

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

DEPARTAMENTO DE ESTATÍSTICA E INFORMÁTICA

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

BANCO DE DADOS (BC3) – 2015.2

**PROJETO**

**AVA**

Alunos: Guilherme Melo

João Nascimento

Paulo Menezes

Recife – 2015

**AMBIENTE VIRTUAL DE APREDIZAGEM (AVA)**

INTRODUÇÃO

O projeto AVA surgiu após a implementação de sistema homônimo na Universidade Federal Rural de Pernambuco (UFRPE). AVA é um acrônimo para Ambiente Virtual de Aprendizagem e é o ambiente da UFRPE usado por docentes e discentes, sendo útil ao primeiro para gerenciamento de conteúdo aos alunos, administração do curso e do acompanhamento do desempenho de estudantes, enquanto que para os estudantes, facilita a comunicação com o professor e demais alunos e da permanência de todo o conteúdo a ser utilizado por ele em sua vida acadêmica num único local. Sendo assim, o AVA funciona de forma similar a Edmodo ou Edulify. Aproveitando o lançamento oficial do AVA no segundo semestre de 2015, foi-se confirmado a escolha pelos estudantes para desenvolvê-lo.

No entanto, o projeto AVA não tem por objetivo reproduzir o AVA, pois aquele possui objetivos distintos desse, pois (o projeto AVA) não é focado única e exclusivamente em funcionalidade do AVA, mas também em funcionalidades do Sig@ (Sistema de Informações e Gestão Acadêmica). Seria até, talvez, mais adequado o uso de Sig@VA ou qualquer aglutinação desses dois siglemas para representar o projeto. Ademais, o projeto AVA tem como objetivos fornecer uma aplicação capaz de modular um ambiente de gerenciamento de informações pelos atores do sistema (professor, aluno e secretária) e permitir a troca dessas informações, construindo um ambiente integrado e uma ferramenta que busca simplificar processos comuns da vida acadêmica.

DETALHES DA ***APLICAÇÃO***

O projeto AVA é desenvolvido sobre o paradigma orientado a objetos em conjunto com o paradigma orientado a aspectos. Esse último, na qual se debruça o projeto para a disciplina de Paradigmas de Programação. O projeto AVA é dividido em camadas, na qual cada uma delas é responsável por agrupar linhas de código com função semelhante no projeto, sendo assim dividido em: dados, negócio e interface gráfica. A camada de dados é gerida pelo Sistema Gerenciador de Banco de Dados (SGBD), que nesse caso, é o MySQL. Sendo ele, o responsável pela parte de persistência dos dados inseridos na aplicação. O motivo do uso SGBD, e não de arquivos ou ainda de outro sistema de gerência, se deve porque, quanto ao primeiro, o projeto AVA está sendo desenvolvido concomitantemente à disciplina de Banco de Dados e, ao segundo, porque esse é o SGDB adotado nessa disciplina.

Na camada de negócios, parte que cuida das regras da aplicação, responsável pelo controle do fluxo do sistema há o uso de aspectos, enquanto que na interface gráfica (GUI), camada responsável por mostrar o conteúdo ao usuário, é usado JavaFX, uma expansão do Java, que permite uma maior flexibilidade e facilidade na criação de GUIs.

Mais especificamente, na parte dos aspectos, o projeto foi desenvolvido considerando os seguintes interesses transversais: autenticação, log de dados, exceções e consultas ao banco de dados.

A autenticação no projeto é usada para validar a entrada de um usuário no sistema, podendo ser o administrador, um aluno ou algum professor. Logo, somente usuários que se encaixem nesse grupo e estejam cadastrados é que terão entrada autenticada com sua identificação.

A parte de log de dados é responsável por armazenar por meio de registros, eventos importantes que o usuário faça na aplicação. São considerados importantes, por exemplo, cadastro de usuários, realização de matrícula, inserção de projeto de pesquisa, solicitação de entrada num projeto etc.

Os aspectos são usados em exceções para avisar a aplicação de um fluxo dissemelhante do esperado, servindo como uma camada extra ao tratamento de erros comuns. Já as consultas ao banco de dados, permite uma abstração maior entre a camada de dados (gerida pelo MySQL) e a camada de negócio do sistema, pois funciona como uma interface, permitindo um código mais limpo, enxuto e menos emaranhado.

PARADIGMA E LINGUAGEM DE PROGRAMAÇÃO

No desenvolvimento de software, há abstrações do mundo real para a representação dos objetos reais pelos paradigmas de programação. Com o paradigma orientado a aspectos, também conhecido como POA, não é diferente. O POA é um paradigma de [programação de computadores](https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_de_computadores" \o "Programação de computadores) de alta abstração que permite aos [desenvolvedores](https://pt.wikipedia.org/wiki/Desenvolvedor" \o "Desenvolvedor) de [software](https://pt.wikipedia.org/wiki/Software" \o "Software) separar e organizar o código de acordo com a sua importância para a aplicação (separação de interesses). Essa separação de interesses, reúne todo o código que seja de comum função em classes distintas num mesmo aspecto ou ainda códigos diferentes numa mesma classe em aspectos distintos, contanto que sejam códigos de caráter secundário, ou seja, que apoie o funcionamento da aplicação, como: log de dados, exceções, autenticação etc. Isso se bem estruturamento torna a manutencionalibidade da aplicação mais fácil futuramente, pois todo o código comum se encontra num mesmo lugar (aspecto).

O motivo do desenvolvimento fazendo uso da linguagem Java atrelado com AspectJ (linguagem usada para o código em aspecto) se deve primordialmente a dois fatores, o primeiro, familaridade com a “linguagem-base” (Java) devido a desenvolvimento de projetos anteriores e, em segundo, a indicação de ambos professores das disciplinas que este projeto foi concomitantemente desenvolvido, recomendarem seu uso. Não se desperdiçando tempo no aprendizado de uma outra linguagem de programação orientada a objetos ou ainda outra linguagem de programação orientada a aspectos, visto que o uso de AspectJ é restrito à Java.

FUNCIONALIDADES

As funcionalidades descritas de forma breve são as seguintes:

* *Cadastro de professores e alunos:* Nessa funcionalidade, o administrador cadastra todos os usuários do sistema, que estão divididos em professores e alunos. Para tal, são passadas informações gerais, exigidas de todos os usuários (CPF, nome, e-mail etc.), ou específicas, para o discente, por exemplo, o curso na qual está matriculado, e o professor, o departamento à qual está ligado.
* *Matrícula de alunos:* Nessa função, os alunos podem se matricular em ofertas do semestre vigente das disciplinas nas quais estão interessados. Para isso, é necessário que esteja em dia com as disciplinas pré-requisitos (disciplinas que são obrigatórias para a realização da disciplina que queira "pagar").
* *Ministrar turma:* Como administrador, é possível associar um professor a uma turma (oferta). Com isso, após o vínculo, o professor se torna responsável por ministrar aula à turma como também executar atividades para sua avaliação, dentre outros quesitos burocráticos.
* *Inserção de notas:* Os professores cadastrados no sistema serão capazes de inserir notas dos alunos que estão matriculados em alguma turma na qual ministra suas aulas.
* *Inserção de bolsas:* Professores podem inserir bolsas de pesquisa científicas para atuação de alunos nas mais diversas áreas. Sendo partilhadas informações do tema da pesquisa, a organização, valor da bolsa, número de vagas etc. do projeto de pesquisa.
* *Solicitação de bolsa:* Aos alunos interessados em alguma proposta de um projeto de pesquisa, poderá informar ao(s) professor(es) do projeto que gostaria de se candidatar à(s) vaga(s) da bolsa através do AVA. O AVA apenas notifica ao(s) professor(es) sobre o interesse do discente em concorrer à vaga, não sendo responsável, por meio dele, de qualquer outra atividade, como realização de um exame, ou ainda, garantia ao aluno que será selecionado.
* *Avisos:* Os avisos são notificações que são direcionados aos usuários do sistema, devido à realização de uma atividade. Uma das atividades na qual se gera um aviso é a de "solicitação de bolsa", na qual o professor recebe uma notificação sobre o interesse de aluno em um projeto por ele desenvolvido. Na contramão, um aviso pode ser gerado ao aluno caso ele seja escolhido como bolsista. Há avisos em outras atividades, como por exemplo, na inserção de uma nova disciplina ou confirmação de matrícula numa disciplina.

INSTRUÇÕES PARA RODAR O PROGRAMA

1. Abra o projeto numa IDE de sua escolha que rode Java (recomendamos Eclipse)
2. Crie uma nova conexão em MySQL Workbench 6.3 CE e insira seu usuário e sua senha ou use uma conexão já estabelecida
3. Abra os arquivos SQLs, encontrados dentro do projeto na pasta “ProjetoBanco” na conexão iniciada
4. Rode todos os scripts. Siga a seguinte ordem na hora de compilá-los: GerarTabelas.sql, PopularTabelas.sql, ConsultarTabelas.sql, Views.sql, calcularMédia.sql, calcularMédiaPorAluno.sql, NovaFunçãoPréRequisito.sql e, por fim, Triggers e Procedures.sql
5. Após isso, retorne a IDE escolhida e em “(nomeProjetoNoComputador)/src/com/ufrpe/ava/aspect/ConexaoMySQL.java” encontre a linha “connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/ava", "root", "");” e onde há root substitua pelo nome do usuário posto na conexão e “” sua senha nessa mesma conexão.

CONSIDERAÇÕES FINAIS

Tantos os objetivos como também as funcionalidades na data de entrega do projeto foram cumpridas. Porém, é notável que ainda é um sistema aquém do AVA. Mas ainda, é válido lembrar que essa não foi a pretensão antes ou no discorrer do projeto para Paradigmas de Programação e Banco de Dados, pois é sabido que um sistema desse porte é necessário um maior tempo ou recurso humano. Com uma base sólida, talvez, possa se expandir o projeto e o desenvolvendo em outras disciplinas do curso de Ciência da Computação e o levando ao mesmo patamar da aplicação original, que deu base ao projeto.

Contudo, o projeto AVA conseguiu implementar todos os objetivos na qual foi focado durante seu desenvolvimento. Logo, chegada a data de sua entrega se conseguiu alcançar as expectativas estabelecidas no início do projeto.