADORES.

**Objetivo:** Permitir que o administrador ou gerente do projeto acate ou rejeite as alterações referentes a símbolos do léxico e cenários, sugeridas pelos usuários participativos.

**Contexto:** Usuário deve ter sido autenticado como administrador ou gerente do projeto.

**Recursos:** lista de alterações sugeridas e justificativas.

**Atores:** usuário, sistema.

**Episódios:**

- 1- Usuário seleciona a opção “Verificar alterações sugeridas”
- 2- O sistema exibe uma lista com todas as alterações sugeridas, tanto para léxicos quanto para cenários. Abaixo de cada alteração sugerida da lista são exibidas duas opções “Aceitar” e “Rejeitar”.
- 3- O usuário seleciona uma das duas opções para cada alteração e clica no botão “Processar”.
- 4- O sistema realmente efetua as alterações acatadas e rejeita as que não foram acatadas pelo usuário.
- 5- Sistema exibe uma tela de sucesso do processamento das alterações.

**Título:** CADASTRAR SÍMBOLO DO LÉXICO

**Objetivo:** Permitir que o usuário inclua um símbolo do léxico em um projeto.

**Contexto:** Projeto cadastrado no sistema.

**Recursos:** dados do léxico.

**Atores:** usuário, sistema.

**Episódios:**

- 1- Usuário seleciona a opção “Novo símbolo do léxico”.
- 2- O sistema exibe um formulário com os seguintes campos: nome, noção, impacto, classificação e sinônimo.
- 3- Usuário submete o formulário preenchido. **Restrição:** Os campos nome, noção e classificação não podem ser deixados em branco.
- 4- Sistema cadastra o novo símbolo do léxico.
- 5- Sistema comunica ao usuário que a inclusão foi realizada com sucesso.

**Exceção:** Se o nome do léxico escolhido ou algum de seus sinônimos já estiver cadastrado no sistema como nome de outro léxico ou sinônimo, então o sistema deve exibir uma mensagem de erro, identificando o que o ocasionou.

**Título:** EXIBIR SÍMBOLO DO LÉXICO E SEUS RELACIONAMENTOS.

**Objetivo:** Exibir as informações de um símbolo do léxico juntamente com seus relacionamentos, utilizando elos para permitir o acesso a outros símbolos referenciados pelo que esta sendo exibido e permitir o acesso a outros elementos do projeto que referenciem o símbolo exibido.

**Contexto:** Símbolo do léxico cadastrado no sistema.

**Recursos:** dados do léxico, nomes e sinônimos de símbolos do léxico e títulos de cenários do projeto.

**Atores:** usuário, sistema.

**Episódios:**

- 1- O usuário seleciona um símbolo do léxico do projeto, exibido no menu lateral.
- 2- O sistema monta a rede de relacionamentos em torno do símbolo selecionado.
- 3- Sistema exibe as informações do símbolo e os relacionamentos mapeados através de elos.

**Título:** ALTERAR SÍMBOLO DO LÉXICO.

**Objetivo:** Permitir que o usuário altere os dados de um símbolo do léxico cadastrado no sistema, através da edição de um formulário preenchido com suas informações cadastradas.

**Contexto:** Símbolo do léxico cadastrado no sistema.

**Recursos:** novos dados do símbolo do léxico.

**Atores:** usuário, sistema.

**Episódios:**

- 1- O usuário seleciona um símbolo do léxico do projeto, exibido no menu lateral.
- 2- O usuário seleciona a opção “Alterar símbolo”
- 3- Sistema exibe um formulário preenchido com dos dados do léxico selecionado.
- 4- Usuário faz as alterações que deseja nas informações do símbolo e submete o formulário. **Restrição:** O campo noção não pode ser deixado em branco.
- 5- O sistema comunica ao usuário que as alterações no símbolo foram feitas com sucesso.

**Título:** REMOVER SÍMBOLO DO LÉXICO.

**Objetivo:** Permitir que o usuário remova um símbolo do léxico do projeto selecionado.

**Contexto:** Símbolo do léxico cadastrado no sistema.

**Recursos:** dados do símbolo do léxico.

**Atores:** usuário, sistema.

**Episódios:**

- 1- O usuário seleciona um símbolo do léxico do projeto, exibido no menu lateral.
- 2- O usuário seleciona a opção “Remover símbolo”
- 3- Sistema solicita a confirmação da exclusão.
- 4- Usuário confirma a exclusão.
- 5- Sistema comunica o usuário que a exclusão do símbolo foi realizada com sucesso.

**Título:** CADASTRAR CENÁRIO

**Objetivo:** Permitir que o usuário inclua um cenário em um projeto.

**Contexto:** Projeto cadastrado no sistema.

**Recursos:** dados do cenário.

**Atores:** usuário, sistema.

**Episódios:**

- 1- Usuário seleciona a opção “Novo cenário”.
- 2- O sistema exibe um formulário com os seguintes campos: título, objetivo, contexto, recursos, atores, episódios e exceção.
- 3- Usuário submete o formulário preenchido. **Restrição:** Os campos título, objetivo, contexto, recursos, atores e episódios não podem ser deixados em branco.
- 4- Sistema cadastra o novo cenário no projeto.
- 5- Sistema comunica ao usuário que a inclusão do cenário foi realizada com sucesso.

**Exceção:** Se o título escolhido para o cenário escolhido já estiver cadastrado no sistema como título de outro cenário ou nome/sinônimo de um léxico, então o sistema deve exibir uma mensagem de erro, identificando o que o ocasionou.

**Título:** EXIBIR CENÁRIO E SEUS RELACIONAMENTOS.

**Objetivo:** Exibir as informações de um cenário juntamente com seus relacionamentos, utilizando elos para permitir o acesso a outros elementos

referenciados pelo cenário que esta sendo exibido e permitir o acesso a outros elementos do projeto que referenciem o cenário exibido.

**Contexto:** Cenário cadastrado no sistema.

**Recursos:** dados do cenário, nomes e sinônimos de símbolos do léxico e títulos de cenários do projeto.

**Atores:** usuário, sistema.

**Episódios:**

- 1- O usuário seleciona um cenário do projeto, exibido no menu lateral.
- 2- O sistema monta a rede de relacionamentos em torno do cenário selecionado.
- 3- Sistema exibe as informações do cenário e os relacionamentos mapeados através de elos.

**Título:** ALTERAR CENÁRIO.

**Objetivo:** Permitir que o usuário altere os dados de um cenário cadastrado no sistema, através da edição de um formulário preenchido com suas informações cadastradas.

**Contexto:** Cenário cadastrado no sistema.

**Recursos:** novos dados do cenário.

**Atores:** usuário, sistema.

**Episódios:**

- 1- O usuário seleciona um cenário do projeto, exibido no menu lateral.
- 2- O usuário seleciona a opção “Alterar cenário”
- 3- Sistema exibe um formulário preenchido com os dados do cenário selecionado.
- 4- Usuário faz as alterações que deseja nas informações do cenário e submete o formulário. **Restrição:** Os campos objetivo, contexto, recursos, atores e episódios não podem ser deixados em branco.
- 5- O sistema comunica ao usuário que as alterações no cenário foram feitas com sucesso.

**Título:** REMOVER CENÁRIO

**Objetivo:** Permitir que o usuário remova um cenário do projeto selecionado.

**Contexto:** Cenário cadastrado no sistema.

**Recursos:** dados do cenário

**Atores:** usuário, sistema.

**Episódios:**

- 1- O usuário seleciona um cenário do projeto, exibido no menu lateral.
- 2- O usuário seleciona a opção “Remover cenário”
- 3- Sistema solicita a confirmação da exclusão.
- 4- Usuário confirma a exclusão.
- 5- Sistema comunica o usuário que a exclusão do cenário foi realizada com sucesso.

### **5.2.2. Divisão dos cenários em grupos**

Foram formados cinco grupos a partir dos cenários que descrevem as situações do sistema. O primeiro grupo (Figura 39) é formado pelos cenários: cadastrar usuário, recuperar senha do usuário, acessar o sistema, alterar dados do usuário e sair do sistema. Estes cenários foram colocados no mesmo grupo, pois considerando uma visão modular do sistema pertenceriam ao módulo usuário, já que todos descrevem situações relacionadas aos dados do usuário.

O segundo grupo (Figura 40) é formado pelos cenários: cadastrar projeto, alterar projeto, remover projeto, selecionar projeto, gerar XML do projeto e gerar grafo do projeto. Estes cenários descrevem situações relacionadas a projetos do sistema, e corresponderiam ao módulo projeto, por isso foram colocados no mesmo grupo.

O terceiro grupo (Figura 41), formado pelos cenários: incluir colaborador no projeto, remover colaborador do projeto e verificar alterações sugeridas pelos colaboradores, corresponde ao módulo de gerência de colaboradores. As situações descritas pelos cenários destes grupos estão relacionadas ao controle da interação entre os usuários nos projetos.

O quarto grupo (Figura 42) corresponde ao módulo léxico do sistema. Os cenários que compõem este grupo são: cadastrar símbolo do léxico, exibir símbolo do léxico e seus relacionamentos, alterar símbolo do léxico e remover símbolo do léxico. Estes cenários descrevem situações relacionadas a símbolos do léxico.

O quinto e último grupo (Figura 43) corresponde ao módulo cenário. Este grupo é formado pelos cenários: cadastrar cenário, exibir cenário e seus relacionamentos, alterar cenário e remover cenário. Estes cenários foram colocados no mesmo grupo, pois descrevem situações relacionadas a cenários.



Figura 39 – Grupo correspondente ao módulo usuário.

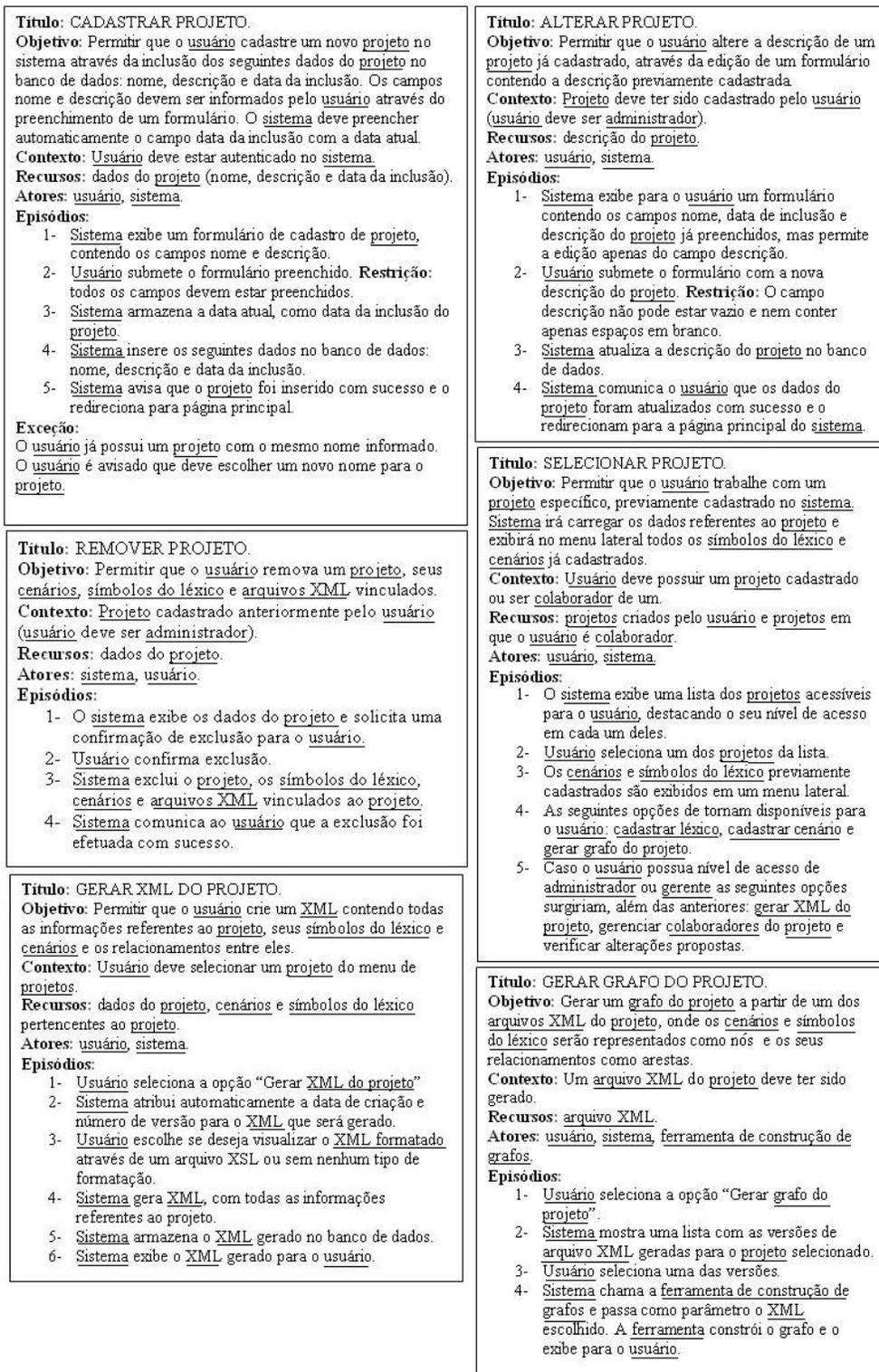


Figura 40 – Grupo correspondente ao módulo projeto.

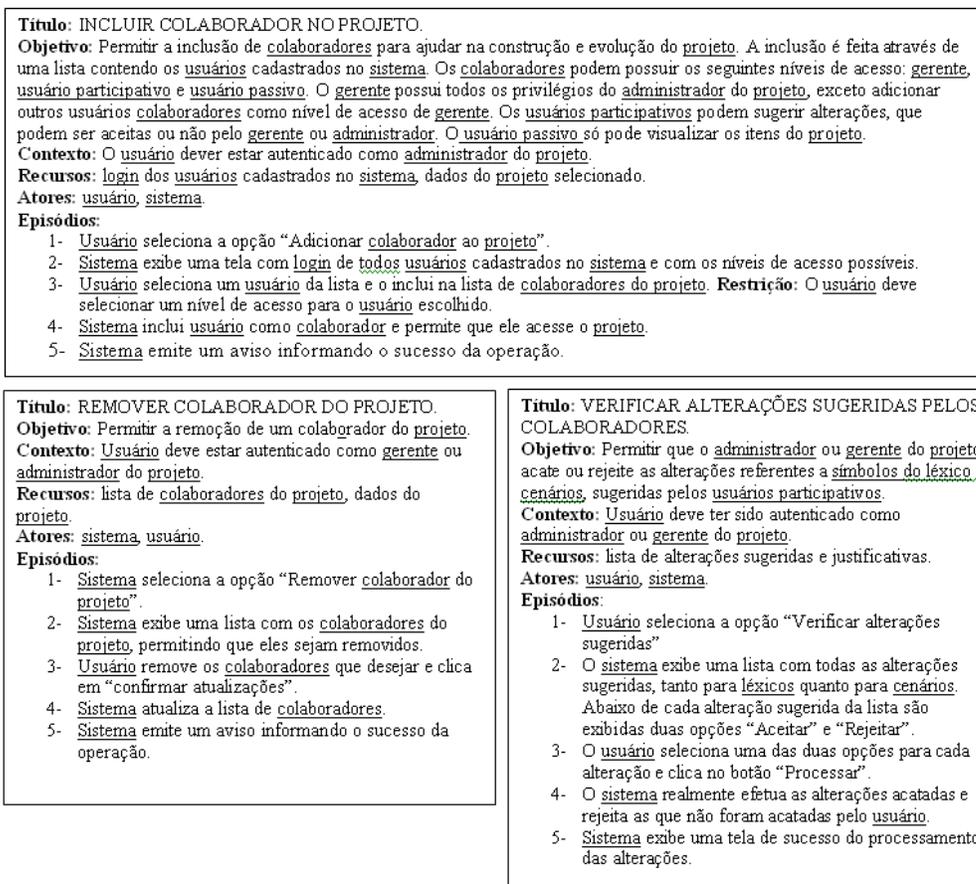


Figura 41 – Grupo correspondente ao módulo administração de colaboradores.

<p><b>Título:</b> CADASTRAR SÍMBOLO DO LÉXICO  <b>Objetivo:</b> Permitir que o usuário inclua um <u>símbolo do léxico</u> em um <u>projeto</u>.  <b>Contexto:</b> Projeto cadastrado no <u>sistema</u>.  <b>Recursos:</b> dados do <u>léxico</u>.  <b>Atores:</b> <u>usuário, sistema</u>.  <b>Episódios:</b></p> <ol style="list-style-type: none"> <li>1- <u>Usuário</u> seleciona a opção “Novo <u>símbolo do léxico</u>”.</li> <li>2- O <u>sistema</u> exibe um formulário com os seguintes campos: nome, <u>noção</u>, <u>impacto</u>, <u>classificação</u> e <u>sinônimo</u>.</li> <li>3- <u>Usuário</u> submete o formulário preenchido.  <b>Restrição:</b> Os campos nome, <u>noção</u> e <u>classificação</u> não podem ser deixados em branco.</li> <li>4- <u>Sistema</u> cadastra o novo <u>símbolo do léxico</u>.</li> <li>5- <u>Sistema</u> comunica ao <u>usuário</u> que a inclusão foi realizada com sucesso.</li> </ol> <p><b>Exceção:</b> Se o nome do <u>léxico</u> escolhido ou algum de seus <u>sinônimos</u> já estiver cadastrado no <u>sistema</u> como nome de outro <u>léxico</u> ou <u>sinônimo</u>, então o <u>sistema</u> deve exibir uma mensagem de erro, identificando o que o ocasionou.</p>	<p><b>Título:</b> ALTERAR SÍMBOLO DO LÉXICO.  <b>Objetivo:</b> Permitir que o usuário altere os dados de um <u>símbolo do léxico</u> cadastrado no <u>sistema</u>, através da edição de um formulário preenchido com suas informações cadastradas.  <b>Contexto:</b> <u>Símbolo do léxico</u> cadastrado no <u>sistema</u>.  <b>Recursos:</b> novos dados do <u>símbolo do léxico</u>.  <b>Atores:</b> <u>usuário, sistema</u>.  <b>Episódios:</b></p> <ol style="list-style-type: none"> <li>1- O <u>usuário</u> seleciona um <u>símbolo do léxico</u> do <u>projeto</u>, exibido no menu lateral.</li> <li>2- O <u>usuário</u> seleciona a opção “Alterar símbolo”</li> <li>3- <u>Sistema</u> exibe um formulário preenchido com dos <u>dados do léxico</u> selecionado.</li> <li>4- <u>Usuário</u> faz as alterações que deseja nas informações do <u>símbolo</u> e submete o formulário. <b>Restrição:</b> O campo <u>noção</u> não pode ser deixado em branco.</li> <li>5- O <u>sistema</u> comunica ao <u>usuário</u> que as alterações no <u>símbolo</u> foram feitas com sucesso.</li> </ol>
<p><b>Título:</b> REMOVER SÍMBOLO DO LÉXICO.  <b>Objetivo:</b> Permitir que o usuário remova um <u>símbolo do léxico</u> do <u>projeto</u> selecionado.  <b>Contexto:</b> <u>Símbolo do léxico</u> cadastrado no <u>sistema</u>.  <b>Recursos:</b> dados do <u>símbolo do léxico</u>.  <b>Atores:</b> <u>usuário, sistema</u>.  <b>Episódios:</b></p> <ol style="list-style-type: none"> <li>1- O <u>usuário</u> seleciona um <u>símbolo do léxico</u> do <u>projeto</u>, exibido no menu lateral.</li> <li>2- O <u>usuário</u> seleciona a opção “Remover símbolo”</li> <li>3- <u>Sistema</u> solicita a confirmação da exclusão.</li> <li>4- <u>Usuário</u> confirma a exclusão.</li> <li>5- <u>Sistema</u> comunica o <u>usuário</u> que a exclusão do <u>símbolo</u> foi realizada com sucesso.</li> </ol>	<p><b>Título:</b> EXIBIR SÍMBOLO DO LÉXICO E SEUS RELACIONAMENTOS.  <b>Objetivo:</b> Exibir as informações de um <u>símbolo do léxico</u> juntamente com seus relacionamentos, utilizando <u>elos</u> para permitir o acesso a outros <u>símbolos</u> referenciados pelo que esta sendo exibido e permitir o acesso a outros <u>elementos</u> do projeto que referenciem o <u>símbolo</u> exibido.  <b>Contexto:</b> <u>Símbolo do léxico</u> cadastrado no <u>sistema</u>.  <b>Recursos:</b> dados do <u>léxico</u>, nomes e <u>sinônimos</u> de <u>símbolos do léxico</u> e <u>títulos</u> de <u>cenários</u> do projeto.  <b>Atores:</b> <u>usuário, sistema</u>.  <b>Episódios:</b></p> <ol style="list-style-type: none"> <li>1- O <u>usuário</u> seleciona um <u>símbolo do léxico</u> do <u>projeto</u>, exibido no menu lateral.</li> <li>2- O <u>sistema</u> monta a rede de relacionamentos em torno do <u>símbolo</u> selecionado.</li> <li>3- <u>Sistema</u> exibe as informações do <u>símbolo</u> e os relacionamentos mapeados através de <u>elos</u>.</li> </ol>

Figura 42 – Grupo correspondente ao módulo léxico.

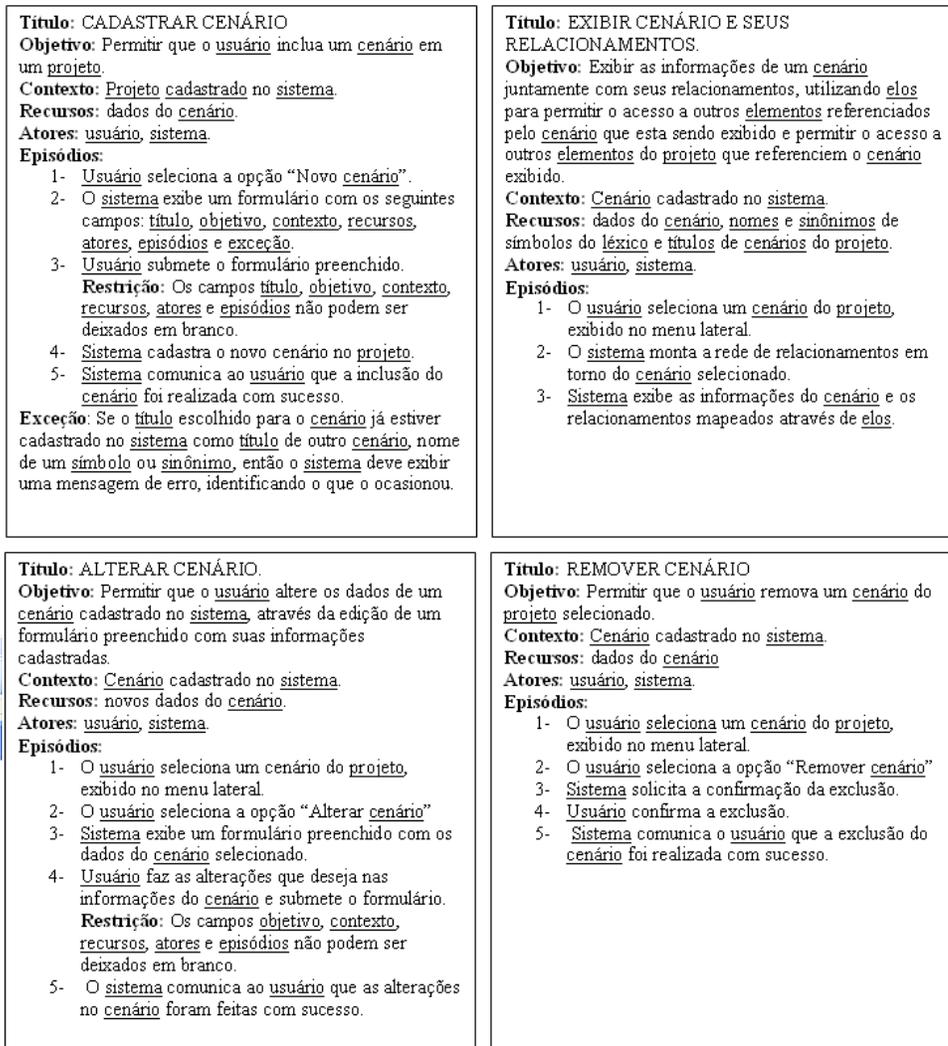


Figura 43 – Grupo correspondente ao módulo cenário.

### 5.2.3. Identificar os cenários raiz

Neste passo, primeiro iremos determinar o relacionamento entre os cenários pertencentes aos grupos identificados e, a partir dos relacionamentos identificados estabeleceremos uma ordem entre estes cenários. O cenário raiz será o cenário que não dependa de nenhum outro cenário do seu grupo.

No primeiro grupo identificamos os relacionamentos mostrados na Figura 44. Com base nestes relacionamentos podemos determinar que o cenário cadastrar usuário deve preceder todos os outros. Com isto, identificamos o cenário cadastrar usuário como cenário raiz deste grupo.

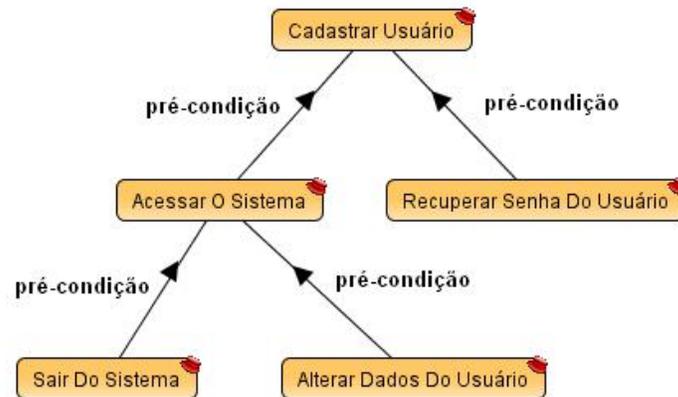


Figura 44 – Relacionamentos entre os cenários do grupo correspondente ao módulo usuário.

Os relacionamentos identificados no segundo grupo podem ser vistos na Figura 45. Podemos observar que o cenário raiz deste grupo é o cenários cadastrar projeto, pois não possui nenhuma pré-condição dentro do seu grupo.

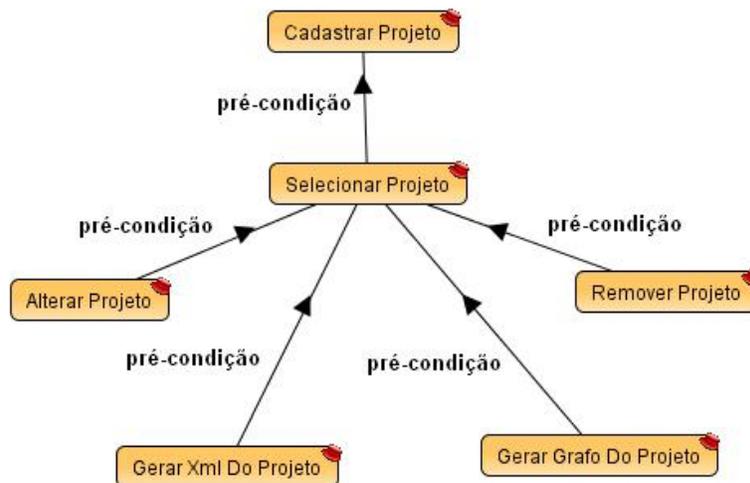


Figura 45 – Relacionamentos entre os cenários do grupo correspondente ao módulo projeto.

De acordo com os relacionamentos identificados entre os cenários do grupo correspondente ao módulo gerência de colaboradores (Figura 46), podemos identificar como cenário raiz o cenário incluir colaborador no projeto, pois este cenário não possui nenhuma pré-condição dentro de seu grupo.

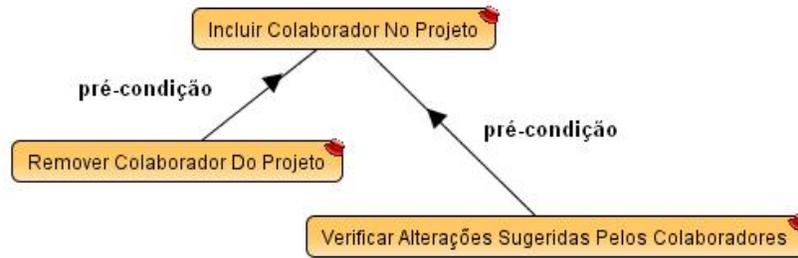


Figura 46 – Relacionamentos entre os cenários do grupo correspondente ao módulo colaboradores.

Na Figura 47 pode-se observar os relacionamentos entre os cenários do grupo referente ao módulo léxico. O cenário raiz deste grupo é o cenário cadastrar símbolo do léxico, pois não possui nenhuma pré-condição entre os cenários de seu grupo.

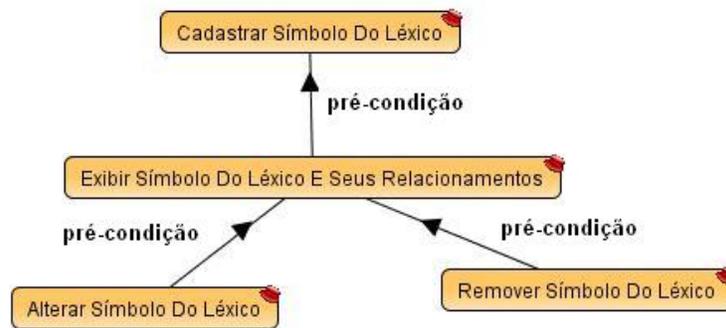


Figura 47 – Relacionamentos entre os cenários do grupo correspondente ao módulo léxico.

O cenário raiz do grupo correspondente ao módulo cenário é o cenário cadastrar cenário. Isto porque, como podemos observar na Figura 48, este cenário não possui nenhuma pré-condição entre os cenários do grupo.

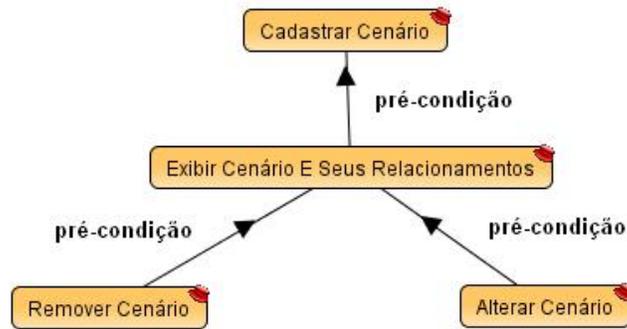


Figura 48 – Relacionamentos entre os cenários do grupo correspondente ao módulo cenário.

#### 5.2.4. Construir cenário integrador

O primeiro passo para a construção do cenário integrador é a identificação dos relacionamentos entre os cenários raiz e sua ordem. Podemos observar na Figura 49 os relacionamentos entre os cenários raiz identificados. Com base na ordem determinada através destes relacionamentos e o modelo apresentado na subseção 4.1.3.3, criamos o cenário integrador para o software C&L. Este cenário pode ser visto na Figura 50.

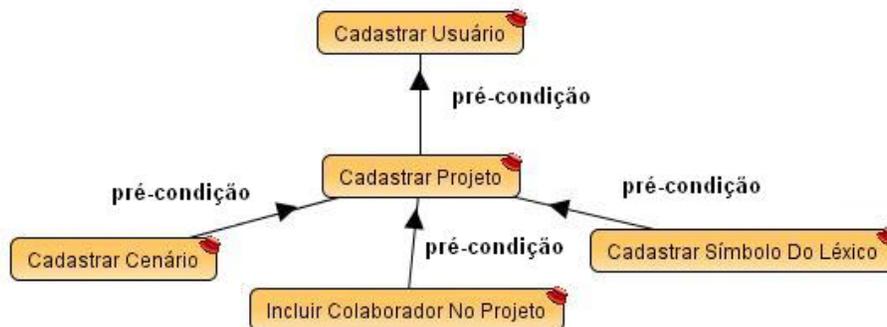


Figura 49 – Relacionamentos entre os cenários raiz.

<p><b>Título:</b> Sistema C&amp;L</p> <p><b>Objetivo:</b> Permitir a colaboração entre diversos <u>usuários</u> na elaboração de <u>projetos</u> utilizando <u>cenários</u> e <u>símbolos do léxico</u>, fornecendo meios para facilitar a integração com outras ferramentas.</p> <p><b>Contexto:</b> - <u>Usuário</u> deseja elicitar a linguagem e as situações de um domínio de aplicação específico.</p> <p><b>Recursos:</b> -</p> <p><b>Atores:</b> -</p> <p><b>Episódios:</b></p> <ol style="list-style-type: none"> <li>1- Se o <u>usuário</u> ainda não possuir cadastro então <u>CADASTRAR USUÁRIO</u>, senão <u>ACESSAR SISTEMA</u>.</li> <li>2- <u>Usuário</u> pode <u>CADASTRAR PROJETO</u> ou <u>SELECIONAR PROJETO</u> se já houver algum cadastrado.</li> <li>3- # [ <u>Usuário</u> pode <u>INCLUIR COLABORADOR NO PROJETO</u> ]</li> <li>4- [ <u>Usuário</u> pode <u>CADASTRAR SÍMBOLO DO LÉXICO</u> no <u>projeto</u>. ]</li> <li>5- [ <u>Usuário</u> pode <u>CADASTRAR CENÁRIO</u> no <u>projeto</u>. ]</li> <li>6- <u>Usuário</u> pode <u>SAIR DO SISTEMA</u>. #</li> </ol>
--

Figura 50 – Cenário integrador do software C&L.

### 5.2.5. Refinamento dos cenários

A seguir mostraremos os cenários divididos em camadas, seguindo o *framework* MVC, obtidos através do refinamento dos cenários construídos inicialmente. Como o número de cenários resultante deste refinamento é muito grande, mostraremos apenas alguns cenários de cada camada do software. Os cenários que mostraremos a seguir são resultantes do refinamento dos seguintes cenários, pertencentes ao módulo léxico: “cadastrar símbolo do léxico”, “exibir símbolo do léxico e seus relacionamentos”, “alterar símbolo do léxico”, “remover símbolo do léxico”.

#### 5.2.5.1. Cenários da camada de visão

**Título:** EXIBIR PÁGINA DE CADASTRO DE NOVO SÍMBOLO DO LÉXICO

**Objetivo:** Permitir que o usuário informe os dados do símbolo do léxico para seu cadastro no projeto. Para tanto, a página exibida conterá um formulário com os seguintes campos: nome, noção, classificação, impacto e sinônimo.

**Contexto:** Pode ser acessado através da opção “Novo símbolo” no menu superior da página principal.

**Atores:** usuário autenticado no sistema.

**Recursos:** script.js e estilo.css

**Episódios:**

- 1- Exibir campo Nome.
- 2- Exibir campo Noção.
- 3- Exibir *combobox* Classificação, com as opções: sujeito, verbo, estado e objeto.
- 4- Exibir campo Impacto.
- 5- Exibir campo Sinônimo(s)
- 6- Ao clicar no botão cadastrar o formulário será submetido e o sistema será redirecionado para o CONTROLE LÉXICO. **Restrição:** Antes de submeter o formulário VERIFICAR FORMULÁRIO DE CADASTRO DE LÉXICO.

**Título:** EXIBIR SÍMBOLO DO LÉXICO E SEUS RELACIONAMENTOS

**Objetivo:** Permitir a visualização dos dados de um símbolo do léxico. Para tanto, a página exibida conterá as seguintes informações: nome, noção, impacto, classificação e sinônimos. Os campos noção e impacto podem conter elos para outros símbolos do léxico.

**Contexto:** Pode ser acessado através da seleção de um símbolo do léxico do menu lateral.

**Atores:** usuário autenticado no sistema.

**Recursos:** script.js e estilo.css

**Episódios:**

- 1- Exibir nome do símbolo.
- 2- Exibir noção do símbolo, já com os elos montados.
- 3- Exibir classificação do símbolo.
- 4- Exibir impacto do símbolo.
- 5- Exibir sinônimos do símbolo.
- 6- Exibir botão “Alterar símbolo”.
- 7- Exibir botão “Remover símbolo”.
- 8- Ao clicar no botão “Alterar símbolo”, o formulário será submetido com o comando alterar e o sistema será redirecionado para o CONTROLE USUÁRIO.
- 9- Ao clicar no botão “Remover símbolo”, o formulário será submetido com o comando remover e o sistema será redirecionado para o CONTROLE USUÁRIO.

**Título:** EXIBIR PÁGINA DE ALTERAÇÃO DOS DADOS DO SÍMBOLO DO LÉXICO.

**Objetivo:** Permitir que o usuário altere os dados do símbolo do léxico. Para isto, será exibido um formulário com os dados do símbolo do léxico já preenchido, permitindo a edição dos campos noção, impacto, classificação e sinônimos.

**Contexto:** Pode ser acessado através do atalho “Alterar símbolo” da página de exibição dos dados do símbolo do léxico.

**Atores:** usuário autenticado no sistema.

**Recursos:** script.js e estilo.css

**Episódios:**

- 1- Exibir campo nome já preenchido.
- 2- Exibir campo noção já preenchido, permitindo sua edição.
- 3- Exibir campo classificação já preenchido, permitindo sua edição.
- 4- Exibir campo impacto já preenchido, permitindo sua edição.
- 5- Exibir campo sinônimo já preenchido, permitindo sua edição.
- 6- Ao clicar no botão “Alterar”, o formulário será submetido com o comando alterar e o sistema será redirecionado para o CONTROLE USUÁRIO. **Restrição:** Antes de submeter o formulário, VERIFICAR FORMULÁRIO DE ALTERAÇÃO DE SÍMBOLO DO LÉXICO.

### 5.2.5.2.

#### Cenário da camada de controle

**Título:** CONTROLE LÉXICO

**Objetivo:** Intermediar a comunicação entre a camada de visão e a camada de modelo do módulo léxico.

**Contexto:** novo\_léxico.html, alterar\_dados\_léxico.lp, erro.lp, página\_principal.lp, sucesso.lp.

**Atores:** camada de modelo e camada de visão.

**Recursos:** dados do léxico, informações para as páginas de erro, informações para as páginas de sucesso.

**Episódios:**

- 1- Se a requisição recebida for para cadastrar símbolo do léxico então CHECAR NOME DO LÉXICO. Se nome disponível então CADASTRAR SÍMBOLO DO LÉXICO.
- 2- Se o símbolo for cadastrado corretamente então EXIBIR PÁGINA DE SUCESSO, senão EXIBIR PÁGINA DE ERRO.

- 3- Se a requisição recebida for para remover símbolo então REMOVER SÍMBOLO DO LÉXICO.
- 4- Se o símbolo for removido com sucesso então EXIBIR PÁGINA DE SUCESSO, senão EXIBIR PÁGINA DE ERRO.
- 5- Se a requisição recebida for para alterar símbolo do léxico então ALTERAR SÍMBOLO DO LÉXICO.
- 6- Se a requisição recebida for para exibir dados do símbolo e seus relacionamentos, então RECUPERAR DADOS DO SÍMBOLO E MONTAR SEUS RELACIONAMENTOS. Após a recuperação dos dados, EXIBIR SÍMBOLO DO LÉXICO E SEUS RELACIONAMENTOS.

### 5.2.5.3.

#### Cenários da camada modelo

**Título:** CADASTRAR SÍMBOLO DO LÉXICO

**Objetivo:** Cadastrar símbolo do léxico no projeto do usuário, através da inserção de seus dados no banco de dados.

**Contexto:** É necessário o preenchimento do formulário presente em EXIBIR PÁGINA DE CADASTRO DE NOVO SÍMBOLO DO LÉXICO.

**Atores:** CONTROLE LÉXICO.

**Recursos:** dados do símbolo do léxico, nome do projeto (nome, noção, classificação, impacto e sinônimos), banco de dados.

**Episódios:**

- 1- Receber os dados do símbolo do léxico e o nome do projeto repassados pela CAMADA LÉXICO.
- 2- Conectar ao banco de dados. **Restrição:** Banco de dados deve estar disponível.
- 3- Inserir dados do símbolo do léxico no banco de dados.
- 4- Inserir sinônimos do símbolo do léxico no banco de dados.
- 5- Desconectar do banco de dados.
- 6- Informar a camada CONTROLE LÉXICO que os dados foram inseridos.

**Exceção:**

O nome do símbolo do léxico informado já existe no banco de dados. Informamos a camada CONTROLE LÉXICO que não foi possível inserir os dados porque o nome já existe.

**Título:** REMOVER SÍMBOLO DO LÉXICO

**Objetivo:** Remover um símbolo do léxico do projeto do usuário, através da remoção de seus dados do banco de dados.

**Contexto:** Dados do símbolo devem estar cadastrados no sistema.

**Atores:** CONTROLE LÉXICO.

**Recursos:** nome do símbolo, nome do projeto em que está cadastrado, banco de dados.

**Episódios:**

- 1- Receber o nome do símbolo e do projeto da camada CONTROLE LÉXICO.
- 2- Conectar ao banco de dados. **Restrição:** Banco de dados deve estar disponível.
- 3- Remover símbolo do léxico do banco de dados.
- 4- Desconectar do banco de dados.
- 5- Informar a camada CONTROLE LÉXICO que o símbolo foi removido.

**Título:** ALTERAR SÍMBOLO DO LÉXICO

**Objetivo:** Alterar os dados de um símbolo do léxico cadastrado no projeto, através de uma atualização de seus dados cadastrados no banco de dados.

**Contexto:** Símbolo do léxico cadastrado no sistema.

**Atores:** CONTROLE LÉXICO.

**Recursos:** dados alterados do símbolo do léxico, nome do projeto, banco de dados.

**Episódios:**

- 1- Receber os dados do símbolo do léxico alterados e o projeto a que pertence.
- 2- Conectar ao banco de dados. **Restrição:** Banco de dados deve estar disponível.
- 3- Alterar os dados do símbolo no do banco de dados.
- 4- Desconectar do banco de dados.
- 5- Informar a camada CONTROLE LÉXICO que o símbolo foi alterado.

**Título:** RECUPERAR DADOS DO SÍMBOLO E MONTAR SEUS RELACIONAMENTOS.

**Objetivo:** Recuperar os dados do símbolo do léxico do banco de dados e criar elos nos textos recuperados dos campos noção e impacto, através da identificação da ocorrência de nomes ou sinônimos de outros símbolos do léxico nestes textos.

**Contexto:****Atores:** CONTROLE LÉXICO.**Recursos:** nome do símbolo, nome do projeto, nome dos símbolos do projeto, sinônimos dos símbolos do léxico.**Episódios:**

- 1- Receber o nome do símbolo e do projeto da CAMADA CONTROLE.
- 2- Recuperar o nome dos símbolos do léxico do projeto.
- 3- Recuperar os sinônimos dos símbolos do léxico do projeto.
- 4- COLOCAR ELOS NO TEXTO impacto.
- 5- COLOCAR ELOS NO TEXTO noção.

Encaminhar os dados do símbolo do léxico, já com os elos montados, a camada CONTROLE LÉXICO.

**5.2.6.****Operacionalização dos cenários**

Se fossemos mostrar todos os cenários operacionalizados nesta subseção teríamos que mostrar o código fonte de todo o software, o que a tornaria muito extensa, por isso decidimos mostrar um exemplo através da operacionalização do cenário “exibir símbolo do léxico e seus relacionamentos”, que é um dos principais do software. Este cenário abrange a principal característica do C&L, que é uma estratégia de rastreamento automático [Sayão06]. Esta estratégia se baseia na identificação automática dos relacionamentos entre símbolos do léxico e cenários e a montagem de elos, que são fundamentais na implementação da estratégia, pois permitem a navegação entre os símbolos e cenários relacionados.

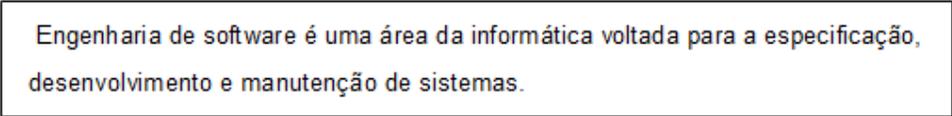
Antes de mostrarmos a operacionalização do cenário iremos apresentar o problema que é resolvido por ele. Depois mostraremos o algoritmo que era utilizado antes da re-engenharia do C&L e a falha que apresentava. Por fim, apresentaremos o novo algoritmo, que foi implementado no novo C&L.

**5.2.6.1.****Apresentação do problema**

Um projeto do C&L pode possuir vários elementos cadastrados, que podem ser símbolos do léxico ou cenários. Estes elementos possuem seus identificadores únicos dentro do projeto. No caso de símbolos do léxico, este identificador é o seu nome e no caso de cenários, este identificador é seu título.

O desejado é que o C&L identifique automaticamente a ocorrência destes identificadores no texto dos elementos que estão sendo exibidos e que, caso encontre algum identificador, o substitua por um elo para o elemento correspondente. Estes elos representam os relacionamentos que podem ocorrer entre dois cenários, entre dois símbolos do léxico ou entre um cenário e um símbolo do léxico. Os identificadores dos elementos podem ser compostos de várias palavras. Neste caso o C&L deve dar preferência aos identificadores compostos de mais palavras.

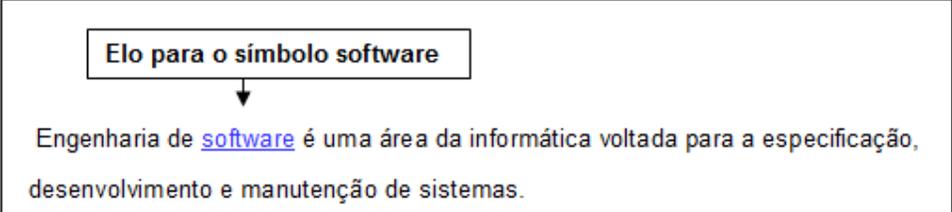
Para exemplificar, suponhamos que exista um projeto com apenas um símbolo do léxico cadastrado, cujo nome seja “software”, e que estejamos visualizando o texto da Figura 51.



Engenharia de software é uma área da informática voltada para a especificação, desenvolvimento e manutenção de sistemas.

Figura 51 – Exemplo de texto sendo visualizado.

Neste caso, o termo “software” deveria ser substituído por um elo para o símbolo software (Figura 52).



Elo para o símbolo software

Engenharia de [software](#) é uma área da informática voltada para a especificação, desenvolvimento e manutenção de sistemas.

Figura 52 – Exemplo de texto com o elo software.

Suponhamos que mais adiante, o usuário cadastre um novo símbolo de nome “engenharia de software” no projeto, e que novamente estejamos visualizando o texto da Figura 51. Agora, o C&L deve substituir o termo “engenharia de software” por um elo para o símbolo “engenharia de software” e não teríamos mais uma referência ao termo “software” (Figura 53).

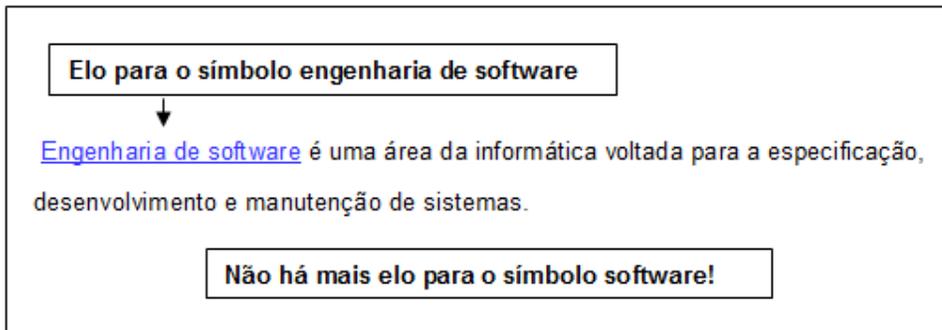


Figura 53 – Exemplo de texto com o elo engenharia de software.

### 5.2.6.2. Algoritmo antigo

Para facilitar, mostraremos, através de um exemplo, o funcionamento do algoritmo utilizado pelo C&L antes da re-engenharia e o erro que ocorria. Neste exemplo, iremos considerar um projeto com os seguintes símbolos do léxico cadastrados: “engenheiro”, “engenheiro de software”, “engenheiro de software experiente”, “software” e “engenharia de software”. Vamos considerar também que estamos visualizando o texto apresentado na Figura 54.

O engenheiro de software experiente é um engenheiro de software com, no mínimo, cinco anos de experiência na prática da engenharia de software

Figura 54 – Exemplo de texto sendo visualizado.

O primeiro passo do algoritmo é criar um vetor, colocar os identificadores dos elementos cadastrados no projeto neste vetor e ordená-los pelo número de palavras (Figura 55).

**Vetor** = [engenheiro de software experiente, engenheiro de software, engenharia de software, engenheiro, software]

Figura 55 – Vetor de identificadores ordenado.

Depois da ordenação, o algoritmo percorre o vetor, começando do termo composto de mais palavras, e cada termo do vetor encontrado no texto é substituído por um elo para o elemento correspondente (Figura 56).



Figura 56 – Passos da aplicação do algoritmo antigo.

Podemos ver que no segundo e quarto passo, a *tag* HTML `<a>` (aqui simplificada sem seus atributos, apenas a título de exemplo), inserida para criar

o elo, fez com que o algoritmo funcionasse corretamente. Entretanto, no quinto passo a palavra “software” aparece no meio do termo “engenheiro de software experiente”, conseqüentemente sem a *tag* <a> para diferenciá-la. Por isso é criado um elo dentro de outro elo, o que ocasiona um erro. Este erro demorou muito para ser percebido, pois é uma situação que ocorre raramente no uso normal do software.

### **5.2.6.3. Algoritmo novo**

Assim como feito anteriormente, também usaremos um exemplo para explicar o funcionamento do novo algoritmo. Para evidenciar as diferenças entre os dois algoritmos, utilizaremos o mesmo exemplo do algoritmo antigo.

O primeiro passo do novo algoritmo é, assim como o antigo, montar um vetor com todos os identificadores dos elementos cadastrados no projeto e ordená-los de acordo com seu número de palavras. Também devemos criar um vetor auxiliar inicialmente vazio. Este vetor será utilizado no decorrer do algoritmo.

Após a criação do vetor auxiliar, iremos percorrer o vetor ordenado a partir do identificador com maior número de palavras até os com menor número de palavras, procurando a ocorrência deles no texto. Caso seja encontrada uma ocorrência, o identificador do símbolo do léxico será colocado no vetor auxiliar e sua ocorrência no texto será substituída pelo código xzzxk\*kxy, onde o \* deverá ser substituído pela posição que o identificador ocupará no vetor auxiliar. Caso o identificador seja o título de um cenário, o procedimento será o mesmo só mudará o código utilizado, que deverá ser wczxk\*kxyyc, onde o \* deve ser substituído pela posição que o identificador ocupará no vetor auxiliar. Os passos da aplicação deste algoritmo podem ser vistos na Figura 57.

**Primeiro passo:**

**Vetor** = [engenheiro de software experiente, engenheiro de software, engenharia de software, engenheiro, software]

**Vetor auxiliar** = [engenheiro de software experiente]

O *xzzxk1kxy* é um engenheiro de software com, no mínimo, cinco anos de experiência na prática da engenharia de software.

**Segundo passo:**

**Vetor** = [engenheiro de software experiente, engenheiro de software, engenharia de software, engenheiro, software]

**Vetor auxiliar** = [engenheiro de software experiente, engenheiro de software]

O *xzzxk1kxy* é um *xzzxk2kxy* com, no mínimo, cinco anos de experiência na prática da engenharia de software.

**Terceiro passo:**

**Vetor** = [engenheiro de software experiente, engenheiro de software, engenharia de software, engenheiro, software]

**Vetor auxiliar** = [engenheiro de software experiente, engenheiro de software, engenharia de software]

O *xzzxk1kxy* é um *xzzxk2kxy* com, no mínimo, cinco anos de experiência na prática da *xzzxk3kxy*.

**Quarto passo:**

**Vetor** = [engenheiro de software experiente, engenheiro de software, engenharia de software, engenheiro, software]

**Vetor auxiliar** = [engenheiro de software experiente, engenheiro de software, engenharia de software]

O *xzzxk1kxy* é um *xzzxk2kxy* com, no mínimo, cinco anos de experiência na prática da *xzzxk3kxy*.

**Quinto passo:**

**Vetor** = [engenheiro de software experiente, engenheiro de software, engenharia de software, engenheiro, software]

**Vetor auxiliar** = [engenheiro de software experiente, engenheiro de software, engenharia de software]

O *xzzxk1kxy* é um *xzzxk2kxy* com, no mínimo, cinco anos de experiência na prática da *xzzxk3kxy*.

Figura 57 – Passos da aplicação do algoritmo novo.

Após a execução destes cinco passos teremos um texto com códigos e o vetor auxiliar com os identificadores correspondentes a estes códigos. Para finalizar basta substituir os códigos por eles para os elementos correspondentes, como mostra a Figura 58.

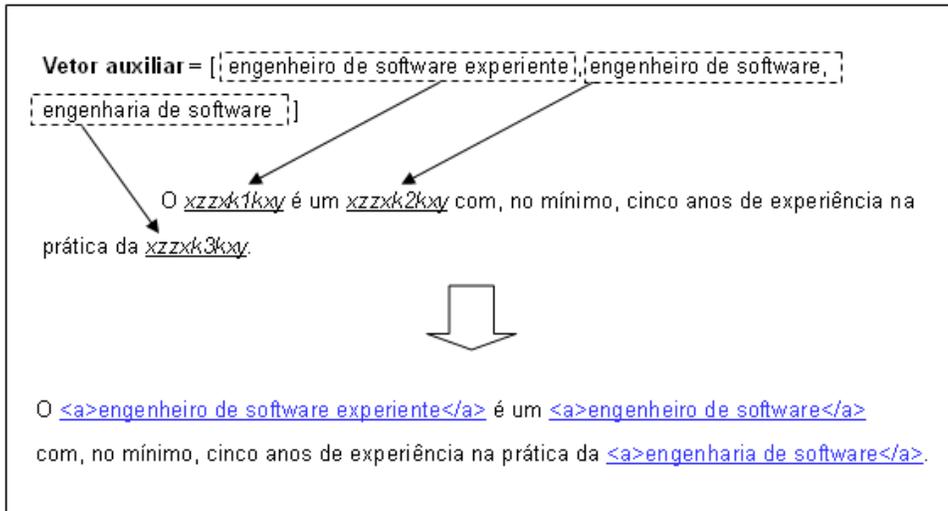


Figura 58 – Montagem dos elos do algoritmo novo.

A grande diferença do algoritmo novo para o antigo é a substituição das ocorrências dos identificadores no texto por códigos. Desta forma, ao realizarmos uma nova busca no texto, já não consideramos mais os identificadores que já foram encontrados, como acontecia anteriormente.

#### 5.2.6.4. Implementação do algoritmo

A seguir mostraremos a operacionalização do cenário “Colocar atalhos no texto”, responsável pela descrição do algoritmo apresentado na subseção anterior. Este cenário faz uso do relacionamento de subcenários para destacar algumas funcionalidades e recuperar dados do banco de dados. Os subcenários “selecionar todos os léxicos do banco de dados”, “selecionar sinônimos do banco de dados” e “selecionar cenários do banco de dados” recuperam, respectivamente, todos os símbolos do léxico, sinônimos de símbolos e cenários de um determinado projeto. O subcenário “organizar tabela” organiza uma tabela de acordo com o número de palavras de seus elementos. E o subcenário “contar palavras” para determina o número de palavras um elemento específico. O relacionamento entre o cenário e seus subcenários pode ser visto na Figura 59.

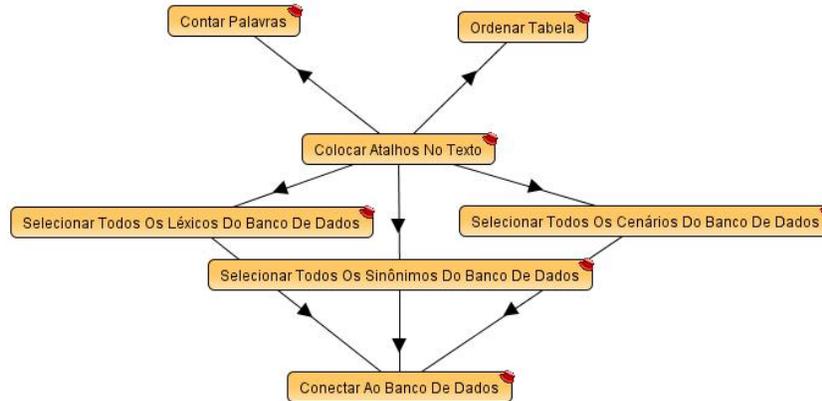


Figura 59 – Relacionamentos entre os cenários do algoritmo.

```

--[[
@Titulo: Colocar elos no texto
@Objetivo: Substituir a ocorrência de nomes de termos do léxico e seus sinônimos
ou de títulos de cenários, já cadastrado no projeto, por um atalho que leve para
definição do termo ou para descrição do cenário. Para tanto, cada elemento
cadastrado no projeto (termos do léxico e cenários) é procurado no texto, caso
alguma ocorrência seja encontrada é criado um atalho no texto para definição deste
elemento.
@Contexto: O usuário deve possuir um projeto com pelo menos um elemento
cadastrado.
@Atores: selecionar_todos_lexicos_bd, selecionar_sinonimos_projeto_bd,
selecionar_todos_cenarios_bd, ordenar_tabela, contar_palavras.
@Recursos: Identificador do projeto (parâmetro id_projeto), texto em que os
elementos serão procurados (parâmetro texto), tipo de elemento que foi selecionado
pelo usuário: termo do léxico ou cenário (parâmetro tipo) e banco de dados.
]]-
function colocar_elos (id_projeto, texto, tipo)

--@Episódio 1: SELECIONAR TODOS OS LEXICOS DO BANCO DE DADOS
  local lexicos = selecionar_todos_lexicos_bd(id_projeto);

--@Episódio 2: SELECIONAR TODOS OS SINÔNIMOS DO BANCO DE DADOS
  local sinonimos = selecionar_sinonimos_projeto_bd(id_projeto);

--@Episódio 3: SELECIONAR TODOS OS CENÁRIOS DO BANCO DE DADOS
  local cenarios = selecionar_todos_cenarios_bd(id_projeto);

--@Episódio 4: Criar uma tabela única, contendo nome dos termos do léxico e seus
sinônimos.
  for index, sinonimo in pairs(sinonimos) do
    table.insert(lexicos, sinonimo);
  end
  local lexicos_e_sinonimos = lexicos;

--@Episódio 5: ORDENAR TABELA sinonimos.
  sinonimos = ordenar_tabela(sinonimos, 1, table.maxn(sinonimos) ,"lexico" )

--@Episódio 6: ORDENAR TABELA cenários.
  cenarios = ordenar_tabela(cenarios, 1, table.maxn(cenarios) ,"cenario" )

--@Episódio 7: ORDENAR TABELA lexicos_e_sinonimos.
  lexicos_e_sinonimos = ordenar_tabela(lexicos_e_sinonimos, 1,
table.maxn(lexicos_e_sinonimos),"lexico" )

--@Episódio 8: Obter o tamanho da tabela lexicos_e_sinonimos, da tabela cenários e
o total, que corresponde a soma dos tamanhos das duas tabelas.
  local tam_lexicos_e_sinonimos = #lexicos_e_sinonimos;
  local tam_cenarios = #cenarios;
  local tam_total = tam_lexicos_e_sinonimos + tam_cenarios ;

--@Episódio 9: Criar tabelas que irão armazenar os links criados para termos do
léxico e para cenários.

```

```

local tabela_links_lexico = {};
local tabela_links_cenario = {};

if ((tipo == "lexico") or (tam_cenarios == 0)) then
--@Episódio 10: Se não há cenários cadastrados no projeto então apenas a lista de
léxicos e sinônimos será percorrida.
    for index, lexico in pairs(lexicos_e_sinonimos) do

--@Episódio 11: A variável nome_lexico recebe o nome do termo do léxico ou
sinônimo que está sendo verificado no momento.
        local nome_lexico = lexico["NOME"];

--@Episódio 12: Uma expressão regular é montada com o nome do termo do léxico ou
sinônimo. Esta expressão regular evita erros devido a pontuação e espaços em
branco na hora da busca.
        local expressao_regular =
            "[%p%s]"..nome_lexico..("[%p%s]";

        if (string.find(texto, expressao_regular) ~= nil) then

            local nome_lexico_link = "";

--@Episódio 13: Se alguma ocorrência da expressão regular for encontrada no texto,
então esta expressão é substituída pelo código (wzzxk*kxy), onde o * será
substituído por um número que identificará a posição do termo do léxico ou
sinônimo na tabela léxicos e sinônimos.
            texto = string.gsub(texto, expressao_regular,
                "%1"..wzzxk"..index.."kxy".."%2");

--@Episódio 14: Verifica se a expressão encontrada é realmente um termo do léxico
ou é um sinônimo de um termo do léxico.
            if (lexico["LEXICO"] ~= nil) then

--@Episódio 15: Se a expressão encontrada for um sinônimo então o nome_lexico_link
deve armazenar o nome do termo do léxico a que pertence o sinônimo. Isto é feito
para que possamos criar um atalho para o termo do léxico a que o sinônimo
corresponde.
                nome_lexico_link = lexico["LEXICO"]
            else

--@Episódio 16: Se a expressão encontrada for um termo do léxico então o
nome_lexico_link deve armazenar o nome do termo do léxico.
                nome_lexico_link = nome_lexico;
            end

--@Episódio 17: Criar o atalho para a expressão encontrada no texto.
            link = "<a title=\"Léxico\"
href=\"../visao/exibe_lexico.lp?id_projeto"..id_projeto.."&nome"..nome_lexico_link.."\">\"..nome_lexico
..\"</a>";

--@Episódio 18: Inserir o atalho criado na tabela_links_lexico na mesma posição
que o elemento encontrado ocupa na tabela lexicos_e_sinonimos. Isto permitirá que
identifiquemos a que código (wzzxk*kxy) do texto este atalho corresponde.
            table.insert(tabela_links_lexico, index, link);
        end --if

    end --for

else

    if (tam_lexicos_e_sinonimos == 0 and tam_cenarios > 0) then

--@Episódio 19: Se não há léxicos no projeto então apenas a lista de cenários será
percorrida.
        for index, cenario in pairs(cenarios) do

--@Episódio 20: A variável nome_cenario recebe o título do cenário que está sendo
verificado no momento.
            nome_cenario = cenario["TITULO"];

--@Episódio 21: Uma expressão regular é montada com o título do cenário. Esta
expressão regular evita erros devido à pontuação e espaços em branco na hora da
busca.
            expressao_regular =
                "[%p%s]"..nome_cenario..("[%p%s]";

            if (string.find(texto, expressao_regular) ~= nil)
then

```

```

--@Episódio 22: Se alguma ocorrência de expressão regular for encontrada no texto,
então um atalho é montado com o nome do cenário.
link = "<a title=\"Cenário\"
href=\"../visao/exibe_cenario.lp?id_projeto=
..id_projeto..\"&titulo=\"..titulo_cenario..\"&
omando=exibir\">\"..titulo_cenario..\"</a>";

--@Episódio 23: O atalho montado montado é inserido na tabela_links_cenario na
mesma posição que o título do cenário encontrado ocupa na tabela cenarios.
table.insert(tabela_links_cenario, index,
link);

--@Episódio 24: A expressão é substituída pelo código (wzczxk*kxyyc), onde o *
será substituído por um número que identificará a posição do título do cenário na
tabela cenários.
texto = string.gsub(texto, expressao_regular,
"%1..\"wzczxk\"..i..\"kxyyc\"..\"%2");
end --if

end -- for

elseif (tam_total > 0) then

--@Episódio 25: Se há léxicos e cenários no projeto então tanto a tabela cenarios
quanto a lexicos_e_sinonimos serão percorridas.
i = 1;
j = 1;
contador = 1;
while (contador <= tam_total) do

--@Episódio 26: Verifica se a tabela cenarios e a tabela lexicos_e_sinonimos já
foram percorridas.
if ( ( i <= tam_lexicos_e_sinonimos ) and ( j <=
tam_cenarios) ) then

--@Episódio 27: Se a tabela cenarios ainda não foi completamente percorrida e a
tabela lexicos_e_sinonimos também não foi completamente percorrida então CONTAR
PALAVRAS do título do cenário e CONTAR PALAVRAS do nome do léxico ou sinônimo.
if ( contar_palavras(cenarios[j][\"TITULO\"])
<=
contar_palavras(lexicos_e_sinonimos[i][\"NOME\"
]) ) then

--@Episódio 28: Se o título do cenário atual possui um número menor ou igual de
palavras que o nome do termo do léxico, então procuraremos a ocorrência deste
léxico no texto.
nome_lexico =
lexicos_e_sinonimos[i][\"NOME\"];

--@Episódio 29: Uma expressão regular é criada com o nome do termo do léxico
atual. Esta expressão regular impede que erros sejam causados por causa da
pontuação e espaços em branco.
expressao_regular =
\"([%p%s])..nome_lexico..\"([%p%s])\";

if( string.gfind(texto,
expressao_regular) ~= nil ) then

--@Episódio 30: Se uma ocorrência da expressão regular é encontrada no texto então
a expressão é substituída pelo código (wzzxk*kxy), onde o * será substituído por
um número que identificará a posição do nome do termo do léxico ou sinônimo na
tabela lexicos_e_sinonimos.
texto = string.gsub(texto,
expressao_regular,
"%1..\"wzzxk\"..i..\"kxy\"..\"%2\"
);

--@Episódio 31: É feita um verificação para saber se o termo encontrado é um termo
do léxico ou um sinônimo de um termo do léxico.
if
(lexicos_e_sinonimos[i][\"LEXIC
O\"] ~= nil) then

--@Episódio 32: Se o elemento encontrado for um sinônimo então o termo do léxico a
que este sinônimo corresponde é armazenado na variável nome_lexico_link. Caso

```

contrário, se o elemento encontrado for um termo do léxico, então ele é armazenado na variável nome\_léxico.

```

nome_lexico_link =
lexicos_e_sinonimos[i]
["LEXICO"];
else
nome_lexico_link =
nome_lexico;
end

--@Episódio 33: Um atalho para o termo encontrado é criado e colocado na tabela
links_lexico, na mesma posição que o léxico atual ocupa na tabela lexicos e
sinônimos. Isto permitirá que identifiquemos a que código (wzzxk*kxy) do texto
este atalho corresponde.
link = "<a title=\"Léxico\"
href=\"../visao/exibe_lexico.l
p?id_projeto=\"..id_projeto..\"&
nome=\"..nome_lexico_link..\"\">
\"..nome_lexico..\"</a>";
table.insert(tabela_links_lexi
co, i, link);
end --if
i = i + 1;

else --if
--@Episódio 34: Se o título do cenário atual possui um maior de palavras que o
nome do léxico atual, então procuraremos a ocorrência do título deste cenário no
texto.
titulo_cenario =
cenarios[j]["TITULO"];

--@Episódio 35: Uma expressão regular é criada com o título do cenário atual. Esta
expressão regular impede que erros sejam causados por causa da pontuação e espaços
em branco.
expressao_regular =
"([%p%s])"..titulo_cenario.."([%p%s])
";

if( string.gfind(texto,
expressao_regular) ~= nil ) then

--@Episódio 36: Caso encontremos uma ocorrência do cenário atual no texto, um
atalho será criado e armazenado no tabela tabela_links_cenario, na mesma posição
que o cenário atual ocupa na tabela cenários.
link = "<a title=\"Cenário\"
href=\"../visao/exibe_cenario.
lp?id_projeto=\"..id_projeto..\"
&titulo=\"..titulo_cenario..\"\">
\"..titulo_cenario..\"</a>";
table.insert(tabela_links_cena
rio, j, link);

--@Episódio 37: A expressão é substituída pelo código (wzczxk*kxyyc), onde o *
será substituído por um número que identificará a posição do título do cenário na
tabela cenários.
texto = string.gsub(texto,
expressao_regular,
"%1"..wzczxk"..j..kxyyc"..%
2");

end --if
j = j + 1;

end --if

elseif( tam_lexicos_e_sinonimos == i-1 ) then

--@Episódio 38: Se a tabela de termos do léxico chegou ao fim e a tabela de
cenários ainda possui cenários que ainda não foram procurados, então devemos
continuar percorrendo apenas a tabela cenários.
titulo_cenario = cenarios[j]["TITULO"];

--@Episódio 39: Uma expressão regular é criada com o título do cenário atual. Esta
expressão regular impede que erros sejam causados por causa da pontuação e espaços
em branco.
```

```

expressao_regular =
"([%p%s])"..titulo_cenario.."([%p%s])";

if( string.gfind(texto, expressao_regular) ~=
nil ) then

```

--@Episódio 40: Caso seja encontrada uma ocorrência do cenário atual no texto, um atalho será criado e armazenado na tabela `tabela_links_cenario`, na mesma posição que o cenário atual ocupa na tabela `cenários`.

```

link = "<a title=\"Cenário\"
href=\"../visao/exibe_cenario.lp?id_p
rojeto=\"..id_projeto..&titulo=\"..tit
ulo_cenario..\">\"..titulo_cenario..\"
</a>";
table.insert(tabela_links_cenario, j,
link);

```

--@Episódio 41: A expressão é substituída pelo código (`wzczxk*kxyyc`), onde o \* será substituído por um número que identificará a posição do título do cenário na tabela `cenarios`.

```

texto = string.gsub(texto,
expressao_regular,
"%1"..wzczxk"..j"..kxyyc"..%2");

```

```

end --if

```

```

elseif( tam_cenarios == j-1 ) then

```

--@Episódio 42: Se a tabela de cenários chegou ao fim e a tabela `lexicos_e_sinonimos` ainda possui elementos que ainda não foram procurados, então devemos continuar percorrendo apenas a tabela `lexicos_e_sinonimos`.

```

nome_lexico = lexicos_e_sinonimos[i]["NOME"];

```

--@Episódio 43: Uma expressão regular é criada com o termo do léxico atual. Esta expressão regular impede que erros sejam causados por causa da pontuação e espaços em branco.

```

expressao_regular =
"([%p%s])"..nome_lexico.."([%p%s])";

```

```

if( string.gfind(texto, expressao_regular) ~=
nil ) then

```

--@Episódio 44: Caso seja encontrada uma ocorrência do termo do léxico atual no texto, A expressão é substituída pelo código (`wzzxk*kxy`), onde o \* será substituído por um número que identificará a posição do nome do termo do léxico na tabela `léxicos`.

```

texto = string.gsub(texto,
expressao_regular,
"%1"..wzzxk"..i"..kxy"..%2");

```

```

if (lexicos_e_sinonimos[i]["LEXICO"]
~= nil) then

```

--@Episódio 45: Se o elemento encontrado for um sinônimo então o termo do léxico a que este sinônimo corresponde é armazenado na variável `nome_lexico_link`. Caso contrário, se o elemento encontrado for um termo do léxico, então ele é armazenado na variável `nome_léxico`.

```

nome_lexico_link =
lexicos_e_sinonimos[i]["LEXICO"];
else

```

```

nome_lexico_link =

```

```

nome_lexico;

```

```

end

```

--@Episódio 46: Um atalho para o termo do léxico encontrado é criado e armazenado na tabela `tabela_links_lexico`, na mesma posição que o léxico atual.

```

link = "<a title=\"Léxico\"
href=\"../visao/exibe_lexico.lp?id_pr
ojeto=\"..id_projeto..&nome=\"..nome_l
exico_link..\">\"..nome_lexico..\"</a>\";

```

```

table.insert(tabela_links_lexico, i,

```

```

link);

```

```
        end --if

        i = i + 1;

        end --if
        contador = contador + 1;

    end --while

    end --if
end--if

contador = 1;

--@Episódio 47: A tabela links lexico é percorrida e os códigos inseridos
anteriormente nos texto são substituídos pelos links armazenados na tabela. O
número que se encontra no meio do código corresponde a posição na tabela de links
que o atalho que será inserido se encontra.
    for i, link in pairs(tabela_links_lexico) do
        expressao_regular = ("([%p%s])"..wzzxk"..i.."kxy"..("[%p%s])");
        texto = string.gsub(texto, expressao_regular, "%1"..link.."%2");
    end
--@Episódio 48: A tabela links cenários é percorrida e os códigos inseridos
anteriormente nos texto são substituídos pelos links armazenados na tabela. O
número que está no meio do código corresponde a posição na tabela de links que o
atalho que será inserido se encontra.
    for i, link in pairs(tabela_links_cenario) do
        expressao_regular = ("([%p%s])"..wzczxk"..i.."kxyyc"..("[%p%s])");
        texto = string.gsub(texto, expressao_regular, "%1"..link.."%2");
    end
--@Episódio 49: O texto com os links é retornado para o usuário.
    cgilua.put(texto);

end
```