

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
UNIVERSIDADE FEDERAL DE VIÇOSA - **CAMPUS FLORESTAL**
INSTITUTO DE CIÊNCIAS EXATAS E TECNOLÓGICAS¹
CURSO TÉCNICO EM INFORMÁTICA

Squink

Deivid Felipe dos Santos Sales
Lara Geovana Almeida Melo
Mateus Mendes Alves Cabral
Pedro Lucas Miranda Rufino
Pedro Lucas Paulino

Maria Amélia Lopes Silva

Este relatório faz parte dos requisitos da disciplina de Projetos II, como projeto de integração da prática profissional em forma de Trabalho de Conclusão do Curso Técnico em Informática.

SUMÁRIO

RESUMO DO TRABALHO	3
ABSTRACT	4
1 - INTRODUÇÃO	5
2 - REVISÃO TEÓRICA	5
3 – METODOLOGIA / MATERIAL E MÉTODO	6
3.1 Diagrama Entidade Relacionamento	7
4 – DESENVOLVIMENTO / RESULTADO E DISCUSSÃO	8
4.1 Requisitos funcionais	8
4.2 Requisitos não funcionais	9
4.3 Regras de negócio	11
5 - TELAS	12
5.1 Tela Inicial	12
5.2 Tela de “Inscrever-se”	13
5.3 Tela de “Login”	14
5.5 Tela de operações	15
6 - DIAGRAMAS	16
6.1 Diagrama de casos de uso	16
6.2 Diagrama de classes	17
7 - CÓDIGOS	17
8.1 Executando o Squink	18
8. 2 Configurações do Pycharm	19
8.3 Sincronização	19
8.4 Saindo o aplicativo	20
8.5 Ao abrir novamente o aplicativo	20
8 – REFERÊNCIAS	22

RESUMO DO TRABALHO

A configuração de um software normalmente é a primeira parte a ser realizada para que seja possível fazer o uso correto de um aplicativo. Entretanto, as configurações minuciosas demandam tempo para que o App fique do seu jeito, esse fato é prejudicial quando se há necessidade de desinstalar e instalar novamente um software, visto que é necessário configurar tudo novamente. Com nosso software pretendemos reverter esse cenário, aqueles que desejam ser produtivos e ainda assim trabalhar de forma a não perder o conforto de suas configurações poderão fazê-lo. Com essa ideia, planejamos poupar o tempo de nosso usuário que é gasto configurando aplicativos complexos. Expressamos nosso intuito através de nosso lema “*configured once, use forever*”.

palavras chave: Configurações, Aplicativos, Java

ABSTRACT

Software configuration is usually the first part that needs to be done in order to make correct use of an application. However, detailed configurations take time for the App to get your way, this fact is detrimental when there is a need to uninstall and reinstall a software, since it is necessary to configure everything again. With our software we intend to reverse this scenario, those who wish to be productive and still work in a way that does not lose the comfort of their settings will be able to do so. With this idea, we plan to save our user's time that is spent configuring complex applications. We express our intention through our motto "configured once, use forever".

Keywords: Settings, Applications, Java

1 - INTRODUÇÃO

Atualmente, um usuário de desktop utiliza uma gama de softwares em seu dia, esses que são devidamente personalizados para o gosto do usuário. Entretanto, devido a motivos alheios, caso o usuário precise desinstalar o software de seu computador, ao instalar novamente, se o programa não salvar em nuvem as configurações do usuário, suas configurações serão perdidas. Para tanto, ao visualizar esse problema que persiste entre os usuários de diversos softwares, resolvemos criar nosso software, Squink, que tem como objetivo salvar em nuvem as configurações feitas por um usuário de um aplicativo qualquer. Possibilitando, assim, que o usuário que por ventura tenha a necessidade de desinstalar o aplicativo, ao instalá-lo novamente, o mesmo possa em poucos clicks retomar as configurações antes .

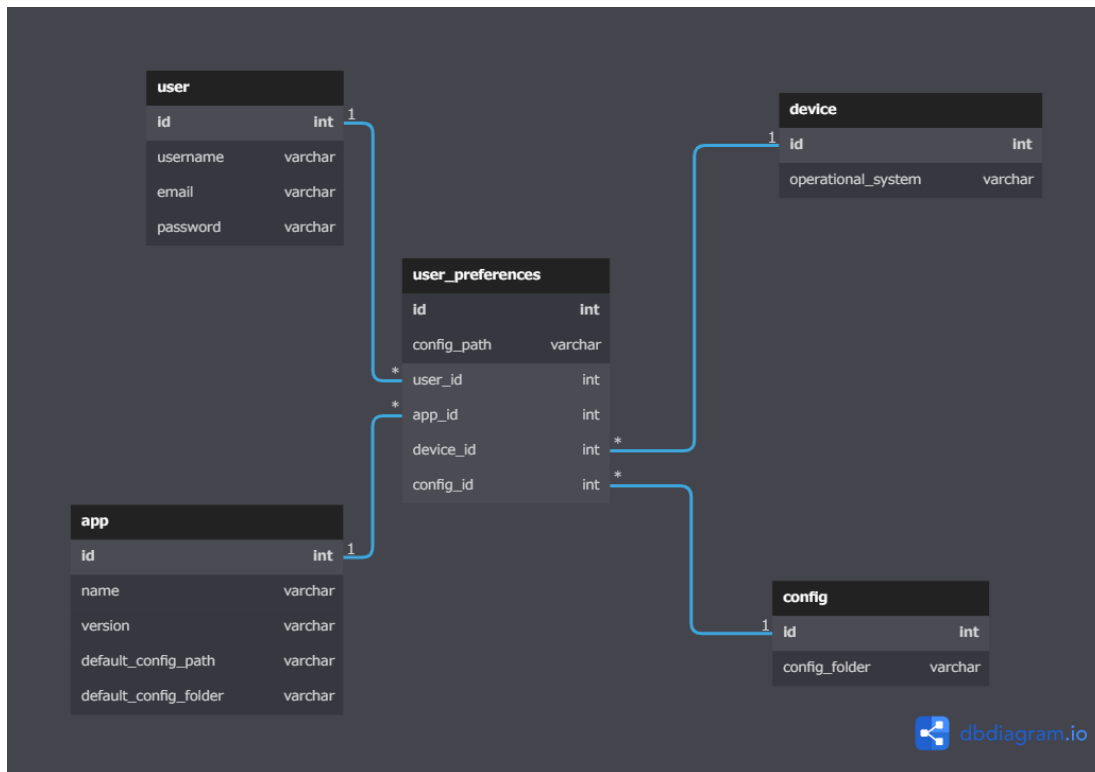
2 - REVISÃO TEÓRICA

Um dos principais problemas que encontramos hoje ao fazer a manutenção de um computador que precise ser formatado é perder todas as informações. Os arquivos de texto, imagens entre outros, são passíveis de restauração, através do backup, entretanto, com aplicativos, tais como: Pycharm, NetBeans, não há a possibilidade de se fazer o backup sem que haja o auxílio de um aplicativo como o Squink. Esse problema foi visualizado por nossa equipe que procurou resolvê-lo. Apesar dessa mazela ser recorrente a todos os usuários de diversos aplicativos, não encontramos trabalhos que realizassem a mesma função do Squink, entretanto, existem diversos softwares de qualidade que apresentam um sistema parecido, mas salvam somente as configurações de seus usuários. Podemos citar o Pacote da adobe, que são um conjunto de softwares focados em edição de imagens e vídeos, o pacote é constituído pelos seguintes softwares: Acrobat Pro, Photoshop, InDesign entre outros, em que esses softwares podem ser desinstalados do computador, entretanto ao se instalar novamente, se logado com a mesma conta, é possível encontrar suas próprias configurações e também os arquivos anteriormente editados. Além desse, outro software que apresenta uma funcionalidade parecida ao que o Squink procura solucionar, são os jogos da empresa Riot Games, em que todos seus jogos, como por exemplo, Valorant, League of Legends, possuem as configurações de jogo salvas e podem ser acessadas de qualquer lugar em qualquer máquina, claro, se o usuário faça o login com suas credenciais.

3 – METODOLOGIA / MATERIAL E MÉTODO

O Squink precisa ser rápido, simples e conciso, por isso a linguagem de programação Java foi utilizada no desenvolvimento do aplicativo, juntamente com o uso de ferramentas como o Scene Builder e banco de dados. O principal problema a ser resolvido foi localizar a pasta com informações, em nosso trabalho optamos por focar apenas nas informações de configurações do software Pycharm, copiar suas configurações e enviar para a nuvem. Esse problema foi sanado com diversas pesquisas que nos levou a utilizar algumas bibliotecas como "Java.io.InputStream, Java.io.OutputStream" que tem como função ler os dados, já o OutputStream tem como finalidade escrever os dados em outro local. A partir disso conseguimos copiar o arquivo do diretório padrão e enviar para a nuvem, como nos foi orientado utilizamos o google drive. Com esse processo em funcionamento realizamos o processo reverso para salvar as configurações novamente no aplicativo.

3.1 Diagrama Entidade Relacionamento



A partir do banco de dados descrito pelo Diagrama ER acima foi possível interligar todas as peças-chave necessárias para o armazenamento da informação desejada. A partir do diagrama é possível visualizar como o software funcionará, por exemplo, seja a pasta de configuração do aplicativo Pycharm, esse se localiza no diretório C:\ProgramData, a partir disso podemos relacionar as informações do app ao usuário. .

Em teoria, ao usar o Squink pela primeira vez, deveria oferecer ao usuário uma lista com todos os programas instalados, porém, como já dito, optamos em focar somente no Pycharm, portanto somente esse aparecerá para o usuário, após selecionar o aplicativo deseja salvar. O próprio sistema fornece o diretório do programa selecionado pelo usuário, a partir disso será feito um processo de tratamento, a pasta de configuração será enviada para a nuvem. Quando necessário restaurar as configurações armazenadas, o Squink vai substituir a pasta que se encontra no computador pela aquela salva na nuvem.

- Tecnologias utilizadas
 - Linguagem de programação Java.
 - Scene Builder.
 - Banco de dados SQL
 -

4 – DESENVOLVIMENTO / RESULTADO E DISCUSSÃO

4.1 Requisitos funcionais

Os requisitos funcionais são aqueles que descrevem o comportamento do sistema, suas ações para cada entrada, ou seja, é aquilo que descreve o que tem que ser feito pelo sistema. Descrevem as funcionalidades que o sistema deve dispor.

Identificador	Descrição	Prioridade
RQ001	O sistema deve vasculhar os diretórios dos programas, selecionados pelo usuário, em busca dos arquivos de configuração.	Alta
RQ002	Deve ser feita a análise dos dados. Verificando a presença de Malware, arquivos corrompidos, tentativa de sql injection ou configurações equivocadas.	Alta
RQ003	Os dados coletados pelo sistema deverão ser armazenados em 'data base'. Fazendo a sincronização de dados para o usuário no site do sistema.	Alta

4.2 Requisitos não funcionais

Os requisitos não funcionais são aqueles que expressam como deve ser feito. Em geral se relacionam com padrões de qualidade como confiabilidade, performance, robustez, etc. São muito importantes, pois definem se o sistema será eficiente para a tarefa que se propõe a fazer ou não. Um sistema ineficiente certamente não será usado. Neles também são apresentados restrições e especificações de uso para os requisitos funcionais.

Identificador	Descrição	Categoria
RNF001	Caso o software não encontre uma pasta específica de um programa em um tempo (x) deve devolver uma mensagem de ("arquivo não encontrado")	Usabilidade
RNF002	O software deve mostrar a frequência com a qual não se realizou a tarefa	Usabilidade
RNF003	O software deve exigir o mínimo de ações humanas	Usabilidade
RNF004	. O software deve ter seu código, comentado em todas as ações.	Manutenibilidade
RNF005	O software deve seguir um padrão na criação de variáveis	Manutenibilidade
RNF006	. O software deve estar sempre refatorado.	Manutenibilidade
RNF007	Caso o software por algum erro "trave" este deve após um tempo (x) o software deve fechar-se.	Confiabilidade

RNF008	O software não salvará arquivos que sejam interrompidos no meio da alocação no Banco de dados.	Confiabilidade
RNF009	O software será compatível com o sistema operacional Windows 10 64X.	Portabilidade
RNF0010	O software poderá ser instalado em outra maquina e quando logado o usuário poderá Buscar todos os arquivos já salvos anteriormente.	Portabilidade
RNF0011	O usuário só poderá ter acesso a seus arquivos quando o login seja feito previamente e seu usuário e senha só poderão ser restaurados através de um código de acesso que o software gerará no primeiro acesso.	Segurança

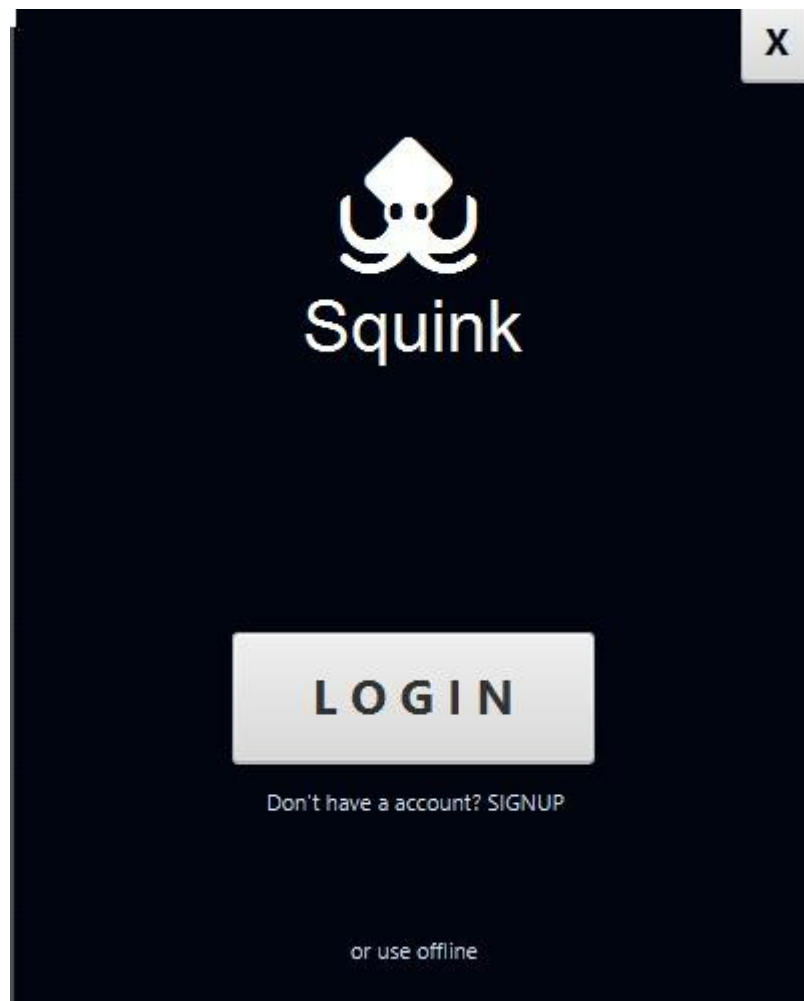
4.3 Regras de negócio

Declarações sobre políticas ou condições que devem ser satisfeitas. Regra de negócio é uma restrição imposta pelo negócio que regulamenta o comportamento de um procedimento operacional do negócio

Identificador	Descrição	Prioridade
RN001	Ao logar em um novo dispositivo, o programa instala todas as configurações salvas na nuvem automaticamente.	Alta
RN002	Por padrão, o programa vem com o tema escuro ativo.	Baixa
RN003	Salvar as configurações de um programa, não exige plano pago. Para até 3 configurações exige plano Premium. Para mais de 3 programas e/ou uso simultâneo em mais de um dispositivo, exige plano corporativo.	Alta
RN004	Se algum problema for diagnosticado durante a verificação periódica, o programa notifica o usuário através do email cadastrado.	Média

5 - TELAS

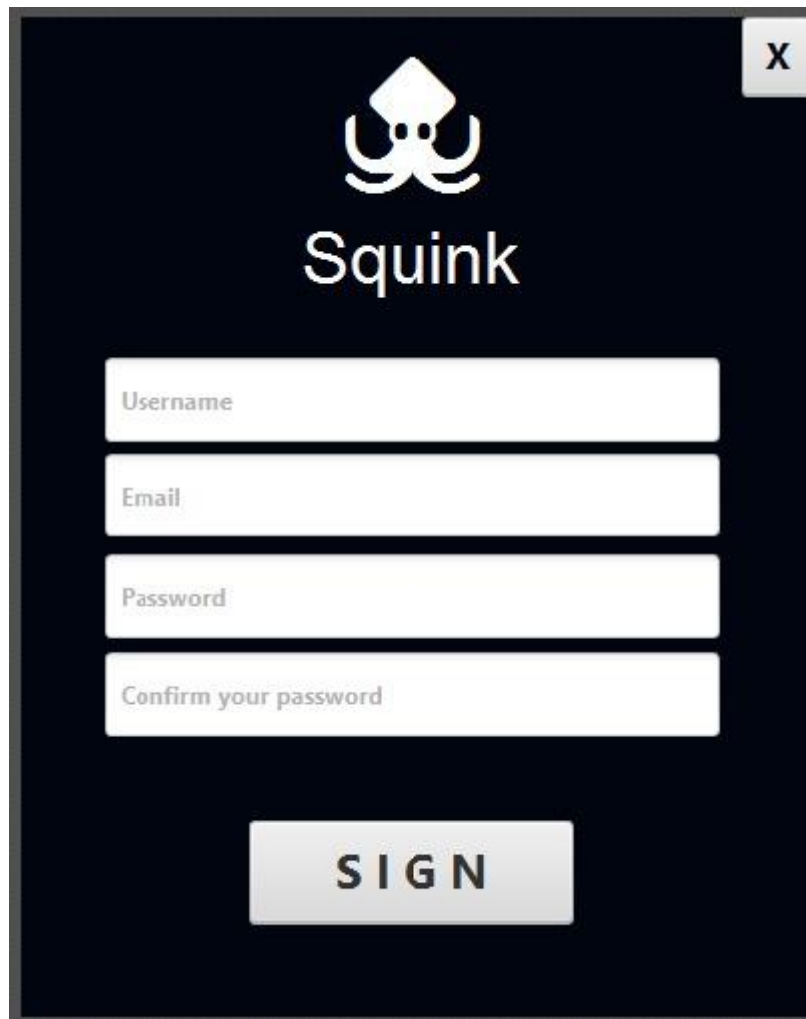
5.1 Tela Inicial



Ao abrir nosso software o usuário se depara com três opções para o uso de nosso aplicativo, sendo duas delas funcionais e uma representativa que será aplicada no futuro.

1. Login: Aquele usuário que já é cadastrado pode realizar o login.
2. Signup: Aquele que ainda não é cadastrado será redirecionado para a página de Cadastro.
3. Offline: Futuramente pretendemos implantar a funcionalidade de utilização offline que se baseia na ideia baixar as configurações que existem em um aplicativo, envia-lá para uma mídia externa para que seja possível aplicar em outras máquinas.

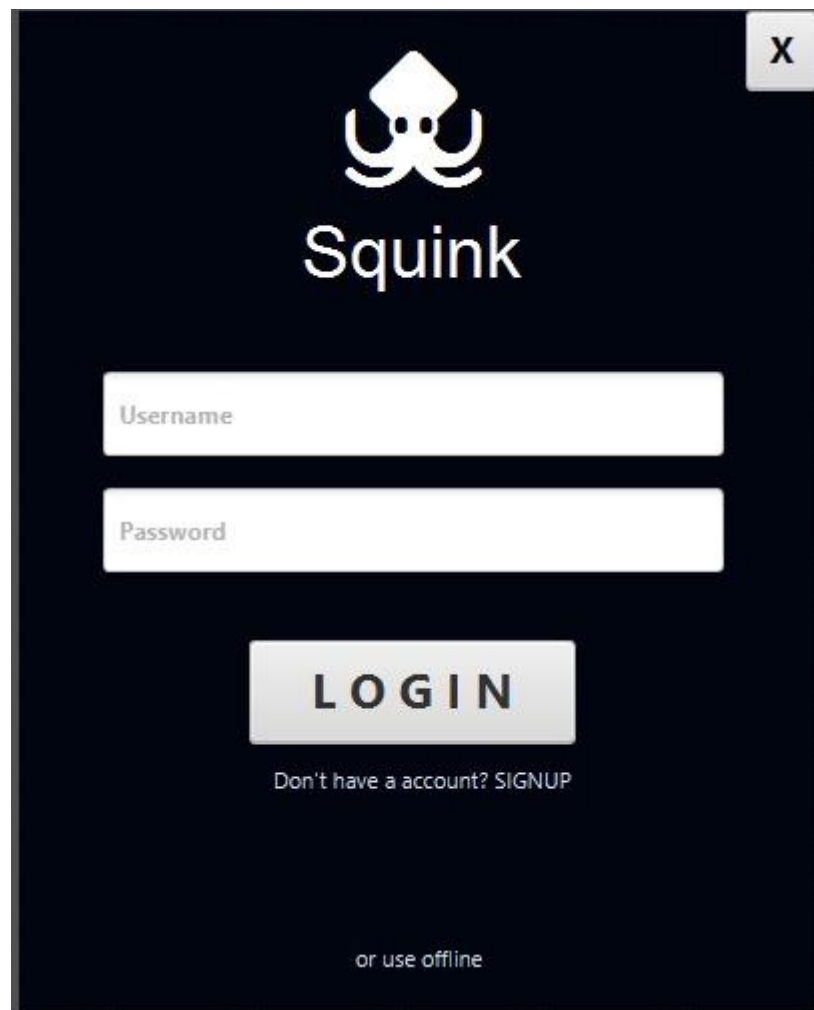
5.2 Tela de “Inscrever-se”




The image shows a registration form for a service called "Squink". The form is displayed in a dark-themed window with a close button (X) in the top right corner. At the top center is the Squink logo, which consists of a stylized octopus head. Below the logo, the word "Squink" is written in a large, white, sans-serif font. The form contains four input fields, each with a light gray placeholder text: "Username", "Email", "Password", and "Confirm your password". These fields are stacked vertically. At the bottom center of the form is a large, rectangular button with a gradient from light gray to white, containing the text "SIGN" in a bold, black, sans-serif font.

A tela de cadastro vai salvar as informações do nosso usuário em nosso banco de dados, assim permitindo que o mesmo consiga acessar todas as funcionalidades do programa.

5.3 Tela de “Login”



X



Squink

Username

Password

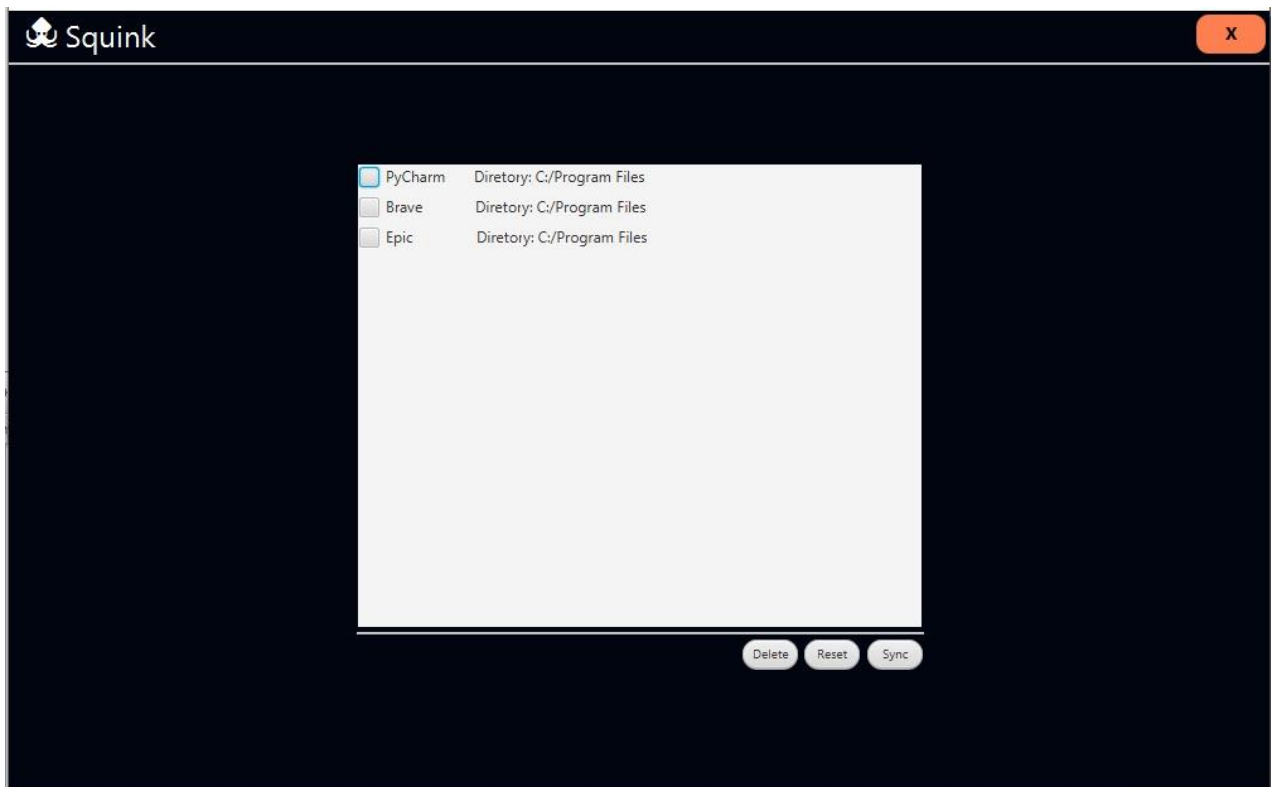
LOGIN

Don't have a account? [SIGNUP](#)

or use offline

Nessa página o usuário que já se cadastrou poderá realizar o login, assim entrando em sua conta

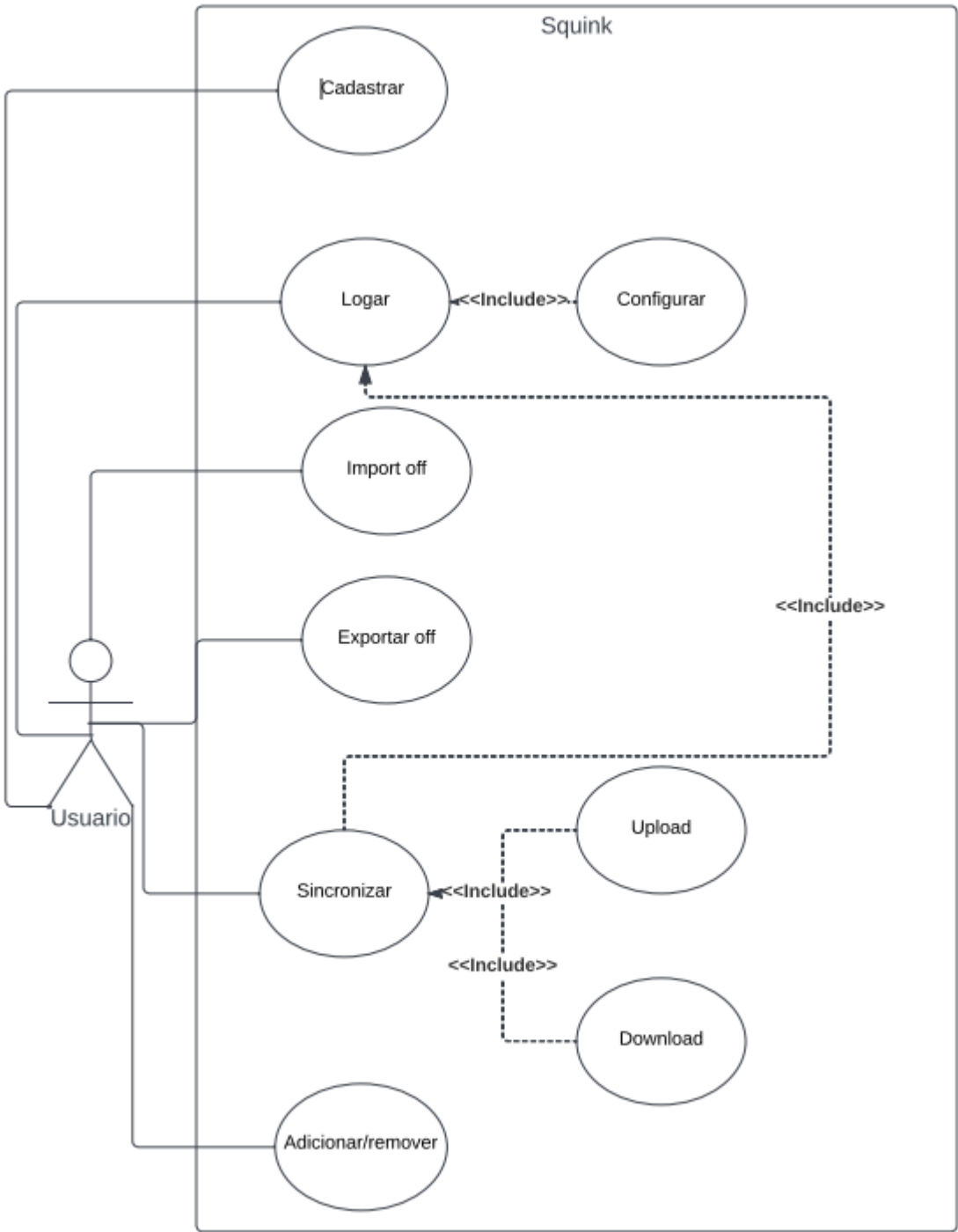
5.5 Tela de operações



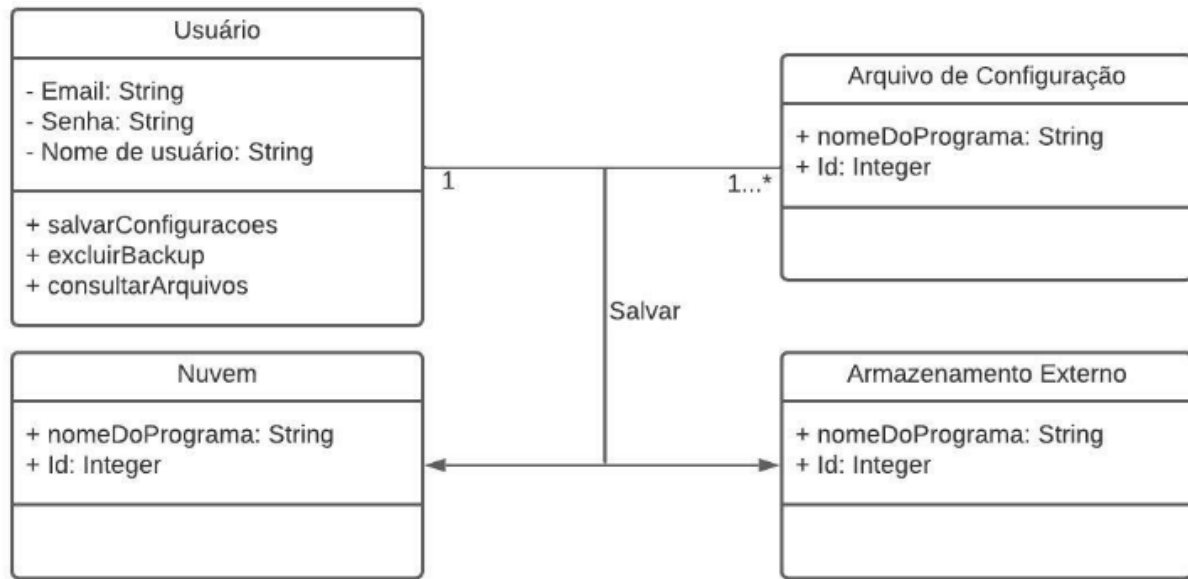
Pode-se considerar esta página como a principal de nosso programa, pois será nela que o usuário vai escolher quais serão os arquivos que serão salvos na nuvem. Em nosso trabalho usamos o software Pycharm como exemplo, em que após selecionar o aplicativo é possível deletá-lo da nuvem, resetá-lo, ou seja, redefinir as configurações para originais do programa e por fim sincronizá-lo, cerne de nosso programa, que busca sincronizar as informações que temos de dado programa e substituir na pasta de configuração.

6 - DIAGRAMAS

6.1 Diagrama de casos de uso



6.2 Diagrama de classes



7 - CÓDIGOS

Em nosso trabalho buscamos construir um código com poucas linhas, porém com grande eficiência, para isso buscamos implementar bibliotecas que contribuíssem para que nosso propósito se concretizasse. Nesse âmbito trouxemos o seguinte método que só é possível por intermédio das bibliotecas `Java.io.InputStream`, `Java.io.OutputStream`.

Nesse trecho de nosso código utilizamos o método `copyDir` que tem como finalidade criar a pasta em que os arquivos de configuração serão armazenados.

```
public static void copyDir(Path src, Path dest) throws IOException {
    Files.walk(src)
        .forEach(source -> {
            try {
                Files.copy(source, dest.resolve(src.relativize(source)),
                    StandardCopyOption.REPLACE_EXISTING);
            } catch (IOException e) {
                e.printStackTrace();
            }
        });
}
```

Nesse trecho os arquivos são direcionados da pasta de origem para a nova.

```
File from = new File("C:\\Users\\ragan\\AppData\\Roaming\\JetBrains");
File to = new File("G:\\Meu Drive\\Squink\\JetBrains");
```

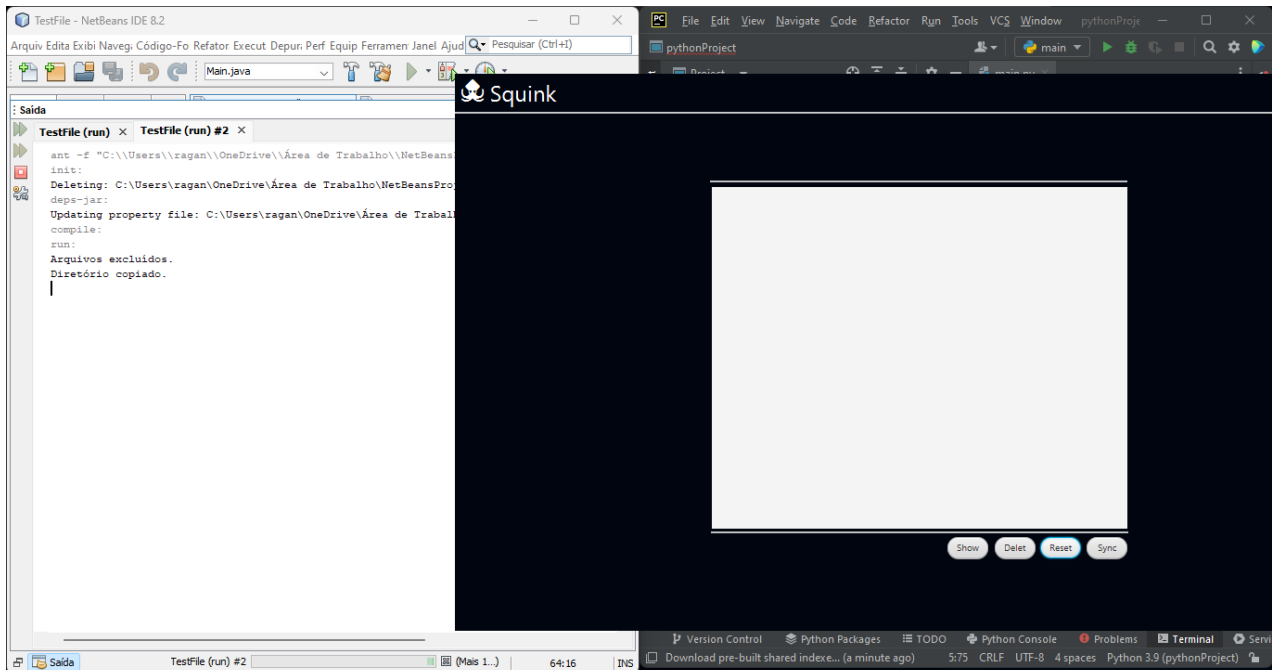
O método é aplicado

```
copyDir(from.toPath(), to.toPath());
```

8 - EXECUTANDO O CÓDIGO

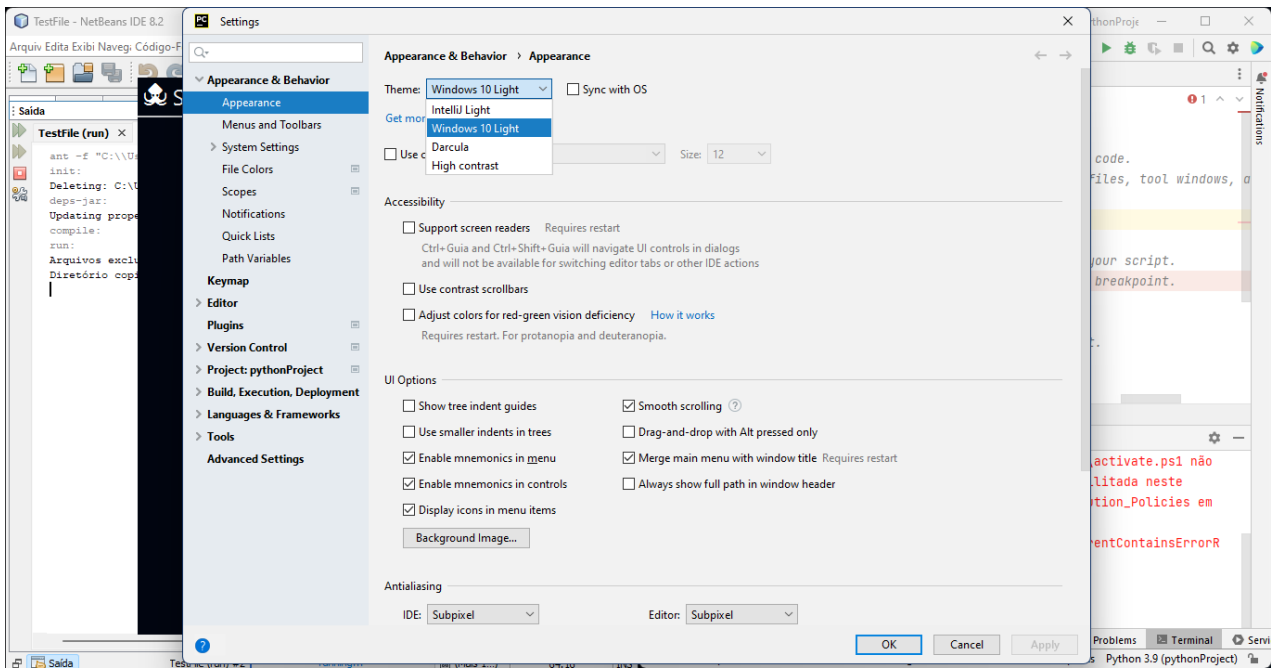
Nesse momento pretendemos apresentar nosso programa em execução, para tanto utilizaremos o programa Pycharm como exemplo.

8.1 Executando o Squink



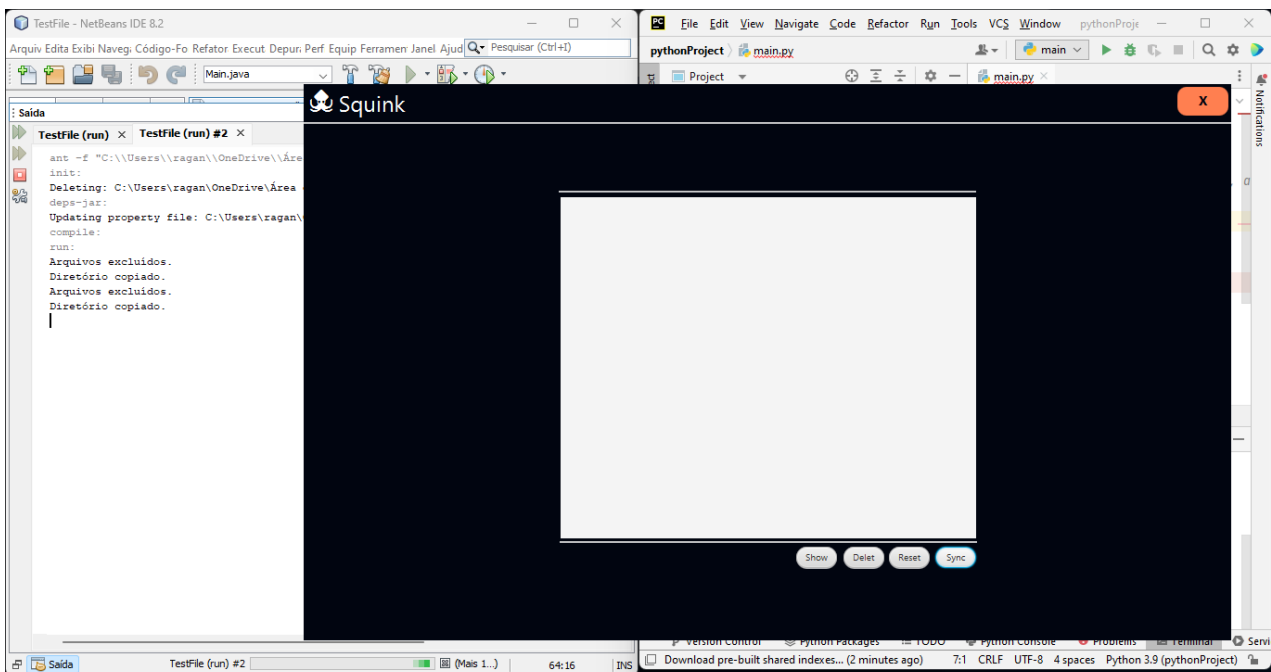
Ao executar o programa e realizar o login seremos redirecionados para a tela de escolha dos softwares que realizaremos a sincronização das configurações com a nuvem, em nosso caso o Pycharm.

8.2 Configurações do Pycharm



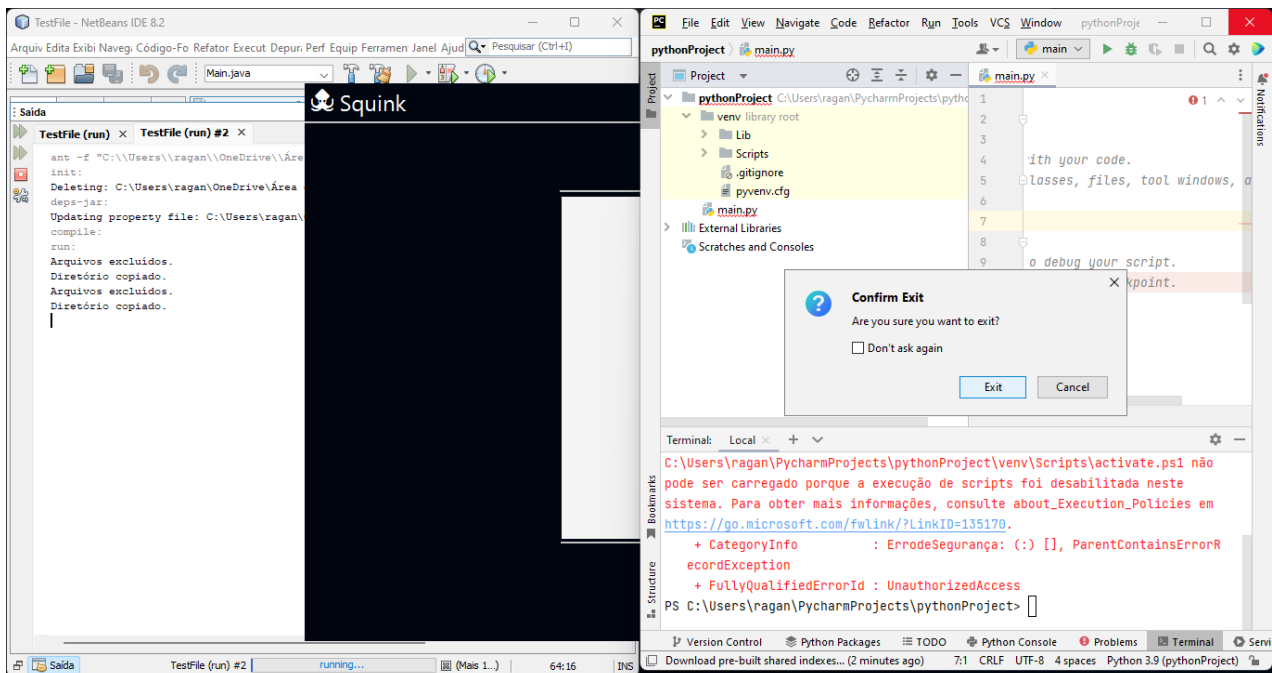
Para exemplificar a utilização do Squink, vamos substituir as configurações do usuário, que já foi sincronizada, por uma pré definida “Windows 10 Light”.

8.3 Sincronização



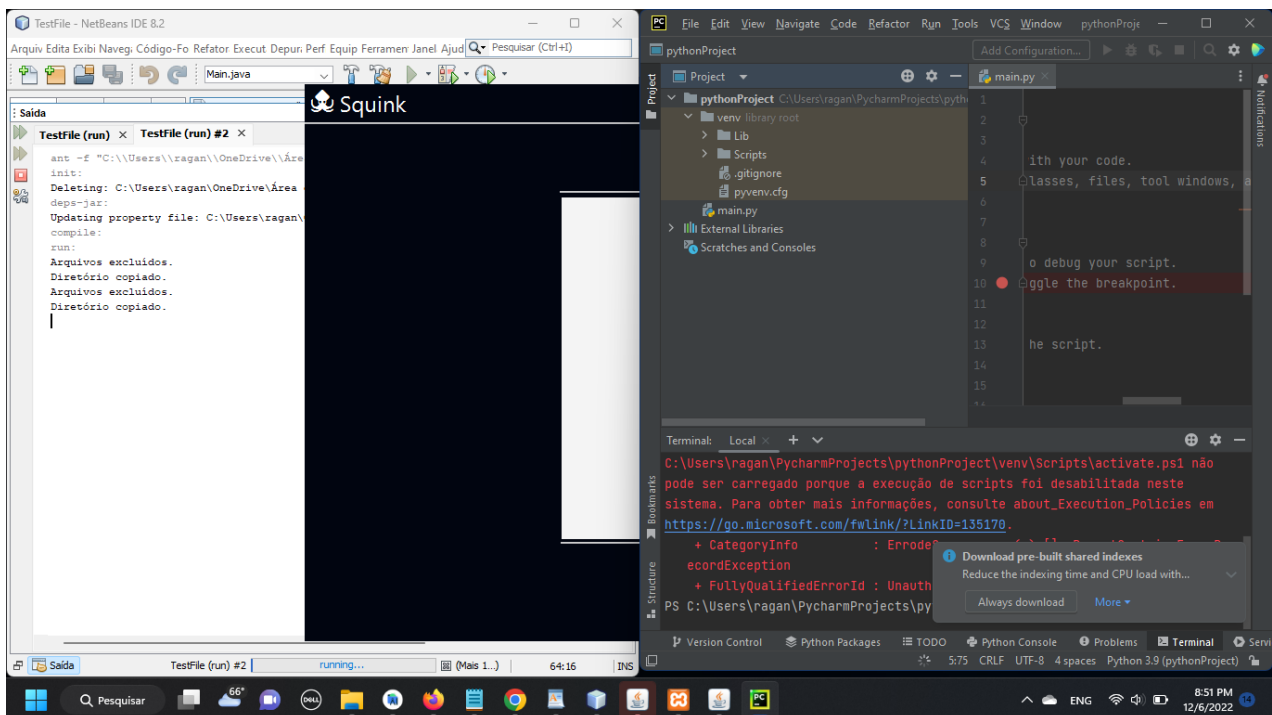
Após a mudança no Pycharm, para voltarmos às configurações anteriores devemos clicar sobre o botão “Sync”, assim o usuário sincronizará as configurações escolhidas anteriormente com o pycharm, ou seja, substituiremos a configuração pré definida “Windows 10 Light” pela configuração que o usuário havia sincronizado anteriormente.

8.4 Saindo o aplicativo



Após a sincronização podemos fechar o software pois o processo já se concluiu.

8.5 Ao abrir novamente o aplicativo



Após abrir novamente o PyCharm podemos ver que as configurações já foram substituídas, mostrando, dessa maneira, a eficácia do Squink.

7 – CONSIDERAÇÕES FINAIS

Apesar de nosso software ir além do que nos foi apresentado no curso de informática. Acreditamos que nosso trabalho tenha suprido nossas expectativas, visto que apesar de existirem diversos problemas, tal como a dificuldade em compactar a pasta de configurações, conseguimos contornar tal problema através de um loop que lê todos os arquivos da pasta de configurações, copiando e os enviando para a nuvem. Assim, sendo possível realizar nosso propósito, ou seja, possibilitar que os usuários do Squink possam usufruir dessa ferramenta otimizando seu tempo e também facilitando o processo de configuração do software. Além disso, o desenvolvimento do Squink, contribuiu para nosso engrandecimento pessoal, nos ajudando a trabalhar em equipe e aprender mais sobre o vasto universo da programação.

8 – REFERÊNCIAS

Curso de UML - Diagrama de Casos de Uso - Exemplo Básico - YouTube. Disponível em:

<<https://www.youtube.com/watch?v=tezLX9quOVc>>. Acesso em: 28 jun. 2022.

Curso de UML - O que são Diagramas de Casos de Uso - YouTube. Disponível em:

<<https://www.youtube.com/watch?v=K-BaRfFx0mA>>. Acesso em: 28 jun. 2022.

Diagrama de Pacotes. Disponível em:

<https://homepages.dcc.ufmg.br/~amendes/GlossarioUML/glossario/conteudo/pacotes/diagrama_de_pacotes.htm>. Acesso em: 28 jun. 2022.

Diagrama de classes (UML): Orientações básicas na elaboração. Disponível em:

<<https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama-de-classes/37224>>. Acesso em: 28 jun. 2022.

Diagramas de colaboração UML, exemplos gratuitos. Disponível em:

<<https://www.edrawsoft.com/pt/uml-collaboration.html>>. Acesso em: 28 jun. 2022.

MER e DER: Modelagem de Bancos de Dados. Disponível em:

<<https://www.devmedia.com.br/mer-e-der-modelagem-de-bancos-de-dados/14332>>. Acesso em: 28 jun. 2022.

Produtos da Adobe: aplicativos para desktop, Web e dispositivos móveis | Adobe. Disponível

em: <<https://www.adobe.com/br/products/catalog.html>>. Acesso em: 28 jun. 2022.

Software online de diagramas e comunicação visual | Lucidchart. Disponível em:

<https://www.lucidchart.com/pages/pt/landing?utm_source=google&utm_medium=cpc&utm_campaign=_chart_pt_allcountries_mixed_search_brand_exact_&km_CPC_CampaignId=1500131167&km_CPC_AdGroupId=59412157138&km_CPC_Keyword=lucidchart&km_CPC_MatchType=e&km_CPC_ExtensionID=&km_CPC_Network=g&km_CPC_AdPosition=&km_CPC_Creative=294337318298&km_CPC_TargetID=kwd-33511936169&km_CPC_Country=1031619&km_CPC_Device=m&km_CPC_placement=&km_CPC_target=&gclid=CjwKCAjwzeqVBhAoEiwAOrEmzeHddmMOc04U_KkxilxKMrlNAbrh7yvJ_5Z7f_A8Kt9D6vyi-3r-bBoCYn8QAvD_BwE>.

Acesso em: 28 jun. 2022.

dbdiagram.io - Database Relationship Diagrams Design Tool.

Disponível em: <<https://dbdiagram.io/home>>. Acesso em: 28 jun. 2022.

ANEXOS

Caso seja opção do aluno, no anexo podem conter figuras /imagens relativas ao desenvolvimento do trabalho, como telas, normas técnicas, fotos, e demais informações sobre o trabalho. Estas informações em forma de anexo permite que o texto da metodologia não fique “quebrado” pelo excesso de figuras contidas nela. Assim , se forem utilizadas muitas informações na seção de metodologia, uma boa opção é transportá-las para a seção de ANEXOS.

Assinatura do(a) Aluno(a)