

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE SÃO PAULO

EDITOR DE ITENS INEP

Estudante-pesquisador: Gabriel de Moraes Garcia

Orientador: Italo S. Vega

RELATÓRIO PARCIAL DE INICIAÇÃO CIENTÍFICA

SÃO PAULO

MARÇO DE 2018

Relatório Intermediário

Sumário

1. Relatório de Atividades	1
1.1. Diário Resumido	1
1.2. Resultados e dificuldades encontradas	1
1.3. Cronograma e progresso	1
1.4. Revisão do Cronograma	2
2. Relatório Científico	3
2.1. Introdução	3
2.2. Itens do INEP	3
2.3. Matrizes de Referência	4
2.4. Questionário MCQ	4
2.5. Requisitos Funcionais do Editor MCQ	4
2.6. Modelo de Domínio	5
2.7. Arquitetura do Sistema	6
3. Modelo de Implementação	8
3.1. Protótipos de Interface	8
4. Bibliografia	10

1. Relatório de Atividades

1.1. Diário Resumido

O projeto de pesquisa começou em agosto de 2017, com encontros mensais com o professor orientador. Além de diretivas sobre como realizar um trabalho de pesquisa, aspectos técnicos de Engenharia de software também eram discutidos a fim de traçar o rumo do desenvolvimento do trabalho.

1.2. Resultados e dificuldades encontradas

O começo da pesquisa se mostrou um pouco desafiadora, pois foi descoberto que o principal documento de referência para a elaboração do modelo de domínio, o Guia de Elaboração de Itens do Inep, não era mais disponibilizado pelo instituto para consulta pública. Dessa forma, foi necessária a busca de outras fontes bibliográficas sobre o tema, o que tornou o processo de formação de intuição de domínio mais complicada.

Pode-se considerar que o objetivo de aprimoração de habilidades e capacidades de modelagem de aplicações utilizando técnicas e princípios de engenharia de software foi alcançado pois foi botado em prática, no trabalho sendo elaborado, os conceitos abordados em matérias da graduação de Ciência da Computação, como “Engenharia de Software-Modelagem” e “Engenharia de Software-Tecnologia”, assim como assuntos sobre arquitetura de sistemas, banco de dados e programação abordados em outras matérias. Em adição, fontes bibliográficas adicionais foram fundamentais a fim de aprimorar a modelização, como o livro *Patterns of Enterprise Application Architecture*, utilizado como base para a criação do modelo arquitetural do projeto. Desta forma, os conhecimentos prévios combinados com novas fontes bibliográficas pesquisadas, permitiram o desenvolvimento do modelo de domínio, arquitetural e de implementação deste projeto. Entretanto continua sendo um processo de aprendizado em progresso, ainda não finalizado.

1.3. Cronograma e progresso

Tabela 1. Estado das tarefas.

Não Iniciado	(N)
Em Andamento	(A)
Concluído	(C)

Plano de Trabalho

1. (A1) Princípios de engenharia de software --- Estudo e exploração tecnológica
 - a. Modelos de domínio (C)
 - b. Modelos de aplicações orientados a objetos (C)
2. (A2) Questionários de Múltipla Escolha --- MCQ
 - a. Estudo do Guia de Elaboração e Revisão de Itens do INEP (C)
 - b. Modelo de domínio MCQ (C)

3. (A3) Editor de Questionários
 - a. Requisitos do editor MCQ (C)
 - b. Modelo arquitetural (C)
 - c. Modelo de implementação (A)
 - d. Código-fonte (N)
 - e. Verificação e validação (N)
4. (A4) Registro do trabalho de pesquisa

Figura 1. Cronograma revisado.

Atividades	Meses											
	2017					2018						
	AGO	SET	OUT	NOV	DEZ	JAN	FEV	MAR	ABR	MAI	JUN	JUL
A1-a	X											
A1-b	X											
A2-a		X	X									
A2-b			X	X	X							
A3-a					X							
A3-b					X	X				X	X	X
A3-c						X				X	X	X
A3-d						X	X	X		X	X	X
A3-e							X	X	X	X	X	X
A4			X	X	X			X	X	X	X	X

Fonte: elaborado pelo autor

1.4. Revisão do Cronograma

Devido às dificuldades em encontrar fontes confiáveis alternativas ao Guia de Elaboração do INEP, as atividades A2-a e A2-b tiveram o tempo de execução estendidos impactando no início das tarefas seguintes. A ajuda do orientador para guiar os orientandos na busca fontes alternativas não só foi fundamental para a completude da atividade, mas se mostrou também uma lição valiosa quanto ao processo de desenvolvimento de uma pesquisa acadêmica. A atividade A3-c não foi terminada e a A3-d, que supostamente teria seu início em Janeiro de 2018, ainda não foi começada. Portanto, prevê-se que a execução da atividade A3-d com tempo reduzido poderá implicar apenas em uma solução parcial, não garantindo assim a conclusão total da atividade inicialmente proposta. Mediante a situação descrita acima, o cronograma deverá ser revisto. Originalmente previu-se a implementação das seguintes macro-funções da aplicação: Editor de matrizes, Editor de Itens e Gerador de Questionários. Propõe-se portanto a redução desse escopo para o Editor de Itens e o Gerador de Questionários, mantendo-se o tempo da atividade A4, conforme o cronograma abaixo:

Figura 2. Cronograma proposto.

Atividades	2018				
	MAR	ABR	MAI	JUN	JUL
A1-a					
A1-b					
A2-a					
A2-b					
A3-a					
A3-b			X	X	X
A3-c	X		X	X	X
A3-d	X	X	X	X	X
A3-e	X	X	X	X	X
A4	X	X	X	X	X

Fonte: elaborado pelo autor

2. Relatório Científico

2.1. Introdução

O principal objetivo do trabalho é modelar e desenvolver uma ferramenta para a edição de Itens do INEP, para a elaboração de questionários múltipla escolha, utilizando como referência de formato e normas o Guia de Elaboração e Revisão de Itens do INEP. Um Item consiste na unidade básica de um instrumento de coleta de dados, que pode ser uma prova, um questionário etc (Brasil. Inep, 2006), e é a peça de estudo principal do projeto sendo desenvolvido, com foco nos Itens elaborados em questionários de múltipla escolha. Desta forma, deseja-se como resultado final desta pesquisa, modelar e desenvolver uma ferramenta que ofereça, aos autores de Itens, suporte à concepção de questionários com itens de múltipla escolha.

2.2. Itens do INEP

Os Itens do INEP são compostos por três partes principais: texto-base, enunciado e alternativas, as quais se dividem em gabarito, distratores e justificativa (Brasil. Inep, 2010). O texto-base é uma unidade comunicativa que auxilia na contextualização do problema chave da questão, podendo apresentar, além de textos, imagens, gráficos e tabelas. O enunciado do item descreve o problema a ser resolvido na questão, de forma objetiva e clara. As alternativas são divididas entre o gabarito, com a resposta correta que o aluno deverá assinalar, e os distratores, que são as alternativas incorretas.

2.3. Matrizes de Referência

O guia de elaboração de itens indica que o primeiro passo para se elaborar um item é o conhecimento da Matriz de Referência como um todo, avaliando todas as habilidades descritas para se conhecer o que os Itens que serão elaborados devem avaliar. O segundo passo é escolher uma habilidade descrita em uma Matriz de Referência, que será utilizada para a elaboração do item. Para cada prova elaborada, o INEP fornece um conjunto de matrizes que irão compor as habilidades sendo avaliadas na prova. O elaborador portanto precisa que a consulta às matrizes seja fácil e objetiva, pois as consultará com recorrência.

Ao se elaborar um Item, deve-se escolher uma habilidade descrita em uma Matriz de Referência. Para cada prova elaborada, o INEP fornece um conjunto de matrizes que irão compor as habilidades sendo avaliadas na prova. O elaborador precisa que a consulta às matrizes seja fácil e objetiva, no momento do cadastro do item, portanto. O termo matriz de referência é utilizado especificamente no contexto das avaliações em larga escala para indicar habilidades a serem avaliadas (Brasil, Inep, 2010). A Matriz de Referência é composta por dois eixos: das competências relativas a uma área de conhecimento e das capacidades cognitivas de quem está sendo avaliado. Ao se combinar esses dois eixos, se obtém uma matriz de habilidades do estudante.

2.4. Questionário MCQ

Um questionário do tipo *Multiple Choice Questions* (MCQ) é uma forma de avaliação de estudantes no qual as questões possuem múltiplas respostas predefinidas, e o aluno sendo avaliado deve escolher a alternativa que melhor responde o que foi perguntado no enunciado.

Para avaliações que serão aplicadas para um grande número de estudantes, o modelo de questionário com múltiplas escolhas, pode ser uma alternativa recomendada pois torna possível a automatização da correção das avaliações, diminuindo consideravelmente o esforço de elaboração e correção. Inúmeros softwares, como o *Eyegrade*, conseguem realizar a correção de forma simples e rápida, fazendo com que o processo se torne bem mais curto.

Tal modelo é amplamente utilizado em provas de avaliação de Educação Básica, como a Provinha Brasil (INEP), para ingresso à Universidades como o ENEM (INEP) e para avaliação de Educação Superior como o ENADE (INEP).

2.5. Requisitos Funcionais do Editor MCQ

Os Requisitos Funcionais são as funções que o sistema deve realizar, captura a relação entre o sistema e seu ambiente (ROMAN, 1984). Para realizar a análise dos requisitos funcionais desse projeto, a leitura dos guias de elaboração do INEP foi a principal fonte de informação, a ponto que tais guias fornecem ao elaborador o método que deve ser seguido para a construção de um item, bem como sua estrutura. Tal método descrito, dá origem ao requisitos funcionais do Editor MCQ, pois a aplicação deverá cumprir com as restrições descritas nos guias, para que as funções que realiza estejam de acordo com que é previsto pelo INEP na construção de um Item.

A aplicação deverá fornecer um meio para cadastro de Matrizes de Referência pelo elaborador, possibilitando o cadastro das competências, capacidades cognitivas e as habilidades. O elaborador deve poder cadastrar uma nova matriz e editar matrizes previamente cadastradas. No momento de

salvar, a aplicação deverá realizar validações quanto o preenchimento de todas as competências e capacidades cognitivas, avaliando se há pelo menos uma habilidade atrelada a cada um deles, fornecendo ao elaborador a oportunidade de revisar as edições realizadas ao salvar a matriz.

Quanto ao cadastro de itens, a aplicação deve permitir o cadastro de cada parte componente do item: texto-base, enunciado, gabarito e distratores. Após o cadastro, o elaborador deverá poder editar Itens previamente cadastrados, para possíveis ajustes e melhoras. O autor primeiramente deverá escolher uma habilidade em uma Matriz de Referência, na qual se baseará para construir o item. Caso tal habilidade já tenha sido usada em uma elaboração anterior, deve-se mostrar um aviso ao autor. A aplicação deve mostrar todas as matrizes cadastradas, disponibilizando um meio de fácil entendimento para a seleção da habilidade desejada. Os campos da estrutura do item só devem ser habilitados após a escolha de uma habilidade.

Como indicado no Guia de elaboração de itens, o texto-base poderá conter imagens, gráficos e tabelas, fazendo-se fundamental que aplicação forneça tal suporte para o elaborador, avaliando a qualidade da imagem sendo inserida, evitando imagens com deformações e resolução baixa. Para tanto, a aplicação deve fornecer um editor interativo que possibilite a inserção de imagens e tabelas, semelhante a editores de e-mail, verificando o tamanho das imagens carregadas. Ainda sobre o texto base, deverá ser fornecido um meio para inserção de possíveis referências bibliográficas, conforme as normas da ABNT.

O gabarito e os distratores terão estruturas semelhantes de cadastro, com um campo para o texto da alternativa e um segundo campo de justificativa, os quais a aplicação deverá validar que não estão vazios e que possuem extensões semelhantes.

A aplicação deverá realizar a validação dos textos, tanto nos campos cadastrados na Matriz de Referência quanto nos campos cadastrados no Item, verificando ortografia, gramática e pontuação, sugerindo ao elaborador possíveis correções.

Após o cadastro das matrizes e dos itens, a aplicação deverá fornecer a possibilidade da elaboração de um questionário MCQ. O elaborador selecionará uma das Matrizes de Referência cadastradas e o número de Itens que deseja utilizar no questionário. A aplicação deverá montar um questionário com Itens de todas as dificuldades, buscando um número similar de Itens por dificuldade, possibilitando o elaborador substituir ou remover Itens. Assim que a estrutura do questionário estiver pronta, o elaborador deve poder imprimir ou gerar um arquivo com o questionário montado.

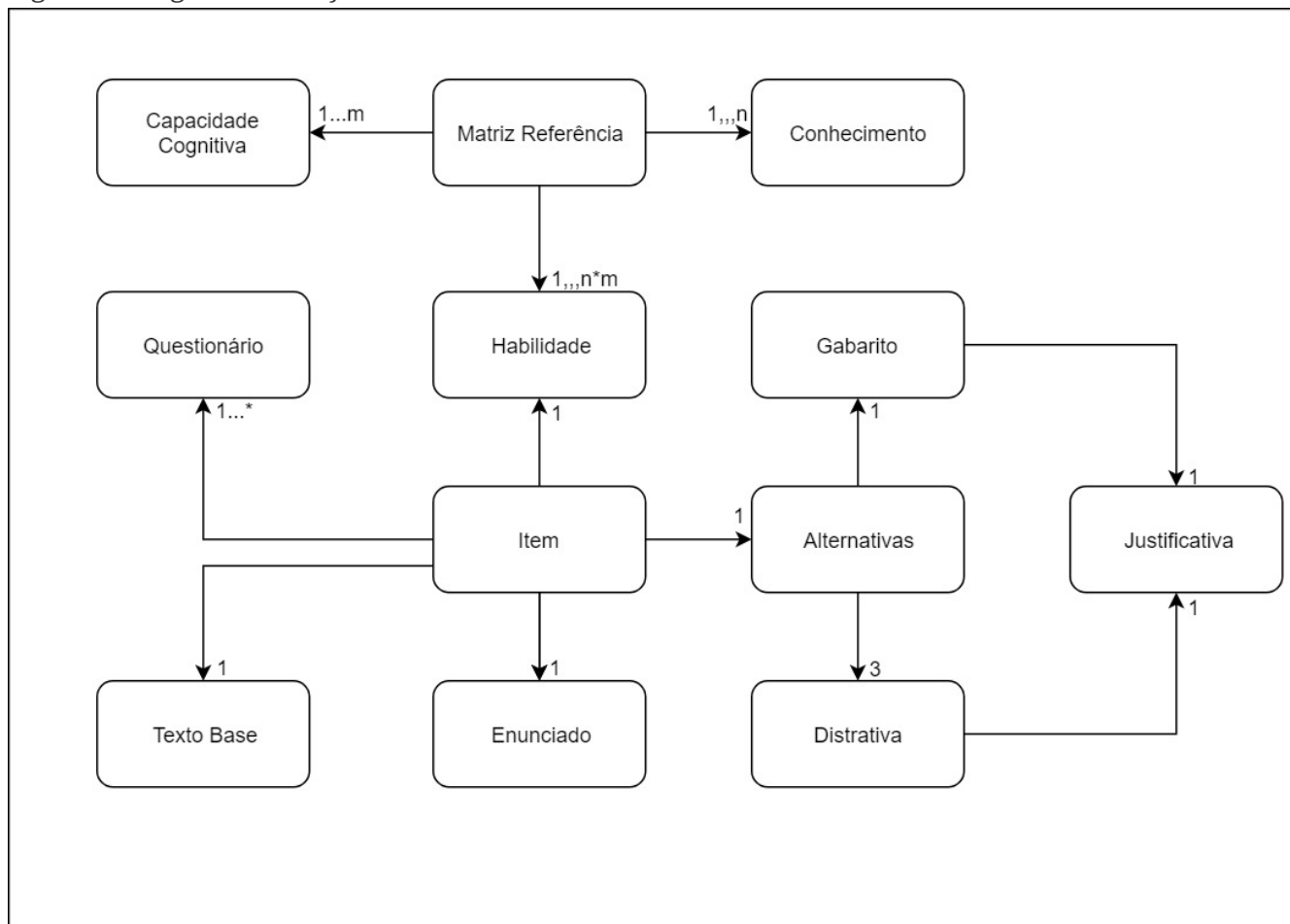
2.6. Modelo de Domínio

O Modelo de Domínio representa visualmente conceitos do mundo real do problema sendo estudado. As modelagens foram realizadas a partir do estudo da estrutura de uma Matriz de Referência e o Item descritos nos guias de elaboração do INEP.

A Figura 1 evidencia a relação dos objetos conceituais do domínio, denotando a relação quantitativa. A Matriz de Referência possui m Capacidades Cognitivas e n Conhecimentos e portanto poderá ter até $m*n$ Habilidades descritas. O Item sempre terá apenas uma Habilidade, Texto-base, Enunciado e quatro Alternativas, e como descrito no guia de elaboração de Itens são divididas em 3 Distratores e 1 Gabarito, todos esses possuindo uma justificativa.

O Questionário possuirá quantos Itens o elaborador achar necessário, variando pelo tipo de prova, tempo para executá-la, por exemplo.

Figura 3. Diagrama de Objetos do Domínio.



Text

Fonte: elaborado pelo autor

2.7. Arquitetura do Sistema

A arquitetura em camadas é uma técnica amplamente utilizada para separar as funções de um sistema de software. Analogamente a um bolo, os principais subsistemas do software são dispostos em camadas, como bolos em camadas, onde cada fica em cima da outra. As camadas mais altas interagem com as camadas adjacentes, mas desconhecem a existências das camadas seguintes (FOWLER, 2002).

O padrão arquitetural escolhido para implementação do Editor de Itens foi o *Model-View-Presenter* (MVP). O MVP define a divisão da camada de apresentação (*Presenter*) das camadas de dados e negócio, composto por três partes principais:

- *Model*: São os objetos que implementam os conceitos abstraídos modelo de domínio. No projeto sendo desenvolvido será separado em em *Model* da *view*, que são os objetos utilizado pela *View* e o *Model* propriamente dito que são relativos aos dados e as regras de negócio.
- *View*: Responsável pela interação com o usuário, definindo como os dados serão exibidos e

recebendo os inputs, notificando o Presenter da mudança de seu estado. Nesse padrão a *View* é passiva, ou seja, não tomará decisões sobre o que exibir, mas sim como exibir os dados fornecidos pelo Presenter.

- *Presenter*: Tem como objetivo fazer a conexão entre a *View* e os objetos do modelo. Se encarregará de realizar a transformação entre o *ViewModel* e o *Model*. Define o que deverá ser exibido pela View e ao mesmo tempo faz chamadas para a camada de Serviço.

A *View* se comunica com o *Presenter* através de um contrato, ou seja, por meio de uma *Interface*. Dessa forma haverá um desacoplamento do modo que a implementação da *View* realiza suas responsabilidades com as regras de apresentação e negócio, resultando em uma arquitetura mais flexível para implementação de diferentes interfaces com o usuário. Essa característica pode se apresentar interessante ao ponto que apesar do projeto prever a implementação de uma aplicação local, tal flexibilidade fornece a opção de fácil implementação de uma nova interface web ou de aplicativo smart-phone, por exemplo. Dessa forma, tal restrição mostra um ganho substancial quanto a facilidade de implementação de novas interfaces com o usuário sem aumentar muito a complexidade da aplicação. Além do padrão MVP, serão utilizados os padrões da Camada de Serviço e Repositório.

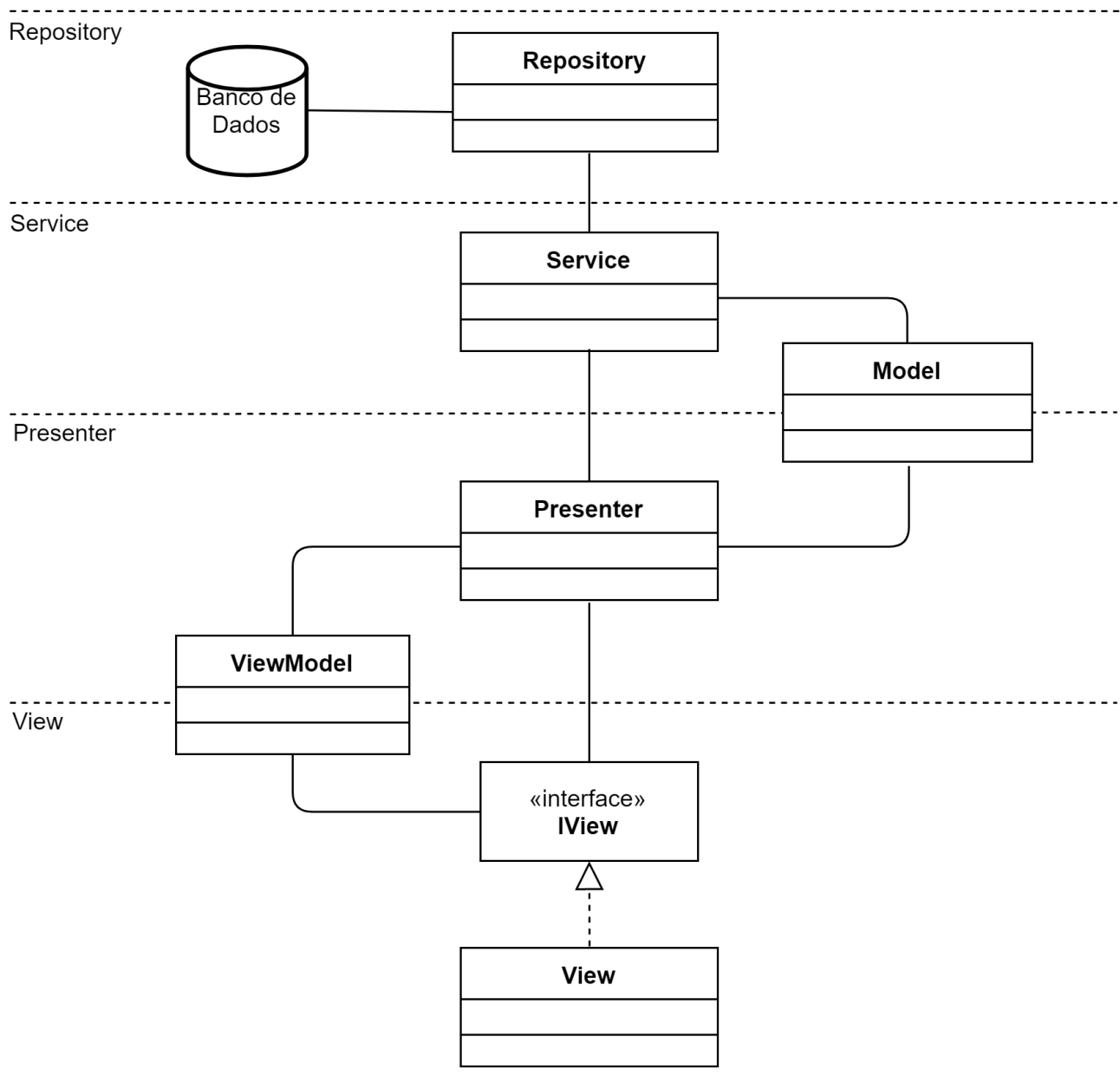
Um objeto do serviço está diretamente atrelado a um objeto de modelo, e realiza validações, cálculos e transformações sobre os objetos do modelo, realizando chamadas ao repositório para obter ou salvar os dados, devolvendo os objetos do modelo à camada de Apresentação. A camada de Serviço, portanto, realiza todas as regras de negócio da aplicação, intermediando a interação da camada de apresentação com os objetos do modelo. Dessa forma, a arquitetura define um escopo claro de onde os requisitos funcionais relativos às regras definidas no guia de elaboração serão implementados.

O Repositório está na camada de Persistência, responsável pela conexão e operações com o banco de dados, desacoplando assim a forma que os serviços realizam suas regras da obtenção dos dados. Um repositório é um objeto que realiza as conversões do objeto do modelo para o objeto de banco de dados e vice-versa. Conceitualmente, o repositório encapsula os objetos persistidos no banco de dados e todas as operações realizadas sobre tais objetos, como criação, obtenção, alteração e deleção dos dados (FOWLER, 2002). Assim como o serviço, o repositório também está ligado a um objeto do modelo.

Considerando assim o Item do INEP como um objeto do modelo, o Serviço do Item do INEP realizará todas as validações descritas nos guias de elaboração, enquanto o Repositório do Item do INEP, será responsável por obter e salvar os Itens no banco de dados.

A arquitetura proposta contempla a implementação de uma aplicação local, requisito do projeto sendo desenvolvido, ao mesmo tempo que fornece flexibilidade e desacoplamento das camadas para fácil manutenção.

Figura 4. Modelo Arquitetural do Editor de Itens.



Fonte: elaborado pelo autor

3. Modelo de Implementação

3.1. Protótipos de Interface

O diagrama de Interface com o Ambiente (ICA) fornece uma primeira visão de como os requisitos funcionais descritos são aplicados na interação com o usuário. A partir dos requisitos funcionais, são desenhados os campos de cadastro que precisam ser dispostos ao usuário para o cadastro dos objetos do modelo de domínio. Cada campo de cadastro representa uma parte da estrutura do objeto, fornecendo controles distintos conforme os guias de elaboração. Nos ICAs elaborados nas figuras abaixo, mostram protótipos da tela de cadastro de matrizes, cadastro de Itens e montagem do questionário. No cadastro de Matrizes de Referência, o elaborador fornecerá um nome, e poderá adicionar quantas colunas de competência e linhas de capacidades cognitivas que necessitar, criando os campos de habilidade. Em cada uma das células, poderá realizar a edição do texto para posteriormente salvar. O ICA do cadastro de Itens, mostra todos os campos necessários para a

elaboração do item, separando o que é a estrutura do Item (Texto-base, Enunciado e Alternativas) das informações relativas às pré-condições para sua elaboração.

Figura 5. ICA do Editor de Itens.

Editor de Matriz de Referência

Nome:

Competências \ Capacidades Cognitivas	K1	K2	K3	K4	K5	
C1	H1	H2	H3	H4	H5	+
C2	H6	H7	H8	H9	H10	
C3	H11	H12	H13	H14	H15	
+						

Salvar

Fonte: elaborado pelo autor

Figura 6. ICA do Editor de Matriz de Referência.

Editor de Itens

Matriz: ▾ Habilidade: ▾

Dificuldade: ▾

Texto-Base

Inserir Imagem Inserir Referência bibliográfica

Enunciado

Gabarito

Texto

Justificativa

Distratores

Texto

Justificativa

Texto

Justificativa

Texto

Justificativa

Salvar

Fonte: elaborado pelo autor

4. Bibliografia

FOWLER, M. et al. Patterns of enterprise application architecture: subtítulo do livro. Edição. [S.L.]: Addison Wesley, 2002. 560 p.

GAMMA, E. et al. Design patterns: Elements of Reusable Object-Oriented Software. 37 ed. [S.L.]: Addison Wesley, 1994. 395 p.

ROMAN, Gruia-Catalin. A Taxonomy of Current Issues in Requirements Engineering. Washington University in St. Louis, [S.L.], set./mar. 2018.

INEP. Guia de elaboração de itens provinha brasil. Disponível em: http://download.inep.gov.br/educacao_basica/provinha_brasil/documentos/2012/guia_elaboracao_itens_provinha_brasil.pdf.

Acesso em: 11 mar. 2018.

INEP. Matrizes de referência. Disponível em: <http://portal.inep.gov.br/educacao-basica/encceja/matrizes-de-referencia>. Acesso em: 11 mar. 2018.