**Este documento tem como objetivo a definição de uma estratégia para a evolução tecnológica nos conceitos de JEE e SQL dos colaboradores da Logic.**

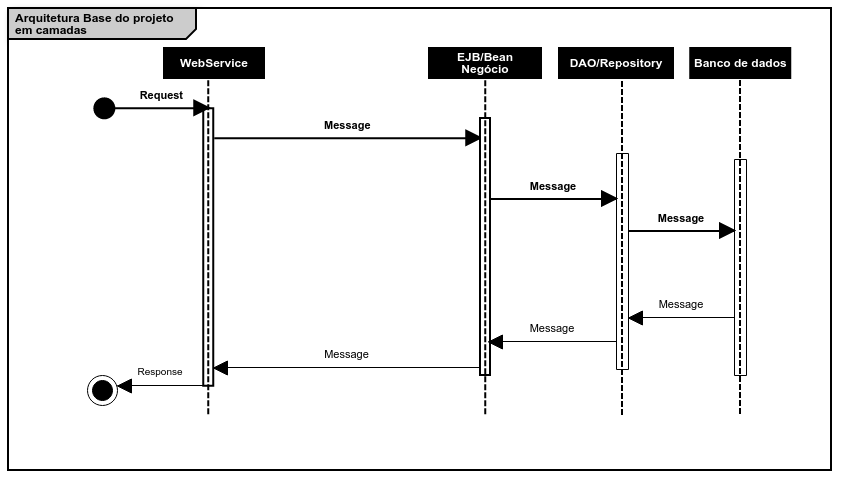
Para definição da estratégia os seguintes pontos são importantes:

* **Metodologia**
* será fornecido material de apoio, junto com um curso a distância que devem ser utilizado como base de conhecimento
* após a conclusão dos estudos do material de apoio e cursos complementares o colaborador deverá executar avaliação prática
* **Material**
* o material a ser utilizado para o treinamento em JEE, será o oficial disponibilizado pelo próprio site da Oracle, o qual se encontra em: <https://docs.oracle.com/javaee/7/JEETT.pdf>
* EAD auxiliar JEE: <https://www.alura.com.br/carreira-desenvolvedor-java-web-enterprise>
* EAD auxiliar JEE: trilha da Red Hat Delivery Specialist - Middleware Application Development MWS-DEL-APD-13849-CUR-EN\_US
* o material a ser utilizado para o treinamento em SQL será o disponibilizado pelo site: <https://www.w3schools.com/sql/default.asp>
* EAD auxiliar SQL: <https://www.alura.com.br/carreira-iniciando-com-sql-e-mysql>
* outras fontes que sejam de interesse do participante podem ser utilizadas

**Avaliações**:

* por não conter framework de interface nesta trilha, todas as funcionalidades devem possuir webservices(REST, Servlet e/ou SOAP) para a realização de chamadas e a validação
* é necessário a criação de um manual detalhado de instalação, configuração e utilização do pacote do projeto
* não é permitido que sejam feitos projetos em grupos de colaboradores
* o banco de dados que deve ser utilizado é HSQLDB
* arquitetura geral do projeto:

o projeto deve seguir a arquitetura especificada abaixo:



considerando o diagrama acima, o projeto, deve ser dividido em no mínimo 3 camadas sendo:

* webService: camada responsável pelo recebimento das chamadas realizadas
* EJB: camada de negócio aonde as validações de entrada, obtenção de objetos e tratativas de negócio devem ser efetuadas
* DAO/Repository: camada responsável pelo acesso ao modelo relacional e dados do produto

**Básica**:

* criar um projeto que possua as seguintes características (utilizar empacotamento ejb + war + ear):
* arquitetura: WebService -> EJB -> JDBC
* utilizar Session Beans Stateless
* considere uma arquitetura utilizando DAO/Repository(utilize CDI para injeção)
* funcionalidades:
* lançamento de contas do mês(nome, data, valor, tipo do lançamento(NORMALIZADO)), considere validações dos campos
* alteração de lançamentos, considere validações dos campos
* exclusão de lançamentos, considere validações dos campos
* pesquisa de lançamentos por período de tempo (dia/mês/ano), considere validações dos campos
* pesquisa de lançamentos por parte do nome do lançamento, considere validações dos campos
* pesquisa por tipo de lançamento, considere validações dos campos
* testes automatizados
* manual de utilização/implantação

**Intermediário**:

* modificar o projeto básico adicionando as seguintes características:
* arquitetura: WebService -> EJB -> Hibernate
* remodelar a aplicação de forma a utilizar o Hibernate para persistência
* funcionalidades adicionais:
* conta corrente, gerenciando o total de gastos ( conta corrente como banco) - utilizar mecanismo de Events CDI para atualizar os valores da conta corrente
* mecanismo de rastreio de todas os lançamentos que alteraram a conta corrente
* este log não precisa ser feito em banco de dados (utilize cache)
* o local de log deve ser único(utilize singleton session bean)
* testes automatizados
* manual de utilização/implantação

**Avançado**:

* modificar o projeto intermediário adicionando/modificando as seguintes funcionalidades:
* arquitetura: WebService -> EJB/JMS -> Hibernate
* remodelar a parte de atualização de valor total da conta corrente para utilização de JMS e MDB(Message driver bean)
* funcionalidades adicionais:
* criar um timer que a cada 10 minutos irá pegar todos os lançamentos no mecanismo de rastreio e imprimir todas as informações do console do servidor(utilize o conceito de schedule)
* utilizando filtros/interceptors criar um mecanismo de segurança para aplicação, o usuário e senha admin/admin123 deve ser válidos
* crie uma pequena interface em HTML+Javascript que utilizando o conceito de websocket, mostre a cada novo rastreio uma notificação no navegador
* testes automatizados
* manual de utilização/implantação

**Entrega, monitoramento/acompanhamento e avaliação do treinamento**

* ao início da parte prática do treinamento o colaborador deve criar um usuário e um novo repositório no [https://www.github.com](https://www.github.com/)
* a cada semana será realizado um acompanhamento do executado VS esperado do treinamento, seguindo o definido cronograma(reunião de 30 minutos)
* ao final do projeto uma apresentação sobre o projeto prática deve ser realizada, considerando:
* arquitetura utilizada
* padrões utilizados
* resumo do funcionamento do projetos
* outros aspectos que o colaborador considerar importantes
* durante o processo de treinamento/avaliação os problemas/dúvidas podem ser direcionados, a qualquer momento, ao responsável elencado para acompanhamento

**Cronograma de entregas/prazos**

* o prazo total para entrega do projeto são de 3 meses, sendo que deste 3 meses as seguintes etapas devem ser cumpridas:
* estudo das tecnologias: primeiro mês
* EAD auxiliar JEE – 10 dias
* EAD JEE RED Hat – 10 dias
* EAD auxilia SQL – 10 dias
* \*Leitura do material – paralelo ao primeiro mês
* desenvolvimento projetos: 2 meses
* projeto básico – 15 dias
* criação do projeto básico – 1 dia
* definição da arquitetura do projeto básico – 3 dias
* funcionalidade 1 básico – 2 dias
* funcionalidade 2 básico – 1 dia
* funcionalidade 3 básico – 1 dia
* funcionalidade 4 básico – 2 dias
* funcionalidade 5 básico – 2 dias
* funcionalidade 6 básico – 2 dias
* manual do projeto básico – 1 dia
* projeto intermediário – 17 dias
* definição da arquitetura intermediário – 4 dias
* ajuste funcionalidade 1 intermediário – 1 dia
* ajuste funcionalidade 2 intermediário – 1 dia
* ajuste funcionalidade 3 intermediário – 1 dia
* ajuste funcionalidade 4 intermediário – 1 dia
* ajuste funcionalidade 5 intermediário – 1 dia
* ajuste funcionalidade 6 intermediário – 1 dia
* conta corrente, gerenciando o total de gastos – 3 dias
* mecanismo de rastreio de todas os lançamentos que alteraram a conta corrente – 3 dias
* manual do projeto intermediário – 1 dia
* projeto avançado – 17 dias
* definição da arquitetura avançada – 4 dias
* remodelar a parte de atualização de valor total da conta corrente para utilização de JMS e MDB – 4 dias
* funcionalidade 1 avançada – 2 dias
* funcionalidade 2 avançada – 2 dias
* funcionalidade 3 avançada – 4 dias
* manual do projeto avançada – 1 dia

**Expectativa de cronograma:**

