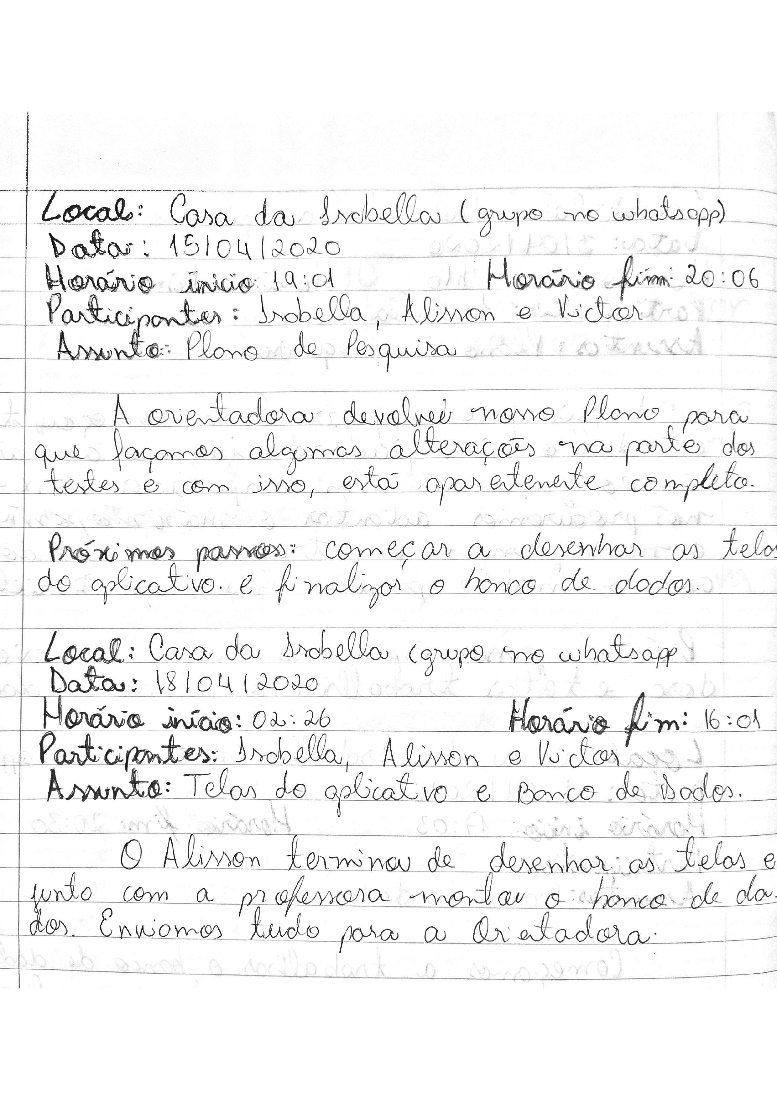
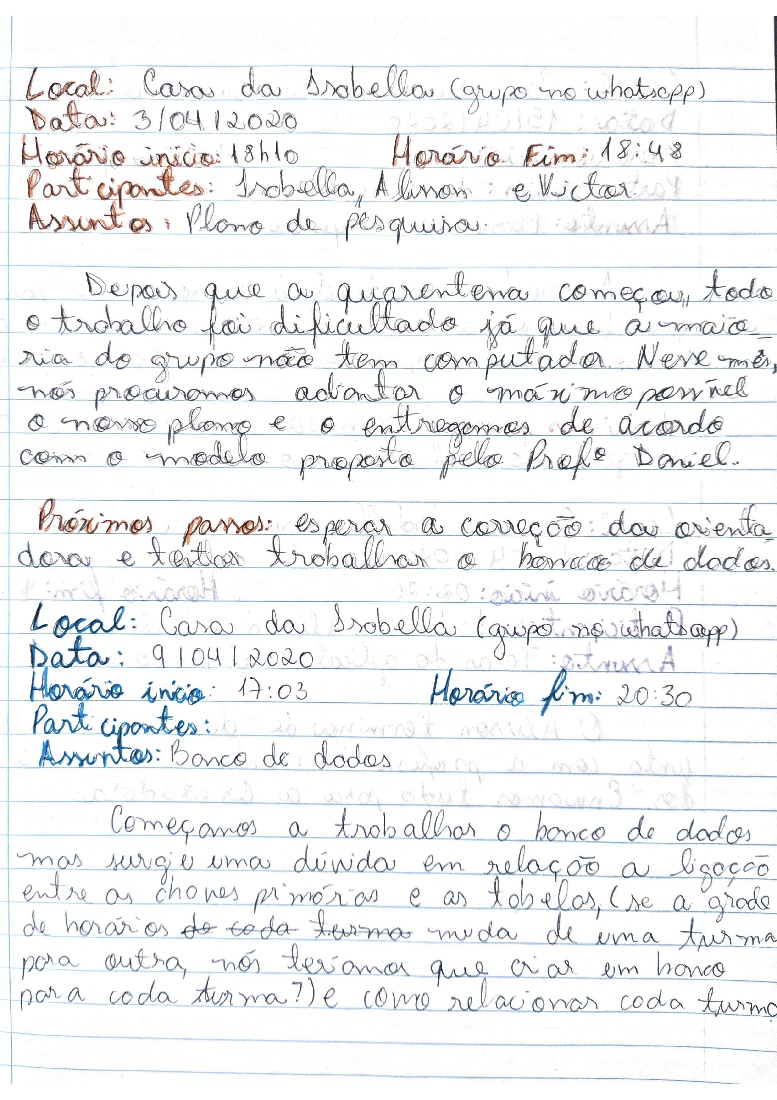
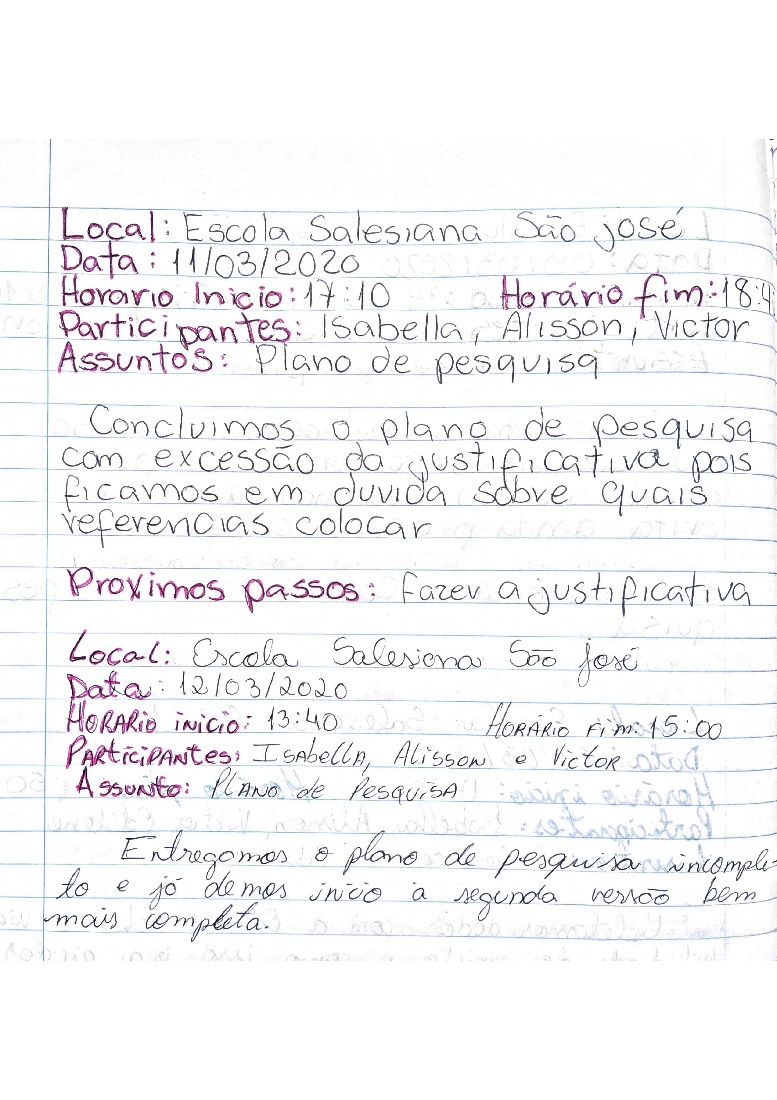
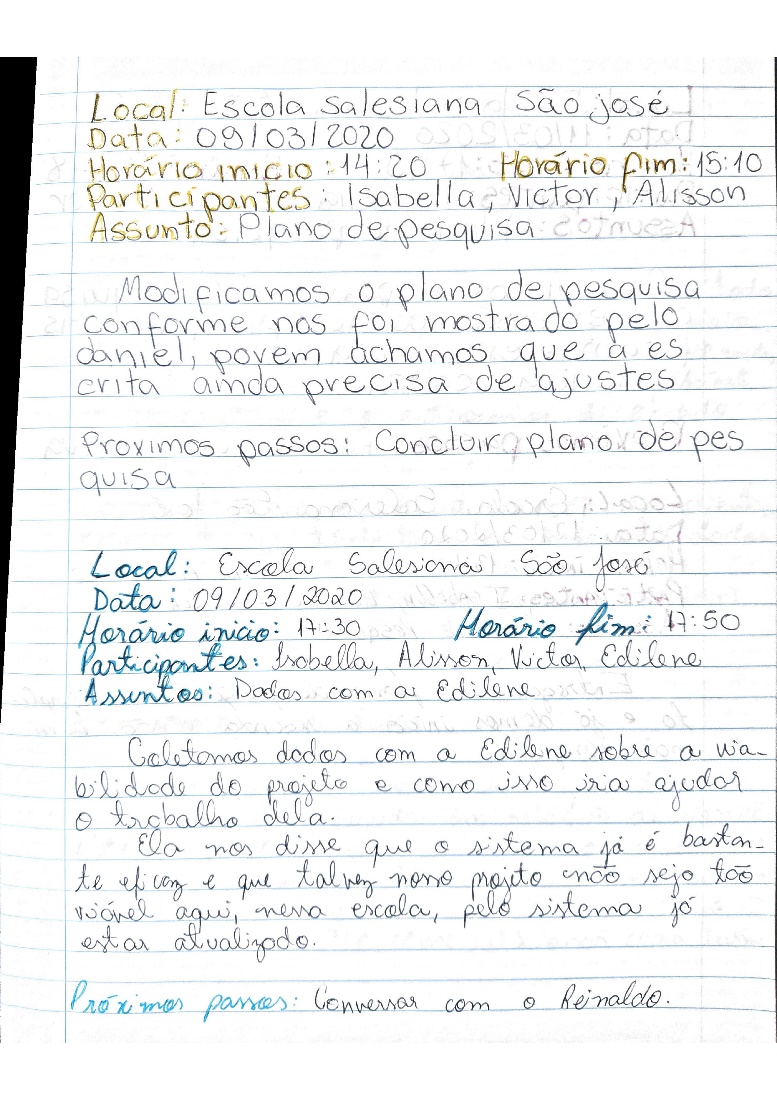
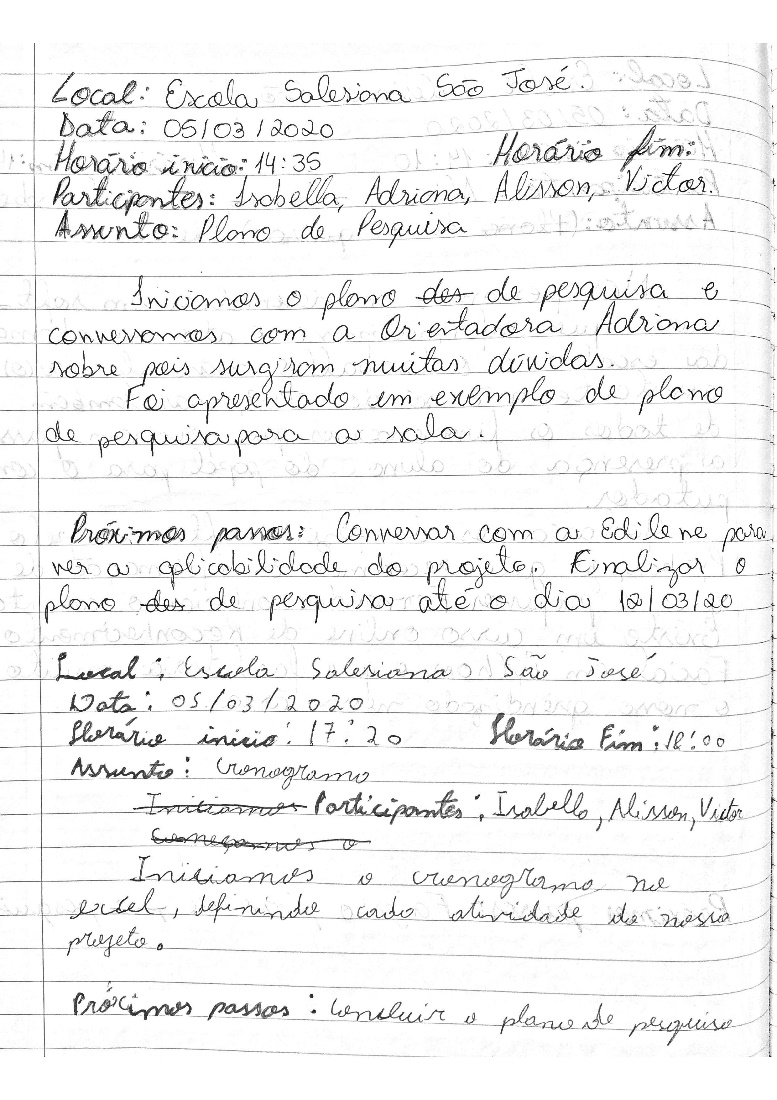
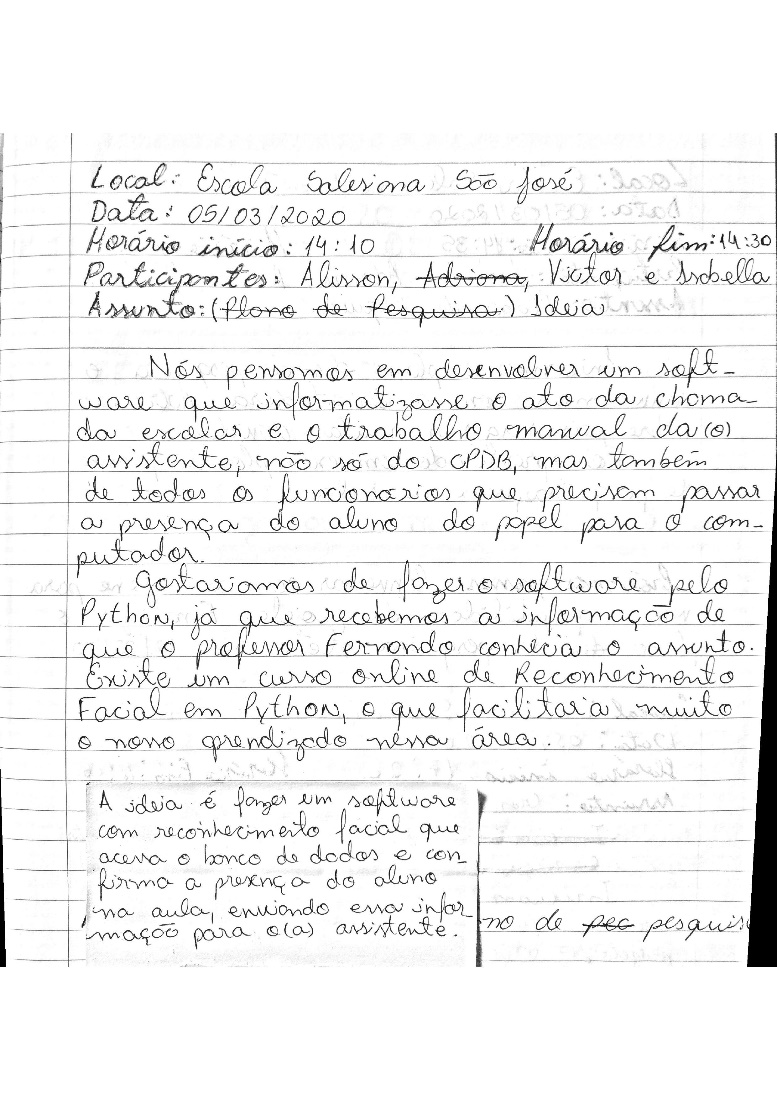
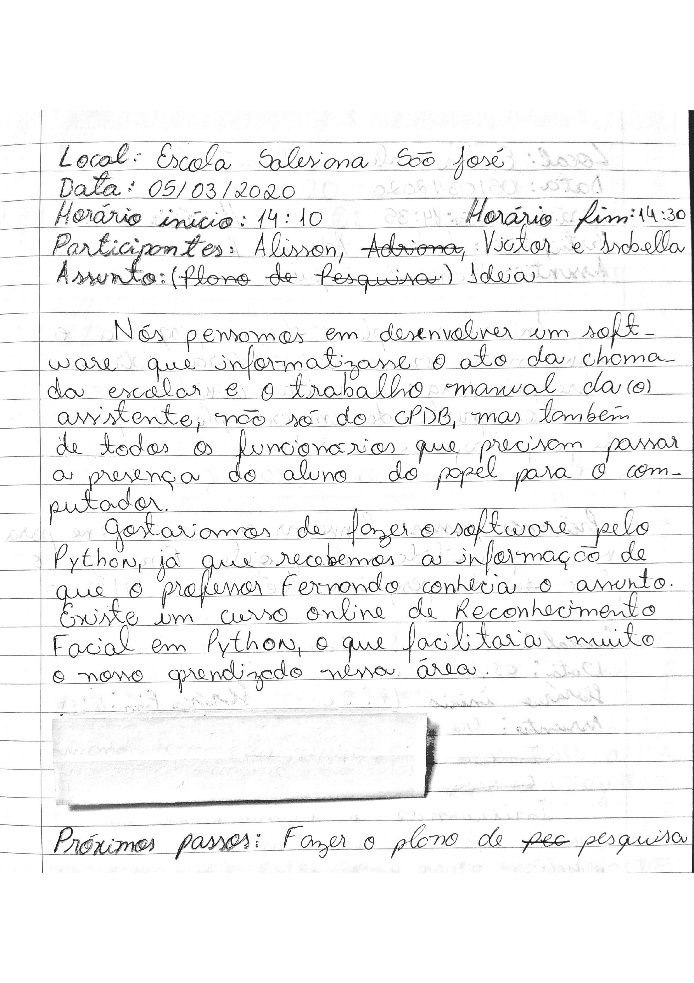
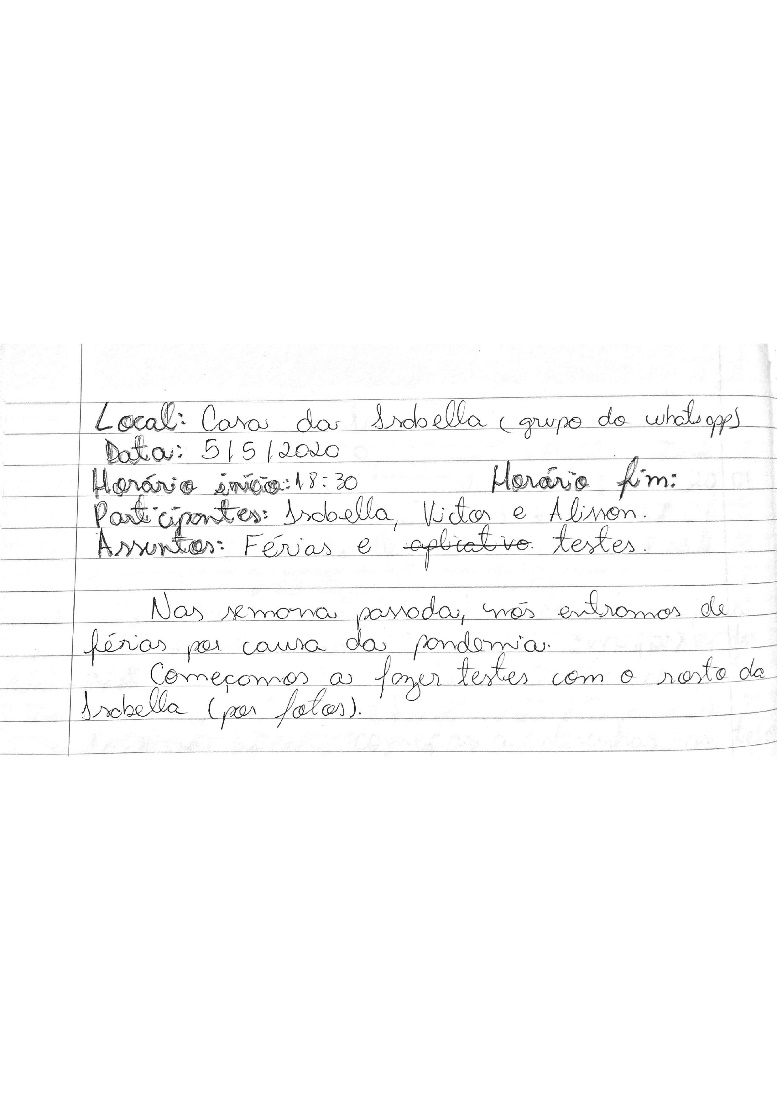
**DIÁRIO DE BORDO**

**Alisson Fantin Rodrigues**  
**Isabella Rosa Lima**

**Victor Monteiro de Oliveira**

**Orientadora: Adriana Coelho Sigrist Silva**





* LOCAL: Sala virtual de Banco de Dados no Teams
* DATA: 04/06
* HORÁRIO INÍCIO: 14:40
* HORÁRIO FIM: 14:50
* PARTICIPANTES: Alisson, Isabella e Victor
* ASSUNTO: Banco de Dados

Conversamos com o gale sobre salvar foto no banco de dados, ele falou q isso não é o adequado se for muitas fotos, pois deixa o baco de dados pesado, ele sugeriu salvar o diretório aonde se encontra essas fotos.

PRÓXIMOS PASSOS: Falar com os orientadores.

* LOCAL: Sala virtual de Banco de Dados no Teams
* DATA: 05/06
* HORÁRIO INÍCIO: 14:30
* HORÁRIO FIM: 14:40
* PARTICIPANTES: Alisson, Isabella, Victor, Prof Adriana e Prof Daniel
* ASSUNTO: Banco de Dados

Conversamos sobre a ideia de que o Gale tinha falado com a gente, salvar o diretório da foto no banco de dados, O Daniel concordo e também deu a ideia de salvar as fotos na nuvem e no banco de dados salvar a localização delas. Optamos pela ideia de salvar o diretório do computador e não o da nuvem como o Daniel sugeriu.

PRÓXIMOS PASSOS: Aplicar a ideia no projeto

* LOCAL: Casa do Victor
* DATA: 02/06
* HORÁRIO INÍCIO: 16:20
* HORÁRIO FIM: 18:00
* PARTICIPANTES: Victor
* ASSUNTO: Python

Foi iniciado o desenvolvimento do sistema de reconhecimento facial em python, com o celular sendo usado como webcam, arquivos haarcascade e codificação baseada no curso de reconhecimento fácil com python disponível na udemy , o sistema reconhece e rostos e grava as fotos para o treinamento dos algoritmos em uma pasta prédefinida.

PRÓXIMOS PASSOS: desenvolver o ambiente de treinamento.

* LOCAL: Casa do Alisson
* DATA: 11/06
* HORÁRIO INÍCIO: 08:30
* HORÁRIO FIM: 10:30
* PARTICIPANTES: Alisson
* ASSUNTO: Banco de Dados, C#

Fiz algumas correções no aplicativo de cadastro, incluindo a parte de salvar o diretório das fotos no banco de dados, e nosso aplicativo já faz o cadastro, atualiza, apaga as informações no banco de dados, para ele ficar completo falta o desenvolvimento da parte de consultar os alunos presentes na aula.

PRÓXIMOS PASSOS: Testar o programa

* LOCAL: Casa do Victor
* DATA: 06/07
* HORÁRIO INÍCIO: 16:20 HORÁRIO FIM: 18:00
* PARTICIPANTES: Victor
* ASSUNTO: C#, banco de dados

Foram feitas modificações no aplicativo de cadastro permitindo apenas usuários identificados como admin fazerem alterações no banco e de dados. No banco de dados foram inseridas em cada uma das tabelas as colunas usrmod e datmod utilizada para armazenar quem foi o usuário que realizou a modificação e a data da última modificação, a gravação dessas informações é feita de forma automática a partir de qualquer alteração feita no aplicativo de cadastro(como novo cadastro e atualizações).

Fragmento Banco:



Fragmento código:



PRÓXIMOS PASSOS: desenvolver o ambiente de treinamento.

* LOCAL: Casa do Alisson
* DATA: 11/07
* HORÁRIO INÍCIO: 16:00
* HORÁRIO FIM: 20:00
* PARTICIPANTES: Alisson
* ASSUNTO: Reconhecimento Facial Python

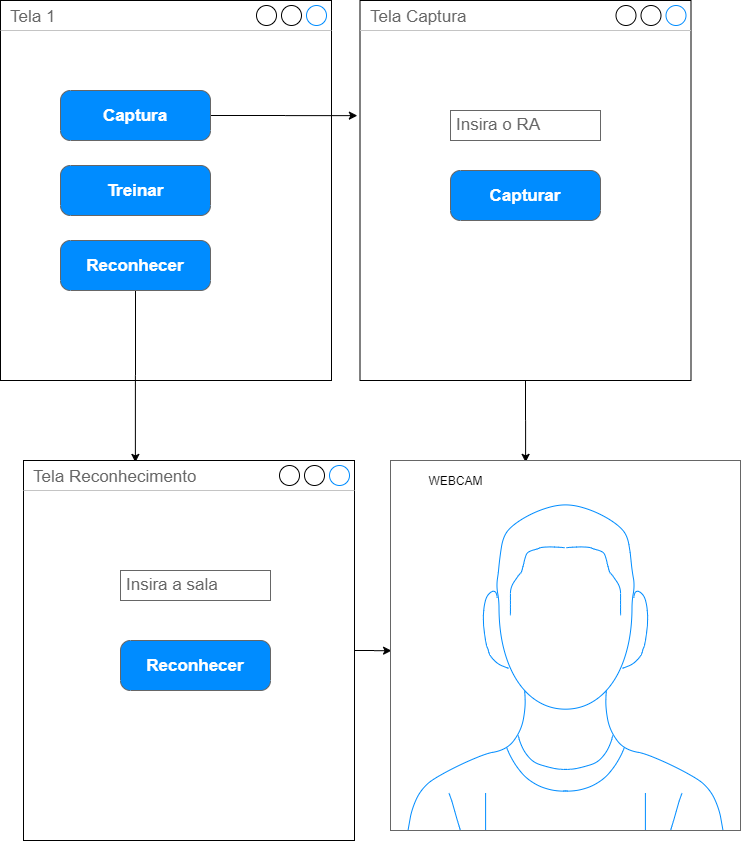
Foi concluído o curso de reconhecimento facial, a partir disso foi desenvolvido um software que reconhece eu e a minha irmã.

PRÓXIMOS PASSOS: Aplicar os algoritmos no projeto.

* LOCAL: Whatsapp
* DATA: 12/07
* HORÁRIO INÍCIO: 13:00
* HORÁRIO FIM: 14:00
* PARTICIPANTES: Alisson, Victor
* ASSUNTO: Interface Python

Discutimos sobre o software em python, e tivemos a ideia de desenvolver uma interface. Nela terá três funções: capturar foto dos alunos, treinar os algoritmos com as fotos capturadas, e reconhecer o aluno para assim dar presença no banco de dados.

Rascunho das interfaces:



PRÓXIMOS PASSOS: Desenvolver a interface em Python.

* LOCAL: TEAMS
* DATA: 17/07
* HORÁRIO INÍCIO: 17:20
* HORÁRIO FIM: 19:00
* PARTICIPANTES: Professora Adriana, professor Daniel, Alisson, Victor, Isabella, Caio, Germano, Drumond
* ASSUNTO: Reunião de Projetos

Foi feito uma reunião com os professores de projetos e os alunos que tinham interesse de participar das feiras, foi mostrado o andamento de cada projeto, apesar do meu nervosismo e problemas com o computador consegui apresentar todos os feitos, tudo indica que os professores gostaram de como estamos andando com o projeto.

PRÓXIMOS PASSOS: pesquisar sobre a biblioteca gráfica tkinter do python, recomendada pelo aluno Caio durante a reunião.

* LOCAL: Casa do victor
* DATA: 18/07
* HORÁRIO INÍCIO: 18:00
* HORÁRIO FIM: 19:00
* PARTICIPANTES: Victor
* ASSUNTO: C#, Banco de dados

Conforme pontuado pelo Prof. Daniel e a Prof. Adriana na reunião do dia anterior no teams, foi adicionado as informações de cadastro o campo “datcria” referente a data de criação do cadastro, o C# grava a hora atual em insere de maneira automática a data de criação não modificável junto com as outras informações.

Fragmento:



PRÓXIMOS PASSOS: testar se o aplicativo está funcional e assertivo, para dar início a estilização.

* LOCAL: Casa do Alisson
* DATA: 19/07
* HORÁRIO INÍCIO: 13:30
* HORÁRIO FIM: 17:00
* PARTICIPANTES: Alisson
* ASSUNTO: Desenvolvimento do software Python

Foi concluído parte do desenvolvimento em Python, o software já reconhece o rosto e coloca presença no banco de dados, porém existe erros ao reconhecer a pessoa: ao fazer teste comigo mesmo muitas vezes fui reconhecido como a minha irmã.

PRÓXIMOS PASSOS: Testar parâmetros e verificar qual algoritmo de reconhecimento é mais eficiente.

* LOCAL: Whatsapp
* DATA: 19/07
* HORÁRIO INÍCIO: 17:30
* HORÁRIO FIM: 18:30
* PARTICIPANTES: Alisson, Victor, Isabella
* ASSUNTO: Discussão sobre o reconhecimento

Discutindo sobre o desenvolvimento do projeto, percebemos que para ele ser aplicado em uma escola deverá ter um computador em cada sala. Isso é algo inviável pra muitas escolas, então foi proposto duas soluções:

Adaptar nosso projeto para um aplicativo mobile, assim precisaria apenas de um tablet em cada sala, o custo seria reduzido e ficaria viável para muitas escolas.

A outra solução seria um computador central que estaria conectado a webcam de cada sala, e ele processaria o reconhecimento facial de todas as webcams ao mesmo tempo.

Vale ressaltar que essas soluções são apenas hipotéticas, e não será aplicada no desenvolvimento do projeto.

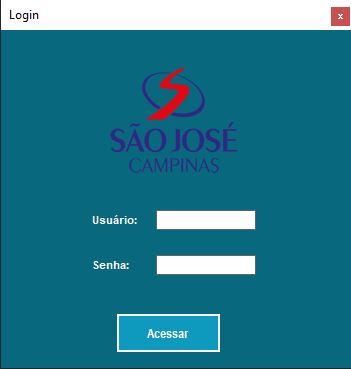
* LOCAL: Casa do victor
* DATA: 22/07
* HORÁRIO INÍCIO: 9:00
* HORÁRIO FIM: 11:00
* PARTICIPANTES: Victor
* ASSUNTO: Teste do aplicativo em C#

Foram checadas as partes do programa com modificações que apresentaram erros que no mesmo dia foram resolvidos, a maioria sendo problemas de escrita na comunicação do aplicativo com o banco de dados.

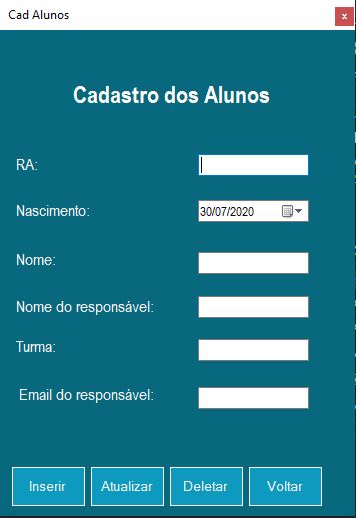
PRÓXIMOS PASSOS: Estilizar o aplicativo em C#

* LOCAL: Casa do victor
* DATA: 22/07
* HORÁRIO INÍCIO: 18:00
* HORÁRIO FIM: 20:00
* PARTICIPANTES: Victor
* ASSUNTO: Estilização do aplicativo em C#

Foram aplicadas cores e fonte nas telas existentes, inspiradas nas telas do site de gestão escolar do essj. E ficaram assim

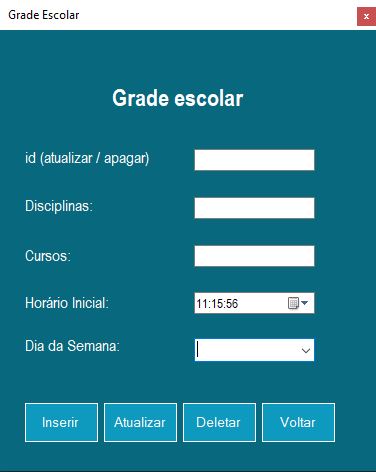












PRÓXIMOS PASSOS: Testar e classificar os algoritmos de reconhecimento C#

* LOCAL: Casa do victor
* DATA: 30/07
* HORÁRIO INÍCIO: 16:40
* HORÁRIO FIM: 18:00
* PARTICIPANTES: Victor
* ASSUNTO: Esclarecimento de relação de tabela no banco

No banco escola as relações ficaram:

Alunos:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RA | nome | cod\_curso | nascimento | responsavel | email\_resp | usrmod | datmod | datcria |

Cursos:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| cod\_curso | descricao | foto | usrmod | datmod | datcria |

Disciplinas:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cod\_disciplina | descricao | usrmod | datmod | datcria |

Frequencia:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| presente | RA | cod\_disciplina | data | usrmod | datamod |

Grade\_escolar:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| id | horario\_inicial | cod\_curso | dia\_semana | cod\_disciplina | usrmod | datmod | datcria |

Login:

|  |  |  |
| --- | --- | --- |
| usuario | senha | permissao |

Prof\_aulas:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| cod\_curso | cod\_disciplina | chapa\_prof | usrmod | datmod | datcria |

Professores:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| chapa\_prof | nome | usrmod | datmod | datcria |

PRÓXIMOS PASSOS: Comentar os códigos.

* LOCAL: Casa do victor
* DATA: 30/07
* HORÁRIO INÍCIO: 18:00
* HORÁRIO FIM: 19:30
* PARTICIPANTES: Victor
* ASSUNTO: Comentarios

Comentei as partes importantes do codigo em C#.

* LOCAL: Casa do victor
* DATA: 31/07
* HORÁRIO INÍCIO: 17:15
* HORÁRIO FIM: 18:00
* PARTICIPANTES: Victor
* ASSUNTO: C#

Observei que no código atual na parte do cadastro de cursos, aulas, professores e disciplinas ele não apresentava erros mesmo se deixássemos uma info do cadastro em branco tentei pensar em uma solução usando ifs e try porém não cheguei em nada efetivo irei tentar de novo amanhã.

Exemplo da solução não funcional:



PRÓXIMOS PASSOS: Pensar em uma nova solução.

* LOCAL: Casa do Alisson
* DATA: 01/08
* HORÁRIO INÍCIO: 16:00
* HORÁRIO FIM: 17:30
* PARTICIPANTES: Alisson
* ASSUNTO: Algoritmos de reconhecimento facial

No aplicativo de reconhecimento facial estamos utilizando a biblioteca opencv, estamos testando 3 principais algoritmos de reconhecimento dessa biblioteca: Eigenface, Fisherface, LBPH.

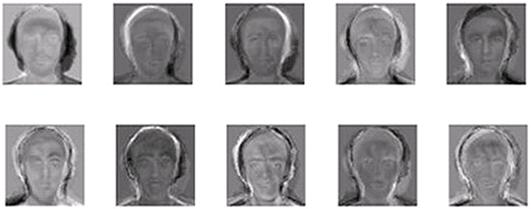
Eigenface: o algoritmo consiste na redução das dimensões da imagem a partir da criação dos eigenfaces.

As imagens de treinamento são submetidas a uma Análise de Componentes Principais (PCA) e a partir dessa análise gera as eigenfaces, elas são geradas a partir das áreas com maior variância do rosto.

Para reconhecer a pessoa, o algoritmo transforma a imagem teste em um eigenface e compara a distância (Cálculo KNN) entre essa nova eigenface com as eigenface criadas no treinamento.

Vale ressaltar que a iluminação interfere no funcionamento desse algoritmo já que ele é um fator de variância no nosso rosto.

Exemplo de Eigenfaces:

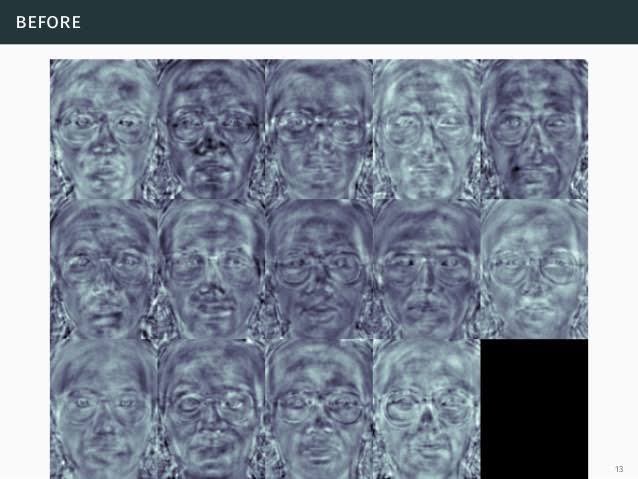


Fisherface: assim como o Eigenface o objetivo desse algoritmo é reduzir as dimensões da imagem, para assim aplicar o reconhecimento.

Para essa redução é utilizado a análise discriminante linear (LDA), diferente do PCA que foca na variação do rosto, essa técnica foca em maximizar as diferenças entre as pessoas.

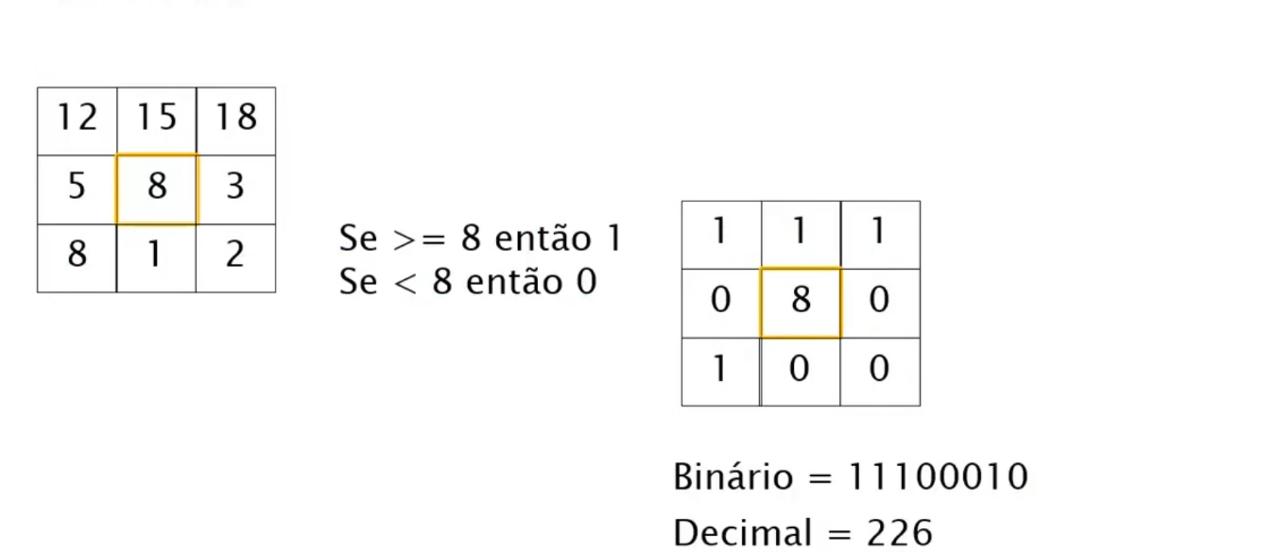
Diferente do Eigenface, a iluminação influencia menos no funcionamento do algoritmo.

Exemplo de Fisherface:

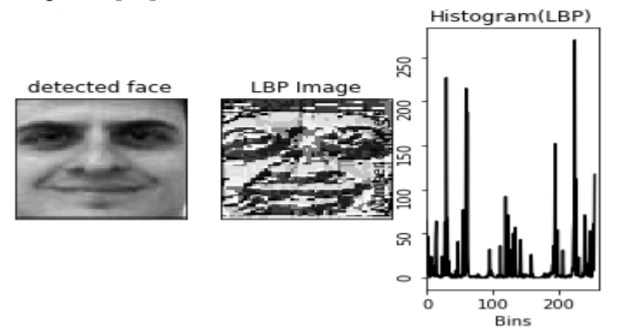


Local Binary Patterns Histogram (LBPH): o algoritmo consiste em pegar o valor LBP, formar um histograma para comparar as imagens.

algoritmo utiliza o valor de um pixel central e compara com os pixels vizinhos se o pixel vizinho for maior que o central é colocado no local 1, se não é colocado 0. A partir disso forma um número binário que é convertido em decimal, esse número em decimal é o LBP.



Com isso é gerado um histograma correspondente com a quantidade de elementos de cada LBP.



E então para reconhecer a pessoa, basicamente é comparado Histograma da imagem teste, com o das imagens de treinamento.

Links consultados sobre o assunto:

<https://repositorio.ufu.br/bitstream/123456789/22158/3/ReconhecimentoFacialAlgotimos.pdf>

<http://www.sinfic.pt/SinficWeb/displayconteudo.do2?numero=44666#:~:text=Esta%20t%C3%A9cnica%20foi%20desenvolvida%20por,combina%C3%A7%C3%A3o%20linear%20de%20diversas%20Eigenfaces>

<https://iq.opengenus.org/lbph-algorithm-for-face-recognition/>

Além dos links muita informação foi retirada do curso de reconhecimento facial da Udemy.

* LOCAL: Casa do Alisson
* DATA: 01/08
* HORÁRIO INÍCIO: 18:00 HORÁRIO FIM: 3:00
* PARTICIPANTES: Alisson
* ASSUNTO: Teste do algoritmo de reconhecimento facial

Foi desenvolvido um aplicativo à parte para verificar qual algoritmo de reconhecimento facial e quais parâmetros os tornam mais eficientes para usar no projeto. Para criação desse aplicativo utilizamos como base no aplicativo teste\_yale, feito durante o curso de reconhecimento facial da Udemy, que foi modificado para corresponder com as nossas necessidades.

Montamos um banco de imagens para testes internos. Para montar o banco de treinamento pegamos 20 pessoas e colocamos 15 fotos de cada uma delas. Para montar o banco de testes selecionamos 15 fotos no total, sendo 10 de pessoas diferentes que o algoritmo foi treinado para reconhecer (vale ressaltar que as fotos de testes, nenhuma foi utilizada para treinamento), e 5 fotos retiradas do site <https://thispersondoesnotexist.com/>.

A ideia é que o algoritmo retorne: quantidade de acertos no reconhecimento, NIC (não identificado certo), NIE (não identificado errado), a média de confiança (threshold).

O cenário perfeito seria um em que o algoritmo retorna: 10 acertos, 5 NIC (Aqui entraria as 5 imagens tiradas do site “thispersondoesnotexist.com”, já que o algoritmo não foi treinado para reconhecer elas), 0 NIE (aqui se encaixa quando uma pessoa era pra ter sido reconhecida, mas não foi).

Para ter uma margem de parâmetros de treinamento que devemos usar nos testes, consultamos dois trabalhos. Para verificar parâmetros do Eigenface e Fisherface, usamos o trabalho “Reconhecimento Facial com os algoritmos Eigenfaces e Fisherfaces” escrito por Thelry David Bissi da Universidade Federal de uberlandia, e para o LBPH, “RECONHECIMENTO DE PADRÕES FACIAIS: UMA SÍNTESE” de VAMBASTER JOSÉ DA COSTA da Universidade Tecnológica Federal do Paraná.

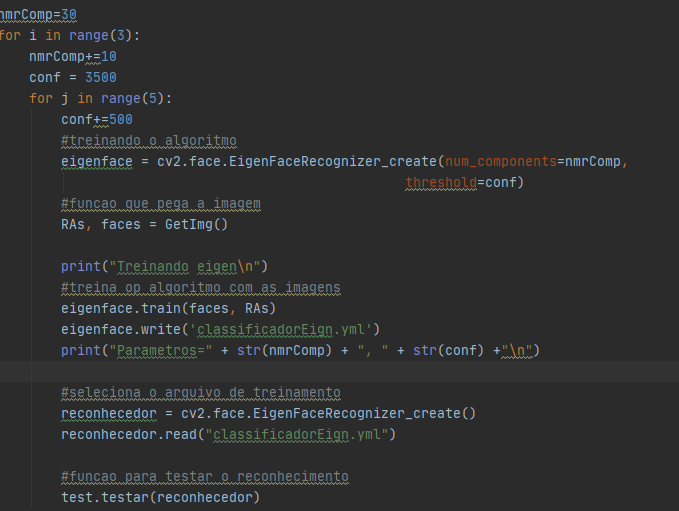
Para o EigenFace usamos de parâmetros: número de componentes = 40, 50, 60; Threshold= 4000, 4500, 5000, 5500, 6000.

Para o Fisherface usamos de parâmetros: número de componentes = 29, 39, 49; Threshold= 1000, 1200, 1400, 1600, 1800.

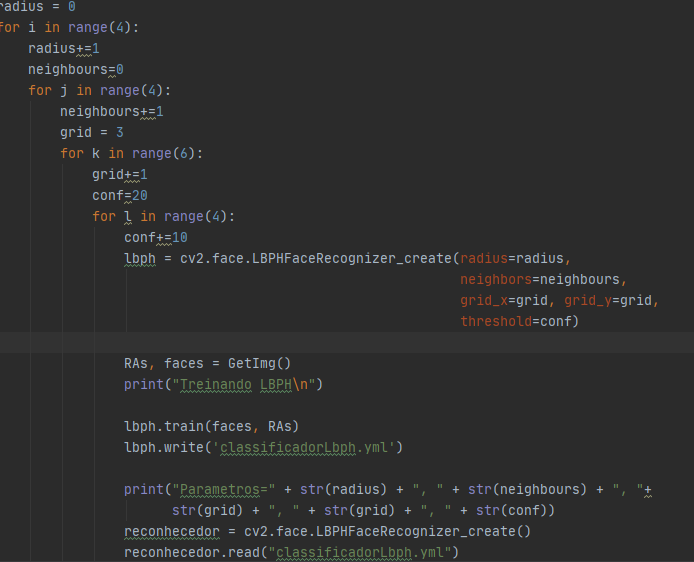
Para o LBPH usamos de parâmetros: raio = 1, 2, 3, 4; vizinho=1, 2, 3, 4; grid x e grid y = 4, 5, 6, 7, 8, 9; Threshold= 30, 40, 50, 60.

O algoritmo retorna com prints no console. Essas informações serão colocadas em uma tabela do Excel para serem analisadas.

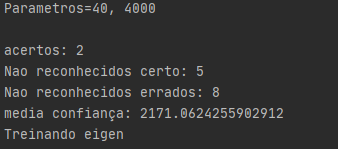
Trechos do codigo:







Rodando o programa:



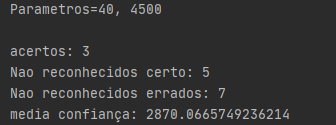
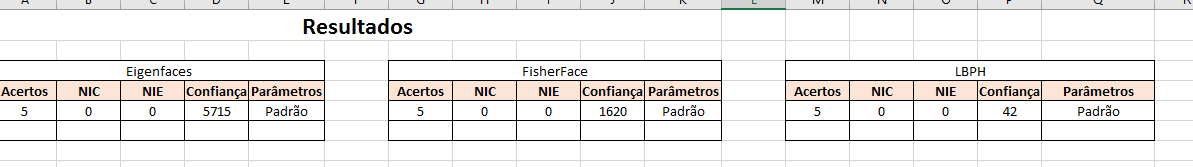


Tabela:



Links dos trabalhos consultados:

<http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/11953/1/SH_COCIC_2019_01_06.pdf>

<https://repositorio.ufu.br/bitstream/123456789/22158/3/ReconhecimentoFacialAlgotimos.pdf>

PRÓXIMOS PASSOS: Rodar o aplicativo e preencher as tabelas no excel.

* LOCAL: Casa do victor
* DATA: 01/08
* HORÁRIO INÍCIO: 16:00 HORÁRIO FIM: 18:00
* PARTICIPANTES: Victor
* ASSUNTO: Modificações do aplicativo em C#

A partir de observações de falhas do aplicativo, percebi que precisávamos modificar as classes e adicionar as verificações, então a aplicação começou a apresentar erros sem conserto, logo passei ele para um novo projeto mais bem escrito.

PRÓXIMOS PASSOS: Remontar as classes aplicativo em C#

* LOCAL: Casa do victor
* DATA: 02/08
* HORÁRIO INÍCIO: 12:00
* HORÁRIO FIM: 18:00
* PARTICIPANTES: Victor
* ASSUNTO: Importação das classes do aplicativo em C#

Inseri no código refatorado as classes com correções, e reconstrui os forms para testes prontos para aplicar a estilização mais bem feita.

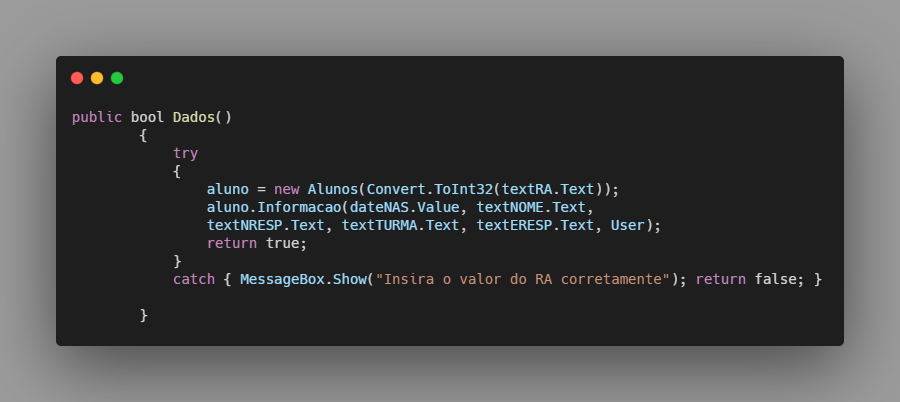
* LOCAL: Whatsapp
* DATA: 02/08
* HORÁRIO INÍCIO: 18:00
* HORÁRIO FIM: 19:00
* PARTICIPANTES: Victor, Alisson
* ASSUNTO: Discussão da importância das classes

Discutimos o encaminhamento do projeto e as modificações futuras, dependendo do andamento do projeto.

PRÓXIMOS PASSOS: Remontar o código dos forms aplicativo em C#

* LOCAL: Casa do victor
* DATA: 03/08
* HORÁRIO INÍCIO: 17:00 HORÁRIO FIM: 20:00
* PARTICIPANTES: Victor
* ASSUNTO: modificação do codigo dos forms C#

Redigitei os códigos de input dos forms e sem muito sucesso, pensei em maneiras de contornar os maiores erros que tive, relacionados a verificação do tipo e valor das textboxes para valores vazios inseri nas classes a verificação que caso fosse falsa retornaria um erro

E para os inputs inteiros a solução que eu pensei na hora vou verificar se era possível converter o número para inteiro e se sim executar o processo, porem este método faz a própria camada de input executar verificações de regras de negócio e isso vai ser alterado futuramente

* LOCAL: Casa do victor
* DATA: 03/08
* HORÁRIO INÍCIO: 17:00
* HORÁRIO FIM: 20:00
* PARTICIPANTES: Victor
* ASSUNTO: modificação do codigo dos forms C#

As mesmas aplicações de verificação na tela de inserção de grades e a de consulta de frequência apresentaram erros que eu não encontrei solução até o momento.

LOCAL: Casa do victor

DATA: 04/08

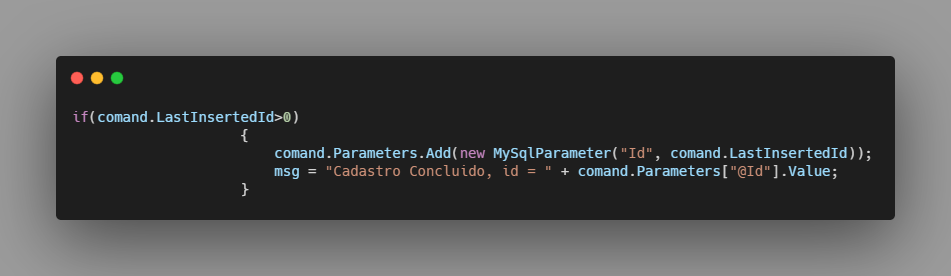
HORÁRIO INÍCIO: 17:00 HORÁRIO FIM: 21:00

PARTICIPANTES: Victor

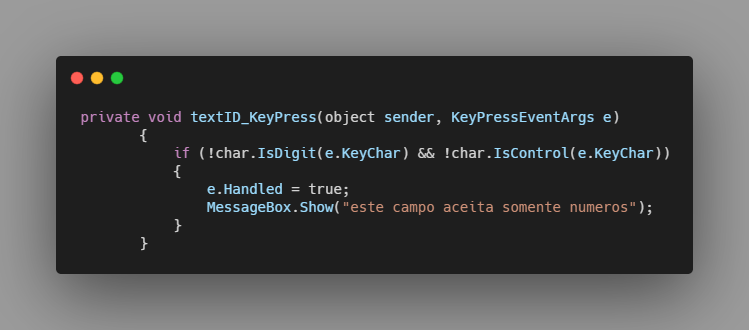
ASSUNTO: modificação do codigo dos forms C#

Pesquisando na internet métodos de retornar o id para o usuário após o cadastro enconteri o artigo: <https://imasters.com.br/back-end/c-obtendo-o-id-do-ultimo-registro-inserido-no-banco-de-dados-mysql>

E implementei a solucão:



Com pequenas alterações para o meu proposito tive êxito em retornar o id corretamente, depois de alguma pesquisa encontrei o método de receber apenas dígitos no textbox que descarta a verificação de uma regra de negócio na camada de input e amanhã vou implementar no código toda já que ela está presente apenas na parte de grade, que tive maiores problemas :



Depois o problema de valores nulos foi resolvido setando o tipo de variáveis inteiras como dynamic, verificando e convertendo na própria classe:

* LOCAL: Casa do victor
* DATA: 05/08
* HORÁRIO INÍCIO: 17:00
* HORÁRIO FIM: 20:00
* PARTICIPANTES: Victor
* ASSUNTO: modificação do codigo dos forms C#

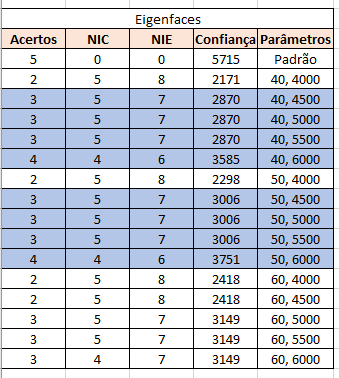
Implementei as modificações em todas as classes e forms.

PRÓXIMOS PASSOS: Testa o aplicativo e determinar se está pronto.

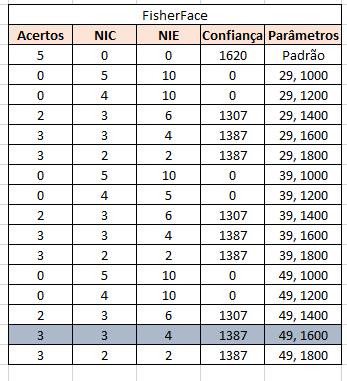
* LOCAL: Casa do Alisson
* DATA: 10/08
* HORÁRIO INÍCIO: 18:00
* HORÁRIO FIM: 19:00
* PARTICIPANTES: Alisson
* ASSUNTO: Resultado dos testes

Levamos como um fator muito importante os acertos do NIC, pois em um sistema real pessoas que não eram pra ser reconhecidas, não podem ser reconhecidas de maneira nenhuma.

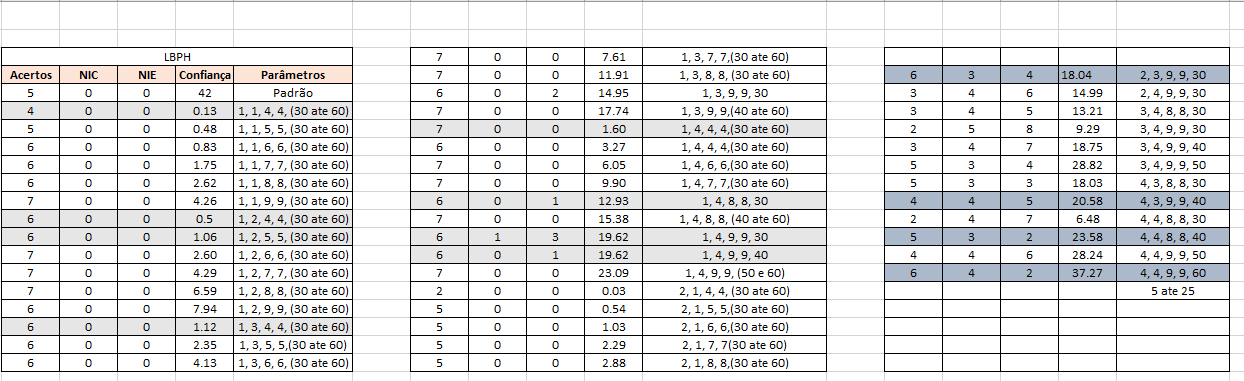
EigenFace: Como vimos os pintados em azul mostram os melhores resultados, não conseguimos uma boa taxa de acerto com esse algoritmo, porém conseguimos que ele não reconhece as pessoas que não eram pra ser reconhecidas.



FisherFace: esse algoritmo aprensentou os resultados mais frustrantes, não teve uma boa taxa de acerto, nem uma boa taxa de NIC.



LBPH: Esse algoritmo pela grande quantidade de parâmetros não registramos todos os resultados nas tabelas, apenas os mais importantes, já que na grande maioria ele reconhecia as pessoas que não eram para ser reconhecidas. Podemos analisar que esse algoritmo com os parâmetros “4, 4, 9, 9, 60”, foi o que atingiu o melhor resultado com 6 acertos e 4 NIC.



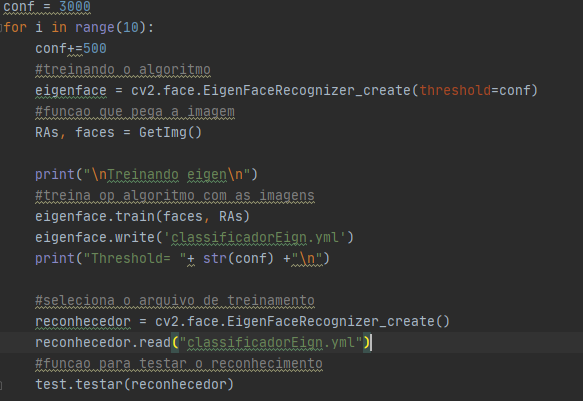
Esperamos um algoritmo que tenha 7 acertos e 4 NIC, como não atingimos esse minimo desejado, iremos continuar os testes.

PRÓXIMOS PASSOS: Novos Testes.

* LOCAL: Casa do Alisson
* DATA: 10/08
* HORÁRIO INÍCIO: 19:00
* HORÁRIO FIM: 20:00
* PARTICIPANTES: Alisson
* ASSUNTO: Novos Testes.

Como podemos observar não conseguimos os resultados mínimos que precisamos, para esses novos testes, de primeiro momento vamos fazer algo que deveríamos ter feito antes: Iremos apenas variar o número de threshold dos algoritmos.

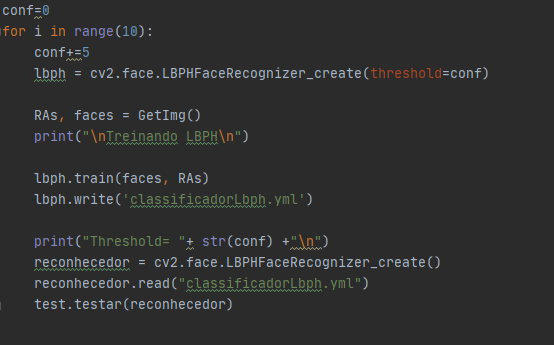
EigenFace: o threshold será variado entre 3500 até 8000 de 500 em 500.



FisherFace: o threshold será variado de 400 até 2200 de 200 em 200.



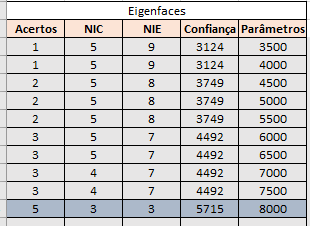
LBPH: o threshold será variado de 5 até 50 de 5 em 5.



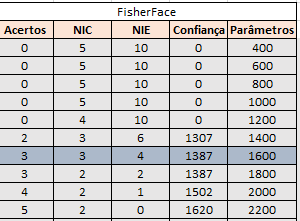
PRÓXIMOS PASSOS: Executar o teste planejado.

* LOCAL: Casa do Alisson
* DATA: 12/08
* HORÁRIO INÍCIO: 13:00
* HORÁRIO FIM: 16:00
* PARTICIPANTES: Alisson
* ASSUNTO: Resultados dos testes.

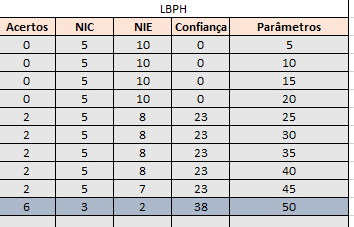
Eigenface:



Fisherface:



LBPH:



**Comparando os dois testes:**

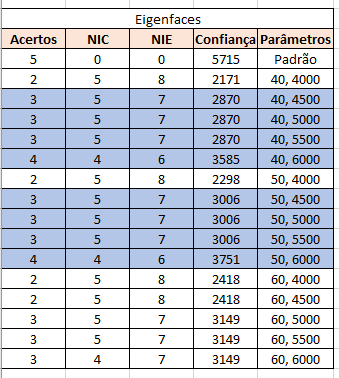
Eigenface melhores resultados:

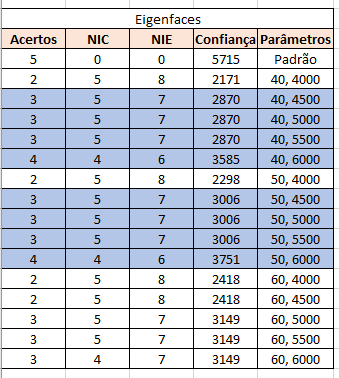
Teste 1

De 10 fotos que eram pra ser reconhecidas: 4 certas, 6 não reconhecidas.

De 5 fotos que não eram para ser reconhecidas: 4 não reconhecidas, 1 sim.

Resumidamente o algoritmo reconheceu errado apenas 1 foto, e não reconheceu 6 que eram pra ter sido reconhecidas.



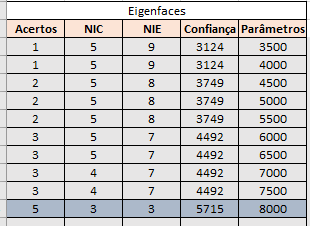


Teste 2

De 10 fotos que eram pra ser reconhecidas: 5 certas, 3 não reconhecidas, 2 reconhecidas errado.

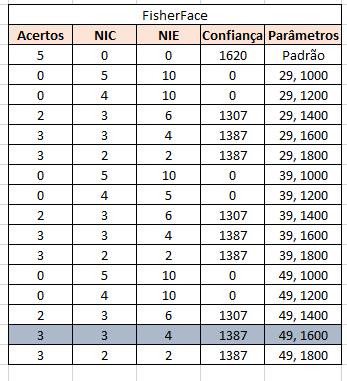
De 5 fotos que não eram para ser reconhecidas: 3 não reconhecidas, 2 sim.

Resumindo o algoritmo reconheceu errado 3 fotos, e não reconheceu 3 fotos que eram para ter sido reconhecida.

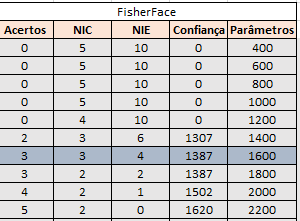


Fisherface melhores resultados:

Teste 1



Teste 2



Resultados iguais em ambos os casos:

De 10 fotos que eram pra ser reconhecidas: 3 certas, 4 não reconhecidas, 3 reconhecidas errado.

De 5 fotos que não eram para ser reconhecidas: 3 não reconhecidas, 2 sim.

Resumindo o algoritmo reconheceu errado 4 fotos, e não reconheceu 7 fotos que eram para ter sido reconhecida.

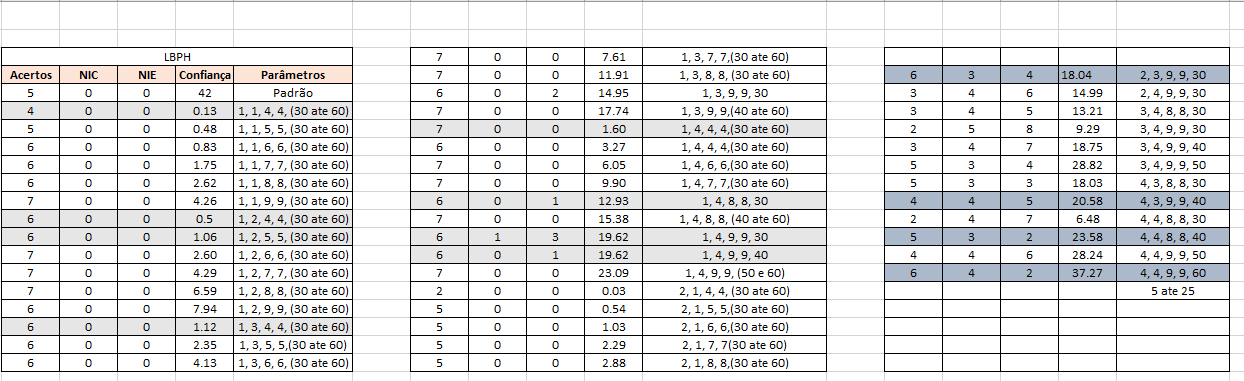
LBPH melhores resultados:

Teste 1:

De 10 fotos que eram pra ser reconhecidas: 6 certas, 2 não reconhecidas, 2 reconhecidas errado.

De 5 fotos que não eram para ser reconhecidas: 4 não reconhecidas, 1 sim.

Resumindo o algoritmo reconheceu errado 3 fotos, e não reconheceu 2 fotos que eram para ter sido reconhecida.

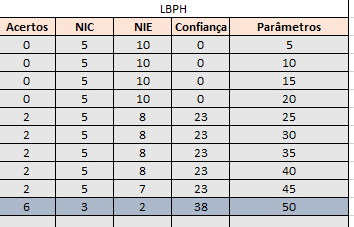


Teste 2:

De 10 fotos que eram pra ser reconhecidas: 6 certas, 2 não reconhecidas, 2 reconhecidas errado.

De 5 fotos que não eram para ser reconhecidas: 3 não reconhecidas, 1 sim.

Resumindo o algoritmo reconheceu errado 3 fotos, e não reconheceu 2 fotos que eram para ter sido reconhecida.



**Conclusão:**

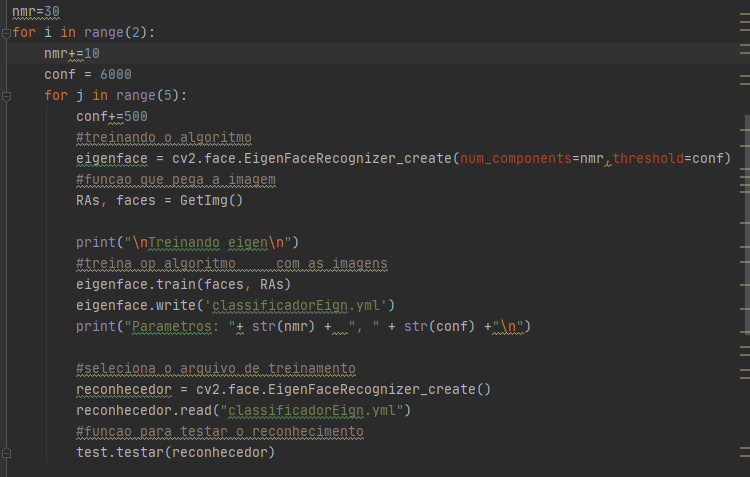
O algoritmo Eigenface e LBPH mostraram resultados melhores no teste 1. então será feito os últimos teste para tentar aperfeiçoar esses algoritmos.

O fisherface não apresentou mudanças, os testes indicam que o que mais influenciou foi o threshold, porém não apresentou resultados satisfatórios, o algoritmo será descartado, e não será usado no projeto.

* LOCAL: Casa do Alisson
* DATA: 13/08
* HORÁRIO INÍCIO: 15:00
* HORÁRIO FIM: 16:00
* PARTICIPANTES: Alisson
* ASSUNTO: Testes finais.

Como analisado, os resultados do primeiro teste foram os mais satisfatórios, e a partir deles vamos fazer novos testes para tentar melhorar ainda mais os algoritmos. Iremos trabalhar apenas com o Eigenface e com o LBPH, já que Fisherface não se mostrou eficiente quando comparado com os outros.

Eigenface: iremos nesse teste utilizar o valor 40 e 50 no número de componentes, e iremos aumentar threshold entre 6500 até 8500 de 500 em 500. A ideia e manter o Número de componentes dos testes que tiveram melhores resultados (40,50), e aumentar o threshold para tentar melhorar os resultados.



LBPH: Já no lBPH, iremos manter o radius em 4 e o neighbours variando entre 3 e 4, o grid x e grid y irá variar entre 9 e 10, e o treshold entre 50 até 80 aumentando de 5 em 5. Nos primeiros testes os resultados com o radius 4, neighbours 3 e 4, e grid 9 tiveram melhores resultados, a ideia é aumentar o grid e o threshold para verificar se terá melhoria.



* LOCAL: Casa do Victor
* DATA: 14/08
* HORÁRIO INÍCIO: 10:00
* HORÁRIO FIM: 12:00
* PARTICIPANTES: Victor
* ASSUNTO: mudança no banco

A partir das pontuações feitas pela professora adriana, na reunião na quarta-feira passada as chaves foram modificadas tornando a tabela grade escolar uma tabela de chaves compostas, assim evitando verificaçoes do aplicativo e tornando os metodos de gravação mais simples, tambem foram adicionadas as colunas usrcria e na tabela frequencia a coluna hora.

Alunos:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RA | nome | cod\_curso | nascimento | responsavel | email\_resp | usrmod | datmod | datcria | usrcria |

Cursos:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| cod\_curso | descricao | foto | usrmod | datmod | datcria | usrcria |

Disciplinas:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| cod\_disciplina | descricao | usrmod | datmod | datcria | usrcria |

Frequencia:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| presente | RA | cod\_disciplina | data | usrmod | datamod | usrcria | hora |

Grade\_escolar:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| horario\_inicial | cod\_curso | dia\_semana | cod\_disciplina | usrmod | datmod | datcria | usrcria |

Login:

|  |  |  |
| --- | --- | --- |
| usuario | senha | permissao |

Prof\_aulas:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| cod\_curso | cod\_disciplina | chapa\_prof | usrmod | datmod | datcria | usrcria |

Professores:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| chapa\_prof | nome | usrmod | datmod | datcria | usrcria |

As colunas em ouro são chaves primarias.

* LOCAL: Casa do Victor
* DATA: 20/08
* HORÁRIO INÍCIO: 14:00
* HORÁRIO FIM: 20:00
* PARTICIPANTES: Victor
* ASSUNTO: Função de salvar em C#

Utilizando o pacote nuget filehelpers, desenvolvi uma função dentro na parte de frequência que exporta como csv os dados que aparecem no datagrid após uma pesquisa. Codigo:



* LOCAL: TEAMS
* DATA: 25/08
* HORÁRIO INÍCIO: 17:10
* HORÁRIO FIM: 18:30
* PARTICIPANTES: Adriana, Isabella, Alisson, Victor, Daniel, Geraldo e outros grupos
* ASSUNTO: Documentação para Bragantec e relatório.

Nos foi informado o necessário para a inscrição da feira Bragantec, nós começamos o plano de pesquisa do modelo deles e estamos trabalhando na introdução do relatório.

Próximos passos: entregar o plano, fazer a inscrição da feira e acabar a introdução.

* LOCAL: TEAMS
* DATA: 01/09
* HORÁRIO INÍCIO: 17:10
* HORÁRIO FIM: 18:30
* PARTICIPANTES: Adriana, Isabella, Alisson, Victor, Daniel, Geraldo e outros grupos
* ASSUNTO: Entrega do Plano da Bragantec

Apresentamos o plano da Bragantec, fizemos alguns ajustes e entregamos para os professores.

A introdução e resumo do relatório seguem sendo ajustados, acredito que na semana que vem já conseguimos partir para os próximos capítulos.

* LOCAL: TEAMS
* DATA: 02/09
* HORÁRIO INÍCIO: 14:30
* HORÁRIO FIM: 16:00
* PARTICIPANTES: Isabella, Alisson, Victor
* ASSUNTO: Ajustes relatório de Pesquisa

Nos reunimos para arrumar o relatório como foi orientado, acreditamos já ter finalizado o resumo e a introdução, partiremos para a pesquisa e escrita do próximo capítulo.

* LOCAL: Casa da Isabella
* DATA: 02/09
* HORÁRIO INÍCIO: 19:30
* HORÁRIO FIM: 22:00
* PARTICIPANTES: Isabella
* ASSUNTO: Interface em Python

Comecei o desenvolvimento da interface na linguagem de programação Python, usando a ferramente tkinter com a seguinte playlist de vídeo aula: <https://www.youtube.com/playlist?list=PLesCEcYj003ShHnUT83gQEH6KtG8uysUE>

Dentro dessa ferramenta, existem 2 tipos de posicionadores dos componentes de tela, sendo eles place (usa coordenadas X e Y) e o anchor (usa os parâmetros de bussúla, isso é, norte, sul, leste, oeste etc). Sabendo disso vimos um problema, tendo em vista que para remanejar a interface, usando place, teríamos que sempre mudar as coordenadas, porém o anchor nos limita muito a usar as bordas da interface. Então tive a ideia de usar labels para ajustar na posição que queríamos.

Próximos passos: finalizar a interface para uso contínuo do projeto.

* LOCAL: Whatsapp
* DATA: 07/09
* HORÁRIO INÍCIO: 15:00
* HORÁRIO FIM: 16:00
* PARTICIPANTES: Isabella, Alisson e Victor
* ASSUNTO: Novas Ideias

Pensamos em uma maneira de diminuir o custo de investimento do nosso projeto, tirando as câmeras de cada sala e colocando-a na entrada da escola. Imaginamos as mudanças e tentaremos habilitar essa alternativa de implantação no projeto, abrindo oportunidades para implantar em escolas que não tem tanto capital para investir em algo desse porte.

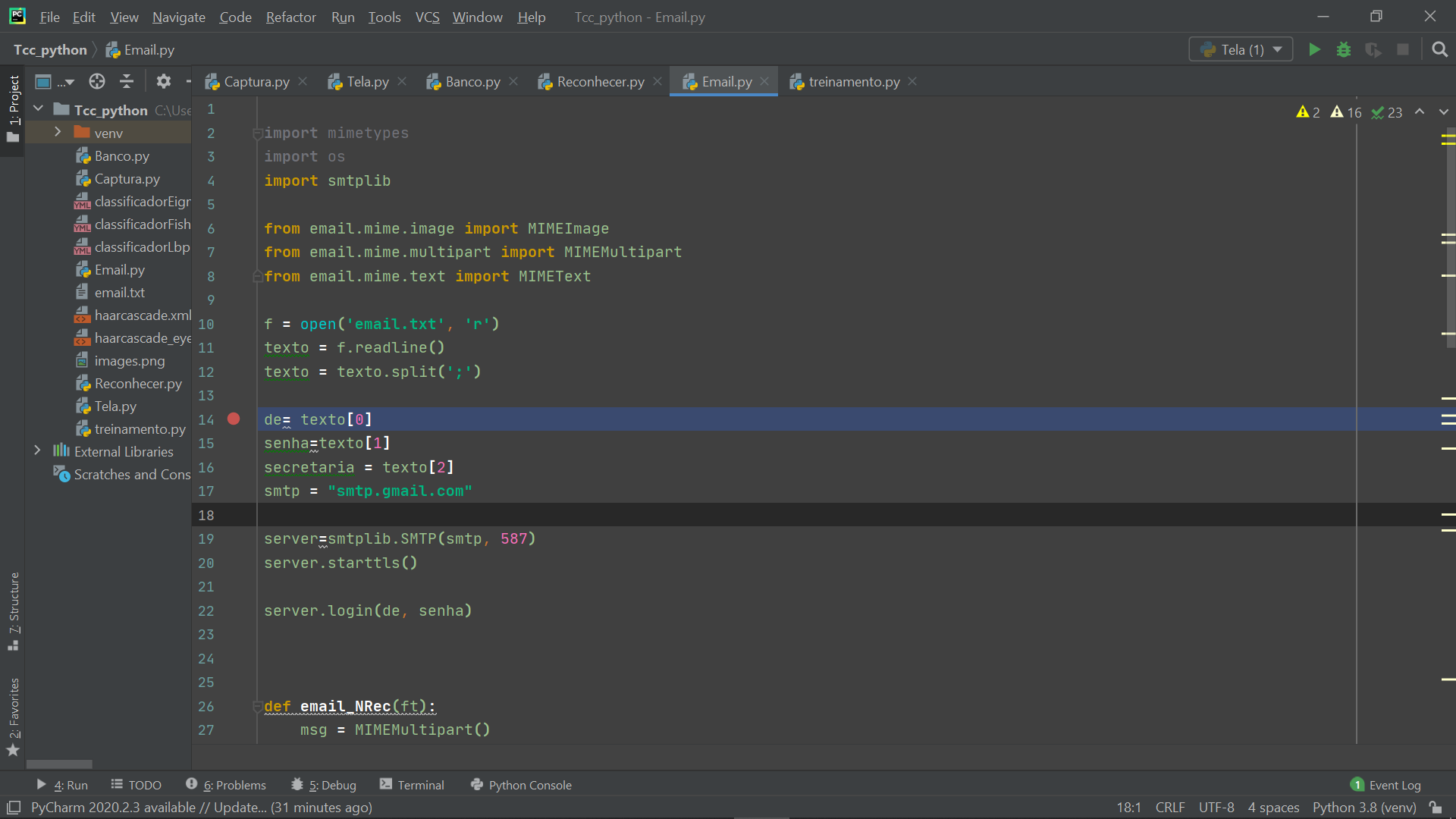
* LOCAL: TEAMS
* DATA: 08/09
* HORÁRIO INÍCIO: 17:10
* HORÁRIO FIM: 18:00
* PARTICIPANTES: Isabella
* ASSUNTO: Reunião

As reuniões após a aula foram canceladas por motivos maiores, entretanto os orientadores fizeram as inscrições hoje na feira da Bragantec.

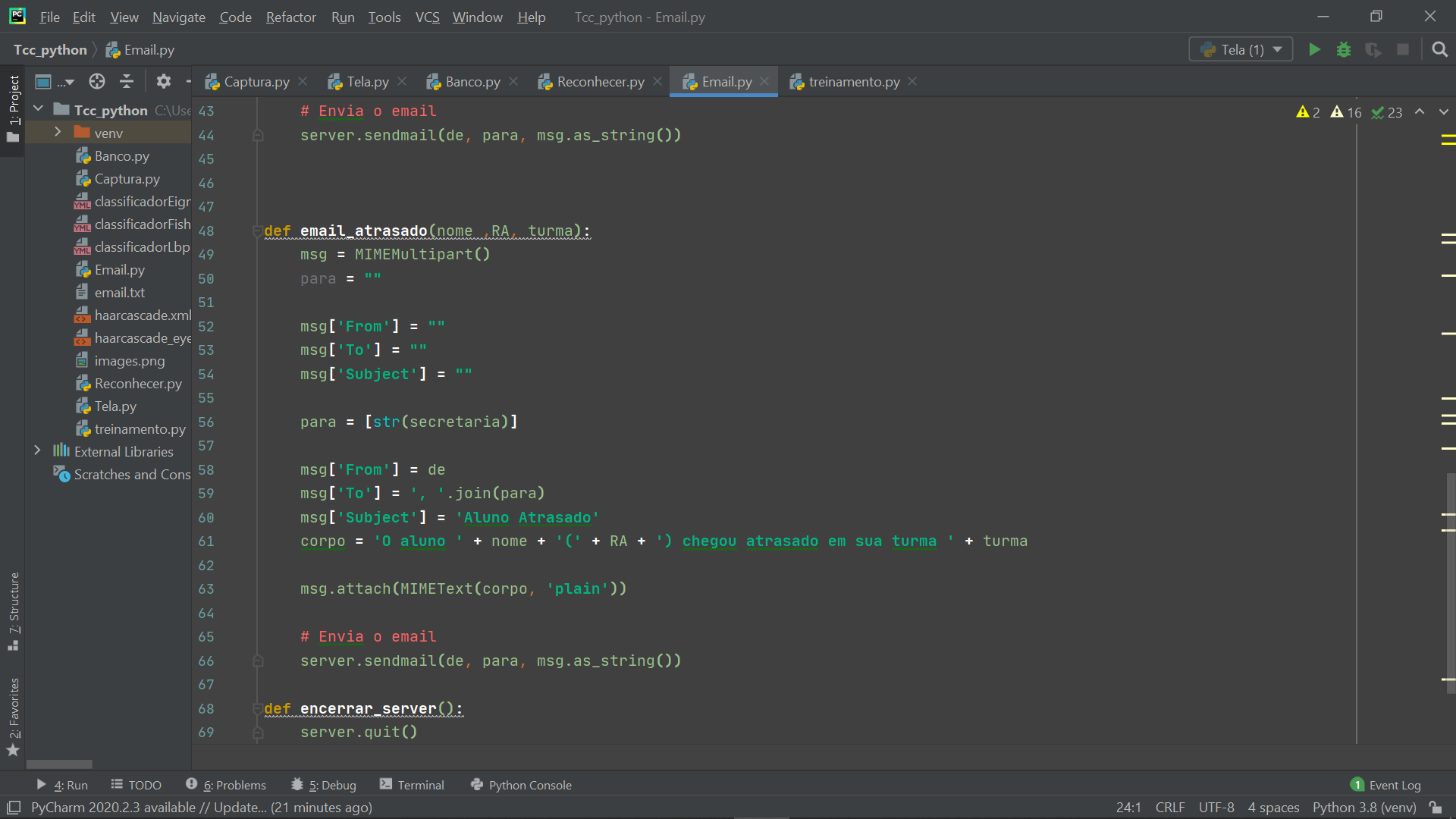
* LOCAL: Casa do Alisson
* DATA: 14/09
* HORÁRIO INÍCIO: 14:00
* HORÁRIO FIM: 20:00
* PARTICIPANTES: Alisson
* ASSUNTO: Funções de e-mail Python

Foi desenvolvido 3 funções de e-mail: uma para avisar os pais que aluno está presente, outra para enviar foto do aluno não reconhecido para a secretaria da escola, e outro para avisar a secretaria que o aluno chegou atrasado.

Ao testar a função de enviar email para os pais percebemos uma lentidão atrapalhando assim o reconhecimento facial, então descartamos essa função, e deixamos como uma funcionalidade adicional que precisa de uma máquina mais potente para funcionar.

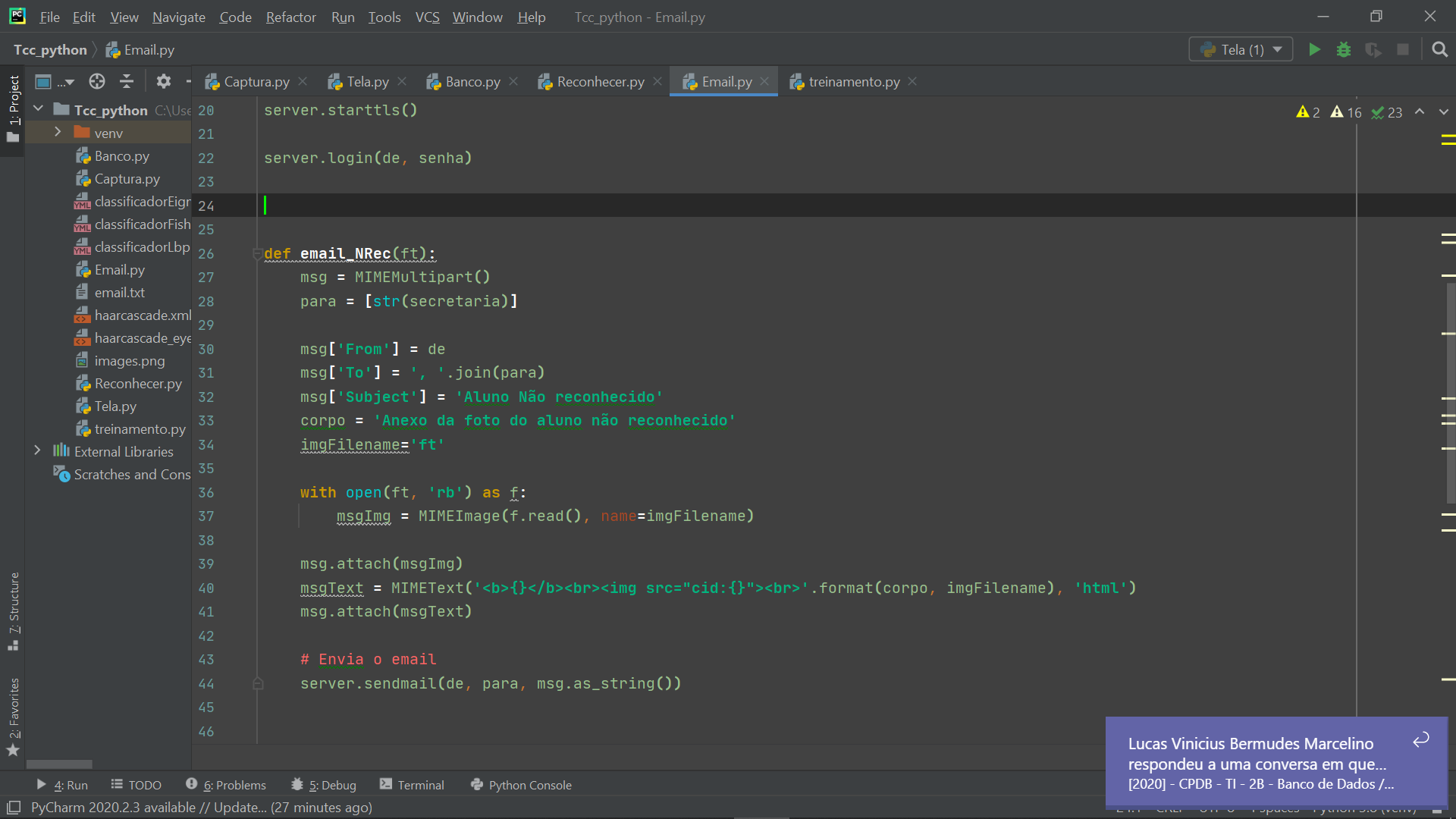
Para pegar as informações de email, a gente usa um arquivo texto que contém um email e senha que será responsável por enviar os emails, e o email da secretaria que é quem vai receber o email. O arquivo é salvo nesse formato: email;senha;email\_secretaria.

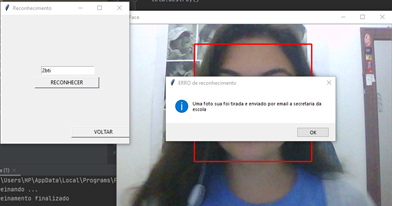
Prints da função de enviar email quando o aluno está atrasado:





Prints da função de enviar email quando o aluno não foi reconhecido:







* LOCAL: Teams
* DATA: 30/10
* HORÁRIO INÍCIO: 17:10
* HORÁRIO FIM: 18:00
* PARTICIPANTES: Alisson, Victor, professores
* ASSUNTO: Ideias

Surgiu a oportunidade de acrescentarmos mais uma função ao projeto, a medição de temperatura dos alunos no mesmo momento do reconhecimento, tentaremos implementar já que o professor Daniel tem um conhecimento grande em arduíno...

* LOCAL: Teams
* DATA: 10/11
* HORÁRIO INÍCIO: 17:10
* HORÁRIO FIM: 19:00
* PARTICIPANTES: Alisson, Victor, Isabella, professores, outros grupos
* ASSUNTO: Ideias

Modificação do vídeo e tentativa de inclusão da ideia de medição de temperatura com o arduino, foram colocadas ideias de construção do sistema de teste com um laptop e colar o sistema por trás da tela

Os professores estão estudando a ideia de irmos até a escola fazer as apresentações das feiras mas não será possível, estamos procurando outra alternativa.

* LOCAL: Teams
* DATA: 11/11
* HORÁRIO INÍCIO: 15:30
* HORÁRIO FIM: 16:20
* PARTICIPANTES: Alisson, Victor, Isabella, professores, outros alunos
* ASSUNTO: Apresentação final

Apresentamos o TCC, recebemos muitas dicas, tais como uma nova biblioteca em Python que pode nos ajudar, o reconhecimento também do professor, e ideias pra ajudar na apresentação das feiras.  
Precisamos atualizar o relatório com a ideia da temperatura, acrescentar o estudo do arduíno e finalizar para enviar para as feiras. Também é necessário finalizar o banner.