

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Enola: Geração procedural de jogos sérios para a prática de Banco de Dados

Jonas Gabriel Maia Bruno Nepomuceno

JUIZ DE FORA
SETEMBRO, 2021

Enola: Geração procedural de jogos sérios para a prática de Banco de Dados

JONAS GABRIEL MAIA BRUNO NEPOMUCENO

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Sistemas de Informação

Orientador: Jairo Francisco de Souza

JUIZ DE FORA
SETEMBRO, 2021

ENOLA: GERAÇÃO PROCEDURAL DE JOGOS SÉRIOS PARA A PRÁTICA DE BANCO DE DADOS

Jonas Gabriel Maia Bruno Nepomuceno

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM SISTEMAS DE INFORMAÇÃO.

Aprovada por:

Jairo Francisco de Souza
Doutor em Informática

Victor Ströele de Andrade Menezes
Doutor em Engenharia de Sistemas e Computação

Igor de Oliveira Knop
Doutor em Modelagem Computacional

JUIZ DE FORA
08 DE SETEMBRO, 2021

Aos amigos que fiz ao longo da jornada.

À família, pela vida.

À minha esposa Ritielen, pelo suporte incansável.

“The cave you fear to enter holds the treasure you seek.”

Joseph Campbell

Conteúdo

Lista de Figuras	4
Lista de Abreviações	5
1 Introdução	6
1.1 Justificativa	7
1.2 Objetivos	8
2 Fundamentação Teórica e Trabalhos Relacionados	9
2.1 Ensino em computação	9
2.2 Ensino de banco de dados	10
2.3 Abordagens modernas de ensino em banco de dados	12
2.4 Aprendizado baseado em jogos (GBL)	14
2.5 Jogo de dedução	16
2.6 Trabalhos relacionados	17
3 Criação de narrativas para ensino de SQL	19
3.1 Método de Pesquisa	19
3.2 Proposta de desenvolvimento	21
3.3 Arquitetura proposta	25
3.3.1 Ferramental utilizado	29
3.3.2 Configuração de partida	29
3.3.3 Partida	30
4 Experimentos	34
4.1 Prova de conceito com novas narrativas	34
4.1.1 Instanciação do metamodelo para duas narrativas	34
4.1.2 Detalhamento da implementação da Narrativa 1	36
4.2 Aplicação em sala de aula	41
5 Conclusões e Trabalhos Futuros	43
A Apêndice	47
A.1 Formulário para levantamento de requisitos com professores de banco de dados	47
A.2 Formulário para avaliação de Enola junto a alunos utilizadores após aplicação	53
A.3 Arquivo de configuração Context	58
A.4 Arquivo de configuração <i>Quotes</i>	60
A.5 Arquivo de configuração <i>Switchers</i>	65
A.6 Arquivo de configuração <i>Templating</i>	67
A.7 Metamodelo narrativo	69
A.8 Esquema do Metamodelo narrativo	70
A.9 Resolução do enredo de <i>SQL Murder Mystery</i>	71

Lista de Figuras

2.1	Triângulo didático.	9
2.2	Grafo de Modelo de Decisão de Markov para recomendar consultas. Fonte: Lavbič, Matek, e Zrnec, 2017.	13
2.3	Sistema de engajamento em jogos. Fonte: Jabbar e Felicia, 2015.	15
3.1	Facilidade de compreensão de cada conceito pelos alunos, segundo os professores.	21
3.2	Metamodelo narrativo.	24
3.3	Arquitetura do sistema desenvolvido.	25
3.4	Modelo entidade-relacionamento inspirado na narrativa de <i>SQL Murder Mystery</i>	31
4.1	Narrativa 2: Embarque no Navio Pirata.	35
4.2	Esquema do banco de dados gerado para a narrativa Embarque no Navio Pirata.	35
4.3	Narrativa 1: “Encontre a Criatura Encantada”.	36
4.4	Esquema de banco de dados da narrativa “Encontre a Criatura Encantada”.	37
4.5	Avaliação feita por alunos após uma sessão de partidas de Enola.	42

Lista de Abreviações

API	<i>Application Programming Interface</i>
DSR	<i>Design Science Research</i>
GBL	<i>Game Based Learning</i>
PBL	<i>Project Based Learning</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	Standard Query Language

1 Introdução

Métodos de ensino de tecnologia têm buscado a renovação de metodologias tradicionais utilizadas dentro de sala de aula para melhor atender à natureza interativa e colaborativa da atividade de lecionar computação. Castro e Siqueira (2017) evidenciam que alunos de computação reagem melhor a estímulos de natureza ativa do que a lições meramente expositivas. Como demonstram de Castro, Siqueira, e da Silva Monteiro (2020), o volume de conteúdo produzido para atingir essa necessidade de ensino ativo tem crescido nos últimos anos de forma evidente para temas como desenvolvimento de *software* e algoritmos. Segundo Huang e Leng (2019), o mesmo ocorre para banco de dados, área de mesma importância dentro do ensino de computação. Banco de dados não é um curso de fácil aprendizado e depende de estratégias inovadoras para levar o ensino de forma satisfatória aos estudantes.

Algumas propostas de ensino como a de Martinhago, Smarzarzo, Lima, e Guimarães (2014) se valem de plataformas e *softwares* que preocupam em exercitar conceitos de bancos de dados através de abstrações, e com isso estimulam o pensamento de banco de dados dentro de um espaço mais amigável. Mas, apesar de ter sua efetividade equiparada ao treinamento dentro de um gerenciador de banco de dados ao discutir-se os conceitos aprendidos, essa abordagem atrasa o contato com as ferramentas.

Já o aprendizado orientado a projetos (PBL) para Josko (2020) e o aprendizado orientado a jogos (GBL) para Chang, Liang, Chou, e Lin (2017) propõem que o aluno deva vivenciar os desafios inerentes da área dentro de um ambiente próximo do real, assistido por um instrutor ou professor capacitado para apoiar os alunos ao longo de atividades progressivas. Essas abordagens unidas podem estimular os alunos a um aprendizado mais satisfatório.

Buscando alinhar o exercício de conhecimentos básicos de bancos de dados relacionais e a gamificação aplicada ao aprendizado ativo, foi proposto por Joon Park (2019) um jogo de bancos de dados chamado *SQL Murder Mystery*, onde o objetivo do jogador é desvendar um mistério de assassinato utilizando dados fictícios policiais como relatos de

testemunhas, endereços, lugares, eventos e até mesmo documentos pessoais. Todos esses elementos são organizados em tabelas relacionais dentro de um banco de dados e deverão ser recuperados pelo aluno enquanto tenta desvendar quais são as relações entre as pistas por ele catalogadas.

Esse tipo de abordagem é capaz de estimular alunos a pensarem sobre como encontrar de forma rápida e intuitiva informações críticas dentro de uma base de dados, uma habilidade essencial para um profissional da área. Entretanto hoje o jogo se baseia em dados inseridos de forma manual e não possui meios para a reutilização de seu esquema de banco de dados para novas narrativas de investigação.

1.1 Justificativa

De acordo com Castro e Siqueira (2017), a velocidade da troca de informações por meios tecnológicos e a rápida evolução do mercado de trabalho têm modificado a maneira como alunos têm se relacionado com conteúdo e instrutores nos ambientes de ensino. Esses se mostram insatisfeitos com a alta carga de atividades expositivas. Com base no que apresenta Carneiro (2020), numa era dominada por dados, incentivar técnicas e métodos que estimulem a construção do ensino se torna essencial para atender à demanda crescente de um mercado que produz dados massivamente.

À medida em que novas tecnologias em bancos de dados surgem, perdura um receio em relação à qualidade dos fundamentos lógicos e de resolução de problemas ao invés de uma preocupação com o aprendizado sintático de uma ferramenta específica. Os alunos tendem a ter mais problemas em corrigir erros lógicos cometidos do que inerentes de escrita de consultas segundo Taipalus e Perälä (2019). Entretanto, muitas ferramentas e técnicas dedicam-se unicamente ao aprendizado sintático de SQL como sendo o principal fundamento formador de um bom analista, não permitindo o contato com os demais elementos presentes em um SGBD.

1.2 Objetivos

A fim de atestar sua efetividade enquanto ferramenta de ensino, esse trabalho faz uso dos métodos de avaliação propostos por Lima et al. (2020). Os métodos verificam a satisfação dos alunos ao utilizar tal ferramenta para consolidar conceitos de bancos de dados relacional de forma lúdica. A avaliação se dá através de formulários aplicados à alunos após terem finalizado partidas em Enola.

Segue-se então a lista de objetivos deste trabalho:

- Criar um sistema de geração e população de instâncias de bases de dados.
- Implementar a geração procedural de conteúdo temático dentro do atual metamodelo de história, gerando novas possibilidades de tuplas. Essa geração será personalizada a partir de arquivos de configuração juntamente com um metamodelo narrativo.
- Avaliar o uso da ferramenta utilizando como critério avaliativo os métodos citados por Lima et al. (2020) em uma turma prática de banco de dados.

Esse trabalho apresenta uma ferramenta baseada em jogos onde, para revelar o desfecho de uma trama narrativa, é preciso fazer uso de SQL para desvendar pistas necessárias à resolução. A ferramenta geradora de instâncias de jogos sérios investigativos, chamada Enola, tem suas partidas ambientadas em uma base de dados relacional. Essas não dependem de interfaces intermediárias ou instruções prévias para o entendimento dos objetivos do jogo. As instâncias de banco de dados criadas por Enola contribuem para a experiência dos alunos enquanto analistas e dos professores enquanto tutores, objetivando o aprendizado de SQL e reuso da ferramenta para outras práticas junto a turmas iniciais de banco de dados. Enola é resultado de uma pesquisa baseada em *Design Science Research* que visa alinhar as expectativas de professores e alunos em disciplinas de banco de dados.

2 Fundamentação Teórica e Trabalhos

Relacionados

Neste capítulo são apresentados desafios atrelados ao ensino em computação e ao engajamento dos alunos na disciplina de banco de dados. Ao decorrer do capítulo, técnicas modernas de ensino são discutidas. Ao fim do capítulo são apresentadas abordagens de ensino que apoiam o aprendizado baseado em jogos e sua avaliação.

2.1 Ensino em computação

Desde o início do ensino de cadeiras em Ciência da Computação os professores têm procurado formas mais efetivas de não somente passar adiante conhecimentos complexos sobre computação mas fazê-lo de forma a facilitar o entendimento de conceitos de base, segundo Carneiro (2020). Existem diversos níveis de abstração que precisam ser assimilados pelos alunos nos mais diversos campos da ciência exata, lógica, matemática, álgebra linear e teoria dos conjuntos, para exemplificar. Todos esses demandam um alto engajamento dos alunos para que haja uma produtiva consolidação do conhecimento. Para discutirmos os esforços em evoluir a didática computacional, é preciso entender os três pilares do sistema didático: aluno, professor e conteúdo, apresentado em de Castro et al. (2020). Valendo-se

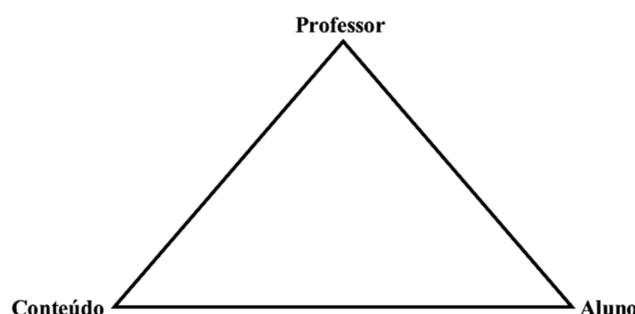


Figura 2.1: Triângulo didático.

da abstração do Triângulo Didático, cada uma de suas arestas indica uma relação que pode ser reforçada dentro da didática.

As estratégias aluno-conteúdo focam em engajar os alunos durante a aprendizagem e facilitam a recuperação dos conhecimentos absorvidos. Já estratégias professor-conteúdo são mais focadas em moldar o conhecimento científico em formas palatáveis e melhor assimiladas pelos alunos, e por fim, um contrato comportamental se forma da relação aluno-professor, que coordena como as interações devem se dar.

Sob esse prisma metafórico coloca-se em evidência as duas primeiras relações, aluno-conteúdo e professor-conteúdo, que por intermédio de sistemas e ferramentas didáticas computacionais podem ser facilitadas. Os ambientes de ensino têm se expandido para fora das fronteiras da sala de aula, criando novas óticas sobre o conteúdo lecionado. Professores e alunos têm mais formas de entrar em contato com o conteúdo de uma disciplina, o que dentro da computação se dá por meios literários, recursos audiovisuais, sistemas *online*, aplicativos móveis de ensino como o de Loewen et al. (2019) e até mesmo jogos didáticos como demonstrado em Ward (2015).

2.2 Ensino de banco de dados

Hamzah, Rukun, Rizal, Purwati, et al. (2019) observam que mesmo com as recentes evoluções e modernizações do ensino, o *status quo* gira em torno de práticas clássicas, atividades simplistas e pouco motivadoras. Ele destaca que o uso de ferramentas e métodos diferentes têm influência positiva sobre os alunos de banco de dados, aumentando a compreensão profunda e resultando em notas mais altas em testes.

No processo de ensinar banco de dados, é preciso que o aluno construa um pensamento crítico que vá além do sistema de conhecimentos a que ele é exposto, segundo Huang e Leng (2019). O mesmo analisa que o pensamento computacional no ensino de banco de dados depende de fatores cruciais listados como decomposição e redução, abstração, conversão e simplificação, classificação e divisão de ideias, pensamento reverso e premissas. Isso indica que, mais que somente uma base estrutural sobre os fundamentos teóricos, é preciso estimular os alunos em diferentes cenários e ambientes para treinamento. Esse permite que todas essas nuances do ensino computacional em banco de dados sejam cobertas, tornando a didática efetiva em toda a sua amplitude de fatores.

Aplicações de ensino de banco de dados têm tido sucesso em construir o pensa-

mento crítico e os fundamentos necessários para alcançar a plenitude nos conhecimentos que requer a área. Mesmo em assuntos abstratos como é a normalização de bancos de dados, os resultados se mostram promissores, conforme Piza-Dávila, Gutiérrez-Preciado, e Ortega-Guzmán (2017).

Antes de discutir sobre as técnicas hoje presentes no ensino de banco de dados, é preciso discutir as expectativas e preocupações de tutores e alunos, de acordo com Prabhu e Jaidka (2019). Segundo pesquisa feita pelos autores, os professores se sentem preocupados em relação ao engajamento gerado nos alunos, na falta de preparação dos mesmos para lidar com projetos e na grande dificuldade em criar formulários de pergunta e resposta uma vez que soluções de consulta em banco de dados podem chegar a resultados corretos por caminhos diversos.

Já os aprendizes sentem falta de materiais para se engajar antes da exposição à sala de aula. Sentem também que os testes finais podem pressionar os possíveis *feedbacks* que receberiam do tutor pois já não há mais tempo hábil para mudanças ou auto-críticas adequadas. Há também relatos de desconexão entre conteúdos, o que pode causar o desengajamento da turma.

Prabhu e Jaidka (2019) ainda discorrem sobre técnicas utilizadas hoje na sala de aula e expõe pontos de atenção a respeito das suas aplicações. São elas: sessão de palestras, aprendizado baseado em projetos, ferramentas *online*, aprendizagem baseada em padrões e liderança compartilhada.

A primeira, sessão de palestras, é meramente expositiva e torna os alunos agentes passivos da informação propagada. Como ponto positivo, ressalta a rápida propagação de conhecimento. Já o ensino baseado em projetos é muito útil para estimular um ambiente real de desenvolvimento, e estimula o pensamento crítico assim como habilidades sólidas. Contudo, caso o instrutor não tenha prévio conhecimento do conteúdo, pode se tornar frustrante para os alunos.

Ferramentas *online* como as que são apresentadas por Carneiro (2020) trazem liberdade e permitem o aluno a aprender no seu próprio ritmo. Existem nelas porém o desafio de auditar respostas a questões abertas de SQL, uma vez que um conjunto de dados pode ser encontrado via várias consultas diferentes, e ainda: quando a ferramenta

apresenta inferências incorretas em questões fechadas traz confusão ao aprendizado.

Outra técnica é o conhecimento baseado em padrões, idealizado para novos aprendizes no contexto de banco de dados. Contudo, com inúmeras possibilidades de consultas SQL, não existe um consenso sobre quais os padrões que devam ser amplamente utilizados.

O autor ainda discorre sobre uma última estratégia que foca em delegar o papel de instrução aos próprios alunos, criando pares de ensino colaborativo, a chamada liderança compartilhada. O ato de incentivar e pensar criticamente sobre as decisões de seus colegas traz reforço e clareza ao próprio conhecimento, mas se torna complexo incentivar aprendizes a participar.

2.3 Abordagens modernas de ensino em banco de dados

Na busca por métodos inovadores no ensino, alguns estudos se propõem a explorar formas mais abstratas de incentivo a fundamentos. Essas abstrações dão um atendimento personalizado e, quando possível, tutoria automatizada aos alunos. A seguir, são discutidas duas abordagens que seguem nesse sentido para trazer inovação ao ensino de banco de dados.

Lavbič et al. (2017) propõem um sistema de recomendação que, utilizando-se de tentativas em resolver um mesmo problema vindas de grupos diferentes de alunos, permite que novatos tenham dicas contextualizadas sem interferência de um instrutor humano. Ao mesmo tempo, a solução se adapta às decisões do aluno, não forçando um meio único de construir consultas SQL definido previamente.

Quando um estudante pede por uma dica sobre a consulta que está a fazer, o sistema aceita a atual consulta como parâmetro de entrada e a partir dela busca um estado parecido em um grafo representando o Processo de Decisão de Markov. Esse então avalia a distância entre o estado atual e consultas fornecidas por estudantes de turmas anteriores para tentar encontrar o novo estado válido mais próximo. A dica então é construída a partir do melhor próximo estado encontrado, e permite que os alunos imprimam o próprio estilo na escrita de sua resposta através de rotulações, por exemplo.

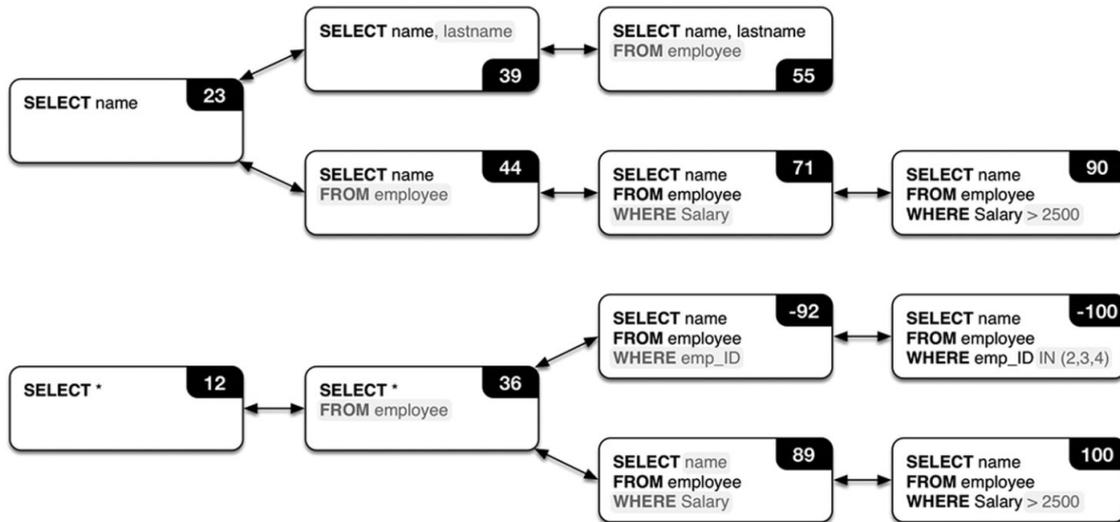


Figura 2.2: Grafo de Modelo de Decisão de Markov para recomendar consultas. Fonte: Lavbič et al., 2017.

Devido à sua natureza de atuação, a ferramenta depende de uma interface de apoio que lerá a consulta e, a partir de um processamento utilizando árvores como estrutura de dados e distância de transformação entre estados, fornecerá dicas em uma interface especializada. Vale ressaltar que seus métodos de processamento podem se tornar muito custosos dado o contexto de atuação.

Apesar de providenciar bastante autonomia e personalização ao ensino de banco de dados, os sistemas de recomendação de dicas para SQL privam o aluno do contato direto com sistemas de gerenciamento de bancos de dados para executar as consultas recomendadas.

Martinhalgo et al. (2014) dedicam-se a resolver questões voltadas à abstração necessária para aprender banco de dados. Através de um estudo semi-experimental, foram passados aos alunos conceitos lúdicos aplicados como um conjunto de atividades a se realizar em sala de aula. Os elementos das atividades foram então personificados pelos próprios alunos de acordo com o nível de abstração da atividade, de um total de quatro atividades desenvolvidas pelo autor. Essas englobavam diversos conceitos comuns em bancos de dados, como a definição sobre dados e informação e as ações “alterar” e “excluir”.

As atividades se estruturaram com um caráter progressivo, e exigiram cada vez mais camadas de abstração dos alunos, culminando em um jogo de dedução, onde a meta era desvendar um crime utilizando os comandos SQL contra uma base de dados

fictícia. O trabalho conclui que apesar de não haver uma melhora significativa entre o grupo de controle e o grupo que aprendeu através de Computação Desplugada, os níveis de satisfação dos alunos foram em média superiores, o que estimula a continuidade dos estudantes em avançar nos estudos da disciplina.

A Computação Desplugada mostra um incentivo maior aos conceitos abstratos de banco de dados, enquanto força os alunos a progressivamente aprofundarem seu nível de entendimento até as camadas elementares de operações. Porém isso também não reflete a realidade de uma análise de banco de dados, e mesmo que os conceitos sejam trabalhados de forma lúdica, foge ao exercício da linguagem de consulta e atrasa o contato com ferramentas complexas e robustas como os SGBDs.

2.4 Aprendizado baseado em jogos (GBL)

Apesar de geralmente serem utilizados de forma recreativa, os jogos permitem a experimentação de conceitos em um ambiente seguro para erros, ao mesmo tempo sendo desafiadores e motivadores. Segundo Christiane Gresse von Wangenheim (2013), o aprendizado baseado em jogos define-se por duas dimensões: é ao mesmo tempo atividade recreativa e ferramenta educacional.

Enquanto jogo, deve possuir um sistema claro e limitado de regras e objetivos que, uma vez cumpridos, colocam o jogador em vantagem competitiva que culminará em sucesso ou fracasso. Já como dispositivo educacional precisa ser projetado para ensinar determinado assunto em seu próprio escopo auto delimitado. Dentro da perspectiva instrucional, os jogos educacionais exercem caráter experimental e podem fomentar profundo entendimento por parte do aluno.

Jogos são compostos por alguns elementos principais, sendo esses os descritos pelo autor: objetivos, regras e restrições, narrativa, interação, desafio e conflito, resultados, recompensas e retornos. Podem ser competitivos, onde os alunos disputam entre si, ou cooperativos, onde os alunos se apoiam em torno de um objetivo comum.

Os jogos educacionais não apresentam grandes diferenças entre as modalidades em termos de ganhos de aprendizado, segundo Lin, Huang, Shih, Covaci, e Ghinea (2017). Contudo, como abordagens de ensino tendem a prezar pelo aumento de satisfação dos alu-

nos envolvidos e como conclusão da pesquisa, jogos competitivos apresentaram resultados melhores entre os participantes. O estudo contudo não apresenta evidências de que jogos competitivos também são os melhores para cenários onde há dificuldade dos alunos em utilizar as ferramentas ou em exercitar os conceitos envolvidos. Nesse trabalho, enfoca-se a cooperação entre os alunos e o apoio de instrutores para resolução dos desafios propostos.

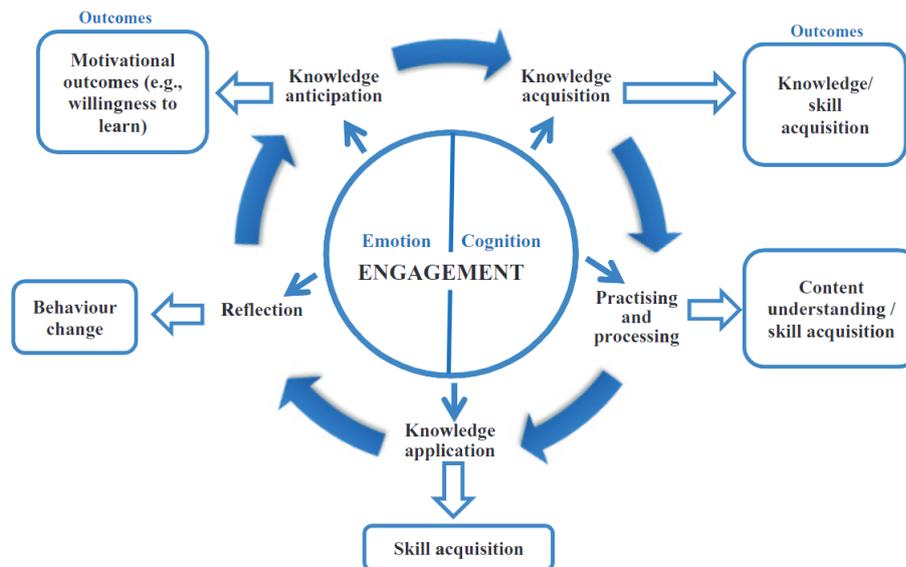


Figura 2.3: Sistema de engajamento em jogos. Fonte: Jabbar e Felicia, 2015.

Tendo ainda em perspectiva o aumento da satisfação e motivação de alunos no ensino, Jabbar e Felicia (2015) definem os pensamentos e emoções como motores centrais do engajamento por aprendizado lúdico. A cognição age como porta de entrada para os conhecimentos fundamentais utilizados no jogo educacional, enquanto a emoção ajuda a moldar o comportamento e cria respostas aos estímulos da atividade. A essa mudança comportamental e de natureza reflexiva se atribui o ganho de conhecimento sobre determinado assunto. Isso não somente reforça o conhecimento, mas permite o aluno antecipar o uso de conceitos e estar apto a absorver novos ensinamentos no mesmo modelo. Ressalta ainda que alunos expostos ao aprendizado baseado em jogos apresentam uma forma estruturada, estratégica e confiante de aprendizagem.

2.5 Jogo de dedução

Sendo temática deste trabalho, o estilo de jogo de dedução consiste em desvendar um mistério através de pistas catalogadas e interrogatórios de personagens. Contudo, esses jogos podem se tornar repetitivos ao passo em que o jogador conclui uma partida, pois os dados apontam as mesmas pistas, sem variações nas informações descobertas.

Uma solução para este problema é proposta por Mohr, Eger, e Martens (2018), e consiste em gerar novos fatos para o mundo do jogo e propõe verificá-las a partir da Lógica Dinâmica Epistêmica, a fim de gerenciar o conhecimento adquirido através do sistema Ostari. Uma partida sempre começa com a geração de personagens para o enredo, montando perfis com propriedades fisionômicas. Assim que os perfis são definidos de forma estocástica, indica-se o conjunto de dados que deve ser investigado e que fazem parte de um crime verificado como solucionável.

Os personagens se tornam então agentes dentro do jogo, tomando ações específicas que podem ou não estar relacionadas ao crime. A forma como esses agentes compartilham informação pode também variar dependendo de sua motivação em omitir, expressa por uma propriedade específica.

O jogo gera evidências nos locais do crime, e caso haja testemunhas, são removidas da cena do crime para aumentar o grau de imersão do jogador e manter o enredo interessante enquanto as testemunhas são interrogadas. O jogador então pode listar evidências e indagar testemunhas na busca para identificar o culpado, mas com um número limitado de ações. Isso permite que a partida não se resuma a acumular o maior número de provas possíveis, mantendo a possibilidade de fracasso caso o suspeito incorreto seja acusado.

A natureza investigativa do *Murder Mystery* e a versatilidade da geração de conteúdo é capaz de ser traduzida para um projeto didático. Esse processo objetiva ensinar e estimular a criação de consultas progressivamente mais desafiantes. Contudo, o trabalho apresentado por Mohr et al. (2018) demonstra uma abrangência em jogabilidade que não seria facilmente transportada para o tempo de uma aula prática de banco de dados. Para tanto, este trabalho pretende simplificar significativamente a abordagem sobre jogos de dedução, atentando-se principalmente à criar partidas com duração adequada à sua aplicação em sala de aula.

2.6 Trabalhos relacionados

Embora em menor número, propostas de jogos sérios na área de banco de dados buscam promover o engajamento dos alunos, em sua maioria, sem promover a interação direta com sistemas gerenciais de banco de dados. A seguir, discute-se os pontos de diferenciação de algumas dessas soluções propostas.

Para desenvolver um jogo sério que foca em ensinar consultas SQL básicas, Soflano (2011) criou uma modificação do jogo *NeverWinter Nights*. Com um cenário medieval e o objetivo de reunir pistas para investigar crimes, mescla as mecânicas presentes no jogo original com caixas de diálogo onde é possível fazer consultas SQL. Os personagens do jogo também auxiliam o jogador através de dicas sobre como construir as consultas. As informações coletadas são automaticamente armazenadas em um banco de dados. Para completar a missão, o jogador precisa prender várias mentes criminosas, e para tanto, obter o mandado que somente pode ser emitido caso o jogador apresente um número mínimo de provas. A adaptação do jogo se deu pelo conjunto de ferramentas chamado de *Aurora Toolset*. Por ser uma adaptação de um jogo que ainda se apoia nas mecânicas do jogo original, acaba por funcionar como extensão dos objetivos originais ao invés de um ambiente focado no ensino de banco de dados. Isso pode vir a desmotivar alunos que não tenham interesse nas atividades inicialmente propostas por *NeverWinter Nights*.

LudoSQL, de Ribeiro e Santos (2016), é um jogo de magia e fantasia ficcional, onde o jogador assume o papel de um mago que usa SQL para realizar feitiços. A narrativa se dá em uma cidade dominada por um mago poderoso, e o objetivo é destroná-lo. Como mecânica do jogo são propostos testes a serem resolvidos. Caso o resultado retornado pela consulta seja satisfatório, o jogador avança para o próximo teste. O sistema de pontos se baseia no tempo de resposta para cada desafio. Como todas as interações do jogo se dão através de uma interface de auxílio, o contato com os artefatos tradicionais de um banco de dados relacional é descartado no processo de ensino de consultas SQL.

Uma ferramenta que consegue unir o exercício lógico em banco de dados e atividades progressivas de consultas SQL é *SQL Murder Mystery*, criado por Joon Park (2019) e previamente citado como um jogo de dedução. *SQL Murder Mystery* é uma base de dados com conjunto único e imutável de tuplas que funciona como um jogo para prática

de consultas. No papel de um investigador que domina SQL, o jogador tem o objetivo de descobrir quem está por trás de um assassinato ocorrido em *SQL City*. Como fonte de informações, o detetive pode explorar a base de dados da polícia, contendo registros de pessoas, licenças de direção, endereços e tabelas de apoio para o cruzamento de informações e descoberta de pistas. Para avançar nas investigações é preciso utilizar as novas informações para explorar a base de dados e encontrar o verdadeiro vilão por trás do crime. O jogo fornece uma abordagem que enfoca o engajamento do aluno em atividades lúdicas através de uma plataforma intermediária web ou através do próprio SGBD. Como ele incentiva a exploração das tabelas e registros de forma progressiva, isso elimina a necessidade inicial de um instrutor. Vale ressaltar que ambas as abordagens exigem conhecimentos prévios de consultas em banco de dados relacionais.

SQL Murder Mystery propõe um ambiente menos abstrato que o descrito na Computação Desplugada, mas onde o pensamento crítico de um analista é igualmente fundamental para resolver o crime. Sem perder o engajamento e satisfação dos alunos, a ferramenta consegue ainda estimular a escrita de consultas e investigação dos dados.

O jogador vence caso a *trigger* de validação de solução identificar o suspeito. Mesmo apresentando uma abordagem próxima do ambiente real de banco de dados, o jogo sofre com a falta de variabilidade de dados. Seu uso se torna limitado uma vez que o conjunto de pistas para resolução da investigação é imutável. Isso dificulta o reuso por parte dos instrutores e a prática recorrente de alunos, que geralmente resolvem o desafio uma única vez antes de perderem o interesse pela ferramenta.

A solução proposta por esse trabalho se baseia em *SQL Murder Mystery* ao gerar bases de dados para descrever um desafio de dedução investigativo que se materializa em tabelas, colunas e registros consultáveis por SQL. Dessa forma, é possível exercitar a criação de consultas em um ambiente próximo do utilizado por analistas profissionais, mesmo que com um contexto lúdico. Como diferencial, essa abordagem permite a reutilização de estruturas de enredo e flexibilização do contexto da base de dados ao gerar proceduralmente dados fictícios para partidas personalizadas através de parâmetros de configuração. Isso permite que a ferramenta seja versátil e que apoie o desenvolvimento da capacidade técnica do aluno enquanto não fica distante da realidade letiva do professor.

3 Criação de narrativas para ensino de SQL

Este capítulo explicita a metodologia de pesquisa e proposta de desenvolvimento para uma ferramenta de geração procedural de jogos em SGBDs. Também trata da suíte de consultas projetada para a progressão do aluno durante uma partida em instância gerada. O capítulo fala ainda do projeto arquitetural e desenvolvimento de Enola, concluindo com comentários a respeito de como se dão as partidas dentro de um SGBD.

3.1 Método de Pesquisa

Esta pesquisa é pautada no *Design Science Research* (DSR), o qual é um método orientado à solução de problemas específicos, buscando uma solução satisfatória para o problema de forma iterativa. O uso de DSR para desenvolvimento de projetos voltados ao ensino assistido por jogos tem sido incentivado. Ceccon e Porto (2020) demonstram como essa metodologia pode apoiar o desenvolvimento de um artefato que atenda aos requisitos didáticos e cause um impacto positivo no aprendizado. Para o presente trabalho, foram adotadas as seguintes etapas da pesquisa: (i) Definição do problema; (ii) Proposta de solução; (iii) Desenvolvimento; (iv) Avaliação; (iv) Proposta de *redesign*.

A etapa de definição do problema constitui-se do estudo sobre a problemática do ensino de SQL em nível superior e o uso de jogos para ensino de SQL. Essa definição se deu por meio de pesquisa bibliográfica, levantamento de recomendações da literatura e coleta de dados com professores. A proposta de solução derivada objetivou aumentar o envolvimento dos alunos através da prática de SQL por meio do ensino baseado em jogos.

A essa conclusão associa-se pesquisas anteriores, como a de Taipalus e Perälä (2019). Com base na aplicação de testes de consultas SQL a alunos, o estudo identificou que erros de sintaxe, expressões e projeções destacam-se entre os erros mais frequentes de alunos de banco de dados quando se trata de realizar consultas SQL. Tendo isso em mente, se torna um requisito que dentro da ferramenta final haja a possibilidade de criar desafios que trabalhem essas áreas de dificuldade comum aos alunos.

Realizou-se um levantamento com professores a respeito das necessidades que são observadas dentro de sua realidade de ensino. A proficiência observada dos alunos em sala de aula ao longo da disciplina de banco de dados e os aspectos qualitativos necessários em uma ferramenta de apoio ao ensino foram os dois pontos focais do questionário. A primeira etapa do levantamento abriu espaço para os professores discorrerem sobre o perfil atual dos métodos de ensino utilizados, as dificuldades em lecionar banco de dados e quais ferramentas são hoje por eles utilizadas. Enquanto o formulário se encontra integralmente no Apêndice I, seguem os resultados compilados das respostas dos 9 participantes, sendo esses professores do ensino superior de diferentes universidades.

Os resultados obtidos podem ser verificados na Tabela 3.1 e mostram que a metodologia mais comum entre professores é a de aulas expositivas, seguida de práticas baseadas em projeto. Quando se trata de abordagens baseadas em jogos em banco de dados, mesmo que exista um interesse em utilizá-las muitos nem mesmo as encontram. Essas poderiam apoiar outras necessidades identificadas, como a falta de vínculo entre conceito e prática e consultas SQL. A partir desses resultados, é possível inferir que há uma demanda em relação a materiais lúdicos que explorem tópicos como modelagem de banco de dados e normalização.

Perguntas	Resultados
Métodos utilizados em sala de aula	Aulas expositivas e práticas (100%), abordagens de ensino baseadas em projeto (55,6%) e baseadas em jogos (22,2%)
Dificuldades de assimilação mais comuns na disciplina	Vínculo entre conceitos e prática, conceitos matemáticos e lógicos, entendimento de comandos e modelagem
Conceitos mais difíceis de lecionar	Consultas SQL, Modelagem conceitual, normalização
Uso de ensino baseado em jogos na área de TI	Não utiliza, porém há interesse (44,4%), Faz uso frequente (33,3%), Já fez uso no passado (22,2%)
Uso de ensino baseado em jogos em disciplina de BD	Não utiliza, porém há interesse (44,4%), Não encontrou abordagens em jogos para BD (33,3%), Faz uso frequente (22,2%)

Tabela 3.1: Resultado das questões respondidas por professores.

A segunda parte do questionário baseou-se no sistema proposto por Taipalus e Perälä (2019), que discrimina tipos de consulta por elementos de complexidade na construção de consultas, como número de tabelas envolvidas, uso de equações ou funções específicas como agrupamento ou ordenação. Foi utilizada a escala *Likert* para os professores responderem sobre a dificuldade dos alunos com conceitos da disciplina.

Ao analisar os dados apresentados na Figura 3.1 pode-se inferir que (i) os conceitos mais revisados ao longo de uma disciplina, como tabelas e chaves primárias se tornam de fácil assimilação para os alunos e (ii) é evidente a necessidade de uma ferramenta que incentive ao menos parte do conjunto proeminente, nomeadamente a prática de consultas SQL com expressões (booleanas, numéricas e data), aninhamento de expressões e “subconsultas”.

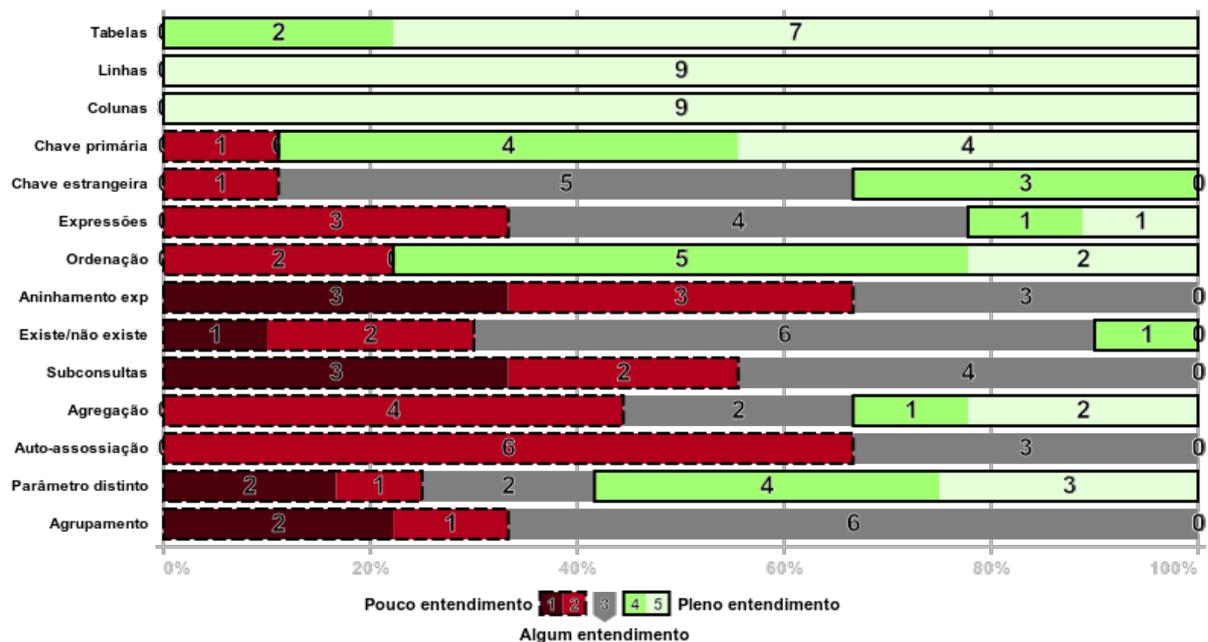


Figura 3.1: Facilidade de compreensão de cada conceito pelos alunos, segundo os professores.

3.2 Proposta de desenvolvimento

As próximas duas etapas da metodologia adotada levam em consideração a análise dos requisitos levantados e descrevem a criação do artefato computacional.

Tendo em vista as dificuldades e necessidades identificadas por meio de pesquisa e questionários, foram definidos os requisitos do sistema. Esse deve permitir a progressiva prática de consultas SQL, um potencial para uso colaborativo e reutilização em sala de aula. Para garantir que a narrativa explorada em Enola aborde os conceitos de SQL de maneira progressiva, é proposto um metamodelo narrativo (Figura 3.2) que permite o reuso da ferramenta para criação de infinitas partidas distintas sem que as mesmas

respostas possam ser inferidas sem o uso de investigação. O mesmo pode ser analisado no Apêndice A.7. Com o metamodelo, é possível garantir que o aluno usará das consultas de acordo com a progressão esperada, mas sempre encontrará diferentes resultados. O metamodelo trata do caminho esperado a ser percorrido pelo utilizador da ferramenta, que parte da tabela contendo os desafios gerados e garante que haja ao menos um caminho viável até a solução esperada.

O metamodelo utilizado como base deste trabalho é capaz de criar tabelas com nomes, colunas e tuplas de acordo com configurações externas. O metamodelo atua como um modelador de consultas, restringindo as possibilidades de pesquisas em base de dados para um conjunto de consultas de complexidade esperada. Isso significa que é possível inferir qual a faixa de dificuldade das consultas que serão realizadas pelos alunos.

O aluno terá um trabalho muito mais exaustivo para encontrar informações através de consultas de complexidade menor do que a planejada no metamodelo. Isso se dá pela criação de tuplas de ruído. Para exemplificar: num cenário onde a informação procurada depende de tuplas em mais de uma tabela, os dados de ruído impedirão a análise manual. O aluno será então induzido a exercitar funções *JOIN* com filtros específicos ao invés de diversos comandos *SELECT*.

Para fins de viabilização do protótipo, a versão atual do artefato suporta apenas um metamodelo, como mostra a figura 3.2. Porém, com este metamodelo é possível gerar diversos enredos distintos sem fugir à estrutura de consultas esperada. A cada execução de Enola, pode ser gerada uma nova instância do banco de dados com um conjunto de pistas gerados randomicamente para um mesmo enredo. O metamodelo explorado nesse trabalho foi construído sobre a narrativa de *SQL Murder Mystery*.

A partir de uma descrição do desafio, o aluno precisa recuperar através de uma consulta o objetivo da sua investigação. A tabela *Challenge* guarda as tuplas de descrição de desafios gerados pela junção dos arquivos de configuração e o metamodelo em código. Essa descrição contextualiza o aluno sobre as primeiras informações a serem pesquisadas na tabela de cenários *Scenario*. São demonstrados resultados possíveis na Tabela 3.2.

A figura 3.2 apresenta uma visualização do metamodelo. Esse abrange as tabelas e propriedades consultáveis, formando a estrutura genérica do enredo.

Consulta	Resultado esperado
<pre> SELECT description FROM challenge WHERE id = -GIVENID- </pre>	<p>A crime has taken place and the detective needs your help. The detective gave you the crime scene report, but you somehow lost it. You vaguely remember that the crime was -CRIMEARTICLE- -CRIMETYPE- that occurred sometime on -REFDATE- and that it took place in -CITY- City. Start by retrieving the corresponding crime scene report from the police department's database.</p>
<pre> SELECT * FROM crime_scene_report WHERE date = -REFDATE- AND type = -CRIMETYPE- AND city = -CITY- </pre>	<p>Security footage shows that there were 2 witnesses. The first witness lives at the -POSITION1- house on -STREETNAME1-. The other named -NAME2-, lives somewhere on -STREETNAME2-</p>
<pre> SELECT transcript FROM interview WHERE person_id = C1:id </pre>	<p>I -CRIMENOTICE- and then saw a -GENDERTYPE- run out. -GENDERSUBJECTCAPS- had a -CLUBTYPE- bag. The membership number on the bag -MEMBERSHIPPARTIALREF- '-MEMBERSHIPPARTIALVAL-'. Only -CLUBSTATUS- members have those bags. The -GENDERTYPE- got into a car with a plate that included '-PLATEPARTIALVAL-'.</p>

Tabela 3.2: Exemplos de Consultas do modelo narrativo.

Os conjuntos de dados P referem-se a propriedades a serem desvendadas. As setas indicam que haverão uma ou mais consultas SQL para acessar essas informações, que podem ou não representar todas as propriedades de uma mesma tupla. Já o conjunto C representa os personagens. Esses podem ser classificados como informantes ou personagens-alvo, e todos os personagens podem fornecer informações através de diálogos.

As informações iniciais do desafio são diretamente fornecidas pela sua descrição na tabela *Challenge*. Essas propriedades não são representadas como um conjunto a ser explorado de pistas P, e sim uma consulta direta à tabela *Scenario*.

A partir da tupla encontrada, é preciso inferir o próximo conjunto válido de informações, P0. Esse descreve o perfil de personagens a serem investigados. Essa consulta deve ser feita na tabela (*Character*) retornando informações de C1 e C2.

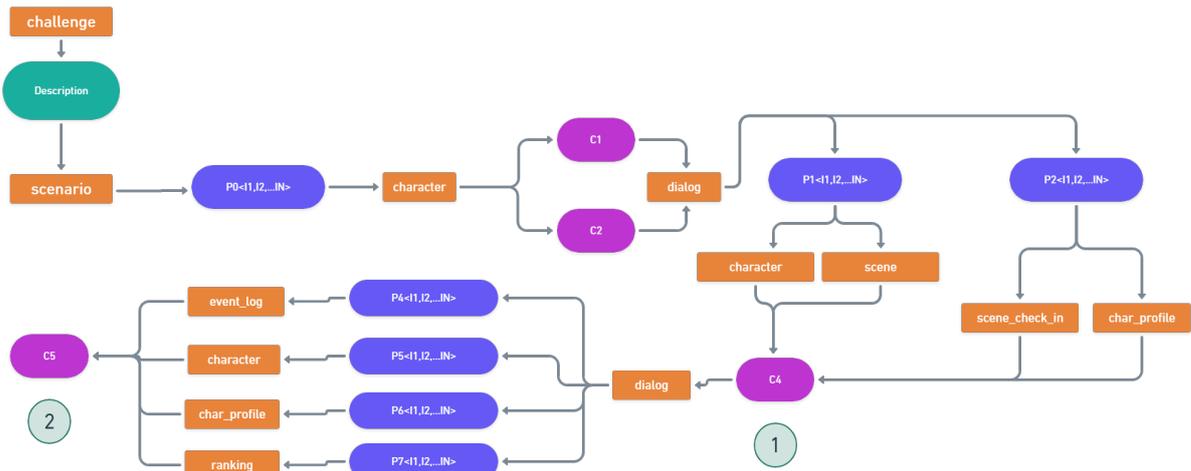


Figura 3.2: Metamodelo narrativo.

As propriedades dos personagens devem ser cruzadas com a tabela *Dialog* para recuperar seus respectivos diálogos. Dos diálogos encontrados, as pistas P1 precisam ser recuperadas através de *JOINS* entre as tabelas *Character* e *InstitutionMembership*. Esses dados retornados por si mesmos geram ambiguidade, e só podem indicar o personagem C3 caso sejam inferidas pistas através do *JOIN* entre as tabelas *InstitutionMembership* e *Profile*. Essa última consulta é proveniente dos dados inferidos de P2.

Como especificado, ao invés de levar a um novo conjunto de dados, o agrupamento dessas inferências levam a um terceiro personagem. Seu diálogo, presente na tabela *Dialog* indica agora 4 conjuntos distintos de propriedades, que levarão ao personagem C4, o real alvo da investigação narrativa.

Esses conjuntos, conforme exibido na 3.2, são: os eventos em que compareceu C3, presentes em *EventLog*; as características fisiológicas de C3 em *Profile*, *Character*; o perfil de C3 em *Profile*; e, por fim, qual o status de C3 em *Ranking*. Os resultados precisam ser agrupados e devolver como resposta quem é C3 em uma única consulta, limitando as operações do aluno e estimulando sua capacidade de sintetizar os conceitos exercitados durante o desafio.

Deve-se assumir que toda vez em que um conjunto novo de informações precise ser descoberto, o módulo responsável pelo conteúdo se preocupa em produzir dados de ruído para os arredores do contexto da busca por meio de força. Isso permite que o aluno exercite os conceitos de consulta em banco de dados conforme planejado pelo metamodelo.

Para tanto, o sistema gerará tuplas de mesmo tipo e valores randômicos.

Além de garantir que haja as delimitações necessárias para que a base de dados represente um jogo interessante e motivador para alunos de bancos de dados, a ferramenta também faz com que as consultas sejam executadas em conformidade com o entendimento técnico dos alunos. Isto quer dizer que, além de modelar o desafio, o metamodelo representa a progressão de complexidade de consultas de forma implícita.

Um exemplo disso é o conjunto de informações necessários para encontrar C3: somente é possível alcançá-lo através da interseção dos resultados encontrados de consultas utilizadas anteriormente. Essa interseção se dá por meio de *JOINS*, e qualquer ação manual para alcançar esse resultado a partir do conjunto de propriedades P1 e P2 se torna exaustivo.

O jogo motiva o aluno a sempre investigar quais são as ferramentas adequadas para um caminho mais curto até a resolução de cada etapa do problema. Com isso, reforça a intuição e capacidade técnica, habilidades fundamentais à análise de banco de dados. Ao decorrer da partida, busca-se também reutilizar sistematicamente os conceitos exercitados. Sucessivamente, a próxima consulta tende a ter complexidade sempre igual ou maior que a anteriormente utilizada.

3.3 Arquitetura proposta

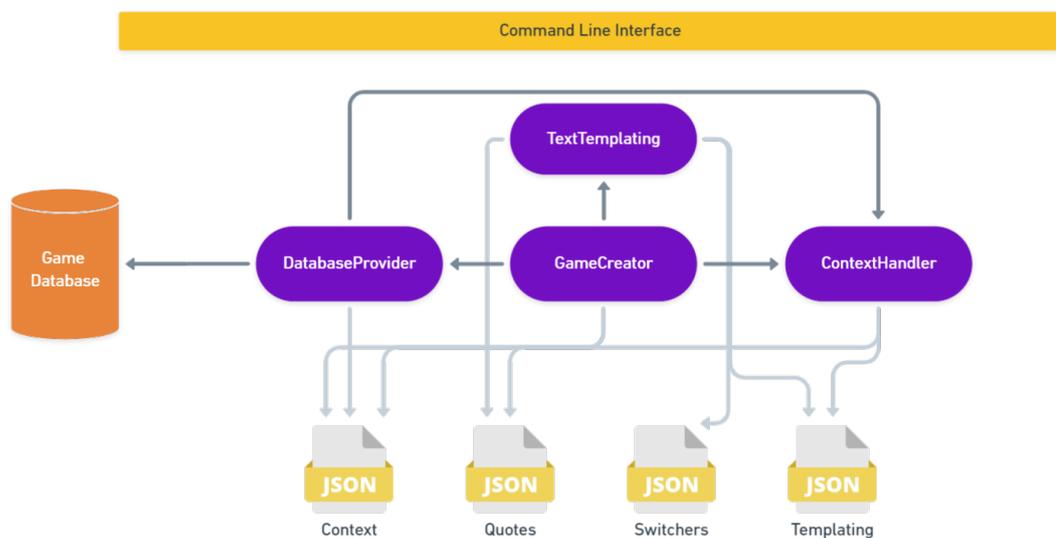


Figura 3.3: Arquitetura do sistema desenvolvido.

O sistema proposto possui uma arquitetura modular, conforme mostra a Figura 3.3. Por se tratar de uma aplicação de geração de banco de dados a serem utilizados em ambiente local, os módulos dentro da aplicação comunicam entre si de forma acoplada. A execução da aplicação produz uma base de dados relacional a partir de migrações.

Uma migração descreve o formato esperado de tabelas e colunas da base de dados, para recriação e histórico de modificações de um esquema de banco de dados. No contexto deste trabalho, foram utilizadas migrações para a geração de bases de dados.

Ao instanciar a base de dados, inicialmente vazia, a migração realiza as customizações descritas no arquivo *schema*. Este arquivo possui os nomes das tabelas e colunas, e é possível estender o módulo *schema* para que haja ainda customização de tipos das colunas.

Logo depois, executam-se operações de inserção de dados de ruído em quantidades pré-determinadas pelo instrutor nos arquivos de configuração. Vale ressaltar que o SGBD escolhido para o desenvolvimento e testes em sala de aula foi o *SQLite3*, que não suportou cargas de dados maiores que 1000 tuplas por tabela na fase de desenvolvimento.

Essas operações de inserção ocorrem por tabela, e permitem que o conteúdo das tuplas seja similar ao esperado do contexto. A elas dá-se o nome de *seeds*. Cada *seed* usa objetos modelados utilizando *Faker*¹ para geração do conteúdo de cada propriedade e entidade randomicamente definida, e insere esse objeto conforme o modelo relacional. Esses fazem uso de transformação dos dados para melhor adequar cada propriedade ao seu conjunto de dados, ou seja, cada tupla criada possui dados de mesma classe semântica. Caso a propriedade se refira a um nome, é possível que todas as instâncias geradas por um *seed* sejam também algum nome. Assim é com numerais, endereços e informações de identificação única, sempre seguindo o que estipulam as configurações de customização.

Uma vez que todas as tabelas foram inseminadas com a carga de dados solicitada, a base é liberada para estruturação da narrativa de acordo com o módulo *Game Creator*, responsável por coordenar a criação de uma partida e o fluxo de execução do metamodelo, além de persistir as entidades envolvidas. A partida é criada a partir de novas inserções na base de dados, atendendo as regras da narrativa proposta.

¹<https://github.com/marak/Faker.js/>

Essa inserção ocorre hoje através de consultas utilizando *Knex*,² sendo essas tuplas compreendidas pelo metamodelo apresentado anteriormente nesse trabalho. A responsabilidade do módulo *Game Creator* é estruturar relacionamentos entre entidades customizadas na base de dados, tanto para o fluxo principal narrativo quanto para impedir resoluções manuais dos desafios de consulta.

Para criar diálogos diversificados, o *Game Creator* depende de outro módulo, *Text Templating*, que sintetiza textos proceduralmente à partir de configuração de contextos específicos. O módulo de modelagem de texto extrai os modelos informados no arquivo de configuração de frases. Esse arquivo contém uma lista de textos customizáveis que obedecem o seguinte formato JSON:

```
1 {
2   "name" : "example",
3   "base_quote" : "Lorem ipsum sit <REPLACEMENT>.",
4   "entities" : {
5     "name" : REPLACEMENT,
6     "type" : "VALOR",
7     "context" : "EXEMPLO.JARGAO"
8   }
9 }
```

Listagem 3.1: Exemplo de configuração de Modelagem de texto

Na listagem 3.1, *name* é o identificador do texto a ser customizado. A propriedade *base_quote* é o texto modelável a sofrer as substituições de acordo com a especificação de entidades. *Entities* consiste na lista de entidades a atuarem como variáveis de substituição de conteúdo nos modelos textuais. Essas representam o formato do elemento a ser provisionado pela aplicação como substituto no texto. Elas podem possuir vários tipos e o formato de seu valor varia conforme o tipo do elemento substituto. Os tipos suportados são *OptionsElement*, *NativeType*, *Partial* e *Switcher*.

O primeiro, *OptionElement*, é o mais customizável dos tipos por permitir que o especificador do jogo defina um vetor de opções possíveis para substituição, mas restringe

²<https://knexjs.org/>

a faixa de tipos possíveis a que pertence a entidade em questão. Já *NativeType* representa uma série de tipos de elemento pré-implementados e expostos pela aplicação como alternativa para o valor do elemento substituto.

Para entender os demais tipos é preciso entender seu contexto de uso: *Partial* é apenas a representação de parte do elemento substituto e é utilizado em situações onde a informação completa precisa ser descoberta por investigação do jogador; e *Switcher* representa um grupo de elementos substitutos pertencentes a uma mesma classe, e é utilizado quando diversos elementos substitutos trabalharão em conjunto para prover uma harmonia entre elementos narrativos. Por sua vez, *context* é um atributo opcional e representa a tabela e coluna em que o elemento substituto será persistido, e.g. tabela EXEMPLO e coluna JARGÃO. Este, em conjunto com o identificador *name*, apoia a construção de objetos concisos dentro da narrativa.

Uma vez que o sistema processa as configurações, as entidades a serem substituídas no modelo de texto são selecionadas estocasticamente, dado os seletores especificados e faixa de operação descrita na configuração. Isso se dá pelo *Context Handler*, componente responsável por gerar dados de acordo com o modelo esperado pelo *Text Templating* e por também prover o conjunto de informações que posteriormente serão distribuídas pelas entidades como pistas para a resolução da narrativa em questão pelo *Game Creator*. Esses componentes hoje trabalham de forma intimamente conectada.

Ao mesmo tempo em que o *Text Templating* percorre elementos dos textos modeláveis, requisita ao *Context Handler* que gere conteúdo para esses elementos e os insere em um saco de propriedades para posterior consolidação, isto é, inserção coerente com a narrativa dentro do *Game Creator*. O *Context Handler* atua como a implementação do metamodelo dentro dos diálogos e agregador de pistas narrativas. As informações em texto que serão relevantes como propriedades são armazenadas numa estrutura auxiliar para consolidação posterior.

Em seguida, as entidades centrais da narrativa são instanciadas. Após a sua inserção no banco de dados, a *trigger* de verificação de solução é atualizada com as informações de conclusão da narrativa. Essas informações mudam os dados referentes à diálogos especiais que somente são inseridos na tabela de solução caso personagens-alvo

sejam inseridos. Ela garante a verificabilidade da solução a cada etapa.

3.3.1 Ferramental utilizado

No desenvolvimento utilizou-se a linguagem Typescript com ambiente de execução NodeJS como apresentado em Bierman, Abadi, e Torgersen (2014). A geração de bancos de dados relacionais utilizou como SGBD *SQLite3*, e *Knex*. *Knex* é uma abstração da camada de persistência que permite realizar operações comuns de bancos de dados sem a necessidade de comprometer-se com uma sintaxe específica. Para a inserção de dados de ruído e geração procedural de informações relevantes para a contextualização da narrativa, foi utilizada a biblioteca *Faker*, responsável por funções de criação de conteúdo que fornecem uma grande variedade de conjuntos de dados.

3.3.2 Configuração de partida

O banco de dados, bem como as entidades persistidas relacionadas à narrativa e a *trigger* verificadora de solução constituem os elementos de configuração da partida. A partir de um arquivo de configuração contendo os nomes das tabelas a serem criadas, a aplicação executa a migração responsável pela geração das tabelas ligadas ao jogo em questão. São criadas duas tabelas para apoiar o jogador ao início e ao decorrer das partidas.

A primeira tabela serve para registrar desafios, isto é, jogos disponíveis na instância em questão. A segunda tem como propósito validar informações encontradas durante partidas, onde uma *trigger* é acionada a cada tentativa de inserção. Essa é responsável por verificar se a informação é um resultado parcial ou final para o desafio. Além dessas, são criadas tabelas relacionadas diretamente com a resolução do desafio, e que consequentemente deterão os dados a serem pesquisados pelo jogador com o objetivo de avançar na narrativa do desafio.

Para fins de prototipação, a estrutura das tabelas e seus respectivos relacionamentos foram especificados sem a possibilidade de customização através de arquivos. Após a etapa de migração das tabelas, fica estabelecida a inserção em cada tabela do contexto do desafio de um número pré-determinado de tuplas na base de dados. Essas servem como impedimento à resolução manual do desafio em questão, uma vez que torna impraticável

a análise de dados puramente humana. Entretanto, ao longo das investigações realizadas pelo jogador, é perceptível que esses dados não têm o objetivo de prover contexto ao desafio em si. Essa etapa se torna possível graças ao uso da biblioteca de geração procedural de conteúdo *Faker*, que permite que os dados sejam similares no formato de seu conteúdo aos dados que são relevantes à investigação.

3.3.3 Partida

Ao concluir a etapa de configuração, o instrutor pode então gerar uma instância de partida para o SGBD escolhido e percorrer a base de dados resultante através de comandos SQL. O início da partida acontece ao escolher a narrativa a ser solucionada na tabela de desafios. Essa apresenta uma breve descrição e as primeiras pistas para iniciar a narrativa.

Para melhor compreensão da sequência de consultas descrita a seguir é possível ver a resolução de SQL Murder Mystery no anexo A.9. O enredo que inspirou este trabalho possui uma trama análoga à descrita pelo metamodelo, e portanto serve para representar as consultas aqui exploradas. À medida que o usuário executa novas consultas na base de dados, este explora registros únicos e que não podem ser facilmente encontrados utilizando uma simples varredura manual dos registros. Assim, a cada retorno correto, uma nova pista é proposta, com o objetivo de avançar no contexto da narrativa até que se encontre um ou mais alvos.

Alvos podem ser confirmados através da sua inserção na tabela de solução: é através de uma *trigger* verifica se o alvo faz parte do conjunto de solução. Caso esse alvo seja um intermediário na narrativa, a *trigger* irá inserir na tabela de soluções um registro contendo a descrição do resultado juntamente com novas pistas. Mas caso esse seja o alvo principal da narrativa o jogo chega ao seu fim.

Cada alvo encontrado possui sua própria curva de dificuldade no uso de SQL, de acordo com o metamodelo. Vale ressaltar aqui que o jogo dispensa a natureza punitiva utilizada comumente em jogos, sendo que dentro de uma partida não é possível perder pontos. O que quer se minimizar aqui é o tempo de procura pelas pistas e alvos da narrativa.

Cada configuração de partida gera conjuntos de dados únicos e a mudança de con-

texto nas configurações pode estender a longevidade da ferramenta enquanto um recurso letivo para os professores de banco de dados.

O conjunto de consultas esperadas em cada etapa do metamodelo permite perceber a relação entre a evolução da narrativa e aumento da dificuldade das consultas. A primeira consulta é direcionada, ou seja, o instrutor indica ao aluno qual tupla deve ser selecionada da tabela *Challenge*. Seu esquema entidade-relacionamento se dá como na Figura 3.4, e com os arquivos de configuração é possível alterar tanto a possibilidade de novos dados para esse enredo quanto novos enredos para o mesmo metamodelo.

A partir daqui, vale a interpretação por parte do aluno das informações disponíveis, incentivando o senso crítico e investigativo: através das propriedades e tabelas identificadas no texto o aluno precisa realizar uma seleção um pouco mais complexa, utilizando o comando *WHERE* e filtros lógicos encadeados na tabela referente ao cenário encontrado. Ao fazê-lo, irá se deparar com o primeiro conjunto de pistas. Para encontrá-las é preciso distinguir as tuplas desejadas através de duas seleções distintas, ou uma união entre seleções.

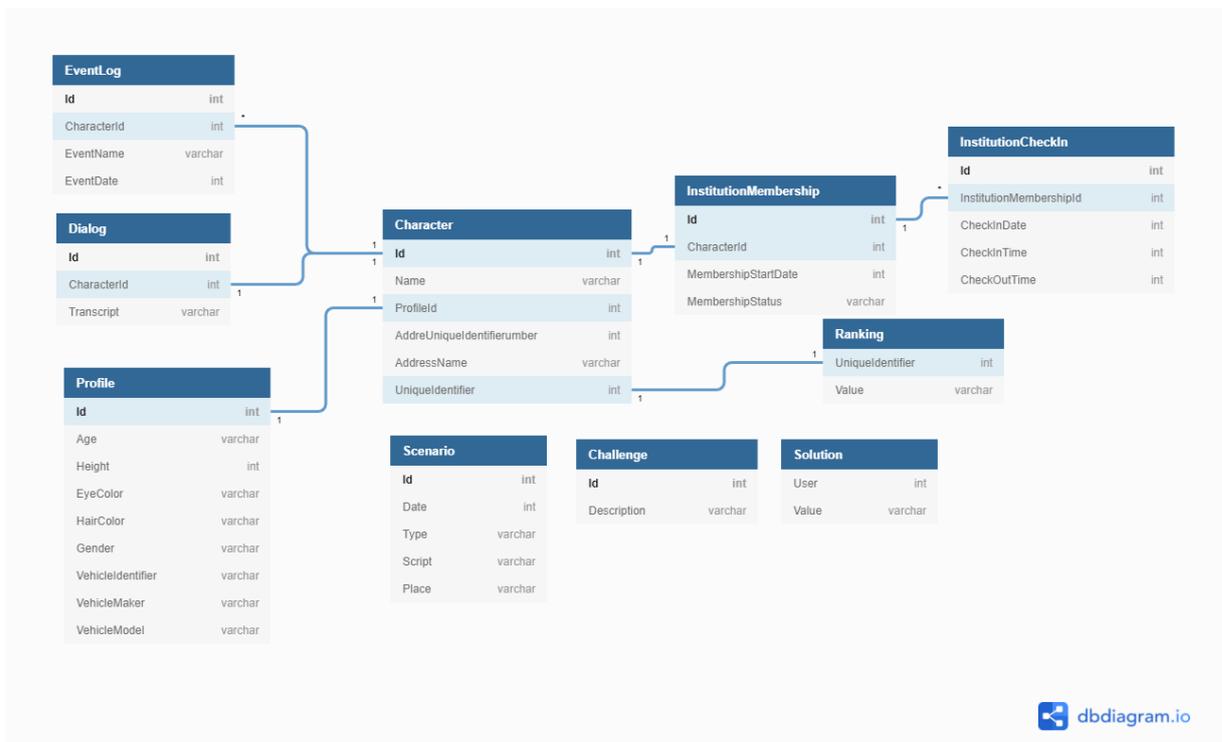


Figura 3.4: Modelo entidade-relacionamento inspirado na narrativa de *SQL Murder Mystery*.

A primeira consulta consiste em um *SELECT* com *WHERE* e precisa que as

tuplas retornadas sejam ordenadas por uma propriedade específica. A segunda também é um *SELECT* com *WHERE*, mas que agora faz uso do operador *LIKE*. Pequenas variações no uso de uma técnica-base são realizadas de modo a exercitar os conceitos de SQL, sem que o aluno se encontre em cenários completamente diferentes a cada nova consulta.

A partir dos personagens descobertos, é possível ler a transcrição de seus diálogos utilizando dois *SELECTs* distintos ou um *SELECT* e o operador *IN* para descobrir o que dizem os personagens na tabela *Dialog*. Uma terceira opção, um pouco mais complexa, seria trazer os resultados anteriores de uma união das “subconsultas” também utilizando o operador *IN*.

Contudo, o metamodelo propõe qual será o limite inferior de complexidade de consulta aceitável para a resolução de um passo do desafio, não impedindo que alunos de maior experiência alcancem resultados equivalentes mais rapidamente.

Vale reiterar que, a cada novo diálogo, existe uma interação lúdica e que exige do aluno atenção ao extrair informações da narrativa. Para o próximo conjunto de consultas a complexidade aumenta consideravelmente, uma vez que ambos diálogos anteriores abrem conjuntos de pistas que exigem distintas técnicas SQL para descobrir o novo personagem.

Conforme o enredo exige, a ferramenta garante que haverá ruído nas tuplas retornadas para que nenhuma consulta de complexidade menor que a esperada alcance o objetivo investigativo por força bruta. O conjunto de pistas P1 é resolvido minimamente por uma seleção e “subconsulta” com *WHERE* e operador *IN*. Já o conjunto de pistas P2 é resolvido por encadeamento de filtros de data e diferença em uma consulta à tabela *InstitutionCheckIn* seguido de nova consulta com os resultados obtidos à tabela *Character*.

Vale ressaltar, não é exigido que a tupla-objetivo seja obtida em uma única consulta. Ao fim da exploração das pistas fornecidas pelos diálogos, é preciso fazer o *JOIN* das propriedades resultantes para encontrar o próximo personagem da narrativa, e por fim seu diálogo na tabela *Dialog*.

Este fornece ao aluno-investigador quatro novos conjuntos de informações em um único diálogo, e cada conjunto terá de ser explorado individualmente nas tabelas *Ranking*, *Profile*, *Character* e *InstitutionCheckIn*, valendo-se dos operadores *DISTINCT*, *JOIN*, *WHERE*, *IN* e “subconsultas” com *JOIN*, *WHERE*, operadores lógicos e aritméticos.

Para esse desafio é exigido que os resultados sejam alcançados em uma única consulta final. A fim de proporcionar uma melhor compreensão do enredo de propósito genérico, um enredo concreto é discutido no capítulo 4.

A progressão de dificuldade nas consultas utilizadas para solucionar o desafio apresentado foi definida com base nos resultados encontrados junto aos professores. O banco supracitado será o ambiente utilizado pelo aluno para praticar a consulta em SQL, e define os limites do jogo. Isso permite que o aluno possa validar suas habilidades em SQL, utilizando *SQLite3* como seu SGBD. Também permite que o instrutor possa se envolver ativamente no apoio à alunos e viabiliza sua autonomia, uma vez que faz uso de ferramentas conhecidas do dia-a-dia do professor.

4 Experimentos

Este capítulo dedica-se a documentar a aplicação da ferramenta com alunos e sua respectiva avaliação. Também explora uma nova narrativa para o metamodelo proposto, e demonstra como através de arquivos de configuração é possível customizar o fluxo da história mesmo sem modificar o metamodelo do banco de dados.

4.1 Prova de conceito com novas narrativas

Nessa seção propõe-se dois novos modelos narrativos, com propósito de demonstrar como outras histórias podem ser produzidas a partir do metamodelo. O experimento traz um olhar voltado para a criação de novas configurações por parte do instrutor ao utilizar Enola. Os fluxos das histórias aqui apresentadas diferem do modelo principal inspirado em *SQL Murder Mystery*, aplicado em sala de aula e utilizado neste trabalho para explicar o desenvolvimento do projeto até então.

4.1.1 Instanciação do metamodelo para duas narrativas

Narrativa 1: A história é sobre a capitã de um navio pirata que, antes de começar a navegar, busca em uma Companhia Pirata informações sobre sua aventura. Para essa história foi escolhida randomicamente a Companhia Tesoureira do Ouro, onde as missões são de caça ao tesouro. Esse modelo narrativo tem sua progressão conforme descrito na Figura 4.1, e é análogo em termos de complexidade ao metamodelo principal. As diferenças aqui ocorrem dentro da perspectiva lúdica e terão impacto significativo sobre as propriedades textuais. Como os tipos se mantêm, as propriedades que mudaram de nome sofrem pequeno ou nenhum impacto do ponto de vista de consulta, mas foram modificadas para se adequar melhor ao enredo de pirataria.

O desafio a ser resolvido pelo aluno utilizador sempre será como o texto de uma carta de curso. O personagem principal aqui é o navio, assim como os personagens alvo. A companhia da qual pertence o navio principal armazenam suas rotas de comércio lis-

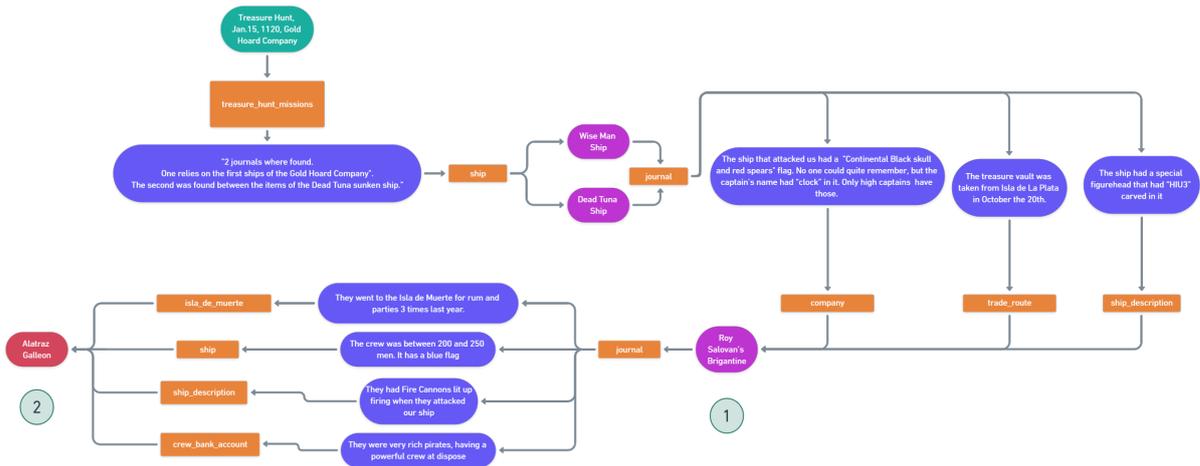


Figura 4.1: Narrativa 2: Embarque no Navio Pirata.

tadas no que antes era dentro do metamodelo principal a tabela *InstitutionCheckIn*. As descrições de diálogo ou narração da história são feitas em diários de bordo. Os eventos aqui ocorrem em uma ilha, a *Isla de Muerte*, onde festas só possuem registro de início, sem se saber quando acabaram. O esquema do banco de dados final é exemplificado na Figura 4.2.

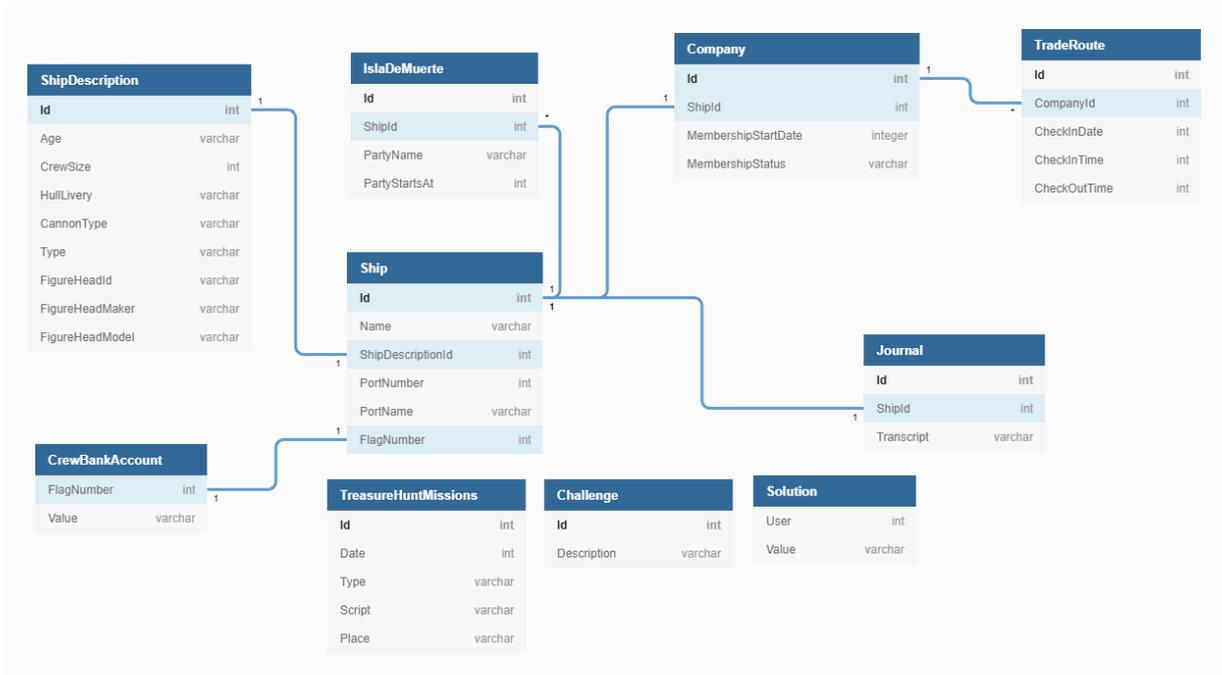


Figura 4.2: Esquema do banco de dados gerado para a narrativa Embarque no Navio Pirata.

Narrativa 2: A história coloca o utilizador no papel de um gnomo que está a procura de uma fada dentro do bosque encantado. Ele deve conversar com as criaturas do lugar para encontrar pistas da fada e levá-la ao rei. O que acontece é uma pequena mu-

dança de planos que o faz entrar numa conferência de fadas. Essa narrativa é demonstrada pela Figura 4.3.

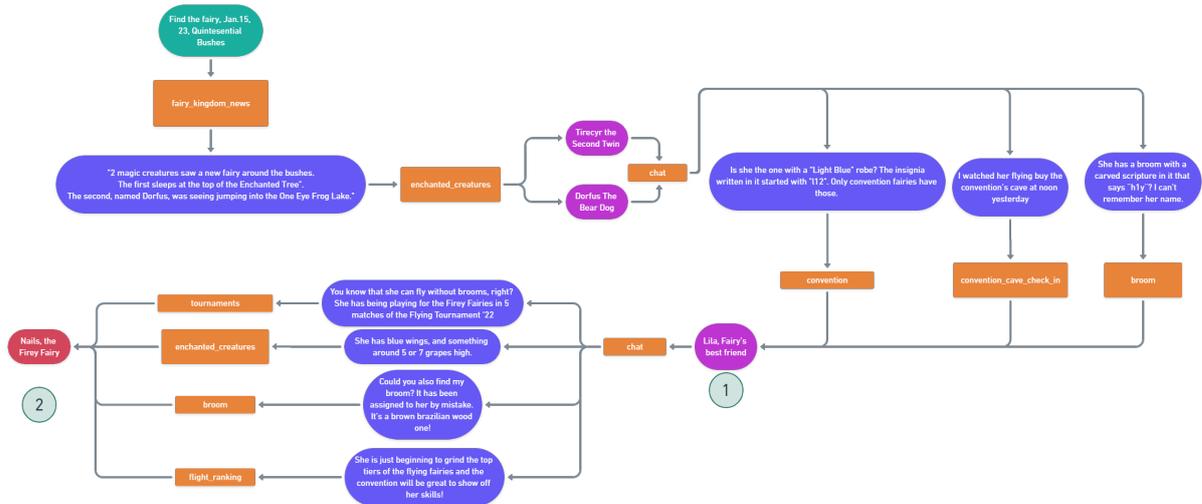


Figura 4.3: Narrativa 1: “Encontre a Criatura Encantada”.

Tanto o nome de tabelas quanto de colunas fazem alusão ao cenário de fantasia. Um exemplo claro é o identificador único da tabela *Profile* que apesar de continuar a ser uma propriedade do tipo inteiro agora representa algo menos convencional: o número de castanhas da Árvore da Vida no momento em que a criatura encantada nasceu. É esse nível de detalhe e cuidado com a coesão textual que eventualmente se refletirá num alto nível de imersão dos alunos. Mesmo em um ambiente limitador como o de um esquema de banco de dados, é possível criar narrativas com características de interesse dos utilizadores. Por sua vez, o esquema de banco de dados final será o demonstrado pela Figura 4.4.

4.1.2 Detalhamento da implementação da Narrativa 1

Considerando a Narrativa 1, podemos explorar em detalhes o passo a passo da criação de uma nova narrativa em Enola. Quatro configurações precisaram ser criadas, *context*, *quotes*, *switchers* e *templating*. Todas essas configurações são arquivos JSON que representam as diferentes estruturas citadas anteriormente para a customização de cada narrativa. Um exemplo completo de todos os arquivos de configuração da narrativa do modelo principal, inspirado em *SQL Murder Mystery*, está no Apêndice A.3 para eventuais consultas. A seguir é descrito o passo-a-passo para criação da narrativa “Encontre a Criatura Encantada” e são destacados de seus arquivos de configuração trechos de código que auxiliam

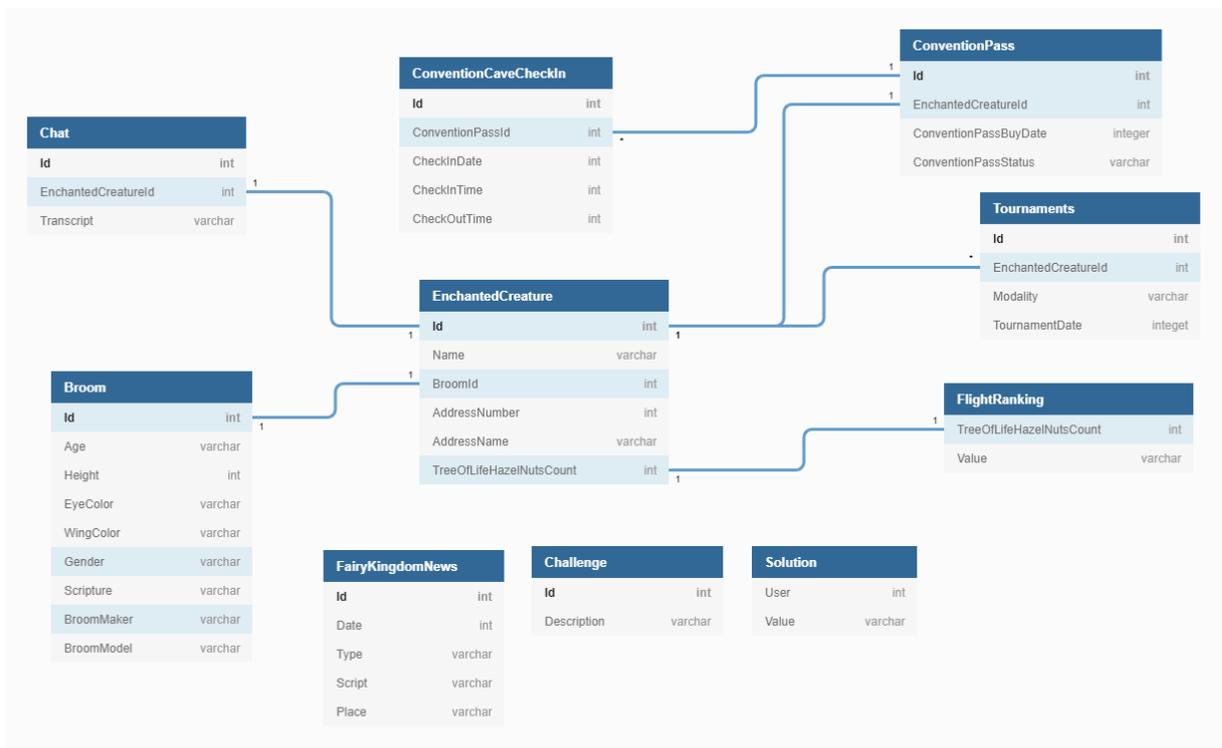


Figura 4.4: Esquema de banco de dados da narrativa “Encontre a Criatura Encantada”.

no entendimento da utilização de Enola.

Context é responsável pela personalização do contexto do banco de dados em si, isto é, nomes de tabelas e colunas do esquema. Dada as limitações do protótipo implementado, no momento não é possível alterar o tipo das colunas de forma fácil. Os tipos seguem o mesmo esquema de banco de dados que pode ser encontrado no Apêndice A.8. É preciso intervir no módulo *Game Creator* para realizar quaisquer alterações no modelo entidade-relacional. O formato do objeto esperado é como demonstrado pela listagem 4.1.

Listagem 4.1: Exemplo de configuração customização de Esquema do banco de dados

```

2  "profile": {
3    "name": "broom",
4    "columns": [
5      "age",
6      "height",
7      "eye_color",
8      "wing_color",
9      "sign",
10     "scripture",
11     "broom_maker",
12     "broom_model"
13   ]
14 },

```

A primeira propriedade do objeto representa o nome da tabela a ser gerada no

modelo final, que substituirá a predefinição do metamodelo na etapa de migração. Isso permite a manipulação das tabelas dentro da ferramenta sem que se determine o nome final da tabela dentro do esquema de banco de dados da instância gerada. Assim também é para as colunas. No exemplo acima a tabela predefinida *Profile* será agora chamada de *Broom* no esquema final. A lista *columns* representa ordenadamente os novos nomes das colunas da tabela *Profile* do esquema padrão. O objetivo aqui não é trazer uma diversidade de tipos para as consultas do metamodelo, e sim uma possibilidade de percorrer o fluxo narrativo novo através de consultas análogas.

O esquema de banco de dados dinâmico garante que quando as migrações das tabelas ocorrerem e a instância for gerada, a mesma terá os novos valores. Esses serão embutidos também nas chaves estrangeiras que compõem as relações entre entidades do metamodelo. Dentro do *Game Creator* e do *Text Templating*, as informações descritas no arquivo *Context* acessadas através do objeto de configuração de esquema vão ser inseridas dentro de tuplas da narrativa conforme especificações do arquivo *Templating*.

Listagem 4.2: Exemplo de configuração de Modelagem de texto

```
216 {
217   "name": "target_one_found",
218   "base_quote": "You run across the dirty road at the enter of the cave and see a fairy that fits
                the description. When you touched her shoulder, how surprised you were! You didn't found
                the {FAIRY_TYPE}, but wait, not everything is lost. Talk to the {FAIRY_FRIEND} to find the
                remarkable fairy the king is so eager to know. If you feel especially confident in your
                SQL skills, try to complete this final step with no more than 2 queries. Use this same
                INSERT statement with the name of the fairy to check your answer.",
219   "entities": [
220     {
221       "name": "FAIRY_TYPE",
222       "switcher": "FAIRY.TYPE"
223     },
224     {
225       "name": "FAIRY_FRIEND",
226       "switcher": "FAIRY.FRIEND"
227     }
228   ]
229 },
```

Quotes é responsável pela customização dos diálogos da narrativa, e indica as variáveis de substituição a serem utilizadas, assim como quais delas se tornarão propriedades de entidades no futuro, as propriedades de contexto. Como exibido na listagem 4.2, a propriedade *name* faz referência direta ao metamodelo, e não deve ser alterada. O *Game Creator* criará uma entidade fada com o nome correspondente dentro da base de

dados. O mesmo garante que o nome esteja na tabela de solução no momento em que a fada amiga for encontrada pelo utilizador e a *trigger* de verificação seja ativada por tentativa de inserção manual. Já a propriedade *base_quote* contém a narrativa do encontro entre o utilizador enquanto no papel do gnomo enviado pelo rei, e a criatura encantada que equivocadamente encontrou pensando ser o objetivo da missão. Existe ainda um conjunto de entidades descritas no arquivo. Essas dizem respeito a substituição de variáveis na frase, e também qual será a estratégia de geração de conteúdo associada. Nesse caso é um *switcher*. As variáveis *FAIRY_TYPE* e *FAIRY_FRIEND* serão substituídas por um dos possíveis conjuntos descritos no arquivo *Switchers*.

Switchers guardam as estruturas de comutação de variáveis que possuem um cenário compartilhado, onde não faz sentido geração de conteúdo de forma aleatória. Dos *switchers* planejados para essa narrativa, o *switcher CLUB* é usado como exemplo na listagem 4.3. *Switchers* podem ter diversos valores, mas para a história da fada em específico, foi considerado somente um cenário. Vale ressaltar que quanto mais cenários possíveis forem descritos no arquivo, mais diversificada se tornam as instâncias de jogo. Um dos componentes do módulo *Text Templating*, chamado também de *Switcher* fica responsável por fazer a comutação de cenários por execução.

Listagem 4.3: Exemplo de configuração de *Switcher*

```
37 {
38   "name": "CLUB",
39   "cases": [
40     {
41       "TYPE": "convention",
42       "ACTIVITY": "flying by the convention's cave"
43     }
44   ]
45 },
```

A próxima modificação necessária na ferramenta é a criação de um arquivo *Templating*. Esse guarda customizações de opções de geração procedural para diálogos e propriedades de contexto que serão posteriormente recuperadas pelos componentes *Fake* e *Utils*. Não somente essas opções moldam a narrativa descrita na instância, mas também todas as entidades geradas para causar ruído no retorno de consultas manuais que pretendam resolver o desafio com força bruta. O *Game Creator* faz uso dos componentes alimentados por esse arquivo para instanciar as tuplas descritas pelo metamodelo utili-

zado. A listagem 4.4 exemplifica o *templating* de geração de nome de torneios que serão inseridos na correspondente tabela *Event* do metamodelo, chamado de *generateEventName*.

Listagem 4.4: Exemplo de configuração de *Templating*

```
37 {
38   "name": "generateEventName",
39   "options": [
40     "Wings",
41     "Aces",
42     "Hummingbirds",
43     "Nightingales",
44     "Eagles",
45     "Hawks"
46   ]
47 },
```

Com essas adequações aos arquivos de configuração, é possível obter variações da narrativa “Encontre a Criatura Encantada” dentro do metamodelo estabelecido por padrão a cada nova execução.

Para executarmos Enola, é preciso que o ambiente NPM esteja configurado e os pacotes instalados. Depois, basta executarmos o *script* principal:

```
$ npm run enola
```

Esse *script* desencadeia uma sequência de outros *scripts* a serem executados sequencialmente: a limpeza de outras instâncias anteriores de base de dados, a migração que criará a base de dados *SQLite3* com o esquema já customizado, o *seeding* da base e o *start* da aplicação principal. Ao fim, uma instância contendo o jogo correspondente pode ser encontrada na raiz do projeto.

Na possibilidade de que alguma propriedade da narrativa precise ter tipos diferentes, uma tabela precise ter mais (ou menos) colunas, ou que tabelas sejam removidas ou adicionadas essas alterações podem ser feitas usando a API de construção de consultas do *Knex* nos arquivos do componente *database*. No arquivo *tables* é possível alterar o formato das tabelas, e as instâncias randômicas a serem inseridas pelo *seeder* estão descritas nos arquivos *seed*. Se o instrutor pretende ainda alterar ou expandir o esquema de banco de dados a ser utilizado, é preciso reimplementar o módulo *Game Creator*. Quaisquer alterações no esquema de banco de dados também acarretam em modificações nas consultas planejadas para o metamodelo vigente.

A partir de um metamodelo inspirado na iniciativa de *SQL Murder Mystery* foi criada uma ferramenta versátil, capaz de engajar o professor na sua reutilização com diversas turmas e narrativas. Por se tratar de uma aplicação lúdica em banco de dados que sempre reproduz o esquema determinado pelo módulo *Game Creator* que está aberto para as edições do instrutor, Enola dá também vias de controle da aplicação sem dependência direta da narrativa gerada.

4.2 Aplicação em sala de aula

As duas últimas etapas da metodologia adotada dizem respeito à avaliação da solução e da proposta de *redesign* com base nos resultados encontrados. Para avaliação do artefato, foi realizado um experimento com 31 alunos de cursos de Ciência da Computação, Sistemas de Informação e Engenharia Computacional da Universidade Federal de Juiz de Fora, como uma atividade do curso de Banco de Dados I. Todos os participantes estavam na nona semana de aula, onde tinham terminado de estudar os conceitos básicos de SQL e não tinham ainda praticado em ambiente real assistido por SGBD. Por conta do cenário pandêmico de 2020 e 2021, a sessão de prática foi realizada de forma remota e síncrona.

A problemática foi apresentada aos alunos e estes foram divididos em grupos de até cinco participantes. Em seguida, cada grupo foi associado a uma sala virtual reservada (*breakout rooms*) para que pudessem discutir e colaborar para chegar à solução do problema. Uma instância do banco de dados utilizando como enredo a trama de *SQL Murder Mystery* foi gerada automaticamente para cada grupo. O mesmo nível de dificuldade se mantém em cada instância, mas cada uma contém tuplas distintas das demais.

Os grupos terminaram a atividade entre 85 e 110 minutos e todos os grupos conseguiram alcançar os dois alvos do problema. Ao final da sessão de prática, os participantes responderam a um questionário avaliativo sobre a ferramenta e experiência de uso. O questionário foi desenvolvido com base na meta-análise proposta por Lima et al. (2020), que reuniu critérios essenciais à jogos sérios: a consolidação do conhecimento, usabilidade/autonomia, diversão/motivação, satisfação, diversidade de desafios, imersão, relevância, clareza, percepção de progresso, interação social, confiança e concentração. O

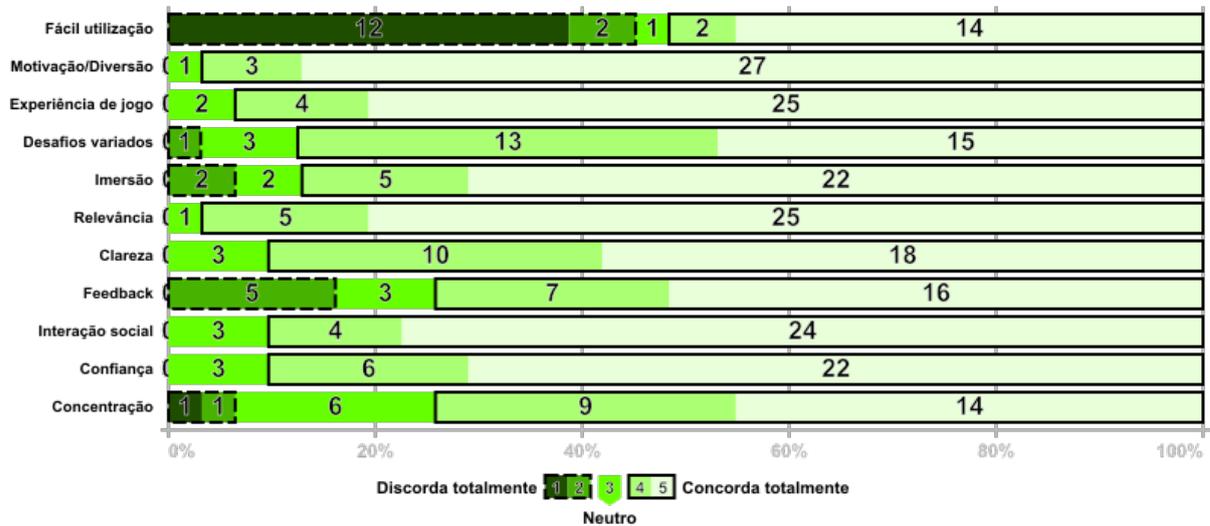


Figura 4.5: Avaliação feita por alunos após uma sessão de partidas de Enola.

mesmo encontra-se no Apêndice A.1 para eventuais consultas.

Como demonstrado na Figura 4.5, os critérios analisados foram atendidos de forma satisfatória, tendo, porém, uma diferença significativa na média de usabilidade. Essa pode ser explicada pela falta de contato prévio com SGBDs e pela necessidade de realizar as operações sobre um sistema de banco de dados sem uma camada de apresentação facilitadora. Entretanto, é perceptível que isso não acarretou em uma insatisfação dos alunos, que avaliaram Enola positivamente em todos os demais critérios.

Por fim, na etapa de proposta de *redesign*, tendo como base o resultado da sessão prática, identificou-se novos requisitos para uma segunda iteração da metodologia. Verificou-se a necessidade de mais metamodelos junto ao professor aplicador, assim como uma maior quantidade de fluxos narrativos. Observou-se ainda pelo professor que algumas das possibilidades de consulta poderiam ser identificadas através de pesquisa manual, e isso poderia acarretar em encontrar os dados sem que os conceitos desejados fossem exercitados. O código do artefato foi disponibilizado no Github³.

³<https://github.com/jgabrielmaia/enola>

5 Conclusões e Trabalhos Futuros

Utilizar um jogo sério para transmitir os conceitos básicos da área de banco de dados pode ser essencial para o estímulo e engajamento de alunos em sala de aula. Enola se propõe a exercer essa função aliando narrativas lúdicas a um ambiente próximo do que é vivido por instrutores do meio acadêmico e profissionais de mercado.

Esse trabalho coloca a disposição de professores uma ferramenta de código aberto customizável para a criação de jogos investigativos em bases de dados. Diante das necessidades levantadas pelos professores e seu desconhecimento de ferramentas de apoio ao ensino similares, neste trabalho foi feita uma pesquisa baseada em *Design Science Research*. A partir dos resultados obtidos, foi desenvolvido um sistema que produz instâncias de bases de dados contendo narrativas fictícias. Focando-se nas deficiências de conhecimento relacionadas a consultas em banco de dados, Enola faz uso de um metamodelo para a criação de narrativas customizáveis. Essas promovem o reuso da ferramenta em sala de aula por parte dos professores e aumentam a motivação de alunos na prática de SQL.

Como etapa de avaliação, jogos gerados foram aplicados a uma turma de banco de dados para resolução do desafio narrativo. Atingiu-se alto índice de satisfação entre alunos durante as partidas. Tendo em vista os resultados majoritariamente positivos ficou demonstrado o potencial uso para o engajamento de estudantes da área de banco de dados e a possibilidade de sua consolidação entre professores de forma abrangente.

Ficou evidente pelo retorno do aplicador que a ferramenta apresentou limitações do ponto de vista narrativo, uma vez que os enredos gerados se além aos fornecidos pela aplicação ou são manualmente descritos pelo aplicador. De forma similar, o metamodelo não é flexível quanto ao progresso esperado e possibilidades de desafio.

Durante a aplicação em sala perceberam-se *bugs* na parte de produção de tuplas de apoio à narrativa principal e isso alertou para a necessidade de automatizar uma verificação das tuplas geradas para que se garanta a qualidade da solução.

Em relação à arquitetura da aplicação existem algumas limitações no cerne da customização. Hoje essa configuração se dá através de diversos arquivos JSON, o que

traz desgastante para a utilização do aplicador e se torna um impedimento para o melhor aproveitamento da ferramenta. Existe ainda uma forte dependência com duas ferramentas externas, *Faker* e *Knex*. Em relação à primeira, o acoplamento com o projeto vigente se torna um risco para a sua evolução, uma vez que a faixa de operação do conteúdo procedural gerado fica atrelado à capacidade oferecida pelo *Faker*. Em relação ao *Knex*, hoje é impossível gerar um banco de dados ou executar os comandos SQL sem a API fornecida. As opções de SGBDs ofertadas diferentes de *SQLite3* não foram considerados.

Do ponto de vista didático, os critérios de avaliação utilizados não conseguem atribuir à Enola a responsabilidade por um eventual aumento na taxa de aprendizagem dos alunos. Os critérios, contudo, demonstram que há um interesse elevado em utilizar ferramentas de mesma função para engajar e promover a prática em banco de dados. Conceitos como a criação de *triggers*, *stored procedures*, manutenção de índices e o uso de cursores não são atualmente explorados pela ferramenta.

A respeito da progressão das consultas realizadas dentro da ferramenta não é possível garantir que o aluno siga pela estrutura de consultas imaginada como ideal para solucionar um desafio, uma vez que há formas de solução por força bruta sem que o ambiente do jogo impeça esse tipo de prática pois o mesmo se trata de um SGBD. Para viabilizar tais verificações, uma camada de aplicação lógica em torno do banco de dados se faria necessária, fugindo ao objetivo inicial de induzir os alunos a praticarem em um SGBD real sem interfaces intermediárias.

Em trabalhos futuros pode-se explorar o desenvolvimento de uma linguagem específica de domínio para a criação de metamodelos narrativos. Pode-se ainda considerar outro conjunto de critérios avaliativos para incluir métricas de ganho educacional e comparar os resultados obtidos por Enola a exercícios práticos em sala de aula. Dado o requisito de garantir a viabilidade da narrativa enquanto fomentadora do exercício de SQL, pode-se também propor uma ferramenta de verificação da narrativa proposta.

Seria ainda interessante a criação de uma aplicação específica para a geração procedural de narrativas que ao fazer uso de metamodelos possa apoiar mais áreas da computação que também são negligenciadas quando o assunto é a criação de jogos sérios como motivador para o ensino.

Bibliografia

- Bierman, G., Abadi, M., e Torgersen, M. (2014). Understanding typescript. In *European conference on object-oriented programming* (pp. 257–281).
- Carneiro, J. (2020). A review on tools for teaching database systems online. *perspective*, 12(14), 16.
- Castro, R., e Siqueira, S. (2017). Aprendizagem ativa em sistemas de informação: Novas técnicas propostas e reflexões sobre as experiências. In *Anais principais do xiii simpósio brasileiro de sistemas de informação* (pp. 535–542).
- Ceccon, D. L., e Porto, J. B. (2020). Bcs: Jogos digitais no auxílio do desenvolvimento de crianças especiais com atraso na linguagem. In *Anais do xxxi simpósio brasileiro de informática na educação* (pp. 522–531).
- Chang, C.-C., Liang, C., Chou, P.-N., e Lin, G.-Y. (2017). Is game-based learning better in flow experience and various types of cognitive load than non-game-based learning? perspective from multimedia and media richness. *Computers in Human Behavior*, 71, 218–227.
- Christiane Gresse von Wangenheim, A. v. W. (2013). *Ensinando computação com jogos*. Saraiva Educação SA.
- de Castro, R. M., Siqueira, S. W. M., e da Silva Monteiro, S. (2020). Didática da computação na perspectiva da aprendizagem ativa. In *Anais dos workshops do ix congresso brasileiro de informática na educação* (pp. 31–40).
- Hamzah, M. L., Rukun, K., Rizal, F., Purwati, A. A., et al. (2019). A review of increasing teaching and learning database subjects in computer science. *Revista ESPACIOS*, 40(26).
- Huang, X.-p., e Leng, J. (2019). Design of database teaching model based on computational thinking training. *International Journal of Emerging Technologies in Learning (iJET)*, 14(08), 52–69.
- Jabbar, A. I. A., e Felicia, P. (2015). Gameplay engagement and learning in game-based learning: A systematic review. *Review of Educational Research*, 85(4), 740–779. Retrieved from <https://doi.org/10.3102/0034654315577210> doi: 10.3102/0034654315577210
- Joon Park, C. H. (2019). *Sql murder mystery*. Retrieved from <https://mystery.knightlab.com/>
- Josko, J. M. B. (2020). A problem-based approach to teach physical database design: an experience report. In *Anais do xxviii workshop sobre educação em computação* (pp. 11–15).
- Lavbič, D., Matek, T., e Zrnc, A. (2017). Recommender system for learning sql using hints. *Interactive Learning Environments*, 25(8), 1048–1064.
- Lima, T., Barradas Filho, A., Barros, A. K., Viana, D., Junior, J. B. B., e Rivero, L. (2020). Avaliando um jogo educacional para o ensino de inteligência artificial-qual metodologia para avaliação escolher? In *Anais do xxviii workshop sobre educação em computação* (pp. 66–70).
- Lin, C.-H., Huang, S.-H., Shih, J.-L., Covaci, A., e Ghinea, G. (2017). Game-based learning effectiveness and motivation study between competitive and cooperative modes. In *2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT)* (pp. 123–127).

- Loewen, S., Crowther, D., Isbell, D. R., Kim, K. M., Maloney, J., Miller, Z. F., e Rawal, H. (2019). Mobile-assisted language learning: A duolingo case study. *ReCALL: the Journal of EUROCALL*, 31(3), 293–311.
- Martinhago, A., Smarzaró, R., Lima, I., e Guimarães, L. (2014). Computação desplugada no ensino de bancos de dados na educação superior. In *Anais do xxii workshop sobre educação em computação* (pp. 21–29).
- Mohr, H., Eger, M., e Martens, C. (2018). Eliminating the impossible: A procedurally generated murder mystery. In *Aiide workshops*.
- Piza-Dávila, H. I., Gutiérrez-Preciado, L. F., e Ortega-Guzmán, V. H. (2017). An educational software for teaching database normalization. *Computer Applications in Engineering Education*, 25(5), 812–822.
- Prabhu, S., e Jaidka, S. (2019). Sql and pl-sql: Analysing teaching methods. In *Citrenz conference (2019)*.
- Ribeiro, P. B., e Santos, E. F. (2016). Ludosql: Jogo de apoio ao aprendizado de comandos básicos do sql. *Caderno de Estudos Tecnológicos*, 4(1).
- Soflano, M. (2011). Modding in serious games: Teaching structured query language (sql) using neverwinter nights. In *Serious games and edutainment applications* (pp. 347–368). Springer.
- Taipalus, T., e Perälä, P. (2019). What to expect and what to focus on in sql query teaching. In *Proceedings of the 50th acm technical symposium on computer science education* (pp. 198–203).
- Ward, P. T. (2015). *Teaching of sql through a game* (Unpublished doctoral dissertation).

A Apêndice

A.1 Formulário para levantamento de requisitos com professores de banco de dados

25/08/2021

Sobre o ensino de banco de dados

Sobre o ensino de banco de dados

*Obrigatório

1. Quais métodos você faz uso em disciplinas de bancos de dados? *

Marque todas que se aplicam.

- Aulas expositivas
- Aulas práticas
- Uso de ferramenta externa de apoio ao ensino (e.g. Repl.it)
- Abordagem de ensino baseada em projeto
- Abordagem de ensino baseada em jogos

2. Quais as dificuldades mais comuns na assimilação do conteúdo entre alunos de uma disciplina de banco de dados? *

3. Quais conceitos de banco de dados considera mais difíceis de ensinar dentro de um ambiente de sala de aula, presencial ou virtual? *

Sobre a dificuldade de conceitos de bancos de dados

25/08/2021

Sobre o ensino de banco de dados

4. Considerando o ensino dos seguintes conceitos de SQL dentro da disciplina de banco de dados, classifique entre 0 e 5 onde 0 representa nenhum entendimento dos alunos e 5 representa pleno entendimento dos alunos *

Marcar apenas uma oval por linha.

	0	1	2	3	4	5
Tabelas	<input type="radio"/>					
Linhas (Registros)	<input type="radio"/>					
Colunas (Atributos)	<input type="radio"/>					
Chave Primária	<input type="radio"/>					
Chave Estrangeira	<input type="radio"/>					
Expressões (booleanas, numéricas, data)	<input type="radio"/>					
Ordenação	<input type="radio"/>					
Expressões Com Aninhamento	<input type="radio"/>					
Existe/Não Existe	<input type="radio"/>					
Subconsultas	<input type="radio"/>					
Agregação (e.g. Soma, Máximo)	<input type="radio"/>					
Auto-Associação (Self-Join)	<input type="radio"/>					
Parâmetro Distinto (Distinct)	<input type="radio"/>					
Agrupamento (Group by)	<input type="radio"/>					

25/08/2021

Sobre o ensino de banco de dados

5. Considerando consultas SQL com as características abaixo, classifique entre 0 e 5, 0 representando que o cenário especificado é de fácil assimilação por parte dos alunos e 5 representando que o cenário especificado é de difícil assimilação por parte dos alunos: *

Marcar apenas uma oval por linha.

	0	1	2	3	4	5
Somente uma tabela; expressões;	<input type="radio"/>					
Somente uma tabela; expressões; ordenação;	<input type="radio"/>					
Somente uma tabela; asteriscos; expressões com aninhamento;	<input type="radio"/>					
Múltiplas tabelas; expressões; lidar com chave estrangeira;	<input type="radio"/>					
Múltiplas tabelas; expressões com aninhamento; ordenação;	<input type="radio"/>					
Múltiplas tabelas; expressões; não existe;	<input type="radio"/>					
Múltiplas tabelas; expressões; não existe; subconsultas do tipo Igual;	<input type="radio"/>					
Somente uma tabela; expressões; funções de agregação;	<input type="radio"/>					
Múltiplas tabelas; expressões; múltiplas tabelas de origem;	<input type="radio"/>					
Múltiplas tabelas; expressões; auto-associação (self-join);	<input type="radio"/>					
Múltiplas tabelas; expressões; auto-associação (self-join); função de agregação avaliada contra um valor comum; subconsulta correlacionada;	<input type="radio"/>					

25/08/2021

Sobre o ensino de banco de dados

Múltiplas tabelas; expressões;
função de agregação avaliada
contra uma constante;
subconsulta não
correlacionada; parâmetro
distinto;

Múltiplas tabelas; múltiplas
tabelas de origem; funções de
agregação; agrupamento;

Múltiplas tabelas; múltiplas
tabelas de origem; funções de
agregação; agrupamento;
restrições de agrupamento;
ordenação;

Sobre abordagens de ensino baseadas em Jogos

6. Você faz uso ou já utilizou abordagens de ensino baseadas em jogos em alguma disciplina? *

Marcar apenas uma oval.

- Sim, faço uso atualmente e pretendo continuar
- Sim, porém não utilizo mais
- Não encontrei abordagens de ensino baseadas em jogos
- Não utilizei, porém tenho interesse em utilizar
- Não pretendo fazer uso de abordagens de ensino baseadas em jogos

25/08/2021

Sobre o ensino de banco de dados

7. Você faz uso ou já utilizou abordagens de ensino baseadas em jogos para lecionar Banco de Dados? *

Marcar apenas uma oval.

- Sim, faço uso atualmente e pretendo continuar
- Sim, porém não utilizo mais
- Não encontrei abordagens de ensino baseadas em jogos para banco de dados
- Não utilizei, porém tenho interesse em utilizar
- Não pretendo fazer uso de abordagens de ensino baseadas em jogos em disciplinas de banco de dados

8. Cogita o uso de abordagens autocorretivas como uma ferramenta de auxílio ao professor e ao aluno? *

Marcar apenas uma oval.

- Sim, cogito, abordagens autocorretivas auxiliam tanto o professor quanto o aluno
- Sim, cogito, porém acredito que auxilia em maior parte o professor
- Sim, cogito, porém acredito que auxilia em maior parte o aluno
- Não cogito, pois ainda não conheço os benefícios do uso
- Não cogito, pois não acredito no uso de abordagens autocorretivas

9. O que você vê como características importantes de um jogo para ensinar consultas SQL? Responda tendo em mente um jogo para apoiar no ensino de SQL ao longo de uma disciplina de banco de dados. *

A.2 Formulário para avaliação de Enola junto a alunos utilizadores após aplicação

25/08/2021

Avaliação da ferramenta

Avaliação da ferramenta

Avaliação a ser realizada por alunos após aplicação em turma de banco de dados.

* Required

1. Acredita ter consolidado os conhecimentos em SQL após uso da ferramenta? *

Mark only one oval.

	1	2	3	4	5	
Nada consolidado	<input type="radio"/>	Muito consolidado				

2. Você sente que a ferramenta é de fácil utilização? *

Mark only one oval.

	1	2	3	4	5	
Fácil utilização	<input type="radio"/>	Difícil utilização				

3. Como você vê o grau de diversão/motivação para utilizar a ferramenta? *

Mark only one oval.

	1	2	3	4	5	
Pouco divertido/Desmotivador	<input type="radio"/>	Muito divertido/Motivador				

25/08/2021

Avaliação da ferramenta

4. Sente que a experiência de jogar o jogo em si é satisfatória? *

Mark only one oval.

	1	2	3	4	5	
Pouco satisfatória	<input type="radio"/>	Muito satisfatória				

5. O jogo oferece novos obstáculos, situações e variações? *

Mark only one oval.

	1	2	3	4	5	
Mesmos desafios	<input type="radio"/>	Muitos desafios novos				

6. Considera a experiência do jogo imersiva? *

Mark only one oval.

	1	2	3	4	5	
Me senti desconectado do jogo	<input type="radio"/>	Me senti totalmente imerso no jogo				

7. O jogo é relevante para os interesses de aprendizagem de alunos como você? *

Mark only one oval.

	1	2	3	4	5	
Pouco relevante	<input type="radio"/>	Muito relevante				

25/08/2021

Avaliação da ferramenta

8. As regras do jogo e objetivos foram claros? *

Mark only one oval.

	1	2	3	4	5	
Pouca clareza	<input type="radio"/>	Muita clareza				

9. Você sente que o jogo lhe fornece os feedbacks necessários sobre o seu desempenho e progresso em direção aos objetivos? *

Mark only one oval.

	1	2	3	4	5	
Pouco feedback	<input type="radio"/>	Muito feedback				

10. Considera que o jogo promove interação entre jogadores, colaborativa ou competitivamente? *

Mark only one oval.

	1	2	3	4	5	
Pouca interação social	<input type="radio"/>	Muita interação social				

11. Se sentiu confiante durante o jogo em relação aos conteúdos abordados e seu exercício? *

Mark only one oval.

	1	2	3	4	5	
Pouco confiante	<input type="radio"/>	Muito confiante				

25/08/2021

Avaliação da ferramenta

12. Sentiu-se desfocado ou sobrecarregado pelos desafios propostos pelo jogo? *

Mark only one oval.

	1	2	3	4	5	
Pouco concentrado	<input type="radio"/>	Muito concentrado				

This content is neither created nor endorsed by Google.

Google Forms

A.3 Arquivo de configuração Context

```
1 {
2   "profile": {
3     "name": "driver_license",
4     "columns": [
5       "age",
6       "height",
7       "eye_color",
8       "hair_color",
9       "gender",
10      "plate_number",
11      "car_maker",
12      "car_model"
13    ]
14  },
15  "character": {
16    "name": "person",
17    "columns": [
18      "name",
19      "address_number",
20      "address_street_name",
21      "ssn"
22    ]
23  },
24  "institutionMembership": {
25    "name": "get_fit_now_member",
26    "columns": [
27      "membership_start_date",
28      "membership_status"
29    ]
30  },
31  "eventLog": {
32    "name": "facebook_event_checkin",
33    "columns": [
34      "event_name",
35      "date"
36    ]
37  },
38  "dialog": {
39    "name": "interview",
40    "columns": [
41      "transcript"
42    ]
43  },
44  "institutionCheckin": {
45    "name": "convention_cave_check_in",
46    "columns": [
47      "check_in_date",
48      "check_in_time",
49      "check_out_time"
50    ]
51  },
52  "scenario": {
53    "name": "crime_scene_report",
54    "columns": [
55      "date",
56      "type",
57      "description",
58      "city"
59    ]
60  },

```

```
61     "ranking": {
62         "name": "income",
63         "columns": [
64             "ssn",
65             "annual_income"
66         ]
67     },
68     "solution": {
69         "name": "solution",
70         "columns": [
71             "user",
72             "value"
73         ]
74     },
75     "generation": 500,
76     "instution": "Get Fit Now Gym"
77 }
```

A.4 Arquivo de configuração *Quotes*

```

1  [
2    {
3      "name": "description",
4      "base_quote": "A crime has taken place and the detective needs your help. The detective gave
        you the crime scene report, but you somehow lost it. You vaguely remember that the crime
        was {CRIME_ARTICLE} {CRIME_TYPE} that occurred sometime on {REF_DATE} and that it took
        place in {CITY} City. Start by retrieving the corresponding crime scene report from the
        police department's database.",
5      "entities": [
6        {
7          "name": "CRIME_TYPE",
8          "switcher": "CRIME.TYPE",
9          "context": "scenario.type"
10       },
11       {
12         "name": "CRIME_ARTICLE",
13         "switcher": "CRIME.ARTICLE"
14       },
15       {
16         "name": "REF_DATE",
17         "type": "referenceDate",
18         "context": "scenario.date"
19       },
20       {
21         "name": "CITY",
22         "type": "city",
23         "context": "scenario.city"
24       }
25     ]
26   },
27   {
28     "name": "scenario",
29     "base_quote": "Security footage shows that there were 2 witnesses. The first witness lives at
        the {POSITION_1} house on {STREET_NAME_1}. The other named {NAME_2}, lives somewhere on {
        STREET_NAME_2}.",
30     "entities": [
31       {
32         "name": "POSITION_1",
33         "options": [
34           "first",
35           "last"
36         ],
37         "context": "character.address_number"
38       },
39       {
40         "name": "STREET_NAME_1",
41         "type": "streetname",
42         "context": "character.address_street_name"
43       },
44       {
45         "name": "NAME_2",
46         "type": "name",
47         "context": "character.name"
48       },
49       {
50         "name": "STREET_NAME_2",
51         "type": "streetname",
52         "context": "character.address_street_name"
53       }
54     ]

```

```

55     },
56     {
57         "name": "source_1_dialog",
58         "base_quote": "I saw the {CRIME_TYPE} happen, and I recognized the {CRIME_COMMITER} from {
                    CLUB_TYPE} when I was {CLUB_ACTIVITY} on {CHECK_IN_DATE}." ,
59         "entities": [
60             {
61                 "name": "CRIME_TYPE",
62                 "switcher": "CRIME.TYPE"
63             },
64             {
65                 "name": "CRIME_COMMITER",
66                 "switcher": "CRIME.COMMITER"
67             },
68             {
69                 "name": "CLUB_TYPE",
70                 "switcher": "CLUB.TYPE"
71             },
72             {
73                 "name": "CLUB_ACTIVITY",
74                 "switcher": "CLUB.ACTIVITY"
75             },
76             {
77                 "name": "CHECK_IN_DATE",
78                 "type": "pastDate",
79                 "context": "institution_checkin.check_in_date"
80             }
81         ]
82     },
83     {
84         "name": "source_2_dialog",
85         "base_quote": "I {CRIME_NOTICE} and then saw a {GENDER_TYPE} run out. {GENDER_SUBJECTCAPS} had
                    a {CLUB_TYPE} bag. The membership number on the bag {MEMBERSHIP_PARTIAL_REF} '{
                    MEMBERSHIP_PARTIAL_VAL}'. Only {CLUB_STATUS} members have those bags. The {GENDER_TYPE}
                    got into a car with a plate that included '{LICENSE_PARTIAL_VAL}'." ,
86         "entities": [
87             {
88                 "name": "CRIME_NOTICE",
89                 "switcher": "CRIME.NOTICE"
90             },
91             {
92                 "name": "CLUB_TYPE",
93                 "switcher": "CLUB.TYPE"
94             },
95             {
96                 "name": "GENDER_TYPE",
97                 "switcher": "GENDER.TYPE",
98                 "context": "profile.gender"
99             },
100            {
101                "name": "GENDER_TYPE",
102                "switcher": "GENDER.TYPE"
103            },
104            {
105                "name": "GENDER_SUBJECTCAPS",
106                "switcher": "GENDER.SUBJECTCAPS"
107            },
108            {
109                "name": "CLUB_STATUS",
110                "options": [
111                    "silver",
112                    "gold",

```

```

113         "platinum"
114     ],
115     "context": "clubMembership.membership_status"
116 },
117 {
118     "name": "MEMBERSHIP.PARTIAL.REF",
119     "partial": "MEMBERSHIP.REFERENCE"
120 },
121 {
122     "name": "MEMBERSHIP.PARTIAL.VAL",
123     "partial": "MEMBERSHIP.VALUE",
124     "context": "clubMembership.id"
125 },
126 {
127     "name": "LICENSE.PARTIAL.VAL",
128     "partial": "profile.VALUE",
129     "context": "profile.plate_number"
130 }
131 ]
132 },
133 {
134     "name": "target_1_dialog",
135     "base_quote": "I was hired by a {GENDER.TYPE} with {INCOME}. I don't know {GENDER.POSSESSIVE}
name but I know {GENDER.SUBJECT}'s around {HEIGHT.LOW} or {HEIGHT.HIGH}. {
GENDER.SUBJECTCAPS} has {HAIR.COLOR} hair, has {EYE.COLOR} eyes and she drives a {CAR}. I
know that {GENDER.SUBJECT} attended the {EVENT} 3 times in {YEAR}.",
136     "entities": [
137         {
138             "name": "GENDER.TYPE",
139             "switcher": "GENDER.TYPE",
140             "context": "profile.gender"
141         },
142         {
143             "name": "GENDER.SUBJECT",
144             "switcher": "GENDER.SUBJECT"
145         },
146         {
147             "name": "GENDER.SUBJECTCAPS",
148             "switcher": "GENDER.SUBJECTCAPS"
149         },
150         {
151             "name": "GENDER.SUBJECT",
152             "switcher": "GENDER.SUBJECT"
153         },
154         {
155             "name": "GENDER.POSSESSIVE",
156             "switcher": "GENDER.POSSESSIVE"
157         },
158         {
159             "name": "INCOME",
160             "options": [
161                 "lots of money",
162                 "plenty of bills to pay"
163             ],
164             "context": "ranking.amount"
165         },
166         {
167             "name": "YEAR",
168             "type": "year",
169             "context": "eventLog.event_dates"
170         },
171         {

```

```

172         "name": "HAIR_COLOR",
173         "options": [
174             "red",
175             "black",
176             "brown",
177             "blonde",
178             "white"
179         ],
180         "context": "profile.hair_color"
181     },
182     {
183         "name": "EYE_COLOR",
184         "options": [
185             "amber",
186             "blue",
187             "brown",
188             "gray",
189             "green",
190             "hazel"
191         ],
192         "context": "profile.eye_color"
193     },
194     {
195         "name": "EVENT",
196         "type": "event",
197         "context": "eventLog.event_name"
198     },
199     {
200         "name": "LICENSE",
201         "type": "unique_identifier",
202         "context": "profile.unique_identifier"
203     },
204     {
205         "name": "HEIGHT_HIGH",
206         "type": "high_height",
207         "context": "profile.high_height"
208     },
209     {
210         "name": "HEIGHT_LOW",
211         "type": "low_height",
212         "context": "profile.low_height"
213     }
214 ]
215 },
216 {
217     "name": "target_one_found",
218     "base_quote": "You run across the dirty road at the enter of the cave and see a fairy that fits
the description. When you touched her shoulder, how surprised you were! You didn't found
the {FAIRY_TYPE}, but wait, not everything is lost. Talk to the {FAIRY_FRIEND} to find the
remarkable fairy the king is so eager to know. If you feel especially confident in your
SQL skills, try to complete this final step with no more than 2 queries. Use this same
INSERT statement with the name of the fairy to check your answer.",
219     "entities": [
220         {
221             "name": "FAIRY_TYPE",
222             "switcher": "FAIRY.TYPE"
223         },
224         {
225             "name": "FAIRY_FRIEND",
226             "switcher": "FAIRY.FRIEND"
227         }
228     ]

```

```
229     },
230     {
231         "name": "target_two_found",
232         "base_quote": "Congrats, you found the brains behind the {CRIME_TYPE}! Everyone in SQL City
                hails you as the greatest SQL detective of all time. Time to break out the champagne!",
233         "entities": [
234             {
235                 "name": "CRIME_TYPE",
236                 "switcher": "CRIME_TYPE"
237             }
238         ]
239     },
240     {
241         "name": "not_found",
242         "base_quote": "That's not the right person. Try again!",
243         "entities": []
244     }
245 ]
```

A.5 Arquivo de configuração *Switchers*

```
1 [
2   {
3     "name": "CRIME",
4     "cases": [
5       {
6         "TYPE": "murder",
7         "ARTICLE": "a",
8         "COMMITTER": "murderer",
9         "NOTICE": "heard a gunshot"
10      },
11      {
12        "TYPE": "robbery",
13        "ARTICLE": "a",
14        "COMMITTER": "robber",
15        "NOTICE": "heard a person screaming loudly"
16      },
17      {
18        "TYPE": "arson",
19        "ARTICLE": "an",
20        "COMMITTER": "arsonist",
21        "NOTICE": "saw a dense smoke coming out of the window"
22      },
23      {
24        "TYPE": "burglary",
25        "ARTICLE": "a",
26        "COMMITTER": "burglar",
27        "NOTICE": "heard the alarms go off"
28      },
29      {
30        "TYPE": "theft",
31        "ARTICLE": "a",
32        "COMMITTER": "thief",
33        "NOTICE": "saw someone climbing a ladder"
34      }
35    ]
36  },
37  {
38    "name": "CLUB",
39    "cases": [
40      {
41        "TYPE": "gym",
42        "ACTIVITY": "working out"
43      }
44    ]
45  },
46  {
47    "name": "GENDER",
48    "cases": [
49      {
50        "TYPE": "woman",
51        "POSSESSIVE": "her",
52        "SUBJECT": "she",
53        "SUBJECTCAPS": "She"
54      },
55      {
56        "TYPE": "man",
57        "POSSESSIVE": "his",
58        "SUBJECT": "he",
59        "SUBJECTCAPS": "He"
60      }
61    ]
62  }
63 ]
```

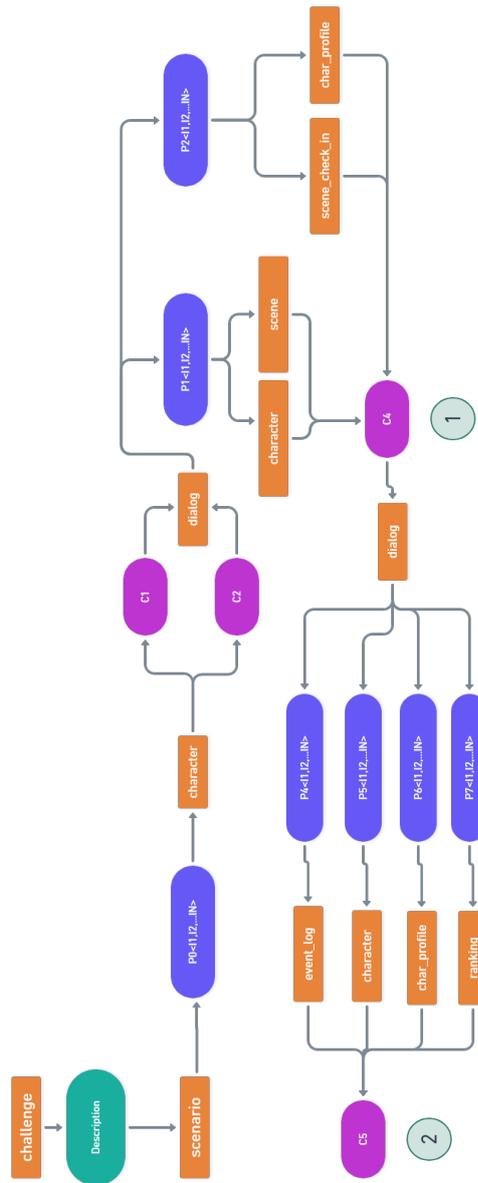
```
61     ]  
62   }  
63 ]
```

A.6 Arquivo de configuração Templating

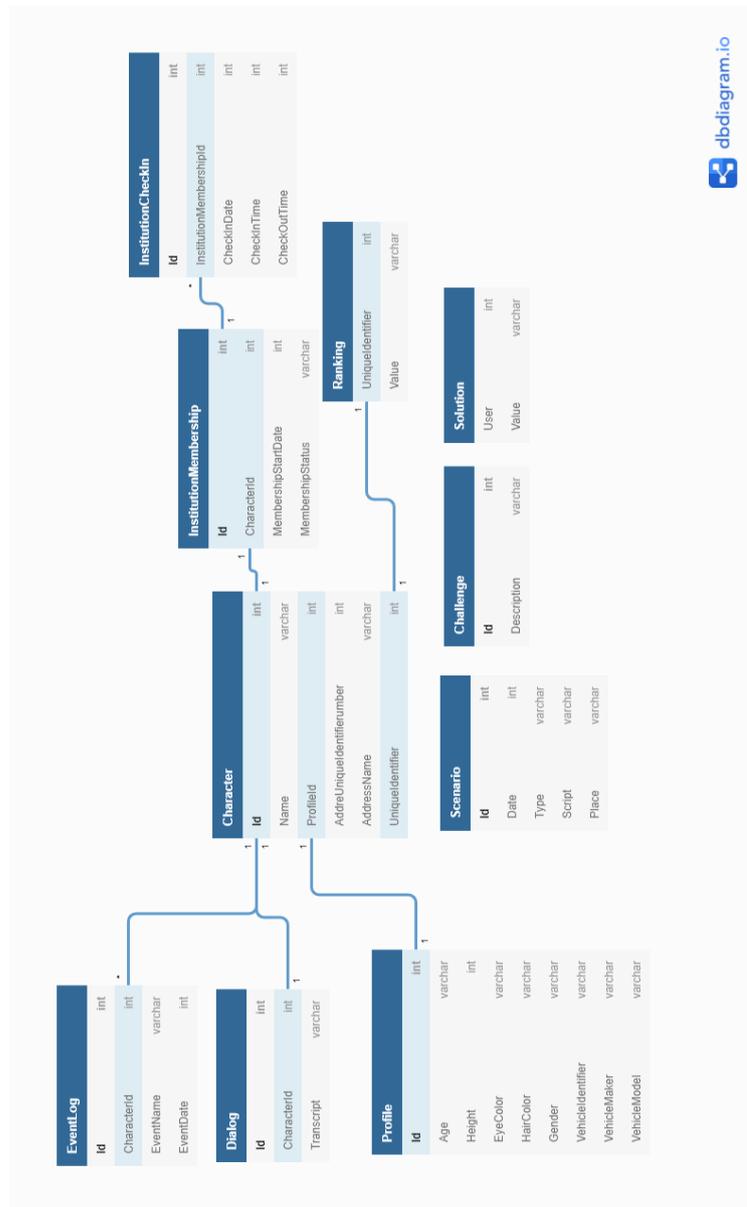
```
1 [
2   {
3     "name": "generateElement",
4     "options": [
5       "streetname",
6       "name",
7       "pastDate",
8       "referenceDate",
9       "companyName",
10      "place",
11      "city",
12      "event",
13      "year",
14      "car",
15      "low_height",
16      "high_height",
17      "default"
18    ]
19  },
20  {
21    "name": "generateGender",
22    "options": ["woman", "man"]
23  },
24  {
25    "name": "generateEventName",
26    "options": [
27      "Concert",
28      "Show",
29      "Tour",
30      "Spectacle",
31      "Rave",
32      "Experience",
33      "Live"
34    ]
35  },
36  {
37    "name": "generateRanking",
38    "options": ["plenty of bills to pay", "lots of money"]
39  },
40  {
41    "name": "generateRanking",
42    "options": [
43      [
44        "amber",
45        "blue",
46        "brown",
47        "gray",
48        "green",
49        "hazel"
50      ],
51      ["red", "black", "brown", "blonde", "white"]
52    ]
53  },
54  {
55    "name": "generateMembership",
56    "options": [
57      [
58        "active",
59        "inactive",
60        "gold",
```

```
61         "platinum"  
62     ]  
63 ]  
64 }  
65 ]
```

A.7 Metamodelo narrativo



A.8 Esquema do Metamodelo narrativo



A.9 Resolução do enredo de *SQL Murder Mystery*

— *A crime has taken place and the detective needs your help.*
 — *The detective gave you the crime scene report,*
 — *but you somehow lost it. You vaguely remember that the crime*
 — *was a murder that occurred sometime on Jan.15, 2018 and that*
 — *it took place in SQL City. Start by retrieving the corresponding*
 — *crime scene report from the police departments database.*

```
select * from crime_scene_report
where date = 20180115
and type = 'murder'
and city = 'SQL_City'
```

— *"Security footage shows that there were 2 witnesses.*
 — *The first witness lives at the last house on "Northwestern Dr".*
 — *The second witness, named Annabel, lives somewhere on*
 — *"Franklin Ave".*

```
select * from person
where address_street_name = 'Northwestern_Dr'
order by address_number desc
```

— *"Northwestern Dr" -> [14887] Morty Schapiro, 11156494, 118009*

```
select * from person
where address_street_name = 'Franklin_Ave'
and name like 'Annabel%'
```

— *"Franklin Ave" + "Annabel" -> [16371] Annabel Miller, 318771143, 490173*

```
select transcript
from interview
where person_id in (14887, 16371)
```

— *Morty Schapiro interview:*
 — *I heard a gunshot and then saw a man run out.*
 — *He had a "Get Fit Now Gym" bag. The membership*
 — *number on the bag started with "48Z". Only gold*
 — *members have those bags. The man got into a car with a*
 — *plate that included "H42W".*

— *Annabel Miller*
 — *I saw the murder happen, and I recognized the killer*
 — *from my gym when I was working out last week on January the 9th.*

```
select check_in_date, check_in_time, check_out_time from get_fit_now_check_in
where membership_id in (select id
from get_fit_now_member
where person_id = 16371)
```

— *20180109: 1600-1700*

```
select membership_id, check_in_time, check_out_time
from get_fit_now_check_in
where check_in_date = 20180109
and check_out_time >= 1600
and membership_id != 90081
```

— *48Z7A 1600 1730*
 — *48Z55 1530 1700*

```
select person_id, name, membership_status
from get_fit_now_member
where id in ('48Z7A', '48Z55')
```

```
— 67318      Jeremy Bowers    gold
— 28819      Joe Germuska    gold
```

```
select
  p.id as PersonId,
  p.name as Name,
  d.id as DriversLicenseId,
  d.plate_number as Plate
from drivers_license d
join person p
  on d.id = p.license_id
where p.id in (67318, 28819)
```

```
— 67318      Jeremy Bowers    423327 0H42W2
```

```
INSERT INTO solution VALUES (1, 'Jeremy_Bowers');
SELECT value as 'Did_you_win?' FROM solution;
```

```
— Congrats, you found the murderer! But wait, there's more...
— If you think you're up for a challenge, try querying the interview
— transcript of the murderer to find the real villain behind this crime.
— If you feel especially confident in your SQL skills, try to complete this
— final step with no more than 2 queries. Use this same INSERT statement with
— your new suspect to check your answer.
```

```
select transcript
from interview
where person_id = 67318
```

```
— I was hired by a woman with a lot of money.
— I don't know her name but I know she's around 5'5" (65")
— or 5'7" (67"). She has red hair and she drives a Tesla Model S.
— I know that she attended the SQL Symphony Concert 3 times in
— December 2017.
```

```
select distinct p.name
from facebook_event_checkin f
join person p
  on f.person_id = p.id
where event_name = 'SQL_Symphony_Concert'
and date > 20170000
and date < 20180000
and person_id in (select p.id
                  from income a
                  join person p
                    on a.ssn = p.ssn
                  join drivers_license d
                    on p.license_id = d.id
                  where d.gender = 'female'
                  and d.hair_color = 'red'
                  and d.height >= 65
                  and d.height <= 67
                  and d.car_model = 'Model_S')
```

```
— Miranda Priestly
```

```
INSERT INTO solution VALUES (1, 'Miranda_Priestly');  
SELECT value as 'Did_you_win?' FROM solution;
```

```
— Congrats, you found the brains behind the murder!  
— Everyone in SQL City hails you as the greatest  
— SQL detective of all time. Time to break out the champagne!
```