
Quero Minha Nota

Estimando a nota do ENEM através de um algoritmo colaborativo

Por

EDUARDO FELIPE EWERT BONET



Departamento de Automação e Sistemas
UNIVERSIDADE FEDERAL DE SANTA CATARINA

Monografia submetida à Universidade Federal de Santa Catarina de acordo com os requerimentos da matéria de graduação PROJETO DE FIM DE CURSO, no curso de Engenharia de Controle e Automação.

JULHO 2015

Quero Minha Nota

Estimando a nota do ENEM através de um algoritmo colaborativo

Por

EDUARDO FELIPE EWERT BONET

Esta monografia foi julgada no contexto da disciplina
DAS5511: PROJETO DE FIM DE CURSO
e aprovada na sua forma final pelo
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Professor Doutor Ubirajara Franco Moreno

BANCA EXAMINADORA

PROFESSOR DOUTOR RICARDO DE ABREU MADEIRA
Orientador na Tuneduc

PROFESSOR DOUTOR UBIRAJARA FRANCO MORENO
Orientador no Curso

Avaliador

Debatedores

RESUMO

O Exame Nacional do Ensino Médio (ENEM) é uma prova padronizada feita por estudantes de todo o Brasil normalmente logo após finalizarem o Ensino Médio. Seu principal objetivo é fornecer um panorama geral da performance dos alunos brasileiros, de forma que professores e os próprios estudantes possam melhorar suas metodologias de ensino e estudo. Em 2009, a prova passou por uma grande reformulação, visando solucionar o problema onde resultados poderiam apenas ser comparados entre aplicantes da mesma prova, dado que a pontuação dependia muito da dificuldade de um exame específico. Para resolver isso, novo método para calcular as notas passou a ser empregado, denominado Teoria de Resposta ao Item. Porém, o processo de calcular as notas usando TRI é baseado em uma análise estatística complexa, criando um novo problema em que estudantes dificilmente conseguem ter uma noção de seu resultado mesmo sabendo quais questões ele acertou. Esse projeto apresenta o QUERO MINHA NOTA, um sistema que usa algoritmos colaborativos e técnicas de análise de dados para dar aos aplicantes uma estimativa de suas notas logo após terem saído da prova.

ABSTRACT

Enem is a standardized test applied every year in Brazil, taken by students usually upon completion of high school. Its main objective is to provide a general vision on the performance of Brazilian students, so that both schools and students themselves could improve their teaching and studying methodology accordingly. In 2009, there was a major overhaul on its structure, aimed to address the problem where results could only be compared within the same exam edition, given that the score was dependent on the difficulty of that specific exam. To solve this, a different approach for scoring was employed, named Item Response Theory, or simply IRT. However, the process of computing scores using IRT is based on a complex statistical analysis, creating an issue where students can rarely have a notion of their scores even knowing which questions they answered correctly. This project introduces QUERO MINHA NOTA, a system that makes use of collaborative algorithms and data science techniques to give these test takers an estimated grade right after leaving the exam room.

DEDICATÓRIA

Dedico essa monografia à minha família, especialmente aos meus pais que me permitiram simplesmente estudar, e que cuja maior herança repassada foi a educação.

Dedico também à minha esposa Amanda Lins Bonet, que me deu suporte durante todos esses anos do curso, e que mostrou extrema paciência em inspecionar detalhadamente esse documento em busca dos infinitos erros de português.

Por último, dedico à equipe da Tuneduc, que confiou na minha capacidade de entregar esse ambicioso projeto mesmo estando em outra região do país.

DECLARAÇÃO DO AUTOR

Declaro que todo trabalho nessa monografia foi desenvolvido por mim, salvo quando feito explícito que uma ferramentas ou trabalho desenvolvido por terceiros ou com a ajuda de terceiros foi utilizado.

ASSINATURA: DATA:

TABELA DE CONTEÚDOS

	Página
Lista de Tabelas	xi
Lista de Figuras	xiii
1 Teoria de Resposta ao item - TRI	3
1.1 Introduzindo a Teoria de Resposta ao Item	4
1.2 Modelagem Matemática do Problema	4
1.2.1 O Modelo Logístico de 3 Parâmetros (3PL)	5
1.2.2 Os Parâmetros do Modelo 3PL	6
1.2.3 Calculando as proficiências	10
1.2.4 Outras aplicações do TRI	11
2 Estimando notas no ENEM	13
2.1 O ENEM	13
2.2 Dados Utilizados	14
2.2.1 Tratamento dos Dados	14
2.2.2 Estimando as notas	16
2.3 Resultados	18
2.4 Comentários	26
3 Quero Minha Nota	27
3.1 Análise de Requisitos	28
3.1.1 Requisitos Funcionais	28
3.1.2 Requisitos Não Funcionais	28
3.2 Topologia	30
3.3 Camada de Interação	33
3.3.1 Funcionalidades da camada de Interação	33
3.3.2 Web	35
3.4 Camada de Interface	35
3.5 Camada de Persistência	38

TABELA DE CONTEÚDOS

3.5.1	Modelo do Banco de Dados	38
3.5.2	Gerenciamento de Atualização de Notas e Parâmetros	39
3.6	Camada de Computação	42
	Bibliografia	45

LISTA DE TABELAS

TABELA	Página
2.1 Variáveis resultantes da transformação dos dados crus	15
2.2 Variáveis necessárias para o estimar a nota do ENEM de um aluno	17
2.3 Comparação entre as distribuições dos erros residuais com os diferentes ajustes . . .	23
2.4 Comparação entre as distribuições dos erros residuais com os diferentes ajustes . . .	24
3.1 Colunas da tabela Aplicante	40
3.2 Colunas da tabela Escola	40
3.3 Colunas da tabela Cidade	40
3.4 Colunas da tabela Mapeamento	41

LISTA DE FIGURAS

FIGURA	Página
1.1 Exemplos de Curvas Características para três questões diferentes	6
1.2 Efeito da variação do parâmetro a nas curvas características, ao manter constantes os parâmetros b e c	7
1.3 Efeito da variação do parâmetro b nas curvas características, ao manter constantes os parâmetros a e c	8
1.4 Efeito da variação do parâmetro c nas curvas características, ao manter constantes os parâmetros a e b	8
1.5 Significado gráfico dos parâmetros de uma curva característica	9
2.1 Diferença na distribuição das notas na prova de matemática entre a população total e uma amostra de 100.000 aplicantes no ENEM 2012	18
2.2 Correlação entre as proficiências reais e as calculadas para a prova de Matemática para o Enem 2012	18
2.3 Correlação entre as proficiências reais e as calculadas para a prova de Matemática para o Enem 2012 com o ajuste de média e desvio padrão ($\mu = 500$ e $\sigma^2 = 100$)	19
2.4 Erro residual $ \theta - \theta' $ ($\mu = 500$ e $\sigma^2 = 100$)	19
2.5 Densidade e Distribuição Acumulada para o erro $ \theta - \theta' $ com ($\mu = 500$ e $\sigma^2 = 100$)	20
2.6 Correlação entre as proficiências reais e as calculadas para a prova de Matemática para o Enem 2012 com o ajuste de média e desvio padrão ($\mu = 505$ e $\sigma^2 = 120$)	21
2.7 Erro residual entre as proficiências reais e estimadas com ajuste na escala ($\mu = 505$ e $\sigma^2 = 120$)	21
2.8 Densidade e Distribuição Acumulada para o erro $ \theta - \theta' $ com ($\mu = 505$ e $\sigma^2 = 120$)	22
2.9 Resultados para a prova de Ciências Naturais	23
2.10 Resultados para a prova de Ciências Humanas	24
2.11 Resultados para a prova de Linguagens	25
3.1 Representação da arquitetura funcional do Quero Minha Nota, com sua divisão em camadas	30
3.2 Representação da arquitetura do Quero Minha Nota, especificando quais tecnologias são utilizadas em cada camada.	31

3.3	Exemplo da diferença de estratégias de escalabilidade	32
3.4	Protótipo da interface para dispositivos móveis - Login e Cadastro	35
3.5	Protótipo da interface para dispositivos móveis - Menu e Gabarito	36
3.6	Protótipo da interface para dispositivos móveis - Gabarito e Estatísticas	36
3.7	Diagrama Relacional do Banco de Dados	39

INTRODUÇÃO

O Exame Nacional do Ensino Médio [4], mais conhecido como ENEM, é uma prova padronizada cujo objetivo é ser uma ferramenta para avaliar o desempenho dos estudantes brasileiros ao completar o ensino médio. Tendo sua primeira edição em 1998, passou por forte reformulação em 2009 para que, entre outros motivos, pudesse passar a ser utilizado como forma de seleção para a entrada em instituições de ensino superior, substituindo ou complementando as provas de vestibular.

Uma das principais modificações introduzidas nessa nova versão é o abandono do cálculo das notas do aluno pelo método clássico (número de acertos) e a adoção da Teoria de Resposta ao Item (TRI). Essa troca teve como principais objetivos permitir que resultados de provas diferentes sejam comparáveis e criar a possibilidade de várias provas por ano [10].

Pelo método antigo, a nota do aplicante dependia fortemente da dificuldade da prova naquele ano. Esse fator impossibilitava, por exemplo, que um aluno que tivesse feito a prova em 2006 usasse seu resultado num processo de seleção de uma universidade em 2008, pois as medidas seriam diferentes. O TRI traz, entre muitas vantagens, uma escala comparativa que resolve esse problema e permite usar o ENEM como mecanismo de seleção para o Ensino Superior.

No entanto, o TRI traz duas grandes desvantagens, tanto para o aplicante quanto para as escolas. Para as escolas, pelo processo antigo elas poderiam melhorar sua qualidade de ensino focando em assuntos que seus alunos tiveram maior taxa de erro. Porém, segundo esse novo método isso não é tão simples, pois o acerto ou erro de um questão tem um efeito diferente para cada aluno.

Já para os aplicantes, é muito difícil ter uma expectativa da sua nota, mesmo tendo conhecimento de quais questões ele acertou, dado que ela não é mais calculada com base em repostas independentes, mas a partir do conjunto das suas respostas e do conjunto das questões de todos os outros aplicantes. Por exemplo, o acerto de uma questão por dois alunos com padrões de resposta diferentes não implica obrigatoriamente em mesmo aumento na nota.

Com o objetivo de minimizar esse problema criou-se o Quero Minha Nota, um sistema ainda em desenvolvimento que tenta dar ao aplicante uma expectativa do seu resultado logo após a realização da prova, precisando apenas de seu gabarito. Esse sistema possui um caráter colaborativo, ou seja, para estimar a nota de um usuário levamos em conta os resultados de todos os outros aplicantes.

Antes de entrar em detalhes sobre o Quero Minha Nota, primeiro será feita no Capítulo 1 uma análise completa sobre o funcionamento do TRI, quais seus pressupostos, vantagens e dificuldades. Entraremos em detalhes também sobre como o acerto ou erro de uma questão impacta nas notas finais.

No Capítulo 2, o conhecimento desenvolvido no Capítulo 1 será utilizado para verificar a viabilidade de estimar notas do Enem, usando para isso os microdados fornecidos pelo governo referentes às provas de 2012.

No Capítulo 3, será realizada uma análise completa sobre o desenvolvimento do Quero Minha Nota. Falaremos sobre os problemas técnicos encontrados na implementação da solução, e quais estratégias utilizamos para trazer a melhor experiência possível para o usuário.

TEORIA DE RESPOSTA AO ITEM - TRI

Ao aplicar uma prova ou teste de múltipla escolha, o método mais comum de medir a performance de um aluno é atribuir uma pontuação específica para cada questão e somar os pontos das questões que o aluno acertou. Essa é a chamada teoria clássica de testes, conhecida também como CTT [22], do inglês *Classical Test Theory*. Apesar de simples e prático, esse método traz três grandes problemas: garantir que o aluno realmente domina o conteúdo de uma questão que ele acertou, pontuar objetivamente as questões dependendo do seu nível de dificuldade e ter a capacidade de comparar dois testes com diferentes dificuldades.

No primeiro caso, um aplicante pode ter acertado uma questão sem dominar o conhecimento necessário. É o chamado falso positivo. Um exemplo claro disso é o "chute", que é muito comum em provas de múltipla escolha, e fere a real avaliação de desempenho do aluno na prova em questão.

Além disso, numa prova de múltipla escolha questões normalmente não possuem pesos diferentes, apesar de indicarem níveis diferentes de habilidade e conhecimento para resolvê-las. E quando esses pesos existem, são dados a partir de uma análise subjetiva sobre o grupo de questões, o que pode levar a uma estimativa errônea das habilidades de cada aluno.

Por último, se a nota é dada baseada no número de questões corretas, não se pode comparar o resultado de dois aplicantes em testes diferentes, pois eles podem possuir dificuldades diferentes. Por exemplo, um aluno que acertou 80% das questões no Teste 1 não necessariamente sabe mais que o que acertou 60% no Teste 2, sendo que existe a possibilidade de o Teste 1 ter sido muito

mais fácil que o Teste 2.

Por isso, faz-se necessário um método que analise objetivamente a real proficiência de cada aluno, capaz de detectar falsos positivos (quando um aplicante acerta a questão sem ter o conhecimento necessário para isso). Nesse contexto surgiu a Teoria de Resposta ao Item.

1.1 Introduzindo a Teoria de Resposta ao Item

Na tentativa de resolver as restrições criadas pelo CTT apresentadas anteriormente, foi apresentada durante os anos 50 uma nova proposta para a medir a proficiência em exames: a Teoria de Resposta ao Item [18], ou TRI, que traz uma proposta de cálculo das proficiências através análise probabilística do cenário da prova como um todo. Nesse sentido, o TRI tem como base comparar as respostas de cada aplicante em cada questão com as respostas de todos os outros aplicantes, e inferir quais questões são mais difíceis ou mais fáceis bem como quais as notas mais prováveis com base nessas comparações.

O primeiro aspecto que o TRI tenta identificar em um padrão de respostas do aplicante é consistência. Quando se tem um aplicante que erra muitas questões de nível fácil ou mediano, mas acerta uma muito difícil, a probabilidade de que esse acerto na muito difícil seja um falso positivo é alta. Já se o aluno acerta consistentemente questões até uma dificuldade específica, existe alta probabilidade de que ele tenha acertado as questões deliberadamente, não por acaso. Ambos os aplicantes receberão pontos por terem acertado a questão, mas o impacto é maior para o aluno com maior consistência.

Outro princípio explorado pelo TRI é considerar as dificuldades das questões como variáveis do sistema, sendo calculadas junto com a proficiência dos alunos. Ou seja, tanto as dificuldades de cada questão quanto a proficiência de cada aplicante são calculadas a posteriori, isto é, a dificuldade de cada questão é obtida a partir do nível de proficiência dos alunos que a acertaram e as proficiências a partir das dificuldades das questões que cada aluno acertou.

1.2 Modelagem Matemática do Problema

O primeiro passo para resolver o problema de avaliação de uma forma mais matemática é descrever a probabilidade que um aplicante com proficiência θ tem de acertar um item através de uma função $P(\theta)$. Chamaremos essa função de curva característica do item ou questão.

Nesse estudo, utilizaremos o modelo conhecido como "3PL", por ser o utilizado pelo INEP no cálculo das notas do ENEM. Existem diversos outros modelos disponíveis na literatura, como

"2PL", "4PL" e "Rasch"[21]. A escolha de um ou outro depende de quais aspectos do item deseje-se representar, bem como da importância dada a cada aspecto ou fator. Detalhes sobre outros modelos podem ser encontrados em [19].

1.2.1 O Modelo Logístico de 3 Parâmetros (3PL)

Para encontrar a função que melhor representa um item, é preciso primeiro analisar quais os fatores que influenciam a probabilidade de um aplicante responder uma questão da forma com que respondeu. Para o modelo de 3 Parâmetros, como o nome diz, são consideradas três características: dificuldade, diferenciação e acerto ao acaso.

Naturalmente, quanto maior a proficiência de um aluno maior deve ser a chance de ele acertar uma questão. Com isso, sabemos que a curva é monotonamente crescente, ou seja, se $x > y \Leftrightarrow f(x) > f(y)$. Analogamente, quanto mais difícil a questão, maior será a proficiência necessária para existir uma chance de acerto razoável.

Além disso, existem questões que representam melhor o conteúdo analisado, fazendo com que um aluno que tenha acertado uma delas tenha maior chance de ter uma proficiência maior do que o que não acertou. Por outro lado, questões mal elaboradas ou que englobam uma fração pequena da habilidade analisada (em um teste com muitas questões por exemplo), não causam tanta diferença na possível proficiência do aluno.

Por último, existe sempre a possibilidade de o aluno acertar a questão sem ter proficiência nenhuma, especialmente em provas de múltipla escolha. A curva característica deve identificar também a taxa de acertos que não são explicados pela proficiência do aluno, ou a probabilidade de acerto ao acaso. A Figura 1.1 mostra algumas curvas características que consideram os fatores mencionados.

Nessa seção, considera-se que a distribuição de θ tenha média 0 e desvio padrão 1. Dessa forma, na Figura 1.1, tem-se três curvas características distintas. Na curva em vermelho, um aluno mediano com $\theta = 0$ tem em torno de 30% de chance de ter acertado a questão. Porém, qualquer aluno tem chance mínima de 18% de acertar a questão, não importa sua proficiência. Observa-se também que para um aluno ter uma probabilidade de 70% de ter acertado a questão, ele devia ter uma proficiência de pelo menos 2. Essa curva representaria uma questão difícil com uma chance relativamente alta de acerto por "chute".

Porém, na curva verde alunos de proficiência baixa dificilmente acertariam a questão. Aluno medianos tem exatamente 50% de chance de acertar a questão, enquanto que alunos com proficiência em torno de 2 tem quase 100% de chance de ter acertado. Esse é o caso de uma

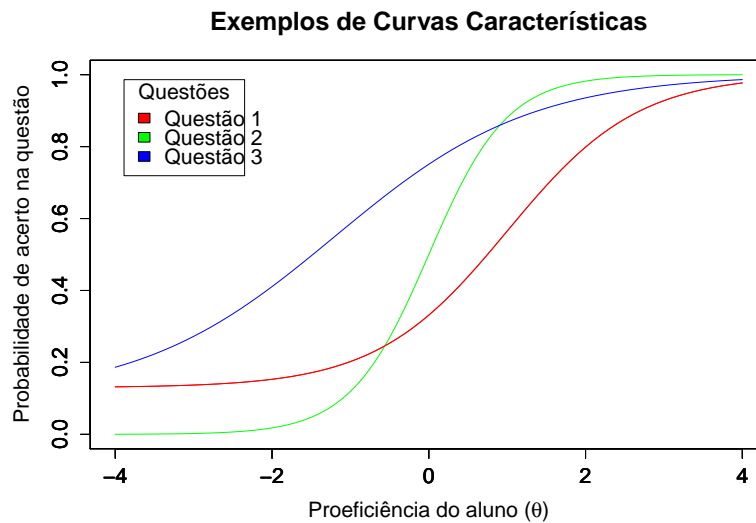


Figura 1.1: Exemplos de Curvas Características para três questões diferentes

questão ideal que identifica quase perfeitamente alunos acima ou abaixo da média.

Na questão em azul a curva tem uma inclinação muito baixa, sendo pouco elástica com relação à proficiência θ . Quando menor a inclinação, menor é o fator de explicação daquela questão na proficiência do aluno.

1.2.2 Os Parâmetros do Modelo 3PL

Como o nome diz, o modelo 3PL usa três parâmetros para descrever a curva característica: a (grau de diferenciação), b (dificuldade) e c (acerto ao acaso).

- PARÂMETRO a : Grau de diferenciação da questão

O parâmetro a mede o grau de diferenciação de uma questão, ou seja, o impacto que acertar ou não o item causa na proficiência. Isso é representado por uma subida mais íngreme ou mais lenta na curva, onde quanto maior o valor de a , maior a diferença de proficiências entre aplicantes que acertaram e erraram a questão. A Figura 1.2 mostra a a curva característica para diferentes valores de a .

Na Figura 1.2, a curva vermelha possui uma inclinação muito forte. Por causa disso, um aplicante com $\theta = -1$ possui probabilidade quase nula de acerto, enquanto que é quase certo que um com $\theta = 1$ tenha escolhido a opção correta. Já para um valor de a mediano, como o da curva verde,

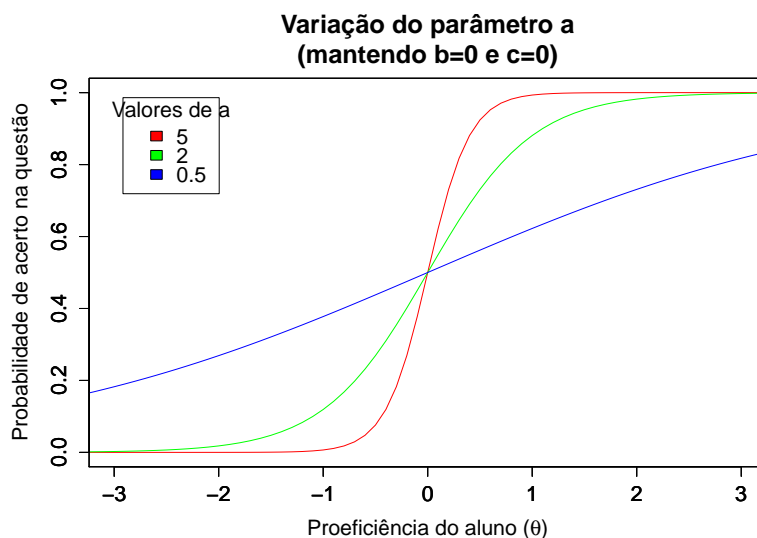


Figura 1.2: Efeito da variação do parâmetro a nas curvas características, ao manter constantes os parâmetros b e c .

a diferença de probabilidade de acerto entre os aplicantes com $\theta = 1$ e $\theta = -1$ fica em torno de 70%. No caso da curva azul, essa diferença é muito menor, pouco mais de 20%. Ou seja, é muito mais difícil afirmar que o aluno que acertou o item azul tem uma proficiência maior do que o que errou, enquanto que para o item vermelho essa correlação é quase certa.

- PARÂMETRO b : Dificuldade da questão

O parâmetro b representa a dificuldade de uma questão. Quanto maior o valor de b , mais difícil é a questão e maior a proficiência requerida para uma probabilidade de acerto. Para isso, o parâmetro b desloca a curva característica no eixo θ . O resultado da variação do valor de b é mostrado na Figura 1.3.

Na Figura 1.3, para o item representado pela curva vermelha, um aplicante com $\theta = 0$ tem 90% de probabilidade de acerto. Em contrapartida, esse mesmo aplicante tem 50% de acertar o item com a curva em verde, que possui um valor maior para b . Observando a curva em azul, com um b maior ainda, esse aplicante teria apenas 10% de probabilidade de acertar a questão.

- PARÂMETRO c : Probabilidade de acerto ao acaso.

Para modelar a probabilidade de acerto que não é explicado pela proficiência do aluno,

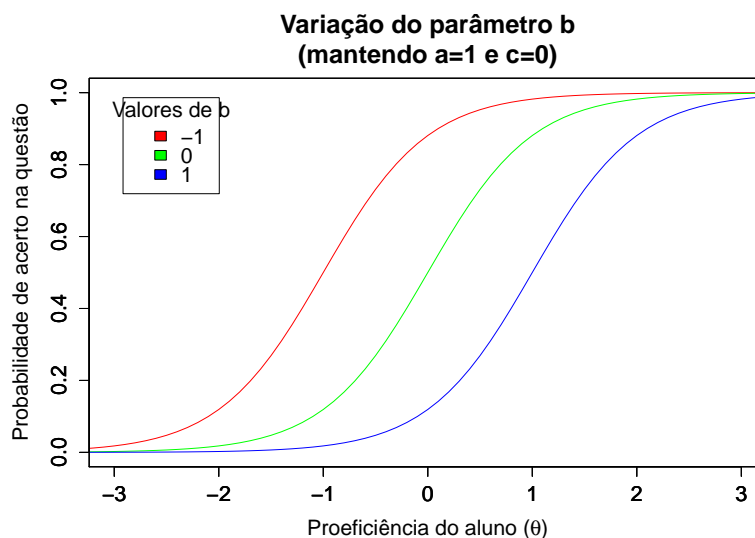


Figura 1.3: Efeito da variação do parâmetro b nas curvas características, ao manter constantes os parâmetros a e c .

usaremos o parâmetro c . O parâmetro c simplesmente desloca o ponto 0 da função para cima, como mostrado na Figura 1.4. Com isso, por menor que seja a proficiência do aluno, ele ainda apresentará uma chance mínima de acerto.

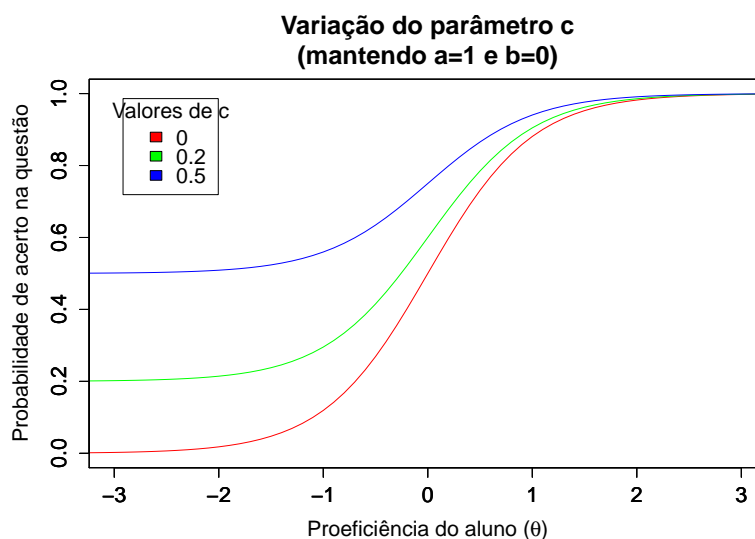


Figura 1.4: Efeito da variação do parâmetro c nas curvas características, ao manter constantes os parâmetros a e b .

Na Figura 1.4, não importa o quão baixa é a proficiência de um aplicante ele sempre terá pelo menos 50% de chance de acertar o item com curva em azul, o que não acontece para o item em vermelho, que possui $c = 0$.

1.2.2.1 Obtendo Graficamente os Parâmetros de uma questão

Quando se tem a curva característica, pode-se obter os parâmetros da questão graficamente. A Figura 1.5 descreve o significado gráfico de cada um dos parâmetros.

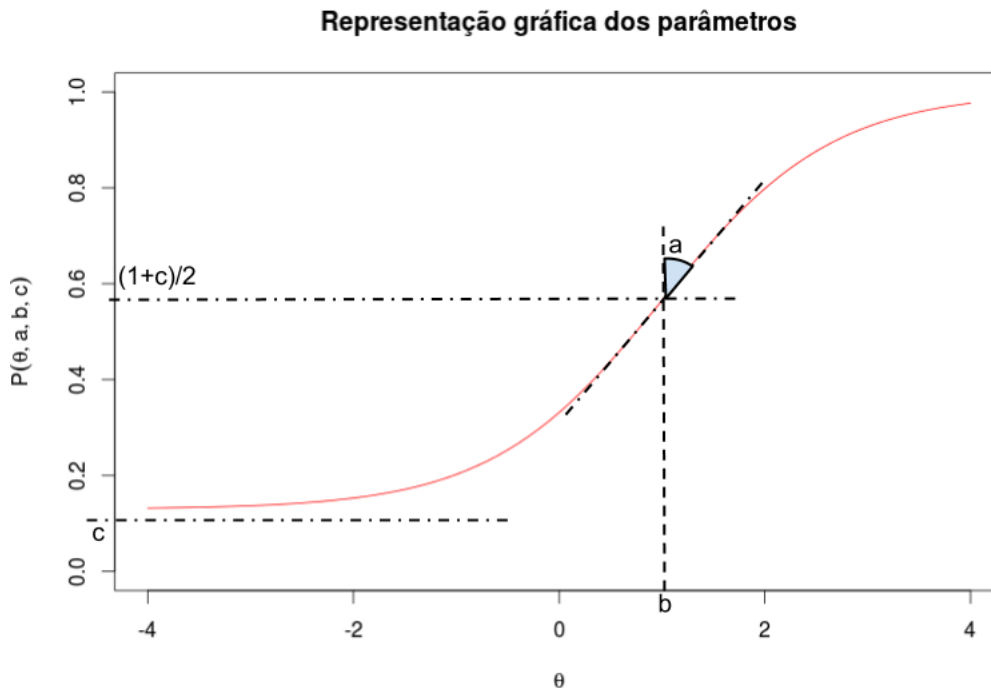


Figura 1.5: Significado gráfico dos parâmetros de uma curva característica

O parâmetro c é a probabilidade de acerto do aluno que possui $\theta = -\infty$, ou seja, é o menor valor de probabilidade possível, ou então a probabilidade para o ponto onde a derivada da função θ é a menor possível.

O parâmetro b é igual à proficiência do aluno que possui uma proficiência θ_b cuja probabilidade de acerto equivale a $P(\theta) = (1 + c)/2$, que é ponto de maior inclinação da curva característica.

Finalmente, o parâmetro α é a derivada da curva no ponto de maior inclinação, que é o mesmo de onde se obtém o valor de b .

1.2.2.2 Equacionando o 3PL

Sabendo o que cada um dos parâmetros significa, a Equação 1.1 é a equação que representa a curva característica da questão em definida pelos parâmetros a , b e c em função da proficiência θ .

$$(1.1) \quad P(\theta, a, b, c) = c + \frac{1 - c}{1 + e^{-a(\theta - b)}}$$

Analogamente, a função que descreve a probabilidade de um aluno com proficiência θ ter errado uma questão é dada pela Equação 1.2

$$(1.2) \quad Q(\theta, a, b, c) = 1 - P(\theta, a, b, c)$$

1.2.3 Calculando as proficiências

Após de definir o modelo utilizado para uma questão, através da escolha da curva característica, segue-se para o problema de como calcular as proficiências dos alunos pelo TRI.

Supõe-se que uma prova com n questões foi respondida por m aplicantes. Nessa prova os aplicantes ou erraram ou acertaram cada uma questão, considerando questões deixadas em branco ou com mais de uma alternativa marcada como erradas, assim como é feito no ENEM.

Não há preocupação sobre qual alternativa o aplicante marcou, apenas se ele acertou o não a questão. As respostas de cada aluno s_i são guardadas em um vetor resposta u_i , de ordem n , onde o elemento u_{ij} representa o acerto ou erro do aluno s_i à questão a_j . Esses vetores resposta são então agrupados em uma matriz $U_{m \times n}$, chamada de matriz resposta.

As variáveis do sistema são os vetores θ , onde θ_i é a proficiência do aluno s_i , e os vetores a , b e c onde cada elemento a_j , b_j , e c_j correspondem aos parâmetros da questão q_j , que são desconhecidos.

Para estimar as variáveis, utiliza-se o método da máxima verossimilhança, que visa otimizar os parâmetros e proficiências de forma a maximizar ao máximo a probabilidade da matriz resposta observada.

Assim, a Equação 1.3 define o objetivo de otimização para o TRI.

$$(1.3) \quad \max \prod_m \prod_n P(\theta_i, a_j, b_j, c_j)^{u_{ij}} \times Q(\theta_i, a_j, b_j, c_j)^{1-u_{ij}}$$

O primeiro termo da multiplicação só faz efeito quando o aplicante acertou a questão, ou seja, quando $U_{ij} = 1$. Da mesma forma, o segundo termo só aparece quando $U_{ij} = 0$, quando o aluno errou a questão.

Porém, para uma alta quantidade de questões ou aplicantes, a probabilidade se torna tão pequena que o computador não consegue mais representar o valor. Por isso, modifica-se a equação de otimização fazendo uso do logaritmo natural de cada uma das probabilidades, transformando os produtórios em somatórios, sem alterar a natureza da interação entre as variáveis. Assim, a versão final do objetivo de otimização é dado na Equação 1.4.

$$(1.4) \quad \max \sum_m \sum_n \ln(P(\theta_i, a_j, b_j, c_j)^{u_{i,j}} \times Q(\theta_i, a_j, b_j, c_j)^{1-u_{i,j}})$$

Caso queira-se usar algum outro modelo, como o "Rasch" por exemplo, basta substituir na função de otimização as equações das curvas características $P(\theta)$ e $Q(\theta)$ pela equações específicas do modelo escolhido.

1.2.4 Outras aplicações do TRI

Apesar existir aqui um foco no uso do TRI para medir conhecimento de alunos sobre um determinado assunto em uma prova, essa não é a única aplicação possível.

[16] discute o uso do TRI para identificar proficiência em ambientes hospitalares, como por exemplo atendimentos clínicos.

O TRI é amplamente também utilizado na psicologia para estudar os resultados de pesquisas de opinião, por exemplo. [17] traz uma análise completa do uso do TRI para identificar quais os melhores métodos de aprendizado para um conjunto de alunos, e em como identificar quais suas áreas cognitivas fortes e fracas.

ESTIMANDO NOTAS NO ENEM

Tendo conhecimento das premissas e propostas da Teoria de Resposta ao Item, mostradas no Capítulo 1, podemos agora partir para um exemplo prático: estimar as notas do ENEM a partir de dados oficiais publicados pelo INEP.

2.1 O ENEM

Como discutido, a partir de 2009 o ENEM passou por uma grande reformulação e, dentre outras medidas, passou a adotar a Teoria de Resposta ao Item para o cálculo das proficiências. No entanto, a análise feita pelo INEP possui alguns diferenciais à aquela apresentada no Capítulo 1.

As proficiências no ENEM não possuem média 0 e desvio padrão 1 como consideramos no estudos dos parâmetros do TRI. Segundo o guia do estudante divulgado pelo INEP [7], um aluno que tirou uma nota 500 tem a mesma proficiência que a média dos alunos de que completaram o terceiro ano em escolas públicas no ano de 2009. Além disso considera-se que o desvio padrão desse alunos foi 100.

O ENEM é, na verdade, uma composição de 5 provas: Ciências da Natureza, Ciências Humanas, Matemática, Linguagens e Redação. Como o cálculo da nota de Redação é feita a partir de um outro tipo de análise, está fora do escopo deste trabalho. Os cálculos das notas das provas de matemática, ciências humanas e ciências naturais são feitos utilizando TRI seguindo o procedimento apresentado no Capítulo 1.

Porém, estimar a nota da prova de Linguagens traz uma dificuldade a mais: os alunos podem escolher se respondem às questões da prova de Língua Inglesa ou da prova de Língua Espanhola, embora essa diferenciação não apareça no resultado final. Nesse caso, existem 40 questões que todos os aplicantes fizeram, e cujos parâmetros não devem mudar entre os dois grupos, mas um conjunto de 5 questões específicas para cada conjunto.

2.2 Dados Utilizados

Para desenvolver o algoritmo de estimativa, foram usados os microdados dos ENEM de 2012 e 2013 publicados pelo INEP. Esses dados são de alta granularidade, informando inclusive qual alternativa cada aluno marcou em cada questão, mas sem identificar quem é o aluno, apenas em que escola ele estuda e qual ano cursava quando fez a prova.

Esses dados são liberados normalmente 1 ano após a realização da prova, embora esse período seja inconstante e possa ser muito maior (os dados referentes ao Enem de 2013 foram disponibilizados somente no final de Maio de 2015). Eles estão disponíveis para download público no link [2].

Tanto em 2012 quanto em 2013 os dados da realização da prova (questões marcadas, id dos alunos, etc) foram incluídos em um único arquivo no formato csv. A tabela referente a 2012 possui cerca de 5,8 milhões de linhas, uma linha por aplicante, e cerca de 160 colunas trazendo as mais diversas informações. Para 2013, foram mais de 7,1 milhões de entradas, com 166 colunas.

Para facilitar a leitura das tabelas, junto aos arquivos com os dados crus é fornecido um dicionário de colunas, que explica o significado e o formato de cada uma das informações fornecidas.

As colunas dividem-se em 3 grupos principais: informações sobre o estudo aluno (escola, se é pública ou privada, ano em que completou o ensino médio), informações do questionário sócio-econômico do aplicante e informações sobre a prova do aplicante (cor da prova, gabarito, presença, alternativas marcadas).

2.2.1 Tratamento dos Dados

Embora exista uma grande quantidade de informação disponível, só uma porção deles é realmente necessária, a que é utilizada para o cálculo da Teoria de Resposta ao Item. A Tabela 2.1 mostra quais são os dados e variáveis fundamentais para que se possa fazer o cálculo das proficiências.

Para conseguirmos fazer a análise TRI, os dados vitais da Tabela 2.1 são os vetores resposta das matérias objetivas, que são uma sequência de 45 valores binários, indicando o acerto ou erro de cada questão. No caso da prova de Linguagens, os 40 primeiros valores correspondem às

Nome da Variável	Tipo	Descrição
id	numérica	ID do aluno no ENEM
escolaPublica	0 ou 1	Escola pública ou privada
concluiu2012	0 ou 1	Ano de conclusão do ensino médio
nota_LI	numérica	Nota na prova de Linguagens
nota_CN	numérica	Nota na prova de C. Naturais
nota_CH	numérica	Nota na prova de C. Humanas
nota_MT	numérica	Nota na prova de Matemática
presenca_LI	binário	Presente na prova de Linguagens
presenca_CN	binário	Presente na prova de C. Naturais
presenca_CH	binário	Presente na prova de C. Humanas
presenca_MT	binário	Presente na prova de Matemática
respostas_LI	vetor binário	Vetor respostas em Linguagem
respostas_CN	vetor binário	Vetor respostas em C. Naturais
respostas_CH	vetor binário	Vetor respostas em C. Humanas
respostas_MT	vetor binário	Vetor respostas Matemática
lingua	0 ou 1	Língua Enstrangeira escolhida

Tabela 2.1: Variáveis resultantes da transformação dos dados crus

questões de língua portuguesa, e as 5 últimas às questões da língua estrangeira. Para essa prova, a variável *lingua* indica qual o a língua estrangeira que cada aluno escolheu.

Aplicantes que faltaram em alguma das provas (indicados pelas variáveis de presença) são desconsiderados no cálculo das proficiências naquela prova, pois o fato de um aluno não ter nenhum acerto por não ter ido não implica que a probabilidade de que um aplicante erre todos as questões seja maior.

As variáveis de *escolaPublica* e *concluiu2012* são usadas para selecionar quais os padrões de respostas podem ser usados no cálculo das notas, seguindo a regra do INEP que apenas alunos concluintes no Ensino Médio do ano da prova provindos de escola pública tem seus dados usados para estimar os parâmetros TRI.

Após aplicar o algoritmo, a variável *id* permite parear entrada e saída, enquanto as variáveis de *nota* são usadas para validar as estimativas.

Uma vez que os microdados não são fornecidos nesse formato, é preciso transformar as informações originais para a tabela acima. As variáveis *id*, *escolaPublica*, *concluiu2012*, *nota*, *presenca* e *lingua* tem correspondentes diretos nos microdados originais.

Porém, encontrar o vetor resposta das provas requer mais esforço. Primeiramente, nos dados

originais as respostas dos alunos são fornecidas como um vetor onde cada entrada é alternativa marcada pelo aplicante na questão, junto com um outro vetor com as alternativas corretas da prova que o aluno fez. Comparando os dois vetores conseguimos encontrar um vetor binário de acertos e erros.

No entanto, os vetores resultantes se referem à cor específica da prova do aplicante, o que implica em ordem diferente de questões. A questão 24 da prova amarela pode ser a questão 36 da prova azul, por exemplo. Fazer o mapeamento de provas é impossível somente com as tabelas fornecidas nos microdados, mas em anexo a elas o INEP fornece também todas as provas, de todas as cores, permitindo fazer esse processo manualmente.

Tendo o dicionário de variáveis como base, para conseguir os dados mencionados acima precisaríamos das informações contidas nas colunas mostradas na tabela 2.2. A nomenclatura das colunas se refere ao ano de 2012, sendo que para o ano de 2013 alguns nomes sofreram alteração, mas as naturezas das variáveis são as mesmas.

2.2.2 Estimando as notas

A linguagem escolhida para prototipar o cálculo do TRI foi a linguagem R [20]. R é uma linguagem desenvolvida para computação estatística, que ganhou grande adoção tanto pela academia quanto por empresas que trabalham com dados.

Além disso, R possui uma biblioteca de código livre feita especificamente para rodar análises TRI. A biblioteca `mirt`[15] está em contínuo desenvolvimento por Phil Chalmers, e possui forte suporte acadêmico.

Outra opção considerada inicialmente era a biblioteca `Guacamole`[6], desenvolvida como parte da iniciativa Open Source da Khan Academy. Essa biblioteca implementa o TRI usando Python de forma altamente escalável.

Porém, a biblioteca `mirt` tem um caráter mais exploratório e generalístico, permitindo testar diferentes equações características e customizadas, em adição a um grande conjunto de funções que facilitam a análise dos resultados, de grande utilidade no momento de prototipação.

Apesar de possuir uma série de funções extremamente interessantes para o desenvolvimento do algoritmo, o R traz uma grande restrição: para que muitas de suas funções funcionem com a facilidade desejada, ele opera totalmente em memória RAM. Lembrando que a tabela usada tem um tamanho aproximado de 6gb, mesmo um computador potente teria problema só em carregar

Nome da Variável	Tipo	Descrição
NU_INSCRIÇÃO	numérica	ID do aluno no ENEM
ID_DEPENDENCIA_ADM	enum	Escola é privada, estadual, federal ou municipal
ST_CONCLUSAO	enum	Ano de conclusão do ensino médio
NU_NT_LI	numérica	Nota na prova de Linguagens
NU_NT_CN	numérica	Nota na prova de C. Naturais
NU_NT_CH	numérica	Nota na prova de C. Humanas
NU_NT_MT	numérica	Nota na prova de Matemática
IN_PRESENÇA_LI	binário	Presente na prova de Linguagens
IN_PRESENÇA_CN	binário	Presente na prova de C. Naturais
IN_PRESENÇA_CH	binário	Presente na prova de C. Humanas
IN_PRESENÇA_MT	binário	Presente na prova de Matemática
TX_RESPOSTAS_LI	vetor de caracteres	Alternativas marcadas em Linguagem
TX_RESPOSTAS_CN	vetor de caracteres	Alternativas marcadas em C. Naturais
TX_RESPOSTAS_CH	vetor de caracteres	Alternativas marcadas em C. Humanas
TX_RESPOSTAS_MT	vetor de caracteres	Alternativas marcadas em Matemática
ID_PROVA_LI	inteiro	Cor da prova de Linguagens
ID_PROVA_CN	inteiro	Cor da prova de C. Naturais
ID_PROVA_CH	inteiro	Cor da prova de C. Humanas
ID_PROVA_MT	inteiro	Cor da prova de Matemática
TP_LINGUA	0 ou 1	Língua Estrangeira escolhida
DS_GABARITO_LI	Vetor de Caracteres	Gabarito da prova de Linguagens
DS_GABARITO_CH	Vetor de Caracteres	Gabarito da prova de C. Humanas
DS_GABARITO_CN	Vetor de Caracteres	Gabarito da prova de C. Naturais
DS_GABARITO_MT	Vetor de Caracteres	Gabarito da prova de Matemática

Tabela 2.2: Variáveis necessárias para o estimar a nota do ENEM de um aluno

os dados para a memória, e ainda mais para operar sobre eles.

A solução mais aconselhada para essa situação é trabalhar com amostras. Escolhendo 100.000 participantes de forma aleatória, a distribuição das notas se mantém a mesma, como mostrado na Figura 2.1, mesmo com o valor 100.000 sendo uma fatia relativamente pequena do total .

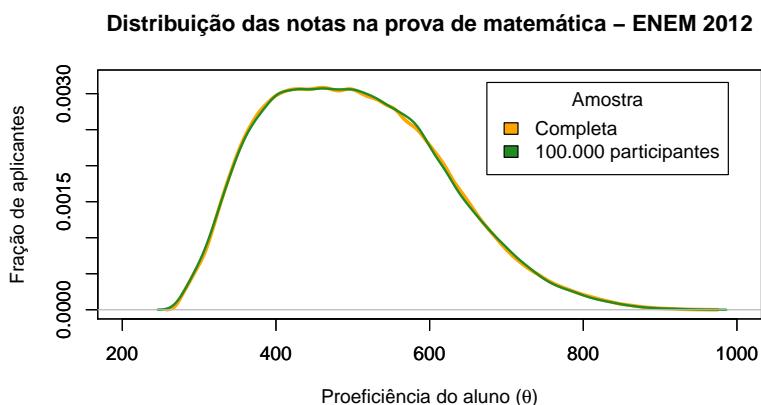


Figura 2.1: Diferença na distribuição das notas na prova de matemática entre a população total e uma amostra de 100.000 aplicantes no ENEM 2012

2.3 Resultados

A seguir é apresentado um estudo mais detalhado dos resultados obtidos utilizando a prova de Matemática de 2012. O resultado para as outras provas é descrito em sequência.

Após o tratamento e seleção dos dados, utiliza-se a biblioteca mirt para fazer a otimização do TRI. A relação entre a nota obtida e a nota real é mostrada na Figura 2.2. O algoritmo foi configurado para rodar por no máximo 500 iterações ou enquanto a diferença máxima entre duas proficiências calculadas em iterações consecutivas for menor que 10^{-5} .

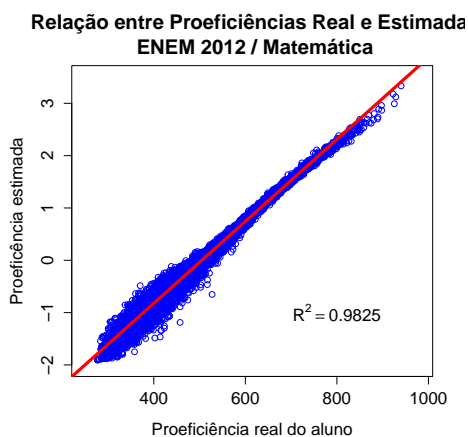


Figura 2.2: Correlação entre as proficiências reais e as calculadas para a prova de Matemática para o Enem 2012

Apesar de existir erro, a aproximação por mínimos quadrados entre as proficiências reais e estimadas possui um R^2 de 0.9825, indicando que elas possuem uma forte correlação linear. A

linha em vermelho indica o modelo linear que melhor aproxima as duas funções.

No entanto, não é desejável que proficiências estimadas possuam média 0 e desvio padrão 1. Falar para um aplicante que ele teve média 0 ou média negativa pode inclusive trazer a ideia errada de que ele não acertou nenhuma questão. Por isso, o resultado precisa ser traduzido em uma escala que faça sentido. Inicialmente, define-se como média a proficiência 500 e o desvio padrão como 100. O resultado é mostrado na Figura 2.3.

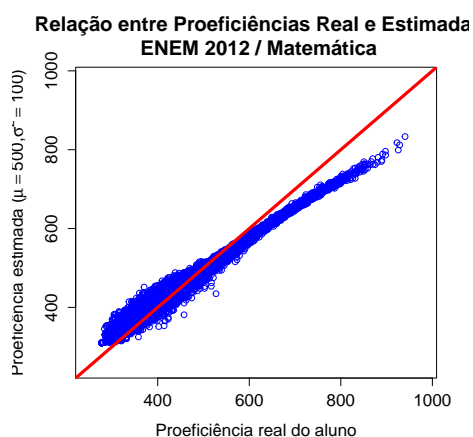


Figura 2.3: Correlação entre as proficiências reais e as calculadas para a prova de Matemática para o Enem 2012 com o ajuste de média e desvio padrão ($\mu = 500$ e $\sigma^2 = 100$)

A linha em vermelho indica nesse caso os pontos de correlação perfeita. Observa-se erro alto devido a diferença nas escalas, com erro padrão de 27.75 pontos. A Figura 2.4 mostra o erro absoluto.

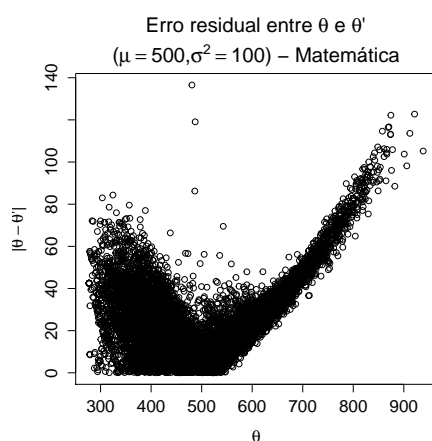
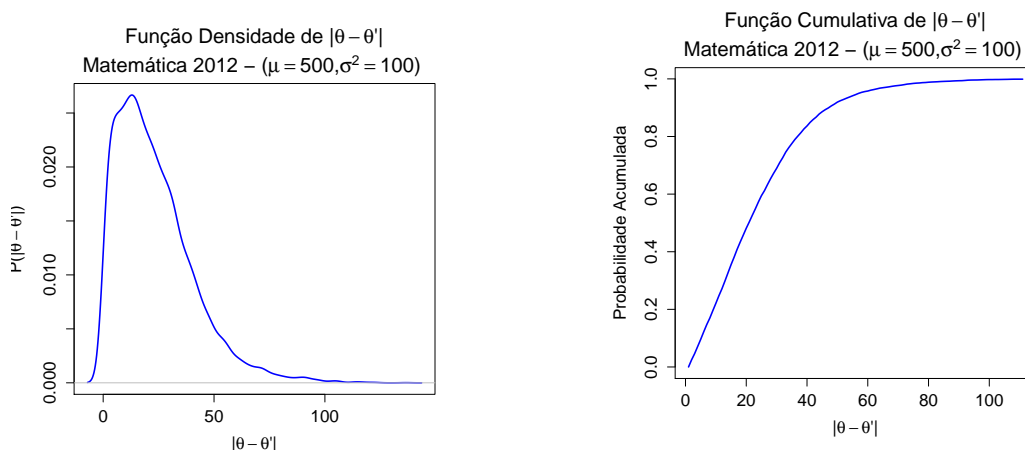


Figura 2.4: Erro residual $|\theta - \theta'|$ ($\mu = 500$ e $\sigma^2 = 100$)

A visualização do erro na Figura 2.4 não traz muita informação além do máximo e mínimo. Para observar melhor o erro geral, a Figura 2.5 mostra a distribuição do erro. Pela Figura 2.5(a), observa-se que, apesar de existirem aplicantes que apresentam erro maior do que 100 pontos, grande parte das estimativas possuem erro menor que 50. Para esse ajuste, obteve-se um erro padrão de 27.75, o que indica que 95% das estimativas possui erro menor que 53.4, relação que pode facilmente ser verificada na distribuição acumulada da Figura 2.5(b).



((a)) Densidade do erro $|\theta - \theta'|$ ($\mu = 500$ e $\sigma^2=100$) ((b)) Distribuição acumulada do erro $|\theta - \theta'|$ ($\mu = 500$ e $\sigma^2=100$)

Figura 2.5: Densidade e Distribuição Acumulada para o erro $|\theta - \theta'|$ com ($\mu = 500$ e $\sigma^2=100$)

Porém, a média 500 e desvio padrão 100 se referem apenas aos alunos de escola pública que fizeram a prova em 2009 e finalizaram o ensino médio naquele ano. Na realidade, ao observar os dados aos longos dos anos, os valores reais são bem diferentes. Por exemplo, para a prova de matemática a média fica em 505 e o desvio padrão em 120. Esse desvio padrão é especialmente alto porque a média das escolas particulares (que não entra no cálculo) é bem mais alta. O resultado no novo ajuste de escala é mostrado na Figura 2.6, e o erro residual é mostrado na Figura 2.7.

As distribuições do erro para esse caso são mostradas na Figura 2.8. Observa-se um erro muito menor, o que é confirmado pela Figura 2.8(b), onde 95% das estimativas possuem erro absoluto menor que 25 pontos.

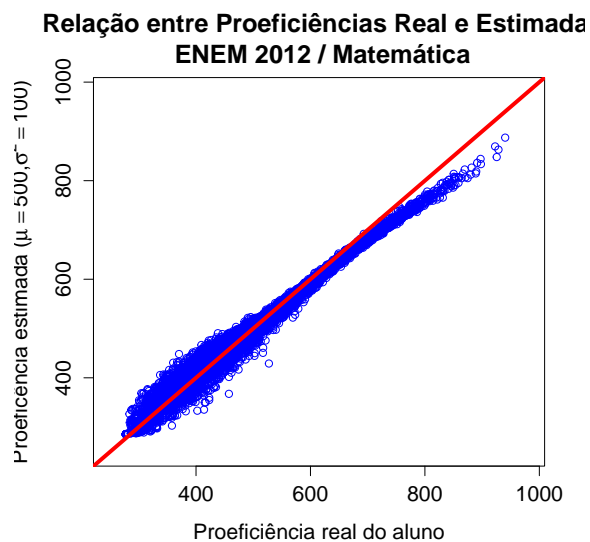


Figura 2.6: Correlação entre as proficiências reais e as calculadas para a prova de Matemática para o Enem 2012 com o ajuste de média e desvio padrão ($\mu = 505$ e $\sigma^2 = 120$)

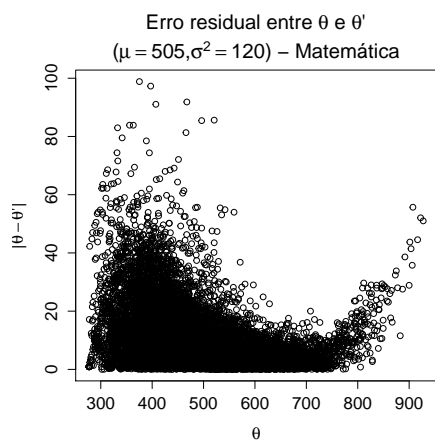
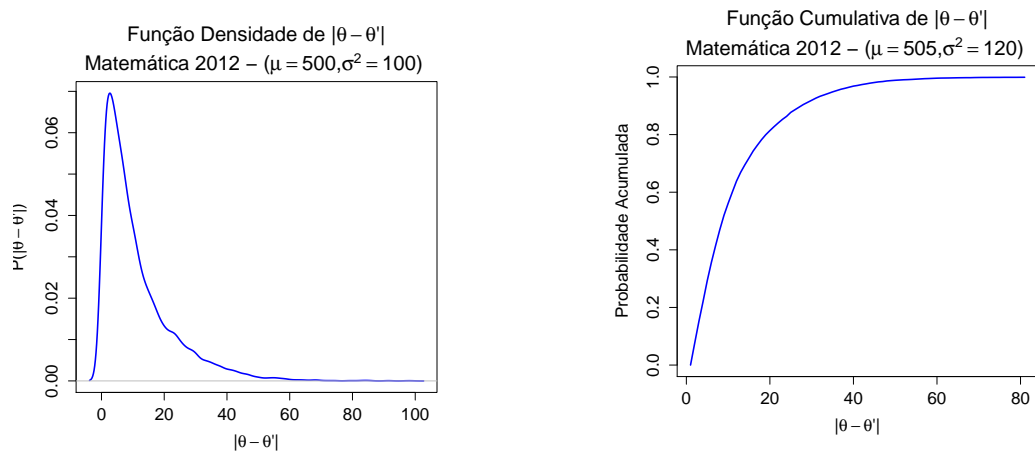


Figura 2.7: Erro residual entre as proficiências reais e estimadas com ajuste na escala ($\mu = 505$ e $\sigma^2 = 120$)



((a)) Densidade do erro $|\theta - \theta'|$ ($\mu = 505$ e $\sigma^2=120$) ((b)) Distribuição acumulada do erro $|\theta - \theta'|$ ($\mu = 505$ e $\sigma^2=120$)

Figura 2.8: Densidade e Distribuição Acumulada para o erro $|\theta - \theta'|$ com ($\mu = 505$ e $\sigma^2=120$)

A Tabela 2.3 mostra as relações entre os residuais com o ajuste de escala com ($\mu = 500$, $\sigma^2 = 100$), ($\mu = 505$, $\sigma^2 = 120$) e o melhor ajuste possível, obtido através do método dos mínimos quadrados entre θ e θ' .

Ajuste	Mínimo	Média	Mediana	Máximo	Desvio Padrão
Modelo Linear	0	10.86	7.46	104.30	15.25
($\mu = 500$, $\sigma^2 = 100$)	0	22.70	19.94	126.40	27.75
($\mu = 505$, $\sigma^2 = 120$)	0	13.86	11.08	98.43	17.98

Tabela 2.3: Comparação entre as distribuições dos erros residuais com os diferentes ajustes

A estimativa para as outras provas segue a mesma lógica para o ajuste da escala. Os resultados para a prova de Ciências Naturais, Ciências Humanas e Linguagens são mostrados nas Figuras 2.9, 2.10 e 2.11, respectivamente.

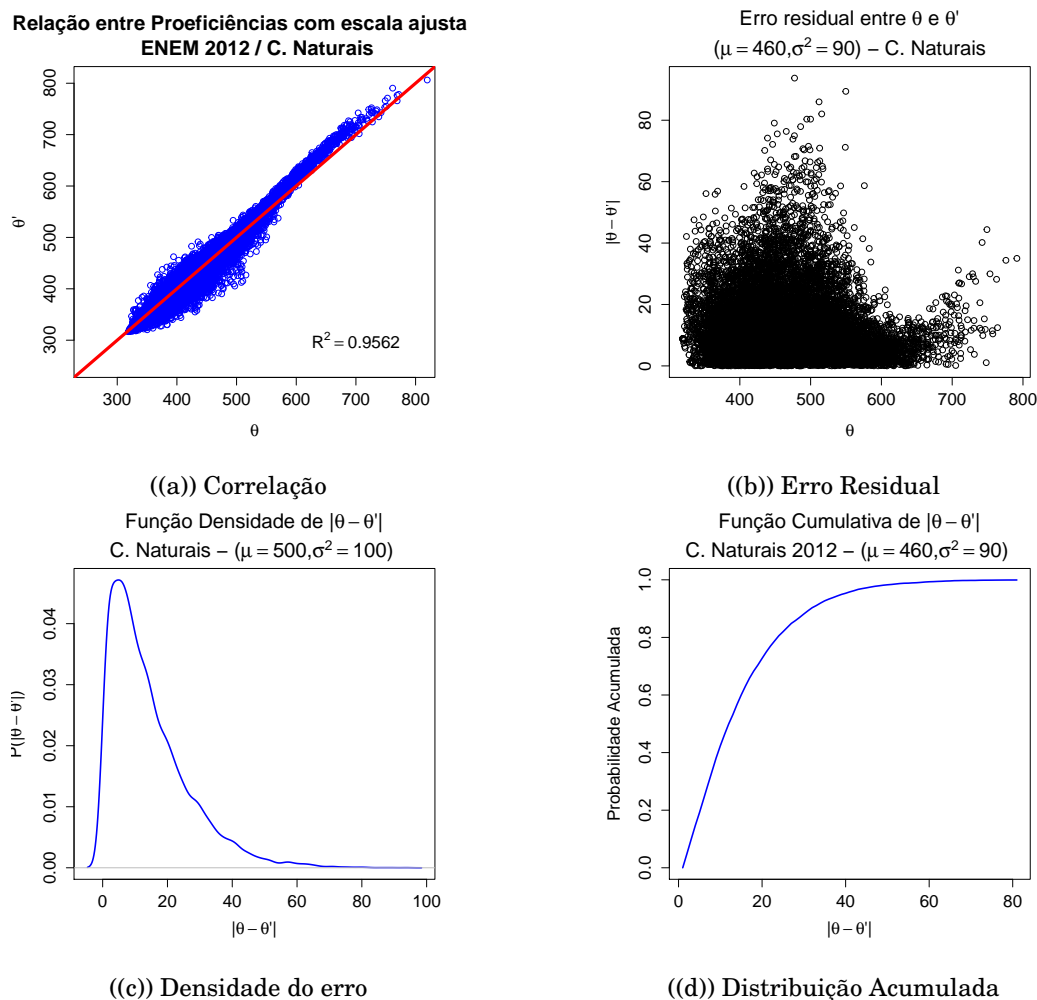


Figura 2.9: Resultados para a prova de Ciências Naturais

A Tabela 2.4 traz os resultados numéricos encontrados para as três provas.

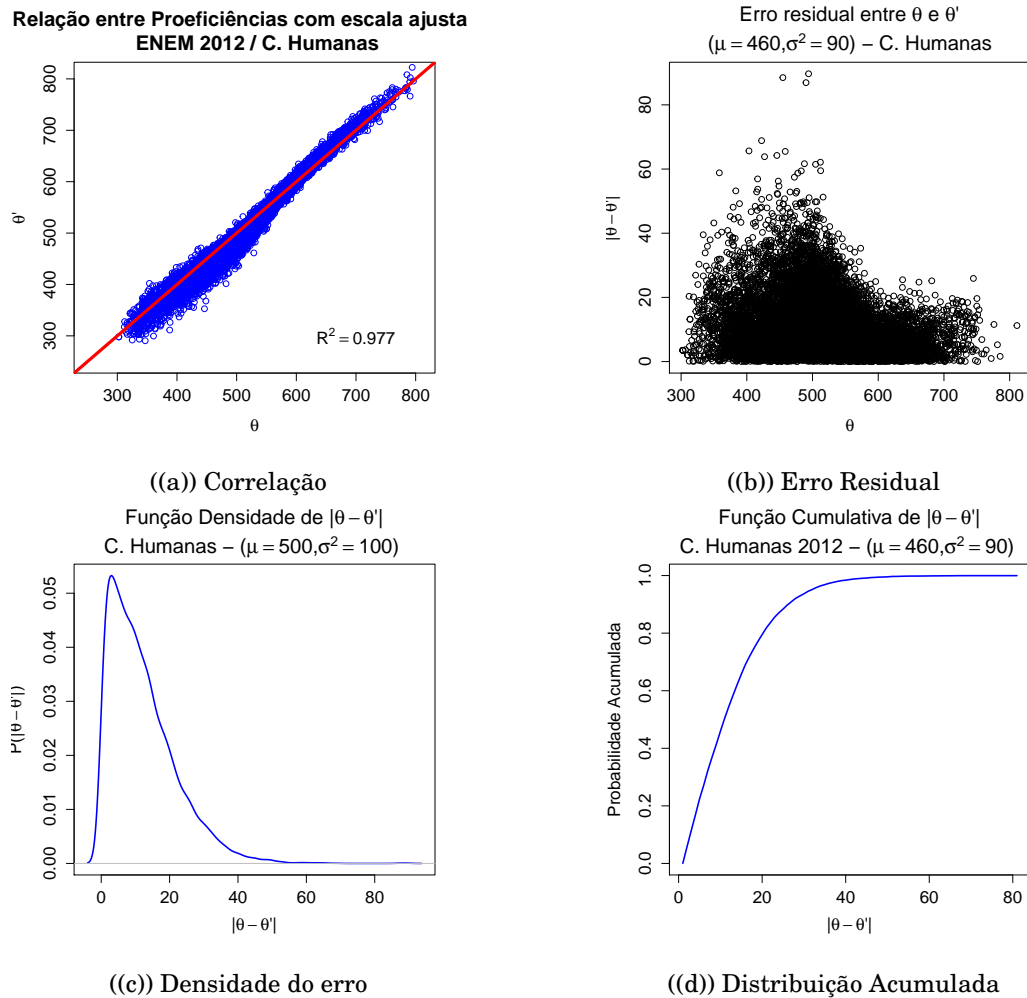
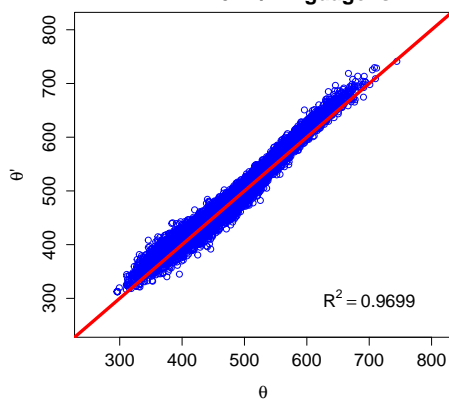


Figura 2.10: Resultados para a prova de Ciências Humanas

Prova	R^2	Mínimo	Média	Mediana	Máximo	σ^2
C.Humanas	0.977	0	12.79	9.87	93.6	14.90
C.Naturais	0.9562	0	14.12	11.74	95.57	18.31
Linguagens	0.9656	0	10.94	9.19	62.55	12.49

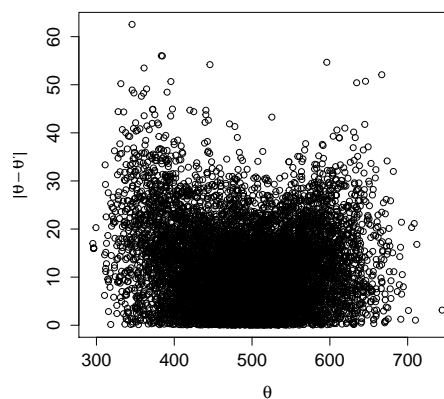
Tabela 2.4: Comparação entre as distribuições dos erros residuais com os diferentes ajustes

Relação entre Proeficiências com escala ajusta
ENEM 2012 / Linguagens



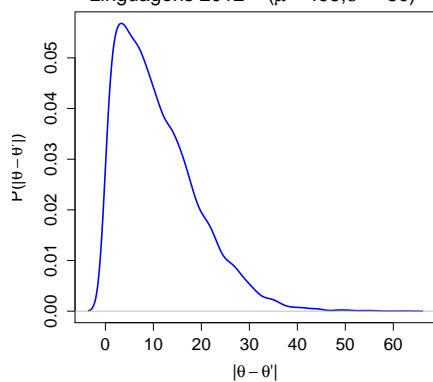
((a)) Correlação

Erro residual entre θ e θ' ($\mu = 495, \sigma^2 = 80$) – Linguagens



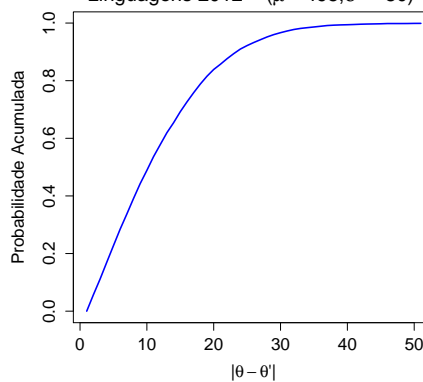
((b)) Erro Residual

Função Densidade de $|\theta - \theta'|$
Linguagens 2012 – ($\mu = 495, \sigma^2 = 80$)



((c)) Densidade do erro

Função Cumulativa de $|\theta - \theta'|$
Linguagens 2012 – ($\mu = 495, \sigma^2 = 80$)



((d)) Distribuição Acumulada

Figura 2.11: Resultados para a prova de Linguagens

2.4 Comentários

Apesar de ter alcançado um bom resultado, ainda existem fontes de erro difíceis de serem corrigidas. Por exemplo, o ajuste da escala é feito com relação à prova de 2009. Para isso, são selecionadas questões que são consideradas âncoras para a comparação entre dois anos diferentes. Por exemplo, uma equipe pedagógica escolhe duas questões, uma de cada prova, que são consideradas iguais. Seguindo o mesmo procedimento do cálculo do TRI para Linguagens, os parâmetros dessa questão na prova que se deseja analisar são fixados, gerando uma equivalência entre as duas escalas. Como não temos conhecimento sobre quais são as questões âncora ou como elas são escolhidas, é impossível ajustar as escalas perfeitamente.

QUERO MINHA NOTA

Após validar o algoritmo de cálculo do TRI no Capítulo 2, o foco mudará agora para o estudo do Quero Minha Nota, um sistema em desenvolvimento criado com o intuito de permitir que aplicantes do ENEM tenham uma estimativa de sua nota antes do lançamento oficial.

No sistema, após sair da prova o usuário fornece, através de uma interface gráfica, as alternativas assinaladas por ele em cada uma das questões das 4 provas objetivas. Após fazer cadastro e inserir as informações necessárias, ele é informado quais as questões ele acertou ou errou, a partir de um gabarito formulado por um parceiro do projeto. Junto com o diagnóstico da prova do aplicante, o sistema retorna também as proficiências ENEM estimadas.

A maior dificuldade dessa proposta reside em lidar com o princípio do TRI de uma análise estatística da população como um todo. Tentar dar ao aplicante uma nota baseada apenas em seu próprio padrão de resposta é impossível. Por isso, o Quero Minha Nota é uma aplicação que depende fortemente da adesão de uma grande quantidade de usuários.

Esse requisito traz consigo uma série de desafios a serem enfrentados. O primeiro a ser pensado é como permitir o acesso à maior quantidade possível de usuários, oferecendo a eles uma boa experiência de uso independente de como acessam o sistema. O segundo é como atender a todos esses usuários de forma eficaz, com respostas rápidas às suas interações. Por último, o maior desafio é como estimar as notas de todos esses usuários de forma eficiente e com precisão aceitável.

3.1 Análise de Requisitos

Tendo conhecimento da proposta do Quero Minha Nota, o primeiro passo para começar a sua implementação é definir os seus requisitos.

3.1.1 Requisitos Funcionais

- RF-1 Ao acessar o sistema pela primeira vez, o usuário não precisar imediatamente criar uma conta. Quando não autenticado, o usuário ainda tem acesso à função de editar seu gabarito.
- RF-2 Para que o usuário possa ter sua nota calculada, acesso ao gabarito oficial e sincronização de dados entre plataformas, é obrigatório que ele tenha inserido suas informações através do cadastro.
- RF-3 Um usuário deve poder inserir as alternativas escolhidas em cada uma das 4 provas objetivas do ENEM: Matemática, Linguagens, Ciências Humanas e Ciências Naturais.
- RF-4 Alterações feitas por um usuário tanto nos dados de sua conta quando no seu gabarito devem ser persistidas.
- RF-5 Após a criação da conta e a inserção dos gabaritos, o sistema deve ser capaz de estimar a nota do usuário para as quatro provas separadamente.
- RF-6 O usuário deve ser capaz de escolher entre as duas provas de Línguas Estrangeiras, e o sistema calculará a nota da prova de Linguagens considerando essa escolha.
- RF-7 Funcionamento offline: O usuário pode inserir informações mesmo não estando conectado à internet. Porém, as ações tomadas a partir do servidor por essas novas informações só serão realizadas após a conexão ter sido reestabelecida e os dados enviados.

3.1.2 Requisitos Não Funcionais

- RNF-1 Operabilidade: O sistema possui apenas um tipo de usuário. Esse usuário tem acesso total aos dados inseridos por referentes à sua conta.
- RNF-2 Autenticação: Para acessar os dados de uma conta, o usuário deve ter sua identidade verificada.
- RNF-3 Segurança: O sistema deve possuir mecanismos de proteção contra possíveis ataques mal-intencionados, como tentativa de acesso a informações de forma indevida ou de tentativas de

incapacitar os servidores com o uso de DDoS.

- RNF-4 Precisão: Deve ser fornecido também ao usuário além das estimativas de notas as faixas de erro da estimativa.
- RNF-5 Verificabilidade: Deve-se, através do número da inscrição no Enem do aluno, poder verificar a precisão obtida no sistema, com a saída dos dados oficiais referentes ao ano de 2015.
- RNF-6 Disponibilidade: O sistema deve no mínimo estar disponível para as plataforma Android e iOS. Idealmente, ele estará disponível também para Windows Phone e Web.
- RNF-7 Consistência: Alterações feitas por um usuário em um canal específico devem ser propagadas para todos os canais de acesso que ele usa.
- RNF-8 Custos: O sistema deve ser planejado a minimizar os custos de operação, mas sem desprezitar os outros requisitos funcionais e não funcionais.
- RNF-9 Escalabilidade: O sistema deve ter a capacidade de se adaptar a mudanças grandes no número de acessos, sem perder consistência, robustez e velocidade de resposta.
- RNF-10 Robustez: o sistema deve ter mecanismos de prevenção a falhas. Como falhas, podemos citar queda de conexão de internet por parte do usuário e queda de máquinas de servidor.
 - RNF-10.1 Queda de conexão de servidores: Caso o usuário esteja com problemas de conexão à rede, as alterações que ele fez devem ser salvas localmente no canal que o usuário está utilizando, para que os dados sejam atualizados quando a conexão for reestabelecida.
 - RNF-11.2 Indisponibilidade do Banco de Dados: o sistema deve implementar soluções para que, quando o usuário estiver conectado à internet, o banco de dados e a aplicação possam sempre atender às suas interações.
- RNF-12 Usabilidade: O sistema deve ser projetado para facilitar ao máximo o uso pelo usuário.
- RNF-13 Velocidade de resposta: A resposta à uma chamada de calcular nota não deve demorar mais que 5s para ser respondida.
- RNF-14 Para que de uma prova possa ser calculada, o usuário deve ter informado a situação de conclusão no Ensino Médio em 2015, a escola em que estuda e as suas respostas na prova

cuja nota será calculada. Para o caso da prova de Linguagens, é necessário também a Língua Estrangeira.

- RNF-15 Privacidade: Os dados fornecidos pelo usuário não deverão ser divulgados de forma pública ou privada para terceiros de forma que o usuário possa ser identificado correlacionado com esses dados. Toda a política de como esses dados serão ou não usados estarão presentes nos Termos de Uso do sistema.

3.2 Topologia

A arquitetura do sistema pode ser dividida em 4 camadas distintas, como mostrado na Figura 3.1. O objetivo dessa separação é permitir a divisão de responsabilidades, facilitando a escalabilidade para alto tráfego de informações. A comunicação entre as camadas é feita inteiramente através de chamadas a API's REST, e todos os dados enviados e recebidos entre elas encontram-se no formato JSON.

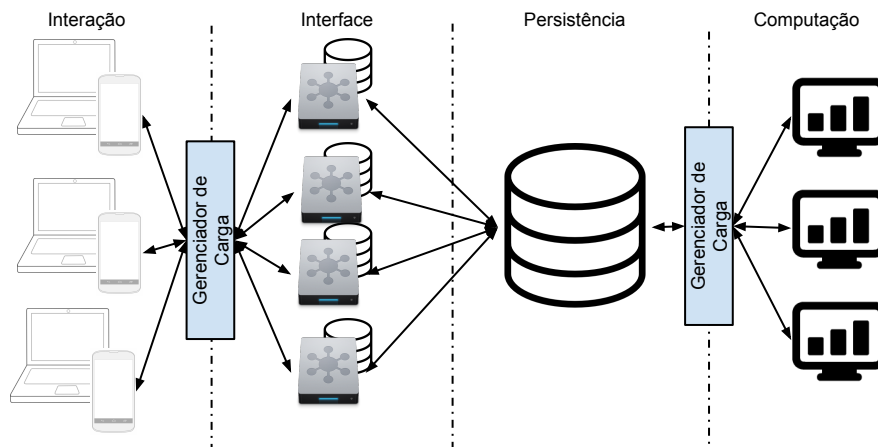


Figura 3.1: Representação da arquitetura funcional do Quero Minha Nota, com sua divisão em camadas

- Camada de Interação: Via de entrada para o usuário utilizar o sistema;
- Camada de Interface: Gerencia a comunicação entre as camadas de interação e persistência;
- Camada de Persistência: Armazena os dados da aplicação;
- Camada de Computação: Cálculo da análise TRI.

Enquanto a Figura 3.1 traz uma ilustração da topologia em um nível mais abstrato, a Figura 3.2 mostra as tecnologias empregadas em cada uma das camadas. Os motivos por trás da escolha de cada uma dessas tecnologias serão apresentados ao longo da descrição detalhada da operação de cada uma das camadas.

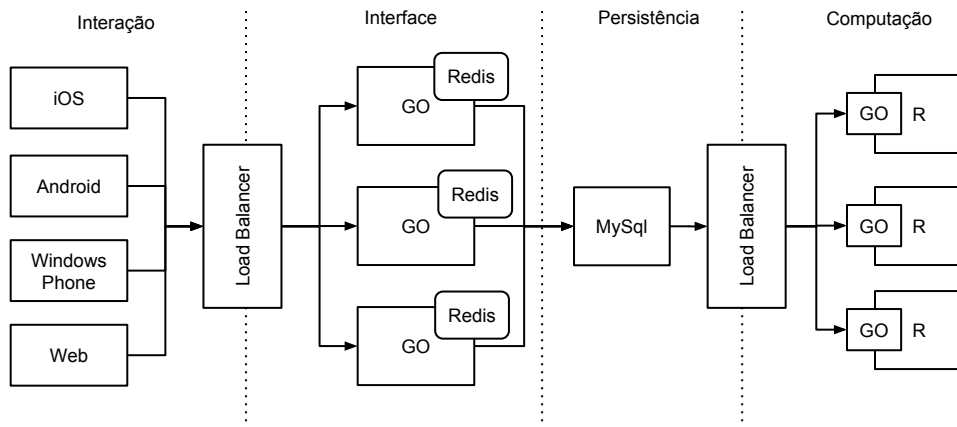


Figura 3.2: Representação da arquitetura do Quero Minha Nota, especificando quais tecnologias são utilizadas em cada camada.

Linguagem de Desenvolvimento

Foram cogitadas diversas linguagens para o desenvolvimento dos servidores. Grande parte dos produtos da empresa são desenvolvidos usando Ruby on Rails, framework muito popular para desenvolvimento web, por Ruby ser uma linguagem de alto nível e por possuir muitas bibliotecas prontas, sendo muito rápido colocar um sistema em produção. Isso vem com o custo de ser uma framework pesada, onde cada requisição do usuário requer uma quantidade razoável de memória e processamento para ser atendida.

Pela natureza da aplicação, espera-se receber uma quantidade enorme de requisições em um curto espaço de tempo. Se Ruby on Rails fosse escolhida nesse cenário dificilmente o servidor aguentaria. Python sofreria o mesmo problema.

Uma alternativa recente que vem ganhando bastante adesão por parte da comunidade é a linguagem denominada Go [5]. Criada por desenvolvedores da Google e lançada em 2012, é uma linguagem compilada de baixo nível, sendo extremamente veloz. Além disso, possui excelente suporte para desenvolvimento de servidores HTTP e, mais importante, foi desenvolvida pensando em concorrência. Por isso, essa linguagem foi a linguagem escolhida para as três camadas de back-end do Quero Minha Nota.

Estratégia de Escalabilidade

Devido à grande quantidade de aplicantes no Enem, espera-se que o sistema ganhe grande adesão rapidamente, implicando em um tráfego de dados enormes repentinamente. Para dar conta de tanto movimento, é preciso projetar as camadas de forma escalável, seguindo uma das duas estratégias possíveis: escalabilidade **vertical** e **horizontal**.

Escalar verticalmente (*"scaling up"*) um sistema significa aumentar a capacidade de processamento da máquina responsável pela computação de forma a minimizar o gargalo sofrido. Por exemplo, se a causa do atraso de respostas for a quantidade de acessos, pode-se aumentar a memória RAM. Agora se o problema forem cálculos muito pesados, pode-se melhorar o processador ou adicionar uma GPU.

Escalar Horizontalmente (*"scaling out"*) significa adicionar máquinas iguais que possuem as mesmas responsabilidades. Por exemplo, em vez de aumentar a memória da máquina responsável pelo atendimento a requisições, pode simplesmente adicionar uma máquina a mais e ter um algoritmo que redirecione uma requisição nova sempre para a máquina menos carregada.

A Figura 3.3 ilustra a diferença nas estratégias de escalabilidade existentes.

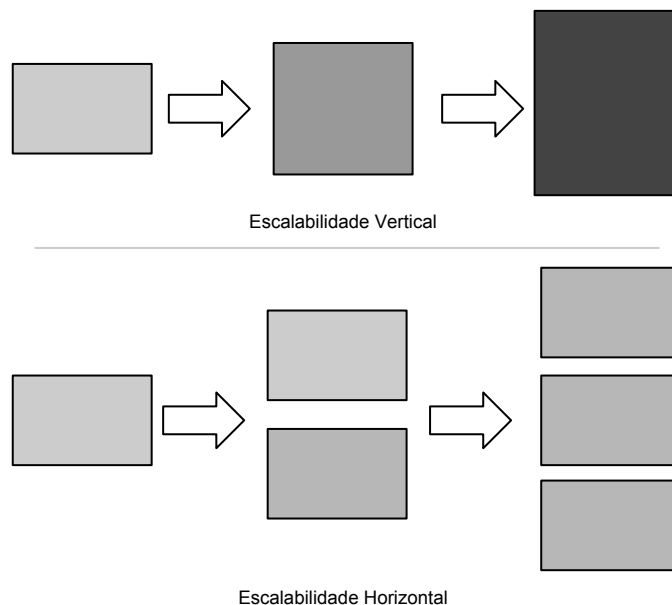


Figura 3.3: Exemplo da diferença de estratégias de escalabilidade

Autorização

Com relação à privacidade e segurança do usuário, cada conta é associada a um email e a uma senha. Porém pedir para que o usuário que forneça suas credencias toda vez que ele abrir o

website ou o aplicativo traz uma péssima experiência.

Como as camadas conversam exclusivamente através de APIs e trocam informações no formato JSON, a solução utilizada para resolver isso é autenticação via JWT [8]. Após o usuário fornecer sua senha e e-mail e os mesmos serem validados pela camada de persistência, é gerado um token criptografado contendo id do usuário. A criptografia é feita usando o algoritmo HS256, com base em uma string conhecida apenas pelo servidor.

Uma grande vantagem dessa estratégia é que o Token pode ser descriptografado e verificado pela camada de interface, evitando ter de fazer um acesso à camada de persistências para toda requisição privada que acontecer. Além disso, esse formato de autenticação é simples de implementar em todas as camadas.

3.3 Camada de Interação

De modo simples, a camada de interação é a parte do sistema acessível ao usuário. Compreende os meios pelos quais ele se cadastra no sistema, insere suas informações e tem acesso aos dados estimados com base em sua prova.

3.3.1 Funcionalidades da camada de Interação

A camada de interação deve permitir que o usuário realize as seguintes operações:

- Criar uma conta única e privada para ele

O usuário deve também ser informado de possíveis erros na criação da conta, como possíveis duplicatas.

- Acessar sua conta criada anteriormente

Após criar uma conta, o usuário pode acessá-la através de qualquer canal, e os dados mostrados a ele serão consistentes . Por exemplo, se o aplicante insere seu gabarito através do navegador em seu computador, esse gabarito deve estar disponível a ele também em um aplicativo de smartphone logado com sua conta, sem haver a necessidade de reinsertão.

- Inserir as alternativas marcadas por ele nas quatro provas objetivas

O usuário deve poder inserir as alternativas marcadas com base na cor de sua prova e língua estrangeira, e posteriormente ter a possibilidade de alterar esses dados.

- Visualizar de forma clara seus erros e acertos
- Visualizar informações sobre as questões das provas

Informações sobre as questões incluem estatísticas como enunciado, porcentagem de acerto e dificuldade, por exemplo.

- Visualizar suas notas estimadas
- Visualizar estatísticas gerais referente a prova

Dá-se o nome de estatísticas gerais para as estimativas que não são referentes exclusivamente ao usuário, como por exemplo média da escola e distribuição geral das notas na prova.

Canais de Distribuição

Como explicado anteriormente, para que o Quero Minha Nota possa calcular as notas com maior precisão, é necessário ter uma grande quantidade de usuários utilizando o sistema. Por isso, é necessário permitir que usuários possam acessar o sistema pela maior quantidade de canais possível. Com isso em mente, a interface do sistema é desenvolvida de duas formas: webapp e aplicativo para smartphones.

Aplicativo

Desenvolver uma versão de um sistema traz sempre uma pergunta importante: para quais plataformas desenvolver. No caso do Quero Minha Nota, quantos mais aplicantes puderem usar o sistema, melhor serão as estimativas das notas, o que levaria a desenvolver para o maior número de plataformas possível. Porém, desenvolver para multiplataformas normalmente requer trabalho duplicado: um app feito para Android não pode ser portado para iOS e vice-versa.

Uma solução viável é desenvolver em uma tecnologia híbrida. Apesar de normalmente terem uma performance muito pior do que aplicativos nativos, aplicativos desenvolvidos com tecnologias híbridas permitem disponibilizar o mesmo sistema para múltiplas plataformas, reduzindo muito o tempo de desenvolvimento. A framework mais comum que funciona dessa forma é a Apache Cordova [1] (antigo PhoneGap), que permite o desenvolvimento de apps híbridos usando Javascript, HTML e CSS, mas existem outras soluções open source como Qt [11] (C++), Kivy [9] (Python), RubyMotion [13] (Ruby) e outras comerciais, como Xamarim [14] (C#).

Entre todas as opções, foi escolhida a Ionic.js, framework construída em cima de Cordova e

AngularJS, essa última sendo a framework usada no desenvolvimento da plataforma web. Dessa forma, grande parte do código pode ser totalmente reutilizada.

A versão mobile será disponibilizada para Android, iOS e Windows Phone, o que representa 99% do mercado nacional.

As Figuras 3.4, 3.5 e 3.6 mostram o protótipo funcional da interface gráfica para o sistema mobile.

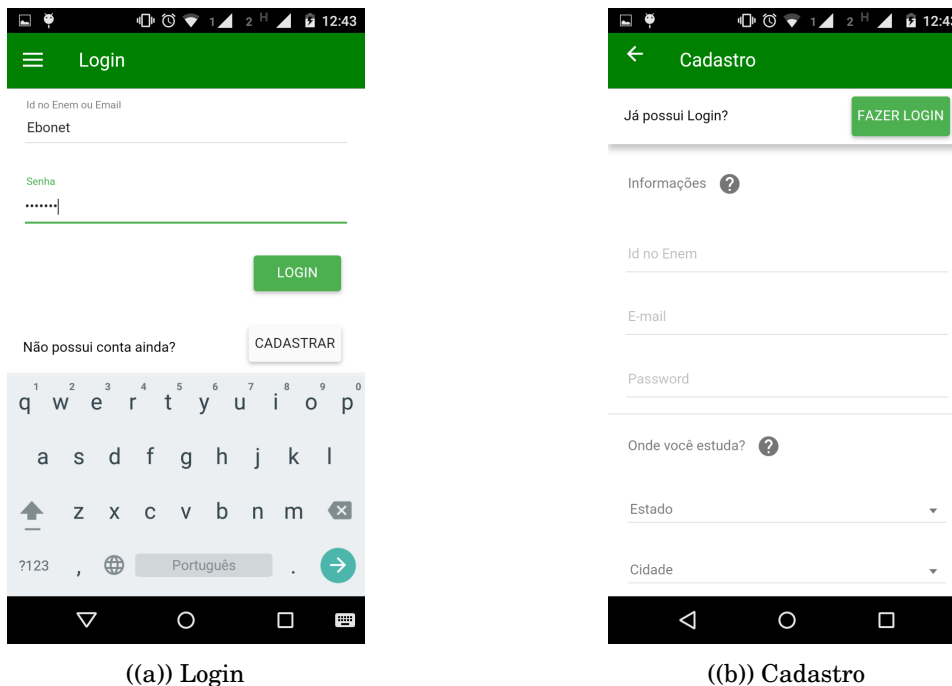


Figura 3.4: Protótipo da interface para dispositivos móveis - Login e Cadastro

3.3.2 Web

Para dispositivos móveis, é necessário ainda implementar as interfaces de informações sobre questões. Porém, esse tipo de informação é considerado um bônus do funcionamento do sistema, e tem prioridade baixa.

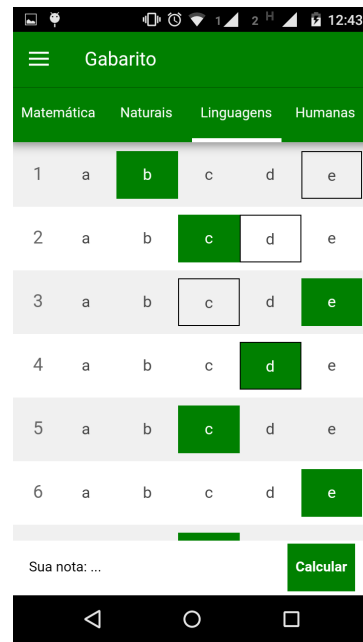
A versão Web está atualmente em desenvolvimento, mas permitirá o usuário usufruir de todas as funcionalidades da versão mobile, com usabilidade otimizada para navegadores. Ela está sendo construída usando AngularJS, de forma a reaproveitar ao máximo o código.

3.4 Camada de Interface

Situada entre a camada de Interação e a Camada de Persistência, a camada de interface tem como responsabilidade minimizar as requisições ao banco de dados, para evitar que o sistema

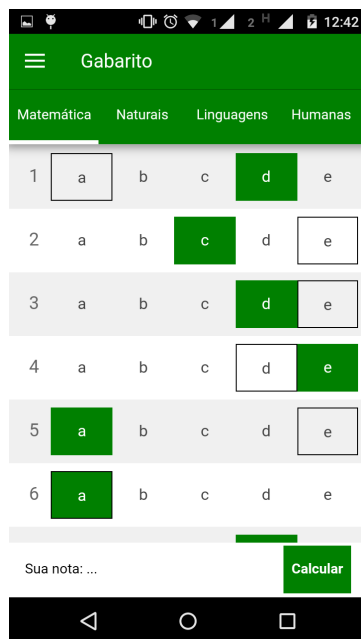


((a)) Menu

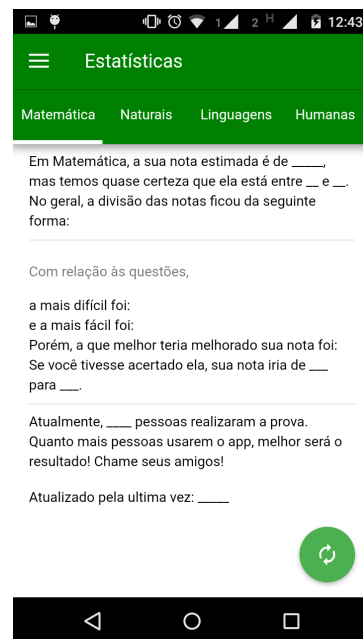


((b)) Questões assinaladas indicando quais são as corretas

Figura 3.5: Protótipo da interface para dispositivos móveis - Menu e Gabarito



((a)) Questões assinaladas



((b)) Estatísticas gerais

Figura 3.6: Protótipo da interface para dispositivos móveis - Gabarito e Estatísticas

sobrecarregue. Essa camada foi implementada pensando em velocidade de resposta, através do cacheamento de informações estáticas.

Camada de Interface: Escalabilidade

Essa é a camada que receberá a maior quantidade de requisições, concentrando um tráfego de dados enorme. Espera-se que essa camada atenda mais de 1000 requisições por minuto no período de uma semana.

A estratégia de escalabilidade horizontal é a mais indicada para essa camada, por diversos fatores. Primeiramente, essa camada não possui um estado a ser mantido. Ela simplesmente otimiza a comunicação entre a camada de interação e o banco de dados através de estratégias de *caching*, mas essas informações não são vitais para o funcionamento da camada. Adicionar uma máquina não traz nenhuma complexidade adicional para o funcionamento do sistema, já que os componentes da camada não precisam trocar informação um com o outro.

Em segundo lugar, adicionar uma máquina igual é muito mais barato que dobrar a capacidade de processamento da máquina, já que máquinas maiores geralmente requerem uma infraestrutura melhor preparada. Além disso, pode-se simplesmente ligar e desligar máquinas conforme ditar a demanda. Isso minimiza custos por provedores de computação em nuvem que cobram por hora de máquina ligada, e não por instância.

Por exemplo, pode-se ter 5 máquinas de interface preparadas, mas só duas ativas normalmente. Nos horários de pico de transações, as outras 3 máquinas são ligadas automaticamente para acelerar a resposta aos usuários, sem atrapalhar o funcionamento das outras duas. Porém, com o tempo essa demanda diminui, e as máquinas extras não são mais necessárias. Essa flexibilidade é muito mais difícil de alcançar com arquiteturas de escalabilidade vertical.

Para gerenciar o direcionamento das requisições para as diferentes máquinas ligadas, utiliza-se o serviço Load Balancer [3] fornecido pela Amazon Web Services.

Camada de Interface: Estratégia de cacheamento

Como mencionado, a função principal dessa camada é o cacheamento de requisições para o banco de dados. Quando um usuário requisita algum dado através da camada de interação, a camada de interface inicialmente procura os dados em seu cache, possibilitando economizar chamadas ao bando de dados. Apesar de existirem informações extremamente dinâmicas que não podem ser cacheadas, existem outras, como os gabaritos das provas e enunciados das questões, por exemplo, que nunca mudam. Algumas dessas informações são inclusive extremamente pesadas e requerem alta capacidade computacional para o cálculo ou envio.

Para fazer esse cache, emprega-se o REDIS [12], um banco de dados NoSQL no formato chave/valor

otimizado para rápida leitura. Para atingir seu objetivo, o REDIS opera totalmente em memória, fazendo com que as operações de leitura e escrita sejam extremamente velozes. A desvantagem é que ele deixa de ser um banco de dados com persistência, pois caso a máquina seja desligada todos os dados são perdidos e o cache é zerado.

Como pretende-se ligar e desligar máquinas com frequência para evitar custos, uma instância recém ativada precisaria inicialmente fazer muitos acessos para o banco de dados, por não possuir nenhuma informação em seu cache. Para resolver o problema, no processo de desligamento da máquina um processo copia automaticamente todos os dados do cache para o disco. Durante o processo de ligar a máquina, esses dados são transferidos de volta para o REDIS. Apesar de inserir um overhead na inicialização devido a esse passo adicional, o ganho em chamadas para o banco de dados evita a compensação.

Embora o sistema tenha muitas informações dinâmicas, o Redis possui também um mecanismo para expirar entradas. Por exemplo, pode-se cachear notas de padrões resposta e definir o tempo de invalidação para 2 horas. Dessa forma, tem-se um cache com garantia de que as informações não sejam desatualizadas.

3.5 Camada de Persistência

A camada de persistência tem duas responsabilidades: armazenar os dados fornecidos pelos usuários e coordenar a atualização das notas. O banco de dados utilizado foi o MySQL, por ser uma tecnologia usada em outros produtos da empresa.

A estratégia de escalabilidade adotada para essa camada é a escalabilidade vertical. Algo que dificulta a estratégia horizontal para esse caso é a necessidade dos dados serem replicados para todas as máquinas, o que envolve diversos problemas como consistência entre as instâncias. Uma possível alternativa para isso seria o chamado *sharding*, onde cada máquina armazena uma parte específica dos dados. Mas isso aumenta muito o tempo de requisições do tipo JOIN, por exemplo, além de não permitir que máquinas sejam ligadas e desligadas automaticamente.

3.5.1 Modelo do Banco de Dados

A Figura 3.7 mostra o diagrama relacional do banco de dados usado.

- **Aplicante:** Representa um usuário do sistema que fez a prova do ENEM.
- **Escola**

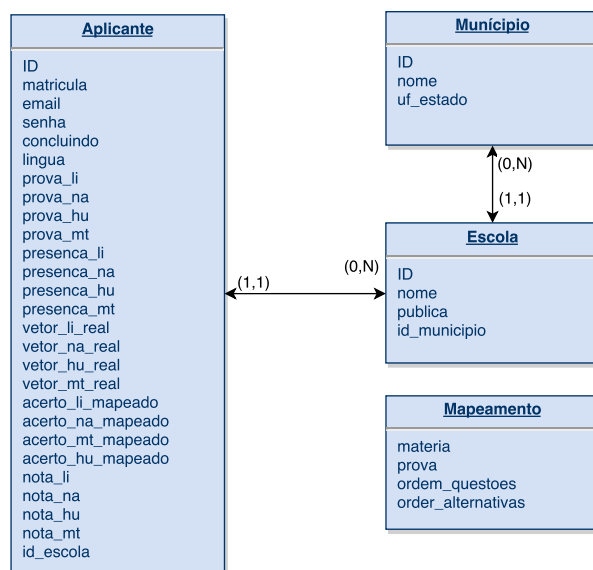


Figura 3.7: Diagrama Relacional do Banco de Dados

- Cidade
- Mapeamento

A tabela de mapeamento guarda as transformações necessárias para transformar os vetores de alternativas marcadas de cada tipo de prova diferente de uma matéria em um vetor resposta unificado. Por exemplo, caso seja escolhida a prova azul como padrão, por exemplo, essa tabela guarda a posição das questões e alternativas de uma prova qualquer na prova azul.

3.5.2 Gerenciamento de Atualização de Notas e Parâmetros

O processo de estimar os parâmetros das questões e as notas usando TRI com uma amostra de 100.000 alunos pode levar até 5 minutos para finalizar, além de usar muitos recursos do computador. Mesmo o cálculo de uma nota com parâmetros de questão já encontrados possui um overhead grande, especialmente devido ao fato de que o Script R que roda os procedimentos não pode ser compilado com funções extras, fazendo necessário recarregar as bibliotecas utilizadas a cada chamada para cálculo.

Isso torna inviável atualizar as notas cada vez que um participante altera seus padrões de resposta. A solução mais recomendada seria utilizar uma ferramenta melhor, mas trocar o algoritmo implicaria em tempo de desenvolvimento, o que iria extrapolar os prazos de entrega especificados. Além disso, é necessário uma quantidade significativa de dados novos para causar

Coluna	Tipo	Descrição
ID	int	Identificador único para o aluno
matricula_enem	int	Número de inscrição do aluno no ENEM
email	texto	Email do usuário
senha	texto	Senha de acesso do usuário
concluindo	binário	Status de conclusão do EM em 2015
lingua	binário	Língua estrangeira escolhida
presenca_li	binário	Presente na prova de Linguagens
presenca_na	binário	Presente na prova de C. Naturais
presenca_hu	binário	Presente na prova de C. Humanas
presenca_mt	binário	Presente na prova de Matemática
prova_li	int	Cor da prova de Linguagens
prova_na	int	Cor da prova de C. Naturais
prova_hu	int	Cor da prova de C. Humanas
prova_mt	int	Cor da prova de Matemática
vetor_li_real	texto(45)	Alternativas assinaladas em Linguagens
vetor_na_real	texto(45)	Alternativas assinaladas em C. Naturais
vetor_hu_real	texto(45)	Alternativas assinaladas em C. Humanas
vetor_mt_real	texto(45)	Alternativas assinaladas em Matemática
acerto_li_mapeado	binario(45)	Vetor resposta ajustado Linguagens
acerto_na_mapeado	binario(45)	Vetor resposta ajustado C. Naturais
acerto_mt_mapeado	binario(45)	Vetor resposta ajustado C. Humanas
acerto_hu_mapeado	binario(45)	Vetor resposta ajustado C. Matemática
nota_li	float	Nota em Linguagens
nota_na	float	Nota em C. Naturais
nota_hu	float	Nota em C. Humanas
nota_mt	float	Nota em Matemática
id_escola	int	ID Inep da escola do aplicante

Tabela 3.1: Colunas da tabela Aplicante

Coluna	Tipo	Descrição
ID	int	Identificador único da escola segundo o INEP
nome	texto	nome da escola
publica	binário	Escola pública ou privada
id_municipio	texto	ID da cidade onde fica a escola

Tabela 3.2: Colunas da tabela Escola

Coluna	Tipo	Descrição
ID	int	Identificador único da cidade
nome	texto	Nome da cidade
uf_estado	texto(2)	Sigla do estado

Tabela 3.3: Colunas da tabela Cidade

Coluna	Tipo	Descrição
materia	int	LIN (0), NAT (1), CHU(2), MAT(3)
prova	int	Cor da prova
ordem_questoes	texto	Posição de cada questão, separado por virgulas
ordem_alternativas	texto	posição de cada alternativa em cada questão após a transformações das posições, separado por virgulas

Tabela 3.4: Colunas da tabela Mapeamento

uma alteração significativa nos resultados. Dessa forma, é necessário um mecanismo de redução de requisições à camada de computação.

Esse mecanismo atua de duas formas: minimizar a atualização dos parâmetros e minimizar o número de processos gerados para calcular notas a partir de parâmetros já definidos.

- Reduzindo o número de atualizações de parâmetros

Para reduzir o número de atualizações de parâmetros de nota, especifica-se um intervalo fixo de 2 horas entre cada ciclo. Atualizações muito constantes, além de causar o sobrecarregamento das máquinas, não tem informações de usuários adicionais agregadas suficientes para alterar significativamente o resultado da otimização.

Durante o período entre atualizações, os novos padrões de resposta que forem adicionados ao banco de dados tem sua nota calculada com base nos parâmetros encontrados no último ciclo.

- Reduzindo o número de atualizações de notas

Como o grande objetivo do sistema é permitir que os alunos tenham uma estimativa de sua nota no ENEM, serão feitas muitas requisições de cálculo de nota. A execução do cálculo de proficiências de 10 padrões de resposta com parâmetros pré-estimados, leva em torno de 4 segundos, dos quais pelo menos dois são gastos carregando as bibliotecas, um overhead muito grande.

O primeiro passo para reduzir o número de chamadas para a camada de computação é cachear notas para padrões de respostas previamente calculados. A atualização dos parâmetros retorna também as notas dos padrões de resposta usados para otimização. Guardar esses resultados no banco de dados economiza uma quantidade significativa de acessos.

Porém, existem 2^{45} combinações de padrões de resposta possíveis. No ENEM de 2012, o padrão que mais se repetiu na prova de matemática foi o com todas as questões erradas, com um total de aproximadamente 0.03% dos participantes apenas. Portanto, apesar de o cache realmente evitar diversos acessos, a probabilidade de que haja requisição para um padrão de resposta que nunca foi visto pelo sistema ainda é alta.

A diferença de tempo entre calcular 10 ou 100 notas é desprezível. Por isso, a próxima estratégia adotada é agrupar os pedidos de cálculo de nota. Da mesma forma que no caso da atualização de parâmetros, estipulamos um tempo fixo entre requisições para a camada de computação. Em contrapartida, existe um usuário esperando pela resposta do servidor, logo esse intervalo deve ser pequeno, para que a requisição não demore muito pra ser respondida.

Por isso, os processos de cálculo de notas são iniciados a cada 0.5 segundos. As requisições feitas durante esse período são acumuladas e enviadas em conjunto.

3.6 Camada de Computação

Essa camada é responsável por todos os cálculos referentes à Teoria de Resposta ao Item realizadas. Ela possui as mesmas características gerais da camada de interface: máquinas não muito potentes, que não guardam estado, mas onde pode-se ter várias delas funcionando ao mesmo tempo, com chamadas gerenciadas por um Load Balancer.

O código em R é chamado a partir do servidor através de linha e comando. Existem dois procedimentos principais: um para calcular os parâmetros a partir de padrões de resposta e outro para estimar as notas a partir de um conjunto de padrões resposta e os parâmetros das questões pré-estabelecidos. Esses scripts possuem o mesmo princípio de funcionamento do procedimento do cálculo do TRI discutido no Capítulo 2.

Assim como a camada de interface, essa camada também não guarda estado. Por isso, máquinas podem ser ligadas e desligadas conforme a demanda. Por isso, essa camada também possui escalabilidade horizontal.

CONCLUSÃO

No Capítulo 1, foi feita apresentação detalhada sobre o que é a Teoria de Resposta ao Item, e como usá-la pode trazer diversas vantagens para a consistência das notas do Enem ao longo dos anos, especialmente considerando sua nova proposta de ser uma nova alternativa ao vestibular.

Na sequência, foi mostrado no Capítulo 2 que é possível estimar notas do ENEM a partir dos padrões de resposta dos participantes com alta correlação linear. A maior fonte de erro nesse caso é ajustar as escalas, dado que esse processo depende fortemente em alinhar questões âncoras entre os diferentes exames.

Finalmente, o sistema Quero Minha Nota foi introduzido no Capítulo 3 como uma solução para o problema que aplicantes encontram de não entender realmente a nota calculada através do TRI. Para isso, foram alinhadas uma série de estratégias para garantir que usuários tenham uma boa experiência uso, garantindo que ele possa ser acessado tanto a partir de smartphones quanto a partir do navegador, mantendo a consistência e o tempo de resposta baixo, em adição a uma estimativa real de seu resultado.

Próximos Passos

Com relação à estimativa das proficiências TRI, é possível diminuir ainda mais o erro apresentado, com um tempo de execução menor. A biblioteca mirt teve um papel grande em um primeiro momento de estudo e prototipação, mas alguns fatores como o overhead causado pelo R tornam seu uso proibitivo a médio prazo.

Voltando para o Quero Minha Nota, ele claramente ainda não está finalizado. Seu lançamento está previsto para a edição 2015 do Enem, que acontecerá nos dias 24 e 25 de Outubro. Até lá, muito trabalho será empregado pela equipe para melhorar ainda mais a experiência do usuário.

BIBLIOGRAFIA

- [1] *Apache Cordova - apache cordova is a platform for building native mobile applications using HTML, CSS and JavaScript.*
<https://cordova.apache.org/>.
Acesso em: 2015-07-25.
- [2] *Download de Microdados - portal INEP.*
<http://portal.inep.gov.br/basica-levantamentos-acessar>.
Acesso em: 2015-07-25.
- [3] *Elastic load balancer.*
<http://aws.amazon.com/pt/elasticloadbalancing/>.
Acesso em: 2015-07-25.
- [4] *Enem - portal INEP.*
<http://portal.inep.gov.br/web/enem/enem>.
Acesso em: 2015-07-25.
- [5] *Go.*
<https://golang.org/>.
Acesso em: 2015-07-25.
- [6] *Guacamole.*
<https://github.com/Khan/guacamole>.
Acesso em: 2015-08-25.
- [7] *Guia Do Estudante ENEM - Entenda sua nota - portal INEP.*
http://download.inep.gov.br/educacao_basica/enem/guia_participante/2013/guia_do_participante_notas.pdf.
Acesso em: 2015-07-25.
- [8] *JWT - json web tokens.*
<http://jwt.io/>.
Acesso em: 2015-07-25.

BIBLIOGRAFIA

- [9] *Kivy: Cross-platform Python framework for NUI development.*
kivy.org/.
Acesso em: 2015-07-25.
- [10] *Nota técnica - teoria de resposta ao item (TRI).*
http://download.inep.gov.br/educacao_basica/enem/nota_tecnica/2011/nota_tecnica_tri_enem_18012012.pdf.
Acesso em: 2015-07-25.
- [11] *Qt.*
<http://www.qt.io/>.
Acesso em: 2015-07-25.
- [12] *Redis - an open source, bsd licensed, advanced key-value cache and store.*
<http://redis.io/>.
Acesso em: 2015-07-25.
- [13] *Rubymotion.*
<http://www.rubymotion.com/>.
Acesso em: 2015-07-25.
- [14] *Xamarin: Mobile app development app creation software.*
<http://xamarin.com/>.
Acesso em: 2015-07-25.
- [15] R. P. CHALMERS, *mirt: A multidimensional item response theory package for the R environment*, Journal of Statistical Software, 48 (2012), pp. 1–29.
- [16] S. M. DOWNING, *Item response theory: applications of modern test theory in medical education*, Medical Education, 37 (2003), pp. 739–745.
- [17] J. LEIGHTON AND M. GIERL, *Cognitive diagnostic assessment for education: Theory and applications*, Cambridge University Press, 2007.
- [18] F. M. LORD, M. R. NOVICK, AND A. BIRNBAUM, *Statistical theories of mental test scores.*, (1968).
- [19] I. PARTCHEV, *A visual guide to item response theory*, (2004).
Acesso em: 2015-07-25.
- [20] R CORE TEAM, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2015.
- [21] G. RASCH, *Probabilistic models for some intelligence and attainment tests.*, ERIC, 1993.

- [22] R. E. TRAUB, *Classical test theory in historical perspective*, Educational Measurement: Issues and Practice, 16 (1997), pp. 8–14.

