

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
GOIÁS CAMPUS ITUMBIARA
ENGENHARIA DE CONTROLE E AUTOMAÇÃO



TRABALHO DE CONCLUSÃO DE CURSO

Desenvolvimento de um *software* interativo para automação da avaliação de processos de horas complementares

Mateus Lucas de Almeida

Itumbiara/GO
2023



INSTITUTO FEDERAL
Goiás

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
SISTEMA INTEGRADO DE BIBLIOTECAS

TERMO DE AUTORIZAÇÃO PARA DISPONIBILIZAÇÃO NO REPOSITÓRIO DIGITAL DO IFG - ReDi IFG

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia de Goiás, a disponibilizar gratuitamente o documento no Repositório Digital (ReDi IFG), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IFG.

Identificação da Produção Técnico-Científica

- | | |
|--|---|
| ☐ Tese | ☐ Artigo Científico |
| ☐ Dissertação | ☐ Capítulo de Livro |
| ☐ Monografia – Especialização | ☐ Livro |
| ☒ TCC - Graduação | ☐ Trabalho Apresentado em Evento |
| ☐ Produto Técnico e Educacional - Tipo: _____ | |

Nome Completo do Autor: MATEUS LUCAS DE ALMEIDA

Matrícula: 20161040100267

Título do Trabalho: Desenvolvimento de um software interativo para automação da avaliação de processos de horas complementares

Autorização - Marque uma das opções

1. (**x**) Autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso aberto);
2. () Autorizo disponibilizar meu trabalho no Repositório Digital do IFG somente após a data ____/____/____ (Embargo);
3. () Não autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso restrito).

Ao indicar a opção **2 ou 3**, marque a justificativa:

- (**x**) O documento está sujeito a registro de patente.
() O documento pode vir a ser publicado como livro, capítulo de livro ou artigo.
() Outra justificativa: _____

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a autor/a declara que:

- i. o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
- ii. obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia de Goiás os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
- iii. cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia de Goiás.

_____, 27 / 06 /2023.
Local Data

Mateus Lucas de Almeida

Assinatura do Autor e/ou Detentor dos Direitos Autorais

MATEUS LUCAS DE ALMEIDA

**DESENVOLVIMENTO DE UM SOFTWARE INTERATIVO PARA
AUTOMAÇÃO DA AVALIAÇÃO DE PROCESSOS DE HORAS
COMPLEMENTARES**

Trabalho de conclusão de curso apresentado à banca examinadora, em cumprimento às exigências legais como requisito parcial à obtenção do grau de Bacharel em Engenharia de Controle e Automação no Instituto Federal de Educação, Ciência e Tecnologia de Goiás – Campus Itumbiara.

Prof. Me. Wellington do Prado - Orientador
Prof. Dr. Thales Lima Oliveira - Coorientador

Dados Internacionais de Catalogação na Publicação (CIP)

A477d Almeida, Mateus Lucas de
Desenvolvimento de um software interativo para automação da
avaliação de processos de horas complementares. / Mateus Lucas
de Almeida. -- Itumbiara, 2023.
45p. : il.

Trabalho de conclusão de curso (Bacharel em Engenharia de
Controle e Automação) - Instituto Federal de Educação, Ciência
e Tecnologia de Goiás, Câmpus Itumbiara, 2023.

Orientador: Prof. Me. Wellington do Prado.
Coorientador: Thales Lima Oliveira.

1. Software - Desenvolvimento. 2. Administração pública.
3. Horário de trabalho. I. Título. II. Prado, Wellington do.
III. Oliveira, Thales Lima. IV. Instituto Federal de Educação,
Ciência e Tecnologia de Goiás.

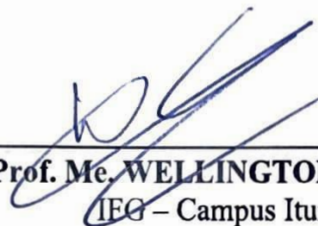
CDD 006.76

Instituto Federal de Educação, Ciência e Tecnologia de Goiás
Biblioteca Maria Gabriela Pacheco Pardey / Câmpus Itumbiara
Bibliotecária: Rosiane Gonçalves de Lima CRB1/1684

MATEUS LUCAS DE ALMEIDA
Desenvolvimento de um software interativo para a automação da avaliação de processos de horas complementares

Trabalho de Conclusão de Curso apresentado à banca avaliadora, parte dos requisitos para a obtenção do título de bacharel em Engenharia de Controle e Automação junto ao Instituto Federal de Educação, Ciência e Tecnologia de Goiás (IFG), Câmpus Itumbiara.

Aprovado em 23 de Junho de 2023.

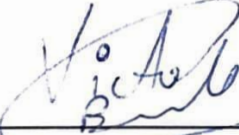


Prof. Me. WELLINGTON DO PRADO
IFG – Câmpus Itumbiara
Orientadora

Thales Lima
Oliveira:05908
429600

Assinado digitalmente por Thales Lima
Oliveira:05908429600
ND: CN=Thales Lima Oliveira:05908429600,
OU=UFU - Universidade Federal de
Uberlândia, O=ICPEdu, C=BR
Razão: Eu estou aprovando este documento
Localização:
Data: 2023.06.29 11:05:42-03'00'
Foxit PDF Reader Versão: 12.1.0

Prof. Dr. Thales Lima Oliveira
IFG – Câmpus Itumbiara
Coorientadora



Prof. Dr Victor Reges Bernadeli
IFG – Câmpus Itumbiara
Avaliadora



Prof. Dr Claudio Roberto Pacheco,
IFG – Câmpus Itumbiara
Avaliador

Itumbiara/GO
2023

Dedico este trabalho à minha família, que sempre me incentivou a buscar a educação e me deu o suporte necessário para que eu pudesse alcançar meus objetivos. Aos meus amigos, que me ajudaram a manter o equilíbrio e a motivação durante os períodos de estudo intenso. Aos meus professores e orientadores, pela orientação, inspiração e paciência ao longo deste trabalho. Dedico este trabalho também à comunidade acadêmica, por proporcionar um ambiente de aprendizagem enriquecedor e estimulante.

AGRADECIMENTOS

Gostaria de começar agradecendo a Deus por me conceder a oportunidade de contar com orientadores tão dedicados e competentes como vocês, Wellington do Prado e Thales Lima Oliveira, durante todo o processo de elaboração do meu TCC. Sem a sua orientação e o seu amor, nada seria possível.

Além disso, gostaria de expressar minha gratidão aos meus amigos, colegas e professores das demais áreas que me apoiaram e me encorajaram durante todo o processo. Seus conselhos e incentivos foram fundamentais para que eu persistisse e não desistisse em momentos difíceis.

Agradeço também aos colegas de curso que se dedicaram e compartilharam conhecimentos e experiências comigo. Juntos, pudemos enfrentar os desafios que surgiram ao longo do caminho e aprendemos muito uns com os outros.

Por fim, não posso deixar de mencionar a gratidão que sinto pelos professores das demais áreas que me ensinaram e me inspiraram a buscar a excelência em meu trabalho. Seus conhecimentos e sabedoria foram fundamentais para minha formação e crescimento como estudante e como pessoa.

Wellington e Thales, vocês foram exemplos de dedicação, amor e paixão pelo que fazem. O comprometimento com o meu aprendizado e sucesso acadêmico foi inspirador e me motivou a buscar sempre o melhor.

Mais uma vez, agradeço a Deus por ter colocado vocês em meu caminho e por permitir que eu tivesse a honra de tê-los como meus orientadores. Agradeço também a todos aqueles que me apoiaram nessa jornada. Juntos, conseguimos alcançar esse importante marco em minha vida acadêmica e profissional.

"O temor do Senhor é o princípio da sabedoria; todos os que cumprem os seus preceitos revelam bom senso. Ele será louvado para sempre!" - Salmos 111:10

RESUMO

De acordo com o tema trabalhado no desenvolvimento de um software interativo para automação da avaliação de processos de horas complementares, situamos o contexto sobre o desenvolvimento de uma ferramenta computacional livre e interativa que visa automatizar o processo de avaliação das horas complementares. A escolha desse tema se deu pelo fato de que essa ferramenta pode contribuir para o aumento da eficiência e redução de erros na administração pública, permitindo a inclusão de novos algoritmos e soluções. O objetivo geral é desenvolver um software livre e de código aberto que possibilite a avaliação das horas complementares pela CAAC de forma interativa. Com isso, o produto estará apto a consultar o banco de dados de processos anteriores, evitando erros e proporcionando a possibilidade de interatividade entre os avaliadores.

Palavras-chave: Software. Avaliação de processos de horas complementares. Ferramenta computacional.

ABSTRACT

According to the theme addressed in the development of an interactive software for automation of complementary hours evaluation processes, we outline the context of developing a free and interactive computational tool that aims to automate the evaluation process of complementary hours. The choice of this theme was made based on the fact that this tool can contribute to increased efficiency and error reduction in public administration, allowing the inclusion of new algorithms and solutions. The overall objective is to develop a free and open-source software that enables the evaluation of complementary hours by CAAC in an interactive manner. With this, the product will be able to consult the database of previous processes, avoiding errors and providing the possibility of interactivity among the assessors.

Keywords: Software. Evaluation of additional hours processes. Computational tool.

LISTA DE IMAGENS

Figura 01: Janela de compilação do código.....	24
Figura 02: Interface de desenvolvimento do CodeLite.....	27
Figura 03: Área de projeto.....	29
Figura 04: Avaliador CAAC.....	31
Figura 05: Novo processo.....	31
Figura 06: Avaliador CAAC.....	32
Figura 07: Buscar no Drive.....	33
Figura 08: Comentário do Processo.....	33
Figura 09: Avaliador CAAC.....	34
Figura 10: Relatório de validação.....	34

LISTA DE SIGLAS E ABREVIATURAS

API	<i>Application Programming Interface</i> (Interface de Programação de Aplicações)
Bell Labs	Centro de pesquisa da empresa americana Bell Telephone Laboratories.
Bitmap	Imagem digital composta por pixels
.cpp	<i>C++ Source File</i> (Arquivo de Código-fonte em C++)
C++	Linguagem de programação C++
C++11	Versão 11 da linguagem C++
C++14	Versão 14 da linguagem C++
C++17	Versão 17 da linguagem C++
CAAC	Comissão de Avaliação de Atividades Complementares
Clang:	<i>C Language Family Frontend for LLVM</i> (Frontend da Família de Linguagem C para o LLVM - compilador LLVM para linguagens C, C++, Objective-C e Objective-C++)
ClibPDF	Biblioteca em linguagem C para manipulação de arquivos PDF
CMYK	<i>Cyan, Magenta, Yellow, Black</i> (Ciano, Magenta, Amarelo, Preto)
Codelite	É um IDE gratuito de código aberto e multiplataforma para a linguagem de programação C/C++
DTD	<i>Document Type Definition</i> (Definição de Tipo de Documento)
Eclipse:	Ambiente de Desenvolvimento Integrado de Código Aberto
etc:	<i>et cetera</i> (e outros)
FLTK	<i>Fast Light Toolkit</i> (Kit de ferramentas para desenvolvimento de interface gráfica do usuário, de baixo consumo de recursos)
FOSS	<i>Free and Open-Source Software</i> (Software Livre e de Código Aberto)
FPDF	<i>Free PDF</i> (PDF livre - Biblioteca de código aberto para PHP que permite gerar arquivos PDF a partir de dados dinâmicos)
FTP	<i>File Transfer Protocol</i> (Protocolo de transferência de arquivos)
GB	<i>Gigabyte</i> (Unidade de medida de armazenamento de dados, equivalente a aproximadamente 1 bilhão de bytes)

GCC	<i>GNU Compiler Collection</i> (Coleção de Compiladores GNU - Compilador GNU para várias linguagens de programação)
GIF	<i>Graphics Interchange Format</i> (Formato de Intercâmbio de Gráficos)
Git	<i>Global Information Tracker</i> (Rastreador Global de Informações)
Google Drive	Ferramenta de armazenamento em nuvem
GPL	<i>GNU General Public License</i> (Licença Pública Geral GNU)
GTK+	<i>GIMP ToolKit+</i> (Kit de ferramentas para desenvolvimento de interface gráfica do usuário, originalmente desenvolvido para o GIMP)
GUI	<i>Graphical User Interface</i> (Interface Gráfica do Usuário)
HTML:	<i>HyperText Markup Language</i> (Linguagem de marcação de hipertexto)
.h	<i>Header File</i> (Arquivo de Cabeçalho)
IDE:	<i>Integrated Development Environment</i> (Ambiente de desenvolvimento integrado)
IFG	Instituto Federal de Educação, Ciência e Tecnologia de Goiás
iOS:	<i>iPhone Operating System</i> (Sistema operacional móvel da Apple)
Java	Linguagem de programação para aplicações desktop
JavaScript	Linguagem de programação de scripts
JPEG	<i>Joint Photographic Experts Group</i> (Grupo de Especialistas em Fotografia Conjunta)
JSON	JavaScript Object Notation (Notação de Objetos JavaScript)
Mercurial:	<i>Distributed Version Control System</i> (Sistema de Controle de Versão Distribuído)
MFC	<i>Microsoft Foundation Class</i> (Conjunto de classes de programação C++ usadas para desenvolvimento de aplicativos para Windows, originalmente desenvolvido pela Microsoft)
Numpy	Numerical Python (Python Numérico)
Objective-C	Linguagem de programação para aplicativos iOS
Open-source	Software de código aberto
OpenType	Padrão de fonte escalável desenvolvido pela Microsoft e Adobe
OSX:	<i>Mac Operating System</i> (Sistema Operacional Mac)

PDF	<i>Portable Document Format</i> (Formato Portátil de Documento)
PDFlib	Biblioteca de geração de PDFs
PHP	<i>Hypertext Preprocessor</i> (Pré-processador de Hipertexto - linguagem de programação de código aberto, especialmente projetada para desenvolvimento de aplicações web e de servidor)
Phyton	Linguagem de programação
PNG	<i>Portable Network Graphics</i> (Gráficos Portáteis de Rede)
POO	Programação Orientada a Objetos
pytest	<i>Testing framework for Python</i> (Framework de teste para Python)
PyTorch	<i>Python-based scientific computing package for machine learning</i> (Pacote de computação científica baseado em Python para aprendizado de máquina)
Qt	<i>Biblioteca Qt</i> (framework multiplataforma de desenvolvimento de aplicativos)
RAM	<i>Random Access Memory</i> (Memória de Acesso Aleatório)
RapidXml	Biblioteca de análise de XML de código aberto e rápida, que oferece uma maneira eficiente de analisar documentos XML
RGB	<i>Red, Green, Blue</i> (Vermelho, Verde, Azul)
Rust	Linguagem de programação multiparadigma e compilada
SDK:	<i>Software Development Kit</i> (Kit de desenvolvimento de software)
Std	Biblioteca padrão de C++
STL	<i>Standard Template Library</i> (Biblioteca de modelos padrão)
SVN:	<i>Apache Subversion</i> (Sistema de controle de versão centralizado)
TDM-GCC	<i>Toolchain Distribution Manager GCC</i> (Distribuição do Compilador GCC pelo Gerenciador de Cadeias de Ferramentas)
TensorFlow	<i>Open-source machine learning framework</i> (Framework de aprendizado de máquina de código aberto)
The Linker:	<i>The Linker</i> (O vinculador, um programa que converte o código fonte em código de máquina binário)
TICs	Tecnologias de Informação e Comunicação
TrueType	Tipo de fonte escalável
Type1	Tipo de fonte de formato de arquivo específico
Unicode	Padrão de codificação de caracteres universal

Visual C++:	<i>Visual C++ Compiler</i> (Compilador Visual C++ - Ambiente de desenvolvimento integrado da Microsoft para a linguagem C++)
Visual Studio:	Ambiente de Desenvolvimento Integrado da Microsoft
wizard:	Assistente de criação de projetos
WMF	<i>Windows Metafile Format</i> (Formato de Arquivo de Metafile do Windows)
wxcrafter	Ferramenta de criação de interfaces gráficas de usuário usando o framework (Conjunto de Ferramentas e Bibliotecas) wxWidgets
wxGTK	<i>wxWidgets Gimp Toolkit</i> (Kit de ferramentas wxWidgets para o GIMP)
wxMac	<i>wxWidgets Macintosh</i> (Kit de ferramentas wxWidgets para Macintosh)
wxMotif	<i>wxWidgets Motif</i> (Kit de ferramentas wxWidgets para Motif).
wxMSW	<i>wxWidgets Microsoft Windows</i> (Kit de ferramentas wxWidgets para Microsoft Windows)
wxOSX/Carbon	<i>wxWidgets Mac OS X Carbon</i> (Kit de ferramentas wxWidgets para Mac OS X Carbon)
wxOSX/Cocoa	<i>wxWidgets Mac OS X Cocoa</i> (Kit de ferramentas wxWidgets para Mac OS X Cocoa)
wxPdfDocument	Classe C++ para geração de arquivos PDF utilizando wxWidgets.
wxQt	<i>wxWidgets Qt</i> (Kit de ferramentas wxWidgets para Qt)
wxWidgets	<i>Cross-Platform GUI Library</i> (Biblioteca de Interface Gráfica de Usuário Multiplataforma)
wxX11	<i>wxWidgets X11</i> (Kit de ferramentas wxWidgets para X11)
XLM	<i>Extensible Markup Language</i> (Linguagem de marcação extensível - É uma linguagem de marcação usada para armazenar e transmitir dados)
XML Schema	Esquema XML

SUMÁRIO

1 INTRODUÇÃO	14
2 WXWIDGETS.....	16
2.1 Desenvolvimento multiplataforma com wxWidgets: Opções de compilação e utilização em aplicativos comerciais e de código aberto.....	19
2.2 Desenvolvimento de aplicações desktop: Segurança e estabilidade para o usuário	20
2.3 Desenvolvimento do Audacity com wxWidgets: Um estudo de caso de sucesso multiplataforma.....	21
2.4 CodeLite: Um IDE de plataforma aberta com suporte para wxWidgets	22
3 FERRAMENTAS E BIBLIOTECAS UTILIZADAS NO DESENVOLVIMENTO DO SOFTWARE	24
3.1 Rapidxml: uma biblioteca eficiente para parsing de arquivos xml	25
3.2 Wxpdfdocument: biblioteca C++ para geração de arquivos pdf com wxwidgets	26
4 DESENVOLVIMENTO DO SOFTWARE	28
4.1 WxCrafter: Ferramenta visual para desenvolvimento de interfaces gráficas	29
4.2 Software de Avaliação de Atividades Complementares (CAAC): Recursos e Funcionalidades.....	31
4.3 C++: Uma linguagem de programação de alto desempenho e flexibilidade	36
4.4 PYTHON: Uma linguagem versátil e de fácil aprendizado	38
5 CONCLUSÃO.....	41
REFERÊNCIAS.....	43
APÊNDICES.....	46

1 INTRODUÇÃO

A adoção de tecnologias de informação e comunicação (TICs) em organizações do setor público tem sido frequentemente associada a programas de reforma com o objetivo de reduzir as ineficiências geradas pela carga burocrática.

A implementação de TIC para automatizar os procedimentos administrativos existentes poderia melhorar a eficiência e eficácia do sistema administrativo. Mecanismos de monitoramento e controle mais eficazes, eficientes habilitados pelas tecnologias de TIC, podem de fato, provar soluções valiosas para o design e a implementação de organizações burocráticas mais funcionais, aumentando a homogeneidade e previsibilidade dos procedimentos administrativos e seu alinhamento com o quadro normativo e legal que rege cada burocracia do setor público.

Destaca-se a implementação de TICs com a utilização de programas gratuitos e de código aberto (FOSS, do inglês *Free and Open Source Software*). Os FOSSs são soluções ideais para aplicação na área educacional e para a pesquisa, pois permite ao programador a possibilidade de uma completa modificação e adaptação do programa para adequação aos problemas estudados.

Observa-se que esse processo realizado de forma manual é ineficiente e sujeito a erros. É nesse contexto que a proposta de trabalho se encontra sustentação: o desenvolvimento de uma ferramenta computacional livre e interativa que automatize o processo de avaliação das horas complementares é de fundamental importância para o serviço público no âmbito do IFG.

E o desenvolvimento de um software que automatize o processo de avaliação de horas complementares e gere interação entre os avaliadores para uma correta revisão do trabalho é de fundamental importância no âmbito do IFG, contribuindo para o aumento da eficiência e diminuição de erros da administração pública. Além disso, o software desenvolvido de forma livre e de código aberto permitirá a inclusão de novos algoritmos e soluções, evitando assim sua obsolescência.

O principal objetivo deste projeto é desenvolver um software livre e de código aberto que permita a avaliação das horas complementares pela CAAC de forma interativa, eficiente e automatizada. No produto final, será possível inserir as descrições de forma automatizada, aplicar restrições e contabilizar a pontuação do

estudante automaticamente, consultar os bancos de dados de processos anteriores para evitar erros e a possibilitar a interatividade entre os avaliadores.

Como método para a criação do software proposto, será usado somente um computador com as ferramentas de desenvolvimento necessárias para a sua implementação. O programa será desenvolvido principalmente na linguagem de programação C++, por possuir as seguintes características principais. Com isso a linguagem de programação não se torna obsoleta, uma vez que novas soluções são implementadas na biblioteca std (Padrão, do inglês, *Standard*). Também é previsto a utilização da linguagem Python, para desenvolvimento de scripts necessários para integração, por meio de uma API (Interface de Programação de Aplicações, do inglês *Application Programming Interface*) com o Google Drive, o qual será utilizado como banco de dados e ferramenta de integração com os avaliadores. Visa-se a utilização de bibliotecas de domínio público e de código aberto, permitindo assim, o desenvolvimento de um software completamente gratuito e sem restrições de licenças protetoras.

Para produzir a GUI (Interface Gráfica do Usuário, do inglês, *Graphical User Interface*) para inserção dos documentos do processo e interface com os demais controles do usuário, será empregado o wxWidgets, que fornece aplicações com aparência nativa, ou seja, os elementos da interface gráfica, como abas, barras de rolagem, menus, etc. São iguais ao tema de cada sistema operacional. Isso é devido à utilização de widgets nativos sempre que possível durante o desenvolvimento do wxWidgets (SMART e HOCK, 2003). Para administração dos dados que serão gravados no disco e no Google Drive, é utilizada a biblioteca RapidXML (KALICINSKI, 2020), empregada na gravação e recuperação dos arquivos no disco por meio do padrão XML. Finalmente, para geração automática do relatório final a ser enviado pela CAAC, será utilizada a biblioteca wxPdfDocument

E como resultado esperado, será disponibilizada uma ferramenta computacional livre e de código aberto, em que será possível inserir as descrições dos documentos relativos às horas complementares de forma automatizada, aplicar as restrições e contabilizar a pontuação do estudante automaticamente, consultar aos bancos de dados de processos anteriores para evitar erros e a possibilitar a interatividade entre os avaliadores da CAAC.

2 WXWIDGETS

Ao longo dos anos, o wxWindows evoluiu para se tornar o wxWidgets. A versão atual é a 3.1.4, lançada em dezembro de 2020. Ele continua sendo desenvolvido e mantido pela comunidade de desenvolvedores, incluindo uma série de novos recursos e melhorias, entre eles a adição de suporte para Unicode, melhorias na performance, suporte para o uso de múltiplas janelas, adição de recursos de depuração e gerenciamento de projetos, melhorias na documentação e suporte para aplicativos nativos para mobile (ZEITLIN, 2023).

De acordo com Zeitlin (2023) o Wxwidgets é voltado para programadores, que permite desenvolver aplicativos que possam funcionar no Windows e Unix/Linux, além de Mac OS X, possuindo linguagem de programa C++, Python, Perl e C#/.NET, sendo um aplicativo grátis com código aberto.

Segundo Silva (2011) cita em seu contexto que o Wxwidgets foi iniciado em 1992 na universidade de Edimburgo, por Julian Smart, a qual deu início a um projeto que teria como objetivo criar um aplicativo portátil em Unix/Linux e Windows, crescendo e oferecendo suporte até mesmo a Mac OS X e outras plataformas, que desde então tem sido amplamente utilizada em aplicativos comerciais. Ele conecta-se a uma biblioteca a qual oferecer ajuda online e muita transparência na parte de carregamento e salvamento de imagens e muito mais.

As plataformas que são suportadas pelo Wxwidgets são (ZEITLIN, 2023):

- wxGTK: é uma porta para a biblioteca wxWidgets que é recomendada para o sistema operacional Linux e algumas variantes do Unix.
- wxMSW: é uma porta para a biblioteca wxWidgets que é recomendada para o sistema operacional Windows de 32 e 64 bits, sendo compatível com todas as versões do Windows (XP, Vista, 7, 8, 10 e 11).
- wxOSX/Cocoa: é uma porta para a biblioteca wxWidgets que é usada para desenvolver aplicativos para Mac OS X 10.10 e versões superiores, utilizando o framework Cocoa.
- wxQt: é uma porta para a biblioteca wxWidgets que é usada para a biblioteca Qt.

- wxX11: é uma porta para a biblioteca wxWidgets que é recomendada para variantes do Linux e Unix que usam o servidor de janelas X11.
- wxMotif : é uma porta para a biblioteca wxWidgets que é recomendada para variantes do Linux e Unix que usam o toolkit Motif.

suportadas pela antiga versão estável 3.0:

- wxMSW: é uma porta para a biblioteca wxWidgets que é recomendada para versões herdadas do Windows de 32 bits, como Windows 95, Windows 98 e Windows 2000.
- wxMac: é uma porta para a biblioteca wxWidgets que é usada para fornecer aplicativos Carbon no Mac OS X 10.2 a 10.6.
- wxOSX/ Carbon: é uma porta para a biblioteca wxWidgets que é usada para fornecer aplicativos baseados em Carbon de 32 bits no Mac OS X 10.5 e superior.

De acordo com Oliveira (2021), a biblioteca wxWidgets é escrita em C++ e fornece uma interface para a GUI nativa em várias plataformas, onde os aplicativos criados com wxWidgets terão um aspecto e comportamento semelhante aos aplicativos nativos em cada plataforma, sendo licenciada de código aberto similar à GPL. Isso significa que os aplicativos criados com wxWidgets podem ser distribuídos gratuitamente e modificados livremente, desde que as modificações sejam disponibilizadas sob a mesma licença.

Segundo Zeitlin (2023), entre os benefícios que o wxWidgets pode oferecer é que ele está disponível em todas as plataformas de desktop, é gratuito, sua fonte é fácil de ser modificada, possui mais de 100 programas como exemplo, seu sistema em visão de layout é simples, porém poderoso, sistema flexível, seu Api de desenho é baseado em 2D, suporte integrado e muito mais. Há alguns requisitos que vale a pena citar, assim como a questão da velocidade de CPU, quanto mais rápido for, mais rápido a biblioteca será compilada, necessitando obter RAM suficiente, principalmente em plataformas Windows, e em todas as plataformas há necessidade de obter grande espaço livre, pelo menos 1 GB ou mais.

Segundo Silva (2019) o desenvolvimento de aplicativos com wxWidgets é facilitado por diversas ferramentas e recursos, incluindo um designer de interface gráfica, documentação extensa e uma comunidade ativa de desenvolvedores, tornando a biblioteca uma escolha popular para desenvolvedores de aplicativos multiplataforma.

De acordo com Chiconi (2020) o wxWidgets é comparável a outras bibliotecas de interface gráfica de usuário (GUI) como Qt, GTK+, FLTK e MFC, pois todas elas oferecem a capacidade de desenvolvimento multiplataforma e uma variedade de componentes de interface gráfica. No entanto, cada biblioteca tem suas próprias características e vantagens únicas, assim explicando cada uma:

- O Qt é uma biblioteca de interface gráfica de usuário mais avançada, com uma ampla variedade de componentes de interface gráfica e recursos de desenvolvimento, a qual possui uma licença comercial, o que significa que os desenvolvedores precisam pagar por licenças para usar o Qt em aplicativos comerciais, suas desvantagens é que possui licença comercial, alguns componentes podem ser considerados mais complexos de usar e pode ser mais pesado em termos de recursos do sistema.
- GTK+ é uma biblioteca de interface gráfica de usuário popular para desenvolvimento de aplicativos Linux e Unix, conhecida por sua integração nativa com o ambiente de desktop do Gnome e por sua licença de código aberto, e sua desvantagens é que pode ser mais difícil de usar em comparação com outras bibliotecas, como o Qt ou o wxWidgets, e pode ter um desempenho mais lento em algumas plataformas.
- FLTK é uma biblioteca leve e rápida que é fácil de usar e possui uma curva de aprendizado mais baixa do que outras bibliotecas, como o Qt ou o wxWidgets, suas desvantagens é que tem uma menor quantidade de componentes de interface gráfica de usuário e recursos avançados em comparação com outras bibliotecas, como o Qt ou o wxWidgets.
- O MFC (Microsoft Foundation Class) é uma biblioteca de interface gráfica de usuário específica para desenvolvimento de aplicativos Windows, fornecido como parte do Visual Studio da Microsoft e é usado principalmente para desenvolvimento de aplicativos comerciais no Windows.

Mesmo obtendo benefícios, o wxWidgets possui algumas limitações, são elas (CHICONI, 2020):

- Suporte limitado para plataformas móveis.
- Personalização de aparência limitada.
- Curva de aprendizado.
- Desempenho.
- Suporte limitado para animações e efeitos visuais.

2.1 Desenvolvimento multiplataforma com wxWidgets: Opções de compilação e utilização em aplicativos comerciais e de código aberto

Após a instalação do Wxwidgets, deve-se construir a biblioteca compilando a versão mais recente ou usar os binários, que são uma opção mais simples. No entanto, a instalação dependerá muito da plataforma utilizada. Depois de obter as fontes, deve-se selecionar o sistema de compilação e, assim, criar seu aplicativo por meio de linhas de comando (ZEITLIN, 2023).

Segundo Heart (2022) o wxWidgets é utilizado tanto em aplicativos comerciais quanto em aplicativos de código aberto. É uma biblioteca de interface gráfica de usuário (GUI) amplamente utilizada e confiável, com uma longa história e uma comunidade ativa de desenvolvedores. Nos aplicativos comerciais, o wxWidgets é amplamente utilizado devido à sua capacidade de desenvolvimento multiplataforma e sua variedade de componentes de interface gráfica. É utilizado em muitos setores, como finanças, indústria, saúde e educação. Alguns dos aplicativos comerciais desenvolvidos com wxWidgets incluem o software de análise de dados e estatística, o software de gerenciamento de projetos Planner e o software de gerenciamento de laboratório LabCollector.

De acordo com Heart (2022), o wxWidgets é suportado e mantido principalmente pela comunidade de desenvolvedores, que trabalha voluntariamente reportando bugs, proporcionando soluções, fornecendo feedback, contribuindo com código e documentação e transparentemente compartilhando tudo com a comunidade de desenvolvedores.

2.2 Desenvolvimento de aplicações desktop: Segurança e estabilidade para o usuário

De acordo com Heart (2022) o fato de as empresas utilizarem vários aplicativos e softwares básicos como Word, Excel, entre outros, até mesmo programas mais avançados que buscam controlar todo um sistema interno, faz com que todo usuário queira se tornar o menos vulnerável possível aos ataques e roubos de hackers. Com isso, os softwares de desktop podem ser um grande aliado quando se trata de segurança da informação.

Ainda seguindo o contexto do autor Heart (2022), é importante frisar que as aplicações de desktop são muito poderosas e oferecem estabilidade, uma vez que fornecem recursos na área de trabalho que facilitam a vida do usuário, e quando são executadas, agem como um programa.

A linguagem de programação utilizada pela maioria das aplicações de desktop é o C++, e esse código é traduzido pelo computador e entendido como bits e bytes em software ou hardware, que são representados apenas por um e zero (HEART, 2022).

Segundo Heart (2022) os códigos fonte utilizados pelos desenvolvedores são compilados em “código de máquina”, que passa a ser interpretado por um programa chamado The Linker, que faz a conversão em código de máquina binário para que o computador possa entender as aplicações desktop.

De acordo com Heart (2022), as aplicações desenvolvidas para desktop são utilizadas com uma interface gráfica de usuário (GUI), onde os usuários escolhem itens como botões, caixa de texto, entre outros recursos necessários para operar corretamente o sistema. Lembrando que as aplicações desktop são autônomas, ou seja, não dependem de nenhum outro programa já instalado em seu computador para que possam ser executadas.

Segundo Heart (2022), as tecnologias mais utilizadas para o desenvolvimento de aplicações desktop são Java, QT e Objective-C, cada uma com suas vantagens e desvantagens, dependendo do objetivo do usuário. O QT é o mais popular para aplicações de desktop, por existir há muito tempo e ser mais familiar para os usuários. Já o Java é uma linguagem de programação poderosa, porém mais complexa de aprender, enquanto o Objective-C é mais utilizado no iOS SDK, que tem pouca demanda. Vale destacar que, para evidenciar a diferença entre aplicações desktop e

web, as aplicações desktop não são conectadas em rede por padrão, ao contrário das aplicações web que são projetadas para estar conectadas.

Temos alguns exemplos de aplicativos desenvolvidos com o wxWidgets que incluem (ZEITLIN, 2023):

- Audacity: editor de áudio de código aberto e multiplataforma que permite aos usuários gravar, editar e processar áudio.
- Code: Blocks: ambiente de desenvolvimento integrado (IDE) de código aberto para desenvolvimento de aplicativos C++.
- FileZilla: gerenciador de FTP de código aberto e multiplataforma que permite aos usuários enviar e baixar arquivos de um servidor FTP.
- Notepad++: editor de texto de código aberto e multiplataforma que oferece recursos avançados, como destaque de sintaxe e suporte para vários idiomas.
- WxMaxima: interface gráfica para o Máxima, um sistema de álgebra computacional de código aberto.
- WxMusic: aplicativo de música de código aberto e multiplataforma que permite aos usuários reproduzir suas músicas e listas de reprodução.

Esses são apenas alguns exemplos de aplicativos desenvolvidos com o wxWidgets, porém, existem muitos outros aplicativos desenvolvidos com essa biblioteca, desde simples até os mais complexos.

2.3 Desenvolvimento do Audacity com wxWidgets: Um estudo de caso de sucesso multiplataforma

Segundo Audacity (2022), uma empresa que utiliza o wxWidgets é a Audacity Team, que desenvolveu o aplicativo de edição de áudio Audacity com o wxWidgets. O Audacity é um editor de áudio de código aberto e multiplataforma que permite aos usuários gravar, editar e processar áudio, estando disponível para Windows, Mac e Linux.

De acordo com Audacity (2022), a equipe de desenvolvimento do Audacity escolheu o wxWidgets devido à sua compatibilidade multiplataforma e à facilidade de uso. O wxWidgets permitiu que a equipe desenvolvesse o Audacity uma vez e o executasse em várias plataformas sem a necessidade de modificações significativas.

Além disso, o wxWidgets ofereceu uma grande quantidade de componentes de interface gráfica de usuário prontos para uso.

Neste estudo de caso, vamos explorar como a Audacity Team, que segundo Audacity (2022), utilizou o wxWidgets para desenvolver seu popular aplicativo de edição de áudio, o Audacity, contando com a ajuda de uma equipe de desenvolvedores voluntários que trabalham em seus tempos livres para manter e melhorar o Audacity.

O suporte para eventos do wxWidgets também foi fundamental para o desenvolvimento do Audacity, permitindo que a equipe criasse uma interface gráfica de usuário reativa, o que é essencial para um aplicativo de edição de áudio. O suporte para vários idiomas também foi importante, pois permitiu que a equipe tornasse o Audacity disponível para usuários de vários idiomas (AUDACITY, 2022).

Segundo Audacity (2022), além disso, a equipe de desenvolvimento do Audacity também teve acesso à documentação e suporte fornecidos pelo wxWidgets, o que foi fundamental para o sucesso do projeto. A equipe também pôde contribuir para o desenvolvimento do wxWidgets, reportando bugs e proporcionando soluções. Em síntese, o wxWidgets foi uma escolha ideal para o desenvolvimento do Audacity devido à sua compatibilidade multiplataforma, facilidade de uso e suporte para eventos e vários idiomas.

2.4 CodeLite: Um IDE de plataforma aberta com suporte para wxWidgets

De acordo com Oliveira (2021), o CodeLite é um ambiente de desenvolvimento integrado (IDE) de plataforma que possui código aberto, sendo especializado em linguagem de programação C, C++, Rust, Python, PHP e JavaScript, as quais funcionam nas plataformas OSX, Windows e Linux. É um projeto que foi criado para ser leve e rápido, oferecendo uma série de recursos para ajudar os desenvolvedores a escreverem, depurarem e testarem seu código.

Algumas das principais funcionalidades do CodeLite, citadas por Oliveira (2021), são incluir suporte para vários compiladores, incluindo GCC, Clang e Visual C++, para depuração integrada, controle de versão, incluindo Git, SVN e Mercurial, suporte para vários projetos, entre eles o C++, PHP e WxWidgets, refatoração de código, para snippets de código e análise de código estático. As principais vantagens do CodeLite são ser simples e fácil de utilizar, oferecendo uma série de recursos que têm como objetivo ajudar os desenvolvedores a escrever, depurar e testar seu código.

É uma ferramenta de código aberto, gratuita, com suporte para múltiplas plataformas, podendo ser usada no Windows, Linux e Mac. Em relação às suas desvantagens, diz-se que ele pode ser menos completo e menos poderoso do que outros IDEs, como o Visual Studio ou o Eclipse. Além disso, por ser uma ferramenta de código aberto, pode haver falta de suporte e documentação em comparação com as soluções comerciais.

O CodeLite possui uma opção de criação de projetos wxWidgets, que permite criar um projeto wxWidgets usando um assistente, sem precisar configurar manualmente as opções. Além disso, ele também oferece suporte para depuração e teste de projetos wxWidgets, assim como outros tipos de projetos (OLIVEIRA, 2021).

3 FERRAMENTAS E BIBLIOTECAS UTILIZADAS NO DESENVOLVIMENTO DO SOFTWARE

De acordo com Oliveira (2021), o wxWidgets é composto por uma série de bibliotecas que tem a finalidade de fornecer uma API de fácil utilização para construção de interface gráfica utilizando a linguagem C++. O mesmo código utilizado para compilar aplicações no Windows pode ser compilado em distribuições do Linux e macOS. A grande vantagem do wxWidgets é que ele gera uma interface nativa (e não emulada, como Java ou Qt).

Segundo Oliveira (2021), o CodeLite é um IDE (Ambiente Integrado de Desenvolvimento), ou seja, é um software criado para facilitar a vida dos programadores. Além disso ele possui o wxCrafter, programa que auxilia (e muito!) a criação dos formulários e janelas utilizando as bibliotecas do wxWidgets.

Abaixo segue todo o processo de instalação do wxWidgets e CodeLite, o qual é realizado por atividades:

Atividade 1: Compilação do wxWidgets

Para utilizar a última versão do wxWidgets no Windows, é preciso compilar o seu código fonte. Para isso, deve-se efetuar o download do compilador de C++ e do código fonte. O compilador que será utilizado é o tdm-gcc, que permite a compilação para as versões 32-bits e 64-bits. Sendo assim, deve-se realizar a instalação do compilador tdm-gcc e reiniciar o computador. Em seguida, insere-se no prompt de comando o seguinte código: **set PATH=C:\TDM-GCC-32\bin;%PATH%**

Utilizando o prompt, constrói-se o arquivo “setup.h” na pasta “...\wxWidgets\build\msw”, com o seguinte comando: **mingw32-make -f makefile.gcc setup_h SHARED=0 MONOLITHIC=1 UNICODE=1 BUILD=release**

Ao abrir o arquivo “setup.h”, o **wxUSE_GRAPHICS_CONTEXT** é alterado para valor 1. Exemplo: **#define wxUSE_GRAPHICS_CONTEXT 1**.

Utilizando o prompt (ainda na pasta “...\wxWidgets\build\msw”), compila-se o wxWidgets utilizando o seguinte comando: **mingw32-make -j8 -f Makefile.gcc SHARED=0 UNICODE=1 MONOLITHIC=1 BUILD=release VENDOR=x64 CXXFLAGS="-fno-keep-inline-dllexport -std=c++17"**.

Esse processo pode demorar até algumas horas.

Atividade 2: Instalando e configurando o CodeLite.

Primeiramente, realiza-se o download e instala-se a última versão do CodeLite no seguinte link: <https://downloads.codelite.org/>

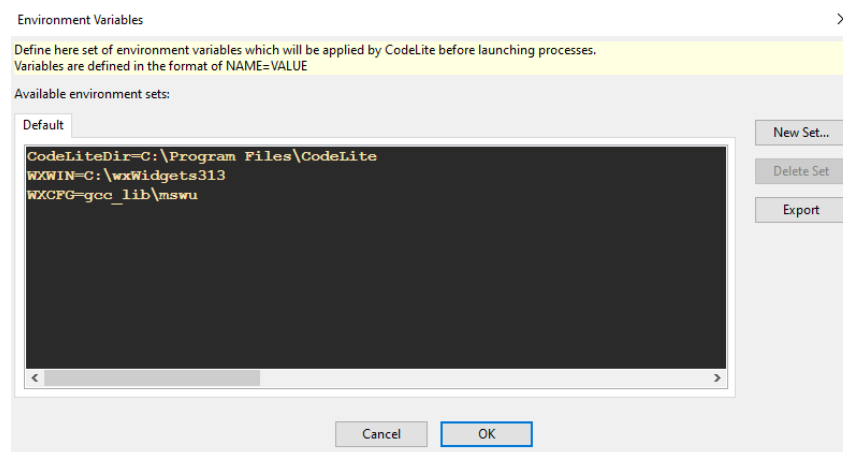
De acordo com a figura 01, que também está nos apêndices, temos a descrição de que, ao abrir o CodeLite, o programa automaticamente irá encontrar o compilador (tdm-gcc). Por último, no menu “Settings” e depois em “Environment Variables”, abrem-se as variáveis de ambiente do CodeLite e configura-se da seguinte maneira:

CodeLiteDir=C:\Program Files\CodeLite

WXWIN=C:\wxWidgets-3.1.4

WXCFG=gcc_lib\mswu

Figura 01: Janela de compilação do código.



Fonte: Próprio autor (2023)

Com essas etapas, é possível criar um projeto utilizando a GUI (Interface Gráfica de Usuário) do wxWidgets e compilar.

3.1 Rapidxml: uma biblioteca eficiente para parsing de arquivos xml

Segundo Silva (2019) explica em seu contexto que RapidXml é uma biblioteca de parsing XML de código aberto escrita em C++, simples de usar e fácil de entender. Ela foi projetada para ser rápida e fácil de usar, e é uma boa escolha para aplicativos que precisam lidar com arquivos XML de tamanho médio a grande, o que significa que ele lê e analisa o arquivo XML diretamente na memória, sem precisar carregá-lo inteiro em uma estrutura de dados antes de analisá-lo. Isso o torna muito eficiente em termos de uso de memória e tempo de processamento. Ele também oferece uma boa quantidade de recursos, incluindo suporte para atributos, elementos de texto,

comentários e espaços em branco, suporta a validação DTD, mas não suporta a validação XML Schema.

3.2 Wxpdfdocument: biblioteca C++ para geração de arquivos pdf com wxwidgets

De acordo com Telle (2021) o wxPdfDocument é uma classe C++ que permite que aplicativos wxWidgets gerem arquivos PDF. O código é uma porta de FPDF - uma classe PHP gratuita para gerar arquivos PDF - para C++ usando a biblioteca wxWidgets; não faz uso de nenhuma biblioteca como PDFlib ou ClibPDF, que exige uma taxa pelo menos para uso comercial.

O wxPdfDocument oferece todas as vantagens do FPDF. Vários scripts PHP complementares encontrados no site da FPDF estão incorporados ao wxPdfDocument. As principais características são (TELLE, 2021):

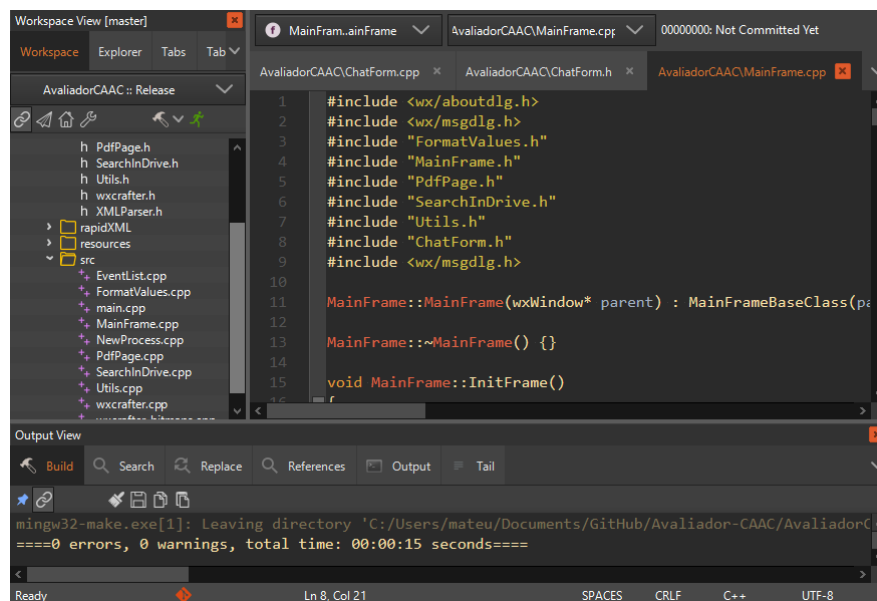
- Escolha da unidade de medida, formato da página e margens
- Gerenciamento de cabeçalho e rodapé da página
- Quebra de página automática
- Quebra de linha automática e justificação de texto
- Suporte de imagem (GIF, JPEG, PNG e WMF)
- Cores (tons de cinza, RGB, CMYK, cores exatas)
- Links (internos e externos)
- 14 fontes padrão da Adobe
- Fontes TrueType e Type1 (com ou sem incorporação) e suporte para codificação
- Fontes TrueType Unicode e Type0 (para chinês, japonês e coreano) na compilação Unicode
- Suporte a fontes OpenType Unicode na compilação Unicode
- Compressão de página
- Primitivas gráficas para a criação de desenhos simples
- Definição de áreas de recorte
- Marcadores para estruturar o documento
- Rotação
- Protegendo o documento por senhas e/ou permissões de acesso
- Anotações de texto

- Formulários PDF (tipos de campo suportados: texto, caixa de combinação, caixa de seleção, botão de opção, botão de pressão)
- JavaScript
- Gradientes de preenchimento
- Modelos
- Camadas (grupos de conteúdo opcionais)
- Padrões de bitmap simples como cores de desenho e preenchimento

4 DESENVOLVIMENTO DO SOFTWARE

Abaixo encontra-se a Figura 02, presente nos apêndices I e II, que representa a estrutura de programação para o desenvolvimento da interface gráfica utilizando o CodeLite com o wxWidgets.

Figura 02: Interface de desenvolvimento do CodeLite



Fonte: Próprio autor (2023)

A interface do CodeLite é dividida em três partes principais: árvore de projetos, área de edição e área de mensagens. Cada parte tem um propósito específico e juntas formam uma interface de usuário intuitiva e eficiente para o desenvolvimento de software.

Segundo Oliveira (2021), a árvore de projetos é usada para navegação e gerenciamento de projetos. É aqui que o usuário pode visualizar e adicionar arquivos e pastas a um projeto, bem como acessar arquivos e pastas recentes. Ela também contém outros recursos úteis, como a pesquisa de arquivos, a visualização de erros e avisos e a navegação pelo código. Na área de projetos, é possível observar alguns arquivos específicos, ou seja, arquivos (.h) e arquivos (.cpp). Os arquivos (.h) são conhecidos como cabeçalho ou header files, e são usados para declarar funções, classes, variáveis e outras informações que serão usadas em outros arquivos do projeto. O arquivo (.h) geralmente inclui a definição de classes, protótipos de funções, constantes, variáveis globais, entre outros. Esse arquivo é importante porque permite

a separação entre a implementação (código-fonte) e a declaração (cabeçalho), tornando mais fácil a manutenção e reutilização do código.

De acordo com Oliveira (2021), os arquivos (.cpp), por outro lado, são o arquivo de implementação e é onde o código real é escrito. Ele inclui as definições das funções e classes declaradas no arquivo (.h). É neste arquivo que o programador escreve as instruções que serão executadas pelo computador.

Já os arquivos wxcrafter (.h) e wxcrafter (.cpp) são gerados pelo CodeLite e são usados para a criação de interfaces gráficas de usuário utilizando o framework wxWidgets. O arquivo wxcrafter (.h) é um cabeçalho que contém a declaração da classe da interface gráfica e as associações entre os controles da interface e os eventos gerados por eles. O arquivo wxcrafter (.cpp) é o arquivo de implementação da classe da interface gráfica, onde o programador escreve o código para lidar com eventos de interface como cliques em botões ou atualizações de campos de texto (OLIVEIRA, 2021).

E conforme a instalação realizada, a parte direita é a área de edição de código e é usada para escrever e editar o código-fonte. É aqui onde o usuário pode aproveitar recursos como destaque de sintaxe, verificação de sintaxe em tempo real, autocompletar de código e gerenciamento de breakpoints. A área de edição de código também permite que o usuário organize a tela em várias abas para trabalhar em vários arquivos ao mesmo tempo.

A barra de status ou área de mensagens é usada para exibir informações importantes sobre o estado atual do CodeLite. É aqui onde o usuário pode ver informações sobre a linha e coluna atual, o estado do projeto (compilação e execução), bem como mensagens de erro e aviso. A barra de status também permite que o usuário execute tarefas como compilar, executar e depurar o código.

4.1 WxCrafter: Ferramenta visual para desenvolvimento de interfaces gráficas

Segundo Ventura e Suquizaqui (2020), wxCrafter é uma ferramenta de desenvolvimento de aplicativos gráficos de interface com o usuário baseada na biblioteca wxWidgets. Ele fornece uma interface visual fácil de usar para a criação de interfaces gráficas para aplicativos, sem a necessidade de escrever código manualmente. É possível criar layouts personalizados, adicionar controles, como botões, caixas de seleção, campos de texto, entre outros, e criar eventos para

- **Painel de edição:** É onde você pode visualizar e editar a aparência da interface gráfica de usuário. Esta área mostra uma representação gráfica do aplicativo, permitindo que você adicione, mova ou exclua elementos da interface, como botões, rótulos e caixas de texto.

- **Painel de propriedades:** Parte da interface que exibe as propriedades dos componentes selecionados na área de edição. Aqui você pode personalizar as propriedades dos componentes, como cor, tamanho, texto, entre outros.

4.2 Software de Avaliação de Atividades Complementares (CAAC): Recursos e Funcionalidades

A imagem a seguir (Figura 04, que também se encontra nos apêndices) representa a tela principal do software “Avaliador CAAC”. Nele, estão presentes todos os itens necessários para a criação de um novo processo ou o download e upload de um processo já existente. Também é possível editar dados e gerar relatórios. Esses recursos são chamados por meio dos menus “Arquivo” e “Relatório”.

No menu “Arquivo”, encontram-se as seguintes opções:

- **Novo Processo:** Cria uma tarefa (processo).
- **Salvar Processo:** Armazena um processo criado.
- **Abrir Processo:** Responsável por abrir um processo já existente em um dispositivo.
- **Abrir Processo do Drive:** Faz o download de um processo do Google Drive.

O menu “Relatório” inclui dois itens:

- **Editar Dados do Processo:** É quando deseja-se alterar alguma informação do número de protocolo, nome do aluno ou a resolução de um processo existente.
- **Gerar Relatório:** Gera o relatório de validação das atividades complementares.

Figura 04: Avaliador CAAC

Item	Escolha o item...
Evento	Nome do evento...
Descrição do evento	
Turno	1
Evento do IFG	True
Instituição	Nome da instituição...
Data do evento	-
Carga horária (h)	0
Invaliar CH	False

Total solicitado = 0,0 h
 Total validado = 0,0 h
 Carga horária limitada pela resolução
 Restrições de carga horária ignoradas
 Carga horária invalidada
 Item não impresso no relatório

Fonte: Próprio autor (2023)

Na figura 05 a seguir (também se encontra no apêndice I), a janela “Novo Processo”, é chamada por meio do menu “Arquivo”. Nela é possível inserir os dados de processo como o número de protocolo, nome do estudante, número de documentos e a resolução referente, ou seja, se é aluno de curso superior ou técnico.

Figura 05: Novo processo

Número do protocolo
 23376.001838/2020-70

Nome do estudante
 Mateus Lucas de Almeida

Número de documentos
 2

Resolução
 Resolução n°16 de dezembro de 2011 (Cursos Superiores)

OK Cancelar

Fonte: Próprio autor (2023)

De acordo com a figura abaixo (Figura 06, que também se encontra nos apêndices), após a confirmação dos dados, a janela principal é habilitada para inserir

as informações referentes às atividades complementares que o aluno participou, por exemplo, o nome do evento, descrição, instituição e carga horária. Os botões “Inserir Item” e “Remover Item” permitem adicionar ou remover mais de um evento. Preenchido todos os dados, é possível salvar o processo no menu “Arquivo”. O processo é salvo tanto no dispositivo desktop como no Google Drive.

Figura 06: Avaliador CAAC

Fonte: Próprio autor (2023)

Se o avaliador precisar revisar ou editar algum processo específico, é possível baixá-lo do Drive como mostra a imagem a seguir (Figura 07, que também se encontra nos apêndices). A Tela “Buscar no Drive” é chamada por meio do menu “Arquivo”. Nela é sugerida uma das informações de busca, ou seja, número de protocolo ou o nome do estudante. Se o processo procurado existir no Drive, o número de protocolo e o nome do estudante são mostrados na janela inferior ao botão “Buscar”. Não inserindo nenhuma informação, a janela irá mostrar todos os processos que foram salvos no Drive. Após selecionar um número de protocolo e clicar no botão “OK”, os dados referentes ao processo são mostrados e habilitados para alteração.

Figura 07: Buscar no Drive

Protocolo	Nome
23376.001838/2020-70	Mateus Lucas de Almeida

Fonte: Próprio autor (2023)

Durante a avaliação de processos de horas complementares, é comum que o avaliador cometa erros de informação, como inserir uma data incorreta, um item equivocado ou deixar de registrar algum documento. Portanto, é necessário implementar no software uma forma de interatividade entre os avaliadores, permitindo que outro avaliador que tenha acesso ao mesmo documento possa apontar todos os erros e deixá-los como observação, para que o primeiro avaliador que gerou o processo possa corrigi-los.

A janela "Comentário do Processo" permite que o usuário insira o nome do avaliador e o comentário, ou seja, as observações sobre o processo. Depois de clicar no botão "Enviar", a mensagem do avaliador é salva na tela principal do programa.

Figura 08: Comentário do Processo

Nome do Avaliador:
Thales Oliveira

Comentário:
Por favor! Alterar o Item 2 para: Atividade Prática em Campo.

Fonte: Próprio autor (2023)

A mensagem inserida pelo avaliador pode ser visualizada na Figura 09, a qual também se encontra nos apêndices. Após o processo ser salvo, outros avaliadores terão acesso ao documento e poderão inserir suas observações.

Figura 09: Avaliador CAAC

Avaliador CAAC

Arquivo Relatório Ajuda

Processo: 23376.001838/2020-70

Aluno: Mateus Lucas de Almeida

Item I - Visitas técnicas

Evento Nome do evento...

Descrição do evento

Turno 1

Evento do IFG ☒

Instituição Nome da instituição...

Data do evento -

Avaliador: Thales Oliveira

Comentário: Por favor! Alterar o Item 2 para: Atividade Prática em Campo.

Item **Evento** **T/**

1 III - Participação e Festival de Artes de Goiás

2 I - Visitas técnicas Web Semântica

Total solicitado = 44,00 h

Total validado = 44,00 h

Carga horária limitada pela resolução

Restrições de carga horária ignoradas

Carga horária invalidada

Item não impresso no relatório

Fonte: Próprio autor (2023)

Finalizado o processo, no menu “Relatório”, é possível gerar o relatório de validação de atividades complementares, conforme a Figura 10, que também se encontra no Apêndice I.

Figura 10: Relatório de validação

INSTITUTO FEDERAL
Goiás
Câmpus Itumbiara

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS
Campus Itumbiara

PROCESSO: Nº 23376.001838/2020-70

ASSUNTO: AVALIAÇÃO DE ATIVIDADES COMPLEMENTARES

A Comissão de Avaliação das Atividades Complementares (CAAC) dos Cursos de Graduação do Instituto Federal de Educação, Ciência e Tecnologia de Goiás, de acordo com o Regulamento das Atividades Complementares dos Cursos de Graduação (Resolução nº 16 de 26/12/2011) e Orientações para as Atividades Complementares dos Cursos de Graduação, após analisar os certificados apresentados no presente processo, valida o cumprimento de 44 horas para o aluno Mateus Lucas de Almeida.

A seguir a relação dos certificados recebidos e avaliados:

EVENTO	INSTITUIÇÃO	PERÍODO	Carga Horária Apresentada	Carga Horária Validada
Festival de Artes de Goiás - Acompanhamento dos estudantes às atividades de artes	IFG	-	40	40
Web Semântica - Acompanhamento dos estudantes às atividades de artes	ULBRA	-	4	4
TOTAL			44	44

Observações:

- Os certificados analisados estão apresentados na tabela acima de acordo com a ordem de apresentação no processo.
- A carga horária validada está de acordo com a Resolução nº 16 do Regulamento das Atividades Complementares.

Inumbiara/GO, 12 de fevereiro de 2023.

Carlos Antunes de Queiroz Junior Fernanda Hein da Costa Thales Lima Oliveira

Fonte: Próprio autor (2023)

4.3 C++: Uma linguagem de programação de alto desempenho e flexibilidade

De acordo com Stroustrup (2014), a linguagem C++ é uma linguagem de programação de alto desempenho e de propósito geral, desenvolvida por Bjarne Stroustrup, da Bell Labs, nos anos 80. Ela é uma extensão do C, com a adição de recursos orientados a objetos, como classes, herança e polimorfismo,

Segundo Stroustrup (2014), C++ é utilizado amplamente em aplicações como jogos, sistemas operacionais, drivers, aplicativos de desktop e muito mais. Sua capacidade de controlar baixo nível de hardware e sua capacidade de ser otimizado para desempenho fazem dele uma escolha popular para desenvolvedores de sistemas.

Segundo Dias (2022) já o C++11, lançado em 2011, introduziu recursos adicionais, como lambda expressions, move semantics e suporte para concorrência. O C++14 e o C++17 foram lançados posteriormente, adicionando mais recursos e melhorias para ajudar a tornar o código mais conciso e seguro. C++ é uma linguagem de programação de alto nível e de propósito geral, que permite ao desenvolvedor controlar baixo nível de hardware e oferece uma grande flexibilidade para a criação de diferentes tipos de aplicações. É uma linguagem compilada, o que significa que o código fonte é convertido em código de máquina antes de ser executado. Isso geralmente resulta em desempenho mais rápido e menos dependência de bibliotecas externas.

Ainda o autor Dias (2022) cita que uma das principais características de C++ é sua capacidade de programação orientada a objetos (POO). Isso permite que o desenvolvedor crie classes e objetos, herde atributos e métodos de classes-base e utilize polimorfismo para criar código mais reutilizável e flexível. Além disso, C++ oferece suporte para templates genéricos, permitindo escrever código que pode ser usado com diferentes tipos de dados sem a necessidade de criar várias versões da mesma função. Outra característica importante do C++ é sua capacidade de trabalhar com ponteiros e alocação de memória manual. Isso permite ao desenvolvedor ter um maior controle sobre como a memória é gerenciada e alocada, o que pode ser útil em aplicações com requisitos de desempenho rigorosos.

De acordo com Silva (2017), C++ também oferece uma grande variedade de bibliotecas prontas para uso, incluindo a STL (Biblioteca de Modelos Padrão, do inglês, *Standard Template Library*), que fornece uma série de classes e funções para

tarefas comuns, como gerenciamento de containers, algoritmos e streams de entrada e saída. Desta forma, entende-se que C++ é uma linguagem poderosa e flexível, que oferece uma grande variedade de recursos e ferramentas para desenvolvimento de aplicações. Sua capacidade de programação orientada a objetos, suporte para templates genéricos e capacidade de trabalhar com ponteiros e alocação de memória manual a tornam uma escolha popular para desenvolvedores de sistemas e jogos.

Algumas vantagens de se utilizar a linguagem C++ incluem (SILVA, 2017):

- Desempenho: C++ é uma linguagem compilada, o que geralmente resulta em desempenho mais rápido do que linguagens interpretadas. Além disso, a capacidade de trabalhar com ponteiros e alocação de memória manual permite ao desenvolvedor ter um maior controle sobre como a memória é gerenciada e alocada, o que pode ser útil em aplicações com requisitos de desempenho rigorosos.

- Flexibilidade: C++ é uma linguagem de propósito geral, o que significa que pode ser usada para desenvolver uma ampla variedade de aplicações, como jogos, sistemas operacionais, drivers, aplicativos de desktop e muito mais. Sua capacidade de programação orientada a objetos, suporte para templates genéricos e capacidade de trabalhar com ponteiros e alocação de memória manual tornam-na uma escolha popular para desenvolvedores de sistemas e jogos.

- Grande variedade de bibliotecas e ferramentas: C++ oferece uma grande variedade de bibliotecas prontas para uso, incluindo a STL (Standard Template Library), que fornece uma série de classes e funções para tarefas comuns, como gerenciamento de containers, algoritmos e streams de entrada e saída.

Alguns dos motivos que levam ao uso de C++ são (REIS, 2020):

- Baixo nível de acesso: C++ oferece acesso abaixo nível de hardware, o que permite ao desenvolvedor escrever código eficiente em termos de desempenho e otimizar o uso de recursos.

- Portabilidade: C++ é uma linguagem compilada, o que significa que o código pode ser compilado em diferentes sistemas operacionais e plataformas, o que permite ao desenvolvedor escrever código que pode ser executado em diferentes sistemas sem a necessidade de modificações significativas.

- Grande comunidade: C++ tem uma grande comunidade de desenvolvedores, o que significa que existem muitos recursos, bibliotecas e ferramentas disponíveis para ajudar os desenvolvedores a resolver problemas e acelerar o desenvolvimento de aplicações.

Suporte para multi-threading: O C++11 introduziu o recurso de suporte para programação concorrente, permitindo que os desenvolvedores escrevam código que pode ser executado em vários threads simultaneamente. Isso é especialmente útil em aplicações que precisam aproveitar o poder de processamento de múltiplos núcleos. Inovação: C++ é uma linguagem em constante evolução, com as versões C++14 e C++17 adicionando novos recursos e melhorias, incluindo suporte para multi-threading, melhorias na segurança do código, entre outros.

Durante o desenvolvimento do software, são utilizados conceitos de programação orientada a objetos, como a definição de classes e objetos, herança e polimorfismo. Também são empregados métodos e funções responsáveis por toda a lógica do programa, além de outros recursos da linguagem C++, como ponteiros e referências, de acordo com o código principal de execução **"MainFrame.cpp"**.

4.4 PYTHON: Uma linguagem versátil e de fácil aprendizado

Conforme os autores Campos e Oliveira (2016), Python é uma linguagem de programação de alto nível, de código aberto e de propósito geral, criada por Guido Van Rossum em 1989, que tem uma sintaxe simples e intuitiva, tornando-a fácil de aprender para iniciantes. Além disso, a comunidade Python é muito ativa e existem muitos recursos e bibliotecas disponíveis para ajudar os desenvolvedores a construir aplicações em vários campos. A seguir, segue a explicação de cada um:

- Ciência de dados: Python é uma das principais linguagens utilizadas para análise de dados, devido à sua capacidade de lidar com grandes conjuntos de dados e à existência de bibliotecas como Pandas, Numpy, Scipy.

Inteligência artificial: Com bibliotecas como TensorFlow, Keras, PyTorch, Python é uma das principais linguagens utilizadas para desenvolvimento de modelos de aprendizado de máquina.

- Automatização de testes: Python é uma das principais linguagens utilizadas para automatização de testes, devido à sua capacidade de se integrar com várias ferramentas e sistemas.

Desenvolvimento web: Com bibliotecas como Django, Flask e Pyramid, Python é uma das principais linguagens utilizadas para desenvolvimento de aplicações web.

Segundo Campos e Oliveira (2016), uma das principais vantagens de Python é sua grande variedade de bibliotecas e frameworks disponíveis para uso, que

possibilitam a realização de tarefas específicas de forma rápida e fácil. Algumas das principais bibliotecas de Python incluem:

- NumPy, para trabalhar com arrays e matrizes multidimensionais
- Pandas, para análise e manipulação de dados
- Matplotlib, para criação de gráficos e visualizações de dados
- TensorFlow, Keras, PyTorch para inteligência artificial
- Django, Flask, Pyramid para desenvolvimento web
- Selenium e Pytest para automação de testes

Outra vantagem de Python é sua comunidade ativa e colaborativa, que fornece suporte e recursos para desenvolvedores iniciantes e experientes. Isso inclui uma ampla variedade de documentação, tutoriais, fóruns e comunidades online, além de conferências e eventos para a comunidade Python.

Segundo Divino (2021), Python é uma linguagem amplamente utilizada em muitas indústrias, incluindo tecnologia, finanças, ciência de dados, inteligência artificial, automação de testes, desenvolvimento web e muito mais. Isso significa que existem muitas oportunidades de emprego para desenvolvedores Python e uma grande demanda por habilidades relacionadas a essa linguagem.

De acordo com Kriger (2022), uma das principais vantagens do Python é sua comunidade ativa e a grande quantidade de bibliotecas e frameworks disponíveis. Isso permite aos desenvolvedores aproveitar o trabalho de outros desenvolvedores e economizar tempo e esforço no desenvolvimento de seus próprios aplicativos. Além disso, Python é fácil de aprender, especialmente para aqueles que já têm alguma experiência com programação, devido à sua sintaxe simples e legível. Ele também é uma boa escolha para a iniciação em programação para aqueles que estão começando.

Na programação, a linguagem Python, é utilizada para permitir a comunicação entre o software com o GoogleDrive. É utilizado a biblioteca Google Drive API escrita em Python, que permite realizar ações como autenticação, listagem de arquivos, upload, download, remoção e busca de arquivos.

O código começa com a importação de módulos necessários para a funcionalidade da biblioteca, incluindo o "googleapiclient.discovery", "google_auth_oauthlib.flow" e "googleapiclient.http".

A função "auth" é usada para autenticar o acesso à API do Google Drive, verificando se há credenciais válidas armazenadas localmente no arquivo "token.pickle" ou solicitando a autenticação do usuário através de "credentials.json".

As funções "listFiles", "fileUpload", "fileRemove", "fileDownload" e "searchFiles" permitem, respectivamente, listar arquivos, fazer upload de arquivos, remover arquivos, baixar arquivos e buscar arquivos no Google Drive. A função "listFilesInFolder" lista os arquivos em uma pasta específica no Google Drive. A função "searchFilesInFolder" permite buscar arquivos específicos em uma pasta.

Além disso, o código também define o escopo de acesso ao Google Drive com a constante "SCOPES" e o ID da pasta "CAACFolder".

5 CONCLUSÃO

O presente trabalho tem como objetivo o desenvolvimento de uma ferramenta computacional livre e de código aberto que permita a avaliação das horas complementares pela CAAC de forma interativa, eficiente e automatizada.

Com relação ao desenvolvimento da ferramenta, pode-se afirmar que foi possível inserir as descrições dos documentos de horas complementares de forma automatizada, aplicar restrições e contabilizar a pontuação do estudante automaticamente de acordo com a regulamentação vigente, consultar os bancos de dados de processos já realizados para evitar erros e a possibilidade de interatividade entre os avaliadores.

O objetivo final do produto é servir como uma ferramenta permanente para o âmbito do IFG, visando aumentar a eficiência da administração e reduzir erros nos processos de avaliação da CAAC. O desenvolvimento de um software interativo para automação da avaliação de processos de horas complementares trouxe diversos benefícios, como agilidade, uma vez que a automação do processo permite acelerar a avaliação das horas complementares, eliminando a necessidade de avaliação manual, além de eficiência, em que o software ajuda a garantir uma avaliação mais precisa e consistente das horas complementares, reduzindo erros e inconsistências.

Um ponto que também merece destaque é a questão da redução de custos, em que a automação do processo pode ajudar a reduzir os custos associados à avaliação manual das horas complementares, como tempo e recursos humanos.

Para trabalhos futuros, seria interessante explorar mais a fundo o desempenho e eficiência de outras bibliotecas em diferentes plataformas e cenários, além de comparar a usabilidade e facilidade de aprendizado para iniciantes em programação. Outra possibilidade seria analisar o impacto das licenças em projetos comerciais e em colaboração com outras bibliotecas e frameworks.

Por exemplo, uma análise comparativa entre a biblioteca wxWidgets e o Qt revela que ambas são bibliotecas de interface gráfica de usuário (GUI) que oferecem suporte a várias plataformas, incluindo Windows, Linux e Mac OS X. No entanto, há algumas diferenças importantes entre elas. Enquanto o wxWidgets é escrito em C++, o Qt é escrito em C++ e suporta outras linguagens de programação, como Python e Java. Além disso, o Qt também oferece suporte a outras plataformas, como Android e iOS. Outra diferença significativa é a licença - o wxWidgets é licenciado sob a Licença

Pública Geral Menor (LGPL), enquanto o Qt é licenciado sob a licença GPL e a licença comercial. Quando se trata de desempenho, o Qt é geralmente considerado mais rápido em alguns cenários. Ambas as bibliotecas oferecem documentação detalhada e suporte de comunidade. Em relação aos componentes de interface gráfica, ambas as bibliotecas oferecem um conjunto completo, mas o Qt tem uma quantidade maior de componentes e um designer gráfico de interface do usuário integrado, tornando mais fácil projetar interfaces visualmente.

REFERÊNCIAS

ADRIGM. Codelite. 2013. Disponível em: <https://www.genbeta.com/desarrollo/codelite-un-ide-multiplataforma-para-c-c> Acesso em: 15 de maio 2023.

ANGELIS, José de.; SILVA, Osmar. Como baixar, compilar a biblioteca e criar um projeto C++ wxWidgets usando o Visual Studio 2017. 2023. Disponível em: <http://arquivodecodigos.com.br/dicas/3696-wxwidgets-como-baixar-compilar-a-biblioteca-e-criar-um-projeto-c-wxwidgets-usando-o-visual-studio-2017.html> Acesso em: 12 de abr. 2023.

AUDACITY. Audacity. 2022. Disponível em: <https://www.audacityteam.org/download/> Acesso em: 15 de jan. 2023.

BÍBLIA. Português. Bíblia sagrada. Tradução de João Ferreira de Almeida. Revista e atualizada no Brasil. 2. ed. São Paulo: Sociedade Bíblica do Brasil, 1993. Salmo 111:10.

CAMPOS, Victor H. G. de; OLIVEIRA, Rosevaldo de. Computação científica usando Python. 1ª Edição. Rondonópolis. 2016.

CHICONI, Natali. Bibliotecas para interfaces gráficas (GUI). 2020. Disponível em: <https://br.ccm.net/faq/10999-bibliotecas-para-interfaces-graficas-gui> Acesso em: 15 de jan. 2023.

CODEBLOKS. Codebloks. 2022. Disponível em: https://wiki.codeblocks.org/index.php/Main_Page Acesso em: 15 de maio 2023.

DELGADO, Armando L. N. Linguagem C++: notas de aula. 2018. Disponível em: <https://www.inf.ufpr.br/ci208/NotasAula.pdf> Acesso em: 10 de jan. 2023.

DIAS, Vinicius. C++: Vale a pena aprender atualmente? Por que e onde usar. 2022. Disponível em: <https://www.alura.com.br/artigos/formacao-linguagem-c-plus-plus> Acesso em: 20 de março 2023.

DIVINO, Bruno. Python – uma introdução à linguagem. 2021. Disponível em: <https://www.alura.com.br/artigos/python-uma-introducao-a-linguagem> Acesso em: 26 de março 2023.

FILEZILLA. Filezilla. 2023. Disponível em: <https://filezilla-project.org/> Acesso em: 15 de maio 2023.

FLTK. FLTK. 2023. Disponível em: <https://www.fltk.org/> Acesso em: 15 de maio 2023.

HEART, Brain. Aplicações desktop – o que são e como funcionam? 2022. Disponível em: <https://wktechnology.com.br/aplicacoes-desktop-o-que-sao-e-como-funcionam/> Acesso em: 10 de jan. 2023.

QT GROUP. Qt Group. 2020. Disponível em: <https://www.qt.io/,%20Dom%C3%ADnio%20p%C3%BAblico,%20https://commons.wikimedia.org/w/index.php?curid=50058603> Acesso em: 15 de maio 2023.

KRIGER, Daniel. O que é Python, para que serve e por que aprender? 2022. Disponível em: <https://kenzie.com.br/o-que-e-python/> Acesso em: 10 de jan. 2023.

MÁXIMA. Um sistema de álgebra computacional. 2015. Disponível em: <https://maxima.sourceforge.io/pt/index.html> Acesso em: 15 de maio 2023.

MICROSOFT. MFC. 2023. Disponível em: <https://learn.microsoft.com/en-us/cpp/media/index/logo-mfc.svg?view=msvc-170&viewFallbackFrom=msvc-170,%20Public%20Domain,%20https%3A%2F%2Fcommons.wikimedia.org%2Fw%2Findex.php%3Fcurid%3D125101497> Acesso em: 15 de maio 2023.

NILSSON, Andreas. File: GTK logo.svg. 2023. disponível em: <https://commons.wikimedia.org/w/index.php?curid=3605698> Acesso em: 15 de maio 2023.

NOTEPAD. What is notepad++. 2023. Disponível em: <https://notepad-plus-plus.org/> Acesso em: 15 de maio 2023.

OLIVEIRA, Marcos. Os 32 melhores IDEs/Editores de texto para C++. 2021. Disponível em: <https://terminalroot.com.br/2021/12/os-32-melhores-ides-editores-de-texto-para-cpp.html> Acesso em: 10 de jan. 2023.

PYTHON. Python. 2023. Disponível em: <https://www.python.org/community/logos/,%20GPL,%20https://commons.wikimedia.org/w/index.php?curid=34991637> Acesso em: 15 de maio 2023.

SAUTER, Esquia; AZEVEDO, Fabio S. de; KONZEN, Pedro H. de A. Computação científica em linguagem C. 2020. Disponível em: <https://www.ufrgs.br/reatmat/ComputacaoCientifica/livro/livro.pdf> Acesso em: 12 de abr. 2023.

SILVA, Maniero. Bibliotecas standards em C/C ++. 2019. Disponível em: <https://pt.stackoverflow.com/questions/321816/bibliotecas-standards-em-c-c> Acesso em: 15 de jan. 2023.

SILVA, Jailson C. Ensino de programação para alunos do ensino básico: um levantamento das pesquisas realizadas no Brasil. 2017. Disponível em: <https://repositorio.ufpb.br/jspui/bitstream/123456789/3328/1/JCS14062017.pdf> Acesso em: 20 de março 2023.

STROUSTRUP, Bjarne. A linguagem de programação C++. Quarta edição. 2013.

REIS, Fábio. Introdução à linguagem de programação C++. 2020. Disponível em: <http://www.bosontreinamentos.com.br/programacao-em-c/introducao-a-linguagem-de-programacao-c/> Acesso em: 25 de março 2023.

SILVA, Danilo R. WxWidgets com café. 2011. Disponível em: <https://widgetscomcafe.blogspot.com/2011/04/primeiro-programa-com-wxwidgets.html> Acesso em: 15 de jan. 2023.

TELLE. Ulrich. WxPdfDocument. 2021. Disponível em: <https://utelle.github.io/wxpdfdoc/> Acesso em: 15 de jan. 2023.

WIKIPÉDIA. Ficheiro: ISSO C++ Logo.svg. 2023. Disponível em: BÍBLIA. Português. Bíblia sagrada. Tradução de João Ferreira de Almeida. Revista e atualizada no Brasil. 2. ed. São Paulo: Sociedade Bíblica do Brasil, 1993. Salmo 111:10. Acesso em: 15 de maio 2023.

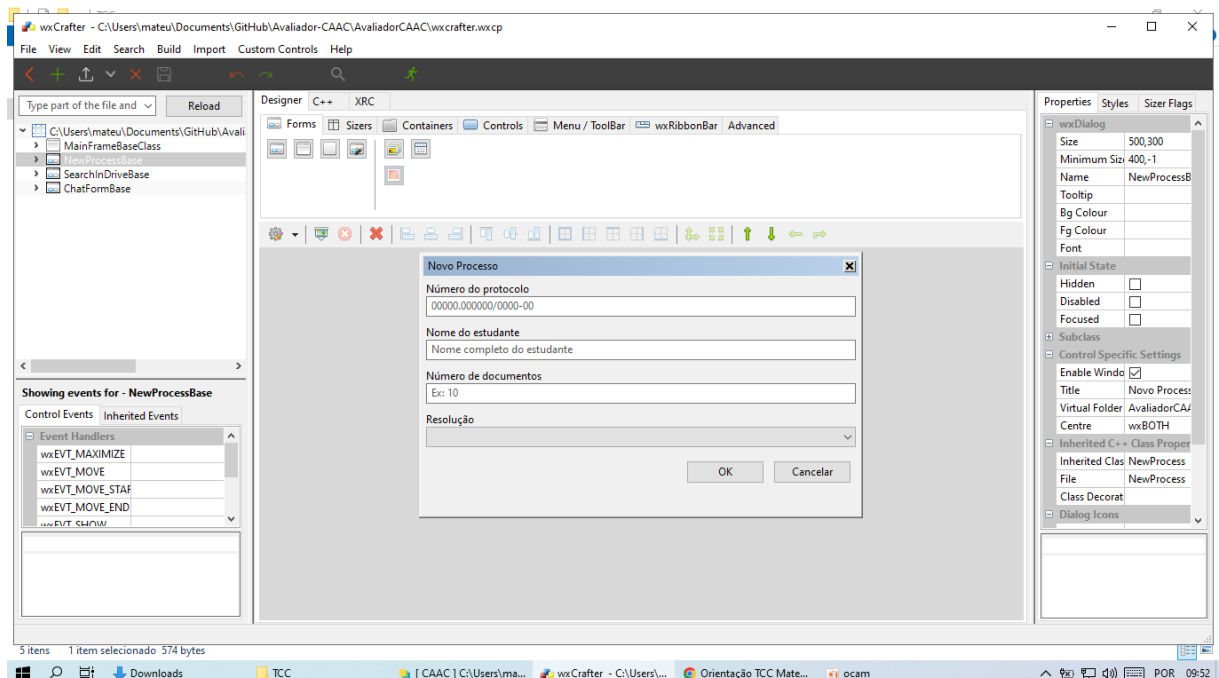
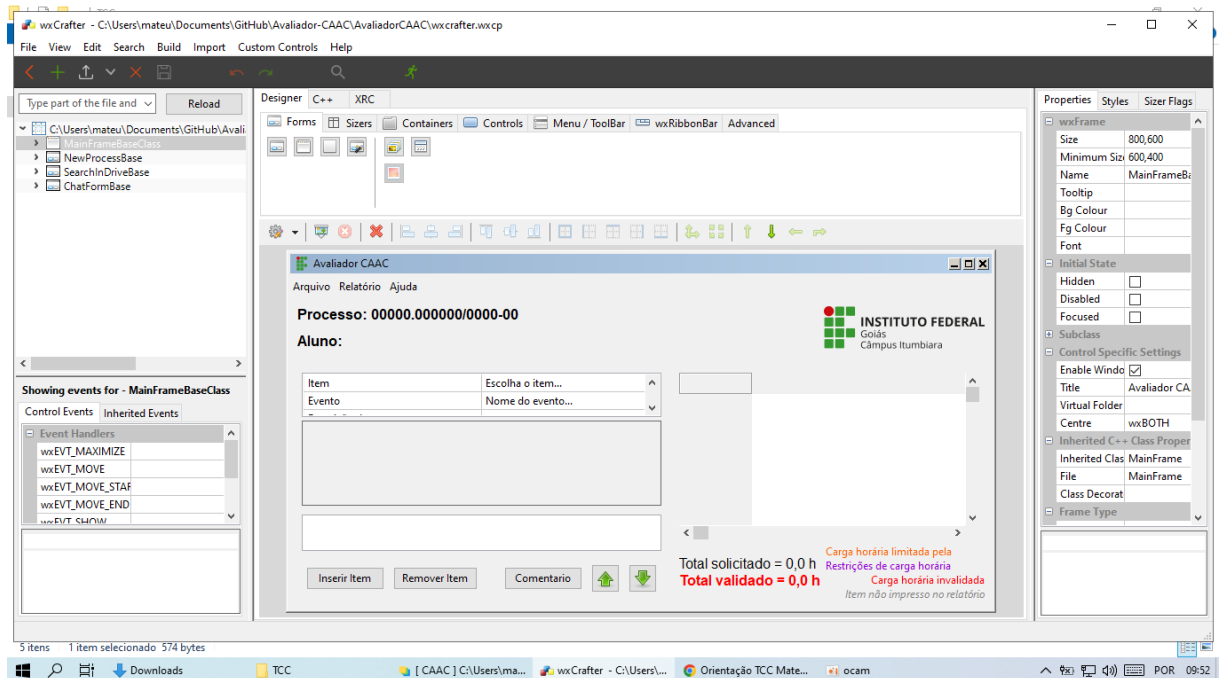
WIKIPEDIA. Audacity. 2023. Disponível em: <https://pt.wikipedia.org/wiki/Audacity> Acesso em: 15 de maio 2023.

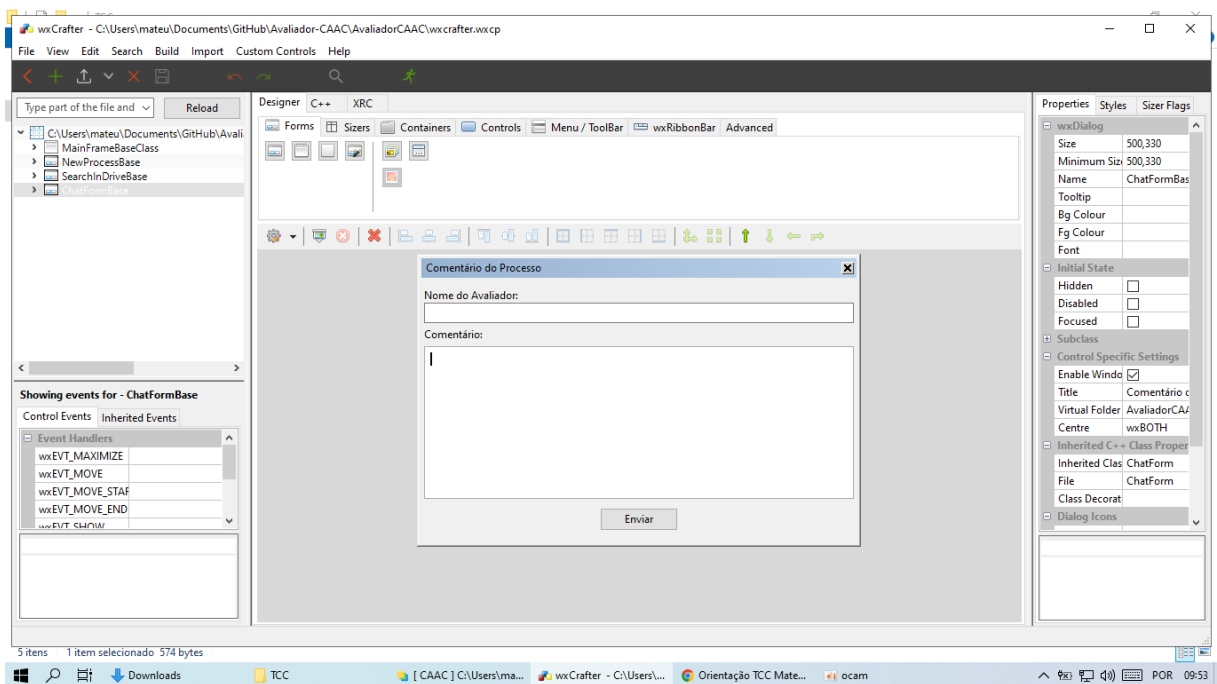
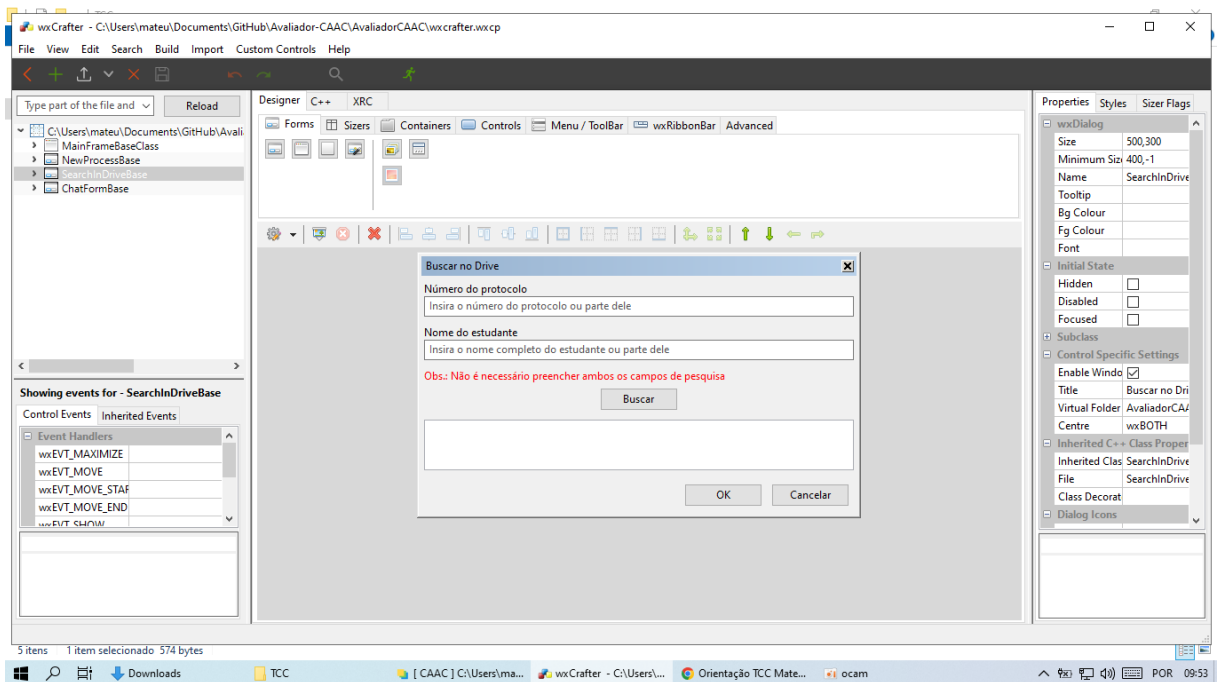
ZEITLIN, Vadim. Wxwidget. 2023. Disponível em: <https://www.wxwidgets.org/about/> Acesso em: 10 de jan. 2023.

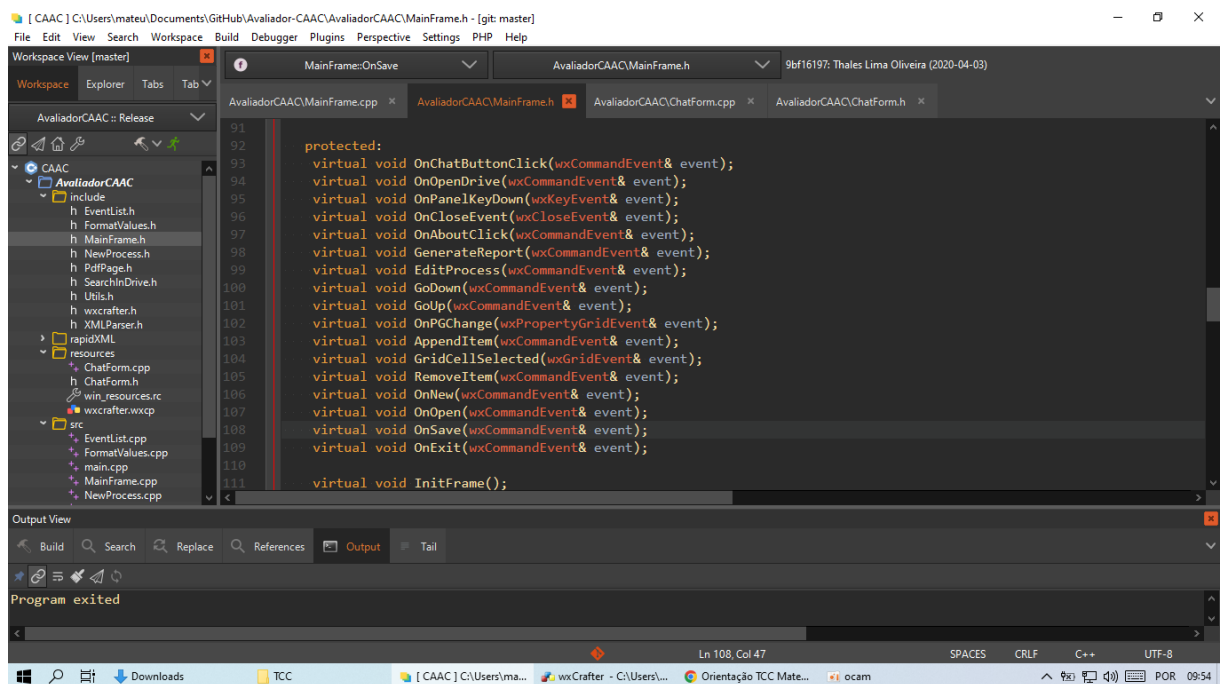
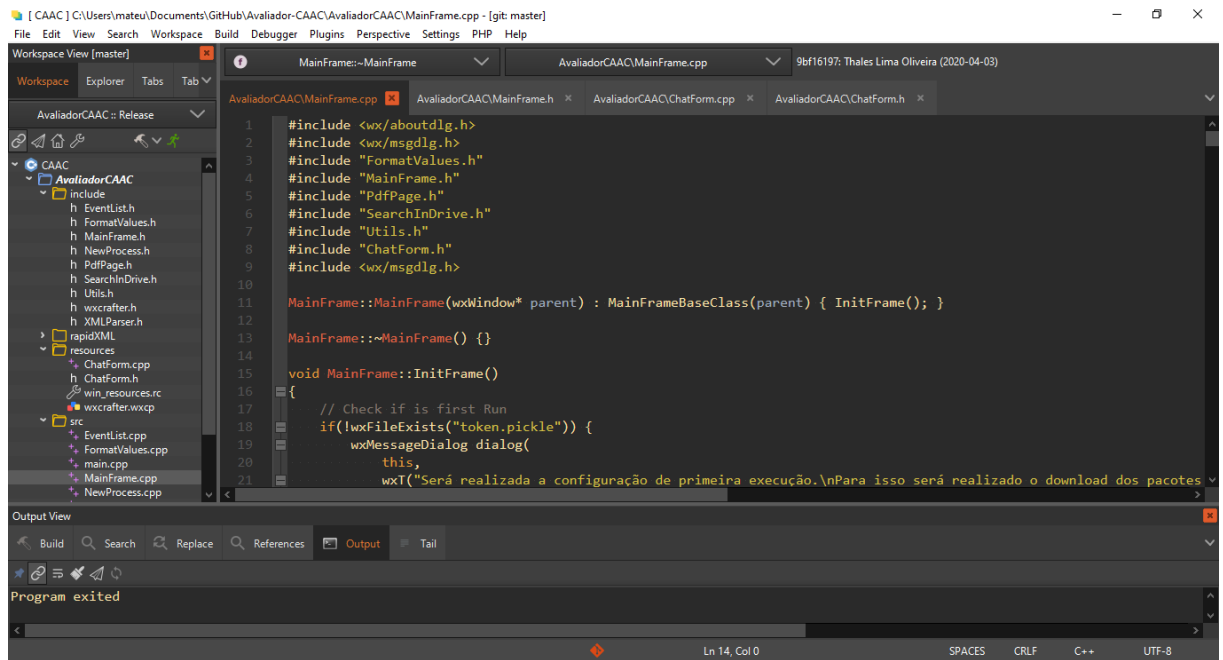
ZEITLIN, Vadim. Controles personalizados em caixas de diálogo de arquivo na versão 3.2.0. 2022. Disponível em: <https://www.wxwidgets.org> Acesso em: 16 de jan. 2023.

APÊNDICES

Apêndice I: Imagens da execução do programa







The screenshot shows the Visual Studio Code editor with the file `AvaliadorCAAC/ChatForm.cpp` open. The code implements the `OnToSendButtonClick` method for the `ChatForm` class. The code is as follows:

```

1 #include "ChatForm.h"
2
3 ChatForm::ChatForm(wxWindow* parent, wxRichTextCtrl* richTextCtrlChat)
4     : ChatFormBase(parent)
5 {
6     m_richTextCtrlChat = richTextCtrlChat;
7 }
8
9 ChatForm::~ChatForm()
10 {
11 }
12
13 void ChatForm::OnToSendButtonClick(wxCommandEvent& event)
14 {
15     m_chat = m_richTextCtrlChat->GetValue();
16     m_avaliador = m_textCtrlName->GetValue();
17     m_richTextCtrlChat->AppendText(wxT("Avaliador: ") + m_avaliador + "\n");
18     m_richTextCtrlChat->AppendText(wxT("Comentário: ") + m_chat + "\n" + .....
19     EndModal(wxID_OK);
20 }
21

```

The Output View at the bottom shows the message "Program exited".

The screenshot shows the Visual Studio Code editor with the file `AvaliadorCAAC/ChatForm.h` open. The code defines the `ChatForm` class, which inherits from `ChatFormBase`. The code is as follows:

```

4 class ChatForm : public ChatFormBase
5 {
6 public:
7     ChatForm(wxWindow* parent, wxRichTextCtrl* richTextCtrlChat);
8     virtual ~ChatForm();
9
10     //const wxString GetChat() {return "Avaliador: " + m_avaliador + " \n\n" + "Comentário: " + m_chat + " " + m_msg ;}
11
12 protected:
13     virtual void OnToSendButtonClick(wxCommandEvent& event);
14
15     wxString m_chat;
16     wxString m_avaliador;
17     wxString m_msg;
18     wxRichTextCtrl* m_richTextCtrlChat;
19 };
20

```

The Output View at the bottom shows the message "Program exited".



INSTITUTO FEDERAL
Goiás
Campus Itumbiara

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS
Campus Itumbiara

PROCESSO: Nº 23376.001838/2020-70

ASSUNTO: AVALIAÇÃO DE ATIVIDADES COMPLEMENTARES

A Comissão de Avaliação das Atividades Complementares (CAAC) dos Cursos de Graduação do Instituto Federal de Educação, Ciência e Tecnologia de Goiás, de acordo com o Regulamento das Atividades Complementares dos Cursos de Graduação (Resolução nº 16 de 26/12/2011) e Orientações para as Atividades Complementares dos Cursos de Graduação, após analisar os certificados apresentados no presente processo, valida o cumprimento de 44 horas para o aluno Mateus Lucas de Almeida.

A seguir a relação dos certificados recebidos e avaliados:

EVENTO	INSTITUIÇÃO	PERÍODO	Carga Horária Apresentada	Carga Horária Validada
Festival de Artes de Goiás - Acompanhamento dos estudantes às atividades de artes.	IFG	-	40	40
Web Semântica - Acompanhamento dos estudantes às atividades de artes	ULBRA	-	4	4
TOTAL			44	44

Observações:

1. Os certificados analisados estão apresentados na tabela acima de acordo com a ordem de apresentação no processo.
2. A carga horária validada está de acordo com a Resolução nº 16 do Regulamento das Atividades Complementares.

Itumbiara/GO, 12 de fevereiro de 2023.

Carlos Antunes de Queiroz Junior

Fernanda Hein da Costa

Thales Lima Oliveira

Apêndice II: Desenvolvimento do código

Segue abaixo o desenvolvimento do código MainFrame:

```
#include <wx/aboutdlg.h>
#include <wx/msgdlg.h>
#include "FormatValues.h"
#include "MainFrame.h"
#include "PdfPage.h"
#include "SearchInDrive.h"
#include "Utils.h"
#include "ChatForm.h"
#include <wx/msgdlg.h>

MainFrame::MainFrame(wxWindow* parent) : MainFrameBaseClass(parent) {
    InitFrame(); }

MainFrame::~MainFrame() {}

void MainFrame::InitFrame()
{
    // Check if is first Run
    if(!wxFileExists("token.pickle")) {
        wxMessageDialog dialog(
            this,
            wxT("Será realizada a configuração de primeira execução.\nPara isso será
realizado o download dos pacotes "
            "Python necessários\ne em seguida as credenciais do e-mail da CAAC
serão solicitados."),
            wxT("Atenção!"), wxICON_WARNING | wxOK);
        dialog.ShowModal();
        wxExecute(wxString::Format("cmd.exe /c firstRun.bat"), wxEXEC_SYNC /*|
m_consoleStatus*/);
    }
}
```

```

m_locale.Init();
m_iflcon.LoadFile("logo128.png", wxBITMAP_TYPE_PNG);
EnableAll(false);
BuildGrid();
m_pgPropIFGEvent->SetAttribute(wxPG_BOOL_USE_CHECKBOX, true);
m_pgPropInvalidate->SetAttribute(wxPG_BOOL_USE_CHECKBOX, true);
m_pgPropShowInReport->SetAttribute(wxPG_BOOL_USE_CHECKBOX, true);
m_pgPropIgnoreRestrictions->SetAttribute(wxPG_BOOL_USE_CHECKBOX,
true);

// Load Event list from drive
if(m_eventList.Load(this)) {
    m_pgPropEvent->SetAttribute(wxPG_ATTR_AUTOCOMPLETE,
m_eventList.GetEventList());
}

// Initialize item list
// Res 16
m_itensResN20.Add(wxT("Escolha o item..."));
m_itensResN20.Add(wxT("I - Visitas técnicas"));
m_itensResN20.Add(wxT("II - Atividade prática de campo"));
m_itensResN20.Add(
    wxT("III - Participação em eventos técnicos, culturais, científicos, acadêmicos,
artísticos e esportivos.));
m_itensResN20.Add(wxT("IV - Participação em comissão organizadora de
eventos institucionais e outros.));
m_itensResN20.Add(wxT("V - Apresentação de trabalhos em feiras, congressos,
mostras, seminários e outros.));
m_itensResN20.Add(wxT("VI - Intérprete de línguas em eventos institucionais e
outros. ));
m_itensResN20.Add(wxT("VII - Monitorias por período mínimo de um semestre
letivo.));

```

```

m_itensResN20.Add(wxT("VIII - Participação em projetos e programas de
iniciação científica e tecnológica como aluno "
    "do projeto, bolsista ou voluntário."));
m_itensResN20.Add(wxT("IX - Participação em projetos de ensino, pesquisa e
extensão com duração mínima de um "
    "semestre letivo."));
m_itensResN20.Add(wxT("X - Cursos e minicursos."));
m_itensResN20.Add(wxT("XI - Estágio curricular não obrigatório igual ou superior
a cem horas"));
m_itensResN20.Add(wxT("XII - Participação como representante de turma por um
período mínimo de um semestre letivo"));
m_itensResN20.Add(wxT("XIII - Participação como representante discente nas
instâncias da Instituição por um período "
    "mínimo de um semestre letivo."));
m_itensResN20.Add(wxT("XIV - Participação em órgãos e entidades estudantis,
de classe, sindicais ou comunitárias."));
m_itensResN20.Add(wxT("XV - Realização de trabalho comunitário."));
m_itensResN20.Add(wxT("XVI - Atividades profissionais comprovadas na área de
atuação do curso."));
m_itensResN20.Add(wxT("Atividade não pontuada"));

// Res 20
m_itensResN16.Add(wxT("Escolha o item..."));
m_itensResN16.Add(wxT("I - Visitas técnicas"));
m_itensResN16.Add(wxT("II - Atividade prática de campo"));
m_itensResN16.Add(
    wxT("III - Participação em eventos técnicos, culturais, científicos, acadêmicos,
artísticos e esportivos."));
m_itensResN16.Add(wxT("IV - Participação em comissão organizadora de
eventos institucionais e outros."));
m_itensResN16.Add(wxT("V - Apresentação de trabalhos em feiras, congressos,
mostras, seminários e outros."));
m_itensResN16.Add(wxT("VI - Intérprete de línguas em eventos institucionais e
outros. "));

```

```

        m_itensResN16.Add(wxT("VII - Monitorias por período mínimo de um semestre
        letivo."));
        m_itensResN16.Add(wxT("VIII - Participação em projetos e programas de
        iniciação científica e tecnológica como aluno "
                                "titular do projeto, bolsista ou voluntário."));
        m_itensResN16.Add(wxT("IX - Participação em programa de iniciação
        a docência."));
        m_itensResN16.Add(wxT("X - Participação em projetos de ensino, pesquisa e
        extensão com duração 60 horas mínima de um "
                                "semestre letivo."));
        m_itensResN16.Add(wxT("XI - Cursos e minicursos."));
        m_itensResN16.Add(wxT("XII - Estágio curricular não obrigatório igual ou superior
        a cem horas"));
        m_itensResN16.Add(wxT("XIII - Participação como representante de turma por
        um período mínimo de um semestre letivo"));
        m_itensResN16.Add(wxT("XIV - Participação como representante discente nas
        instâncias da Instituição por um período "
                                "mínimo de um semestre letivo."));
        m_itensResN16.Add(wxT("XV - Participação em órgãos e entidades estudantis,
        de classe, sindicais ou comunitárias."));
        m_itensResN16.Add(wxT("XVI - Realização de trabalho comunitário."));
        m_itensResN16.Add(wxT("XVII - Participação como ouvinte em defesas de
        trabalhos acadêmicos."));
        m_itensResN16.Add(wxT("Atividade não pontuada"));
    }

void MainFrame::OnExit(wxCommandEvent& event)
{
    wxUnusedVar(event);
    Close();
}

void MainFrame::OnNew(wxCommandEvent& event)
{

```

```

if(IsSaveNeeded())
    ShowSaveDialog();
else
    SaveNeeded();

NewProcess process(this, &m_protocolNumber, &m_studentName, &m_numDocs,
&m_resolution);

if(process.ShowModal() == wxID_OK) {
    ClearAll();
    EnableAll();

    auto choices = m_pgPropItem->GetChoices();
    choices.Clear();
    if(m_resolution == N_16)
        choices.Set(m_itensResN16);
    else if(m_resolution == N_20)
        choices.Set(m_itensResN20);

    m_pgPropItem->SetChoices(choices);

    wxMessageDialog dialog(this,
        wxT("Deseja buscar o aluno no Drive?\nCaso o aluno já possua
processos anteriores os "
        "comprovantes poderão ser importados."),
        wxT("Atenção!"), wxICON_WARNING | wxYES_NO);
    if(dialog.ShowModal() == wxID_YES) {
        wxString* fileID = new wxString("");
        wxString* fileName = new wxString("");
        SearchInDrive driveDialog(this, fileID, fileName);
        driveDialog.FillProtocolNum(m_protocolNumber);
        driveDialog.FillStudentName(m_studentName);
        driveDialog.DoSearch();
        if(driveDialog.ShowModal() == wxID_OK) {

```

```

wxBusyInfo busy(GetBusyInfoFlags(this, wxT("Baixando processo...")));
wxString path = "downloads\\" + *fileName + ".xml";
wxExecute(wxString::Format("%s Download \"%s\" \"%s\"", m_runDriveAPI,
*fileID, path),
          wxEXEC_SYNC | m_consoleStatus);

// Before open save the new basic data:
wxString newProtocol = m_protocolNumber;
wxString newName = m_studentName;
long int newNumDocs = m_numDocs;
Resolution newResolution = m_resolution;
// Open process
OpenProcess(path);
// Return to new values
m_protocolNumber = newProtocol;
m_studentName = newName;
m_numDocs = newNumDocs;
m_resolution = newResolution;

// Hide all activities from report
for(auto it = m_activityList.begin(), itEnd = m_activityList.end(); it != itEnd;
++it) {
    Activity* activity = *it;
    activity->SetShowInReport(false);
    if(m_index <= activity->GetId()) m_index = activity->GetId() + 1; // Set
m_index to the higher value
}
}
}
m_staticTextProcessNumber->SetLabel(wxT("Processo: ") +
m_protocolNumber);
m_staticTextStudentName->SetLabel(wxT("Aluno: ") + m_studentName);
ValidateAllActivities();
}

```

```

    for(int i = 0; i < m_numDocs; ++i) DoAppendItem();
}

void MainFrame::OnOpen(wxCommandEvent& event)
{
    wxFileDialog openFileDialog(this, _("Abrir arquivo XML"), "", "", "Arquivos XML (*.xml)|*.xml",
                                wxFD_OPEN | wxFD_FILE_MUST_EXIST);
    if(openFileDialog.ShowModal() == wxID_CANCEL) return; // the user changed
    idea...

    if(!OpenProcess(openFileDialog.GetPath())) {
        wxMessageDialog dialog(this, wxT("Não é possível abrir o arquivo."),
        wxT("Erro"), wxICON_ERROR);
        dialog.ShowModal();
        return;
    }
}

bool MainFrame::OpenProcess(wxString path)
{
    ClearAll();
    EnableAll(true);

    rapidxml::xml_document<> doc;
    rapidxml::file<> xmlFile(path.mb_str());
    doc.parse<0>(xmlFile.data());

    auto rootNode = doc.first_node("Processo");
    if(!rootNode) return false;
    auto numProtocolNode = rootNode->first_node("NumProtocolo");
    if(!numProtocolNode) return false;
    m_protocolNumber = wxString::FromUTF8(numProtocolNode->value());
    auto studentNameNode = rootNode->first_node("NomeEstudante");

```



```

if(!studentNameNode) return false;
m_studentName = wxString::FromUTF8(studentNameNode->value());
m_resolution = static_cast<Resolution>(XMLParser::GetNodeValueInt(rootNode,
"NumResolucao"));

auto activitiesNode = rootNode->first_node("Atividades");
if(!activitiesNode) return false;
auto activityNode = activitiesNode->first_node("Atividade");
while(activityNode) {
    // ID
    int id = XMLParser::GetAttributeValueInt(activityNode, "ID");
    // New activity
    Activity* activity = new Activity(id);
    // Item
    activity->SetItemCode(XMLParser::GetNodeValueInt(activityNode, "Item"));
    // Evento
    auto eventName = activityNode->first_node("Evento");
    if(!eventName) return false;
    activity->SetEventName(wxString::FromUTF8(eventName->value()));
    //Descricao evento
    auto eventDesc = activityNode->first_node("DescricaoEvento");
    if(eventDesc){
        activity->SetEventDescription(wxString::FromUTF8(eventDesc->value()));
    }
    // Num shifts
    activity->SetShiftNumber(XMLParser::GetNodeValueDouble(activityNode,
"NumTurnos"));
    // Evento IFG
    activity-
>SetIfgEvent(static_cast<bool>(XMLParser::GetNodeValueInt(activityNode,
"EventoIFG"))));
    // Instituicao
    auto institutionNode = activityNode->first_node("Instituicao");
    if(!institutionNode) return false;

```

```

activity->SetInstitutionName(wxString::FromUTF8(institutionNode->value()));
// Data
auto dateNode = activityNode->first_node("Data");
if(!dateNode) return false;
activity->SetDate(wxString::FromUTF8(dateNode->value()));
// Num shifts
activity->SetChPresented(XMLParser::GetNodeValueDouble(activityNode,
"CH"));
// Atividade valida
activity-
>SetInvalidateCH(static_cast<bool>(XMLParser::GetNodeValueInt(activityNode,
"AtividadeValida"))));
// Exibir no relatorio
activity-
>SetShowInReport(static_cast<bool>(XMLParser::GetNodeValueInt(activityNode,
"ExibirNoRelatorio"))));
// Ignorar restricoes
activity-
>SetIgnoreRestrictions(static_cast<bool>(XMLParser::GetNodeValueInt(activityNode,
"IgnorarRestricoes"))));
// Obs
auto obsNode = activityNode->first_node("Observacoes");
if(!obsNode) return false;
activity->SetObs(wxString::FromUTF8(obsNode->value()));

//Comentário
// auto commentNode = rootNode->first_node("Comentario");
//if(!commentNode) return false;
// activity->SetComment(wxString::FromUTF8(commentNode->value()));
//m_richTextCtrlChat->SetValue(wxString::FromUTF8(commentNode->value()));

m_activityList.push_back(activity);

// Append new row

```

```

        BypassSelectionEvent();
        DoAppendItem(false, false);
        // Set new row ID
        m_grid->SetCellValue(m_grid->GetNumberRows() - 1, 0,
wxString::Format("%d", id));

        activityNode = activityNode->next_sibling("Atividade");
    }

    auto commentNode = rootNode->first_node("Comentario");
    if(!commentNode) return false;
    // activity->SetComment(wxString::FromUTF8(commentNode->value()));
    m_richTextCtrlChat->SetValue(wxString::FromUTF8(commentNode-
>value()+"\n"));

    auto choices = m_pgPropItem->GetChoices();
    choices.Clear();
    if(m_resolution == N_16)
        choices.Set(m_itensResN16);
    else if(m_resolution == N_20)
        choices.Set(m_itensResN20);

    m_pgPropItem->SetChoices(choices);

    FillAllRows();
    ValidateAllActivities();
    BypassSelectionEvent(false);
    UpdateCH();

    m_staticTextProcessNumber->SetLabel(wxT("Processo: ") + m_protocolNumber);
    m_staticTextStudentName->SetLabel(wxT("Aluno: ") + m_studentName);

    return true;
}

```

```

void MainFrame::OnSave(wxCommandEvent& event)
{
    FormatValues fValues(m_protocolNumber, m_studentName);
    wxFileDialog saveFileDialog(this, wxT("Salvar arquivo XML"), "", "", "Arquivo XML
(*.xml)|*.xml",
                                wxFD_SAVE | wxFD_OVERWRITE_PROMPT);
    saveFileDialog.SetFilename(wxString::Format(wxT("%s[%s]"),
fValues.GetProtocol(), fValues.GetName()));
    if(saveFileDialog.ShowModal() == wxID_CANCEL) return;

    SaveProcess(saveFileDialog.GetPath(), saveFileDialog.GetFilename());

    // Get event list to save
    wxArrayString eventListArray;
    for(auto it = m_activityList.begin(), itEnd = m_activityList.end(); it != itEnd; ++it) {
        eventListArray.Add((*it)->GetEventName());
    }
    m_eventList.Save(this, eventListArray);
}

bool MainFrame::SaveProcess(wxString path, wxString fileName)
{
    // Erase the file (if exists or not) and write the initial data
    std::ofstream writeProjectsFile(path.mb_str());
    writeProjectsFile.close();

    rapidxml::xml_document<> doc;
    rapidxml::file<> xmlFile(path);
    doc.parse<0>(xmlFile.data());

    rapidxml::xml_node<>* decl = doc.allocate_node(rapidxml::node_declaration);
    rapidxml::xml_attribute<>* ver = doc.allocate_attribute("version", "1.0");
    rapidxml::xml_attribute<>* encoding = doc.allocate_attribute("encoding", "utf-8");

```

```

decl->append_attribute(ver);
decl->append_attribute(encoding);
doc.append_node(decl);

rapidxml::xml_node<>* rootNode = doc.allocate_node(rapidxml::node_element,
"Processo");
doc.append_node(rootNode);

rapidxml::xml_node<>* processNameNode = XMLParser::AppendNode(doc,
rootNode, "Nome");
XMLParser::SetNodeValue(doc, processNameNode, fileName);

rapidxml::xml_node<>* processNumber = XMLParser::AppendNode(doc,
rootNode, "NumProtocolo");
XMLParser::SetNodeValue(doc, processNumber, m_protocolNumber);

rapidxml::xml_node<>* studentName = XMLParser::AppendNode(doc, rootNode,
"NomeEstudante");
XMLParser::SetNodeValue(doc, studentName, m_studentName);

rapidxml::xml_node<>* resNumber = XMLParser::AppendNode(doc, rootNode,
"NumResolucao");
XMLParser::SetNodeValue(doc, resNumber, static_cast<int>(m_resolution));

rapidxml::xml_node<>* activitiesNode = XMLParser::AppendNode(doc, rootNode,
"Atividades");

for(auto it = m_activityList.begin(), itEnd = m_activityList.end(); it != itEnd; ++it) {
    Activity* activity = *it;
    // ID
    rapidxml::xml_node<>* activityNode = XMLParser::AppendNode(doc,
activitiesNode, "Atividade");
    XMLParser::SetNodeAttribute(doc, activityNode, "ID", activity->GetId());
    // Item

```

```

        rapidxml::xml_node<>* itemCode = XMLParser::AppendNode(doc, activityNode,
"Item");
        XMLParser::SetNodeValue(doc, itemCode, activity->GetItemCode());
        // Evento
        rapidxml::xml_node<>* eventName = XMLParser::AppendNode(doc,
activityNode, "Evento");
        XMLParser::SetNodeValue(doc, eventName, activity->GetEventName());
        // Descricao do evento
        rapidxml::xml_node<>* eventDesc = XMLParser::AppendNode(doc,
activityNode, "DescricaoEvento");
        XMLParser::SetNodeValue(doc, eventDesc, activity->GetEventDescription());
        // Turnos/apresentação/participação
        rapidxml::xml_node<>* numShift = XMLParser::AppendNode(doc, activityNode,
"NumTurnos");
        XMLParser::SetNodeValue(doc, numShift, activity->GetShiftNumber());
        // Evento IFG
        rapidxml::xml_node<>* ifgEvent = XMLParser::AppendNode(doc, activityNode,
"EventoIFG");
        XMLParser::SetNodeValue(doc, ifgEvent, static_cast<int>(activity-
>IsIFGEvent()));
        // Instituicao
        rapidxml::xml_node<>* institutionNode = XMLParser::AppendNode(doc,
activityNode, "Instituicao");
        XMLParser::SetNodeValue(doc, institutionNode, activity->GetInstitutionName());
        // Data
        rapidxml::xml_node<>* dateNode = XMLParser::AppendNode(doc,
activityNode, "Data");
        XMLParser::SetNodeValue(doc, dateNode, activity->GetDate());
        // CH apresentada
        rapidxml::xml_node<>* ch = XMLParser::AppendNode(doc, activityNode, "CH");
        XMLParser::SetNodeValue(doc, ch, activity->GetChPresented());
        // Atividade valida
        rapidxml::xml_node<>* validAct = XMLParser::AppendNode(doc, activityNode,
"AtividadeValida");

```

```

XMLParser::SetNodeValue(doc, validAct, static_cast<int>(activity-
>IsInvalidateCH()));
    // Exibir no relatorio
    rapidxml::xml_node<>* shownInReport = XMLParser::AppendNode(doc,
activityNode, "ExibirNoRelatorio");
    XMLParser::SetNodeValue(doc, shownInReport, static_cast<int>(activity-
>IsShownInReport()));
    // Ignorar restricoes
    rapidxml::xml_node<>* ignoreRestrictions = XMLParser::AppendNode(doc,
activityNode, "IgnorarRestricoes");
    XMLParser::SetNodeValue(doc, ignoreRestrictions, static_cast<int>(activity-
>IsRestrictionsIgnored()));
    // Obs
    rapidxml::xml_node<>* obs = XMLParser::AppendNode(doc, activityNode,
"Observacoes");
    XMLParser::SetNodeValue(doc, obs, activity->GetObs());
}
//Comentario
rapidxml::xml_node<>* commentNode = XMLParser::AppendNode(doc, rootNode,
"Comentario");
XMLParser::SetNodeValue(doc, commentNode, m_richTextCtrlChat->GetValue());

std::ofstream writeXML(path.mb_str());
writeXML << doc;
writeXML.close();

FormatValues fValues(m_protocolNumber, m_studentName);
SaveAtDrive(path, wxString::Format("%s[%s].xml", fValues.GetProtocol(),
fValues.GetName()));

SaveNeeded(false);
return true;
}

```

```

void MainFrame::EnableAll(bool enable)
{
    m_pgMgr->Enable(enable);
    m_pgPropItem->Enable(enable);
    m_pgPropEvent->Enable(enable);
    m_pgPropEventDesc->Enable(enable);
    m_pgPropShiftN->Enable(enable);
    m_pgPropIFGEvent->Enable(enable);
    m_pgPropInstitution->Enable(enable);
    m_pgPropDate->Enable(enable);
    m_pgPropCH->Enable(enable);
    m_pgPropInvalidate->Enable(enable);
    m_pgPropShowInReport->Enable(enable);
    m_pgPropIgnoreRestrictions->Enable(enable);
    m_pgPropObs->Enable(enable);

    m_buttonAppendItem->Enable(enable);
    m_buttonChat->Enable(enable);
    m_buttonRemoveItem->Enable(enable);
    m_bmpButtonUp->Enable(enable);
    m_bmpButtonDown->Enable(enable);

}

void MainFrame::BuildGrid()
{
    m_grid->SetSelectionMode(wxGrid::wxGridSelectRows);
    m_grid->SetDefaultCellAlignment(wxALIGN_CENTRE, wxALIGN_CENTRE);

    //->HideRowLabels();
    m_grid->SetRowLabelSize(wxGRID_AUTOSIZE);
    m_grid->SetLabelFont(m_grid->GetLabelFont().Bold());
    m_grid->AppendCols(9, false);
    m_grid->SetColMinimalAcceptableWidth(100);

```



```

m_grid->SetColLabelValue(0, wxT("ID"));
// m_grid->SetColSize(0, 10);
m_grid->HideCol(0);
m_grid->SetColLabelValue(1, wxT("Item"));
m_grid->SetColSize(1, 100);
m_grid->SetColLabelValue(2, wxT("Evento"));
m_grid->SetColSize(2, 200);
m_grid->SetColLabelValue(3, wxT("TAP"));
m_grid->SetColSize(3, 100);
m_grid->SetColLabelValue(4, wxT("Evento IFG"));
m_grid->SetColSize(4, 100);
m_grid->SetColLabelValue(5, wxT("Instituição"));
m_grid->SetColSize(5, 200);
m_grid->SetColLabelValue(6, wxT("CH Apresentada"));
m_grid->SetColSize(6, 100);
m_grid->SetColLabelValue(7, wxT("Data Evento"));
m_grid->SetColSize(7, 200);
m_grid->SetColLabelValue(8, wxT("CH Validada"));
m_grid->SetColSize(8, 100);
m_grid->AutoSizeColumns();
// m_grid->ForceRefresh();
}

void MainFrame::GridCellSelected(wxGridEvent& event)
{
    if(!IsSelectionEventBypassed()) {
        long int id = -1;
        if(m_grid->GetCellValue(event.GetRow(), 0).ToLong(&id)) FillPG(id);
    }
    BypassSelectionEvent(false);

    event.Skip();
}

```

```
void MainFrame::AppendItem(wxCommandEvent& event) //Inseir Item
{
    DoAppendItem();

    event.Skip();
}
```

```
void MainFrame::RemoveItem(wxCommandEvent& event) //Remover Item
{
    DoRemoveItem();

    event.Skip();
}
```

```
void MainFrame::OnPGChange(wxPropertyGridEvent& event) {
    UpdateChangedPG(); }
```

```
void MainFrame::ChangePropertyText(wxPGProperty* property, wxString newText)
{
    wxPGCell cell = property->GetCell(0);
    cell.SetText(newText);
    property->SetCell(0, cell);
}
```

```
void MainFrame::EnableAllPG(bool enable)
{
    m_pgPropShiftN->Enable(enable);
    m_pgPropIFGEvent->Enable(enable);
    m_pgPropCH->Enable(enable);
}
```

```
void MainFrame::ValidateCH(Activity* activity)
{
    if(activity->IsInvalidateCH()) {
```

```

    activity->SetChValidated(0.0);
    activity->SetLimitedCH(true);
    return;
}

```

```

double chLanguage = 0.0;
double chCourses = 0.0;
double chComWork = 0.0;
double chProfActWork = 0.0;
double chTCC = 0.0;
double chEventNonIFG = 0.0;

```

```

for(auto it = m_activityList.begin(), itEnd = m_activityList.end(); it != itEnd; ++it) {
    Activity* activity = *it;
    activity->SetLimitedCH(false);
    if(activity->IsInvalidateCH()) {
        activity->SetChValidated(0.0);
        activity->SetLimitedCH(true);
    } else {
        switch(activity->GetItemCode()) {
            case 1:
            case 2:
            case 5: {
                activity->SetChValidated(activity->GetShiftNumber() * 4.0);
                activity->SetChPresented(activity->GetShiftNumber() * 4.0);
            } break;
            case 3:
            case 4: {
                double ch = activity->GetChPresented();
                if(!activity->IsIFGEvent() && !activity->IsRestrictionsIgnored()) {
                    ch *= 0.8;
                    double oldChEventNonIFG = chEventNonIFG;
                    chEventNonIFG += ch;
                    if(chEventNonIFG > m_chCompActivity * 0.5) {

```

```

        double partialCH = m_chCompActivity * 0.5 - oldCHEventNonIFG;
        if(partialCH > 0) {
            activity->SetChValidated(partialCH);
            activity->SetLimitedCH(true);
        } else {
            activity->SetChValidated(0.0);
            activity->SetLimitedCH(true);
        }
    } else {
        activity->SetChValidated(ch);
    }
    activity->SetLimitedCH(true);
} else
    activity->SetChValidated(ch);
} break;
case 6:
case 11:
case 16:
case 19: {
    double ch = activity->GetChPresented();
    double oldCH = 0.0;
    double accCH = 0.0;
    if(activity->GetItemCode() == 6) {
        oldCH = chLanguage;
        chLanguage += ch;
        accCH = chLanguage;
    } else if(activity->GetItemCode() == 11) {
        oldCH = chCourses;
        chCourses += ch;
        accCH = chCourses;
    } else if(activity->GetItemCode() == 16) {
        oldCH = chComWork;
        chComWork += ch;
        accCH = chComWork;
    }
}

```

```

    }
    else if(activity->GetItemCode() == 19) {
        oldCH = chProfActWork;
        chProfActWork += ch;
        accCH = chProfActWork;
    }
    if((accCH > m_chCompActivity * 0.4) && !activity-
>IsRestrictionsIgnored()) {
        double partialCH = m_chCompActivity * 0.4 - oldCH;
        if(partialCH > 0) {
            if(partialCH < 0.5) partialCH = 0.0; // Remove garbage result (i.e.
when "m_chCompActivity * 0.4 - oldCH" get very close to zero)
            activity->SetChValidated(partialCH);
            activity->SetLimitedCH(true);
        } else {
            activity->SetChValidated(0.0);
            activity->SetLimitedCH(true);
        }
    } else
        activity->SetChValidated(ch);
} break;
case 7:
case 8:
case 9: {
    activity->SetChValidated(60.0);
    activity->SetChPresented(60.0);
} break;
case 10:{
    double ch = activity->GetChPresented();
    activity->SetChValidated(60.0);
    activity->SetChPresented(ch);
} break;
case 12: {
    activity->SetChValidated(60.0);

```

```

    } break;
    case 13:
    case 14:
    case 15: {
        activity->SetChValidated(30.0);
        activity->SetChPresented(30.0);
    } break;
    case 17: {
        double ch = activity->GetShiftNumber() * 2.0;
        activity->SetChPresented(ch);
        double oldCHTCC = chTCC;
        chTCC += ch;
        if((chTCC > m_chCompActivity * 0.2) && !activity-
>IsRestrictionsIgnored()) {
            double partialCH = m_chCompActivity * 0.2 - oldCHTCC;
            if(partialCH > 0) {
                activity->SetChValidated(partialCH);
                activity->SetLimitedCH(true);
            } else {
                activity->SetChValidated(0.0);
                activity->SetLimitedCH(true);
            }
        } else {
            activity->SetChValidated(ch);
        }

    } break;
    case 18: {
        activity->SetChValidated(0.0);
        activity->SetChPresented(activity->GetChPresented());
    } break;
    default:
        break;
}

```

```

    }
}
SaveNeeded();
}

void MainFrame::ValidateAllActivities()
{
    for(auto it = m_activityList.begin(), itEnd = m_activityList.end(); it != itEnd; ++it) {
        Activity* activity = *it;
        if(activity->GetItemCode() != 0) {
            ValidateCH(activity);
            UpdateRowCH(activity);
        }
    }
}

void MainFrame::UpdateRowCH(Activity* activity)
{
    int rowNumber = -1;
    for(int i = 0; i < m_grid->GetNumberRows(); ++i) {
        long int id = -1;
        if(m_grid->GetCellValue(i, 0).ToLong(&id)) {
            if(id == activity->GetId()) { rowNumber = i; }
        }
    }
    if(rowNumber == -1) return;

    m_grid->SetCellValue(rowNumber, 8, wxString::Format("%f", activity-
>GetChValidated()));

    UpdateRowColour(activity, rowNumber);

    m_grid->Refresh();
}

```

```

void MainFrame::FillPG(int id)
{
    // Find activity
    for(auto it = m_activityList.begin(), itEnd = m_activityList.end(); it != itEnd; ++it) {
        Activity* activity = *it;
        if(activity->GetId() == id) {
            m_pgPropItem->SetValueFromInt(ConvertItemCodeToProp(activity-
>GetItemCode(), m_resolution));
            m_pgPropEvent->SetValueFromString(activity->GetEventName());
            m_pgPropEventDesc->SetValueFromString(activity->GetEventDescription());
            m_pgPropShiftN->SetValueFromInt(activity->GetShiftNumber());
            m_pgPropIFGEvent->SetValueFromInt(activity->IsIFGEvent());
            m_pgPropInstitution->SetValueFromString(activity->GetInstitutionName());
            m_pgPropDate->SetValueFromString(activity->GetDate());
            m_pgPropCH->SetValueFromString(wxString::Format("%f", activity-
>GetChPresented()));
            m_pgPropInvalidate->SetValueFromInt(activity->IsInvalidateCH());
            m_pgPropShowInReport->SetValueFromInt(activity->IsShownInReport());
            m_pgPropIgnoreRestrictions->SetValueFromInt(activity-
>IsRestrictionsIgnored());
            m_pgPropObs->SetValueFromString(activity->GetObs());

        }
    }

    UpdateChangedPG();
}

void MainFrame::UpdateChangedPG()
{
    EnableAllPG(false);
    bool shiftN = false;
    bool ifgEvent = false;

```



```

bool ch = false;

switch(ConvertPropToItemCode(m_pgPropItem->GetValue().GetInteger(),
m_resolution)) {
    case 0: {
        return;
    } break;
    case 1:
    case 2: {
        ChangePropertyText(m_pgPropShiftN, wxT("Num. Turnos"));
        m_pgPropShiftN->Enable(true);
        shiftN = true;
    } break;
    case 3:
    case 4: {
        ChangePropertyText(m_pgPropShiftN, wxT("Num. Participações"));
        m_pgPropShiftN->Enable(true);
        m_pgPropIFGEvent->Enable(true);
        m_pgPropCH->Enable(true);
        shiftN = true;
        ifgEvent = true;
        ch = true;
    } break;
    case 5: {
        ChangePropertyText(m_pgPropShiftN, wxT("Num. Apresentações"));
        m_pgPropShiftN->Enable(true);
        shiftN = true;
    } break;
    case 6:
    case 11:
    case 16:
    case 19: {
        m_pgPropCH->Enable(true);
        ch = true;
    }
}

```

```

    } break;
    case 7:
    case 8:
    case 9:
    {
        // nothing really matters...
    } break;
    case 10:
    {
        m_pgPropCH->Enable(true);
        ch = true;
    } break;
    case 12: {
        m_pgPropCH->Enable(true);
        ch = true;
    } break;
    case 13:
    case 14:
    case 15: {
        // to me...
    } break;
    case 17: {
        ChangePropertyText(m_pgPropShiftN, wxT("Num. Participações"));
        m_pgPropShiftN->Enable(true);
        shiftN = true;
    } break;
    case 18: {
        m_pgPropCH->Enable(true);
        ch = true;
    }
    default:
        break;
}

```

```

m_pgMgr->Refresh();

if(!m_grid->GetSelectedRows().IsEmpty() && m_pgPropItem-
>GetValue().GetInteger() != 0) {
    // Get current activity from list
    int rowNumber = m_grid->GetSelectedRows()[0];
    Activity* currentActivity = NULL;
    for(auto it = m_activityList.begin(), itEnd = m_activityList.end(); it != itEnd; ++it) {
        Activity* activity = *it;
        long int id = -1;
        if(!m_grid->GetCellValue(rowNumber, 0).ToLong(&id)) break;
        if(activity->GetId() == id) { currentActivity = activity; }
    }

    // Fill the current activity with new informations
    currentActivity->SetItemCode(ConvertPropToItemCode(m_pgPropItem-
>GetValue().GetInteger(), m_resolution));
    currentActivity->SetEventName(m_pgPropEvent->GetValue().GetString());
    currentActivity->SetEventDescription(m_pgPropEventDesc-
>GetValue().GetString());
    currentActivity->SetShiftNumber(m_pgPropShiftN->GetValue().GetDouble());
    currentActivity->SetIfgEvent(m_pgPropIfgEvent->GetValue().GetBool());
    currentActivity->SetInstitutionName(m_pgPropInstitution-
>GetValue().GetString());
    currentActivity->SetChPresented(m_pgPropCH->GetValue().GetDouble());
    currentActivity->SetDate(m_pgPropDate->GetValue().GetString());
    currentActivity->SetInvalidateCH(m_pgPropInvalidate->GetValue().GetBool());
    currentActivity->SetShowInReport(m_pgPropShowInReport-
>GetValue().GetBool());
    currentActivity->SetIgnoreRestrictions(m_pgPropIgnoreRestrictions-
>GetValue().GetBool());
    currentActivity->SetObs(m_pgPropObs->GetValue().GetString());

    ValidateCH(currentActivity);

```

```

if(currentActivity) {
    m_grid->SetCellValue(rowNumber, 1, m_pgPropItem->GetDisplayedString());

    m_grid->SetCellValue(rowNumber, 2, currentActivity->GetEventName());

    wxString shiftNStr = "-";
    if(shiftN) shiftNStr = wxString::Format("%d", static_cast<int>(currentActivity-
>GetShiftNumber()));
    m_grid->SetCellValue(rowNumber, 3, shiftNStr);

    wxString ifgEventStr = "-";
    if(ifgEvent) ifgEventStr = currentActivity->IsIFGEvent() ? wxT("Sim") :
wxT("Não");
    m_grid->SetCellValue(rowNumber, 4, ifgEventStr);

    m_grid->SetCellValue(rowNumber, 5, currentActivity->GetInstitutionName());

    wxString chPresentedStr = "-";
    if(ch) chPresentedStr = wxString::Format("%f", currentActivity-
>GetChPresented());
    m_grid->SetCellValue(rowNumber, 6, chPresentedStr);

    m_grid->SetCellValue(rowNumber, 7, currentActivity->GetDate());

    m_grid->SetCellValue(rowNumber, 8, wxString::Format("%f", currentActivity-
>GetChValidated()));

    // UpdateRowColour(currentActivity, rowNumber);
}
}

ValidateAllActivities();

```

```

m_grid->Refresh();

UpdateCH();
SaveNeeded();
}

void MainFrame::GoUp(wxCommandEvent& event)
{
    if(!m_grid->GetSelectedRows().IsEmpty()) {
        wxArrayInt rowArr = m_grid->GetSelectedRows();
        rowArr.Sort(CompareBackward);
        for(int i = rowArr.Count() - 1; i >= 0; --i) {
            int rowNumber = rowArr[i];
            if(rowNumber != 0) {
                for(int j = 0; j < m_grid->GetNumberCols(); ++j) {
                    auto tmpValue = m_grid->GetCellValue(rowNumber, j);
                    m_grid->SetCellValue(rowNumber, j, m_grid->GetCellValue(rowNumber
- 1, j));

                    m_grid->SetCellValue(rowNumber - 1, j, tmpValue);
                }
                // Swap activity at the list
                auto tmpActivity = m_activityList[rowNumber];
                m_activityList[rowNumber] = m_activityList[rowNumber - 1];
                m_activityList[rowNumber - 1] = tmpActivity;

                BypassSelectionEvent();
                m_grid->SelectRow(rowNumber - 1, true);
                m_grid->DeselectRow(rowNumber);

                UpdateRowColour(m_activityList[rowNumber], rowNumber);
                UpdateRowColour(m_activityList[rowNumber - 1], rowNumber - 1);
            }
        }
    }
    m_grid->Refresh();
}

```

```

        SaveNeeded();
    }
    void MainFrame::GoDown(wxCommandEvent& event)
    {
        if(!m_grid->GetSelectedRows().IsEmpty()) {
            wxArrayInt rowArr = m_grid->GetSelectedRows();
            rowArr.Sort(CompareForward);
            for(int i = rowArr.Count() - 1; i >= 0; --i) {
                int rowNumber = rowArr[i];
                if(rowNumber != m_grid->GetNumberRows() - 1) {
                    for(int j = 0; j < m_grid->GetNumberCols(); ++j) {
                        auto tmpValue = m_grid->GetCellValue(rowNumber, j);
                        m_grid->SetCellValue(rowNumber, j, m_grid->GetCellValue(rowNumber
+ 1, j));
                        m_grid->SetCellValue(rowNumber + 1, j, tmpValue);
                    }
                    // Swap activity at the list
                    auto tmpActivity = m_activityList[rowNumber];
                    m_activityList[rowNumber] = m_activityList[rowNumber + 1];
                    m_activityList[rowNumber + 1] = tmpActivity;

                    BypassSelectionEvent();
                    m_grid->SelectRow(rowNumber + 1, true);
                    m_grid->DeselectRow(rowNumber);

                    UpdateRowColour(m_activityList[rowNumber], rowNumber);
                    UpdateRowColour(m_activityList[rowNumber + 1], rowNumber + 1);
                }
            }
        }
        m_grid->Refresh();
        SaveNeeded();
    }

```

```

int MainFrame::CompareForward(int* a, int* b)
{
    if(*a > *b)
        return 1;
    else if(*a < *b)
        return -1;
    else
        return 0;
}

```

```

int MainFrame::CompareBackward(int* a, int* b)
{
    if(*a > *b)
        return -1;
    else if(*a < *b)
        return 1;
    else
        return 0;
}

```

```

void MainFrame::EditProcess(wxCommandEvent& event)
{
    m_numDocs = m_grid->GetNumberRows();
    NewProcess process(this, &m_protocolNumber, &m_studentName, &m_numDocs,
&m_resolution, false);
    process.FillFields();
    if(process.ShowModal() == wxID_OK) {
        EnableAll();
        m_staticTextProcessNumber->SetLabel(wxT("Processo: ") +
m_protocolNumber);
        m_staticTextStudentName->SetLabel(wxT("Aluno: ") + m_studentName);
    }
    SaveNeeded();
}

```

```

void MainFrame::ClearAll()
{
    EnableAll(false);
    m_grid->DeleteRows(0, m_grid->GetNumberRows());
    for(auto it = m_activityList.begin(), itEnd = m_activityList.end(); it != itEnd; ++it) {
        if(*it) delete *it;
    }
    m_activityList.clear();
    m_index = 0;
}

```

```

void MainFrame::DoAppendItem(bool createActivity, bool selectRow)
{
    m_grid->AppendRows(1, true);
    m_grid->SetCellValue(m_grid->GetNumberRows() - 1, 0, wxString::Format("%d",
m_index));
    m_grid->SetCellAlignment(m_grid->GetNumberRows() - 1, 1, wxALIGN_LEFT,
wxALIGN_LEFT); // Item
    m_grid->SetCellAlignment(m_grid->GetNumberRows() - 1, 2, wxALIGN_LEFT,
wxALIGN_LEFT); // Evento
    m_grid->SetCellAlignment(m_grid->GetNumberRows() - 1, 5, wxALIGN_LEFT,
wxALIGN_LEFT); // Instituição

```

```

    if(createActivity) {
        Activity* newActivity = new Activity(m_index);
        m_activityList.emplace_back(newActivity);
    }
    if(selectRow) {
        BypassSelectionEvent();
        m_grid->SelectRow(m_grid->GetNumberRows() - 1);
    }

```

```

    m_index++;

```



```

        m_grid->SetRowLabelSize(wxGRID_AUTOSIZE);
        SaveNeeded();
    }

void MainFrame::DoRemoveItem()
{
    if(!m_grid->GetSelectedRows().IsEmpty()) {
        wxArrayInt selArr = m_grid->GetSelectedRows();
        selArr.Sort(CompareForward);
        for(int i = selArr.Count() - 1; i >= 0; --i) {
            long int id = -1;
            m_grid->GetCellValue(selArr[i], 0).ToLong(&id);
            // Remove activity from list
            for(auto it = m_activityList.begin(), itEnd = m_activityList.end(); it != itEnd;
++it) {
                Activity* activity = *it;
                if(activity->GetId() == id) {
                    m_activityList.erase(it);
                    break;
                }
            }
            BypassSelectionEvent();
            m_grid->DeleteRows(selArr[i], 1);
        }
    }
    ValidateAllActivities();
    SaveNeeded();
}

void MainFrame::UpdateCH()
{
    double chRequested = 0.0;
    double chValidated = 0.0;
    for(auto it = m_activityList.begin(), itEnd = m_activityList.end(); it != itEnd; ++it) {

```

```

    Activity* activity = *it;
    if(activity->IsShownInReport()) {
        chRequested += activity->GetChPresented();
        chValidated += activity->GetChValidated();
    }
}

m_totalReqCH = chRequested;
m_totalValCH = chValidated;

m_staticTextTotalRequested->SetLabel(wxString::Format(wxT("Total solicitado =
%.2f h"), chRequested));
m_staticTextTotalValidated->SetLabel(wxString::Format(wxT("Total validado = %.2f
h"), chValidated));
}

void MainFrame::GenerateReport(wxCommandEvent& event)
{
    PdfPage pdf;
    pdf.AliasNbPages();
    pdf.AddPage();

    // Logo
    pdf.Image(wxS("logoIFG.png"), 10, 13, 55);
    pdf.SetFont(wxS("Arial"), wxS("B"), 9);
    // Move to the right
    pdf.Cell(65);
    // Title
    pdf.Cell(0, 10, wxS("MINISTÉRIO DA EDUCAÇÃO"), wxPDF_BORDER_NONE, 0,
wxPDF_ALIGN_LEFT);
    // Line break
    pdf.Ln(4);
    pdf.Cell(65);

```

```

pdf.Cell(0, 10, wxS("SECRETARIA DE EDUCAÇÃO PROFISSIONAL E
TECNOLOGICA"), wxPDF_BORDER_NONE, 0, wxPDF_ALIGN_LEFT);
pdf.Ln(4);
pdf.Cell(65);
pdf.Cell(0, 10, wxS("INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DE GOIÁS"), wxPDF_BORDER_NONE, 0,
    wxPDF_ALIGN_LEFT);
pdf.Ln(4);
pdf.Cell(65);
pdf.Cell(0, 10, wxS("Campus Itumbiara"), wxPDF_BORDER_NONE, 0,
wxPDF_ALIGN_LEFT);
pdf.Ln(10);

pdf.Ln(6);
pdf.SetFont(wxS("Arial"), wxS("B"), 12);
pdf.Cell(0, 10, wxString::Format(wxS("PROCESSO: N° %s"), m_protocolNumber),
0, 1);
pdf.SetFont(wxS("Arial"), wxS("B"), 11);
pdf.Cell(0, 10, wxString::Format(wxS("ASSUNTO: AVALIAÇÃO DE ATIVIDADES
COMPLEMENTARES"), m_protocolNumber), 0, 1);
pdf.Ln(1);
pdf.Cell(0, 6.6);
pdf.Ln(6);

pdf.SetFont(wxS("Times"), wxS(""), 11);

wxFileInputStream f(wxS("head.txt"));
int len = f.GetLength();
char* ctxt = new char[len + 1];
f.Read(ctxt, len);
ctxt[len] = '\0';
wxString txt(ctxt, *wxConvCurrent);
txt.Replace("\\courseLevel", m_resolution == N_16 ? wxT("de Graduação") :
wxT("Técnicos"));

```

```

    txt.Replace("\\resNumDate", m_resolution == N_16 ? wxT("16 de 26/12/2011") :
wxT("20 de 26/12/2011"));
    txt.Replace("\\hours", wxString::FromDouble(m_totalValCH));
    txt.Replace("\\studentName", m_studentName);
    // Output justified text
    // pdf.Ln(8);
    pdf.WriteXml(txt);
    pdf.Ln(1);
    pdf.SetFont(wxS("Times"), wxS(""), 12);
    pdf.Cell(0, 0, wxS("A seguir a relação dos certificados recebidos e avaliados:"),
wxPDF_BORDER_NONE, 0,
        wxPDF_ALIGN_CENTER);
    pdf.Ln(6);

    wxFileInputStream fTH(wxS("tableHead.txt"));
    len = fTH.GetLength();
    ctxt = new char[len + 1];
    fTH.Read(ctxt, len);
    ctxt[len] = '\0';
    wxString fTHtxt(ctxt, *wxConvCurrent);

    wxFileInputStream fTB(wxS("tableBody.txt"));
    len = fTB.GetLength();
    ctxt = new char[len + 1];
    fTB.Read(ctxt, len);
    ctxt[len] = '\0';
    wxString fTBtxt(ctxt, *wxConvCurrent);

    wxFileInputStream fTF(wxS("tableFoot.txt"));
    len = fTF.GetLength();
    ctxt = new char[len + 1];
    fTF.Read(ctxt, len);
    ctxt[len] = '\0';
    wxString fTFtxt(ctxt, *wxConvCurrent);

```

```

wxString table = fTHtxt;
int numObs = 2;

// Get the observations that are equals and associate with a number (starting with
2)
std::vector<std::pair<Activity*, int>> obsNumActList; // List of numbers associated
with activity with obs strings
std::vector<std::pair<wxString, int>> obsUniqueList; // List of unique obs strings
for(auto* activity : m_activityList) {
    if(activity->GetObs() != "") {
        wxString obs = activity->GetObs();

        // Check previous list
        bool isRepeatedObs = false;
        int currentObsNum = 0;
        for(auto prevObs : obsUniqueList) {
            if(prevObs.first == obs){
                isRepeatedObs = true;
                currentObsNum = prevObs.second;
            }
        }
        if(!isRepeatedObs) {
            currentObsNum = ++numObs;
            obsUniqueList.push_back(std::make_pair(obs, currentObsNum));
        }

        obsNumActList.push_back(std::make_pair(activity, currentObsNum));
    }
}

for(auto it = m_activityList.begin(), itEnd = m_activityList.end(); it != itEnd; ++it) {
    Activity* activity = *it;

```

```

    if(activity->IsShownInReport()) {
        wxString activityStr = fTBtxt;
        wxString eventName = activity->GetEventName();
        if(activity->GetEventDescription() != "") eventName += " - " + activity-
>GetEventDescription();
        activityStr.Replace("\\event", eventName);
        activityStr.Replace("\\institution", activity->GetInstitutionName());
        activityStr.Replace("\\date", activity->GetDate());
        activityStr.Replace("\\requiredCH", wxString::FromDouble(activity-
>GetChPresented()));
        wxString validatedCHStr = wxString::FromDouble(activity->GetChValidated());
        if(activity->GetObs() != "") {
            // Find obs number in list
            int currentObsNumber = 0;
            for(auto obsNumber : obsNumActList) {
                if(obsNumber.first == activity) currentObsNumber = obsNumber.second;
            }
            validatedCHStr += wxString::Format(wxS("<sup>%d</sup>"),
currentObsNumber);
        }
        activityStr.Replace("\\validatedCH", validatedCHStr);
        table += activityStr;
    }
}

fTFtxt.Replace("\\totalReqCH", wxString::FromDouble(m_totalReqCH));
fTFtxt.Replace("\\totalValCH", wxString::FromDouble(m_totalValCH));

table += fTFtxt;

pdf.SetFont(wxS("Times"), wxS(""), 10);
pdf.WriteXml(table);

pdf.Ln(6);

```

```

pdf.SetFont(wxS("Times"), wxS(""), 12);
pdf.Cell(0, 0, wxT("Observações:"));

pdf.Ln();
wxString olStr = "<ol type='1'>";
olStr += wxT("<li>Os certificados analisados estão apresentados na tabela acima
de acordo com a ordem de "
    "apresentação no processo.</li>");
olStr += wxString::Format(wxT("<li>A carga horária validada está de acordo com a
Resolução nº %s do Regulamento "
    "das Atividades Complementares.</li>"),
    m_resolution == N_16 ? wxT("16") : wxT("20"));
for(auto obs : obsUniqueList) {
    olStr += "<li>" + obs.first + "</li>";
}
olStr += "</ol>";
pdf.WriteXml(olStr);

pdf.Ln(15);
wxDateTime dateNow = wxDateTime::Now();
pdf.Cell(0, 0,
    wxString::Format("Itumbiara/GO, %d de %s de %d.", dateNow.GetDay(),
        wxDateTime::GetMonthName(dateNow.GetMonth()),
dateNow.GetYear()),
    wxPDF_BORDER_NONE, 0, wxPDF_ALIGN_RIGHT);
pdf.Ln(10);

wxFileInputStream fSgn(wxS("signatures.txt"));
len = fSgn.GetLength();
ctxt = new char[len + 1];
fSgn.Read(ctxt, len);
ctxt[len] = '\0';
wxString fSgnTxt(ctxt, *wxConvCurrent);
pdf.WriteXml(fSgnTxt);

```

```

FormatValues fValues(m_protocolNumber, m_studentName);
wxFileDialog saveFileDialog(this, _("Salvar PDF"), "", "", "Arquivos PDF
(*.pdf)|*.pdf",
                           wxFD_SAVE | wxFD_OVERWRITE_PROMPT);
saveFileDialog.SetFilename(wxString::Format(wxT("%s[%s]"),
fValues.GetProtocol(), fValues.GetName()));
if(saveFileDialog.ShowModal() == wxID_CANCEL) return;

// save the current contents in the file;
if(!pdf.SaveAsFile(saveFileDialog.GetPath())) {
    wxMessageDialog dialog(
        this, wxString::Format(wxT("Não é possível salvar o arquivo em '%s'."),
saveFileDialog.GetPath()),
        wxT("Erro"), wxICON_ERROR);
    dialog.ShowModal();
    return;
} else {
    wxLaunchDefaultApplication(saveFileDialog.GetPath());
}
}

void MainFrame::FillAllRows()
{
    for(int i = 0; i < m_grid->GetNumberRows(); ++i) {
        // Get activity
        long int id = -1;
        m_grid->GetCellValue(i, 0).ToLong(&id);
        Activity* activity = NULL;
        for(auto it = m_activityList.begin(), itEnd = m_activityList.end(); it != itEnd; ++it) {
            if((*it)->GetId() == id) {
                activity = *it;
                break;
            }
        }
    }
}

```



```
}

if(activity) {
    bool shiftN = false;
    bool ifgEvent = false;
    bool ch = false;

    switch(activity->GetItemCode()) {
        case 1:
        case 2: {
            shiftN = true;
        } break;
        case 3:
        case 4: {
            shiftN = true;
            ifgEvent = true;
            ch = true;
        } break;
        case 5: {
            shiftN = true;
        } break;
        case 6:
        case 11:
        case 16:
        case 19: {
            ch = true;
        } break;
        case 7:
        case 8:
        case 9: {
            // nothing really matters...
        } break;
        case 10:
        case 12: {
```

```

        ch = true;
    } break;
    case 13:
    case 14:
    case 15: {
        // to me...
    } break;
    case 17: {
        shiftN = true;
    } break;
    case 18: {
        ch = true;
    } break;
    default:
        break;
}

```

```

m_pgPropItem->SetValueFromInt(ConvertItemCodeToProp(activity-
>GetItemCode(), m_resolution));

```

```

m_grid->SetCellValue(i, 1, m_pgPropItem->GetValueAsString());

```

```

m_grid->SetCellValue(i, 2, activity->GetEventName());

```

```

wxString shiftNStr = "-";

```

```

if(shiftN) shiftNStr = wxString::Format("%d", static_cast<int>(activity-
>GetShiftNumber()));

```

```

m_grid->SetCellValue(i, 3, shiftNStr);

```

```

wxString ifgEventStr = "-";

```

```

if(ifgEvent) ifgEventStr = activity->IsIFGEvent() ? wxT("Sim") : wxT("Não");

```

```

m_grid->SetCellValue(i, 4, ifgEventStr);

```

```

m_grid->SetCellValue(i, 5, activity->GetInstitutionName());

```

```

        wxString chPresentedStr = "-";
        if(ch) chPresentedStr = wxString::Format("%f", activity->GetChPresented());
        m_grid->SetCellValue(i, 6, chPresentedStr);

        m_grid->SetCellValue(i, 7, activity->GetDate());

        m_grid->SetCellValue(i, 8, wxString::Format("%f", activity-
>GetChValidated()));

        UpdateRowColour(activity, i);
    }
}
m_grid->Refresh();
}
void MainFrame::OnAboutClick(wxCommandEvent& event)
{
    wxAboutDialogInfo aboutInfo;
    aboutInfo.SetName(wxT("Avaliador CAAC"));
    aboutInfo.SetVersion(MY_APP_VERSION_STRING);
    aboutInfo.SetDescription(
        wxT("Software auxiliar para cálculo e geração de relatórios de avaliação de
horas complementares pela CAAC\n\n"
        "Comissão para Avaliação de Atividades Complementares - Setor da
Indústria\nIFG - Câmpus Itumbiara"));
    aboutInfo.SetCopyright("(C) 2020-2021");
    aboutInfo.AddDeveloper("Thales Lima Oliveira (thales.oliveira@ifg.edu.br)");
    wxGenericAboutDialog dlgAbout(aboutInfo, this);
    dlgAbout.ShowModal();
}

void MainFrame::OnCloseEvent(wxCloseEvent& event)
{
    if(IsSaveNeeded())
        event.Skip(ShowSaveDialog());
}

```

```

else
    event.Skip();
}

bool MainFrame::ShowSaveDialog()
{
    wxMessageDialog msgDialog(this, wxT("Existem dados não gravados no
disco.\nDeseja salvar?"), wxT("Atenção!"),
                               wxYES_NO | wxCANCEL | wxICON_EXCLAMATION);
    switch(msgDialog.ShowModal()) {
        case wxID_CANCEL: {
            return false;
        } break;
        case wxID_YES: {
            FormatValues fValues(m_protocolNumber, m_studentName);
            wxFileDialog saveFileDialog(this, wxT("Salvar arquivo XML"), "", "", "Arquivo
XML (*.xml)|*.xml",
                                       wxFD_SAVE | wxFD_OVERWRITE_PROMPT);
            saveFileDialog.SetFilename(wxString::Format(wxT("%s[%s]"),
fValues.GetProtocol(), fValues.GetName()));
            if(saveFileDialog.ShowModal() == wxID_CANCEL) return false;
            SaveProcess(saveFileDialog.GetPath(), saveFileDialog.GetFilename());
            return true;
        } break;
        case wxID_NO: {
            return true;
        } break;
    }

    return false;
}

void MainFrame::OnPanelKeyDown(wxKeyEvent& event)
{

```

```

if(event.ShiftDown() && event.GetUnicodeKey() == 'E') {
    EventList eventList;
    eventList.Load(this);
}
event.Skip();
}

void MainFrame::SaveAtDrive(wxString path, wxString fileName)
{
    wxArrayString foundIDs;
    if(HaveFileInDrive(fileName, foundIDs)) {
        wxMessageDialog msgDialog(this, wxT("Esse arquivo existe no Drive.\nDeseja sobrescrever?"), wxT("Atenção!"),
                                   wxYES_NO | wxICON_EXCLAMATION);
        switch(msgDialog.ShowModal()) {
            case wxID_YES: {
                wxBusyInfo busy(GetBusyInfoFlags(this, wxT("Substituindo arquivo no Drive...")));
                wxExecute(
                    wxString::Format("%s Replace \"%s\" \"%s\" \"%s\"", m_runDriveAPI,
                    foundIDs[0], fileName, path),
                    wxEXEC_SYNC | m_consoleStatus);
                return;
            } break;
            case wxID_NO: {
                return;
            } break;
        }
    } else {
        wxBusyInfo busy(GetBusyInfoFlags(this, wxT("Gravando arquivo no Drive...")));
        wxExecute(wxString::Format("%s Save \"%s\" \"%s\"", m_runDriveAPI,
        fileName, path),
            wxEXEC_SYNC | m_consoleStatus);
    }
}

```

```
}
```

```
bool MainFrame::HaveFileInDrive(wxString fileName, wxArrayString& fileIDs)
{
    wxBusyInfo busy(GetBusyInfoFlags(this, wxT("Procurando arquivo no Drive...")));
    wxExecute(wxString::Format("%s FindInFolder \"%s\"", m_runDriveAPI, fileName),
wxEXEC_SYNC | m_consoleStatus);
```

```
    wxTextFile txtFile;
    if(txtFile.Open("foundIDs.txt")) {
        fileIDs.Add(txtFile.GetFirstLine());
        while(!txtFile.Eof()) { fileIDs.Add(txtFile.GetNextLine()); }
```

```
        if(fileIDs[0] == "") {
            txtFile.Clear();
            txtFile.Write();
            return false;
```

```
        } else {
            txtFile.Clear();
            txtFile.Write();
            return true;
        }
    }
```

```
}
```

```
    return false;
```

```
}
```

```
wxBusyInfoFlags MainFrame::GetBusyInfoFlags(wxWindow* parent, wxString
message)
```

```
{
```

```
    return wxBusyInfoFlags()
        .Parent(parent)
        .Icon(m_iflcon)
```

```

        .Title(wxString::Format("<b>%s</b>", message))
        .Text(wxT("Aguarde um momento..."))
        .Foreground(*wxWHITE)
        .Background(*wxBLACK)
        .Transparency(4 * wxALPHA_OPAQUE / 5);
    }

void MainFrame::OnOpenDrive(wxCommandEvent& event)
{
    wxString* fileID = new wxString("");
    wxString* fileName = new wxString("");
    SearchInDrive driveDialog(this, fileID, fileName);
    if(driveDialog.ShowModal() == wxID_OK) {
        wxBusyInfo busy(GetBusyInfoFlags(this, wxT("Baixando processo...")));
        wxString path = "downloads\\" + *fileName + ".xml";
        wxExecute(wxString::Format("%s Download \"%s\" \"%s\"", m_runDriveAPI,
*fileID, path),
                wxEXEC_SYNC | m_consoleStatus);
        OpenProcess(path);
    }
}

void MainFrame::UpdateRowColour(Activity* activity, int rowNumber)
{
    wxColour rowColour = *wxBLACK;
    wxFont txtFont = m_grid->GetDefaultCellFont();
    if(!activity->IsShownInReport()) { txtFont.MakeItalic(); }
    if(activity->IsLimitedCH()) { rowColour = wxColour(255, 100, 0); }
    if(activity->IsRestrictionsIgnored()) { rowColour = wxColour(148, 0, 211); }
    if(activity->IsInvalidateCH()) { rowColour = wxColour(255, 0, 0); }
    wxColour disabledRowColour = rowColour;
    disabledRowColour.MakeDisabled();
    for(int i = 0; i < 9; ++i) {

```

```

        m_grid->SetCellTextColour(rowNumber, i, activity->IsShownInReport() ?
rowColour : disabledRowColour);
        m_grid->SetCellFont(
            rowNumber, i,
            activity->IsShownInReport() ? m_grid->GetDefaultCellFont() : m_grid-
>GetDefaultCellFont().MakeItalic());
    }
}

```

```

long MainFrame::ConvertPropToItemCode(const long& id, const Resolution&
resolution)
{
    if(resolution == N_16) {
        return id;
    } else if(resolution == N_20)
    {
        if(id < 9)
            return id;
        else if(id < 16) {
            return id + 1;
        }
        else if(id == 16)
            return 19; // Profesional Activity
        else if(id == 17)
            return 18;
    }
    return 0;
}

```

```

long MainFrame::ConvertItemCodeToProp(const long& id, const Resolution&
resolution)
{
    if(resolution == N_16) {
        return id;
    }
}

```



```

    } else if(resolution == N_20)
    {
        if(id < 9)
            return id;
        else if(id < 16) {
            return id - 1;
        }
        else if(id == 19)
            return 16; // Profesional Activity
        else if(id == 18)
            return 17;
    }
    return 0;
}

void MainFrame::OnChatButtonClick(wxCommandEvent& event)
{
    ChatForm ChatForm(this, m_richTextCtrlChat);
    ChatForm.SetSize(ChatForm.GetBestSize());
    if(ChatForm.ShowModal() == wxID_OK){

        // wxString msg = "";

        // wxMessageBox(msg);

        // msg += " " + ChatForm.GetChat();
    }
}

```

Segue abaixo o desenvolvimento do código Python:

```

from __future__ import print_function
import pickle
import os.path
import os
import io
import sys
from googleapiclient.discovery import build
from google_auth_oauthlib.flow import InstalledAppFlow
from google.auth.transport.requests import Request
from googleapiclient.http import MediaFileUpload, MediaIoBaseDownload
from apiclient import errors

# If modifying these scopes, delete the file token.pickle.
SCOPES = ['https://www.googleapis.com/auth/drive']
CAACFolder = '1ZC6gO1F0Jh0MhJhgtRdXarXB3P9H9fUg'

def auth(service):
    creds = None
    # The file token.pickle stores the user's access and refresh tokens, and is
    # created automatically when the authorization flow completes for the first
    # time.
    if os.path.exists('token.pickle'):
        with open('token.pickle', 'rb') as token:
            creds = pickle.load(token)
    # If there are no (valid) credentials available, let the user log in.
    if not creds or not creds.valid:
        if creds and creds.expired and creds.refresh_token:
            creds.refresh(Request())
        else:
            flow = InstalledAppFlow.from_client_secrets_file(
                'credentials.json', SCOPES)
            creds = flow.run_local_server(port=0)

```

```

# Save the credentials for the next run
with open('token.pickle', 'wb') as token:
    pickle.dump(creds, token)

```

```

service = build('drive', 'v3', credentials=creds)
return service

```

```

def listFiles(size, service):
    results = service.files().list(
        pageSize=size, fields="nextPageToken, files(id, name)").execute()
    items = results.get('files', [])
    if not items:
        print('No files found.')
    else:
        print('Files:')
        for item in items:
            print(u'{0} ({1})'.format(item['name'], item['id']))

```

```

def fileUpload(service, filename, filepath, mimetype, folderID):
    file_metadata = {'name': filename,
                     'parents': [folderID]}
    media = MediaFileUpload(filepath,
                             mimetype=mimetype)
    file = service.files().create(body=file_metadata,
                                  media_body=media,
                                  fields='id').execute()
    return file

```

```

def fileRemove(service, fileID):
    service.files().delete(fileId=fileID).execute()

```

```

def fileDownload(service, fileID, filepath, mimetype=None, folderID=None):
    request = service.files().get_media(fileId=fileID)
    fh = io.BytesIO()

```

```

downloader = MediaIoBaseDownload(fh, request)
done = False
while done is False:
    status, done = downloader.next_chunk()
    print('Download {}%.'.format(int(status.progress() * 100)))
with io.open(filepath, 'wb') as f:
    fh.seek(0)
    f.write(fh.read())

def searchFiles(size, service, filename, ids):
    results = service.files().list(pageSize=size, fields="nextPageToken, files(id, name)",
                                   q="name contains '{}' and trashed =
false".format(filename)).execute()
    items = results.get('files', [])

    if not items:
        print('Não foram encontrados arquivos.')
    else:
        for item in items:
            ids.append(item['id'] + '\n')
        print('IDs dos arquivos gravados em foundIDs.txt')

def listFilesInFolder(service, folderID, ids):
    results = service.files().list(fields="nextPageToken, files(id, name)",
                                   q="{}' in parents and trashed =
false".format(folderID)).execute()
    items = results.get('files', [])

    if not items:
        print('Não foram encontrados arquivos.')
    else:
        for item in items:
            ids.append(item['id'] + '\n')

def searchFilesInFolder(service, folderID, filename, filenameOpt, ids, names):

```

```

results = service.files().list(fields="nextPageToken, files(id, name)",
                                q="{}' in parents and (name contains '{} or name contains '{}')
and trashed = false".format(folderID, filename, filenameOpt)).execute()
items = results.get('files', [])

if not items:
    print('Não foram encontrados arquivos.')
else:
    for item in items:
        ids.append(item['id'] + '\n')
        names.append(item['name'] + '\n')

def main():
    print('Conectando ao Google Drive...')
    service = None
    service = auth(service)
    print('Conexão realizada com sucesso!')

if(len(sys.argv) > 1):
    if sys.argv[1].casefold() == 'firstun':
        print('Google drive autorizado!')
    elif sys.argv[1].casefold() == 'save':
        print('Salvando arquivo no drive')
        fileUpload(service, sys.argv[2],
                    sys.argv[3], 'text/xml', CAACFolder)
        print('Arquivo gravado com sucesso!')
    elif sys.argv[1].casefold() == 'find':
        print('Buscando arquivos no drive...')
        ids = []
        searchFiles(100, service, sys.argv[2], ids)
        # save txt file
        txtFile = open('foundIDs.txt', 'w')
        txtFile.writelines(ids)
        txtFile.close()

```

```

elif sys.argv[1].casefold() == 'listinfolder':
    print('Buscando arquivos na pasta AvaliadorCAAC...')
    ids = []
    listFilesInFolder(service, CAACFolder, ids)
    # save txt file
    txtFile = open('foundIDs.txt', 'w')
    txtFile.writelines(ids)
    txtFile.close()
elif sys.argv[1].casefold() == 'findinfolder':
    print('Buscando arquivos na pasta AvaliadorCAAC...')
    ids = []
    names = []
    searchFilesInFolder(service, CAACFolder, sys.argv[2], "", ids, names)
    # save txt files
    txtFile = open('foundIDs.txt', 'w')
    txtFile.writelines(ids)
    txtFile.close()
elif sys.argv[1].casefold() == 'findinfolder2':
    print('Buscando arquivos na pasta AvaliadorCAAC...')
    ids = []
    names = []
    searchFilesInFolder(service, CAACFolder, sys.argv[2], sys.argv[3], ids,
names)
    # save txt files
    txtFile = open('foundIDs.txt', 'w')
    txtFile.writelines(ids)
    txtFile.close()
    txtFile = open('foundNames.txt', 'w')
    txtFile.writelines(names)
    txtFile.close()
elif sys.argv[1].casefold() == 'delete':
    print('Removendo arquivo...')
    fileRemove(service, sys.argv[2])
elif sys.argv[1].casefold() == 'replace':

```

```
    print('Removendo arquivo anterior...')
    fileRemove(service, sys.argv[2])
    print('Salvando arquivo no drive')
    fileUpload(service, sys.argv[3],
               sys.argv[4], 'text/xml', CAACFolder)
    print('Arquivo gravado com sucesso!')
elif sys.argv[1].casefold() == 'download':
    fileDownload(service, sys.argv[2], sys.argv[3])
return 1

if __name__ == "__main__":
    main()
```