PROJETO DE APLICAÇÃO DE GAMIFICAÇÃO PARA CURSOS DE ENSINO A DISTANCIA. (titulo provisório)

Augusto Coelho Melo

Sobral – CE

2020

LISTA DE SIGLAS E ABREVIATURAS

EAD: Ensino a distancia;

*API: Application Programming Interface*;

*JS: JAVASCRIPT*

*CORS: Cross-Origin Resource Sharing* – compartilhamento de recursos de origem cruzada

*REST: Representational State Transfer* – Transferência Representacional de Estado

LISTA DE ILUSTRAÇÕES

Sumário

[INTRODUÇÃO 5](#_Toc52375029)

[Objetivos gerais: 5](#_Toc52375030)

[Objetivos específicos: 5](#_Toc52375031)

[Justificativa 6](#_Toc52375032)

[Fundamentação teórica: 6](#_Toc52375033)

[Definição de regras de gamificação: 6](#_Toc52375034)

[Obtenção de dados: 8](#_Toc52375035)

[Tratamento e filtragem inicial dos dados: 8](#_Toc52375036)

[Estabelecendo um banco de dados com as informações coletadas: 9](#_Toc52375037)

[Web scraping e tratamento dos dados: 10](#_Toc52375038)

[Python 10](#_Toc52375039)

[Selenium 10](#_Toc52375040)

[Armazenamento dos dados 10](#_Toc52375041)

[Mysql 10](#_Toc52375042)

[Criação de API rest 10](#_Toc52375043)

[Node.js 10](#_Toc52375044)

[EXPRESS 10](#_Toc52375045)

[*React-native* 10](#_Toc52375046)

[Consumo de *API* de *login* de outra parte 10](#_Toc52375047)

[*CORS* 10](#_Toc52375048)

[Aplicativo para exibição aos estudantes 11](#_Toc52375049)

[METODOLOGIA 13](#_Toc52375050)

[Coleta de feedback 13](#_Toc52375051)

[Analise quantitativa 13](#_Toc52375052)

[RESULTADOS 13](#_Toc52375053)

[CONCLUSÃO 13](#_Toc52375054)

[REFERENCIAS 13](#_Toc52375055)

[APENDICE A: exemplo de *script* para *webscrap.* 14](#_Toc52375056)

[APENDICE B: exemplo de *script* para tratamento de dados*.* 17](#_Toc52375057)

[APENDICE C: exemplo de *script* para Inserção em banco de dados*.* 21](#_Toc52375058)

[APENDICE D: exemplo de *script* para formatação de *encode text.* 22](#_Toc52375059)

[APENDICE E: exemplos de consulta e *procedure in query language* do banco de dados (MySQL)*.* 23](#_Toc52375060)

[APENDICE F: Arquivo .bat para agendamento de tarefas 23](#_Toc52375061)

# INTRODUÇÃO

## Objetivos gerais:

* Desenvolver um método para coletar, tratar e disponibilizar os dados relacionados à gamificação de modo automatizado.
* Criar uma plataforma acessível para permitir que os estudantes vejam seu andamento em atividades de gamificação.
* Desenvolver uma forma que permita aos estudantes avaliarem a plataforma de gamificação

## Objetivos específicos:

Utilizando *scripts* em *Python* para obter informações sobre os *logs* (log de dados é uma expressão utilizada para descrever o processo de registro de eventos relevantes num sistema computacional.) dos estudantes de cursos EAD. Tratar esses dados, removendo usuários que não interessam as atividades de gamificação como, por exemplo: professores e tutores. Em seguida normalizar os dados exibidos nos *logs*(registros diários em forma de relatorios)*,* caso haja alguma inconsistência. Para que desta forma seja possível, por fim inseri-los em um banco onde serão agrupados e exibidos para quem possa interessar essa informação, sendo que até o momento, nessa etapa do processo foi definido que diferentes informações terão diferentes níveis de acesso, por exemplo, o aglomerado de ações de uma instituição será visualizada pela diretoria da instituição enquanto as realizações individuais estarão disponíveis apenas aos próprios estudantes que cumpriram as atividades EAD, através de aplicação especifica, sendo esta com a utilização de *JavaScrypt* e suas bibliotecas em especial *React*.

Garantindo que as atividades realizadas sejam atualizadas diariamente, utilizando funcionalidades tanto na forma de *scripts* em *Python* como *procedures* armazenadas no banco de dados, para que os alunos focados em realizar suas atividades consigam acompanhar o seu andamento ao longo do curso, permitindo também que observem seu próprio desenvolvimento em relação a metas estabelecidas e em relação a outros colegas de curso e instituição; preservando, obviamente, a privacidade e a confidencialidade de todos os estudantes envolvidos.

Após a plataforma de gamificação ser disponibilizada, definir mecanismos para coletar o *feedback* dos estudantes de modo a compreender como eles veem a implementação dessas atividades de gamificação e a satisfação com a plataforma e a metodologia empregada.

# JUSTIFICATIVA:

Tendo em foco os objetivos que se pretendem aplicar projetos de gamificação, alguns dos mais pertinentes em relação a plataformas educacionais é permitir que o usuário tenha como manter em vista seus objetivos alcançáveis e os pequenos resultados que irão construir um resultado maior, além disso sempre é interessante que seja possível ao usuário enxergar pontos que possam ser melhorados a partir de metas básicas e objetivos de realização de atividades. Com essas demandas em mente faz-se necessário a implementação de uma plataforma segura e confiável para que os usuários, nesse caso especifico, estudantes EAD possam visualizar, comparar e acompanhar metas em seus cursos de modo a fazer o melhor aproveitamento possível de suas disciplinas cursadas.

Devido a especificidades do sistema e do volume de dados gerados diariamente por uma instituição de ensino superior contando com mais de vinte mil estudantes matriculados e ativamente engajados em atividades virtuais, podem ocorrer dias em que os registros ou logs de atividades ultrapassem cinquenta mil registros diariamente apenas se observarmos atividades pertinentes a tarefas definidas como interessantes para contabilizar pontos em gamificação. Em especifico, para esse projeto serão definidos pesos diferentes para diferentes atividades de criação de conteúdo por parte dos estudantes a ser exibido posteriormente na seção metodologia. Com isso, é possível que a instituição de ensino e os próprios estudantes sejam capazes de observar os avanços alcançados em atividades consideradas fundamentais para a sua formação e sua contribuição para a construção de conhecimento como um todo.

# Fundamentação teórica:

Nesse ponto encontram-se as etapas, informações pertinentes , tecnologias e técnicas empregadas no desenvolvimento desse trabalho, acompanhado de uma sucinta descrição dos mesmos.

## Definição de regras de gamificação:

Uma etapa que precede a realização da aplicação de métodos de gamificação é a seleção de regras e elementos de interesse para a aplicação.

Tendo como público-alvo o corpo estudantil.

E a realização de atividades virtuais na plataforma online da instituição de ensino como comprovação de ação concluída.

Para isso a opinião do corpo docente envolvido na criação de atividades foi consultada, permitindo assim a elaboração do que serão consideradas como ações validas por parte dos alunos e seu impacto na participação do aluno ao longo do período de aulas que será observado, analisado e exibido aos alunos como seus resultados na gamificação.

Portanto,as atividades que necessitam ser realizadas por parte dos alunos e definidas como essenciais, escolhidas pelo corpo docente, tanto para o aprendizado do aluno como para a contabilização de pontos na aplicação, essa pontuação resultante; por sua vez, será exibida aos alunos individualmente.

Na tabela abaixo é possível visualizar quais atividades serão consideradas pertinentes para a contabilização de pontos concedidos pela plataforma de gamificação, a quantidade de pontos a serem dados a cada interação em especifico e suas metas por atividade.

|  |  |  |
| --- | --- | --- |
| **ATIVIDADE** | **PONTOS** | **METAS** |
| Tirar dúvidas com tutores e professores | 1 | 10 interações |
| Participação nos fóruns temáticos das disciplinas | 2 | 15 interações |
| Participação no fórum de café virtual | 2 | 5 interações |
| Interagir na opção “Bloco de Notas” | 5 | 20 interações |
| Interagir na opção “Partilhando Ideias” | 5 | 20 interações |
| Envio de atividade realizada da webconferência | 20 | 4 atividades |
| Interagir na opção “Contribua com um documento” | 10 | 10 documentos |
| Envio de certificado de realização de curso livre | 40 | 2 atividades |

**Tabela 1 – pontuações por cada atividade, definidas pelo corpo docente da instituição.**

## Obtenção de dados:

Uma vez definidas quais atividades devem ser consideradas adequadas para atribuir pontuação da gamificação. A atividade a ser realizada é atender a demanda de obtenção das informações pertinentes a respeito das atividades realizadas pelos estudantes de modo seguro, constante e devidamente tratado para que possam ser exibidos com confiabilidade em uma plataforma atraente, inequívoca e confortável para os usuários; dito isto, inclui-se também relembrar e incentivar a respeito das atividades fazendo uso de metas de objetivos.

Devido ao grande volume de dados (em alguns dias mais de cinquenta mil registros serão tratados e inseridos) e especificidades da situação em que seria aplicada, além da impossibilidade para obter os dados diretamente do banco de dados, a opção é acessar a massa de dados compilada em sua forma bruta que está disponível em forma de relatórios gerados em uma pagina da instituição de ensino, todavia, seu volume massivo tornava inviável a analise diária por um usuário humano.

## Tratamento e filtragem inicial dos dados:

Após obtenção dos dados é necessário tratar-los, corrigir e formatar de modo que possam ser utilizados de modo confiável, com este objetivo em mente, é pertinente comentar a respeito das diversas funcionalidades disponíveis em conjunto com a linguagem *Python* na forma de bibliotecas e módulos que permitem a automação de atividades e o tratamento de dados; optando-se por utilizar o *framework* *Selenium*[1] de modo a criar um *script web crawler (*rastreador ou rastejador web*), que de acordo com VandanaShrivastava* [6]*:*

é um programa que navega a web e realiza o download das referencias de paginas web. Motores de busca executam diversas instancias de rastreadores em uma platitude de servidores para obter informação diversificada para eles. O *web crawler* rastreia de uma pagina a outra na internet, coletando a pagina *web*, carregando o conteúdo da pagina no banco de dados do motor de busca e atribuindo *index* ao mesmo. *Index*, por sua vez, é um imenso banco de dados de palavras e textos que existem em uma pagina web diferente

Para a criação desta ferramente foi escolhida a linguagem *Python,* pois a linguagem oferece um conjunto de soluções aplicáveis segura e com uma curva de aprendizado suave, para a aplicação automatizada de navegar em uma pagina selecionada e realizar atividades de *webscraping* dos dados desejados.

*Webscraping,* que de acordo com Karthikeyan[8]:

É a técnica de extrair informações de vários documentos web automaticamente. Ela obtem os conteúdos baseados na query, agrupa e transforma os dados de um formato desestruturado em uma representação estruturada. Classificação de texto se torna uma fase vital para sumarizar a informação e em categorizar as paginas web de modo adequado.

Ou seja, coletar informações de uma pagina “raspando” o conteúdo web, Disso vem o nome, *Webscraping*. Algo que na situação especifica aplicada nesse projeto constitui-se em obter o relatório diário de atividades realizadas nas plataformas virtuais da instituição, salvando posteriormente esse relatório em um arquivo e realizando o tratamento e classificação inicial, para que possa essas informações recebam o tratamento final em uma etapa posterior, sendo que a plataforma utilizada para a coleta desses dados já disponibiliza o *download* dessas informações como relatórios em formato de ‘*.csv*’ (*commas-separeted values:* valores separados por virgula); isso permite uma interação mais simples por parte do script, caso não houvesse essa opção, ainda assim é relativamente simples adicionar essas funcionalidades com o uso de bibliotecas python como pandas e csv, que também serão usadas em etapas posteriores para o tratamento de dados

## Estabelecendo um banco de dados com as informações coletadas:

Ao concluir-se essa etapa o procedimento seguinte é tratar os dados, para isso Python fornece o modulo *csv*[2] que permite a manipulação de diversos arquivos ‘*.csv’* para remover usuários que não interessam a gamificação no caso em especifico professores, as diversas atividades que fogem do conjunto definido devem ser removidas também e por fim a padronização dos registros pertinentes deve ser garantido para evitar inconsistências nos dados. Depois disso é possível fazer o upload dessas informações em um banco de dados, optou-se para esse projeto por uma vps (virtual private server – servidor privado virtual) na forma de uma droplet na Digital Ocean, que trata-se de uma maquina virtual baseada em Linux executadas em um hardware virtualizado, sendo que cada Droplet criado é um novo servidor podendo ser usado de modo independente ou como parte de uma estrutura maior baseada em nuvem[3].

### Python:

Python é uma linguagem de programação de alto nível, lançada em 1991, por Guido Van Rossum. Tendo como principais objetivos produtividade e legibilidade. Recebe bastante destaque em aplicações de automação, analise de dados, devido ao abundante ecossistema de recursos, bibliotecas e pacotes disponíveis para essas atividades em Python[12]

### Pandas

Em analise de dados com python existem varias bibliotecas que se destacam, dentre essas bibliotecas, Pandas é uma das que mais se destaca, desenvolvida em 2008 e lançada com código aberto em 2009 pelo *AQR Capital Management*, essa biblioteca se destaca tanto no ambiente acadêmico quanto no comercial por oferecer um *dataframe* (sistema matricial de organização de dados, semelhante a planilhas) rápido e eficaz, ferramentas para leitura e escrita de dados, alinhamento de dados, tratamento de dados incompletos ou ausentes, formatação flexível, fatiamento, subdivisão, agrupamento, hierarquia de eixos [13]

### Selenium

### Armazenamento dos dados:

### MySQL

### Droplet

## Criação de *API rest*

*Rest*

### Node.js

Node.js é um ambiente de execução *javascript, open-source cross-plataform* que atua a nível *frontend*[9] (responsável por organizar as entradas do usuário em varias formas e processa-las para uma especificação adequada ao *backend* ) e *backend* (processar as solicitações do *frontend*). Desse modo

### EXPRESS

O Express é um framework para aplicativo da web do Node.js mínimo e flexível que fornece um conjunto robusto de recursos para aplicativos web e móvel.[10]

Permitindo a criação de uma API de modo rápido e fácil.

### *React-native*

## *Login* a partir do consumo de *API* parceira

Existem, basicamente, três tipos de API[11]:

* privada: que são utilizadas internamente entre as aplicações de uma empresa, ou seja, de forma local;
* parceira: que são utilizadas entre parceiros de negócios ou para permitir a integração entre diferentes softwares;
* pública: que podem ser utilizadas livremente. Muitas vezes são disponibilizadas por empresa para que os desenvolvedores possam fazer a integração com outras aplicações

Para oferecer conforto e segurança aos alunos ,quando os mesmos forem fazer uso da aplicação, a opção de fazer uso de informações de credenciais já utilizadas em outros sistemas da instituição de ensino foi escolhida. Como trata-se de uma API desenvolvida e disponibilizada por um setor diferente setor da empresa, trata-se de uma API parceira.

Para isso na aplicação utiliza-se o CORS que permite o consumo desses recursos, a partir da API, sem por em risco a integridade dos dados em ambas as partes.

### *CORS*

Inicialmente o estudante deve fazer login com suas credenciais, por questões de conforto e segurança os dados de matricula e senha foram obtidos utilizando uma *api* de conexão utilizada em outros sistemas da instituição de ensino, para isso é utilizado o *CORS* (compartilhamento de recursos de origem cruzada)

## Aplicativo para exibição aos estudantes

Nas imagens a seguir é possivel observar as telas implementadas para a utilização por parte dos estudantes,

Na imagem abaixo vemos a tela de login onde iremos comparar os dados fornecidos pelo usuário com os dados armazenados em outros bancos da instituição.

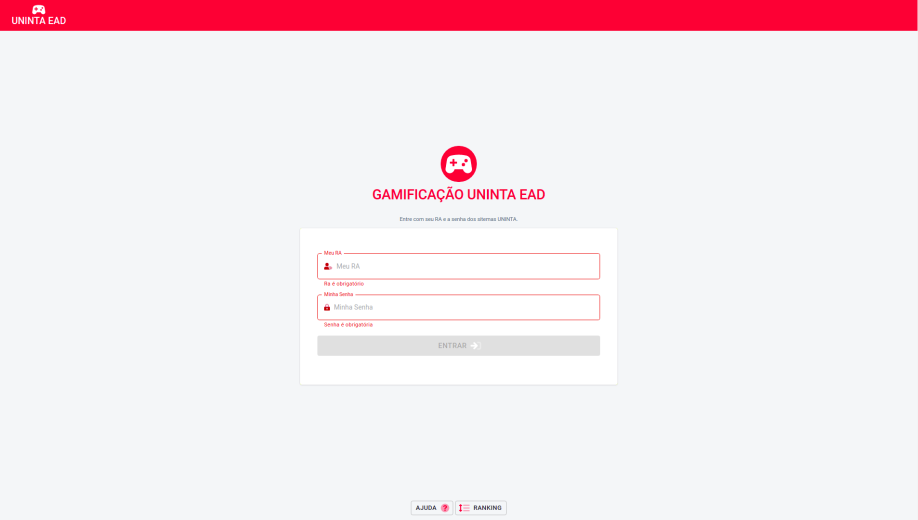


Figura 1 tela de login.

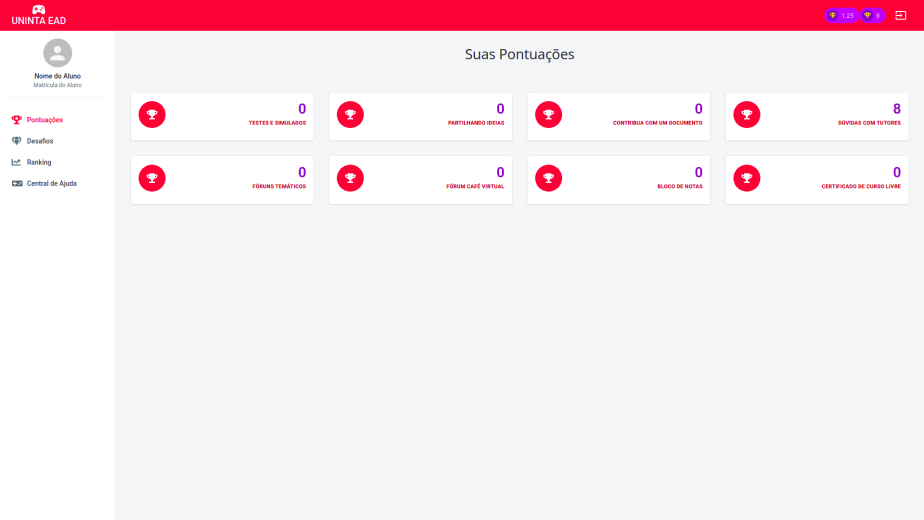


Figura : pagina com a listagem de pontuações do estudante.

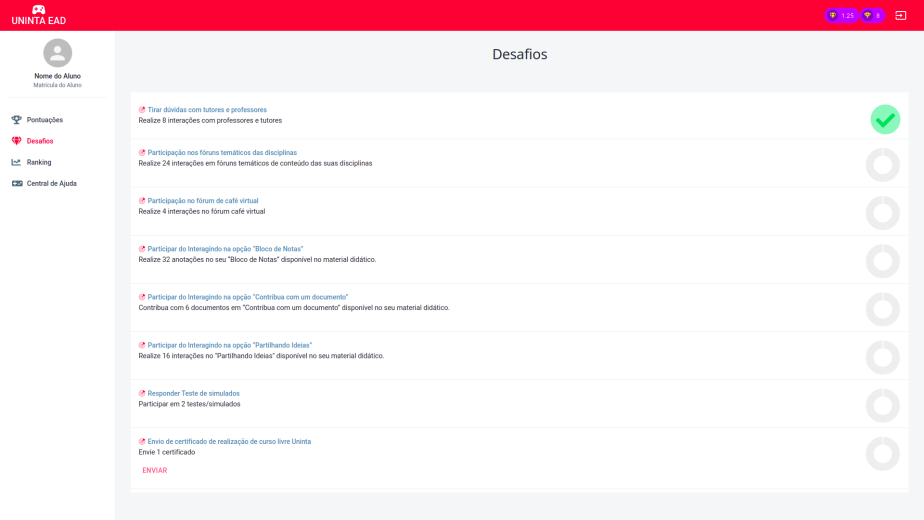


Figura : pagina listando os desafios e o nível de conclusão dos mesmos.

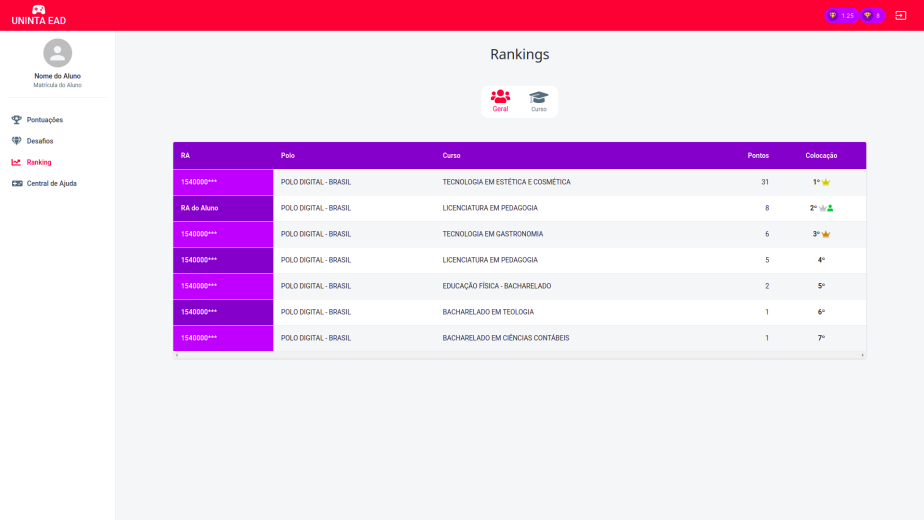


Figura : tela de ranking geral em relação ao polo de ensino, com o registro identificador dos outros alunos parcialmente oculto.

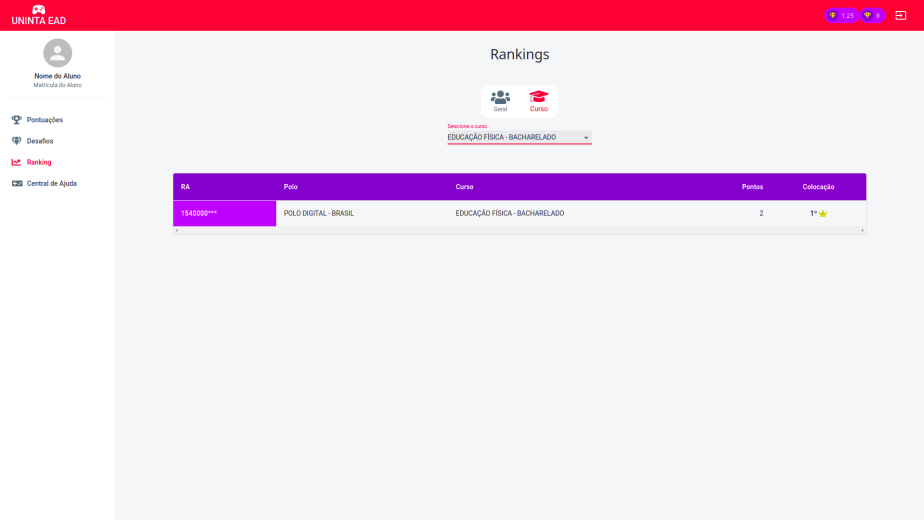


Figura : tela de ranking em relação ao curso, com o registro identificador dos outros alunos parcialmente oculto.

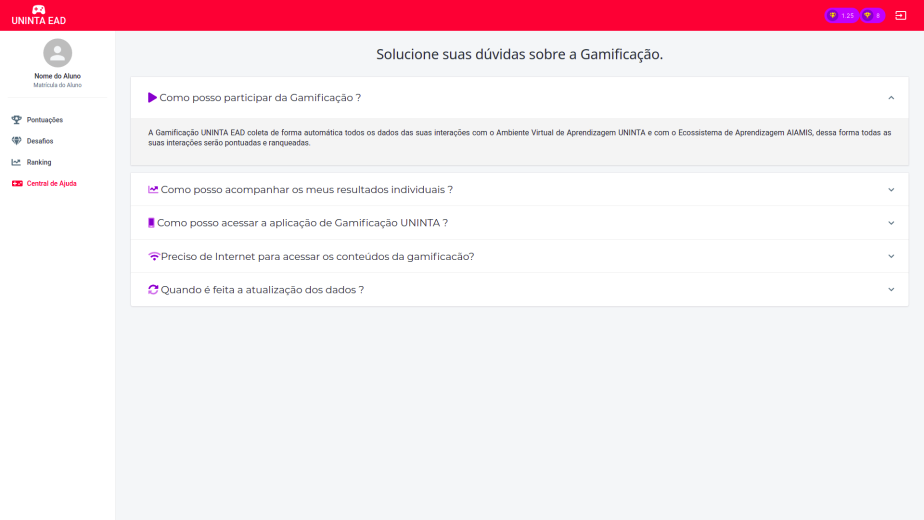


Figura : Tela de informações adicionais e seção tira duvidas

# METODOLOGIA

## 

## Coleta de feedback

## 

## Analise quantitativa

# 

# RESULTADOS

# 

# CONCLUSÃO

# REFERENCIAS

[1]Documentação relacionada ao webdriver Selenium, voltado para automação - https://selenium-python.readthedocs.io/

[2]Documentação modulo csv para Python - https://docs.python.org/3/library/csv.html

[3]Documentação digital ocean - <https://www.digitalocean.com/docs/droplets/>

[4]Documentação JavaScript

[5]documentação React - <https://pt-br.reactjs.org/docs/getting-started.html>

[6]Z. Guojun, J. Wenchao, S. Jihui, S. Fan, Z. Hao and L. Jiang, "Design and application of intelligent dynamic crawler for web data mining," 2017 32nd Youth Academic Annual Conference of Chinese Association of Automation (YAC), Hefei, 2017, pp. 1098-1105, doi: 10.1109/YAC.2017.7967575.

[7] DE MATTOS MARQUES, Clarice Lúcia; ROSA, Aline Verneque. A GAMIFICAÇÃO COMO ESTRATÉGIA FACILITADORA NO PROCESSO ENSINO-APRENDIZAGEM NO ENSINO SUPERIOR: UMA REVISÃO DE LITERATURA. In: **III-Semana Acadêmica-Campus Mesquita**. 2017.

[8] Karthikeyan, T., et al. "Personalized content extraction and text classification using effective web scraping techniques." International Journal of Web Portals (IJWP) 11.2 (2019): 41-52.

[9] NODE.JS: <https://developer.ibm.com/technologies/node-js/>

[10] EXPRESSJS: <https://expressjs.com/pt-br/>

[11] API REST: <https://rockcontent.com/br/blog/rest-api/>

[12] Why Python is Good for Data Analytics?:

<https://thriveglobal.com/stories/why-python-is-good-for-data-analytics/#:~:text=Python%20is%20one%20of%20the,a%20lot%20quick%20and%20convenient>.

[13]Pandas: https://pandas.pydata.org/about/index.html

# APENDICE A: exemplo de *script* para *webscrap.*

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

from selenium.common.exceptions import TimeoutException

from selenium.webdriver.common.by import By

import selenium.webdriver.support.ui as ui

from selenium.webdriver.support.ui import Select, WebDriverWait

import selenium.webdriver.support.expected\_conditions as EC

import os

import time

from datetime import datetime as dt

# a ausencia de estado em alguns usuarios estava quebrando o procedure resultado\_identifica\_por\_cpf

# e todos os posteriores

class forumTesteBot():

"""docstring for forumTesteBot"""

def \_\_init\_\_(self):

options = webdriver.ChromeOptions()

options.add\_argument('--ignore-certificate-errors')

options.add\_argument('--ignore-ssl-errors')

dir\_path = os.path.dirname(os.path.realpath(\_\_file\_\_))

chromedriver = dir\_path + "/chromedriver"

os.environ["webdriver.chrome.driver"] = chromedriver

self.driver = webdriver.Chrome(

options=options, executable\_path=chromedriver)

def timerPratice(self):

time.sleep(3)

def gotosite(self):

self.driver.get("http://pagina para login")

# USER - identificação do elemento

ui.WebDriverWait(self.driver, 15).until(

EC.visibility\_of\_element\_located((By.ID, "username")))

#preenchimento

self.driver.find\_element\_by\_id("username").send\_keys('nome de usuario')

# PASSWORD - identificação do elemento

ui.WebDriverWait(self.driver, 15).until(

EC.visibility\_of\_element\_located((By.ID, "password")))

#preenchimento

self.driver.find\_element\_by\_id("password").send\_keys('senha')

time.sleep(3)

ui.WebDriverWait(self.driver, 15).until(

EC.visibility\_of\_element\_located((By.ID, "btn-login")))

self.driver.find\_element\_by\_id("btn-login").click()

time.sleep(5)

def gotoreport(self):

self.driver.get(

"http://pagina onde estão contidas as informações")

time.sleep(2)

def setData(self):

# entrar data e tipo de ação

# escolher a ação

acao = Select(ui.WebDriverWait(self.driver, 15).until(

EC.visibility\_of\_element\_located((By.ID, "menumodaction"))))

acao.select\_by\_value("c")

# DIA - usar dia anterior

dataDoTrem = Select(ui.WebDriverWait(self.driver, 15).until(

EC.visibility\_of\_element\_located((By.ID, "menudate"))))

#diaLa = "1568602800"

now = dt.now()

dia = dt.date(now)

stamp = time.mktime(dia.timetuple())

afinal = stamp - 86400

diaInt = int(afinal)

diaLa = str(diaInt)

dataDoTrem.select\_by\_value(diaLa)

time.sleep(3)

# botao: obter esses logs

ui.WebDriverWait(self.driver, 15).until(EC.visibility\_of\_element\_located(

(By.XPATH, '/html/body/div[4]/div/div/section/div/form/div/input[5]')))

self.driver.find\_element\_by\_xpath(

"/html/body/div[4]/div/div/section/div/form/div/input[5]").click()

time.sleep(5)

print("wait for it")

def trazPraCasa(self):

time.sleep(2)

print("a espera acabou")

# o botao para download era apenas <input type="submit" value="Download"> por isso .submit e nao .click

ui.WebDriverWait(self.driver, 15).until(EC.visibility\_of\_element\_located(

(By.XPATH, '/html/body/div[4]/div[1]/div[2]/section/div/form[2]')))

#self.driver.timeouts().implicitlywait(15) /html/body/div[4]/div[1]/div[2]/section/div/form[2] /html/body/div[4]/div[1]/div[2]/section/div/form[2]/div

# /html/body/div[4]/div[1]/div[2]/section/div/form[2]/div/button

self.driver.find\_element\_by\_xpath(

'/html/body/div[4]/div[1]/div[2]/section/div/form[2]/div/button').submit()

# mudança no xpath do formulario de download

#

time.sleep(10)

def teardowm(self):

print("ta la")

time.sleep(160)

self.driver.close()

if \_\_name\_\_ == '\_\_main\_\_':

obj = forumTesteBot()

obj.timerPratice()

obj.gotosite()

obj.gotoreport()

obj.setData()

obj.trazPraCasa()

obj.timerPratice()

obj.teardowm()

# APENDICE B: exemplo de *script* para tratamento de dados*.*

import csv

import codecs

import pandas as pd

import tempfile

import shutil

from tempfile import NamedTemporaryFile

import time

import fnmatch

import os

from datetime import date, timedelta

today = date.today()

diaUsado = today.strftime('%Y%m%d')

# print(diaUsado)

print("logs\_"+diaUsado)

diaInt = int(diaUsado)

diaLa = str(diaInt)

# print(diaLa)

#identificado o diretório que se encontra o relatorio

for NomeRelatorio in os.listdir(r'C:\Users\home\Downloads'):

if fnmatch.fnmatch(NomeRelatorio, "logs\_"+diaLa+"\*.csv"):

print(NomeRelatorio)

a = NomeRelatorio

#transferimos para o diretório desejado

cwd = os.getcwd() # Get the current working directory (cwd)

files = os.listdir(cwd) # Get all the files in that directory

print("Files before in %r: %s" % (cwd, files))

print(a)

# time.sleep(5)

shutil.move("C:/Users/home/Downloads/"+a,

"C:/Users/home/OneDrive/projetosUninta/gamefication/"+a)

time.sleep(5)

# cwd = os.getcwd() # Get the current working directory (cwd)

files = os.listdir(cwd) # Get all the files in that directory

#print("Files after in %r: %s" % (cwd, files))

#com csv abrimos o arquivo e iniciamos a remoção de colunas desnecessarias

with open(a, 'r', encoding='utf-8') as file:

reader = csv.DictReader(file)

# print("reeeeeee")

with open('saiuPtBR2.csv', 'w', newline='', encoding='utf-8') as file:

col = ["\ufeffHora", "Nome completo",

"Contexto do Evento", "Nome do evento"]

writer = csv.DictWriter(file, fieldnames=col)

headers = "Hora,NomeCompleto,ContextoEvento,NomeEvento\n"

file.write(headers)

# removendo colunas indesejadas

for line in reader:

del line['Usuário afetado']

del line['Componente']

del line['Descrição']

del line['Origem']

del line['endereço IP']

writer.writerow(line)

#

# removemos também os usuários que não nos interessam

takeThis = ["Webservice User", "-"]

with open('saiuPtBR2.csv', 'r', encoding='utf-8') as infile, open('saiuPtBR3.csv', 'w', newline='', encoding='utf-8') as outfile:

writer = csv.writer(outfile)

for line in csv.reader(infile, delimiter=','):

if not any(takeThis in line for takeThis in takeThis):

writer.writerow(line)

# pandas was used to correct some lines

#keepThis = ['Algum conteúdo foi publicado', 'Journal entry created', 'Mensagem enviada', "Registro criado", "Tentativa do questionário iniciada"]

# "Base de dado Contribua com um documento","Fórum – Café Virtual" agora estao pelo sistema curadoria

keepThis = ["Fórum de discussão ", "Diário: Bloco de Notas", "Fórum Temático", "Fórum: Fórum Partilhando ideias", "Questionário: Simulado", "Sistema"]

#forum = ["Fórum temático", "Fórum : Fórum de Temático 2","Fórum : Fórum de Temático 1"]

time.sleep(2)

# usando pandas para renomear adequar e normalizar algumas informações de modo a garantir a padronizaçao

filename = 'saiuPtBR3.csv'

filenameout = 'saiuPtBR.csv'

data = pd.read\_csv(filename)

data["ContextoEvento"] = data["ContextoEvento"].replace(

"Fórum: Fórum - Café Virtual", "Fórum - Café Virtual")

# metodo que localiza e substitui baseado em string parcial no tocante # contexto de evento ou nome de tutor como por exemplo questionario: simulado

# NO CASO DE ALGUNS TUTORES ESPECIFICOS QUE PODERIAM TER O NOME #ESCRITO EM MAIS DE UMA FORMA NO SISTEMA(POR QUESTÕES DE PRIVACIDADE #OS NOMES FORAM SUBSTITUIDOS)

data.loc[data['NomeCompleto'].str.contains(

'TUTOR - FM', case=False), 'NomeCompleto'] = "TUTOR FM"

data.loc[data['NomeCompleto'].str.contains(

'TUTORA EAD - SM', case=False), 'NomeCompleto'] = "TUTORA EAD - SM"

data.loc[data['NomeCompleto'].str.contains(

'Tutor - FDOA', case=False), 'NomeCompleto'] = "Tutor - FDOA"

# contexto evento

data.loc[data['ContextoEvento'].str.contains(

'questionário: simulado', case=False), 'ContextoEvento'] = "Questionário: Simulado"

data.loc[data['ContextoEvento'].str.contains(

'Diário: Bloco de notas', case=False), 'ContextoEvento'] = "Diário: Bloco de Notas"

data.loc[data['ContextoEvento'].str.contains(

'Fórum: FÓRUM TEMÁTICO', case=False), 'ContextoEvento'] = "Fórum Temático"

data.loc[data['ContextoEvento'].str.contains(

'Fórum: FÓRUM TEMÁTICa', case=False), 'ContextoEvento'] = "Fórum Temático"

data.loc[data['ContextoEvento'].str.contains(

'Fórum: FÓRUM TEMÁTICO', case=False), 'ContextoEvento'] = "Fórum Temático"

data.loc[data['ContextoEvento'].str.contains(

'Fórum: FÓRUM de TEMÁTICO', case=False), 'ContextoEvento'] = "Fórum Temático"

data.loc[data['ContextoEvento'].str.contains(

'Fórum: Partilhando ideias', case=False), 'ContextoEvento'] = "Fórum: Partilhando ideias"

data.loc[data['ContextoEvento'].str.contains('Tarefa: Enviar relato de experiência de utilização profissional de algum recurso do Ecossistema de Aprendizagem AIAMIS',

case=False), 'ContextoEvento'] = "Tarefa: Enviar relato de experiência de utilização profissional"

data.loc[data['ContextoEvento'].str.contains('Tarefa: Envio de atividade realizada da webconferência',

case=False), 'ContextoEvento'] = "Tarefa: Envio de atividade na webconferência"

data.loc[data['ContextoEvento'].str.contains(

'Tarefa: Envio de certificado de realização de curso livre', case=False), 'ContextoEvento'] = "Curso Livre"

data.to\_csv(

r"C:\Users\home\OneDrive\projetosUninta\gamefication\saiuPtBR2.csv", index=False)

filename = 'saiuPtBR2.csv'

data = pd.read\_csv(filename)

# pandas called in itself to selfload with the isin "method"

# mantem linhas que interessam em contexto do evento

#"Tarefa: Enviar relato de experiência de utilização profissional","Tarefa: Envio de resumo de palestra online", "Tarefa: Envio de atividade na webconferência"

#"Base de dados: Contribua com um documento","Diário: Bloco de Notas",

CE = data[data['ContextoEvento'].isin(["Fórum - Café Virtual", "Fórum Temático", "Fórum: Partilhando ideias", "Questionário: Simulado", "Sistema",

"Tarefa: Envio de certificado de realização de curso livre"])]

# mantem linhas q interessam em nome do evento após a filtragem de contexto do evento feita anteriormente

NE = CE[CE['NomeEvento'].isin(['Algum conteúdo foi publicado.', 'Journal entry created', 'Mensagem enviada',

"Registro criado", "Tentativa do questionário iniciada", "Um arquivo foi enviado."])]

# NE.shape

# CONFIRMAR A PASTA EM QUE SERÁ SALVA E SEU NOME

# para debug usar nome q nao caia na pesquisa por arquivo ou colocar em outra pastas, no caso esta em outra pasta

NE.to\_csv(r"C:\Users\home\OneDrive\projetosUninta\gamefication\logs\_"+diaLa+".csv", index=False)

print("it´s over")

# APENDICE C: exemplo de *script* para Inserção em banco de dados*.*

import csv

import pymysql

import pandas as pd

import sys

import os

import time

from datetime import date, timedelta

today = date.today()

diaUsado = today.strftime('%Y%m%d')

print(diaUsado)

print("logs\_"+diaUsado)

diaInt = int(diaUsado)

diaLa = str(diaInt)

print(diaLa)

with open("logs\_" + diaLa + ".csv", "r", encoding='utf-8') as logs:

ler = pd.read\_csv(logs)

time.sleep(3)

# Renomeia os arquivos tratados para logs.csv

for nome in os.listdir('.'):

if (nome == "logs\_" + diaLa + ".csv"):

os.rename("logs\_" + diaLa + ".csv", "logs.csv")

# Insere as tabelas no Banco de Dados SELECIONADO

def csv\_to\_mysql(load\_sql1, host):

'''

Esta função transfere informações de uma tabela no formato CSV para SQL de acordo com a variável load\_sql

'''

try:

con = pymysql.connect(host='ENDEREÇO DO HOST',

port=3306,

user='NOME DO USUARIO',

password='SENHA',

db='NOME DO BANCO DE DADOS',

autocommit=True,

local\_infile=1,

cursorclass=pymysql.cursors.DictCursor)

print('Conectando ao Banco de Dados: {}'.format(host))

cursor = con.cursor()

cursor.execute(load\_sql1)

print('Sucesso na transferência para a tabela relat\_tratados.')

con.close()

except Exception as e:

print('Error: {}'.format(str(e)))

sys.exit(1)

# Execution Example

load\_sql1 = "LOAD DATA LOCAL INFILE 'logs.csv' INTO TABLE NOME\_DA\_TABELA CHARACTER SET utf8mb4 FIELDS TERMINATED BY ',' ENCLOSED BY '\"' LINES TERMINATED BY '\n' ignore 1 lines;"

host = '142.93.244.120'

csv\_to\_mysql(load\_sql1, host)

time.sleep(3)

print("time to kill this buckos: calculado.csv,logs.csv")

for nome in os.listdir('.'):

if (nome == "logs.csv"):

os.rename("logs.csv", "logs\_"+diaLa+".csv")

print("Arquivos renomeados com sucesso!!!")

time.sleep(5)

# APENDICE D: exemplo de *script* para formatação de *encode text.*

Por vezes surge a necessidade de um tratamento adicional, pois os arquivos estão em um *encode* incompatível com os disponíveis para o projeto ou com uma *collation* diferente da disponível no banco de dados o que resulta em uma camada adicional de tratamento.

|  |
| --- |
| import csv  import pymysql  import pandas as pd  import sys  import os  import time  import codecs  blockSize = 1048576  with codecs.open("dados alunos\_21\_09.csv", "r", encoding="mbcs") as sourceFile:  with codecs.open("Dados2020a.csv", "w", encoding="UTF-8") as targetFile:  while True:  contents = sourceFile.read(blockSize)  if not contents:  break  targetFile.write(contents) |

# APENDICE E: exemplos de consulta e *procedure in query language* do banco de dados (MySQL)*.*

Query simples:

SELECT \* FROM `NOME\_TABELA`

Query armazenada para ser utilizada como procedimento:

CREATE DEFINER=`\_gamificacao`@`localhost` PROCEDURE `NOME\_PROCEDURE ` () update resultado, (select RA\_alu,Contexto\_do\_evento, count(Contexto\_do\_evento) as somado from relat\_tratados group by RA\_alu, Contexto\_do\_evento) as t2

set resultado.contagem = IF( resultado.contagem < t2.somado OR resultado.contagem = null, t2.somado, resultado.contagem)

where resultado.RA\_alu = t2.RA\_alu

AND resultado.ContextoEv = t2.Contexto\_do\_evento$$

# APENDICE F: Arquivo .bat para agendamento de tarefas

@echo off

"C:\Users\home\AppData\Local\Programs\Python\Python37-32\python.exe" "C:\Users\home\OneDrive\projetosUninta\gamefication\5 - BotReports.py"

start /wait "C:\Users\home\AppData\Local\Programs\Python\Python37-32\python.exe" "C:\Users\home\OneDrive\projetosUninta\gamefication\5 - BotReports.py" /ALL /Quite

"C:\Users\home\AppData\Local\Programs\Python\Python37-32\python.exe" "C:\Users\home\OneDrive\projetosUninta\gamefication\tratarPlanilha.py"

timeout 5

start /wait "C:\Users\home\AppData\Local\Programs\Python\Python37-32\python.exe" "C:\Users\home\OneDrive\projetosUninta\gamefication\tratarPlanilha.py" /ALL /Quite

"C:\Users\home\AppData\Local\Programs\Python\Python37-32\python.exe" "C:\Users\home\OneDrive\projetosUninta\gamefication\contadorpy.py"

timeout 5

start /wait "C:\Users\home\AppData\Local\Programs\Python\Python37-32\python.exe" "C:\Users\home\OneDrive\projetosUninta\gamefication\contadorpy.py" /ALL /Quite

"C:\Users\home\AppData\Local\Programs\Python\Python37-32\python.exe" "C:\Users\home\OneDrive\projetosUninta\gamefication\importServer.py"

timeout 5

start /wait "C:\Users\home\AppData\Local\Programs\Python\Python37-32\python.exe" "C:\Users\home\OneDrive\projetosUninta\gamefication\importServer.py" /ALL /Quite

"C:\Users\home\AppData\Local\Programs\Python\Python37-32\python.exe" "C:\Users\home\OneDrive\projetosUninta\gamefication\callProc.py"