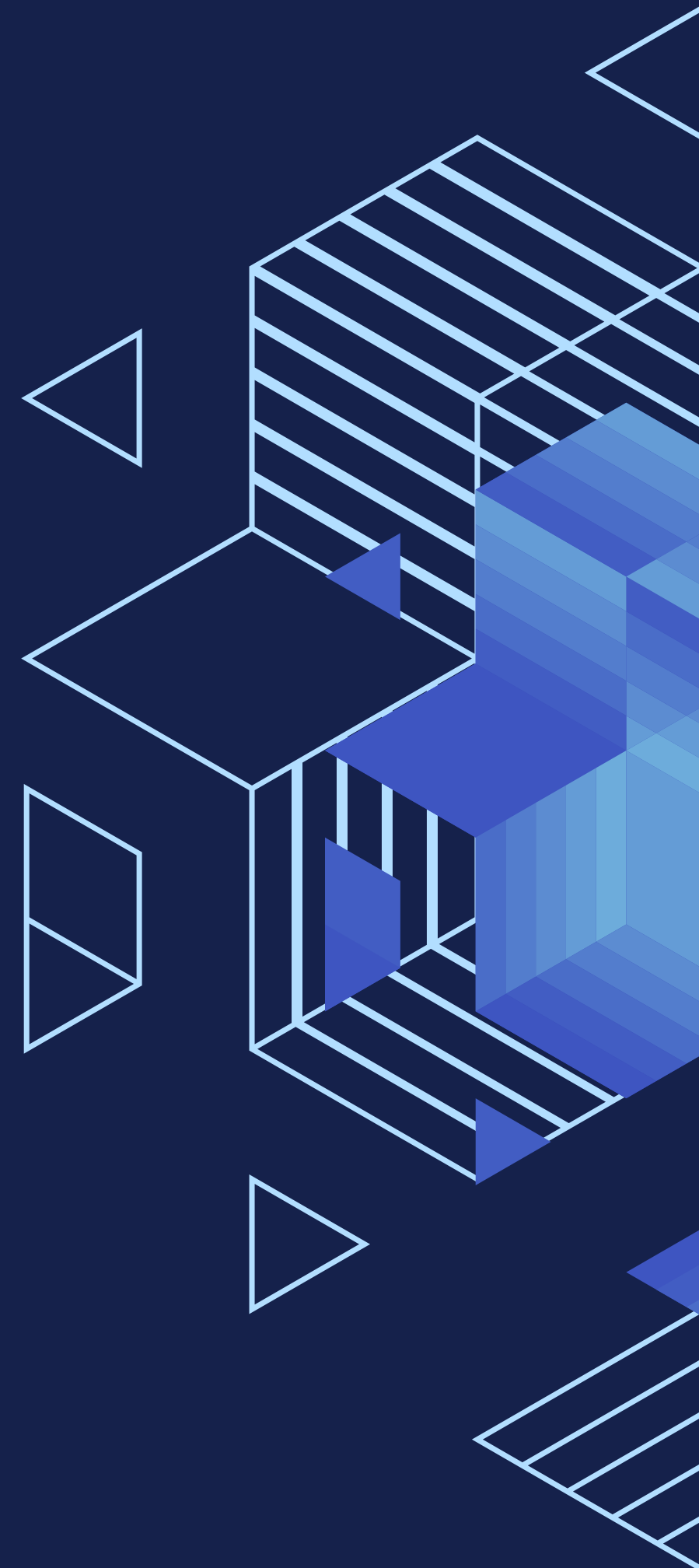


# ARCHICREATORS SPRINT 2



# Schedule

- ▶ Refinement of requirements
- ▶ Simulation
- ▶ Next steps








# Refinement of requirements



- we have added 2 requirements for the Chatbot, one for availability and the other for response time
  - entry, exit and requirement control
- 

# Simulation - ChatBot

```
[ ] stop_words = set(stopwords.words('english'))
stemmer = PorterStemmer()

def preprocess_text(text):
    tokens = word_tokenize(text) # Tokenização
    tokens = [word for word in tokens if word.isalnum()] # Removendo caracteres não alfanuméricos
    tokens = [word.lower() for word in tokens] # Convertendo para minúsculas
    # print("Tokenização: ", tokens)

    tokens = [word for word in tokens if word not in stop_words] # Removendo Stopwords
    # print("Remoção das Stopwords: ", tokens)

    stemmed_tokens = [stemmer.stem(word) for word in tokens] # Aplicando stemming
    # print("Aplicação do Stemming: ", stemmed_tokens)

    return ' '.join(stemmed_tokens)

# Aplicando o método na coluna de interesse
dataframe['GL Description'] = dataframe['GL Description'].apply(preprocess_text)

# Dividindo os dados em conjuntos de treinamento e teste
X_train, X_test, y_train, y_test = train_test_split(dataframe['GL Description'], dataframe['Level 1'], test_size=0.2, random_state=42)

# Vetorizando o texto ponderando pela frequência de ocorrência
tfidf_vectorizer = TfidfVectorizer(max_features=1000)
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
X_test_tfidf = tfidf_vectorizer.transform(X_test)
```

```
# Treinando um modelo Multinomial Naive Bayes
nb_classifier = MultinomialNB()
nb_classifier.fit(X_train_tfidf, y_train)

# Prevendo os valores do conjunto de teste
y_pred = nb_classifier.predict(X_test_tfidf)

# Avaliando o modelo
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

print(f'Acurácia do modelo: {accuracy}')
print(f'Relatório de classificação:\n{report}')
```

```
Acurácia do modelo: 0.8277851753217932
Relatório de classificação:
```

```
precision    recall  f1-score   support
```

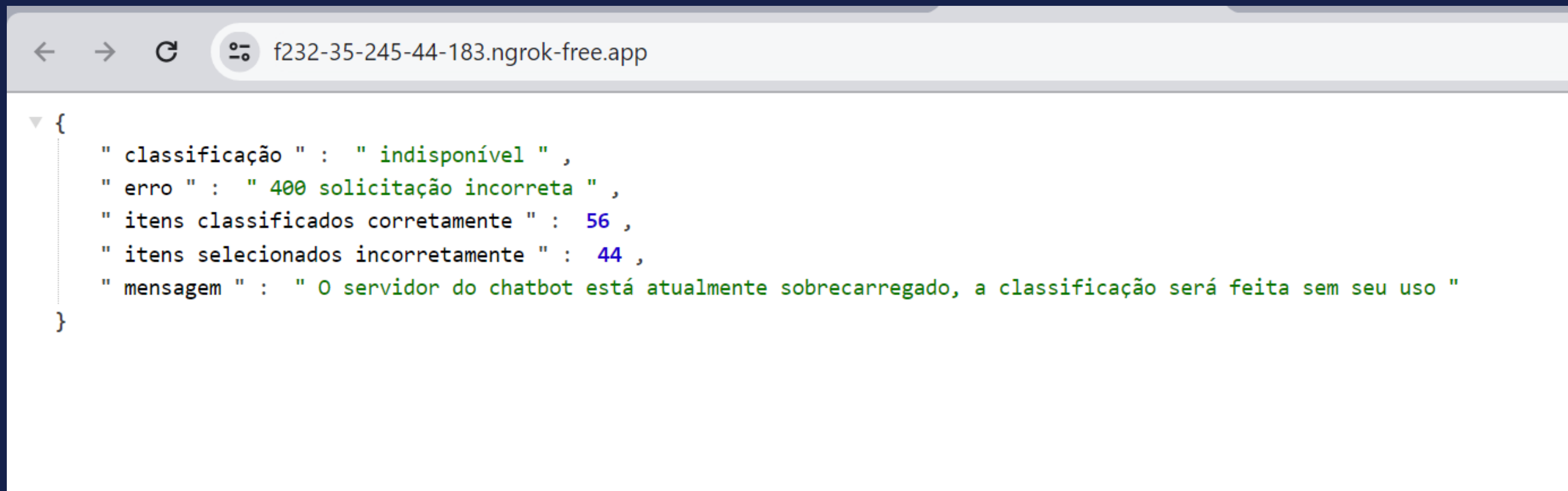
# Simulation - ChatBot

➞ Acurácia do modelo: 0.8277851753217932  
Relatório de classificação:

	precision	recall	f1-score	support
Energy & Utilities	0.00	0.00	0.00	5
Human Resources	0.68	0.62	0.65	82
Logistics	0.45	0.88	0.60	16
Manufacturing	0.00	0.00	0.00	14
Professional Services	0.95	0.92	0.94	1093
R&D Equipment (incl. Equipment Services and Supplies)	0.00	0.00	0.00	9
Real Estate & Facilities	0.69	0.89	0.78	379
Sales, Marketing & Events	0.50	0.54	0.52	80
Technology/Telecom	0.83	0.91	0.87	435
Travel & Expense	0.43	0.08	0.14	37
Uncategorized	0.50	0.09	0.15	103
accuracy			0.83	2253
macro avg	0.46	0.45	0.42	2253
weighted avg	0.81	0.83	0.81	2253

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning:
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning:
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning:
_warn_prf(average, modifier, msg_start, len(result))
```

# Simulation - ChatBot



A screenshot of a web browser window. The address bar shows the URL `f232-35-245-44-183.ngrok-free.app`. The main content area displays a JSON object with the following fields:

```
{
  "classificação" : " indisponível " ,
  "erro" : " 400 solicitação incorreta " ,
  "itens classificados corretamente" : 56 ,
  "itens selecionados incorretamente" : 44 ,
  "mensagem" : " O servidor do chatbot está atualmente sobrecarregado, a classificação será feita sem seu uso "
}
```



# Next steps



SPRINT 1



SPRINT 2



- Implementation
- Automated tests
- New simulation model review



SPRINT 4



SPRINT 5

The background is a dark blue field filled with abstract geometric patterns. At the top, there are large, complex shapes made of white outlines and blue gradients, resembling stylized cubes or architectural structures. In the center, the text 'THANK YOU!' is written in a bold, white, sans-serif font. At the bottom, there are smaller, solid blue geometric shapes, including triangles and parallelograms, arranged in a way that suggests a continuation of the top's patterns.

**THANK YOU!**