

## **TRABALHO DE ENGENHARIA DE SOFTWARE I – 4. BIM**

- Realizar o trabalho em grupo de até 3 alunos;
- Trabalho entregue com atraso: -2 pontos por dia de atraso;
  - a. **CONTA COMO ATRASO QUALQUER TRABALHO ENTREGUE APÓS 10 minutos do início da aula.**
- Eventuais cópias de conteúdo ou trabalho idêntico a de outros grupos, nota zero para todos os envolvidos;
- Em prova será cobrado os conhecimentos adquiridos neste trabalho.
- É obrigatório realizar todos os trabalhos. A falta de um deles, significa trabalho não realizado – nota zero.

Em resumo o trabalho abrange:

- Pesquisa de um tema – Peso 1,0
- Implementação Aplicação Convencional – Peso 2,0
- Implementação utilizando Webservices e SessionBean (RMI) – Peso 2,0
- Utilização de IA e Microservices – Peso 3,0

### **1. Realizar Pesquisa – Peso 1,0 – 13/11/2018**

Para os temas abaixo prepare apresentação com SLIDES para:

#### **A. DEVOPs**

#### **B. Escolha um dos temas abaixo:**

- WebAssembly
- CHATBOT
- Microservices
- ESB (Enterprise Service Bus)

Preparar para os temas (“A” e “B”) slides para apresentação dos temas pesquisados – Preparar slides para até 7 minutos de apresentação.

A nota desta parte será atribuída apenas a quem do grupo apresentar o trabalho – Nota Individual. Então, recomenda-se que todos apresentem algo sobre os temas propostos.

## 2. Estudo de Caso: Controle de Contratos

### Parte 1 – Implementação Sistema básico – Peso 2,0

Nro. Contrato: _____	Data Emissão do Contrato: ____/____/____
Período de Validade: ____/____/____ a ____/____/____	
Tipo de Contrato: <i>(Prestação serviço, Serviços Jurídicos, contratação Mão Obra, etc)</i>	
Cliente: <i>(nome / endereço / cpf ou cnpj / e-mails / fones)</i>	
Valor do Contrato R\$ _____	
Descrição do Contrato: <i>(texto livre)</i>	
_____	
_____	
_____	

Dado a tela acima, modele um sistema para implementá-lo, nos seguintes termos:

**a) Artefatos da fase de INICIAÇÃO**

- Documento de Visão
- Lista de Requisitos (Utilizar a Planilha Template)
- Diagrama de Use cases

**b) Artefatos para Iteração de ELABORAÇÃO**

- Diagrama de classes de negócios (utilizar modelagem padrão pessoa / pessoa física / jurídica)
- Diagrama de Pacotes (Nível classes)
- Modelo E-R
- Diagrama de sequência para o serviço cadastrarContrato
- Utilizando ferramenta Eclipse, crie um projeto para as classes de negócio: SiglaSistema + sufixo BO (Business Object). Neste projeto deve estar implementada todas as classes modeladas no diagrama de classes de negócio. Incluir implements serializable em cada classe.

### c) Implementação da Camada de Negócio / Serviços

Implementar os serviços:

- **cadastrarContrato**
- **consultarContratoPorNumero**

Requisitos serviço: **cadastrarContrato**

Parametro entrada: Objeto contrato com o cliente e demais dados associados

Parâmetro saída: `NegocioException` se houver erro.

Requisitos:

- Não permitir mais que 3 contratos para o mesmo cliente dentro do período de validade;
- No objeto cliente, a cada contrato cadastrado, somar mais um do atributo `QTDECONTRATO` presente no objeto `CLIENTE`;
- Valor do contrato sempre maior que R\$ 1,00
- Demais atributos do contrato sempre com valor válido

Requisitos serviço: **consultarContratoPorNumero**

Parametro entrada: numero do contrato

Parâmetro saída: `NegocioException` se houver erro ou objeto `CONTRATO`

## **Parte 2 – Introduzindo Processamento Distribuido e Microserviços**

### **d) Implemente os serviços utilizando um Servidor de aplicação padrão J2EE – Peso 1,0**

**Para as camadas de negócio:**

**Crie um projeto específico para acionar por WebServices e SessionBean os serviços acima já implementados.**

- Expor os serviços implementados anteriormente em **WebServices**;
- Expor os serviços implementados anteriormente em SessionBean (EJB 3.x, protocolo RMI)
- Adaptar as camadas de negócio, para **uso de Pool de Conexões com o SGBD**. Para isso, será necessário alterar a forma de conexão com o SGBD e obtenção da conexão, para uma chamada ao Pool de Conexões configurado no Servidor de Aplicação.

### **e) Implementação Interface com o Usuário – Peso 1,0**

- Crie **Interface Web**: Criar numa ferramenta ou tecnologia qualquer de front-end uma interface Web para consumir os WebServices implementados de acordo com a tela de interface já proposta anteriormente.
- **Interface Teste**: Neste tipo de interface, consumir os serviços via RMI.

## f) Utilização de Microserviços e IA – Peso 3,0

- Utilizando os microservices da CLOUD (IBM, Google, Amazon, Microsoft ou outra) para reconhecimento de elementos de uma foto.

- Modifique a interface WEB para:

Nro. Contrato: _____	Data Emissão do Contrato: ____/____/____
Período de Validade: ____/____/____ a ____/____/____	
Tipo de Contrato: <i>(Prestação serviço, Serviços Jurídicos, contratação Mão Obra, etc)</i>	
Cliente: <i>(nome / endereço / cpf ou cnpj / e-mails / fones)</i>	
Valor do Contrato R\$ _____	
<b>Informe Imagem (foto):</b> <i>xxxxx.jpg</i>	
Descrição do Contrato: <i>(texto livre)</i>	
_____	
_____	
_____	

Neste caso, dado uma foto qualquer, utilize o microserviço escolhido para fazer reconhecimento da mesma.

A descrição do contrato será realizada automaticamente da seguinte forma:

“O presente contrato abrange “ + todos os elementos separados por vírgula que o microserviço retornar que estiver indicado com taxa de acerto acima de 60%.

Para implementar isso, crie um projeto e crie um serviço chamado:

**gerarDescricaoContrato**

Parâmetro entrada: foto

Parâmetro Saída: string com a descrição do contrato.

### **Executando este trabalho você perceberá:**

- a) Implementação de processamento distribuído. A sua aplicação não será apenas em N Camadas lógica;
- b) Conhecimento sobre Servidores de Aplicação (container Web / container Negócio);
- c) Uso racional do SGBD, pelo fato de usar Pool de Conexões;
- d) Modelo de Arquitetura Evolutiva (evolucionária) . Uma vez, que mais de 90% do código fonte e praticamente a totalidade da análise e design da aplicação são mantidos;
- e) Observar, que os serviços expostos na camada de negócio, poderão ser acessados de qualquer computador (software cliente) ligado a rede onde o servidor de aplicação estiver conectado.
- f) Utilização de Microservicos (na CLOUD) como parte de uma aplicação
- g) Noções de IA e reconhecimento de imagem – apenas utilizando serviços nas nuvens já pre-treinados. Observe aqui que uma implementação mais robusta exigirá treinar o módulo de IA nas fotos e imagens que o sistema pode aceitar;
- h) Compare as duas diferentes interfaces WEB – a tradicional onde o usuário diz tudo e outra onde há “Inteligência” e a partir de uma foto o sistema sugere uma descrição

### **Critérios de Avaliação do trabalho:**

- a) Compatibilidade da implementação com a Arquitetura de Desenvolvimento
- b) Corretude nos artefatos pedidos
- c) As aplicações implementadas e funcionando corretamente
- d) A totalidade do trabalho pedido – ou seja, a ausência de alguma parte pode implicar em desconto na nota das partes realizadas.

## **Procedimentos para poder utilizar os serviços na cloud:**

No caso da google ou IBM, pesquise por:

Google Cloud Vision

IBM Cloud Visual Recognition

No site, crie uma conta (na IBM, a Unioeste tem convenio. Utilize seu e-mail da Unioeste).

Com a conta gerada, você precisará criar uma chave de acesso (ticket) para consumir serviços.

Na IBM será um código a ser incluso no programa que chama o microservice.

No caso da Google um arquivo .JSON que você deverá atribuir a variável de ambiente do SO utilizado.

No google, baixe o developer kit Google Cloud For Eclipse. Na IBM existe um .JAR que disponibilizei no google DRIVE para este fim junto com exemplos de uso das duas clouds:

[https://drive.google.com/drive/folders/1wGf5Mur\\_vExGS5rwGjnNK78kWzyBEs55?usp=sharing](https://drive.google.com/drive/folders/1wGf5Mur_vExGS5rwGjnNK78kWzyBEs55?usp=sharing)

Cuide no cadastro das contas. Não vincule nenhum meio de pagamento. Ambos as clouds para estudos são grátis – mas com limitações. Mas, para nossas experiências serve.