**UNIVERSIDA DE PERNAMBUCO – UPE**

***CAMPUS* GARANHUNS**

**LICENCIATURA EM COMPUTAÇÃO**

**ALISSON DE LIMA ALBUQUERQUE**

**ANDRILEIDE DE SOUZA SERPA**

**ELYZABETT KAROLYNA DE LIMA**

**RIELSON FEITOSA GONÇALVES**

**THIAGO FERREIRA ALMEIDA**

**RELATÓRIO DO PROJETO DA DISCIPLINA PROGRAMAÇÃO III**

**GARANHUNS**

**2019**

Esse projeto foi muito mais trabalhoso que o último. Bem, as diferenças básicas do projeto anterior para este foi a ferramenta de criação de sites, que era mais intuitiva, mais fácil de ser usada e a tecnologia em si que a gente usou no lado do servidor (do *xampp*) era outra linguagem que a gente não tinha explorado muito que foi o *php*. Neste projeto foi adicionada as tecnologias do Java (*servlet* e *jsp*) e o uso do *ide (eclipse)* com o servidor *TomCat*. Em comum, usamos o HTML5 e o CSS3, mas não igual as formas de escrever os códigos, mas continua servindo para o mesmo propósito. O CSS também foi usado para estilizar a página, criamos um arquivo.css que foi linkado no HTML.

No início tivemos dificuldade em usar várias tecnologias de uma vez e foi mais difícil entender as classes e a montar as páginas, também foi complicado mexer no *Servlet* mas depois conseguimos guardar os dados enviados do *jsp* para uma classe *servlet* e por fim usar os dados para gerar outra página com o retorno em contra partida a requisição, enviada pelo HTML5.

Assim tivemos que primeiro fazer um arquivo.jsp para guardar o HTML dentro, e colocamos uma *teg Form* para criar uma página de opções com dois botões (cadastro de artigo e listar todos), sendo assim a página redireciona para a criação do formulário e/ou para listar todos os formulários cadastrados, aí ela retorna como resultado *method=”POST”,* que envia mensagens mais seguras para o servidor. Dentro do *Form* colocamos que era para enviar as informações do artigo (Nome do artigo, nome dos autores, onde foi publicado, ano, localização das páginas, etc). Cada um desses possui um bloco de entrada de dados que é onde o usuário passa os dados, também tem o botão de enviar que pega as informações e manda para o servidor e na página listar todos ela retorna uma lisca com todos os artigos cadastrados na “platafoma”.

Portanto, vamos para a parte de *back-end,* onde essas informações vão chegar. Estendemos a classe *HttpServlet* do pacote *javax*, criando uma outra classe para que passássemos os dados criados pelos usuários, assim armazenamos separadamente e criamos uma *String* para cada informação. Jogamos eles dentro de um objeto, onde ele guarda todas as informações do envio, e fizemos isso usando um *ArrayList* de cadastros. Por fim, serializamos esses objetos. Sendo assim, pegamos o *ArrayList* e usamos a classe *Random,* para sortear um elemento da lista para fazer retorno ao envio que foi feito pelo cliente, assim o arquivo que foi sorteado da lista volta para a máquina do cliente.