



## Trabalho TI2 - Sprint 3

**Bruno Pena Baêta**  
**Eric Azevedo de Oliveira**  
**Felipe Nepomuceno Coelho**



**bigproject**

# Conexão backend - frontend

Código em Java

```
BigProjectA conectar = new BigProjectA(); // conectar com nosso banco de dados
conectar.conectarPost();

/* codigos onde iremos colocar a pagina on */
get("/", (req, res) -> mandarSite.renderContent("/home.html"));
get("/login", (req, res) -> mandarSite.renderContent("/login.html"));
get("/register", (req, res) -> mandarSite.renderContent("/register.html"));
get("/myprojects", (req, res) -> mandarSite.renderContent("/myprojects.html"));
get("/creation", (req, res) -> mandarSite.renderContent("/creation.html"));
get("/project", (req, res) -> mandarSite.renderContent("/project.html"));
get("/mycomments", (req, res) -> mandarSite.renderContent("/mycomments.html"));
}

fetch(`http://localhost:4567/mandarRe?query=${nome},${senha},${email},${file},${tag},${10}` , methodGet)
  .then(res => res.json())
```

# Conexão backend – Banco de Dados

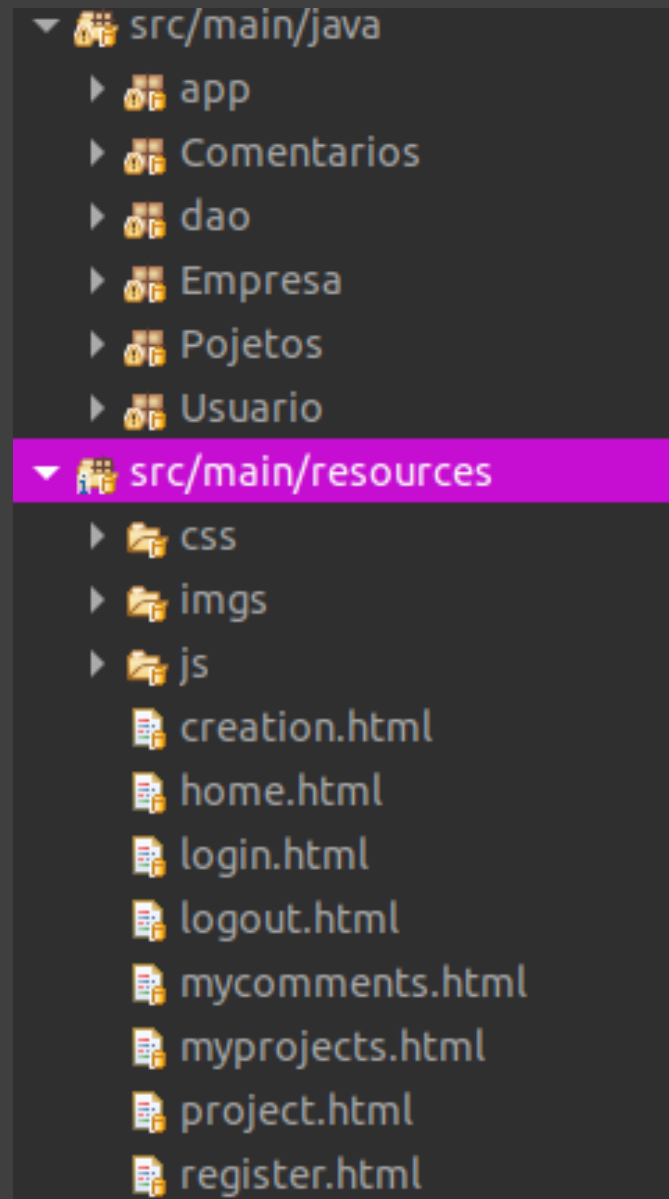
Código em Java

```
// inserir uma nova empresa em nosso BD
public void inserirEmpresa(Empresa empresa) {
    boolean saberVerdade = false;

    try {
        Statement st = conexao.createStatement();
        st.executeUpdate("INSERT INTO empresa (idempresa,nome,email,senha,imagem)" + "VALUES ("
            + empresa.getIdEmpresa() + ", '" + empresa.getNomeEmpresa() + "', '" + empresa.getMailEmpresa()
            + "', '" + empresa.getSenhaEmpresa() + "', '" + empresa.getImagemEmpresa() + "');");

        st.close();
        saberVerdade = true;
    } catch (SQLException u) {
        throw new RuntimeException(u);
    }
}
```

# Pastas do Projeto



# Declaração e Metodos Get/Set

Código em Java

```
public class Projetos {  
  
    private int idProjeto;  
    private String nomeProjeto;  
    private String descricaoProjeto;  
    private String dataInicio;  
    private String dataFim;  
    private int valorProjeto;  
    private String tag;  
    private String imagem;  
    private String rec;  
  
    public Projetos(int idProjeto1, String nomeProjeto1, String descricaoProjeto1, String dataInicio1, String dataFim1,  
        int valorProjeto1, String tag1, String imagem1, String rec) {  
        this.idProjeto = idProjeto1;  
        this.nomeProjeto = nomeProjeto1;  
        this.descricaoProjeto = descricaoProjeto1;  
        this.dataInicio = dataInicio1;  
        this.dataFim = dataFim1;  
        this.valorProjeto = valorProjeto1;  
        this.tag = tag1;  
        this.imagem = imagem1;  
        this.rec = rec;  
    }  
  
    public String getRec() {  
        return rec;  
    }  
  
    public void setRec(String rec) {  
        this.rec = rec;  
    }  
}
```



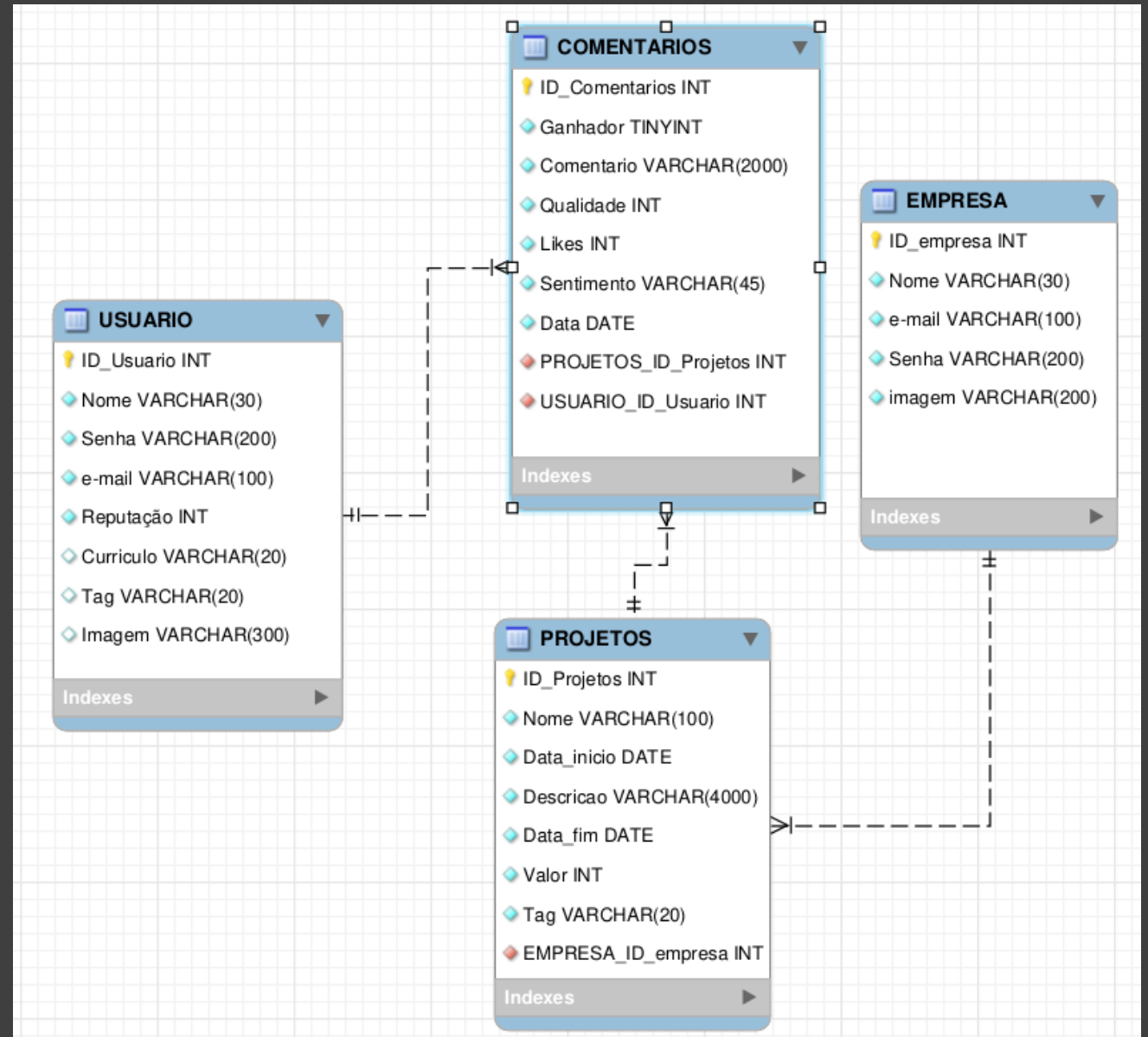
# Trabalho TI2 - Sprint 4

**Bruno Pena Baêta**  
**Eric Azevedo de Oliveira**  
**Felipe Nepomuceno Coelho**



**bigproject**

# Mudança no BD



# Demonstração Implementação do Front-end, Back-end e Banco de Dados



# Demonstração Implementação dos Recursos Inteligentes

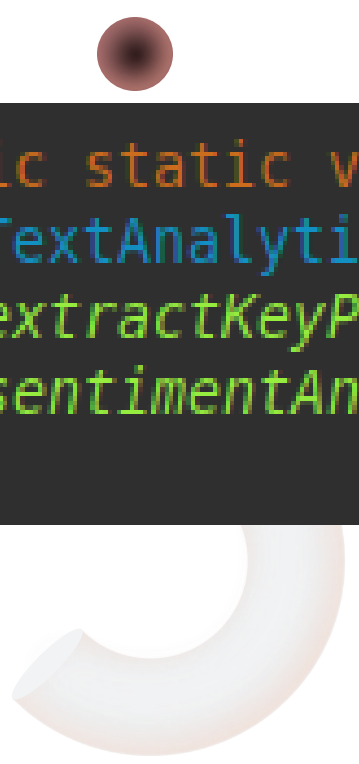
# Requisição

---

```
public static void sentimentAnalysisExample(TextAnalyticsClient client, String pato) {  
    // The text that need be analyzed.  
  
    DocumentSentiment documentSentiment = client.analyzeSentiment(pato);  
    System.out.printf(  
        "Recognized document sentiment: %s, positive score: %s, neutral score: %s, negative score: %s  
        documentSentiment.getSentiment(), documentSentiment.getConfidenceScores().getPositive(),  
        documentSentiment.getConfidenceScores().getNeutral(),  
        documentSentiment.getConfidenceScores().getNegative());  
  
    for (SentenceSentiment sentenceSentiment : documentSentiment.getSentences()) {  
        System.out.printf("Recognized sentence sentiment: %s %n", sentenceSentiment.getSentiment(),  
            sentenceSentiment.getConfidenceScores().getNegative());  
    }  
}
```

# Autenticação do Cliente

---



```
public static void fazerIA(String receber) {  
    TextAnalyticsClient client = authenticateClient(KEY, ENDPOINT);  
    extractKeyPhrasesExample(client, receber);  
    sentimentAnalysisExample(client, receber);  
}
```

<b>Ferramental de IA</b>  Nossa IA consistem em uma API que analisa um fragmento de texto e retorna uma análise de sentimento do texto.	<b>Entradas</b>  As entradas para a IA são "Comentários" que foram feitos em nosso site, em projetos de empresas	<b>Proposição de valor</b>  Facilitar a busca e análise dos comentários para a os donos dos projetos  ◦Principais pontos <ul style="list-style-type: none"><li>• Filtro baseado na análise de emoções</li><li>• Acesso a pontuação de sentimentos para os comentários</li></ul>	<b>Equipe</b>  Analistas : Analisam os resultados e e verificam o funcionamento da API	<b>Clientes</b>  ◦ Empresas ◦MEIs ◦Pessoas que queiram comentar em projetos
	<b>Saídas</b> A saída de dados consiste em uma análise de sentimentos que retorna valores numerais entre 0 e 1 e a emoção com maior valor identificado. Podendo ser positivo, neutro ou negativo.		<b>Stakeholders Chaves</b> <ul style="list-style-type: none"><li>• Donos de Projetos</li><li>• Pessoas que comentam no site</li></ul>	
<b>Custos</b>  Para a operação da API inicialmente estamos utilizando o plano gratuito que oferece 5000 chamadas por 30 dias. Estamos com previsão de mudança para o plano Standard para que o site possa continuar a utilizar a API.			<b>Receitas</b>  A receita é gerada a partir de uma taxação nos projetos, que é calculada com base na pauta do projeto e na requisição curricular	