

Prognóstico de Daltonismo: como a tecnologia pode auxiliar na detecção do distúrbio através da automatização e acessibilidade

Ana Beatriz Soares, Alice Crispim Bruno, Isaac Henrique de Paula
Departamento de Engenharia Eletrônica
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil

Resumo—Este artigo descreve a implementação de um teste clínico utilizado para o diagnóstico dos casos mais comuns de discromatopsia. A aplicação permite um auxílio na compreensão do prognóstico do transtorno, encorajando que o paciente procure um especialista já direcionado para o diagnóstico final que resultará na confecção das lentes de correção.

Palavras-chave — daltonismo, python, método, paciente, automatização, otimização

I. INTRODUÇÃO

O nome *daltonismo* possui origem em um dos primeiros casos relatados do distúrbio da visão, John Dalton, famoso cientista que possui diversas contribuições para a comunidade acadêmica, química e física. Dalton observou que havia alguma anomalia em seu modo de ver o mundo e escreveu um artigo onde descreve como era viver confundindo as cores verde e vermelho.

Notou-se que o daltonismo afeta majoritariamente os homens devido à forma como os genes responsáveis pela percepção das cores são herdados. A principal causa do daltonismo é uma alteração nos genes que estão localizados no cromossomo X. Os homens têm um cromossomo X e um cromossomo Y (XY), enquanto as mulheres têm dois cromossomos X (XX).

Se um homem herda um cromossomo X com o gene alterado para daltonismo, ele manifestará a condição, porque não possui um segundo cromossomo X para compensar o defeito. Em contrapartida, uma mulher precisaria herdar o gene alterado de ambos os pais para ser daltônica, o que é muito menos comum. Se ela herdar apenas um cromossomo X com o gene alterado, o outro cromossomo X normal pode compensar a deficiência, e ela será apenas portadora do gene do daltonismo, sem manifestar a condição.

De acordo com informações do National Eye Institute: "Color blindness is more common in men because the genes responsible for the most common, inherited color

blindness are on the X chromosome. Males only have one X chromosome, while females have two X chromosomes. A single recessive gene on the X chromosome will cause color blindness in males, whereas it would take two such genes to cause color blindness in females" (National Eye Institute, 2021).

Existem vários tipos de daltonismo, que podem ser classificados principalmente em três categorias: deuteranopia, protanopia e tritanopia, além de subtipos relacionados a essas categorias principais. Sendo elas: Deuteranopia e Deuteranomalismo (Daltonismo verde), Protanopia e Protanomalismo (Daltonismo vermelho), Tritanopia e Tritanomalismo (Daltonismo azul-amarelo) e Acromatopsia (Daltonismo total).

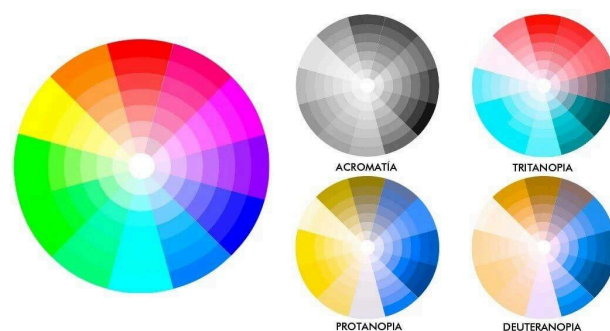


Fig. 1. Círculo cromático para a visão normal à esquerda, e para cada tipo de daltonismo à direita.[4].

Dessa maneira, este trabalho consiste no desenvolvimento de uma aplicação que forneça um prognóstico ao usuário para auxiliá-lo na compreensão da condição do transtorno visual o qual ele é portador. Entretanto, este prognóstico não garante que pessoa realmente possui o distúrbio, por este motivo, a longo prazo, seria importante fazer o desenvolvimento dessa aplicação em conjunto com profissionais da saúde e pesquisadores para que, além de um prognóstico, possamos fornecer uma forma rápida e segura de

diagnóstico que torne a comunicação médico-paciente mais clara e direta.

Os códigos utilizados estão disponíveis em um repositório do GitHub[3] que pode ser acessado por meio deste link: [Teste Daltonismo](#).

II. REFERENCIAL TEÓRICO

O National Eye Institute (NEI)[1] fornece uma descrição abrangente sobre o daltonismo, abordando seus diferentes tipos, causas, sintomas, diagnóstico e tratamento. O NEI explica que o daltonismo é frequentemente causado por fatores genéticos, com genes responsáveis localizados no cromossomo X, o que torna a condição mais comum em homens. Os tipos mais comuns incluem deuteranopia, protanopia, tritanopia e acromatopsia. Os sintomas incluem dificuldade em distinguir certas cores ou, em casos graves, a incapacidade de ver qualquer cor. Para o diagnóstico, são utilizados testes como o de cores de Ishihara. Não há cura para o daltonismo, mas o NEI menciona o uso de óculos ou lentes de contato coloridas para melhorar a percepção de cores e discute novas pesquisas para possíveis tratamentos futuros. Além disso, o NEI aborda o impacto do daltonismo na vida cotidiana e oferece estratégias de adaptação para as pessoas afetadas.

O Colour Blind Awareness (CBA)[2] sobre o daltonismo inclui uma explicação detalhada dos diferentes tipos de daltonismo, sua prevalência, causas, e o impacto na vida cotidiana das pessoas afetadas. A organização também se concentra em aumentar a conscientização e fornecer recursos educacionais.

O CBA detalha os tipos de daltonismo, como deuteranopia, protanopia e tritanopia, explicando as diferenças na percepção de cores para cada tipo. Eles destacam que o daltonismo é mais comum em homens devido à herança genética ligada ao cromossomo X. A prevalência é de aproximadamente 8% nos homens e 0,5% nas mulheres.

A organização também aborda o impacto do daltonismo em várias áreas da vida, como a educação e o trabalho. Eles fornecem orientações para professores e empregadores sobre como apoiar pessoas daltônicas, sugerindo adaptações simples, como o uso de materiais didáticos com alto contraste e a evitação de informações codificadas por cores.

Além disso, o CBA promove a conscientização pública sobre o daltonismo, enfatizando a importância de diagnósticos precoces e o desenvolvimento de produtos e serviços acessíveis. A organização também apoia pesquisas e colabora com outras entidades para melhorar a qualidade de vida das pessoas com daltonismo.

III. METODOLOGIA

O Teste de Ishihara é comumente realizado em um consultório oftalmológico, na presença de um médico oftalmologista, o qual apresenta para o paciente diversos cartões, 38 em seu total. Cada cartão contém um padrão de pontos/círculos com cores estratégicas que formam um número ou símbolo, e o paciente deve informar o que ele está enxergando em cada cartão. Em consultório também é analisada a intensidade da dificuldade do paciente em diferenciar os tons e enxergar os padrões esperados, a fim de compreender o quanto o distúrbio de cores está afetando o cotidiano da pessoa.

Por se tratar de uma análise extensa, onde são mostrados muitos cartões com diferentes possibilidades de resposta cada um, a depender do tipo de daltonismo do paciente, o método atual tem espaço para melhorias, modernização, automatização e acessibilidade.

Desse modo, a proposta deste projeto é implementar um teste de daltonismo, utilizando os cartões do Teste de Ishihara e uma interface gráfica simples que interage com o usuário coletando suas respostas perante a imagem de cada cartão, e fornece um prognóstico para determinado tipo de daltonismo, em formato PDF, com os dados do paciente e o seu resultado do teste, encorajando-o a se consultar com um profissional.

O projeto, apelidado intuitivamente como Teste de Daltonismo, foi implementado utilizando a linguagem de programação Python e sua interface gráfica padrão Tkinter[5] e bibliotecas adicionais que serão citadas ao longo dos tópicos. O Teste é dividido em duas janelas principais: a janela do Usuário e a janela de Ishihara. Ao todo foram escritas 4 classes, separadas em arquivos diferentes, que se relacionam entre si utilizando principalmente herança, um dos pilares da Programação Orientada a Objetos, e composição.

A. Classe Usuario (*usuario.py*)

A classe Usuario é a responsável pela janela que faz a primeira interação com o paciente, onde este insere seus dados: nome completo, data de nascimento, CPF, E-mail e telefone/celular.

Nesta classe foi aplicada a herança da classe TesteDaltonismo, da qual foi herdado o método abstrato *def iniciar_teste*. Este método é responsável pelo que o próprio nome diz, iniciar o teste. Dentro da classe Usuario, a implementação deste método, inicia a janela que recebe as informações pessoais do usuário antes do início do Teste de Daltonismo, utilizando o Tkinter para a criação gráfica da janela. A utilização desse método abstrato demonstra o pilar da Abstração na Programação Orientada a Objeto.

Os dados fornecidos pelo usuário, como nome completo, data de nascimento, telefone e celular, são

armazenados como atributos protegidos, adicionando um único underscore '_' antes do nome do atributo, indicando que eles não podem ser acessados diretamente fora da classe, mas podem ser acessados por subclasses. Já os dados mais sensíveis, como CPF e e-mail, são armazenados como atributos privados, adicionando dois underscores '__' antes do nome do atributo, restringindo ainda mais o acesso. Feito isso, o acesso direto aos dados do usuário fica restrito, podendo ser acessados apenas através de métodos específicos, conhecidos como getters e setters, permitindo a leitura e modificação dos dados de forma controlada e segura, garantindo a integridade dos dados e protegendo-os contra acessos indevidos. Esse uso de atributos protegidos e privados, juntamente com métodos públicos para tratar os dados do paciente, exemplifica o pilar do encapsulamento na Programação Orientada a Objetos.

Fig. 2. Janela Usuario em execução. Imagem do autor.

Algumas funcionalidades da classe Usuario são:

- **Configuração da Janela Principal:** Criação da janela principal com propriedades definidas (título, tamanho, cor de fundo, etc.) e exibe um cabeçalho para que o usuário possa preencher sua ficha de paciente.
- **Coleta de Informações Pessoais:** Campos de entrada para nome, sobrenome, telefone, email e CPF.
- **Validação das Entradas:** O método *def configurar_validacao* garante que apenas caracteres válidos sejam aceitos. Por exemplo, no espaço destinado ao nome não são aceitos caracteres numéricos, enquanto no campo destinado ao CPF não são aceitas letras. Além disso, o método *def check_entradas* monitora os campos obrigatórios e habilita o botão "Iniciar Teste" apenas quando todos os campos estiverem preenchidos, garantindo que todas as informações do paciente sejam coletadas.

- **Início do Teste:** O botão "Iniciar Teste" chama o método *def iniciar_bt*, que executa a função *subprocess.Popen* para iniciar o teste de daltonismo, acessando o arquivo *ishihara.py*. Os atributos coletados do usuário são passados como parâmetros para a classe Ishihara.

B. Classe Ishihara (*ishihara.py*)

Assim como na classe Usuario, a classe Ishihara também herda a classe TesteDaltonismo e seu método abstrato *def iniciar_teste* o implementando de forma diferente, trazendo assim o pilar do Polimorfismo ao código. A classe Ishihara apresenta as placas de Ishihara ao usuário, coleta suas respostas, gera o resultado do teste e habilita o botão para gerar um arquivo PDF com as informações do usuário e seu prognóstico com base em suas respostas.

A classe Ishihara inicia uma nova janela com propriedades definidas (título, tamanho, cor de fundo, etc.). Nela são exibidos o título com o número da placa de Ishihara atual, bem como a placa atualizada. Apresenta uma caixa de seleção onde o usuário pode com facilidade selecionar o número ou símbolo que ele está enxergando. Quando a caixa de seleção é preenchida, um botão para alternar para a próxima imagem é habilitado. O processo se repete para as 38 imagens.

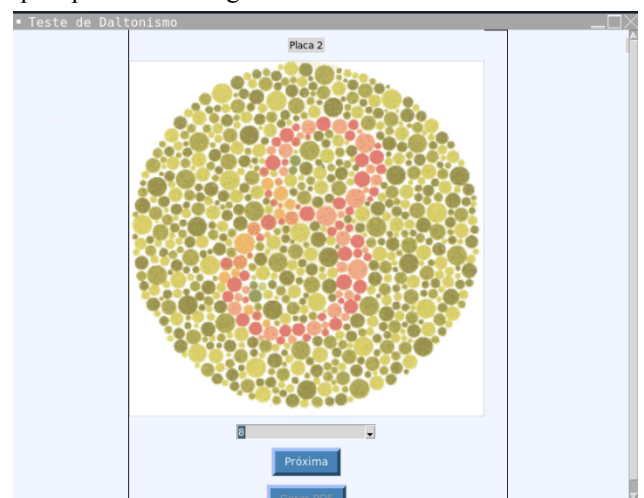


Fig. 3. Janela Ishihara, Placa 2, com a caixa de seleção já preenchida e o botão Próxima habilitado. Imagem do autor.

Algumas funcionalidades da classe Ishihara são:

- **Recebe os Atributos do Usuário:** na classe Usuario, os atributos foram passados como parâmetros para a classe Ishihara, que por sua vez utiliza o módulo *argparse* para 'capturar' esses parâmetros e conseguir utilizá-los dentro da classe Ishihara.
- **Apresentação das Placas de Ishihara:** Imagens das placas de Ishihara são carregadas e exibidas. O usuário insere o que vê em cada placa e as

respostas são analisadas por meio da comparação. Cada imagem tem sua configuração com as prováveis respostas para cada tipo de daltonismo, de forma que quando o usuário seleciona uma das alternativas ela é atribuída a uma variável *count* específica ao seu tipo. No final, a variável com maior valor indica o prognóstico.

- Navegação Entre as Placas: O botão "Próxima" coleta a resposta do usuário para a placa atual, armazena essa resposta e avança para a próxima placa. Quando todas as placas são exibidas, o botão "Finalizar" é habilitado.
- Armazenamento dos Resultados: No método *def mostrar_resultados*, as respostas são salvas em um arquivo de dicionário e impressas no lado direito da mesma janela. O resultado do teste é colocado em uma variável *diagnostico* que é passada para o método *def salvar_dados*. Nesse método é criado um banco de dados, utilizando a biblioteca *SQLite*, que adquire os dados do usuário capturados por Ishihara, e também o resultado do teste armazenado na variável *diagnostico*. Por fim, utilizando novamente a função *subprocess.Popen*, o arquivo que gera o PDF informativo (*documento.py*) é chamado.

C. Classe *TesteDaltonismo* (*InterfaceDaltonismo.py*)

A classe *TesteDaltonismo* é definida no arquivo *InterfaceDaltonismo.py*. Ela é composta por dois métodos: o método abstrato *def iniciar_teste* e o método *def habilitar_botao*.

Quando o método abstrato *def iniciar_teste* é herdado, é implementado por ambas as classes que herdam ele. Como tanto a classe *Usuario* quanto a classe *Ishihara* herdam da classe *TesteDaltonismo*, ambas implementam o método *def iniciar_teste*, mas de formas diferentes. Isso caracteriza o Polimorfismo aplicado a esse método.

O método *def habilitar_botao* é um método criado no intuito de habilitar e desabilitar botões. Também é herdado por ambas as classes, e é chamado sempre que necessário controlar a habilitação/desabilitação dos botões.

D. Classe *Documento* (*documento.py*)

A classe *Documento* é a classe que gera um arquivo PDF contendo as informações do usuário e o seu resultado no Teste de Daltonismo.

Utilizando o módulo *reportlab*, ele gera um arquivo PDF com o nome 'Resultado_{nome do paciente}.pdf' do último usuário adicionado ao banco de dados criado na classe *Ishihara*. Este PDF é salvo automaticamente na

máquina e também pode ser visualizado na Web. O módulo *webbrowser* é o responsável por isso, facilitando o acesso do usuário ao resultado.

E. Esquemático UML

Condensando o que foi exposto a respeito da estrutura do código, o seguinte esquemático do diagrama UML[Fig. 4.] apresenta as interações que ocorrem entre as classes.

As classes *Usuario* e *Ishihara* herdam a classe *TesteDaltonismo*, contida no arquivo *InterfaceDaltonismo.py*, como representado pelas setas em direção a *Interface*. *Usuario* e *Ishihara* tem uma relação de agregação, onde as duas classes estabelecem um vínculo quando *Usuario* passa os parâmetros do usuário para *Ishihara*. O mesmo acontece entre *Ishihara* e *Documento*, através do banco de dados criado em *Ishihara*, que foi utilizado em *Documento* para as informações do arquivo PDF.

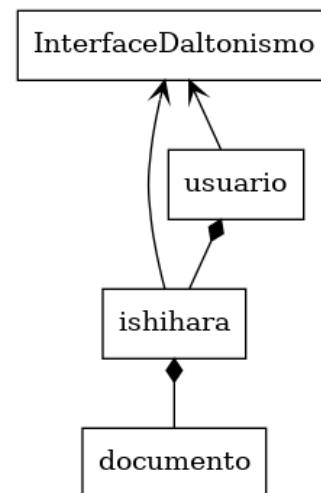


Fig. 4. Esquemático das ligações de herança e composição entre as classes. Imagem do autor.

IV. RESULTADOS

Foram obtidos resultados satisfatórios tanto para o prognóstico quanto para a geração do arquivo PDF.

A linguagem Python e sua interface padrão Tkinter se mostraram suficientes para a implementação do projeto Teste de Daltonismo, trazendo uma interação simplificada, intuitiva e amigável com o usuário, contribuindo com a acessibilidade do teste. Durante os testes realizados, foi possível avaliar o desempenho do sistema em termos de precisão na coleta e armazenamento das informações dos usuários, como nome completo, data de nascimento, CPF, e-mail e telefone. A aplicação do encapsulamento, com atributos privados e métodos de acesso, garantiu a segurança e a integridade desses dados, demonstrando a eficácia dessa prática na proteção contra acessos indevidos.

Durante a fase de avaliação do teste, visando garantir sua eficácia em auxiliar pessoas com daltonismo, foram realizadas cinco execuções distintas do teste. Cada execução teve como objetivo simular uma das possíveis condições de visão afetada pelo daltonismo, incluindo visão normal, protanopia, deuteranopia, tritanopia e também uma execução destinada a situações onde o resultado fosse inconclusivo. Essa abordagem múltipla permitiu uma análise abrangente das capacidades do teste em diferentes cenários de daltonismo, garantindo sua capacidade de identificar e distinguir entre os diferentes tipos dessa condição visual. Cada execução foi cuidadosamente planejada e executada, com os participantes fornecendo respostas de acordo com a condição visual simulada.

| |
|--|
| Placa 1: 12 (Normal) |
| Placa 2: 8 (Normal) |
| Placa 3: 6 (Normal) |
| Placa 4: 29 (Normal) |
| Placa 5: 57 (Normal) |
| Placa 6: 5 (Normal) |
| Placa 7: 3 (Normal) |
| Placa 8: 15 (Normal) |
| Placa 9: 74 (Normal) |
| Placa 10: 2 (Normal) |
| Placa 11: 6 (Normal) |
| Placa 12: 97 (Normal) |
| Placa 13: 45 (Normal) |
| Placa 14: 5 (Normal) |
| Placa 15: 7 (Normal) |
| Placa 16: 16 (Normal) |
| Placa 17: 73 (Normal) |
| Placa 18: Abstrato de cores (Normal) |
| Placa 19: Abstrato de cores (Normal) |
| Placa 20: Abstrato de cores (Normal) |
| Placa 21: Abstrato de cores (Normal) |
| Placa 22: 26 (Normal) |
| Placa 23: 42 (Normal) |
| Placa 24: 35 (Normal) |
| Placa 25: 96 (Normal) |
| Placa 26: Duas linhas de cores diferentes (Normal) |
| Placa 27: Duas linhas (Normal) |
| Placa 28: Abstrato de cores (Normal) |
| Placa 29: Abstrato de cores (Normal) |
| Placa 30: Uma linha (Normal) |
| Placa 31: Uma linha (Normal) |
| Placa 32: Uma linha (Normal) |
| Placa 33: Uma linha (Normal) |
| Placa 34: Uma linha (Normal) |
| Placa 35: Uma linha (Normal) |
| Placa 36: Uma linha (Normal) |
| Placa 37: Uma linha (Normal) |
| Prognóstico Final: Visão Normal |

Fig. 5. Teste com resultado de visão Normal. Imagem do autor.

| |
|---|
| Placa 1: 12 (Normal) |
| Placa 2: 3 (Protanopia) |
| Placa 3: Nada (Protanopia) |
| Placa 4: 70 (Protanopia) |
| Placa 5: 57 (Normal) |
| Placa 6: 2 (Protanopia) |
| Placa 7: 5 (Protanopia) |
| Placa 8: 17 (Protanopia) |
| Placa 9: 21 (Protanopia) |
| Placa 10: Nada (Protanopia) |
| Placa 11: 2 (Deuteranopia) |
| Placa 12: 91 (Deuteranopia) |
| Placa 13: Nada (Protanopia) |
| Placa 14: Nada (Protanopia) |
| Placa 15: Nada (Protanopia) |
| Placa 16: Nada (Protanopia) |
| Placa 17: Nada (Protanopia) |
| Placa 18: Nada (Protanopia) |
| Placa 19: Nada (Protanopia) |
| Placa 20: Nada (Protanopia) |
| Placa 21: Nada (Protanopia) |
| Placa 22: 2 (Deuteranopia) |
| Placa 23: 4 (Deuteranopia) |
| Placa 24: 35 (Normal) |
| Placa 25: 96 (Normal) |
| Placa 26: Uma linha (Deuteranopia) |
| Placa 27: Uma linha e bolinhas (Deuteranopia) |
| Placa 28: Nada (Protanopia) |
| Placa 29: Nada (Protanopia) |
| Placa 30: Nada (Protanopia) |
| Placa 31: Uma linha (Normal) |
| Placa 33: Nada (Protanopia) |
| Placa 34: Dois traços (Deuteranopia) |
| Placa 35: Uma linha (Normal) |
| Placa 36: Nada (Deuteranopia) |
| Placa 37: Uma linha (Normal) |
| Prognóstico Final: Protanopia |

Fig. 6. Teste com resultado Protanopia. Imagem do autor.

| |
|---|
| Placa 1: 12 (Normal) |
| Placa 2: 3 (Deuteranopia) |
| Placa 3: Nada (Deuteranopia) |
| Placa 4: 70 (Deuteranopia) |
| Placa 5: 57 (Normal) |
| Placa 6: 2 (Deuteranopia) |
| Placa 7: 5 (Deuteranopia) |
| Placa 8: 17 (Deuteranopia) |
| Placa 9: 21 (Deuteranopia) |
| Placa 10: Nada (Deuteranopia) |
| Placa 11: 2 (Deuteranopia) |
| Placa 12: 91 (Deuteranopia) |
| Placa 13: Nada (Deuteranopia) |
| Placa 14: Nada (Deuteranopia) |
| Placa 15: Nada (Deuteranopia) |
| Placa 16: Nada (Deuteranopia) |
| Placa 17: Nada (Deuteranopia) |
| Placa 18: Nada (Deuteranopia) |
| Placa 19: Nada (Deuteranopia) |
| Placa 20: Nada (Deuteranopia) |
| Placa 21: Nada (Deuteranopia) |
| Placa 22: 2 (Deuteranopia) |
| Placa 23: 4 (Deuteranopia) |
| Placa 24: 35 (Normal) |
| Placa 25: 96 (Normal) |
| Placa 26: Uma linha (Deuteranopia) |
| Placa 27: Uma linha e bolinhas (Deuteranopia) |
| Placa 28: Nada (Deuteranopia) |
| Placa 29: Nada (Deuteranopia) |
| Placa 30: Nada (Protanopia) |
| Placa 31: Uma linha (Normal) |
| Placa 33: Nada (Protanopia) |
| Placa 34: Dois traços (Deuteranopia) |
| Placa 35: Uma linha (Normal) |
| Placa 36: Uma linha (Normal) |
| Placa 37: Uma linha (Normal) |
| Prognóstico Final: Deuteranopia |

Fig. 7. Teste com resultado Deuteranopia. Imagem do autor.

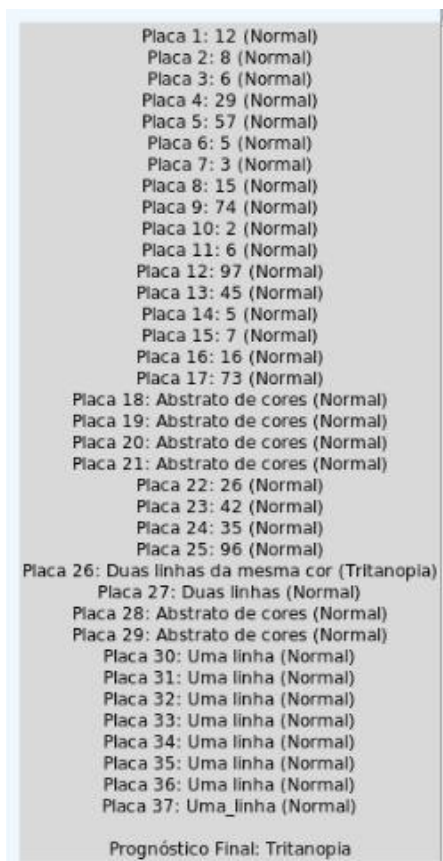


Fig. 8. Teste com resultado Tritanopia. Imagem do autor.

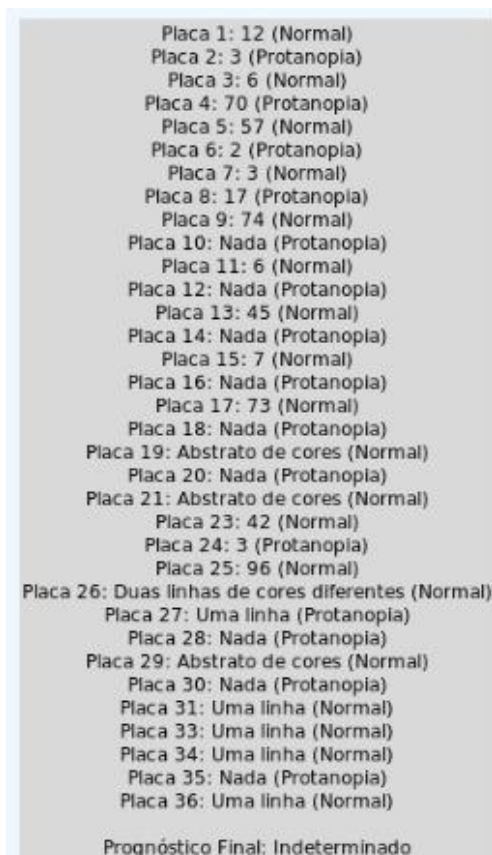


Fig. 9. Teste com resultado Indeterminado. Imagem do autor.

Estes resultados são impressos na janela Ishihara. O arquivo PDF é gerado apenas quando o usuário seleciona o botão Gerar PDF. Um exemplo do arquivo gerado está abaixo.

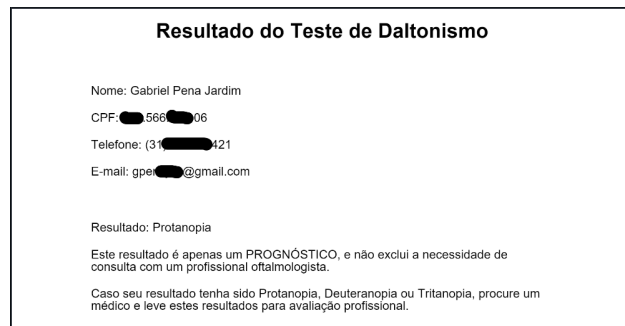


Fig. 10. Arquivo Resultado_Gabriel.pdf. Sendo apenas um prognóstico, é encorajado que o usuário procure um oftalmologista em caso de resultado diferente do Visão Normal.

Com isso, a proposta do projeto Teste de Daltonismo foi concluída com êxito, visando a acessibilidade, modernização, automatização e, não menos importante, o prognóstico. Embora os resultados iniciais sejam promissores, há espaço para melhorias no sistema, como a inclusão de testes adicionais para outros tipos de daltonismo e melhorias na interface gráfica para torná-la ainda mais intuitiva.

Em suma, os resultados obtidos demonstram que o projeto Teste de Daltonismo desenvolvido é uma ferramenta que pode ser utilizada para a detecção e análise de daltonismo. A combinação de tecnologias modernas e princípios sólidos de Programação Orientada a Objetos garantiu a criação de um sistema robusto, seguro e fácil de usar, com potencial para contribuir significativamente para a área de oftalmologia.

REFERÊNCIAS

1. National Eye Institute(<https://www.nei.nih.gov/learn-about-eye-health/eye-conditions-and-diseases/color-blindness>). Acesso em 17 jun 24
2. Colour Blind Awareness(<https://www.colourblindawareness.org/colour-blindness/types-of-colour-blindness/>). Acesso em 17 jun 24
3. Link do github(<https://github.com/anabeatriz-0702/PF-TesteDaltonismo>)
4. Design Inclusivo - Desenhando em cores para todos(<https://brasil.uxdesign.cc/design-inclusivo-desenhando-com-cores-para-todos-3359bcec1a9e>). Acesso em 18 jun 24
5. tkinter - Python Interface to Tcl/Tk(<https://docs.python.org/pt-br/3/library/tkinter.html>). Acesso em 30 mai 24