

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DO AMAZONAS
DEPARTAMENTO DE ENSINO SUPERIOR
CURSO DE TECNOLOGIA EM ANÁLISE E
DESENVOLVIMENTO DE SISTEMAS**

Francisco Marcelo Mendes Damasceno

POSCOMP *Coach* - Uma Plataforma Web para auxiliar na Preparação de
Estudantes para o Exame do POSCOMP.

Manaus, Amazonas

2017

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DO AMAZONAS**
DEPARTAMENTO DE ENSINO SUPERIOR
**CURSO DE TECNOLOGIA EM ANÁLISE E
DESENVOLVIMENTO DE SISTEMAS**

Francisco Marcelo Mendes Damasceno

POSCOMP *Coach* - Uma Plataforma Web para auxiliar na Preparação de
Estudantes para o Exame do POSCOMP.

Trabalho de Conclusão de Curso apresentado
à banca examinadora do Curso Superior de
Tecnologia em Análise e Desenvolvimento de
Sistemas do Instituto Federal de Educação,
Ciência e Tecnologia do Amazonas (IFAM),
Campus Manaus Centro, como requisito para
a disciplina Trabalho de Conclusão de Curso
II - Desenvolvimento de Software.

Profa. Dra. Andréa P. Mendonça

(Orientadora)

Profa. Dra. Elloá Barreto Guedes da Costa

(Coorientadora)

Manaus, Amazonas

2017

POSCOMP *Coach* - Uma Plataforma Web para auxiliar na Preparação de
Estudantes para o Exame do POSCOMP.

Banca Examinadora

**Profa. Dra. Andréa P.
Mendonça**

Orientadora Acadêmica
Instituto Federal de Educação,
Ciência e Tecnologia do Amazonas -
IFAM

**Profa. Dra. Joyce Miranda dos
Santos**

Instituto Federal de Educação,
Ciência e Tecnologia do Amazonas -
IFAM

**Prof. Dr. Jucimar Brito de
Souza**

Instituto Federal de Educação,
Ciência e Tecnologia do Amazonas -
IFAM

Manaus, Amazonas

2017

Resumo

Existe hoje no Brasil uma tendência de crescimento com relação a procura por programas de pós-graduação em computação, e essa tendência tem se confirmado nas últimas duas décadas. O Exame Nacional para Ingresso na Pós-Graduação em Computação (POSCOMP) é um exame idealizado e concebido pela Sociedade Brasileira de Computação (SBC) e utilizado por várias instituições de ensino como instrumento de seleção para os seus programas de Pós-Graduação. O POSCOMP é um exame constituído de 70 questões, distribuídas em três áreas – Matemática, Fundamentos da Computação e Tecnologia. As provas de anos anteriores são disponibilizadas no site da SBC em forma de arquivos PDF, mas atualmente não são disponibilizadas, por parte da SBC, ferramentas que automatizem o processo e ajudem o candidato a gerenciar os estudos com esse material. Embora existam algumas iniciativas de terceiros, estas possuem uma quantidade limitada de recursos e funcionalidades. Portanto, no intuito de mitigar essas dificuldades e melhorar a preparação dos candidatos que serão avaliados pelo seu desempenho no POSCOMP, foi desenvolvida neste trabalho de conclusão de curso uma plataforma Web denominada POSCOMP *Coach* que auxilia candidatos de Pós-Graduação em Computação na preparação para o POSCOMP, provendo suporte no planejamento e gerenciamento de seus estudos, através de criação de simulados com base nas questões de provas anteriores, avaliação de desempenho e histórico de estudos. O processo de desenvolvimento foi guiado por uma metodologia Ágil, por meio do Processo Unificado Ágil (AUP), tendo como tecnologias fundamentais a linguagem *Python*, o *framework* web Django, MySQL como sistema gerenciador de banco de dados, Git para controle de versão de código e \LaTeX para o processo de construção do banco de questões e edição deste documento. A plataforma foi avaliada por estudantes de computação e docentes das cinco regiões do país com respeito a diferentes critérios de usabilidade (prevenção a erros, linguagem, feedback, entre outros), obtendo um percentual de aprovação de, no mínimo, 80% em cada critério avaliado.

Palavras-chave: POSCOMP; Exame do POSCOMP; Pós-Graduação em Computação; SBC; Plataforma Web para o POSCOMP.

Abstract

There is a growing trend in Brazil in relation to demand for graduate programs in computing, and this trend has been confirmed in the last two decades. The National Exam for Post-Graduation in Computing (POSCOMP) is an idealized exam designed by the Brazilian Computer Society (SBC) and used by several educational institutions as a selection tool for its graduate programs. POSCOMP is an exam consisting of 70 items, distributed in three areas – Mathematics, Fundamentals of Computing and Technology. The tests from previous years are available on the SBC website in the form of PDF files, but SBC currently does not provide tools to automate the process and help candidates to manage their studies by using this material. While there are some third-party initiatives, they have a limited features and functionality. Therefore, in order to mitigate these difficulties and improve the preparation of the candidates that will be evaluated for their performance in POSCOMP, a Web Platform called POSCOMP Coach was developed to assist Post-Graduation in Computing students in preparation for the POSCOMP exam, by providing suport in the planning and management of their studies, through the creation of simulated exams based on previous exams, performance evaluation and study history. The development process was guided by an Agile methodology, through the Agile Unified Process (AUP), having as fundamental technologies the Python language, the Django web framework, MySQL as database management system , Git for code version control and L^AT_EX which was extremely important in the process of building the issue bank and editing this document. The web platform was evaluated by computer science students and teachers from all five country regions in regards of several usability criterias (error prevention, language, feedback, among others), obtainng at least 80% approval indice in all of the criterias.

Keywords: POSCOMP; POSCOMP exam; Post-Graduation in Computing; SBC; Web platform for POSCOMP.

Lista de figuras

Figura 1 – Formação de Mestres e Doutores - por período de avaliação	11
Figura 2 – Estrutura de um Item.	16
Figura 3 – Exemplo de QME: Questão extraída do exame POSCOMP 2014 - Fundamentos da Computação.	17
Figura 4 – Exemplo de QMEC: Questão extraída do exame POSCOMP 2015 - Fundamentos da Computação.	17
Figura 5 – Ciclo de vida do AUP.	20
Figura 6 – Composição do Banco de Questões.	32
Figura 7 – Questão 02/POSCOMP 2011 - Formato de Dicionário <i>Python</i>	33
Figura 8 – Fluxo dos dados até a persistência.	34
Figura 9 – PNG vs SVG - Exemplo.	34
Figura 10 – PNG vs SVG - Comparativo em detalhe no POSCOMP Coach.	35
Figura 11 – Itens já persistidos pelo módulo <i>utils.py</i>	35
Figura 12 – Modelo Conceitual - POSCOMP <i>Coach</i>	37
Figura 13 – Diagrama de Pacotes - POSCOMP <i>Coach</i>	38
Figura 14 – Caso de Uso - POSCOMP <i>Coach</i> : Usuário.	40
Figura 15 – Caso de Uso - POSCOMP <i>Coach</i> : Administrador.	41
Figura 16 – Diagrama de domínio - POSCOMP <i>Coach</i> : Módulo <i>base</i>	43
Figura 17 – Diagrama de domínio - POSCOMP <i>Coach</i> : Módulo <i>mock</i>	44
Figura 18 – Diagrama de domínio - POSCOMP <i>Coach</i> : Módulo <i>account</i>	44
Figura 19 – POSCOMP <i>Coach</i> : MER.	46
Figura 20 – Região do país em que cursa ou cursou a graduação.	49
Figura 21 – Curso de Graduação que está cursando ou cursou.	50
Figura 22 – Situação no curso.	50
Figura 23 – Frequência de uso.	52
Figura 24 – UI - POSCOMP <i>Coach</i> : Página inicial (<i>Landing Page</i>).	69
Figura 25 – UI - POSCOMP <i>Coach</i> : Dashboard (Usuário).	70
Figura 26 – UI - POSCOMP <i>Coach</i> : Perfil.	71
Figura 27 – UI - POSCOMP <i>Coach</i> : Meus simulados.	72
Figura 28 – UI - POSCOMP <i>Coach</i> : Simulado por ano (padrão).	73
Figura 29 – UI - POSCOMP <i>Coach</i> : Simulado por Área.	74
Figura 30 – UI - POSCOMP <i>Coach</i> : Personalizado.	75
Figura 31 – UI - POSCOMP <i>Coach</i> : Análise de Desempenho (GERAL).	76
Figura 32 – UI - POSCOMP <i>Coach</i> : Análise de Desempenho (Por Área).	77
Figura 33 – UI - POSCOMP <i>Coach</i> : Progressão individual (Por Área).	78
Figura 34 – UI - POSCOMP <i>Coach</i> : <i>Leaderboard</i>	79

Figura 35 – UI - POSCOMP <i>Coach</i> : Dashboard (Administrador).	80
Figura 36 – UI - POSCOMP <i>Coach</i> : Editar Item.	81
Figura 37 – UI - POSCOMP <i>Coach</i> : Listagem de Exames.	82
Figura 38 – UI - POSCOMP <i>Coach</i> : Estatísticas gerais - Nível de dificuldade das questões	83
Figura 39 – UI - POSCOMP <i>Coach</i> : Estatísticas gerais - Usuários por IES Públi- ca/Privada.	84
Figura 40 – UI - POSCOMP <i>Coach</i> : Distribuição geográfica dos usuários.	85
Figura 41 – POSCOMP <i>Coach</i> : MER Completo.	87
Figura 42 – Exemplo de atributo com @property em Python	89

Lista de quadros

Quadro 1 – Correlação entre entidades do Modelo Conceitual e Diagrama de Classes.	42
Quadro 2 – Correlação Conceito, Classe e Tabela.	45
Quadro 3 – Caso de Uso CDU01: Registrar na plataforma.	61
Quadro 4 – Caso de Uso CDU02: Logar na plataforma.	62
Quadro 5 – Caso de Uso CDU03: Recuperar Senha.	62
Quadro 6 – Caso de Uso CDU04: Editar Perfil.	63
Quadro 7 – Caso de Uso CDU05: Preparar Simulado.	64
Quadro 8 – Caso de Uso CDU06: Realizar Simulado.	65
Quadro 9 – Caso de Uso CDU07: Listar Simulados.	65
Quadro 10 – Caso de Uso CDU08: Visualizar Gabarito.	66
Quadro 11 – Caso de Uso CDU09: Revisar Simulado.	66
Quadro 12 – Caso de Uso CDU10: Visualizar Desempenho.	66
Quadro 13 – Caso de Uso CDU11: Listar exames.	67
Quadro 14 – Caso de Uso CDU12: Pesquisar Itens.	67
Quadro 15 – Caso de Uso CDU13: Editar Item.	68
Quadro 16 – Caso de Uso CDU14: Editar Alternativa.	68

Lista de tabelas

Tabela 1	–	Formação de Mestres e Doutores - por período de avaliação.	11
Tabela 2	–	Tipos de Questões de Múltipla Escolha (QME)	15
Tabela 3	–	Distribuição das questões - POSCOMP)	18
Tabela 4	–	Análise Comparativa do POSCOMP Coach em relação as outras alter- nativas existentes.	19
Tabela 5	–	Artefatos gerados ao longo do processo.	22
Tabela 6	–	Diagramas oficiais da UML	23

Sumário

1	CONTEXTUALIZAÇÃO E PROBLEMÁTICA	11
1.1	Objetivos	13
1.1.1	Objetivo Geral	13
1.1.2	Objetivos específicos	13
1.2	Justificativa	14
1.3	Organização do Documento	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Itens de Múltipla Escolha	15
2.1.1	Estrutura de um Item	16
2.2	O Exame do POSCOMP	16
2.3	Trabalhos Relacionados	18
2.4	Tecnologias adotadas no desenvolvimento do POSCOMP <i>Coach</i>	20
2.4.1	Agile Unified Process	20
2.4.2	Unified Modeling Language	23
2.4.3	Python	24
2.4.4	Django	25
2.4.5	MySQL	26
2.4.6	Git	27
2.4.7	L ^A T _E X	27
2.4.8	Bootstrap	28
2.4.9	ChartJS	29
3	POSCOMP <i>COACH</i>	30
3.1	Visão Geral	30
3.2	Construção do Banco de Questões	31
3.3	Modelo Conceitual	36
3.4	Detalhamento da plataforma	37
3.5	Diagramas de casos de uso	39
3.6	Diagrama de classes	41
3.7	Modelo Entidade Relacionamento - MER	45
3.8	Desenvolvimento incremental e Interfaces	46
4	AValiação DO POSCOMP <i>COACH</i>	48
4.1	Metodologia	48
4.2	Análise dos resultados	49

4.2.1	Perfil dos avaliadores	49
4.2.2	Avaliação de Usabilidade	51
4.2.3	Sugestões e <i>Feedback</i>	52
	Considerações Finais	54
	REFERÊNCIAS	56
	APÊNDICES	60
	APÊNDICE A – ESPECIFICAÇÃO DE CASOS DE USO: REGIS- TRO E AUTENTICAÇÃO	61
	APÊNDICE B – ESPECIFICAÇÃO DE CASOS DE USO: VISÃO DO CANDIDATO	64
	APÊNDICE C – ESPECIFICAÇÃO DE CASOS DE USO: VISÃO DO ADMINISTRADOR	67
	APÊNDICE D – INTERFACES DO USUÁRIO	69
	APÊNDICE E – DIAGRAMAS COMPLEMENTARES	86
	APÊNDICE F – CONTROLE DE ACESSO, PUBLICIDADE E VI- SIBILIDADE DE ATRIBUTOS EM PYTHON	88
	ANEXOS	90
	ANEXO A – POSCOMP COACH - FORMULÁRIO DE AVALIAÇÃO.	91

1 Contextualização e Problemática

Existe hoje no Brasil uma tendência de crescimento com relação à procura por programas de pós-graduação em Computação, e essa tendência tem se confirmado nas últimas duas décadas.

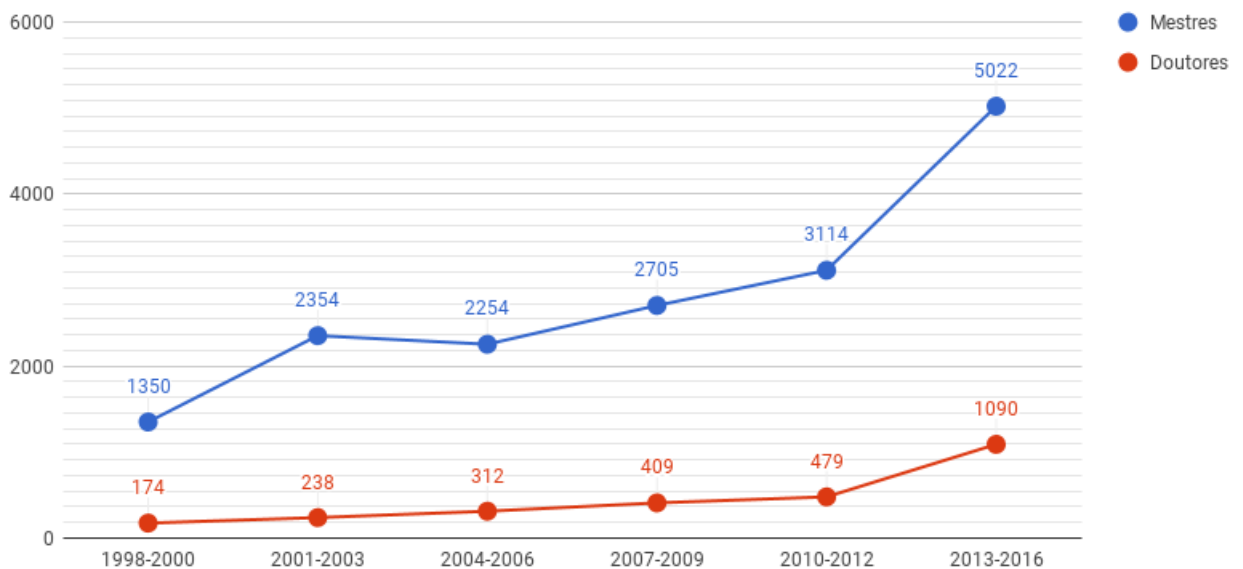
De fato, informações coletadas recentemente da CAPES¹, através de seus relatórios de avaliação periódicos, mostram um significativo aumento no número de pós graduados em Computação ao longo da década. Os dados extraídos dos relatórios da Capes foram sintetizados na Tabela 1. Na Figura 1 temos um gráfico de linha que demonstra de forma mais clara essa tendência de crescimento.

Tabela 1 – Formação de Mestres e Doutores - por período de avaliação.

	1998-2000	2001-2003	2004-2006	2007-2009	2010-2012	2013-2016
Mestres	1350	2354	2254	2705	3114	5022
Doutores	174	238	312	409	479	1090

Fonte: (CAPES, 2017).

Figura 1 – Formação de Mestres e Doutores - por período de avaliação



Fonte: Elaborado pelo autor com base em (CAPES, 2017).

Segundo Medeiros (2011, p.146), "[...] mesmo havendo queda na demanda pela graduação, continua o interesse, no Brasil, por formação avançada. Isso, aliás, é comprovado

¹ Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - <http://www.capes.gov.br/>

pelos números do POSCOMP — um exame aplicado pela SBC em todas as regiões do país".

O Exame Nacional para Ingresso na Pós-Graduação em Computação (POSCOMP) é um exame idealizado e concebido pela Sociedade Brasileira de Computação (SBC²) e utilizado por várias instituições de ensino como um instrumento para auxiliar no processo de seleção para os seus Programas de Pós-Graduação em Computação (PPGC). O POSCOMP consiste de uma prova com 70 questões distribuídas em três áreas fundamentais – Matemática, Fundamentos da Computação e Tecnologia. Ele testa os conhecimentos dos alunos na área de Computação, e tem como objetivo principal avaliar os candidatos postulantes às vagas nos programas de pós-graduação em Computação oferecidos em todo o Brasil.

O POSCOMP é aplicado no Brasil desde 2002 e desde 2006 também é aplicada no Peru graças a uma parceria firmada com a Sociedade Peruana de Computação. No Brasil o exame é aplicado atualmente pela Fundação Universidade Empresa de Tecnologia e Ciências (Fundatec³), e ocorre todos os anos. A prova possui abrangência nacional, e portanto permite ao candidato realizar a prova na cidade onde reside, mesmo que a vaga pleiteada seja em uma instituição localizada em outro estado. Muitas instituições utilizam o POSCOMP como requisito obrigatório para a seleção, mas isso não é regra. Os programas podem optar por utilizar o resultado do exame da maneira que lhe for mais conveniente, estabelecendo ou não sua obrigatoriedade ou mesmo utilizando apenas parte das questões. Pode-se tomar como exemplo o Instituto de Matemática e Estatística da USP (IME-USP⁴) que recomenda fortemente o resultado do POSCOMP para compor seus critérios de seleção, ou o Instituto de Computação da Universidade de Campinas (IC-UNICAMP⁵) que no critério formação acadêmica do candidato considera o histórico escolar, currículo, cartas de recomendação, plano de pesquisa, carta de apresentação e o desempenho do candidato no POSCOMP.

Neste contexto, é possível afirmar que o POSCOMP é hoje elemento fundamental entre os instrumentos disponíveis para que as instituições de ensino possam embasar e fortalecer os seus métodos de seleção para o ingresso nos seus programas de pós-graduação em Computação.

Apesar da importância deste exame no processo seletivo das diversas pós-graduações no país, a preparação para o POSCOMP é uma das maiores dificuldades dos postulantes à uma vaga nestes programas de pós-graduação, isto porque, embora a SBC disponibilize em seu site todos os exames realizados desde sua primeira edição no ano de 2002, estes estão em formato PDF, o que limita as possibilidades de estudo do candidato, que precisa manter

² <http://www.sbc.org.br>

³ <http://www.fundatec.org.br/>

⁴ <https://www.ime.usp.br/dcc/pos/selecao>

⁵ <http://www.ic.unicamp.br/pos/ingresso>

o controle das questões e provas resolvidas e a resolver, bem como computar manualmente acertos, erros e ainda visualizar seu desempenho nas áreas abrangidas pelo exame, e finalmente depende dele próprio estabelecer as estatísticas de seu próprio desempenho para saber onde precisa investir mais esforço e estudo.

No intuito de mitigar essas dificuldades, foi desenvolvido neste trabalho de conclusão de curso uma plataforma Web denominada **POSCOMP *Coach***, que tem como objetivo prover suporte para planejamento e gerenciamento dos estudos para o POSCOMP, permitindo aos candidatos, dentre outras coisas, configurar simulados com gerenciamento das questões por área do conhecimento (Matemática, Fundamentos da Computação e Tecnologias) utilizando as questões de exames anteriores e possibilitando a inclusão de novas questões de exames futuros, manter o registro de questões resolvidas e a resolver, gerar relatórios estatísticos de desempenho, e auxiliar no planejamento dos estudos. Para isto foram utilizados como base tecnológica⁶ HTML5, Javascript (ChartJS), CSS (Bootstrap), Python, Django, MySQL e L^AT_EX.

1.1 Objetivos

Os objetivos deste projeto são os descritos a seguir.

1.1.1 Objetivo Geral

Desenvolver uma plataforma Web que auxilie candidatos de Pós-Graduação em Computação na preparação para o POSCOMP, provendo suporte no planejamento e gerenciamento de seus estudos, através de criação de simulados com base nas questões de provas anteriores, avaliação de desempenho e histórico de estudos.

1.1.2 Objetivos específicos

1. Especificar os requisitos do POSCOMP *Coach* para melhor favorecer a preparação dos candidatos;
2. Implementar um banco de itens povoando-o com questões dos exames anteriores do POSCOMP;
3. Implementar uma interface web que permita aos candidatos a utilização dos recursos disponíveis na plataforma;
4. Implementar recursos para avaliação de desempenho dos candidatos;
5. Implantar a plataforma em um ambiente controlado para fins de demonstração e testes.

⁶ Todas as tecnologias serão abordadas no [Capítulo 2](#) que trata da Fundamentação Teórica

1.2 Justificativa

A procura por pós-graduação em Computação vem aumentando em decorrência da demanda cada vez maior de mão de obra especializada, seja no âmbito acadêmico ou no mercado. Com a importância do POSCOMP e seu uso cada vez maior pelas instituições de ensino nos seus respectivos programas de pós-graduação em Computação, a necessidade de uma preparação mais eficaz por parte dos candidatos também aumenta.

Tendo em vista as ações da SBC no Brasil e como responsável pela realização do POSCOMP, ela poderá disponibilizar os serviços desta plataforma para os interessados nos PPGCs⁷, melhorando consideravelmente a forma como hoje são disponibilizadas as informações dos exames, de forma estática. Além disso, a plataforma contribui também para as próprias instituições de ensino em seus PPGCs que poderão incentivar os alunos na preparação para o exame.

Além disso, o POSCOMP *Coach* beneficia diretamente os candidatos na medida em facilita o acesso aos exames, gerenciamento e planejamento dos estudos. Por meio do uso da plataforma, o candidato terá ao seu dispor índices de desempenho que o ajudarão no planejamento autônomo dos seus estudos.

Podemos dizer que o efeito cascata com POSCOMP *Coach* é benéfico também para o universo acadêmico de Computação como um todo, pois melhora a qualidade do estudo dos candidatos, favorece a obtenção de resultados e pontuações melhores, resultando em candidatos motivados e comunidade científica da Computação fortalecida.

1.3 Organização do Documento

O presente documento está organizado em quatro capítulos. No [Capítulo 2](#) será abordado a fundamentação teórica conceitual, isto é, todos os conceitos teóricos que fundamentam a temática em questão. Os requisitos e modelagem para construção da plataforma POSCOMP *Coach* serão apresentadas no [Capítulo 3](#). No [Capítulo 4](#) é feita uma análise sobre os resultados do questionário avaliativo sobre a usabilidade da plataforma, seguido das considerações finais.

⁷ Programas de Pós-Graduação em Computação

2 Fundamentação Teórica

Neste capítulo são apresentados os fundamentos teóricos nos quais este projeto se embasa. Além destes, também serão apresentadas os trabalhos relacionados e as tecnologias que foram utilizadas no desenvolvimento do POSCOMP *Coach*.

2.1 Itens de Múltipla Escolha

Um item é uma unidade básica de observação em um teste (HALADYNA, 2004). Itens de múltipla escolha, também conhecidos como Questões de Múltipla Escolha (QME), são um dos instrumentos de avaliação educacional existentes mais difundidos e pesquisados (PARKES; ZIMMARO, 2016).

As Questões de Múltipla Escolha são bastante utilizadas em avaliações escolares, mas são utilizadas principalmente em larga escala, como por exemplo os exames de concursos públicos, Exame Nacional do Ensino Médio (ENEM) e o próprio Exame do POSCOMP. Observando-se as vantagens, como por exemplo o tempo de correção e validação num contexto em larga escala, as QMEs podem ser uma efetiva e eficiente maneira de mensurar os resultados de aprendizagem (BRAME, 2013).

Existem muitos tipos de questões de múltipla escolha, desenvolvidas para medir resultados cognitivos complexos. O que diferencia esses tipos são a forma como as opções de resposta estão estruturadas (BERK, 1996). Na Tabela 2 estão listados alguns dos tipos mais utilizados nos livros e estudos sobre testes educacionais, segundo Haladyna, Downing e Rodriguez (2002).

Tabela 2 – Tipos de Questões de Múltipla Escolha (QME)

<i>Formato</i>	<i>Descrição</i>
Múltipla Escolha Convencional	Enunciado com 3 a 5 alternativas
Escolha a alternativa	Enunciado e 2 alternativas
Correspondência	3 a 12 opções precedendo um grupo de enunciados
Múltipla Verdadeira-Falsa	Enunciado com 3 a 20 alternativas. Cada opção é avaliada em termos de sua veracidade
Verdadeiro ou Falso	Declaração afirmativa avaliada em termos de sua veracidade
Item dependente de contexto	Um estímulo seguindo por um ou mais itens A resposta de cada item dependente do material de estímulo
Múltipla Escolha Complexa	Um Enunciado seguido de alternativas que são agrupadas para escolha

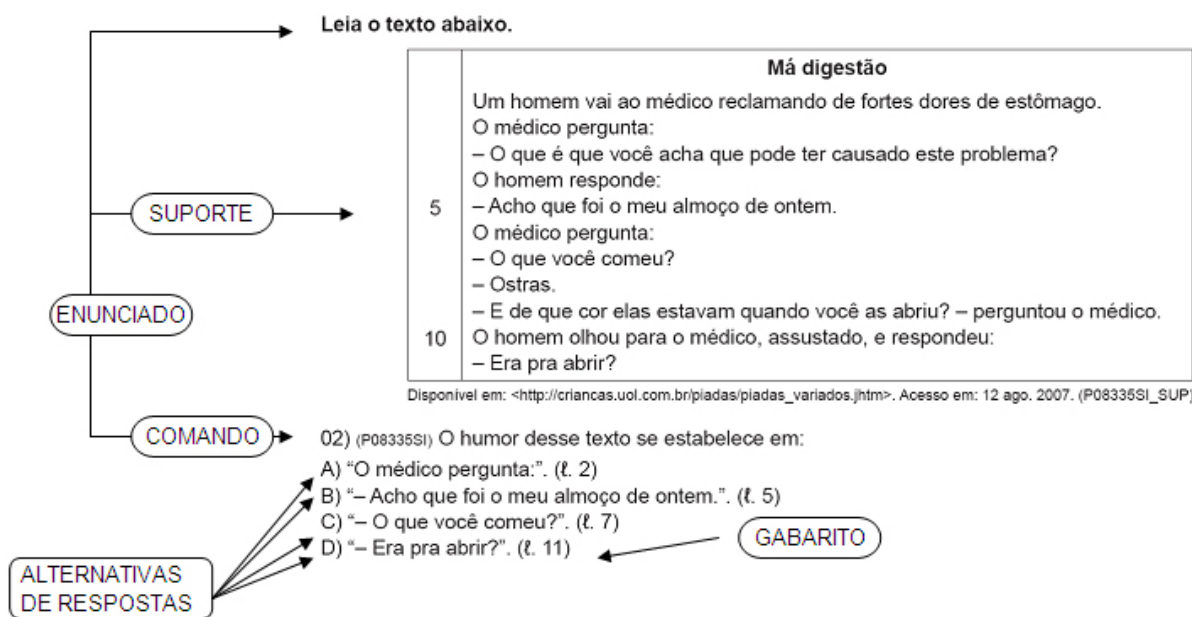
Fonte: Haladyna, Downing e Rodriguez (2002, p. 309-333)

Tendo em vista o contexto deste projeto, que tem como foco os exames do POSCOMP, constarão no banco de itens da plataforma basicamente dois tipos de QME: Múltipla Escolha Convencional e Múltipla Escolha Complexa, conforme serão ilustradas na [seção 2.2](#) O Exame do POSCOMP.

2.1.1 Estrutura de um Item

Uma Item é composto basicamente de um **enunciado**, que compreende um problema ou questão, seguido por um conjunto de alternativas, em que uma destas **alternativas** é a correta, ou **gabarito**, e as outras alternativas são incorretas ou parcialmente corretas, também chamadas de distratores. O enunciado pode conter texto, imagem e outros recursos auxiliares que recebem o nome de **suporte**, e o **comando para resposta** pode ser dado sob a forma de complementação ou de interrogação (CAED, 2012).

Figura 2 – Estrutura de um Item.



Fonte: CAEd (2016).

Uma vez apresentado o conceito e estrutura das questões de múltipla escolha, será apresentado a seguir o exame do POSCOMP, sua estrutura e os tipos de itens que o compõe.

2.2 O Exame do POSCOMP

Conforme descrito anteriormente no [Capítulo 1](#) Contextualização e Problemática, o POSCOMP é um Exame realizado desde 2002 pela Sociedade Brasileira de Computação (SBC) em escala nacional, utilizado por diversas instituições de Ensino Superior em

Computação como instrumento importante, de carácter classificatório, para o ingresso em seus Programas de Pós-Graduação em Computação.

O POSCOMP é realizado por meio de uma prova objetiva, com duração de quatro (4) horas e composta por 70 questões de múltipla escolha, cada uma contendo 5 alternativas das quais apenas uma é a resposta correta, conforme exemplo ilustrado na Figura 3. A prova é subdividida em três áreas do conhecimento: Matemática (20 questões), Fundamentos da Computação (30 questões) e Tecnologia da Computação (20 questões), cada área com suas subáreas e quantidade de questões associadas, conforme detalhado na Tabela 3.

Nas ilustrações Figura 3 e Figura 4 são apresentados os dois tipos de Itens utilizados no POSCOMP – Questão de Múltipla Escolha (QME) e Questão de Múltipla Escolha Complexa (QMEC) respectivamente.

Figura 3 – Exemplo de QME: Questão extraída do exame POSCOMP 2014 - Fundamentos da Computação.

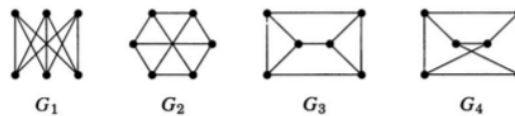
- 24** Sejam uma árvore AVL A , r a raiz de uma subárvore S de A e a_d e a_e , respectivamente, as alturas das subárvores direita e esquerda de S . Em relação a esse tema, assinale a alternativa correta.
- $a_e = a_d$
 - $\frac{a_e + a_d}{2} = 2a_e$
 - Considere que $a_e < a_d$, portanto o valor de a_d pode ser qualquer valor no intervalo $[a_e, 2a_e]$.
 - Considere que $a_e < a_d$, portanto o valor de a_d pode ser qualquer valor no intervalo $[a_e, 2^{a_e}]$.
 - $|a_e - a_d| = 1$

Fonte: POSCOMP 2014.

Figura 4 – Exemplo de QMEC: Questão extraída do exame POSCOMP 2015 - Fundamentos da Computação.

— QUESTÃO 36 —

Considere os grafos, a seguir.



Pela análise desses grafos, verifica-se que

- G_3 e G_4 são grafos completos.
- G_1 e G_2 são grafos isomorfos.
- G_3 e G_1 são grafos bipartidos.
- G_2 e G_3 são grafos planares.
- G_4 e G_1 são multigrafos.

Fonte: POSCOMP 2015.

Tabela 3 – Distribuição das questões - POSCOMP.

Área do Conhecimento	Subárea	Nº Questões
Matemática	Álgebra Linear	20
	Análise Combinatória	
	Cálculo Diferencial e Integral	
	Geometria Analítica	
	Lógica Matemática	
	Matemática Discreta	
	Probabilidade e Estatística	
Fundamentos da Computação	Análise de Algoritmos	30
	Algoritmos e Estrutura de Dados	
	Arquitetura e Organização dos Computadores	
	Circuitos Digitais	
	Linguagens de Programação	
	Linguagens Formais, Autômatos e Computabilidade	
	Organização de Arquivos e Dados	
	Sistemas Operacionais	
	Técnicas de Programação	
	Teoria dos Grafos	
Tecnologia de Computação	Banco de Dados	20
	Compiladores	
	Computação Gráfica	
	Engenharia de Software	
	Inteligência Artificial	
	Processamento de Imagens	
	Redes de Computadores	
	Sistemas Distribuídos	
Total de Questões		70

Fonte: Adaptada de: [SBC \(2016\)](#).

Nesta subseção foram apresentadas informações referentes ao Exame do POSCOMP e sua estrutura básica. Na próxima seção descreve-se os trabalhos relacionados e uma breve comparação de funcionalidades e aspectos relevantes entre esses projetos e o POSCOMP *Coach*, solução proposta neste documento.

2.3 Trabalhos Relacionados

Durante a revisão bibliográfica, foram identificadas algumas plataformas web que possuem funcionalidades similares às desenvolvidas neste projeto, como é caso do *app* Questões para POSCOMP ([SILVA, 2017](#)) e do Ambiente Colaborativo para Treinamento POSCOMP (ACTCOMP) ([de Sordi Junior, 2015](#)). O *app* Questões para POSCOMP ([SILVA, 2017](#)) foi desenvolvido para a plataforma Android e possui questões agrupadas por tópicos – Banco de Dados, Engenharia de Software, Estrutura de Dados, Linguagens Formais, Autônomos e Compiladores, Matemática, Redes e Teoria da Computação, fornecendo ao final da resolução o número de erros e acertos do usuário. A resolução das questões pode ser feita dentro do tempo real da prova (4 horas) e uma solução explicativa é apresentada quando o usuário marca a resposta incorreta. O ACTCOMP, por sua vez,

permite que estudantes resolvam questões no estilo do exame POSCOMP, isto é, questões de múltipla escolha com 5 alternativas e 1 gabarito, porém as mesmas não são oriundas das provas anteriores, mas sim de usuários cadastrados com perfil de professor que colaboram com a criação das questões e com a classificação das mesmas quanto à dificuldade, área do conhecimento e corretude (de Sordi Junior, 2015).

Uma primeira avaliação que pode ser realizada do POSCOMP *Coach* é em termos comparativos com estas alternativas existentes, considerando características e funcionalidades. Esta comparação é sintetizada na Tabela 4.

Tabela 4 – Análise Comparativa do POSCOMP Coach em relação as outras alternativas existentes.

Aspecto Considerado	Questões para POSCOMP	ACTCOMP	POSCOMP Coach
Tipo de Questões	POSCOMP	Criadas colaborativamente	POSCOMP
Base de Dados	76 questões	Não disponível	1120 questões
Soluções das Questões	Detalhadas	Gabarito	Gabarito
Personalização de Simulados	Não	Sim	Sim
Limites de Tempo para Resolução	Sim	Não	Sim
Visualização do Desempenho	Quantitativo	Quantitativo	Gráficos do desempenho geral e por área
Ranking	Não	Não	Sim
Plataforma	Android	Web	Web
Integração com Redes Sociais	Não	Facebook	Facebook e Google
Design Responsivo	Não se aplica	Sim	Sim

Fonte: Elaborado pelo autor.

Embora as soluções propostas na literatura colaborem para a minimização das dificuldades na preparação para o POSCOMP, é possível identificar algumas limitações das mesmas. O fato do Questões para POSCOMP, estar disponível em apenas um tipo de plataforma mobile limita o público alvo. A apresentação do desempenho do candidato apenas em termos quantitativos de erros e acertos dificulta a identificação dos pontos fracos, que demandam melhor preparação por parte dos candidatos. A outra solução existente, o ACTCOMP, não possui implementada a funcionalidade de geração de estatísticas sobre o desempenho, ainda que apresentada no menu ao usuário. Além disso, a criação do banco de questões diversificado depende da colaboração dos usuários do sistema, o que tem se mostrado um limitador em termos práticos visto que o número de questões disponíveis aparenta ser reduzido, ainda que não informado pelos autores.

Nesta seção foram abordados os trabalhos relacionados e seus aspectos, semelhanças e diferenças em relação ao POSCOMP *Coach*. A seguir serão apresentadas as tecnologias utilizadas no desenvolvimento da plataforma Web.

2.4 Tecnologias adotadas no desenvolvimento do POSCOMP Coach

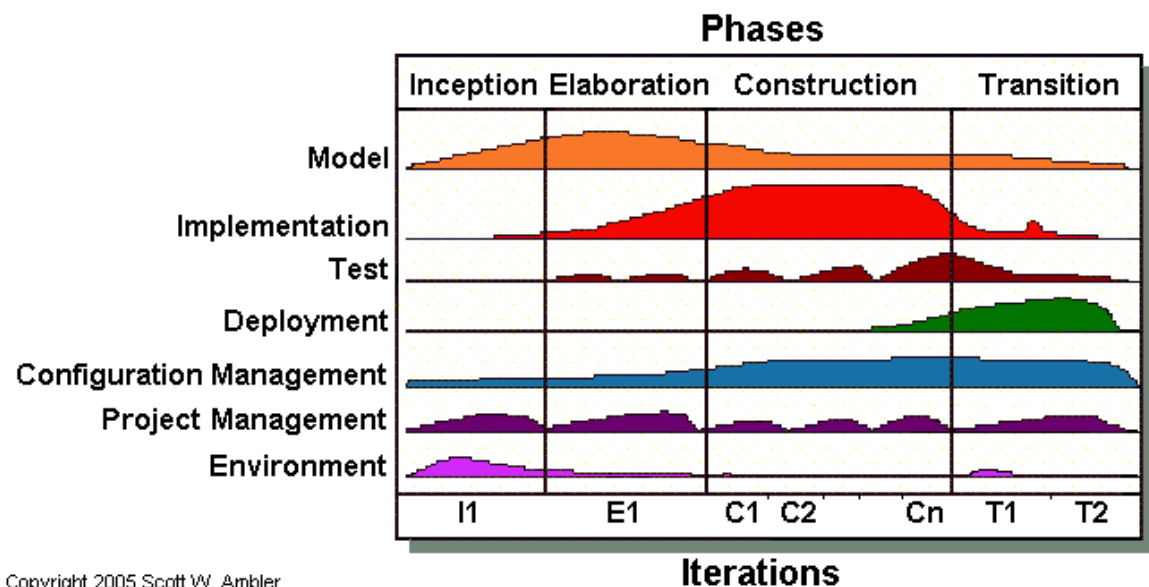
O POSCOMP *Coach* teve seu desenvolvimento guiado pelo Processo Unificado Ágil (Agile Unified Process), foi modelado com a Linguagem de Modelagem Unificada (Unified Modeling Language), implementado utilizando a linguagem de programação Python por intermédio do *framework* Django, HTML5, Javascript (ChartJS), CSS (Bootstrap), MySQL e L^AT_EX. Estes recursos são descritos nas subseções seguintes.

2.4.1 Agile Unified Process

Agile Unified Process (AUP) é uma versão ágil do *Rational Unified Process* (RUP)¹. Ele descreve uma abordagem de desenvolvimento de *software* simples de compreender e utiliza técnicas e conceitos ágeis, porém permanecendo compatível com o tradicional modelo RUP (AMBYSOFT, 2006).

O processo unificado ágil possui uma natureza "serial para o que é amplo" e "iterativa para o que é particular", isto é, uma sequência linear de atividades de engenharia de software (PRESSMAN, 2011) que se desdobra em quatro fases mais amplas, conforme pode ser observado na Figura 5.

Figura 5 – Ciclo de vida do AUP.



Fonte: <http://www.ambysoft.com/unifiedprocess/agileUP.html> (2016).

1. **Inception (Início)** - Que tem como objetivo identificar o escopo do projeto, uma potencial arquitetura para o sistema, e obter os investimentos necessários;

¹ Ver (GORNIK, 2003)

2. **Elaboration (Elaboração)** - Nesta fase o objetivo é validar a arquitetura do sistema;
3. **Construction (Construção)** - Aqui o objetivo é construir software funcional de forma gradual e incremental de forma que atenda as principais prioridades dos investidores;
4. **Transition (Transição)** - Finalmente, esta fase tem como objetivo validar e implantar o sistema em um ambiente de produção.

Também observado na Figura 5 o AUP é composto por um conjunto de disciplinas (*Model, Implementation, Deployment, Configuration Management, Project Management e Environment*). Disciplinas são as atividades realizadas em cada iteração de forma mais particular. "A equipe itera ou se repete para alcançar a agilidade e entregar incrementos de software significativos para usuários finais tão rapidamente quanto possível."(PRESSMAN, 2011). A seguir são apresentadas disciplinas e seus objetivos.

- **Model (Modelo)** - Compreender as regras de negócio do cliente, o domínio do problema sendo tratado pelo projeto, e identificação da solução mais adequada;
- **Implementation (Implementação)** - Transformar os modelos previamente construídos em código fonte executáveis e testáveis;
- **Test (Teste)** - Testar, validar e verificar se os requisitos foram atendidos;
- **Deployment (Implantação)** - Tornar o software disponível para os usuários;
- **Configuration Management (Configuração e Gerenciamento)** - Controlar o andamento geral do projeto, liberação de versões e possíveis mudanças ao longo do processo;
- **Project Management (Gerenciamento do Projeto)** - Gerenciamento dos riscos, atribuição de tarefas, acompanhamento das atividades e coordenação com pessoas e sistemas que não fazem parte do escopo do projeto, assegurando a entrega no prazo estipulado e com os recursos disponíveis alocados;
- **Environment (Ambiente)** - Assegurar que todos os recursos necessários, sejam de hardware, software e etc, estejam disponíveis e para o time quando requisitados.

Seguindo o modelo proposto pelo AUP com relação as disciplinas, e adaptando-a ao contexto do POSCOMP *Coach*, na fase de inicial (**Inception**), foram definidos o escopo geral do trabalho, regras de negócio e identificação do arcabouço tecnológico mais adequado, optando-se pelas tecnologias previamente citadas no início da [seção 2.4](#).

Na fase de elaboração (***Elaboration***), foi desenvolvido um conjunto de *softwares* utilitários para importação de dados das provas do POSCOMP e um protótipo da aplicação com a finalidade de validar a arquitetura da plataforma.

Com o modelo de arquitetura definido, deu-se início então a fase de construção (***Construction***) onde o desenvolvimento e testes da aplicação definitiva teve início de forma incremental e sistemática correspondendo às disciplinas de *Implementation* e *Test* respectivamente. Ainda nesta fase as disciplinas de *Configuration Management*, *Project Management* e *Environment*, não foram seguidas estritamente a risca. No caso da primeira, *Configuration Management*, os *releases* eram avaliados constantemente pelo proponente, com entregas pontuais às orientadoras acadêmicas deste trabalho a fim de validar o andamento. Já as disciplinas *Project Management* e *Environment* são mais apropriadas quando o desenvolvimento se dá em equipe, o que não é o caso deste trabalho.

Na fase final de desenvolvimento (***Transition***) foi preparado um servidor privado remoto (*Virtual Private System*), mantido pelo proponente com recursos próprios, para a implantação e diagnóstico da plataforma em ambiente controlado. Esta atividade corresponde a disciplina (*Deployment*) no AUP.

Durante o processo, ao longo de todas as fases do AUP, foram gerados alguns artefatos, que estão demonstrados na Tabela 5.

Tabela 5 – Artefatos gerados ao longo do processo.

Fase	Artefato
<i>Inception</i>	Definição do escopo geral do projeto; Descrição de casos de uso; Diagrama de casos de uso; Diagrama de classes; Modelo Entidade-Relacionamento (MER).
<i>Elaboration</i>	Descrição da arquitetura da plataforma
<i>Construction</i>	Código-Fonte e Testes (executados pelo desenvolvedor)
<i>Transition</i>	Plano de implantação

Fonte: Elaborada pelo Autor

Para fins deste trabalho, o AUP foi escolhido dentre outras alternativas pelas seguintes características: é um processo mais simples e ágil que outros dirigidos a planos, tal como o RUP. Visa o software como objetivo, e permite alterações incrementais, atendendo plenamente as necessidades deste trabalho.

Nesta seção foi demonstrado como o AUP foi aplicado no contexto de desenvolvimento do POSCOMP *Coach*. Na próxima subseção será abordado a utilização da UML no mesmo contexto.

2.4.2 Unified Modeling Language

A *Unified Modeling Language* (UML) é uma linguagem visual que ajuda na criação e desenvolvimento de projetos de sistema de *software* (FOWLER, 2014). Por definição A UML "[...] é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de software" (BOOCH; RUMBAUGH; JACOBSON, 2006).

A UML surgiu em meados de 1997 com o objetivo de unificar várias linguagens gráficas de modelagem orientadas a objeto que existiam na época, oferecendo um nível de abstração alto o que facilitava as discussões sobre o projeto. Atualmente, controlada pelo OMG² (*Object Management Group*), encontra-se na versão 2.5³ lançada em junho de 2015 (FOWLER, 2014; OMG, 2016).

A UML 2 é constituída por treze (13) digramas oficiais, conforme apresentados na Tabela 6. A UML através de seus diagramas se propõe a representar uma visão ou perspectiva de um sistema, estática ou dinâmica. No entanto, é importante destacar que, é normal que apenas alguns destes diagramas sejam de fato utilizados para formular esta representação (LARMAN, 2007).

Tabela 6 – Diagramas oficiais da UML.

Diagrama	Objetivo	Linhagem
Atividades	Comportamento procedimental e paralelo	Na UML 1
Classes	Classe, características e relacionamentos	Na UML 1
Comunicação	Interação entre objetos; ênfase nas ligações	Diagrama de colaboração da UML 1
Componentes	Estrutura e conexão de componentes	Na UML 1
Estruturas compostas	Decomposição de uma classe em tempo de execução	Novidades da UML 2
Distribuição	Distribuição de artefatos nos nós	Na UML 1
Visão geral da interação	Mistura de diagrama de sequência e de atividades	Novidade da UML 2
Objetos	Exemplo de configurações de instâncias	Extraoficialmente na UML 1
Pacotes	Estrutura hierárquica em tempo de compilação	Extraoficialmente na UML 1
Sequência	Interação entre objetos; ênfase na sequência	Na UML 1
Máquinas de estado	Como os eventos alteram um objeto no decorrer de sua vida	na UML 1
Sincronismo	Interação entre objetos; ênfase no sincronismo	Novidade da UML 2
Casos de uso	Como os usuários interagem com um sistema	Na UML 1

Fonte: Fowler (2014, p. 33).

² <http://www.omg.org>

³ <http://www.omg.org/spec/UML/2.5/>

No escopo deste trabalho, foram utilizados três diagramas para representar a visão sobre o POSCOMP *Coach*: diagrama de pacotes, de classes e de casos de uso, conforme serão abordados no [Capítulo 3](#) - POSCOMP *Coach*.

Nesta subseção foi abordado as características da UML e como se dará sua aplicação no contexto do POSCOMP *Coach*. Na próxima subseção será apresentada a linguagem de programação utilizada como base da plataforma web, Python.

2.4.3 Python

Python⁴ é uma linguagem de programação criada por Guido Van Rossum em 1991. Os objetivos iniciais eram obter máxima produtividade com alta legibilidade de código. Python é orientada a objetos, dinâmica e fortemente tipada. Ela é interpretada, mas também compilada, de forma que seu código fonte é compilado em formato binário (*bytecode*⁵) e então interpretado, mas isso ocorre de forma transparente para o usuário; não é preciso compilar de forma explícita o código fonte ([CASSELL; GAULD, 2014](#)).

A linguagem Python e sua marca são mantidos pela *Python Software Foundation*⁶ (PSF), uma instituição sem fins lucrativos que é responsável por proteger a plataforma Python. É ela quem determina se uma implementação de linguagem pode ser chamada de Python ou não ([Python Brasil, 2008](#)).

Pode-se dizer que Python é usada em diferentes contextos e aplicações. É utilizada, por exemplo, em escolas e universidades, muitas vezes como linguagem introdutória, pela sua facilidade de aprendizagem, chegando a figurar entre as linguagens mais populares no ensino introdutório de Ciência da Computação em instituições Norte-Americanas ([GUO, 2014](#)). Entretanto, Python não é utilizada apenas nos meios educacionais e acadêmicos. Grandes empresas como Google, NASA e Lucasfilm Ltd utilizam Python de forma profissional. Destacando o Google como exemplo, é possível filtrar uma quantidade de aproximadamente 150 repositórios de *software* mantidos e abertamente disponíveis na sua página do GitHub⁷ (Plataforma mundialmente conhecida de compartilhamento de código e repositórios de *software*).

Python é uma linguagem com uma expressividade sintática de altíssimo nível, o que potencialmente aumenta a produtividade. Isso ajuda a entender porquê foi a 4ª linguagem de programação mais utilizada em 2016 ([TIOBE, 2016](#)).

Python executa em diversas plataformas, evitando o *vendor lock-in*⁸. Sua licença *open source* permite o uso, distribuição e venda de suas aplicações desenvolvidas em Python.

⁴ <http://python.org>

⁵ <https://docs.python.org/3/glossary.html>

⁶ <https://www.python.org/psf/>

⁷ <https://github.com/Google?language=python>

⁸ Arquitetura dependente de produto ([Source Making, 2016](#))

Possui sintaxe simples, mas sofisticada, permitindo que se escreva código tanto de forma procedural, quanto orientado a objetos, de forma elegante e conforme a necessidade. Seu interpretador interativo permite testar e experimentar código em tempo real, reduzindo tempo com testes e aumentando a produtividade. O ecossistema Python possui uma enorme quantidade de bibliotecas e módulos disponíveis, e ainda Python *bindings* para diversos *toolkits* gráficos, agrupados em um repositório de módulos Python categorizados⁹ (OLIPHANT, 2007).

Um fator importante a se considerar em relação ao Python é sua comunidade de usuários, entusiastas e desenvolvedores. Considerada uma das mais receptivas e ativas dentre as linguagens de programação, e que além disso tem se dedicado por anos em melhorar os padrões, bibliotecas e práticas que ajudam na hora de construir softwares robustos e confiáveis em curtos espaços de tempo (JAWORSKI; ZIADE, 2016).

O uso do Python (versão 3.6.3) para este projeto foi definido pelos pontos fortes da linguagem, pelas características apresentadas, que atenderam plenamente ao desenvolvimento do POSCOMP *Coach*. Além disso, a afinidade e conhecimento prévio da linguagem pelo desenvolvedor também teve relevante influência nessa escolha.

Algumas informações importantes e complementares para este trabalho no que diz respeito às peculiaridades da linguagem Python em relação à outras linguagens tradicionais orientadas a objetos foram adicionadas no Apêndice F – Controle de acesso, publicidade e visibilidade de atributos em Python.

Na próxima subseção será apresentado o *framework* que juntamente com Python proporcionou a construção da plataforma web POSCOMP *Coach*.

2.4.4 Django

Django é um Framework Web de alto nível escrito em Python, que encoraja um desenvolvimento rápido com um *design* limpo e prático. Tendo sido construído por desenvolvedores experientes, ele abstrai muito dos problemas e dificuldades encontrados no desenvolvimento Web, de modo que o desenvolvedor possa focar na sua aplicação, sem ter que ficar se preocupando em "reinventar a roda".

O *framework* Django foi pensado para ajudar os desenvolvedores de aplicações desde a concepção até o seu desenvolvimento o mais rápido possível. É um *framework* bastante seguro e provê mecanismos para evitar os problemas mais comuns como SQL *injection*¹⁰, *cross-site scripting* e *cross-site forgery*¹¹.

⁹ <https://pypi.python.org/pypi>

¹⁰ Uma técnica de injeção de código SQL arbitrário em formulários de páginas web.

¹¹ São tipos de ataques que exploram vulnerabilidades causadas por falha nas validações de parâmetros de entrada do usuário e resposta do servidor na aplicação web.

Django é muito extensível e escalável, foi projetado para utilizar bem o *hardware*, em qualquer nível de arquitetura, o que significa que é possível adicionar novos servidores de dados, servidores de cache ou servidores de aplicações, tantos quanto forem necessários [Django Project \(2017\)](#).

Pode-se dizer que o Django segue um padrão de arquitetura *Model-View-Controller* (MVC), no entanto, é comum que desenvolvedores se refiram a ele como um *framework Model-Template-View* (MTV). Isso se dá pela escolha dos desenvolvedores em redefinir os nomes dos componentes de acordo com a organização e visão arquitetural do *framework*. Dessa forma temos que a camada *View* dá lugar à *Template* e a camada *Control* é substituída pela *View*, que em Django, efetivamente trata as requisições e descreve os dados que vão ser apresentados ao usuário na resposta repassando-os ao *Template*. Neste caso, como explicado pelos desenvolvedores em seu FAQ¹² o *Controller*, seria o próprio *framework* e ferramentas embutidas nele.

Por ser construído utilizando a linguagem Python, Django compartilha de muitos aspectos da filosofia Python citados anteriormente na [subseção 2.4.3](#). Dentre os principais fatores para escolher Django em detrimento de outros *frameworks* como Web2Py¹³ ou Flask¹⁴ pode-se destacar a maturidade do projeto, comunidade grande e bastante ativa de desenvolvedores, excelente documentação e código aberto.

Nesta subseção foi apresentado o Django, *framework* utilizado para construir, de fato, a plataforma web. Na próxima subseção será abordado o MySQL solução escolhida para persistência das informações de exames, provas e dados de usuários.

2.4.5 MySQL

MySQL é um Sistema Gerenciador de Banco de Dados (SGBD) relacional de código aberto, baseado na SQL (*Structured Query Language*), linguagem de pesquisa declarativa padrão para bancos de dados relacionais. Desenvolvido inicialmente pela empresa MySQL AB foi adquirido pela Sun Microsystems em 2008 e posteriormente comprado pela Oracle em 2010 que o mantém até os dias de hoje.

O MySQL é considerado um SGBD rápido e leve, o que ajudou na sua adoção por diversas plataformas de sites e sistemas gerenciadores de conteúdo (CMS - *Content Manager System*) como Joomla e Wordpress. Embora bastante utilizado na web devido a facilidade de implantação e manuseio o MySQL é um SGBD de propósito geral, sendo utilizado por grandes empresas de tecnologia. Escalabilidade, alta performance e disponibilidade, sistema de transações robusto e fácil gerenciamento são características presentes neste SGBD. ([MySQL, 2017](#))

¹² <https://docs.djangoproject.com/en/1.11/faq/general/>

¹³ <http://www.web2py.com>

¹⁴ <http://flask.pocoo.org>

A adoção do MySQL para este projeto se deu principalmente por ser um excelente SGBD de código aberto que possui suporte as mais diversas linguagens, farta documentação e referências na Web e com excelente suporte à linguagem Python e seus *frameworks*.

2.4.6 Git

Git é um sistema de controle de versão distribuído (CVS - *Control Version System*) de código aberto criado por Linus Torvalds¹⁵ em 2005. Foi projetado para lidar desde projetos pequenos à grandes projetos de forma rápida e eficiente (git-scm, 2017).

Embora existam diversos sistemas de controle de versão disponíveis, Git vem se popularizando cada vez mais em empresas e projetos de código aberto. Isso se dá por diversos fatores, dos quais destacam-se: agilidade com uma interface de linha de comando realmente intuitiva, a possibilidade de trabalhar *offline* por ser um sistema distribuído, reversão simples de erros, *commits* pequenos e significativos ao invés de um grande *commit* grande e confuso, *merge* extremamente eficiente e fluxo de trabalho adaptável ao desenvolvedor.

Pode-se dizer que o Git, hoje, é o padrão de fato para desenvolvimento colaborativo e controle de versão, e portanto, uma escolha natural para auxiliar no desenvolvimento projeto apresentado neste documento.

Encerrando esta subseção sobre o uso do Git para versionamento, foram apresentadas diversas tecnologias envolvidas na construção e desenvolvimento do POSCOMP *Coach*. Na próxima subseção será apresentado o uso do L^AT_EX e seu papel fundamental na viabilidade deste trabalho.

2.4.7 L^AT_EX

Alguns desafios foram encontrados durante o processo de desenvolvimento deste trabalho. Encontrar uma solução viável para extrair o conteúdo das provas do Exame do POSCOMP, e ainda como armazená-las na plataforma mantendo o seu conteúdo o mais fiel possível, foram alguns deles.

Como descrito anteriormente, os Exames do POSCOMP se encontram disponíveis em formato PDF, no site da SBC. O formato PDF é um formato utilizado para distribuição de arquivos com finalidade de visualização e impressão. Mecanismos de *parsing* desse tipo de arquivo existem, mas são limitados. Somando-se ainda o fato de que, como os Exames POSCOMP foram produzidos e aplicados por diferentes entidades ao longo dos anos, existem diferenças na forma como as questões estão organizadas, sem um padrão definido, dificultando ainda mais um processo completamente automatizado.

¹⁵ Também criador do Kernel Linux – <http://kernel.org>

Para resolver esse primeiro desafio, contou-se com o apoio de alunos da graduação em Engenharia de Computação da EST/UEA (Escola Superior de Tecnologia/ Universidade do Estado do Amazonas), que se prontificaram a transcrever as provas para um formato adequado, \LaTeX .

\LaTeX é um sistema de composição de documentos de alta qualidade criado por Leslie Lamport em 1985, que inclui funcionalidades pensadas para a produção de documentação técnica e científica. Ele utiliza o formatador \TeX ¹⁶ como engine de formatação. Pode-se dizer que \LaTeX é o padrão de fato para comunicação e publicação científica ([The LaTeX Project, 2017](#)). Um dos princípios fundamentais do \LaTeX é permitir que o autor foque mais no conteúdo do documento que está sendo escrito, e menos na formatação e aparência, evitando que um tempo precioso seja dispendido com procedimentos complicados, repetitivos e suscetíveis a erros que os processadores de textos tradicionais possuem.

Com o uso do \LaTeX foi possível também solucionar um outro desafio: a armazenagem no banco de dados. Como um arquivo `.tex` (extensão de arquivo \LaTeX) é essencialmente um arquivo de texto puro, foi possível desenvolver um *parser*¹⁷, responsável por ler os arquivos, tratá-los e encapsulá-los em uma estrutura conveniente para ser armazenado e utilizado pela plataforma.

Além do que foi exposto acima, é importante destacar que o \LaTeX foi utilizado também para a construção desse documento. Por se tratar de um trabalho acadêmico, nada mais natural do que utilizar o padrão recomendado para confecção deste tipo de documento.

Por tanto, a opção pelo uso do \LaTeX para este trabalho foi de fundamental importância, pois permitiu solucionar problemas reais, que poderiam impactar negativamente no desenvolvimento da plataforma.

Neste subseção foi apresentado o \LaTeX e sua importância no desenvolvimento do POSCOMP *Coach*. A próxima subseção será focada no Bootstrap, *framework* CSS para construção das interfaces e telas do POSCOMP *Coach*.

2.4.8 Bootstrap

Bootstrap é um *framework* JavaScript, HTML(*Hypertext Markup Language*) e CSS (*Cascading Style Sheets*) criado por Mark Otto e Jacob Thornton em meados de 2011. É considerado o mais popular para o desenvolvimento de aplicações web responsivas e alinhadas com a filosofia mobile. Seu objetivo é tornar o desenvolvimento *frontend* mais

¹⁶ \TeX é um software que foi inicialmente escrito por Donald E. Knuth em 1977, com seu anúncio para o público em 1982, que tinha a finalidade de compor textos com suporte a fórmulas matemáticas, e explorar o potencial dos equipamentos de impressão digitais que surgiam naquela época. ([OETIKER H. PARTL; SCHLEGL, 2016](#)).

¹⁷ Ferramenta de análise sintática que identifica e trata padrões em uma *string* ou texto.

ágil e fácil (SILVA, 2015).

O Bootstrap foi desenvolvido e otimizado para ser usado com marcação HTML5. Contém funcionalidades que facilitam o cumprimento das diretrizes de acessibilidade previstas pelo *World Wide Web Consortium* (W3C), que é uma comunidade internacional, onde seus membros são compostos de diversos órgãos e empresas com o intuito de desenvolver e manter os padrões na Web.

A opção pelo Bootstrap no desenvolvimento deste trabalho se deu, além das características previamente citadas, pela sua popularidade, disponibilidade de boa documentação e conhecimento prévio do desenvolvedor.

Na próxima subseção será apresentada a ferramenta utilizada para geração dos gráficos com informações estatísticas, ChartJS.

2.4.9 ChartJS

ChartJS é uma biblioteca JavaScript que auxilia na criação e apresentação de gráficos para web através do elemento HTML *canvas*. Os gráficos são gerados e customizados apenas CSS e JavaScript, e podem ser exibidos usando um dos vários tipos disponíveis de gráficos como Linear, Barra, Radar, Área polar, Pizza ou *Doughnut*, Bolha e Escala.

No POSCOMP *Coach*, inicialmente foram utilizados basicamente os tipo Pizza ou *Doughnut*, como pode ser observado na Figura 32 do Apêndice D – Interfaces do Usuário, onde está representado as estatísticas de erros e acertos de questões por área do conhecimento.

Neste capítulo foram apresentados os fundamentos teóricos nos quais este trabalho se embasou. Além destes, também foram apresentadas os trabalhos relacionados e as tecnologias que foram utilizadas no desenvolvimento do POSCOMP *Coach*. No Capítulo 3 será abordado a elaboração do POSCOMP *Coach*, visão geral do projeto, diagramas e modelos e demais artefatos.

3 POSCOMP *Coach*

Neste capítulo são apresentadas as características e funcionalidades do POSCOMP *Coach*, com detalhamento dos diagramas e demais artefatos que compõe a fase de análise e modelagem da plataforma. Como resultado do processo de desenvolvimento ágil, com entregas contínuas ao longo das interações, diversos aspectos da plataforma já foram implementados, e serão abordados também neste capítulo.

3.1 Visão Geral

Conforme descrito anteriormente, o POSCOMP *Coach* é uma plataforma web cujo objetivo é prover auxílio a candidatos de Pós-Graduação em Computação na preparação para o exame do POSCOMP, fornecendo suporte no planejamento e gerenciamento de seus estudos por meio da criação de simulados com base nas questões de provas anteriores, avaliação de desempenho e histórico de estudos.

Esta plataforma foi desenvolvida de forma iterativa e incremental, considerando o Processo Unificado Ágil ([LARMAN, 2007](#)). De acordo com este processo, houve quatro fases principais: (1) *Concepção*, na qual foi construída uma visão comum e o escopo básico do projeto; (2) *Elaboração*, na qual foi construída uma visão mais refinada do sistema, foram elucidados a maioria dos requisitos, e houve a implementação iterativa da arquitetura central; (3) *Construção*, em que houve a implementação iterativa das funcionalidades, execução de testes e preparação para a implantação; e (4) *Transição*, na qual houve a realização de testes beta e a posterior implantação.

O POSCOMP *Coach* foi desenvolvido em Python utilizando o *framework* para desenvolvimento web Django, utiliza o MySQL como gerenciador de banco de dados e o Apache como servidor web. Embora estas tenham sido as tecnologias principais para o desenvolvimento, outras tecnologias também foram adotadas, a citar: Git, para controle de versionamento; Bootstrap, para desenvolvimento de projetos web responsivos; e, Chartjs, para plotagem de gráficos, neste caso para demonstrar dados sobre o desempenho dos candidatos que utilizam a plataforma. Testes unitários foram utilizados constantemente ao longo do desenvolvimento da aplicação.

Nas próximas seções serão detalhados aspectos da construção do banco de questões, das funcionalidades disponíveis e uma avaliação comparativa com outras soluções similares existentes na literatura.

3.2 Construção do Banco de Questões

O desenvolvimento de uma aplicação mais complexa frequentemente demanda a criação de *softwares* auxiliares, que ajudam no processo de desenvolvimento, testes e *deploy*. Ao longo deste projeto foi necessário o desenvolvimento de um conjunto de utilitários, na forma de módulos python para viabilizar a manipulação dos dados e realização de testes, sendo eles: *parsetex.py*, *utils.py*, *exporter.py* e *dummydata.py*. A função de cada um deles no processo será demonstrada ao longo deste capítulo.

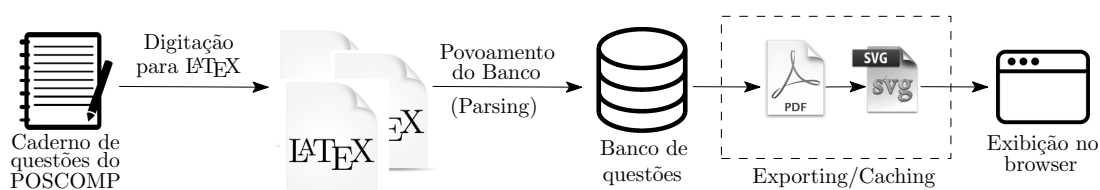
Um dos principais requisitos para que o POSCOMP *Coach* fosse caracterizado como uma plataforma capaz de permitir o bom preparo dos candidatos era a proximidade em termos de similaridade com as provas reais empregadas no POSCOMP. Além disso, um número substancial de questões também era necessário, para garantir que todos os assuntos estariam cobertos e que a diversidade estimulasse os candidatos, evitando a resolução de uma mesma questão repetidas vezes.

Considerando os requisitos previamente expostos, o desafio inicial portanto consistiu em construir uma base de questões levando-se em conta dois principais aspectos, qualitativo e quantitativo. Decidiu-se, então, utilizar as questões dos exames anteriores (de 2002 a 2017) em um total de 1120 questões. Esta proposta atendia perfeitamente o objetivo, porém existia um problema prático que precisaria ser tratado: todos essas questões estavam disponíveis apenas em formato PDF (*Portable Document Format*), o que por si só já dificultaria em muito o trabalho. Somado a isso, tinha o agravante de que os cadernos de questões não seguiam um padrão, nem de formato, nem de número de questões por assunto. Os meios investigados para exportação ou conversão desses formatos acabavam resultando em perda de formatação, não reconhecimento de caracteres especiais, utilizados principalmente na composição de fórmulas matemáticas.

A estratégia adotada para viabilizar a construção desse banco de questões, incluindo a importação de provas futuras, foi segmentada em alguns passos, descritos como uma abstração de alto nível na Figura 6. Inicialmente foi realizada uma força tarefa envolvendo dez alunos da graduação em Engenharia de Computação da EST/UEA (Escola Superior de Tecnologia/ Universidade do Estado do Amazonas) para transcrição das questões dos exames anteriores do POSCOMP para a sintaxe do \LaTeX . Com esta medida, garantiu-se a preservação de todos os elementos gráficos e fórmulas matemáticas contido nas questões, além de manter um padrão de apresentação e formato das questões compatível com o do exame original, isto é, contendo os textos que motivam ou compõe a situação-problema e que, muitas vezes, incluem elementos gráficos, como tabelas, figuras, fórmulas, diagramas, dentre outros.

Uma vez finalizado o processo de transcrição, era necessário a importação dos dados para um SGBD (Sistema Gerenciador de Banco de Dados), no caso deste trabalho

Figura 6 – Composição do Banco de Questões.



Fonte: Elaborada pelo autor.

o MySQL, de forma a permitir o uso das informações pela aplicação. Para este passo, foi desenvolvido um módulo utilitário com a função de *parser*, denominado *parsetex.py*, responsável por varrer os arquivos *.tex (LaTeX) previamente digitados e identificar os metadados da questão, como ano do exame, número da questão, assunto, enunciado, alternativas e gabarito. Os dados são então retornados pelo *parser* na forma de uma estrutura de dados Python denominada dicionário. A Figura 7 ilustra o exemplo de uma estrutura do tipo dicionário Python retornada através do método *get_question* do módulo *parsetex.py*.

O passo seguinte foi a persistência dos dados gerados pelo *parser* no banco de dados. Neste passo um segundo módulo Python desenvolvido para este trabalho denominado *utils.py* persiste no banco o Exame e seu gabarito através dos métodos *persist_exam* e *persist_answerkeys*, respectivamente. Esse processo pode ser visualizado mais explicitamente na Figura 8.

Nesta fase do processo, as informações já estão no banco (vide Figura 11), mas precisam ser preparadas para a exibição no *browser* do usuário. Para uma melhor experiência do usuário, o formato adotado para exibição da questão e alternativas foi o SVG (*Scalable Vector Graphics*). A grande vantagem da exportação de conteúdo LaTeX para SVG em detrimento do PNG é a escalabilidade visual do gráfico, isto por que o SVG é um formato vetorial, e portanto, não possui perda de qualidade quando redimensionado na tela do *browser*, seja ele *desktop* ou *mobile*. Essa característica é evidenciada no exemplo da Figura 9 e demonstrada na Figura 10 na forma de um fragmento de questão renderizada pelo POSCOMP Coach, onde temos a mesma questão representada pelos dois formatos, e aumentada com *zoom*.

Neste momento tem-se uma plataforma funcional, porém a ausência de dados de entrada de usuários para testes iniciais é um problema, uma vez que realizar os simulados exige um tempo considerável. Para contornar essa situação foi desenvolvido o módulo *dummydata.py*. A função desse módulo é popular exames simulados criados por usuários, recebendo como parâmetro a identificação do simulado, e o índice de acerto, de forma que as funcionalidades, como por exemplo os gráficos de análises possam ser verificados.

Alguns testes foram realizados com a geração dos arquivos **.svg** sob demanda, em

Figura 7 – Questão 02/POSCOMP 2011 - Formato de Dicionário *Python*.

```

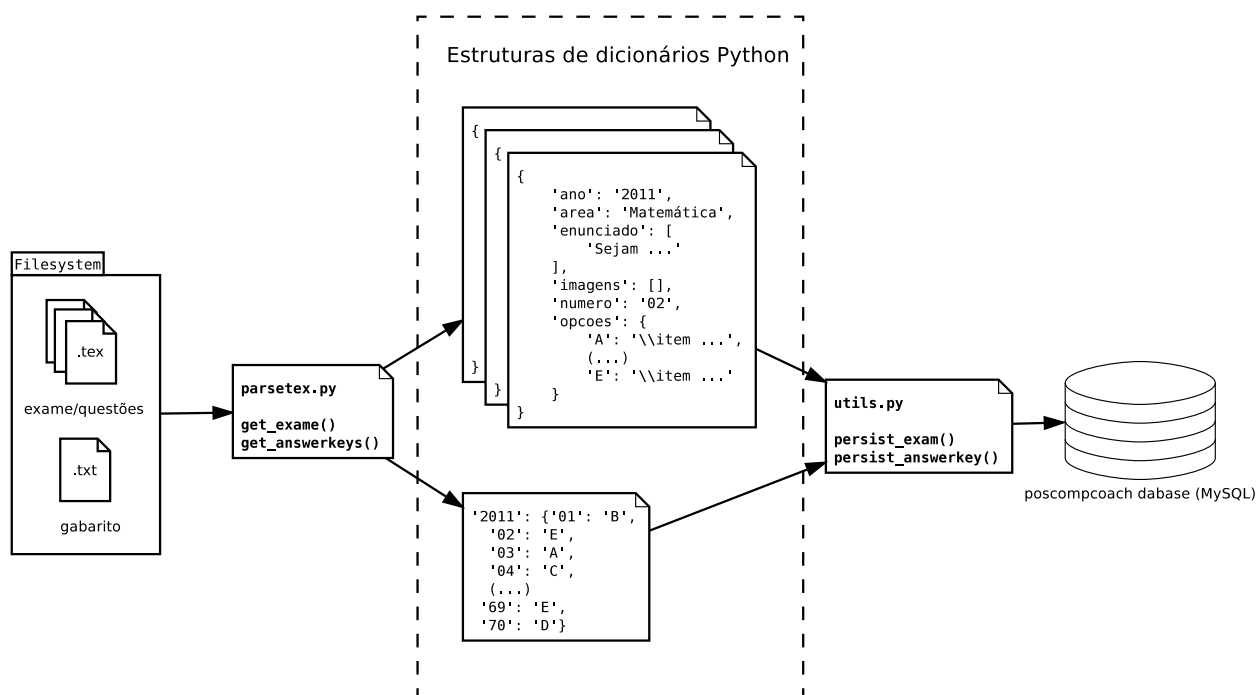
{
    'ano': '2011',
    'area': 'Matemática',
    'enunciado': [
        'Sejam  $a$  e  $b$  números reais não nulas.
        As duas retas perpendiculares à reta  $\frac{x}{a} + \frac{y}{b} = 1$ 
        que formam triângulos de área  $|ab|$  com os eixos
        ordenados são descritas pelas equações:\n'
    ],
    'imagens': [],
    'numero': '02',
    'opcoes': {
        'A': '\item  $ax - by = 1$  e  $-ax + by = 1$ \n',
        'B': '\item  $\frac{x}{a} - \frac{y}{b} = 1$  e  $\frac{x}{a} + \frac{y}{b} = 1$ \n',
        'C': '\item  $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$  e  $\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$ \n',
        'D': '\item  $\frac{x}{b} - \frac{y}{a} = \sqrt{2}$  e  $\frac{x}{b} + \frac{y}{a} = \sqrt{2}$ \n',
        'E': '\item  $\frac{x}{|a|} + \frac{y}{|b|} = \sqrt{2}$  e  $\frac{x}{|a|} - \frac{y}{|b|} = -\sqrt{2}$ \n'
    }
}

```

Fonte: Elaborada pelo autor.

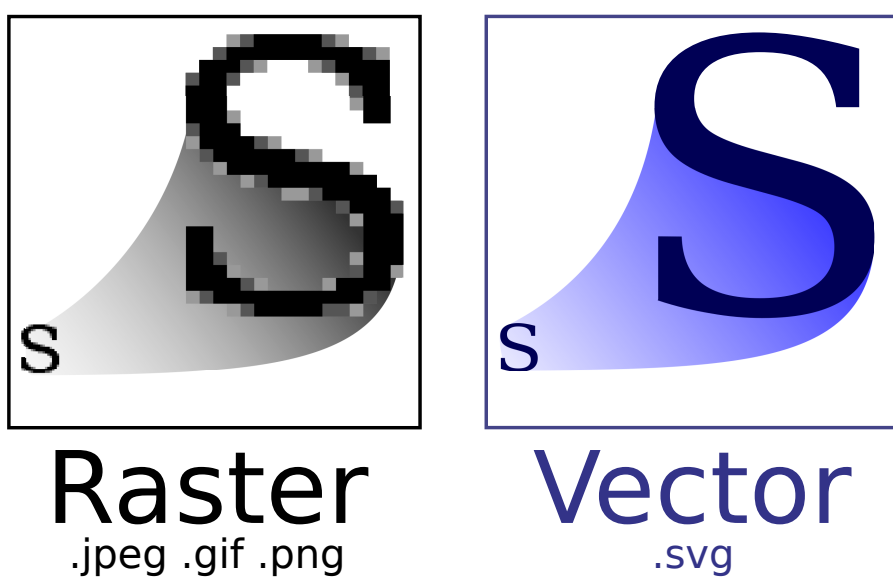
tempo de execução, porém esta abordagem não se mostrou eficiente, porque o processo de geração dos arquivos .svg se dá em dois passos distintos, sendo o primeiro de .tex para .pdf, (vide Figura 6), em seguida de .pdf para .svg, e o tempo despendido neste processo de conversão não foi satisfatório. Para solucionar esse problema e maximizar os recursos de *hardware*, bem como melhorar a experiência do usuário, optou-se pela geração prévia de um *cache* das questões, de modo que a reconstrução fosse garantida apenas quando a questão fosse atualizada ou corrigida pelo administrador. Um terceiro utilitário desenvolvido para este trabalho denominado *exporter.py* é o responsável por realizar estas conversões e geração do cache. Este utilitário é chamado previamente para geração do cache inicial com toda a base de questões, e também em tempo de execução pelo Django no momento da atualização das questões e alternativas, individualmente, por parte do administrador. Ao final, essa abordagem com uso do *cache* permitiu uma navegação mais fluida entre as questões durante a composição de simulados pelos usuários.

Figura 8 – Fluxo dos dados até a persistência.



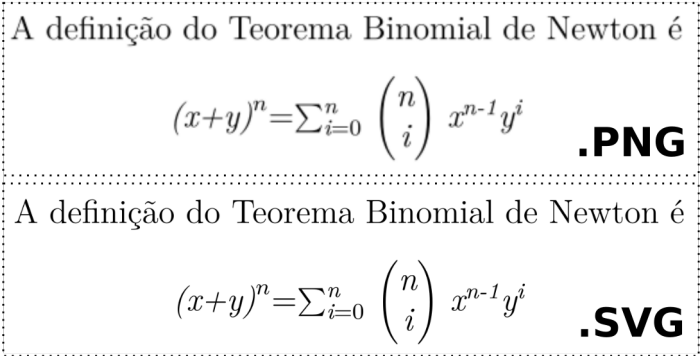
Fonte: Elaborada pelo autor.

Figura 9 – PNG vs SVG - Exemplo.



Fonte: Elaborada pelo autor.

Figura 10 – PNG vs SVG - Comparativo em detalhe no POSCOMP Coach.



Fonte: Elaborada pelo autor.

Figura 11 – Itens já persistidos pelo módulo *utils.py*.

Query 2 ✕

Limit to 1000 rows

```
1 select * from poscomp_coach.base_item;
```

#	id	number	statement	area_id	exam_id
2	2946	32	Quais dos algoritmos de ordenação de ordenação abaixo possuem tempo no pior caso e tempo médio de execução proporcional ...	5	46
3	2947	24	Considere o projeto de um circuito digital que implementa a função \$\$\$ com três variáveis de entrada e satisfazendo as seguintes ... $f(x, y, z) = \begin{cases} 1 & \text{se } x \neq y \wedge \\ 0 & \text{caso c...} \end{cases}$	5	46
4	2948	45	Dentre as características do modelo relacional e do modelo de objetos em bancos de dados, qual afirmação é INCORRETA?	6	46
5	2949	53	Considere uma cena representada no sistema de referência do universo (SRU), uma window definida pelo par de coordenadas (0,...	6	46
6	2950	31	Considere o algoritmo de busca sequencial de um elemento em um conjunto com \$\$\$ elementos. A expressão que representa o te...	5	46
7	2951	49	Considerando a rede de Petri abaixo, quais das alternativas são verdadeiras? $\begin{cases} \text{item } \text{O lugar A está habilitado a disparar.} \\ \text{item } \text{Apenas a transição T1 está habilitada a disparar.} \\ \text{item } \text{A sequência de transições (T1, T2, T3, T...} \end{cases}$	6	46
8	2952	42	Dentre as definições a seguir, ligadas ao conceito de normalização do modelo relacional, qual delas é INCORRETA?	6	46

Fonte: Elaborada pelo autor.

3.3 Modelo Conceitual

Um modelo conceitual é essencialmente a representação de conceitos em um domínio do problema. Pode-se dizer também que é uma descrição de coisas do mundo real do domínio do problema. Nele não são considerados detalhes de implementação ou tecnologias utilizadas, ou seja, enfatiza-se fortemente conceitos do domínio e não entidades de software (LARMAN, 2002).

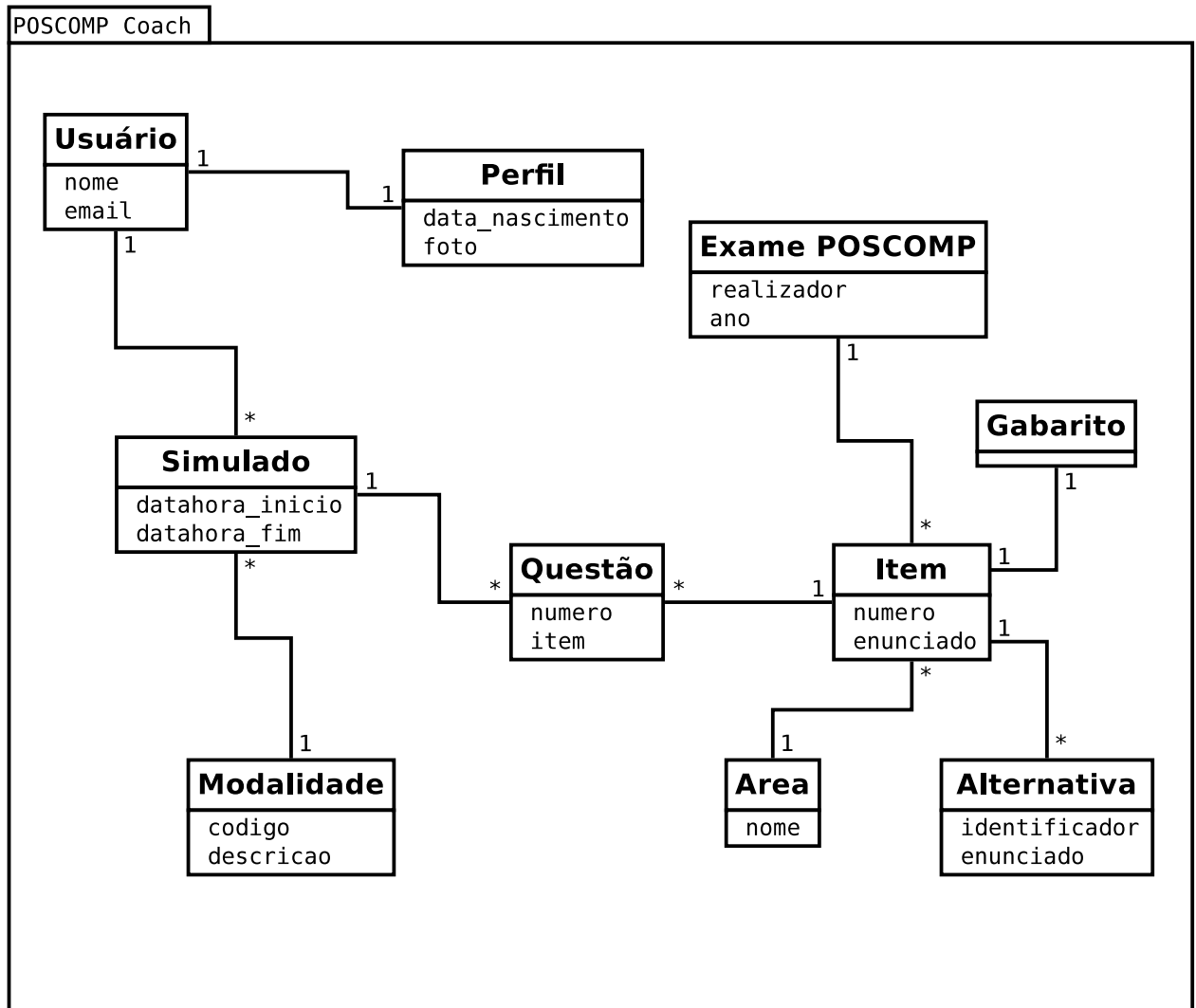
Na Figura 12 é possível observar como esse modelo se aplica ao domínio do POSCOMP Coach. A entidade **Usuário** representa os atores envolvidos na utilização da plataforma, que podem ser do tipo *user*, basicamente o candidato que se registra na plataforma para utilizá-la no seu preparatório, ou *staff*, administrador da plataforma, com privilégios especiais como a edição das questões importadas e configurações gerais da plataforma. Os usuários cadastrados são associados à uma entidade **Perfil** em uma relação de um para um, onde esta entidade tem a responsabilidade de manipular dados adicionais de um usuário, tais como foto, data de nascimento e instituição de ensino.

O conjunto de entidades **Exame**, **Item**, **Alternativa**, **Area** e **Gabarito** representam o exame original POSCOMP processado e importado para a plataforma. Essas entidades formam a base da plataforma, de onde serão extraídos os dados para configuração dos simulados pelos usuários.

As entidades **Simulado**, **Questão** e **Modalidade** representam o simulado configurado pelo usuário. Tomando como base os exames previamente cadastrado os usuários têm a opção de realizar o exame original com base no ano em que ocorreu, ou personalizar uma prova definindo alguns parâmetros como área de conhecimento, número de questões e tempo de prova.

Uma observação importante com relação ao modelo conceitual apresentado na Figura 12 é que os nomes das entidades foram escritos em português, para melhor entendimento e por se tratar de uma abstração de mais alto nível, porém na implementação a nomenclatura das classes e estrutura de codificação estão em inglês, conforme mostrado mais adiante na Seção 3.6 – Diagrama de classes.

Figura 12 – Modelo Conceitual - POSCOMP Coach.



Fonte: Elaborado pelo autor.

3.4 Detalhamento da plataforma

Tomando como base o modelo conceitual, pode-se ter uma ideia geral da arquitetura do *software* e como ele foi projetado. Neste contexto, a plataforma é fundamentalmente composta de quatro módulos, **account**, **base**, **mock** e **stats**. Cada um desses módulos tem uma atribuição importante e específica na plataforma conforme detalhamento a seguir.

O módulo **account**, é responsável por gerenciar o registro de novos usuários, alteração de perfis, restrições, permissões e redefinição de senha. O suporte base para estas funcionalidades é fornecido pelo módulo *admin* do Django Framework.

No módulo **base** estão todas as informações dos exames anteriores do POSCOMP que foram importadas para a plataforma e estão disponíveis para configuração de novos exames simulados por parte do usuário. Aos administradores é permitido a edição das questões para fins de correção apenas, já que o exame deve refletir integralmente a prova

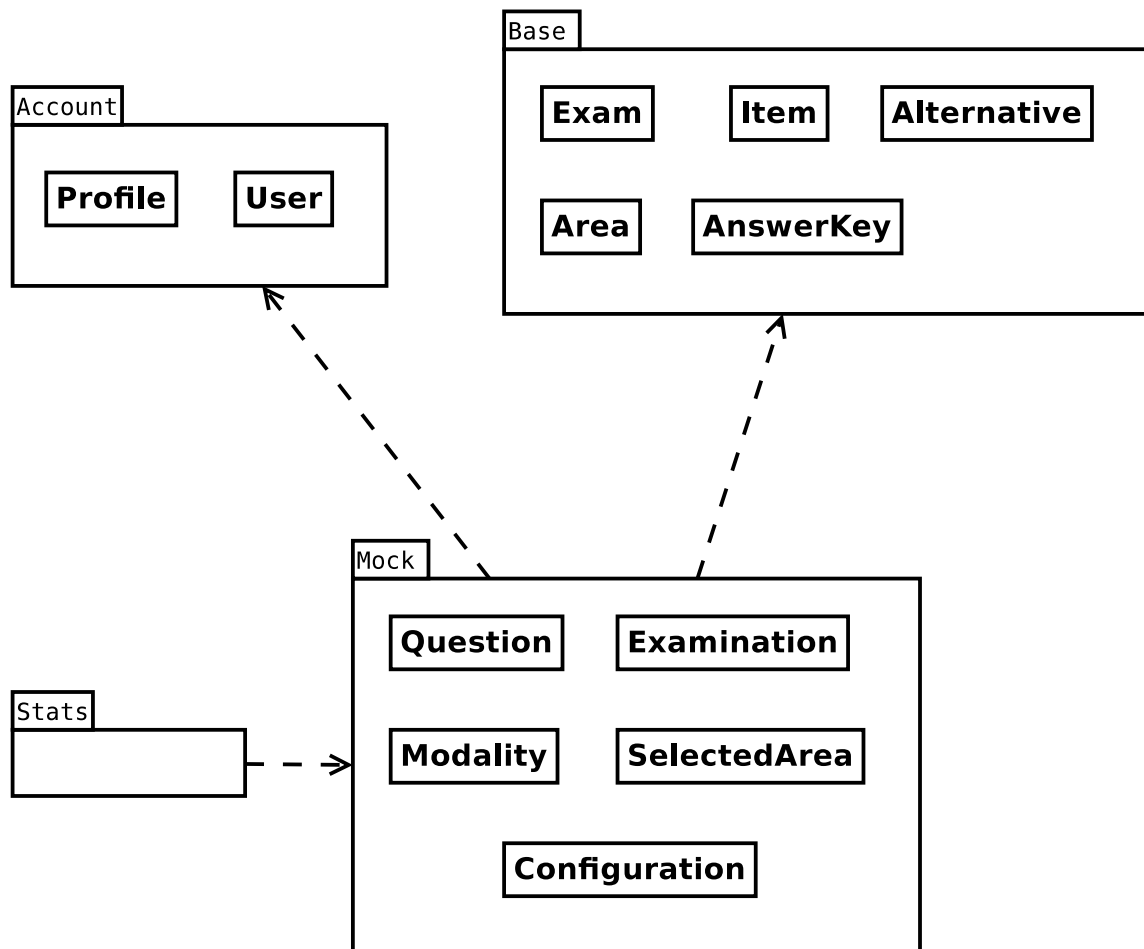
aplicada no POSCOMP no respectivo ano.

O módulo **mock** é a parte mais importante da plataforma, pois é ele que possibilita a configuração e realização do simulados por parte dos usuários. Um simulado configurado toma como base todo o banco de questões, e mais as opções de customização do usuário, que pode escolher entre as três modalidades disponíveis, padrão, por área de conhecimento e personalizado, além de tempo de duração da prova.

O quarto módulo, denominado **stats** é responsável por gerar gráficos para análise de desempenho e dados estatísticos, tendo como base os resultados obtidos pelo usuário em seus exames simulados realizados.

A sumarização desse módulos e suas relações de interdependência são apresentadas no Diagrama de Pacotes, conforme é ilustrado na [Figura 13](#).

Figura 13 – Diagrama de Pacotes - POSCOMP Coach.



Fonte: Elaborado pelo autor.

3.5 Diagramas de casos de uso

Ao passo em que *Casos de Uso* são considerados documentos narrativos que descrevem sequência de eventos realizadas por um ator em um processo durante a utilização de um sistema, um *Diagrama de Caso de Uso* ilustra a relação entre esses atores e as ações executadas por eles, utilizando-se de notações específicas da UML (LARMAN, 2002).

A dinâmica da plataforma consiste nos seguintes passos, o candidato tem interesse em se preparar para o POSCOMP, visando ingressar em um Programa de Pós-Graduação em Computação. Para ter acesso aos recursos do POSCOMP *Coach* ele precisa antes de tudo, se *registrar na plataforma*. Para isto ele acessa o endereço eletrônico da plataforma, e realiza seu cadastro utilizando uma das redes sociais sugeridas, Facebook e Google ou, caso não possua nenhuma destas, faz um cadastro por um formulário de registro próprio disponível pela plataforma.

Uma vez cadastrado é o momento de o candidato *logar na plataforma*, onde serão apresentadas em um *dashboard* as instruções básicas do uso e dos recursos disponíveis. Já logado, o candidato pode dar início aos estudos, começando por *preparar um simulado* e escolher a modalidade dentre as três disponíveis (Padrão POSCOMP, Por Área e Personalizada). Após a preparação do simulado, o candidato pode optar por *realizar o simulado* imediatamente ou posteriormente de acordo com sua conveniência.

Quando houver ao menos um simulado finalizado o candidato poderá *listar os simulados* pendentes ou já realizados, *visualizar o gabarito* dos simulados finalizados e ainda, *visualizar o seu desempenho* individual através de gráficos dinâmicos que lhe permitirão acompanhar o seu rendimento ao longo do tempo. O candidato tem ainda a possibilidade de revisar um simulado já finalizado.

O candidato poderá *editar seu perfil*, alterar informações pessoais, e caso necessário poderá *recuperar sua senha* através do formulário de recuperação de senha.

Em um contexto mais restrito temos o administrador, que é responsável por manter a plataforma. Tendo em vista que o banco de questões é formado exclusivamente por exames realizados do POSCOMP não é permitida alteração de conteúdo dos itens, mas correções de falhas que possam ter ocorrido na importação, ou pequenos erros de digitação do conteúdo original podem ser realizados pelo administrador.

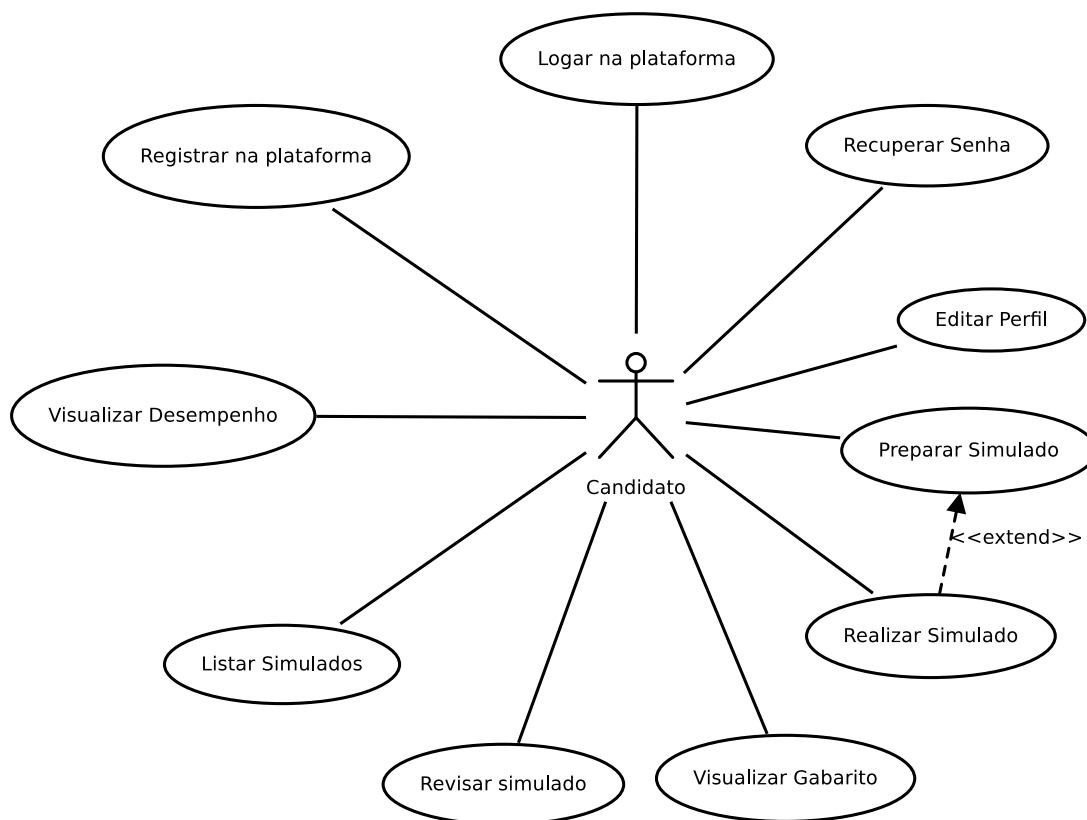
Desta forma, o administrador responsável por manter a plataforma, terá, além de todas as funcionalidades que o candidato possui, acesso às funcionalidades administrativas, bem como acesso aos dados estatísticos dos candidatos cadastrados. O administrador poderá *pesquisar itens* importados, e caso seja necessário alguma correção ele poderá *editar um item* ou *editar alternativas*. Além disso, poderá *listar exames* importados e *manter usuários*, sendo esta última uma funcionalidade provida pelo próprio *framework* Django. Já os dados estatísticos poderão ser utilizados para análise de desempenho coletivo visando

melhorias na plataforma.

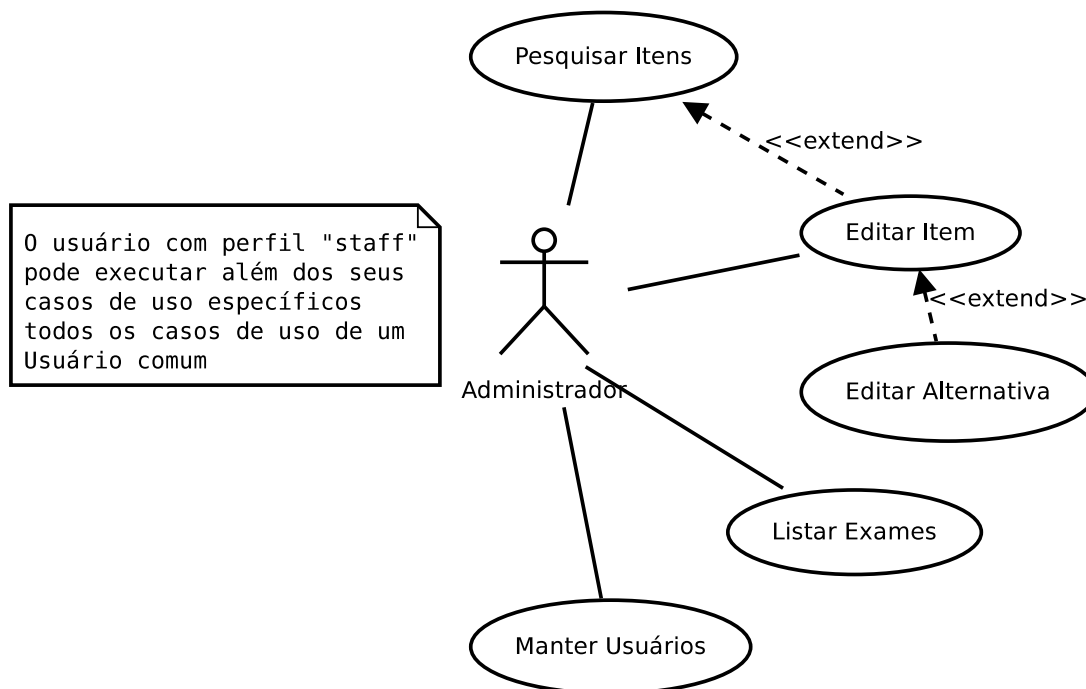
Nas Figuras 14 e 15 estão representados nos diagramas as relações entre os atores (Candidato e Administrador) e suas ações no sistema. As descrições de todos os casos de uso estão disponível nos apêndices, conforme descrito a seguir:

- Apêndice A – Especificação de Casos de Uso: Registro e Autenticação;
- Apêndice B – Especificação de Casos de Uso: Visão do Candidato;
- Apêndice C – Especificação de Casos de Uso: Visão do Administrador.

Figura 14 – Caso de Uso - POSCOMP *Coach*: Usuário.



Fonte: Elaborado pelo autor.

Figura 15 – Caso de Uso - POSCOMP *Coach*: Administrador.

Fonte: Elaborado pelo autor.

3.6 Diagrama de classes

O *Diagrama de Classes* modela definições de recursos essenciais à operação do sistema. Recursos que representam pessoas, materiais, informações e comportamentos. É ponto de partida para a geração de código e o destino para engenharia reversa. Está no núcleo do processo de modelagem de objetos (PENDER, 2003).

Aqui cabe uma observação com relação a apresentação do Diagrama de Classes a seguir. Conforme citado na subseção 2.4.4 – Django, o *framework* Django compartilha de muitos aspectos da filosofia Python. Estas peculiaridades refletem nos artefatos gerados, como por exemplo os atributos públicos nas classes de modelo representadas nos diagramas de classe, que a princípio podem parecer inadequados quando comparados a representação de projetos em outras linguagens de programação, porém, esclarece-se que a representação está correta, e tem como base a documentação oficial do *framework* (Django Documentation, 2017). Para maiores informações recomenda-se a leitura do Apêndice F – Controle de acesso, publicidade e visibilidade de atributos em Python.

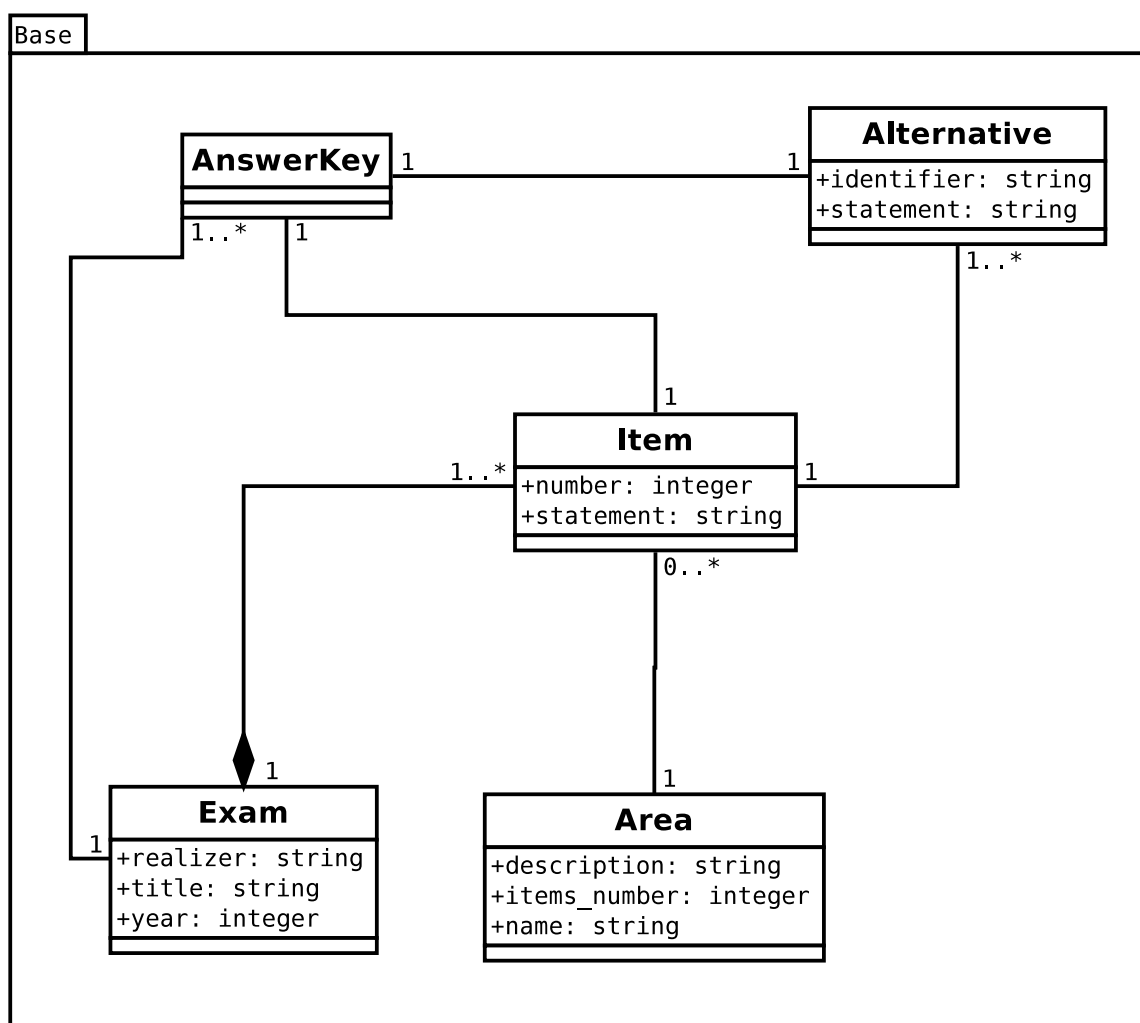
Com base no modelo conceitual ilustrado na Figura 12, apresentado na Seção 3.3 – Modelo Conceitual são apresentados nas Figuras 16, 17 e 18 a representação do Diagrama de Classes dos módulos *base*, *mock* e *account*, que compõe a base do POSCOMP *Coach*.

A semântica das classes, suas atribuições e seu papel na plataforma seguem a mesma dos modelos conceituais apresentados anteriormente, conforme mostrado no Quadro 1.

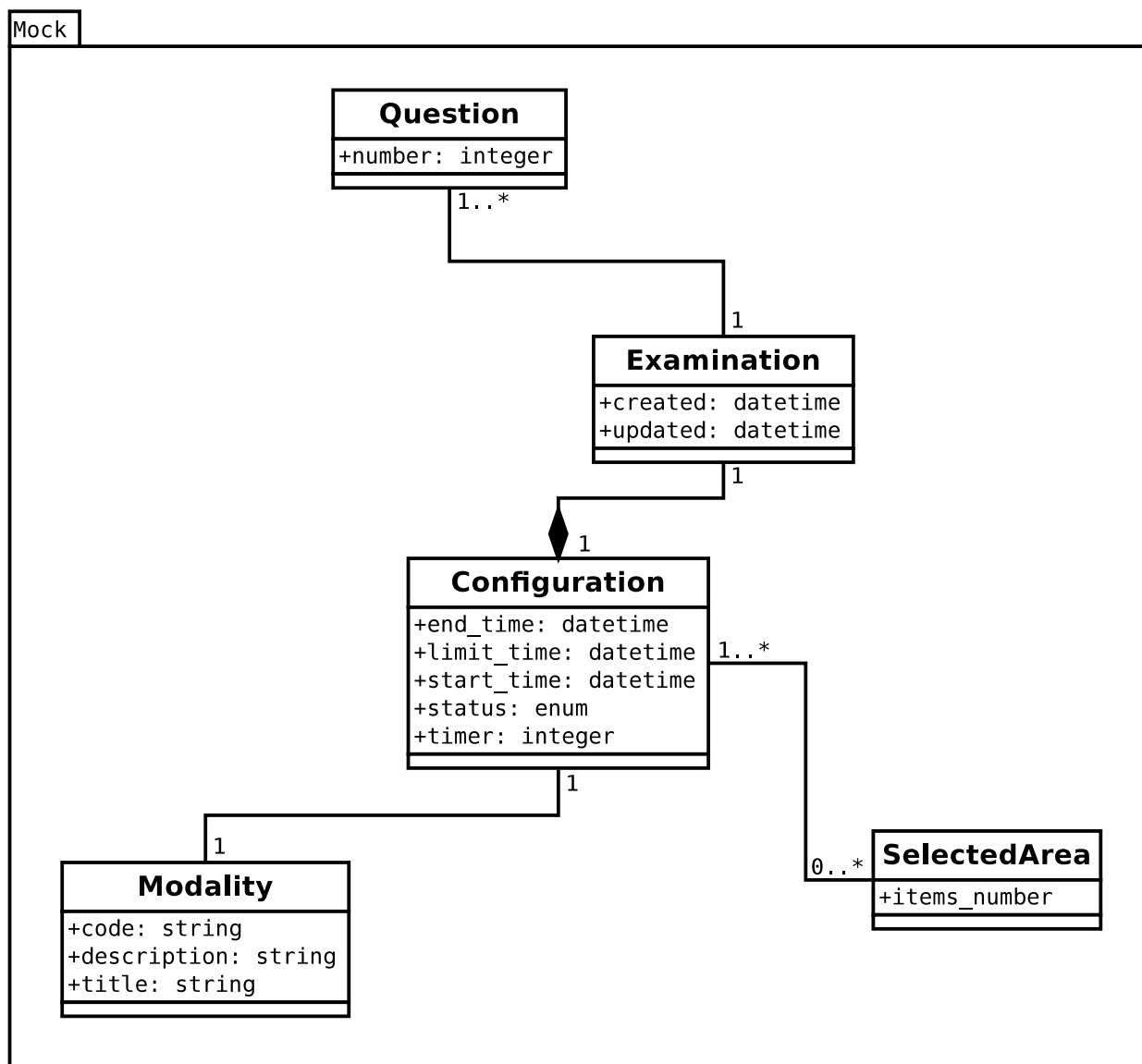
Quadro 1 – Correlação entre entidades do Modelo Conceitual e Diagrama de Classes.

Conceito	Classe	Papel
Usuário	<i>User</i>	Representa um usuário que interage com o sistema. Fornecida pelo <i>framework</i> . Esta entidade deverá estar associada a um perfil, responsável por armazenar informações adicionais do usuário.
Perfil	<i>Profile</i>	Representa um perfil associado ao Usuário. Nesta classe estão vinculados os dados adicionais de um usuário como foto, data de nascimento, e outros.
Exame	<i>Exam</i>	Representa o Exame POSCOMP original persistido no banco com seus metadados.
Item	<i>Item</i>	Esta entidade representa um Item, ou seja, uma das setenta (70) questões que compõe um Exame POSCOMP.
Alternativa	<i>Alternative</i>	Representa uma das cinco (5) alternativas que compõe um Item.
Area	<i>Area</i>	Representa a Área do conhecimento à qual pertence um Item, podendo ser atualmente: Matemática, Fundamentos da Computação e Tecnologia da Computação.
Gabarito	<i>AnswerKey</i>	Resposta correta, gabarito de um Item.
Simulado	<i>Examination</i>	Simulado preparado pelo usuário com base nos Items de Exames originais POSCOMP presentes no banco.
Questão	<i>Question</i>	Questão de um Simulado. Referencia diretamente um Item de um Exame original POSCOMP.
Modalidade	<i>Modality</i>	Representa as modalidades de Simulado disponíveis. Atualmente: Padrão POSCOMP, Por Área do Conhecimento e Personalizado.

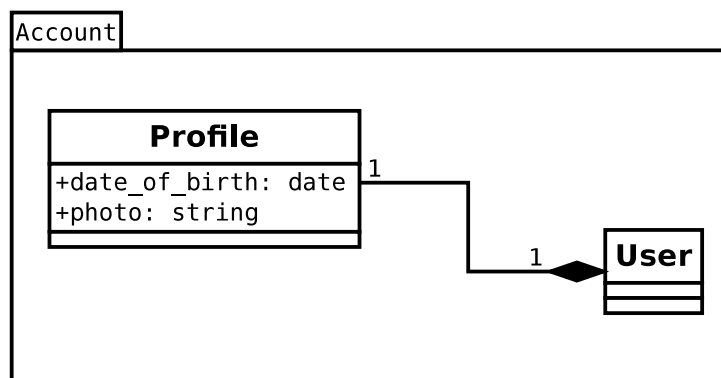
Fonte: Elaborada pelo Autor.

Figura 16 – Diagrama de domínio - POSCOMP *Coach*: Módulo *base*.

Fonte: Elaborado pelo autor.

Figura 17 – Diagrama de domínio - POSCOMP *Coach*: Módulo *mock*.

Fonte: Elaborado pelo autor.

Figura 18 – Diagrama de domínio - POSCOMP *Coach*: Módulo *account*.

Fonte: Elaborado pelo autor.

3.7 Modelo Entidade Relacionamento - MER

O Diagrama entidade-relacionamento, ou Modelo Entidade Relacionamento (MER) é a notação fundamental para a modelagem de dados conceitual. Sendo que seu principal propósito é representar os objetos de dados e suas relações, onde cada entidade termina representada por pelo menos uma tabela de dados, e normalmente expressa um Depósito de Dados (REZENDE, 2005).

Na Figura 19 é mostrado o Modelo Entidade Relacionamento e os relacionamentos que compõe o banco de dados do POSCOMP *Coach*. Analogamente ao comparativo entre Modelo Conceitual e Diagrama de Classes, o MER também possui semântica bastante similar, conforme mostrado no Quadro 2, porém com algumas particularidades que serão descritas a seguir.

O primeiro aspecto que é possível notar neste comparativo é que os nomes das tabelas são prefixados com o nome do módulo ao qual ela pertence, por exemplo: *base_exam*, corresponde à entidade *Exam* do módulo *base*, e assim por diante. Essa nomenclatura é padrão na construção de aplicações web baseadas no *framework* Django, e ajuda na organização e manutenção do banco de dados. Um segundo aspecto é que não estão representadas neste MER as tabelas que não fazem parte do domínio da plataforma POSCOMP *Coach*, deixando o diagrama mais claro e objetivo. Ainda assim, um diagrama ER completo, com todas as tabelas do projeto, incluindo as geradas de forma automática pode ser visualizado no Apêndice E – Diagramas Complementares.

Quadro 2 – Correlação Conceito, Classe e Tabela.

Modelo Conceitual	Diagrama de Classes	Tabela no Banco
Perfil	<i>Profile</i>	<i>account_profile</i>
Exame	<i>Exam</i>	<i>base_exam</i>
Item	<i>Item</i>	<i>base_item</i>
Alternativa	<i>Alternative</i>	<i>base_alternative</i>
Area	<i>Area</i>	<i>base_area</i>
Gabarito	<i>AnswerKey</i>	<i>base_answerkey</i>
Simulado	<i>Examination</i>	<i>mock_examination</i>
Questão	<i>Question</i>	<i>mock_question</i>
Modalidade	<i>Modality</i>	<i>mock_modality</i>

Fonte: Elaborada pelo Autor.

abordagem mostrou-se bastante eficaz, uma vez que dispensou a necessidade de construir uma versão específica para dispositivos móveis, permitindo assim, focar no desenvolvimento das funcionalidades sem comprometer o suporte às diferentes plataformas.

A plataforma foi disponibilizada no endereço eletrônico <http://poscompcoach.com>, e passou por um processo de avaliação por estudantes de computação, conforme será descrito no capítulo seguinte.

4 Avaliação do POSCOMP Coach

No intuito de realizar uma avaliação do POSCOMP *Coach*, a plataforma foi submetida a testes de usabilidade com o público alvo a que se destina – possíveis candidatos do Exame. A metodologia para a realização desta avaliação assim como os resultados obtidos são apresentados neste capítulo.

4.1 Metodologia

A dinâmica do processo avaliativo consistiu basicamente em três etapas: (i) definição de escopo e critérios de avaliação; (ii) elaboração do formulário utilizando os recursos disponibilizados pelo Google Forms¹, ferramenta web de construção e coleta de dados disponibilizada pelo Google²; (iii) convite para participação na pesquisa, realizado principalmente através do envio de e-mails.

Na primeira etapa, foram definidos os critérios e escopo do formulário de avaliação, além das características e funcionalidades que seriam avaliadas. Desta forma, o formulário consistiu então de três partes: identificação dos avaliadores, questões sobre usabilidade, e sugestões/comentários dos avaliadores. Os avaliadores foram instruídos a utilizar a plataforma POSCOMP Coach por pelo menos trinta (30) minutos antes de proceder com o preenchimento do formulário.

A construção do formulário utilizando o Google Forms se deu da seguinte forma. No que tange a identificação dos avaliadores foram solicitados os seguintes dados: nome, e-mail, nome do curso de graduação que está cursando ou cursou, situação no curso e região do país em que cursa ou cursou a graduação. Os itens relativos a usabilidade foram divididos em 9 critérios: (i) falar na linguagem do usuário; (ii) minimizar carga de memória do usuário; (iii) ser consistente; (iv) prover feedback; (v) saídas claramente marcadas e atalhos; (vi) mensagens de erros construtivas e precisas; (vii) compatibilidade do POSCOMP *Coach* com o mundo real; (viii) estética; e (ix) prevenção de erros. Para cada pergunta, foram disponibilizados cinco (5) opções de resposta: Concordo totalmente, Concordo, Indeciso, Discordo e Discordo Totalmente.

A última seção do formulário permitia que o avaliador desse uma nota geral ao POSCOMP *Coach* (0 a 10), assim como permitia que ele pudesse deixar comentários e/ou fazer sugestões de funcionalidades em campos de texto livre.

O formulário de avaliação completo encontra-se no [Anexo A – POSCOMP Coach](#)

¹ <https://www.google.com/intl/pt-BR/forms/about/>

² <https://www.google.com>

- **Formulário de Avaliação**, e sua elaboração seguiu as premissas estipuladas por Jakob Nielsen, no seu artigo intitulado *10 general principles for interaction design*, que são consideradas boas práticas estabelecidas que vão além da usabilidade (NIELSEN, 1995).

Na terceira e última etapa foi realizado o envio de e-mails com convites para participar da avaliação aos egressos de cursos de Computação, estudantes inscritos no ENADE³, finalistas dos cursos de graduação em Computação e docentes da área de Computação e afins.

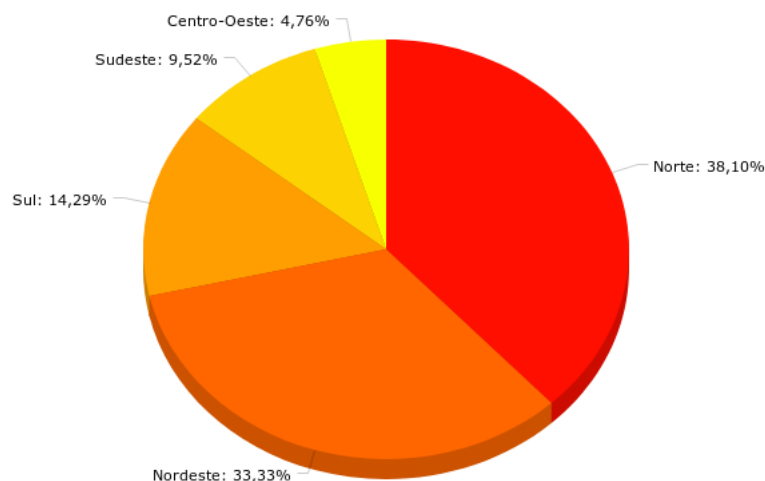
4.2 Análise dos resultados

O questionário avaliativo do POSCOMP Coach foi aplicado no período de 11 de Outubro até 02 de Novembro de 2017. No total, trinta e dois (32) participantes se dispuseram a contribuir com a pesquisa e enviaram suas respostas e sugestões com relação a usabilidade da plataforma.

4.2.1 Perfil dos avaliadores

Um dado interessante da pesquisa é que houve participação de pessoas todas as regiões do Brasil, conforme aponta o gráfico da Figura 20. Esse é um dado importante, pois opiniões de diferentes partes acrescentam diferentes perspectivas sobre a plataforma em relação ao universo acadêmico dessas regiões.

Figura 20 – Região do país em que cursa ou cursou a graduação.



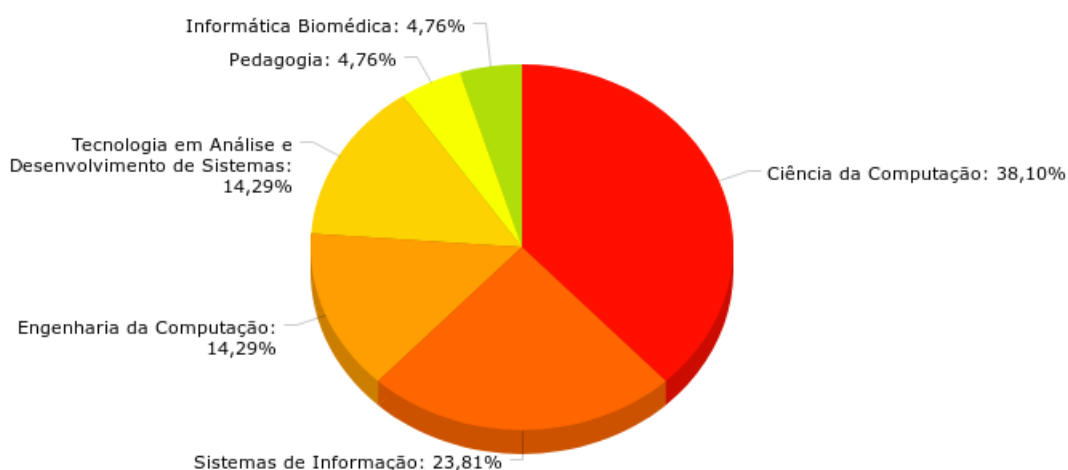
Fonte: Elaborada pelo autor.

Em relação ao perfil dos avaliadores quanto ao curso que está cursando, ou cursou, tem-se que 61,91% dos avaliadores cursam, ou cursaram Ciências da Computação ou

³ Exame Nacional de Desempenho de Estudantes, <http://enade.inep.gov.br/>

Sistemas da Informação. Os demais avaliadores eram originários de cursos de Engenharia da Computação e Tecnologia em Análise e Desenvolvimento de Sistemas, conforme ilustrado na [Figura 21](#).

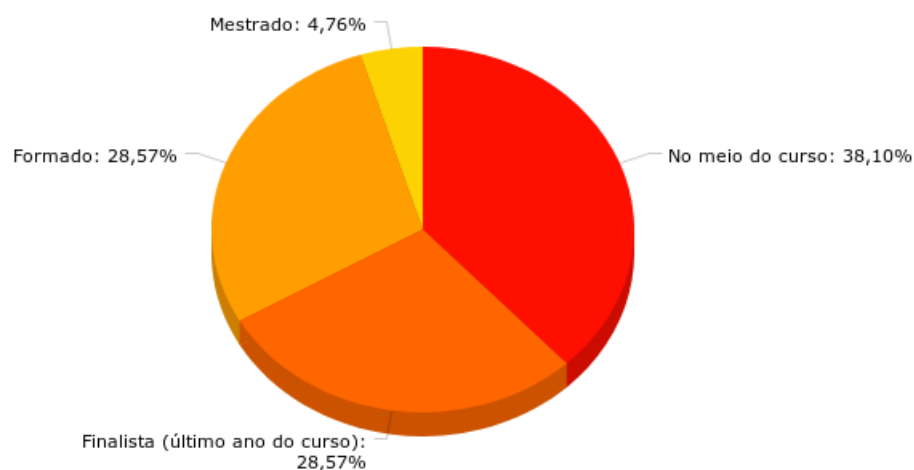
Figura 21 – Curso de Graduação que está cursando ou cursou.



Fonte: Elaborada pelo autor.

Com relação a situação dos avaliadores nos cursos informados, tem-se que graduados e pós-graduados formam 33,33% dos participantes, mas a maioria dos avaliadores está no meio do curso ou finalizando, o que corresponde a 38,10% e 28,57% respectivamente, conforme demonstrado no gráfico da [Figura 22](#).

Figura 22 – Situação no curso.



Fonte: Elaborada pelo autor.

4.2.2 Avaliação de Usabilidade

Segundo os avaliadores, o POSCOMP Coach se mostrou objetivo e claro com relação aos termos utilizados em sua interface, obtendo 100% de aprovação (considerando as respostas: concordo totalmente - 38,10% e concordo - 61,91%).

No que diz respeito a carga de memória para criação de simulados, 80,9% dos avaliadores consideraram pequena a carga de memória exigida para criação de simulados por ano (padrão POSCOMP). Já para simulados por área e por preferência (personalizado) as etapas foram consideradas pequenas por 85,7% dos avaliadores em ambos os casos. Ainda neste quesito, 90,4% dos avaliadores responderam que as etapas para retomar um simulado previamente iniciado é pequena. Todos os avaliadores informaram considerar pequena o número de etapas para consultar o desempenho.

Em relação a consistência, 90,47% dos avaliadores responderam que a plataforma apresenta poucas falhas. Já no caso da interface, 90,47% responderam que botões, menus e símbolos apresentam similaridade entre telas, e que as informações de orientação durante a navegação são suficientes. Durante o período de testes apenas um usuário reportou uma falha na autenticação com as redes sociais, que foi corrigida prontamente no mesmo dia, sem impactos perceptíveis aos outros participantes.

A totalidade dos avaliadores informou que a plataforma possui mecanismos para acompanhamento das tarefas requeridas, e 85,71% dos avaliadores responderam que a plataforma informa de forma rápida e clara quando uma determinada ação está sendo realizada ou foi concluída. Esses resultados evidenciam que os usuários sentem-se situados e com as tarefas facilmente acessíveis, mesmo com pouco tempo de uso da plataforma.

Quanto as mensagens de erro apresentadas aos usuários, 80,95% dos avaliadores responderam que as mensagens de erro são claras e propõe dicas para resolução, enquanto que 76,2% informaram que as mensagens de erro são claras, precisas e bastante explicativas.

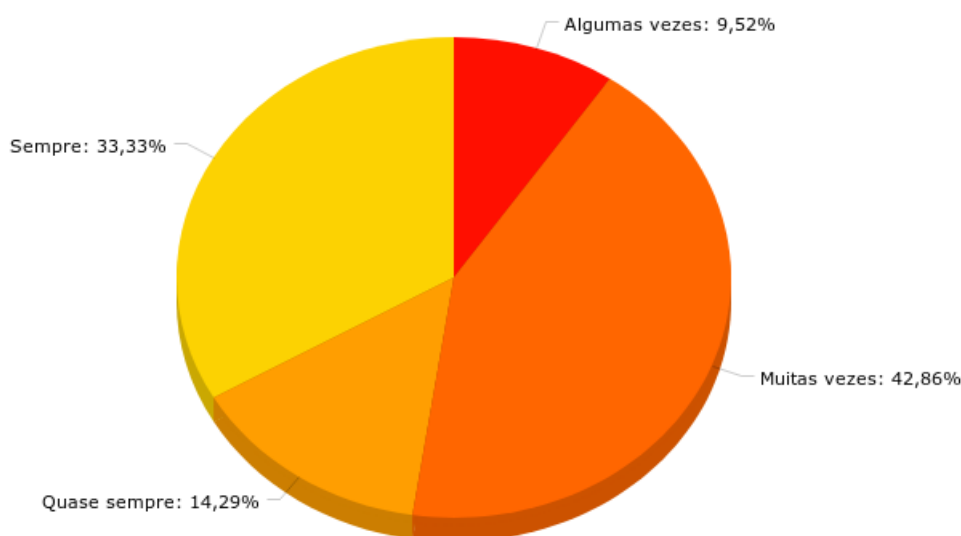
A comunicação com o mundo real, com relação à linguagem utilizada, frases e termos familiares foi aprovada pela totalidade dos avaliadores. A fácil compreensão e entendimento do conteúdo apresentado também foi avaliado positivamente.

Em se tratando da estética, os resultados são bastante positivos. Com relação a disposição dos elementos na tela, como botões, menus, imagens e textos, 80,95% concordaram que estão organizados e agrupados segundo um critério lógico e objetivo. Todos concordaram que os botões e janelas são concisos e 95,23% concordaram que as informações apresentadas são aceitáveis e facilitam entendimento do sistema.

No que tange a prevenção de erros, 85,71% dos avaliadores responderam que a plataforma disponibiliza em sua navegação, mensagens de alerta quando na tentativa de realizar uma atividade inadequada ou de inserir valores inválidos.

Quanto à utilização da plataforma, de acordo com o gráfico apresentado na [Figura 23](#) 90,5% dos avaliadores informaram que usariam a plataforma com bastante frequência. Questionados em relação a uma nota para o POSCOMP Coach, 81% dos atribuíram nota de 8,0 a 10,0. Por fim, ao responderem sobre a indicação da plataforma, todos os avaliadores responderam positivamente.

Figura 23 – Frequência de uso.



Fonte: Elaborada pelo autor.

4.2.3 Sugestões e *Feedback*

Ao final do questionário os avaliadores tiveram a oportunidade de informar quais funcionalidades ou características sentiram falta e não estão presentes nessa versão inicial do POSCOMP Coach, bem como reportar sugestões de melhorias e observações pertinentes.

Os problemas reportados por usuários considerados *bugs* durante o período de avaliação foram corrigidos. Um exemplo, foi o caso de um *bug* na autenticação utilizando redes sociais que utilizavam um e-mail previamente cadastrado como conta avulsa gerava um erro de duplicidade. Esse erro foi corrigido, e um alerta é emitido ao usuário informado que um cadastro com o referido e-mail já existe. Outro caso, mais relacionado a consistência da interface em que um usuário reportou ícones diferentes para a mesma ação de sair da plataforma, que poderia causar confusão, foi corrigido e um mesmo ícone foi usado em ambos os casos.

Com relação as funcionalidades propostas e sugestões de melhorias, duas delas são pertinentes, e poderiam ser incorporadas em versões futuras do POSCOMP *Coach*. Uma delas diz respeito à classificação das questões por assunto por meio de *tags*. Essa

funcionalidade é interessante pois facilitaria a revisão das provas, mas envolve uma mudança significativa no código da aplicação. Outra sugestão interessante foi a de criar uma modalidade para uma quantidade específica de questões de um exame específico, como por exemplo: realizar um simulado com apenas quinze questões de matemática do exame de 2015.

Todas as observações foram cuidadosamente avaliadas, e anotadas. Algumas das situações reportadas, como as descritas anteriormente, geraram a necessidade de correções pontuais de *bugs* ou inconsistências na interface. Já as sugestões pertinentes foram registradas para uma possível implementação em versões futuras do POSCOMP *Coach*, já que se tratam de melhorias ou adição de novas funcionalidades, o que excede o escopo definido inicialmente para o projeto.

Considerações Finais

O intuito deste trabalho foi apresentar o POSCOMP Coach, uma plataforma web para auxiliar os estudantes na sua preparação para o POSCOMP. A plataforma permite que os estudantes realizem exames simulados configuráveis e personalizados. A base de dados contempla as 1120 questões dos exames anteriores do POSCOMP realizados no período de 2002 a 2017. Com controle do tempo disponível e possibilidade de navegar, pelas questões antes de resolvê-las, a plataforma provê ao candidato um ambiente realístico de realização do exame. Além dos aspectos mencionados, a plataforma disponibiliza gráficos de desempenho geral e por área, permitindo ao estudante identificar aspectos da sua preparação que devem ser melhorados.

No desenvolvimento deste trabalho foi utilizada a metodologia Ágil AUP, que permite um desenvolvimento iterativo e incremental, de forma que no estágio atual de desenvolvimento, a plataforma disponibiliza diversos elementos essenciais que ajudam o candidato na sua preparação, dentre os quais destacam-se o registro de usuários utilizando a rede social Google+, Facebook ou formulário próprio, alteração de informações adicionais de perfil. Tela de gerenciamento dos simulados configurados, realizados e por fazer. Realização de Simulados, com possibilidade de suspender e continuar posteriormente. Visualização do resultado do exame mediante gabarito e gráfico de desempenho individual, além de revisão de simulados já realizados. para incentivar o aumento no número de questões resolvidas, e destacar os estudantes com maior índices de acertos. Um cronômetro de prova está disponível, o que auxilia no gerenciamento do tempo de resolução das provas. Para os administradores da plataforma é permitido a pesquisa e edição dos itens (questões) para eventuais correções e gerenciamento dos perfis de usuários. Visualização de estatísticas gerais, e mapa de distribuição geográfica dos participantes.

A versão funcional encontra-se disponível em <http://poscompcoach.com>. O desenvolvimento desta plataforma abre perspectivas de vários trabalhos futuros, dentre eles a divulgação da plataforma entre os estudantes de computação em todo Brasil e a partir de seu uso, investigar matérias e assuntos com índices menores de acertos. Esta investigação pode resultar em uma discussão sobre a formação dos estudantes em nível de graduação.

Outro trabalho seria em relação a manutenção evolutiva da plataforma, com implementação de novas funcionalidades, como por exemplo a classificação por *tags* sugerida por um dos avaliadores, ou ainda a possibilidade de professores adicionarem explicações sobre a resposta das questões, funcionalidade também sugerida por um avaliador. Um outro ponto importante a ser trabalhado, é o investimento em interfaces acessíveis para pessoas com deficiência e, neste caso, há um campo vasto para investimento de pesquisa e

desenvolvimento.

O desenvolvimento deste trabalho resultou em publicação de artigo científico, apresentado no III Fórum de Educação em Engenharia da Computação (FEEC 2017), como parte do XXX Symposium on Integrated Circuits and Systems Design. Chip on the Sands, ocorrido no período de 28 de Agosto a 1 de Setembro em Fortaleza – Ceará, Brasil. (MENDES; MENDONCA; GUEDES, 2017).

Referências

- AMBYSOFT. *The Agile Unified Process (AUP)*. 2006. <<http://www.ambysoft.com/unifiedprocess/agileUP.html>>. Acesso em: 08 Dezembro 2016. Citado na página 20.
- BERK, R. *A Consumer's Guide to Multiple-Choice Item Formats That Measure Complex Cognitive Outcomes*. Pearson Assessment and Information. 1996. <http://images.pearsonassessments.com/images/NES_Publications/1996_12Berk_368_1.pdf>. Acesso em: 05 Dezembro 2016. Citado na página 15.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: guia do usuário*. CAMPUS - RJ, 2006. ISBN 9788535217841. Disponível em: <<https://books.google.com.br/books?id=ddWqxcDKGF8C>>. Citado na página 23.
- BRAME, C. *Writing good multiple choice test questions*. 2013. <<https://cft.vanderbilt.edu/guides-sub-pages/writing-good-multiple-choice-test-questions/>>. Acesso em: 01 Dezembro 2016. Citado na página 15.
- CAED, C. de Políticas Públicas e Avaliação da Educação da Universidade Federal de Juiz de F. *Guia de Elaboração de Itens*. 2012. <http://www.portalavaliacao.caedufjf.net/wp-content/uploads/2012/02/Guia_De_-Elabora%C3%A7%C3%A3o_De_Itens_MT.pdf>. Acesso em: 02 Dezembro 2016. Citado na página 16.
- CAED, C. de Políticas Públicas e Avaliação da Educação da Universidade Federal de Juiz de F. *Item*. 2016. <<http://www.portalavaliacao.caedufjf.net/pagina-exemplo/item/>>. Acesso em: 02 Dezembro 2016. Citado na página 16.
- CAPES. *Relatórios de Avaliação - Ciência da Computação*. 2017. <<http://www.capes.gov.br/component/content/article/44-avaliacao/4656-ciencia-da-computacao>>. Acesso em: 22 Outubro 2016. Citado na página 11.
- CASELL, L.; GAULD, A. *Python Projects*. Wiley, 2014. (Wrox programmer to programmer). ISBN 9781118908891. Disponível em: <<https://books.google.com.br/books?id=8OSfBQAAQBAJ>>. Citado na página 24.
- de Sordi Junior, F. *Desenvolvimento de um ambiente colaborativo de treinamento preparatório para o POSCOMP*. Dissertação (Mestrado) — Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Ciência da Computação, Londrina, 2015. Disponível em <<http://poscomp.uel.br/>>. Acessado em 27 de novembro de 2017. Citado 2 vezes nas páginas 18 e 19.
- Django Documentation. *Coding style*. 2017. <<https://docs.djangoproject.com/en/dev/internals/contributing/writing-code/coding-style/>>. Acesso em: 02 Novembro 2017. Citado na página 41.
- Django Project. *Django Project*. 2017. <<https://www.djangoproject.com>>. Acesso em: 21 Janeiro 2017. Citado na página 26.
- FOWLER, M. *UML Essencial: Um Breve Guia para Linguagem Padrao*. Bookman, 2014. ISBN 9788560031382. Disponível em: <<https://books.google.com.br/books?id=xxoXcuh0oS0C>>. Citado na página 23.

git-scm. *git –fast-version-control*. 2017. <<https://git-scm.com/>>. Acesso em: 22 Janeiro 2017. Citado na página 27.

GORNIK, D. *Ibm rational unified process: Best practices for software development teams*. 2003. Disponível em: <ftp://ftp.software.ibm.com/software/rational/web/whitepapers/2003/rup_bestpractices.pdf>. Citado na página 20.

GUO, P. *Python is Now the Most Popular Introductory Teaching Language at Top U.S. Universities*. 2014. <<https://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-u-s-universities/fulltext>>. Acesso em: 29 Abril 2017. Citado na página 24.

HALADYNA, T. *Developing and Validating Multiple-choice Test Items*. Lawrence Erlbaum Associates, 2004. ISBN 9780805846614. Disponível em: <<https://books.google.com.br/books?id=4fJ2YXMLTrsC>>. Citado na página 15.

HALADYNA, T. M.; DOWNING, S. M.; RODRIGUEZ, M. C. A Review of Multiple-Choice Item-Writing Guidelines for Classroom Assessment. *Applied Measurement in Education*, v. 15, n. 3, p. 309–333, 2002. Disponível em: <http://dx.doi.org/10.1207/S15324818AME1503_5>. Citado na página 15.

JAWORSKI, M.; ZIADE, T. *Expert Python Programming*. Packt Publishing, 2016. ISBN 9781785884399. Disponível em: <https://books.google.com.br/books?id=_tBtDQAAQBAJ>. Citado na página 25.

LARMAN, C. *Utilizando UML e Padrões*. Bookman, 2002. ISBN 9788577800476. Disponível em: <<https://books.google.com.br/books?id=hxl2tmT8QkUC>>. Citado 2 vezes nas páginas 36 e 39.

LARMAN, C. *Utilizando UML e Padrões – Um Guia para a Análise e Projeto Orientados a Objetos*. [S.l.]: Bookman, 2007. Citado 2 vezes nas páginas 23 e 30.

MEDEIROS, C. B. Computação: o terceiro pilar. *Revista USP*, scielo, p. 134 – 147, 05 2011. ISSN 0103-9989. Disponível em: <http://rusp.scielo.br/scielo.php?script=sci_arttext&pid=S0103-99892011000200010&nrm=iso>. Citado na página 11.

MENDES, M.; MENDONCA, A.; GUEDES, E. Poscomp coach: Uma plataforma web para auxílio a estudantes na preparação para o poscomp. *III Fórum de Educação em Engenharia da Computação (FEEC 2017)*. XXX Symposium on Integrated Circuits and Systems Design. Chip on the Sands, 2017. Disponível em: <<http://chiponthesands.lesc.ufc.br/feec-2017/technical-program>>. Citado na página 55.

MySQL. *Top Reasons to Use MySQL*. 2017. <<https://www.mysql.com/why-mysql/topreasons.html>>. Acesso em: 21 Janeiro 2017. Citado na página 26.

NIELSEN, J. *10 Usability Heuristics for User Interface Design*. [S.l.]: Nielsen Norman Group, 1995. Citado na página 49.

OETIKER H. PARTL, I. H. T.; SCHLEGL, E. *The Not So Short Introduction to latex 2*. Packt Publishing, 2016. Disponível em: <<https://tobi.oetiker.ch/lshort/lshort.pdf>>. Citado na página 28.

OLIPHANT, T. E. Python for scientific computing. *Computing in Science and Engg.*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 9, n. 3, p. 10–20, maio 2007. ISSN 1521-9615. Disponível em: <<http://dx.doi.org/10.1109/MCSE.2007.58>>. Citado na página 25.

OMG, O. M. G. *Unified Modeling LanguageTM (UML®)*. 2016. <<http://www.omg.org/spec/UML/>>. Acesso em: 13 Dezembro 2016. Citado na página 23.

Oracle, Java Documentation - The JavaTM Tutorials. *Controlling access to members of a class*. 2017. <<https://docs.oracle.com/javase/tutorial/java/javaOO/accesscontrol.html>>. Acesso em: 17 Outubro 2017. Citado na página 88.

PARKES, J.; ZIMMARO, D. *Learning and Assessing with Multiple-Choice Questions in College Classrooms*. Taylor & Francis, 2016. ISBN 9781317540908. Disponível em: <<https://books.google.com.br/books?id=HKLDCwAAQBAJ>>. Citado na página 15.

PENDER, T. *UML Bible*. Wiley, 2003. (Bible). ISBN 9780764526046. Disponível em: <<https://books.google.com.br/books?id=44IhAQAIAAJ>>. Citado na página 41.

PHILLIPS, D. *Python 3 Object-oriented Programming*. Packt Publishing, 2015. ISBN 9781784395957. Disponível em: <<https://books.google.com.br/books?id=dYloCgAAQBAJ>>. Citado 2 vezes nas páginas 88 e 89.

PRESSMAN, R. S. *Engenharia de Software: Uma abordagem profissional*. [S.l.]: Bookman, 2011. ISBN 9788580550443. Citado 2 vezes nas páginas 20 e 21.

Python Brasil. *Defesa do Python*. 2008. <<http://wiki.python.org.br/DefesaPython>>. Acesso em: 13 Dezembro 2016. Citado na página 24.

Python Software Foundation. *The Python Tutorial*. 2017. <<https://docs.python.org/3.6/tutorial/classes.html>>. Acesso em: 16 Outubro 2017. Citado na página 88.

REZENDE, D. *Engenharia de Software e Sistemas de Informação*. Brasport, 2005. ISBN 9788574522159. Disponível em: <https://books.google.com.br/books?id=rtBvl_L-1mcC>. Citado na página 45.

SBC. *EDITAL SBC N. 001/2016*. 2016. <<http://www.sbc.org.br/documentos-da-sbc/send/186-editais/1015-edital-poscomp-2016>>. Acesso em: 07 Dezembro 2016. Citado na página 18.

SILVA, M. S. *Bootstrap 3.3.5 Aprenda a usar o framework Bootstrap para criar layouts CSS complexos e responsivos*. Novatec Editora LTDA, 2015. ISBN 788575224601. Disponível em: <<http://www.livrosdomaujor.com.br/bootstrap3/codigos.html>>. Citado na página 29.

SILVA, R. S. *Questões para POSCOMP*. 2017. <https://play.google.com/store/apps/details?id=com.ufms_cpppp.poscomp>. Acessado em 27 de novembro de 2017. Citado na página 18.

Source Making. *Vendor Lock-In. AntiPatterns. Software Architecture AntiPatterns*. 2016. <<https://sourcemaking.com/antipatterns/vendor-lock-in>>. Acesso em: 13 Dezembro 2016. Citado na página 24.

The LaTeX Project. *LaTeX – A document preparation system*. 2017. <<https://www.latex-project.org/>>. Acesso em: 17 Março 2017. Citado na página 28.

TIOBE. *TIOBE Index for December 2016*. 2016. <<http://www.tiobe.com/tiobe-index/>>. Acesso em: 13 Dezembro 2016. Citado na página 24.

Apêndices

APÊNDICE A – Especificação de Casos de Uso: Registro e Autenticação

Quadro 3 – Caso de Uso CDU01: Registrar na plataforma.

Ator principal: Candidato

Interessados e Interesses:

Estudantes que desejam melhorar sua preparação para o POSCOMP.

Pré-condições: O candidato deve possuir cadastro nas redes sociais Facebook/Google ou E-mail

Pós-condições (Garantia de Sucesso): Mensagem de *feedback*: "Cadastro efetuado com sucesso"

Cenário de sucesso:

1. Estudante acessa o endereço web da plataforma através do *browser*.
2. Na página principal, são apresentadas as opções de registro através de redes sociais ou formulário próprio da plataforma.
3. Estudante escolhe a que melhor atende suas necessidades, e procede com o preenchimento dos dados, quando requisitados.

Extensões (ou Fluxo alternativo):

Em caso de preenchimento incorreto dos campos obrigatórios uma mensagem de alerta será mostrada, informando que as informações fornecidas são inválidas ou incompletas.

Fonte: Elaborado pelo autor.

Quadro 4 – Caso de Uso CDU02: Logar na plataforma.

Ator principal: Candidato

Pré-condições: Ter executado o caso de uso CDU01 - Registrar-se na plataforma

Pós-condições (Garantia de Sucesso): Mensagem de *feedback*: "login efetuado com sucesso"

Cenário de sucesso:

1. Candidato acessa o endereço web da plataforma através do *browser*.
2. Opta por entrar usando as redes sociais ou o formulário próprio.

Extensões (ou Fluxo alternativo):

Candidato fez cadastro pela plataforma e esqueceu a senha: Caso o candidato tenha esquecido sua senha ele poderá solicitar uma nova, seguindo os passos no caso de uso CDU03.

Fonte: Elaborado pelo autor.

Quadro 5 – Caso de Uso CDU03: Recuperar Senha.

Ator principal: Candidato

Pré-condições: Ter executado o caso de uso CDU01 - Registrar-se na plataforma

Pós-condições (Garantia de sucesso): Mensagem de *feedback* informando que a senha foi redefinida com sucesso. **Cenário de sucesso:**

1. Clica na opção Esqueci minha senha no formulário de login.
2. É apresentado um campo para preenchimento do e-mail utilizado no cadastro.
3. Candidato informa o e-mail, clica em enviar.
4. Mensagem de *feedback* informando sobre instruções de redefinição que são enviadas para o e-mail cadastrado.
5. Candidato verifica o e-mail e segue as instruções para redefinição de senha.

Diversos: Nos casos de cadastro utilizando rede social, o e-mail informado deverá ser o mesmo que foi utilizado para cadastro na rede social utilizada.

Fonte: Elaborado pelo autor.

Quadro 6 – Caso de Uso CDU04: Editar Perfil.

Ator principal: Candidato

Pré-condições:

Ter executado o caso de uso CDU01 - Registrar-se na plataforma

Ter executado o caso de uso CDU02 - Logar na plataforma

Pós-condições (Garantia de sucesso): Mensagem de *feedback* informando que o perfil foi alterado com sucesso.

Cenário de sucesso:

1. Clica em editar no menu lateral Perfil.
2. Candidato é direcionado para tela de edição de perfil.
3. Realiza as alterações desejadas.
4. Clica em salvar alterações.

Fonte: Elaborado pelo autor.

APÊNDICE B – Especificação de Casos de Uso: Visão do Candidato

Quadro 7 – Caso de Uso CDU05: Preparar Simulado.

Ator principal: Candidato

Pré-condições:

Ter executado o caso de uso CDU01 - Registrar-se na plataforma

Ter executado o caso de uso CDU02 - Logar na plataforma

Cenário de sucesso:

1. Seleciona a opção Realizar um novo Simulado no menu lateral Menus Simulados.
2. Seleciona o Tipo de Simulado dentre as opções disponíveis: Padrão POSCOMP, Por assunto ou Personalizado.
3. Clica em Gerar simulado.
4. Mensagem de *feedback* informando que o simulado foi gerado com sucesso.
5. Simulado fica disponível na tela de Meus Simulados aguardando comando do usuário para ser iniciado.

Extensões (ou Fluso alternativo):

– Execução do Caso de uso CDU06: Realizar simulado.

Fonte: Elaborado pelo autor.

Quadro 8 – Caso de Uso CDU06: Realizar Simulado.

Ator principal: Candidato

Pré-condições: Ter executado o caso de uso CDU05 - Preparar Simulado

Cenário de sucesso:

1. Clica na opção Começar agora apresentada ao final do CDU03.
2. Marca as alternativas que julgar corretas.
3. Clica em Finalizar Simulado após a última questão apresentada.

Extensões (ou Fluxo alternativo):

1. Clica no menu lateral Meus simulados.
2. Seleciona o simulado com status de aguardando.
3. Clica em Começar agora.

Diversos: Ao candidato é permitido ir e voltar nas questões para rever suas respostas antes de optar por finalizar o simulado.

Fonte: Elaborado pelo autor.

Quadro 9 – Caso de Uso CDU07: Listar Simulados.

Ator principal: Candidato

Pré-condições: Ter executado o caso de uso CDU06 - Realizar Simulado, ao menos uma vez

Pós-condições: Listagem de todos os simulados, com seus respectivos estados: ativo, aguardando e finalizado.

Cenário de sucesso:

1. Seleciona a opção Meus Simulados no barra lateral ou no menu superior.
2. São listados todos os simulados com seu status.

Fonte: Elaborado pelo autor.

Quadro 10 – Caso de Uso CDU08: Visualizar Gabarito.

Ator principal: Candidato**Pré-condições:**

- Ter executado o caso de uso CDU06 - Realizar Simulado, ao menos uma vez
- Ter executado o caso de uso CDU07 - Listar Simulados

Pós-condições: Listagem do resultado do simulado com gabarito das questões resolvidas e não resolvidas**Cenário de sucesso:**

1. Clica no ícone Detalhes do simulado, na lista de simulados.
2. Ao fim da página é mostrado o gabarito do simulado

Diversos: O resultado do simulado é também exibido no momento após a finalização do mesmo pelo candidato.

Fonte: Elaborado pelo autor.

Quadro 11 – Caso de Uso CDU09: Revisar Simulado.

Ator principal: Candidato**Pré-condições:** Ter executado o caso de uso CDU06 - Realizar Simulado, ao menos uma vez**Pós-condições:** Navegação através de um simulado previamente realizado e finalizado.**Cenário de sucesso:**

1. Clica no ícone Revisar do simulado, na lista de simulados.
2. Usuário tem a possibilidade de navegar pelas questões e revisar suas respostas.

Fonte: Elaborado pelo autor.

Quadro 12 – Caso de Uso CDU10: Visualizar Desempenho.

Ator principal: Candidato**Pré-condições:** Ter executado o caso de uso CDU06 - Realizar Simulado, ao menos uma vez**Pós-condições:** Exibição de gráficos informativos com porcentagem de erros e acertos, no âmbito geral e também por área do conhecimento.**Cenário de sucesso:**

1. Seleciona a opção "Análise Gráfica" no menu "Desempenho".
2. São exibidos gráficos com desempenho Geral e Por assunto.

Fonte: Elaborado pelo autor.

APÊNDICE C – Especificação de Casos de Uso: Visão do Administrador

Quadro 13 – Caso de Uso CDU11: Listar exames.

Ator principal: Administrador

Pré-condições: Possuir perfil administrativo (*staff*)

Pós-condições: Listagem de todos os exames POSCOMP importados para na plataforma

Cenário de sucesso:

1. Administrador acessa o menu Banco de itens no menu lateral esquerdo.
2. Seleciona a opção Exames
3. São listados todos os exames POSCOMP presentes na plataforma.

Fonte: Elaborado pelo autor.

Quadro 14 – Caso de Uso CDU12: Pesquisar Itens.

Ator principal: Administrador

Pré-condições: Possuir perfil administrativo (*staff*)

Pós-condições: Listagem dos itens quem coincidem com o critério de pesquisa

Cenário de sucesso:

1. Administrador acessa o menu Banco de itens no menu lateral esquerdo.
2. Seleciona a opção Itens.
3. É exibida a tela de pesquisa de itens.
4. Administrador insere o termo de pesquisa na caixa de texto e clica e Pesquisar.
5. Listagem com resultado da pesquisa é exibida.

Extensões (ou Fluxo alternativo):

– Execução do caso de uso CDU13: Editar item.

Fonte: Elaborado pelo autor.

Quadro 15 – Caso de Uso CDU13: Editar Item.

Ator principal: Administrador

Pré-condições:

- Possuir perfil administrativo (*staff*)
- Ter executado o CDU12: Pesquisar Itens

Pós-condições: Mensagem de *feedback* informando que o item foi alterado com sucesso.

Cenário de sucesso:

1. Administrador clica na opção Detalhes do item
2. É exibida a tela edição de item.
3. Administrador altera o conteúdo do item, e clica em Pré-visualizar.
4. Repete o passo anterior até concluir que finalizou a edição.
5. Clica em salvar alterações.

Extensões (ou Fluxo alternativo):

- Execução do caso de uso CDU14: Editar alternativa.

Fonte: Elaborado pelo autor.

Quadro 16 – Caso de Uso CDU14: Editar Alternativa.

Ator principal: Administrador

Pré-condições:

- Possuir perfil administrativo (*staff*)
- Ter executado o CDU13: Editar Item

Pós-condições: Mensagem de *feedback* informando que a alternativa foi alterada com sucesso.

Cenário de sucesso:

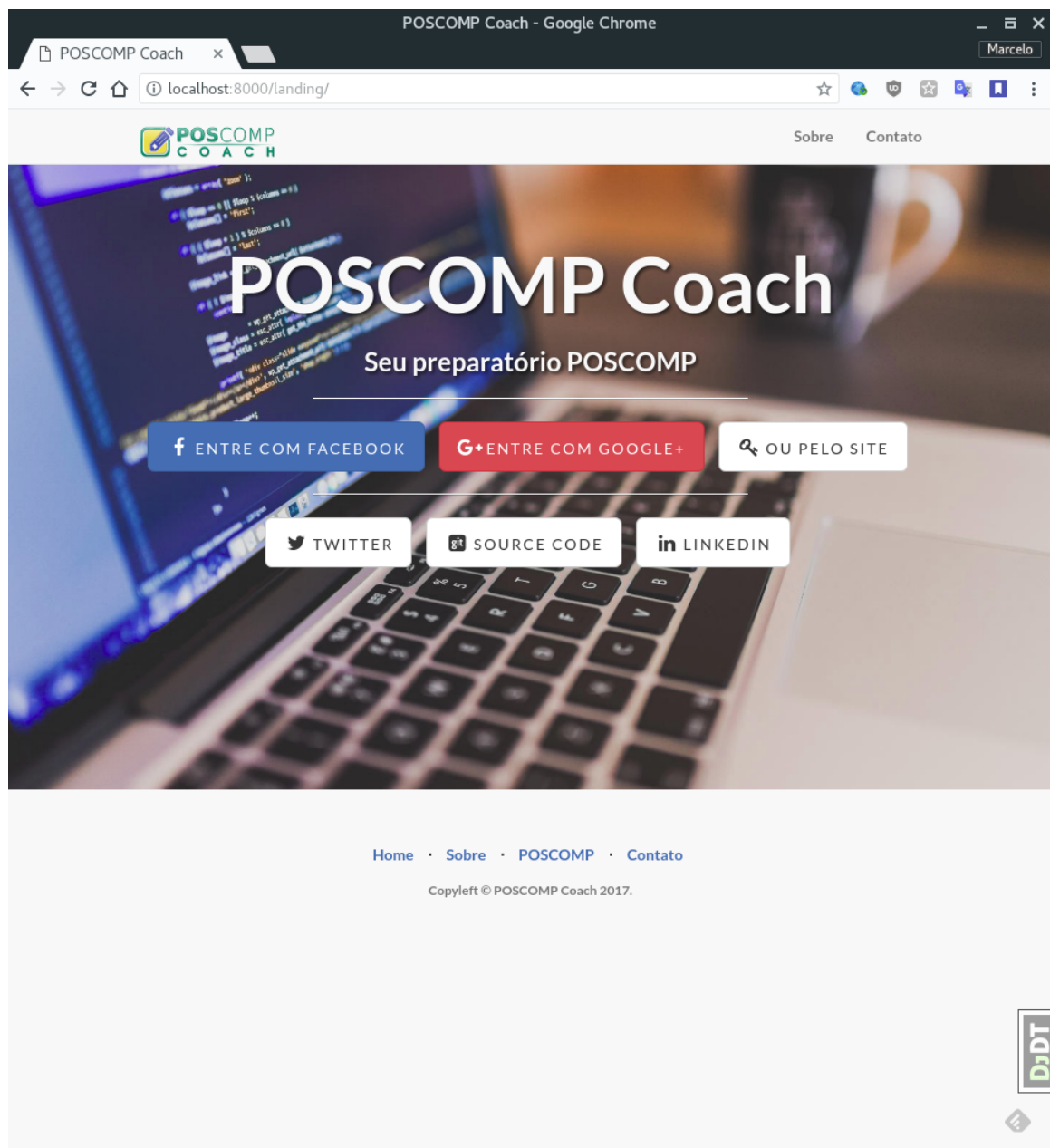
1. Administrador clica na alternativa que deseja editar.
2. É exibida a tela edição de alternativa.
3. Administrador altera o conteúdo do alternativa, e clica em Pré-visualizar item.
4. Repete o passo anterior até concluir que finalizou a edição.
5. Clica em salvar alterações na tela de edição de item.

Fonte: Elaborado pelo autor.

APÊNDICE D – Interfaces do Usuário

A tela inicial (*landing page*) do POSCOMP *Coach*, Figura 24, é o ponto de partida para o candidato, potencial usuário da plataforma. Aqui foram observados aspectos de usabilidade e objetividade, de forma a tornar simples e intuitivo o registro na plataforma.

Figura 24 – UI - POSCOMP *Coach*: Página inicial (*Landing Page*).



Fonte: Elaborado pelo autor.

O Painel Principal (*dashboard*) apresentado na [Figura 25](#) demonstra a visão do candidato após efetuar o primeiro login. Nesta tela o candidato receberá as primeiras instruções sobre o uso da plataforma.

Figura 25 – UI - POSCOMP *Coach*: Dashboard (Usuário).



Fonte: Elaborado pelo autor.

Na Figura 26 é apresentada a tela de edição de perfil, onde o candidato pode fazer a alteração de seus dados, tais como nome e sobrenome, e-mail, foto, data de nascimento. Nesta tela o candidato poderá efetuar também a alteração de senha, caso tenha utilizado o formulário própria fornecido pela plataforma.

Figura 26 – UI - POSCOMP *Coach*: Perfil.

A imagem mostra a interface de edição de perfil de um usuário no sistema POSCOMP Coach, acessada via navegador Google Chrome. O endereço da página é localhost:8000/account/edit/. A interface possui um cabeçalho azul com o título "Meu Perfil" e um menu hambúrguer à esquerda. O conteúdo principal é dividido em duas abas: "informações do usuário" (ativa) e "Segurança e Privacidade".

Na aba "informações do usuário", há um campo "Foto" com uma imagem de perfil e um botão "Alterar imagem". Abaixo, há campos de texto para "Nome" (preenchido com "Marcelo"), "Sobrenome" (preenchido com "Mendes"), "Email" (preenchido com "marcelomendes@gmail.com") e "Nascimento" (preenchido com "06/09/1980"). Um botão verde "Salvar alterações" está no canto inferior direito da seção.

Na barra de rodapé, há links para "Home", "Sobre", "POSCOMP" e "Contato", além do aviso de direitos autorais "Copyright © POSCOMP Coach 2017." e um logo "D3DT" no canto inferior direito.

Fonte: Elaborado pelo autor.

A captura de tela apresentada na Figura 27 mostra a tela "Meus Simulados", onde o candidato pode visualizar o estado dos simulados configurados, aguardando início, ativos ou finalizados.

Figura 27 – UI - POSCOMP *Coach*: Meus simulados.

POSCOMP Coach :: Simulado - Google Chrome

POSCOMP COACH Provas e Simulados Desempenho

Meus Simulados

Simulados finalizados, suspensos, ou não iniciados

#	Modalidade	Gerado	Iniciado	Finalizado	Situação	Ações
42	Padrão POSCOMP	20/08/2017 20:30	20/08/2017 20:30	20/08/2017 20:33	Finalizado	
43	Personalizado	20/08/2017 20:36	20/08/2017 20:36	20/08/2017 20:36	Finalizado	
44	Por Área do Conhecimento	20/08/2017 20:37	20/08/2017 20:38	20/08/2017 20:38	Finalizado	
46	Personalizado	16/09/2017 17:05	16/09/2017 17:05	16/09/2017 17:06	Finalizado	
51	Por Área do Conhecimento	02/10/2017 21:32	02/10/2017 21:40	02/10/2017 21:43	Finalizado	
52	Padrão POSCOMP	07/10/2017 15:26			Suspenso	
56	Personalizado	13/10/2017 17:03	13/10/2017 17:03	13/10/2017 17:09	Finalizado	

[Home](#) · [Sobre](#) · [POSCOMP](#) · [Contato](#)

Fonte: Elaborado pelo autor.

A Figura 28 apresenta a opção de simulado na modalidade "Padrão POSCOMP". Como dito anteriormente, essa modalidade permite ao candidato realizar qualquer exame anterior do POSCOMP, desde a primeira edição em 2002.

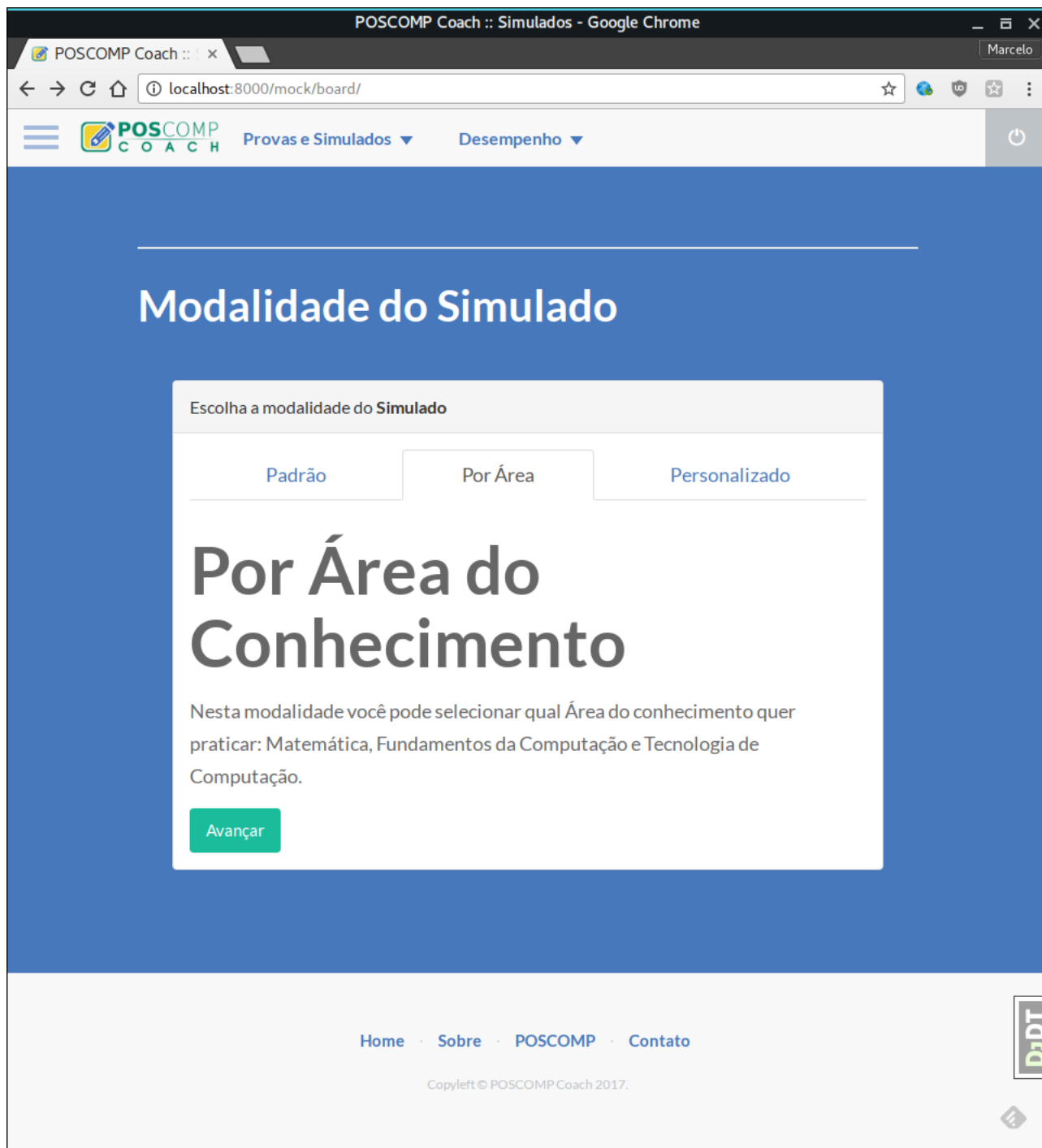
Figura 28 – UI - POSCOMP *Coach*: Simulado por ano (padrão).



Fonte: Elaborado pelo autor.

A Figura 29 apresenta a opção de simulado na modalidade "Por Área do Conhecimento", que possibilita ao candidato realizar um simulado baseado em uma das três áreas do Exame POSCOMP – Matemática, Fundamentos da Computação e Tecnologia de Computação.

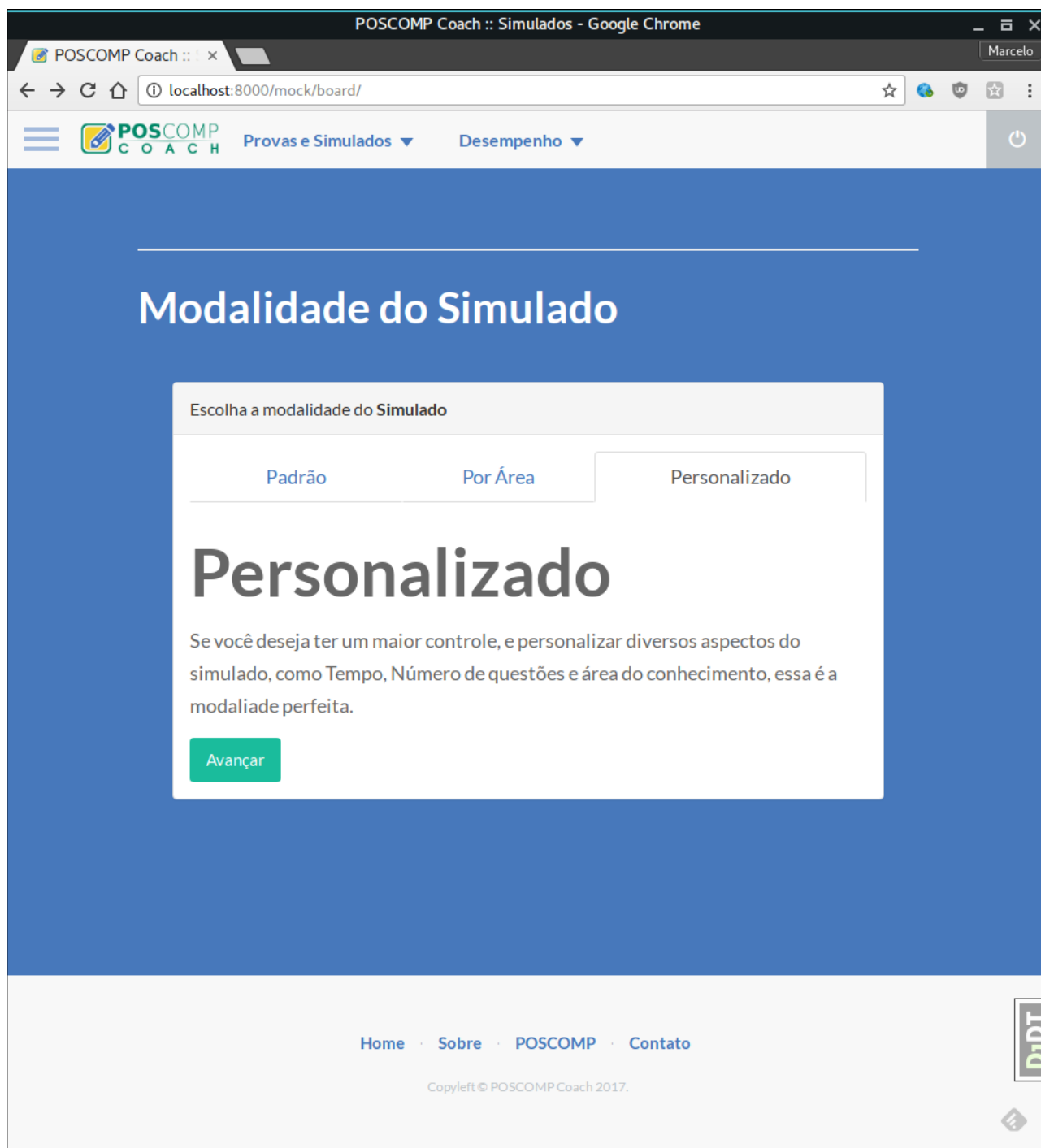
Figura 29 – UI - POSCOMP Coach: Simulado por Área.



Fonte: Elaborado pelo autor.

Na captura de tela apresentada na Figura 30 é mostrado a opção e descrição da modalidade de simulado "Personalizado". Esta modalidade permite ao candidato especificar o número de questões por área, as áreas que deseja incluir e ainda o tempo de prova.

Figura 30 – UI - POSCOMP Coach: Personalizado.



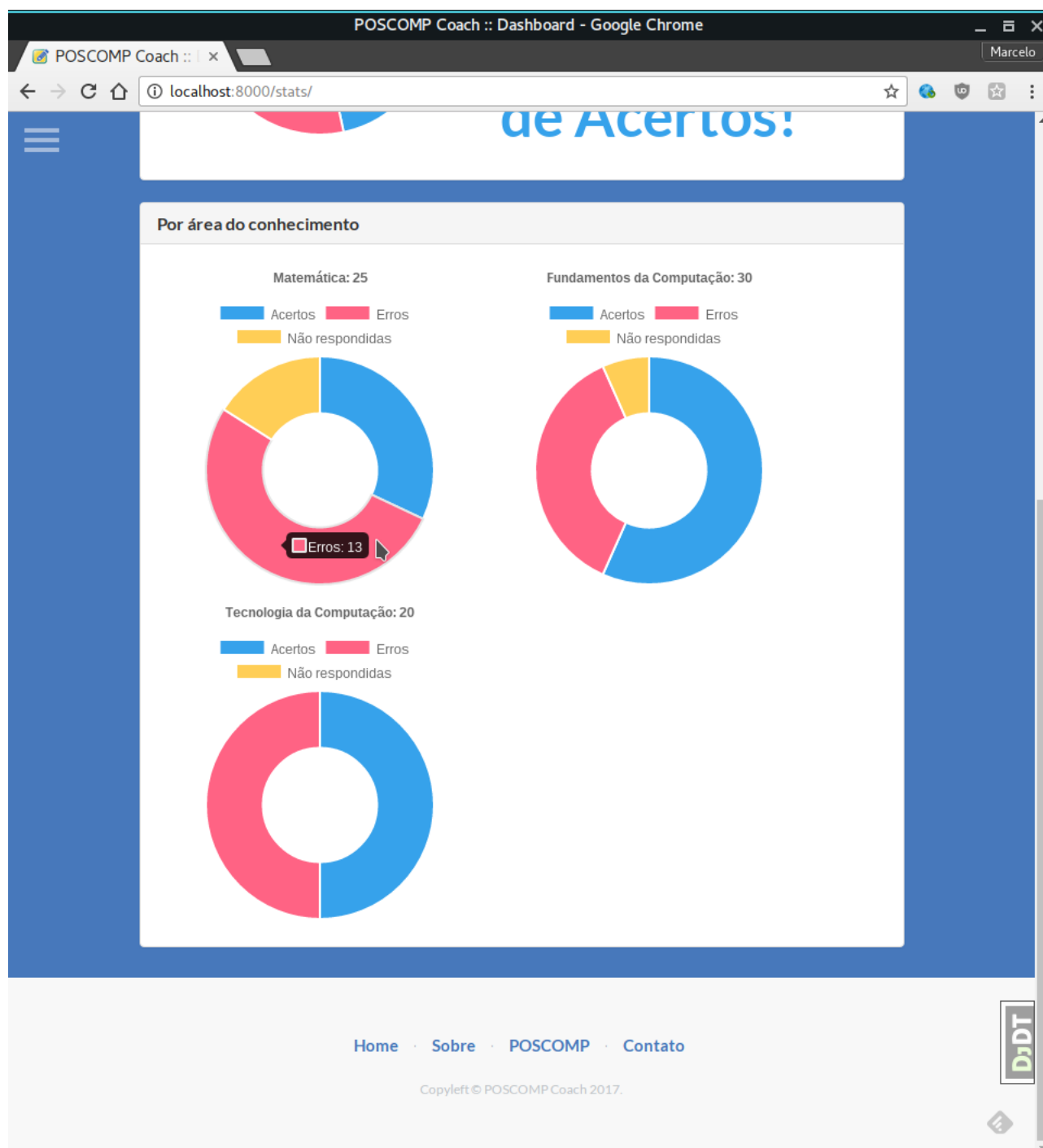
Fonte: Elaborado pelo autor.

Nas Figuras 31 e 32 são mostradas as capturas de tela referentes à análise de desempenho gráfica individual. Neste caso, uma análise geral, com incluindo todos os exames realizados, e uma análise por área do conhecimento.

Figura 31 – UI - POSCOMP *Coach*: Análise de Desempenho (GERAL).



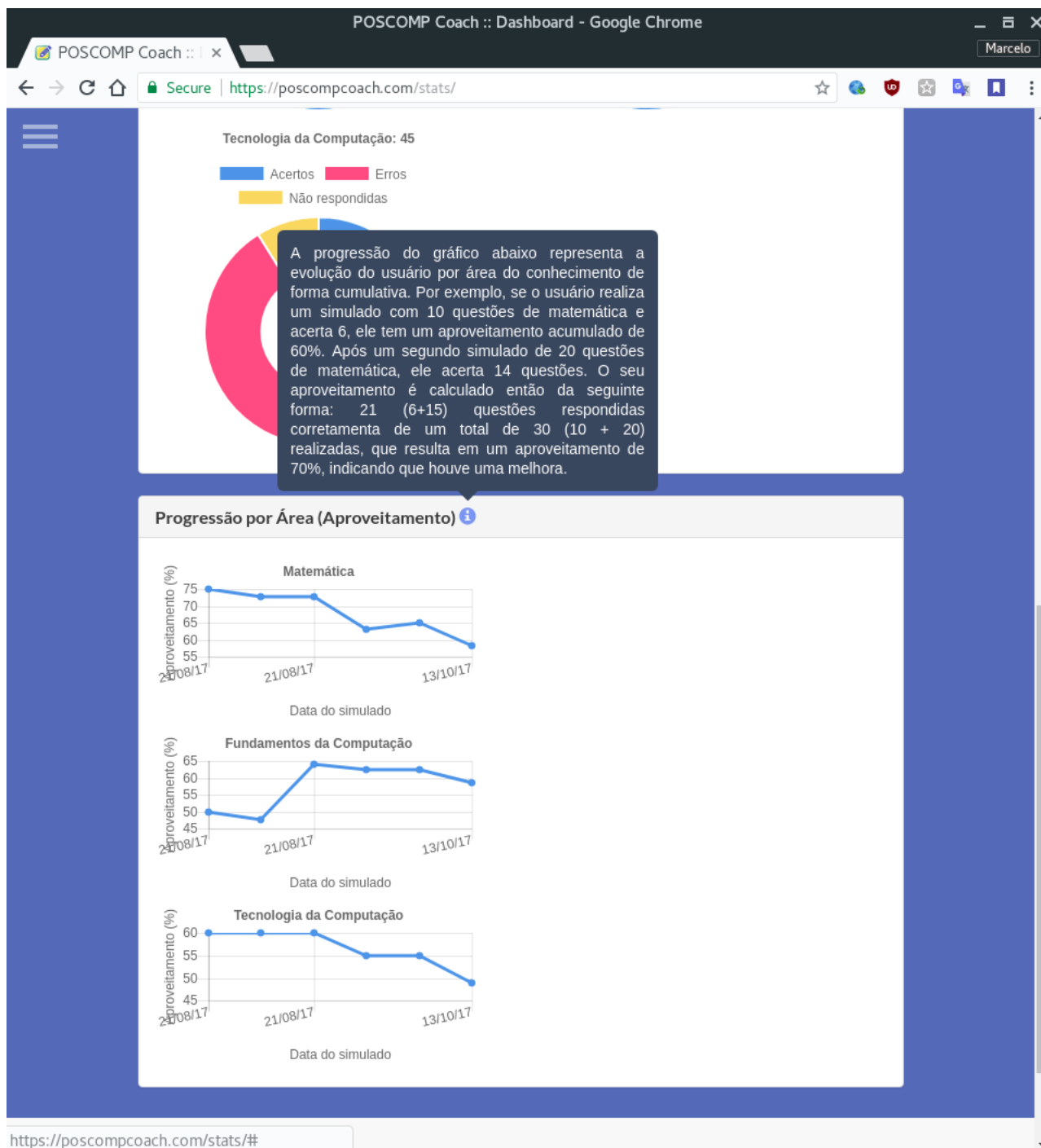
Fonte: Elaborado pelo autor.

Figura 32 – UI - POSCOMP *Coach*: Análise de Desempenho (Por Área).

Fonte: Elaborado pelo autor.

Ainda com relação ao desempenho individual, tem-se na Figura 33 o desempenho progressivo por área.

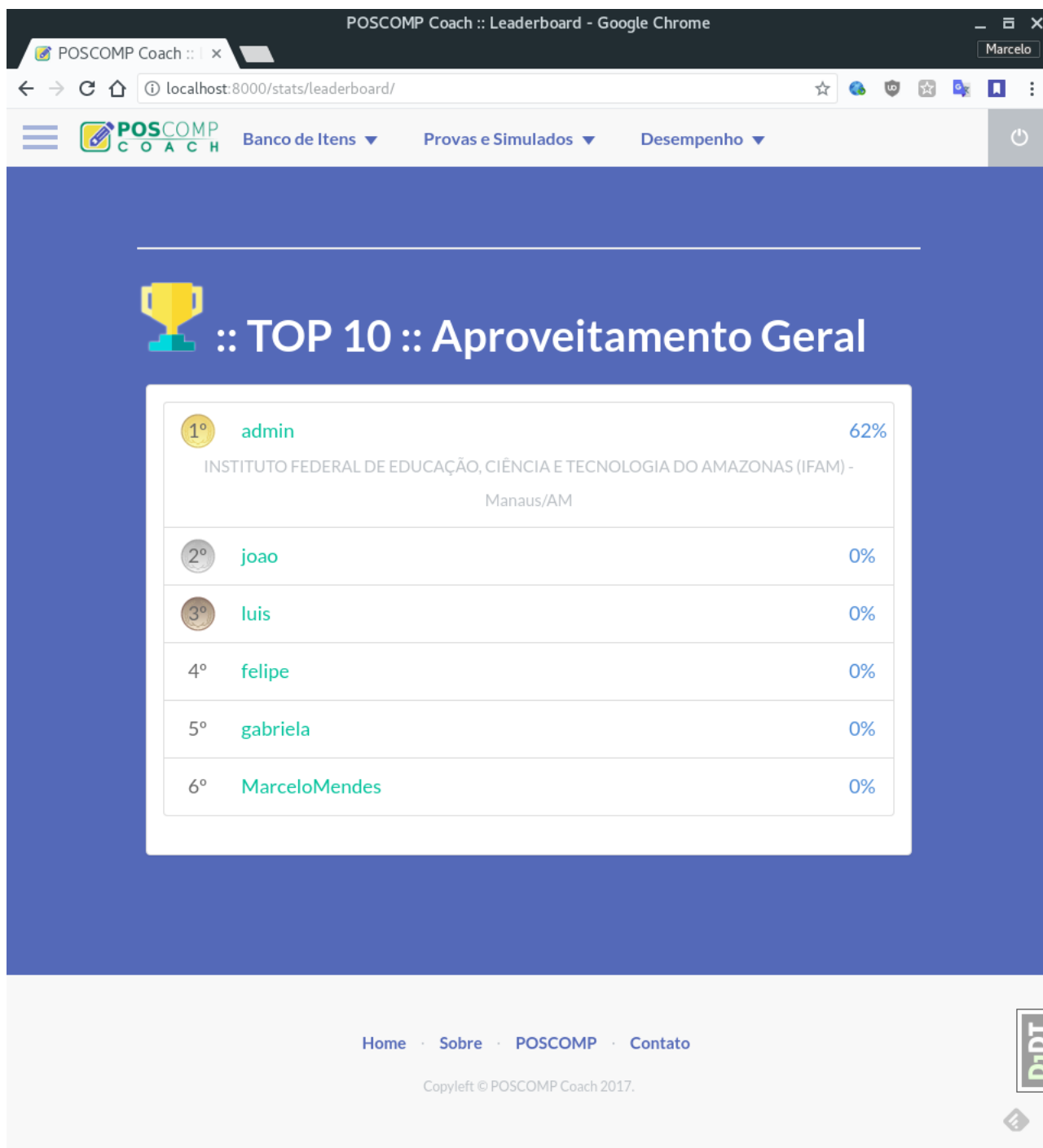
Figura 33 – UI - POSCOMP Coach: Progressão individual (Por Área).



Fonte: Elaborado pelo autor.

Com intuito de incentivar os usuários, foi implementado um *leaderboard* com o ranking dos dez (10) primeiros colocados em relação aos demais, no que diz respeito ao seu desempenho geral. É o que mostra a Figura 34.

Figura 34 – UI - POSCOMP *Coach*: *Leaderboard*.



Fonte: Elaborado pelo autor.

No contexto do Administrador da plataforma, temos inicialmente o "Painel Principal", mostrado na Figura 35. A principal diferença com relação ao Painel do Candidato é a presença do menu "Banco de Itens" na barra de topo da página. Nesse menu, estão as opções de manutenção de Exames, Itens e Alternativas dos Exames POSCOMP importados.

Figura 35 – UI - POSCOMP Coach: Dashboard (Administrador).



Fonte: Elaborado pelo autor.

Também no contexto administrativo tem-se na Figura 36 a tela "Editar Item", onde é mostrado o formulário de edição de um Item e os campos que o compõe.

Figura 36 – UI - POSCOMP *Coach*: Editar Item.

POSCOMP Coach :: Editar Item - Google Chrome

POSCOMP Coach :: x

localhost:8000/base/item/edit/2972/

Editar Item # 2972

Exame: POSCOMP - 2002

Área: Matemática

Número: 3

Enunciado: Seja n um número inteiro positivo. Considere a função f definida recursivamente por

$$f(n) = \begin{cases} 0 & \text{se } n = 1 \\ f(\lfloor \frac{n}{2} \rfloor) + 1 & \text{se } n \geq 2 \end{cases}$$

onde $\lfloor k \rfloor$ é o maior inteiro maior ou igual a k . O valor de $f(25)$ é igual a

A	\item 5	✗
B	\item 4	✓
C	\item 6	✗
D	\item 3	✗
E	\item 2	✗

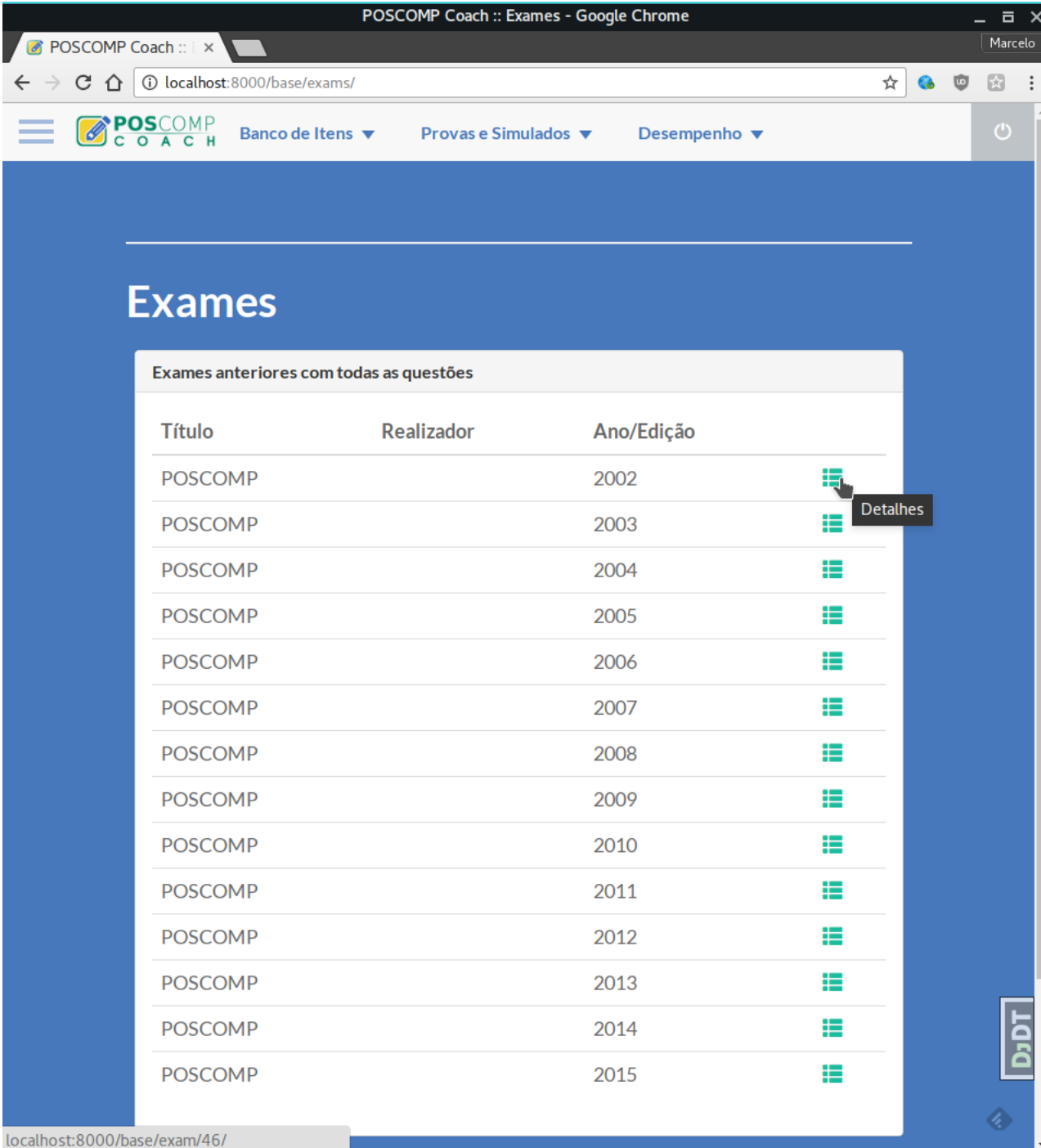
Voltar Pré-Visualizar Salvar alterações

D3DT

Fonte: Elaborado pelo autor.

Ainda no contexto administrativo, é apresentado na Figura 37 a tela com a listagem com todos os Exames POSCOMP cadastrados, totalizando 1120 questões disponíveis para composição dos simulados pelo candidato.

Figura 37 – UI - POSCOMP Coach: Listagem de Exames.



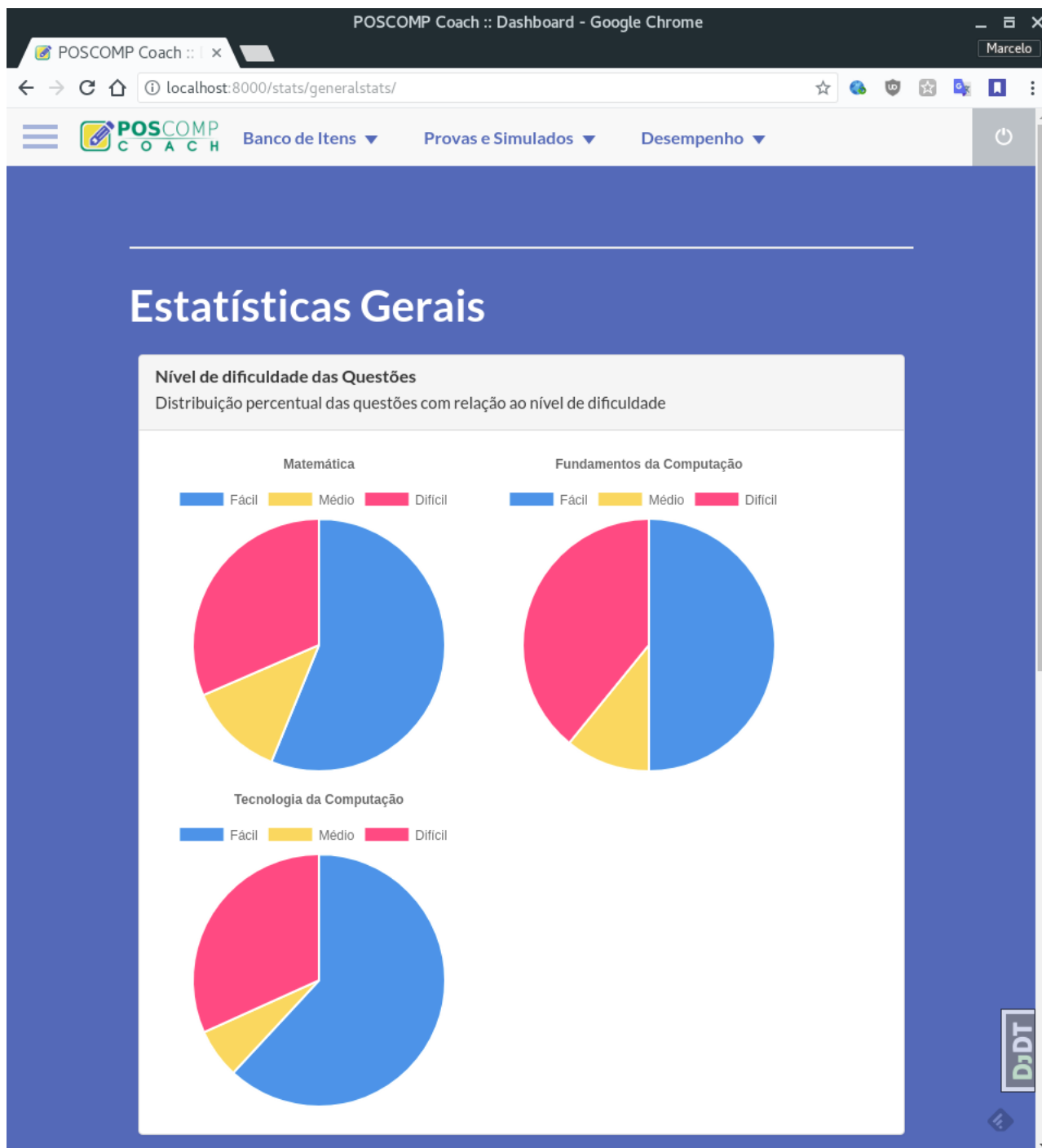
The screenshot shows a web application interface for POSCOMP Coach. The browser address bar indicates the URL is localhost:8000/base/exams/. The page has a blue header with the POSCOMP COACH logo and navigation links: Banco de Itens, Provas e Simulados, and Desempenho. The main content area is titled 'Exames' and contains a table of previous exams with all questions. The table has three columns: Título, Realizador, and Ano/Edição. A tooltip labeled 'Detalhes' is visible over the first row. The footer shows the URL localhost:8000/base/exam/46/.

Título	Realizador	Ano/Edição
POSCOMP		2002
POSCOMP		2003
POSCOMP		2004
POSCOMP		2005
POSCOMP		2006
POSCOMP		2007
POSCOMP		2008
POSCOMP		2009
POSCOMP		2010
POSCOMP		2011
POSCOMP		2012
POSCOMP		2013
POSCOMP		2014
POSCOMP		2015

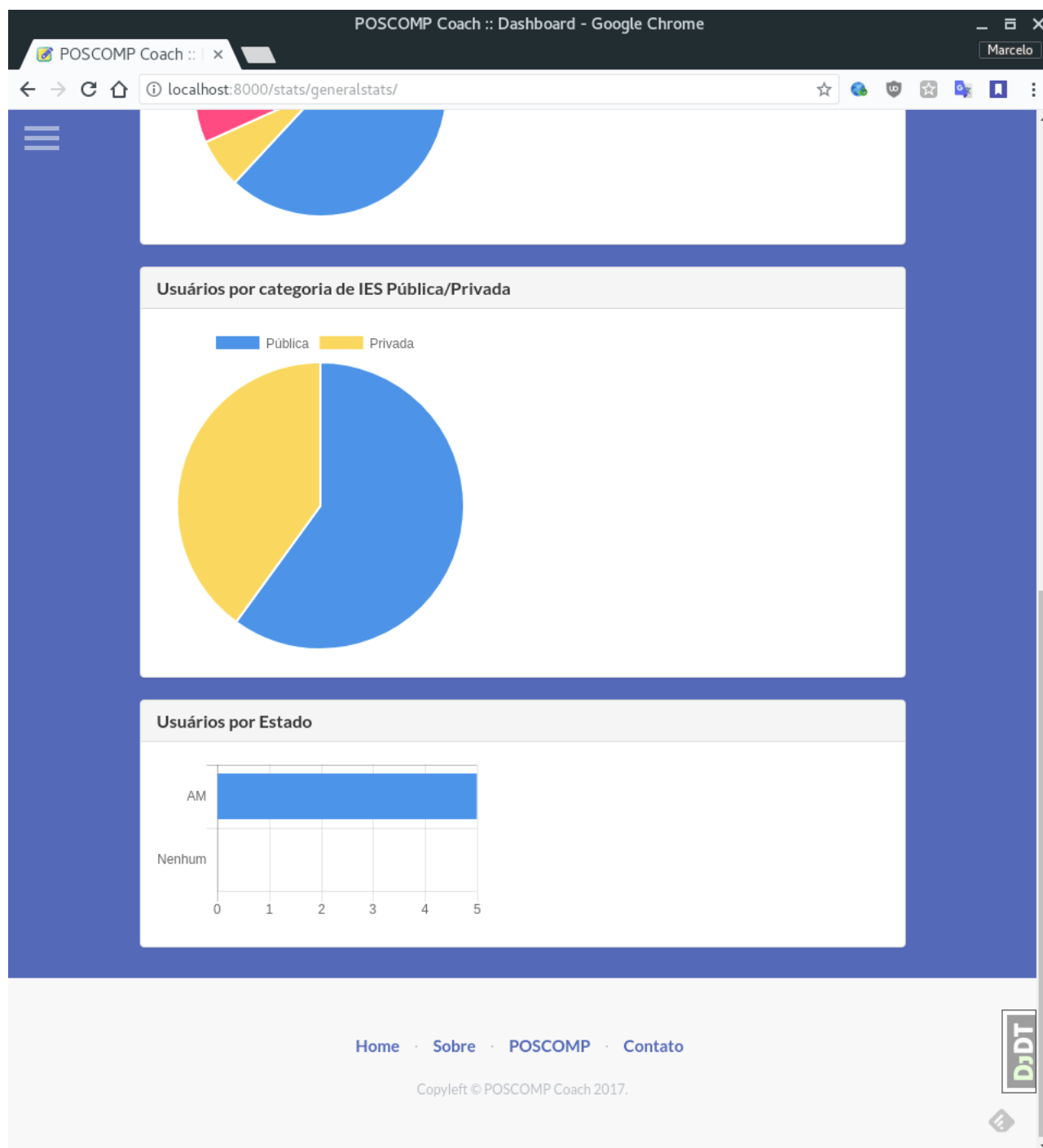
Fonte: Elaborado pelo autor.

Nas Figuras 38 e 39 são mostrados os gráficos estatísticos gerais, como dificuldade dinâmica das questões, baseado na quantidade de acertos/erros quando esta é selecionada e estudantes por Instituição Pública/Privada, e também o número de usuários por estado.

Figura 38 – UI - POSCOMP Coach: Estatísticas gerais - Nível de dificuldade das questões



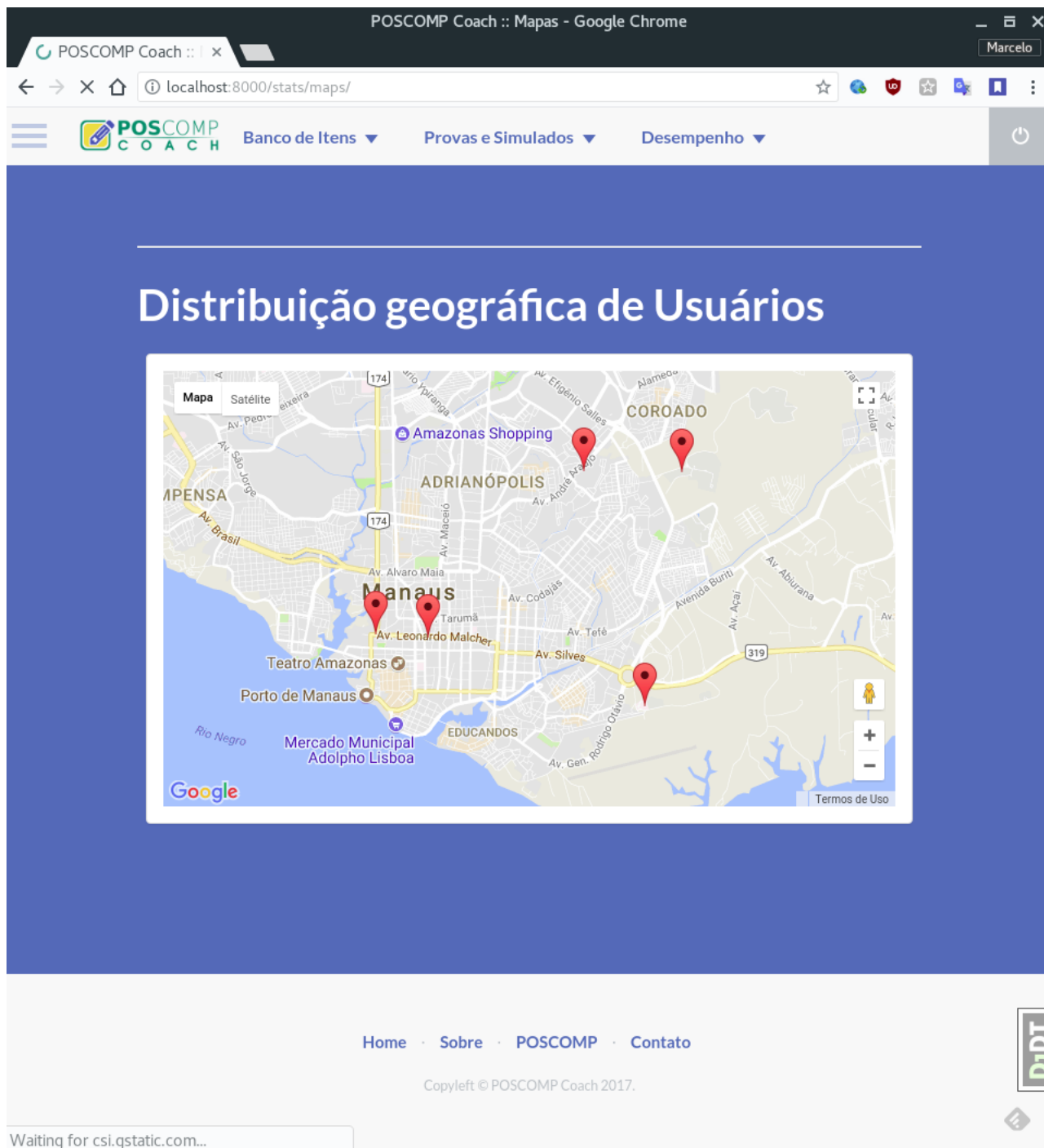
Fonte: Elaborado pelo autor.

Figura 39 – UI - POSCOMP *Coach*: Estatísticas gerais - Usuários por IES Pública/Privada.

Fonte: Elaborado pelo autor.

Na Figura 40 é apresentado ao administrador da plataforma um mapa da distribuição geográfica dos usuários, baseado na Instituição de Ensino selecionada perfil.

Figura 40 – UI - POSCOMP *Coach*: Distribuição geográfica dos usuários.



Fonte: Elaborado pelo autor.

APÊNDICE E – Diagramas Complementares

APÊNDICE F – Controle de acesso, publicidade e visibilidade de atributos em Python

A maioria das linguagens de programação orientadas a objetos possuem um conceito de controle de acesso de métodos e atributos. Na linguagem Java¹, por exemplo, atributos e métodos de um objeto podem ser marcados como *private*, *protected* ou *public*. Quando marcado como *private* indica que esse atributo ou método pode ser acessado somente pelo objeto. Se for marcado como *protected* seus membros serão acessados pela classe e subclasses. Por fim eles podem ser marcados como *public*, indicando que qualquer classe pode acessar seus atributos e métodos. No caso de nenhuma marcação, indica que o acesso está restrito às classes do mesmo pacote (*package*) ([Oracle, Java Documentation - The Java™ Tutorials, 2017](#)).

Em Python esse tipo de controle não está presente. Em vez de regras que obrigam o programador a definir essas restrições, a linguagem estipula *guidelines* e boas práticas. Por convenção, um nome prefixado com um *underscore* (exemplo: `__nome`) deve ser tratado como um membro não-público, seja ele um método ou um atributo, mas não existe nenhum tipo de restrição no interpretador que impeça o acesso. Desta forma, tecnicamente todos os métodos e atributos de uma classe estão acessíveis publicamente ([PHILLIPS, 2015](#)).

Devido algumas situações bem específicas, onde um pouco mais de controle é necessário, como por exemplo para evitar conflito de nomes definidos por uma subclasse, existe um mecanismo chamado de *name mangling*, onde qualquer identificador, atributo ou método, que for prefixado por pelo menos dois (2) *underscores* (exemplo: `__nome`), é substituído por `__nomedaclasse__nome`, onde `__nomedaclasse` é a classe à qual pertence `__nome`.

É importante frisar, porém, que esse mecanismo foi projetado para evitar conflitos de nomes e situações similares. Portanto, mesmo utilizando-se desta funcionalidade, ainda é possível acessar ou modificar o atributo considerado privado ([Python Software Foundation, 2017](#)).

No aprendizado de linguagens como Java, é enfatizado que nunca se deve acessar atributos diretamente, e sim através de métodos públicos, conhecidos como *getters* e *setters*. A razão para isto, é que para os casos em que haja a necessidade de adicionar algum código antes que o atributo seja acessado ou definido, essa modificação seja realizada de

¹ <https://docs.oracle.com/javase/8/docs/technotes/guides/language/index.html>

forma que não haja quebra da API.

No entanto, esse pensamento não faz sentido em Python, uma vez que a linguagem provê um mecanismo para esta situação. Esse mecanismo é chamado *decorator* `@property`. Esta funcionalidade permite ao desenvolvedor escrever seu código normalmente utilizando acesso direto aos membros da classe, e caso algo inesperado surja, como a necessidade de incluir algum código, de validação, por exemplo que precise executar antes de o atributo ser acessado ou definido, será possível readequar o código de forma eficiente sem alterar a interface (PHILLIPS, 2015).

Na Figura 42 é apresentado o exemplo: uma classe `Pessoa` foi definida, e esta possui um único atributo `nome`. Uma alteração foi necessária, e agora o atributo `nome` não deverá mais aceitar entradas em caixa alta. Esta funcionalidade é facilmente alcançada utilizando o mecanismo `@property`, as classes clientes não precisam sequer tomar conhecimento da mudança, e a API permanece inalterada, de forma que `pessoa.nome = "ALGUM NOME"` continuará funcionando normalmente.

Figura 42 – Exemplo de atributo com `@property` em Python

```
1 # Classe normal
2 class Pessoa:
3     def __init__(self, nome):
4         self.nome = nome
5
6 # Uso de @property
7 class Pessoa:
8     def __init__(self, nome):
9         self.nome = nome
10
11     # Método getter, poderia ter uma lógica aqui
12     @property
13     def nome(self):
14         return self.__nome
15
16     # Método setter
17     # Caso o parâmetro nome esteja em caixa alta ele
18     # será mudado para caixa baixa antes de definido
19     # no atributo nome
20     @nome.setter
21     def nome(self, nome):
22         if nome.isupper():
23             self.__nome = nome.lower()
```

Fonte: Elaborado pelo autor.

O exemplo exposto na Figura 42 é simples e descreve uma situação trivial, tem o intuito puro e simplesmente de ilustrar o mecanismo `@property` da linguagem Python. Em uma aplicação de médio porte com milhares de linhas de código, porém, esse mecanismo é extremamente útil, uma vez que dá liberdade ao desenvolvedor de seguir com um código limpo, na forma como as boas práticas e a filosofia da linguagem Python convencionam, evitando criação de métodos *getters* e *setters* triviais que apenas “incham” o código, ao mesmo tempo em que dá poder ao desenvolvedor de, em caso de necessidade, realizar a adequação de código com eficiência, e principalmente, mantendo a consistência da API.

Anexos

ANEXO A – POSCOMP Coach - Formulário de Avaliação.

POSCOMP Coach - Avaliação

Prezado(a) Estudante,

Desenvolvemos uma plataforma denominada POSCOMP Coach com o objetivo de auxiliar estudantes de computação na preparação para o POSCOMP - exame realizado pela Sociedade Brasileira de Computação (SBC) para avaliar candidatos a cursos de Pós-Graduação em Computação no Brasil e que é utilizado por várias universidades brasileiras no processo de seleção de candidatos para mestrado e doutorado em Computação.

Com o POSCOMP Coach, os candidatos poderão ter acesso as provas já realizadas, compor simulados e acompanhar o seu próprio desempenho por meio de estatísticas e gráficos.

Neste momento, precisamos de sua colaboração para avaliar a plataforma do ponto de vista da usabilidade. Esta avaliação exigirá que você utilize o POSCOMP Coach por um tempo mínimo de 30 minutos e depois responda a um questionário. Sua avaliação será fundamental para melhorarmos o POSCOMP Coach e o disponibilizarmos para a comunidade.

Ao efetuar o preenchimento e envio deste questionário para os autores, você concorda com a utilização dos dados informados em pesquisas relacionadas ao POSCOMP Coach. Destacamos que não haverá qualquer identificação das pessoas que responderem ao questionário.

O POSCOMP Coach está disponível em <https://poscompcoach.com/>

Agradecemos sua participação!

***Obrigatório**

1. Você utilizou as funcionalidades do POSCOMP Coach por pelo menos 30 minutos?

Marcar apenas uma oval.

☐

Sim

Ir para a pergunta 2.

☐

Não

Pare de preencher este formulário.

Questionário

2. Nome:

Esta informação não é obrigatória. Identifique-se apenas caso se sinta à vontade.

3. Email:

Esta informação não é obrigatória. Pedimos esta informação apenas para o caso de um possível contato futuro ou para encaminhar os resultados desta pesquisa.

4. Nome do Curso de Graduação que está cursando ou cursou:

Marcar apenas uma oval.

- ☐ Ciência da Computação
- ☐ Sistemas de Informação
- ☐ Engenharia da Computação
- ☐ Engenharia Elétrica
- ☐ Engenharia de Software
- ☐ Licenciatura em Informática
- ☐ Tecnologia em Análise e Desenvolvimento de Sistemas
- ☐ Tecnologia em Redes de Computadores
- ☐ Outro: _____

5. Qual sua situação no curso?

Marcar apenas uma oval.

- ☐ Iniciante (no primeiro ano do curso)
- ☐ No meio do Curso
- ☐ Finalista (último ano do curso)
- ☐ Formado
- ☐ Outro: _____

6. Região do país em que cursa ou cursou a graduação.

Marcar apenas uma oval.

- ☐ Norte
- ☐ Nordeste
- ☐ Centro-Oeste
- ☐ Sul
- ☐ Sudeste

7. Critério de Usabilidade - Falar na linguagem do usuário. *

Marcar apenas uma oval por linha.

	Concordo Totalmente	Concordo	Indeciso	Discordo	Discordo Totalmente
O POSCOMP Coach possibilita o entendimento dos termos utilizados na interface.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

8. Critério de Usabilidade – Minimizar carga de memória do usuário **Marcar apenas uma oval por linha.*

	Concordo Totalmente	Concordo	Indeciso	Discordo	Discordo Totalmente
O número de etapas para a criação de um simulado por ano é pequena.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O número de etapas para a criação de um simulado por área é pequena.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O número de etapas para a criação de um simulado por preferência é pequena	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O número de etapas para retomar um simulado já criado é pequena	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O número de etapas para consultar o resultado do meu desempenho é pequena	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

9. Critério de Usabilidade – Ser consistente **Marcar apenas uma oval por linha.*

	Concordo Totalmente	Concordo	Indeciso	Discordo	Discordo Totalmente
O POSCOMP Coach é confiável – apresenta frequência de falhas baixa ou inexistente?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Os botões, símbolos, menus e demais itens são similares entre uma tela e outra?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O POSCOMP Coach permite fácil recuperação de dados em casos de falhas?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O POSCOMP Coach oferece informações suficientes para orientá-lo (a) durante a navegação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

10. Critério de Usabilidade – Prover feedback **Marcar apenas uma oval por linha.*

	Concordo Totalmente	Concordo	Indeciso	Discordo	Discordo Totalmente
O POSCOMP Coach possui status para acompanhamento das tarefas requeridas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O POSCOMP Coach sempre informa de forma rápida e clara, quando uma determinada ação esta sendo ou foi concluída	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

11. Critério de Usabilidade – Saídas claramente marcadas e atalhos *

Marcar apenas uma oval por linha.

	Concordo Totalmente	Concordo	Indeciso	Discordo	Discordo Totalmente
O POSCOMP Coach possibilita opções de escape ou cancelamento, utilizando teclas fazer – desfazer	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O POSCOMP Coach permite teclas de atalhos para agilizar a navegação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

12. Critério de Usabilidade – Mensagens de erros construtivas e precisas *

Marcar apenas uma oval por linha.

	Concordo Totalmente	Concordo	Indeciso	Discordo	Discordo Totalmente
O POSCOMP Coach produz mensagens de erro claras e propõem dicas para sua resolução	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
As mensagens de erro apresentadas pelo POSCOMP Coach são sempre claras, precisas e bastante explicativas com relação à natureza dos erros cometidos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O POSCOMP Coach sempre indica, por exemplo, através de mensagens, que você está inserindo uma informação errada ou procedendo de maneira inadequada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

13. Critério de Usabilidade – Compatibilidade do POSCOMP Coach com o mundo real *

Marcar apenas uma oval por linha.

	Concordo Totalmente	Concordo	Indeciso	Discordo	Discordo Totalmente
O POSCOMP Coach se comunica de forma clara com frase e termos familiares ao usuário	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Os termos e expressões utilizados no POSCOMP Coach apresentam uma linguagem de fácil compreensão e entendimento	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

14. Critério de Usabilidade – Estética *

Marcar apenas uma oval por linha.

	Concordo Totalmente	Concordo	Indeciso	Discordo	Discordo Totalmente
Os botões, imagens e textos sempre aparecem organizados e agrupados a partir de algum critério lógico? (por exemplo: ordem alfabética, frequência de uso, etc.)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Os nomes dos botões e janelas são concisos, ou seja, breves e precisos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O número de informações apresentado na tela é aceitável, tornando fácil o entendimento do sistema	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

15. Critério de Usabilidade – Prevenção de erros *

Marcar apenas uma oval por linha.

	Concordo Totalmente	Concordo	Indeciso	Discordo	Discordo Totalmente
O POSCOMP Coach disponibiliza em sua navegação, mensagens de alerta quando vamos fazer um procedimento inadequado ou inserindo uma informação errada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

16. Caso estivesse se preparando para o exame do POSCOMP com que frequência você utilizaria a plataforma? *

Marcar apenas uma oval.

- ☐ Algumas vezes
- ☐ Muitas vezes
- ☐ Quase sempre
- ☐ Sempre
- ☐ Não usaria esta plataforma

17. De modo geral, que nota você daria para o POSCOMP Coach? *

Marcar apenas uma oval.

[illegible]

18. Eu indicaria o POSCOMP Coach a amigos ou colegas de curso *

Marcar apenas uma oval.

- ☐ Concordo Fortemente
- ☐ Concordo
- ☐ Não concordo nem discordo
- ☐ Discordo
- ☐ Discordo Fortemente

19. Há alguma funcionalidade que você considera importante e que não viu no POSCOMP Coach?

20. Espaço destinado a comentários ou sugestões.

Powered by

