**Fundação Vale Paraibana de Ensino**

**Colégio Técnico “Antônio Teixeira Fernandes”**

**Curso Técnico em Informática**

**Gabriel Costa Fileno**

**Kaio Eduardo Braga Barbosa**

**Rafael Augusto Guimarães da Silva**

**SISTEMA DE CRIAÇÃO E CORREÇÃO DE PROVAS DE MÚLTIPLA ESCOLHA COM EMBARALHAMENTO DE GABARITOS INTEGRADO**

São José dos Campos, SP

2024

Gabriel Costa Fileno

Kaio Eduardo Braga Barbosa

Rafael Augusto Guimarães da Silva

SISTEMA DE CRIAÇÃO E CORREÇÃO DE PROVAS DE MÚLTIPLA ESCOLHA COM EMBARALHAMENTO DE GABARITOS INTEGRADO

**Relatório Final apresentado ao Colégio Univap – Unidade Centro, como parte das exigências do Curso Técnico em Informática, para obtenção do Título Técnico em Informática.**

**Orientador: Prof. Me. Hélio Lourenço Esperidião Ferreira.**

São José dos Campos, SP

2024

***Frase motivational***

**Agradecimentos**

**Resumo**

O ambiente escolar dispõe de diversas tarefas repetitivas em todos os contextos. Sendo assim, a informatização dos processos que permitem tal implementação resulta em maior desempenho e desfruto da carga laboral de professores e funcionários do corpo educacional. Desta forma, formula-se uma solução indubitavelmente positiva para a redução dos processos repetitivos – neste caso - a criação e correção de avaliações. Utilizando uma abordagem baseada na estrutura *Model-View-Controller* (MVC) através de tecnologias como Python, C#, e MongoDB, que se comunicam através de *WebSockets*. O sistema consiste em uma aplicação *Desktop* que permite ao funcionário manipular uma extensa carga de informações letivas – alunos, avaliações e notas – de maneira fluida e consistente. Neste contexto, este trabalho propõe módulos de criação de provas personificadas, embaralhadas e, por conseguinte, correção destas. Além disso, o monitoramento do momento processual em que a avaliação está presente.

**Palavras-chave:** Criação de provas; Correção de Provas;

**SUMÁRIO**

INTRODUÇÃO

Com base nos dados apresentados na pesquisa "Tempo de trabalho e de ensino: composição da jornada de trabalho dos professores paulistas", fica evidente a extensa carga horária extraclasse realizada pelos docentes, com cerca de 97% dos entrevistados declarando dedicar horas além do contrato para essas atividades. A execução de tarefas mecânicas e unidirecionais resulta no aumento das jornadas de trabalho, diminuindo a disponibilidade para outros processos pedagógicos. Assim, a redução de processos avaliativos repetitivos, como a aleatorização de gabarito e correção de provas objetivas, pode beneficiar o desempenho dos educadores nas áreas onde seu conhecimento é essencial. (BARBOSA, 2020).

O Reconhecimento Óptico de Marca (OMR) é amplamente utilizado em uma variedade de setores e contextos. Na educação, o OMR é comumente utilizado a devido à sua capacidade de processar grandes volumes de respostas de forma rápida e eficiente, fornecendo respostas instantâneas aos alunos e facilitando a avaliação do desempenho. Tal algoritmo passou por evoluções desde sua fundamentação inicial permitindo, atualmente, ser aplicado facilmente em computadores pessoais e scanners de menor complexidade. (DE ELIAS, TASINAFFO e HIRATA, 2021)

O projeto proposto visa desenvolver uma aplicação capaz de aleatorizar provas, gerando gabaritos diferentes e provas personalizadas a partir de documentos do Word. Posteriormente, o software realizará a correção das avaliações escaneadas e gerará os resultados correspondentes, além de possibilitar o retorno desses resultados ao corpo estudantil e a geração de relatórios sobre as avaliações realizadas, visando aprimorar o sistema avaliativo.

Para implementar esse projeto, será utilizada a linguagem de programação Python, devido à sua facilidade de manutenção, orientação a objetos, interpretação e disponibilidade de uma vasta gama de bibliotecas (PYTHON SOFTWARE FOUNDATION, 2024). A biblioteca OpenCV será empregada no processo de correção das avaliações, ao utilizar algoritmos de processamento de imagens que identificarão as provas e as respostas dos alunos. No processo de aleatorização de avaliações, serão utilizadas ferramentas do Python para manipular arquivos do Word através da sua estrutura XML. Além disso, as interfaces gráficas serão desenvolvidas com a ferramenta RAD (*Rapid Application Development*) proporcionada pelo Visual Studio (MICROSOFT, 2023). Por fim, a aplicação utilizará o banco de dados não relacional MongoDB para armazenar as informações necessárias do processo avaliativo. Todo o sistema estará conectado via *websockets* devido a sua propriedade de dinâmica comunicação bidirecional assíncrona que, por sua vez, permite a realização de processos sem que a fluidez da interface visual seja prejudicada.

DESENVOLVIMENTO

FERRAMENTAS E MÉTODOS UTILIZADOS

O Sistema proposto foi construído de forma dedicada ao funcionamento local, em uma composição *hardware* que consiste: Impressora, Scanner e Computador. Sendo assim, proporcionar uma aplicação *Desktop\** é a solução mais adequada, isto se deve pela sua fácil usabilidade por parte do usuário, além da integração com o sistema operacional e os dispositivos conectados sem que haja conexão à internet. Neste contexto, a decisão de utilizar o C# (C Sharp) como linguagem no desenvolvimento *Front-End* corrobora. (MICROSOFT, 2023)

Outrossim, devido a interoperabilidade Microsoft proporcionada pela linguagem C#, o acionamento de tarefas nativas do sistema operacional através do ambiente de desenvolvimento (Visual Studio) e da construção da interface por meio da ferramenta *Rapid Application Development* (RAD) proporcionada pelo mesmo, a manutenção do projeto se torna essencialmente aprimorada. (MICROSOFT, 2023)

Enquanto isso, nos procedimentos automatizados que o sistema promete, a utilização da linguagem de programação Python foi necessária em toda a extensão do sistema. Isto se deve pela vasta gama de biblioteca e integrações da linguagem, visto que, a mesma, é uma linguagem de Fonte Livre, ou seja, está em constante desenvolvimento pela comunidade de desenvolvedores (PYTHON SOFTWARE FOUNDATION, 2024). Sendo assim, a utilização de bibliotecas já existentes permitiu que a criação de avaliações distintas entre si, correção das mesmas e, por conseguinte, o armazenamento e controle dos dados avaliativos se tornou, em suma, de fácil manipulação.

No contexto introduzido, a utilização de uma estrutura de dados não relacional se tornou indispensável devido forte ramificação das informações e necessidade de otimização de buscas e manipulação das informações. Desta forma, a implementação do MongoDB se justifica, tendo em consideração o fato de que o mesmo possui uma arquitetura NoSQL capaz de realizar buscas de forma rápida e reduzir o número de arquivos gerados pelos processos da aplicação.

Por fim, a integração entre os módulos é baseada na arquitetura *Model-View-Controller* (MVC) através de *WebSockets*, que por sua vez, criam um túnel de comunicação assíncrona entre ambos, permitindo assim, a troca de informações em tempo real. Assim sendo, gera uma maior fluidez na interface gráfica ao isolar processos que não dependem de intervenção do usuário.

MVC, WebSockets, Python, C#, MongoDB

DIAGRAMAS

CASOS DE USO

ANÁLISE DE CUSTO

Neste viés, o projeto demanda de um

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PROFISSÃO | SALÁRIO MÉDIO | SALÁRIO/h | HORAS TOTAIS | TOTAL |
| Desenvolvedor Júnior C# | 4000,00 | 25,00 |  |  |
| Desenvolvedor Júnior Python | 2500,00 | 15,62 | 160 + 8 |  |
| Analista Júnior de Banco de Dados | 3000 | 18,75 |  |  |
| Analista Júnior de Sistemas |  |  |  |  |