

## **SISTEMA GERENCIADOR DE BANCAS DE TCC – GBANCAS<sup>1</sup>**

Vagner GOMES<sup>2</sup>  
Eduardo Henrique SILVA<sup>3</sup>

**Resumo:** Este artigo apresenta o desenvolvimento de um sistema web para gerenciar o controle de todas as informações relacionadas a formação de bancas avaliadoras de TCC (Trabalho de Conclusão de Curso). A banca é composta pelo orientador, por professores avaliadores e/ou por participantes convidados que avaliam, através de pesos, o trabalho do aluno orientando. Com base nessa avaliação, é elaborado um comprovante de participação para os professores e um relatório, chamado de Ata de Defesa, que discrimina as notas do trabalho de acordo com os requisitos determinados. A partir dessas informações, foi criado um sistema capaz de gerenciar todo o processo citado, facilitando o processo de emissão do comprovante de participação dos professores e da ata de defesa dos trabalhos, contribuindo para o melhor desempenho das tarefas, facilitando e agilizando a atividade realizada.

**Palavras-chave:** Spring Security; Banca de TCC; Java Web.

### **1 INTRODUÇÃO**

A utilização de sistemas informatizados é um grande aliado no gerenciamento e organização de dados, principalmente em instituições que precisam manter e controlar uma grande base de dados, resultado de suas próprias operações ou não. Quando se trata de instituições de ensino, isto se torna muito importante também, pois nelas existem um grande leque de atividades a serem exercidas e que conseqüentemente geram muitas informações e dados para serem administrados. Se tratando desse assunto, surge a necessidade de gerar processos ágeis e seguros para garantir a produtividade, eficiência e segurança do tráfego de dados e informações da instituição.

Uma das atividades realizadas no Centro Universitário de Patos de Minas, mais especificamente no curso de Sistemas de Informação, que ainda não é informatizado, é o gerenciamento de todas as fases de formação das bancas avaliadoras de trabalho de conclusão de curso. Elas são compostas, geralmente, por professores da própria instituição que se disponibilizam para participarem e assim avaliar os trabalhos, seguindo os critérios necessários, definindo notas para tais e determinando se serão aprovados ou não.

Atualmente, todo o processo supracitado acima é realizado manualmente pelo coordenador de bancas, que utiliza editores de texto, planilhas ou até mesmo sistemas manuscritos. Esse tipo de modelo não é confiável e não atende as necessidades, pois para a formação da banca é necessário verificar a disponibilidade dos integrantes para então poder criá-la, além de garantir que um mesmo avaliador não esteja participando de duas diferentes ao mesmo tempo.

Diante dessas informações, foi desenvolvido um software que auxilia o coordenador a gerenciar todas as etapas de formação das bancas, tais como, o cadastramento dos alunos, os respectivos trabalhos, orientadores e todos os componentes necessários para formação da banca, facilitando o processo de emissão de comprovante de participação dos professores e da ata de defesa dos trabalhos, contribuindo para o melhor desempenho das tarefas, facilitando e agilizando a atividade realizada.

A principal justificativa para a construção desse projeto era a falta de praticidade em relação a como ele é feito, utilizando de materiais manuscritos ou planilhas, métodos que, se comparado com as tecnologias dispostas no momento, não são confiáveis e eficientes.

## **2 MODEL-VIEW-CONTROLLER**

Com o aumento da complexidade dos sistemas desenvolvidos hoje, é necessário cada vez mais procurar maneiras de dividir grandes problemas em problemas menores para se tornarem fáceis de resolver e de menor complexidade. Dessa forma, qualquer tipo de alteração em alguma parte do projeto não interfere nas demais, facilitando a manutenção, atualização ou até alterações necessárias mais tarde. Para a implementação de grandes aplicações geralmente é utilizado o padrão *Model-View-Controller* (MVC), pois ele separa as camadas, facilitando na divisão das tarefas e consequentemente na eficiência do desenvolvimento do sistema.

Segundo Erich Gamma (2000, p. 20) “A abordagem MVC separa Visão e Modelo pelo estabelecimento de um protocolo do tipo inserção/notificação entre eles. Uma visão deve garantir que a sua aparência reflita o estado do modelo [...]”.

Algumas vantagens na utilização desse padrão:

- Facilita o reaproveitamento de código;

- Facilita na manutenção e adição de recursos;
- Maior integração do time;
- Melhor divisão de tarefas;
- Diversas tecnologias adotaram essa arquitetura;
- Facilidade em manter o código sempre limpo.

A abordagem MVC é composta por três tipos de objetos, segundo Erich Gamma (p. 21), são eles:

- **Modelo:** É o objeto de aplicação. Contém as regras de negócio, lógica e funções. Ela tem acesso a toda e qualquer informação vinda do banco de dados.
- **Visão:** É a apresentação na tela. Responsável por qualquer saída de representação dos dados.
- **Controlador:** É o que define a maneira como a interface do usuário reage às entradas do mesmo. Responsável por fazer a mediação de entradas de dados entre as camadas de visão e de modelo.

### **3 WEB SERVICE**

Muitas empresas evitavam prover funcionalidades na internet por insegurança de expor seus dados, porém, com as vantagens impostas por novas tecnologias e novos modelos de negócios baseados na internet, é quase impossível não ceder. Com advento dos *Web Services*, elas podem disponibilizar serviços de forma simples e totalmente isolados da sua base de dados, utilizando para isso o padrão XML e SOAP (protocolo).

Web Services são utilizados como forma de integração e comunicação de sistemas, de modo que um sistema possa realizar uma chamada para um serviço de outro sistema a fim de obter informações. Estas chamadas podem enviar e receber informações em diversos formatos, sendo que atualmente os mais populares são XML e JSON. (LECHETA, 2015).

Com a utilização desta tecnologia toda e qualquer comunicação entre sistemas passa a ser dinâmica e principalmente segura, pois não há intervenção humana.

#### **4 EXTENSIVE MARKUP LANGUAGE (XML)**

Segundo Bray et al. (*Extensible Markup Language (XML)*, 2004), XML foi desenvolvida pelo *XML Working Group* (originalmente conhecido como SGML Editorial Review Board) formado sobre o patrocínio do *World Wide Web Consortium (W3C)* em 1996. Este grupo foi presidido por Jon Bosak, da Sun Microsystems, com participação ativa do *XML Special Interest Group*, também organizado pelo W3C. Em fevereiro de 1998 tornou-se uma tecnologia recomendada pelo W3C.

O surgimento da XML como um padrão para representação de dados na Web é muito esperado para facilitar a publicação de dados eletrônicos pelo fornecimento de uma sintaxe simples para dados que seja legível tanto por pessoas quanto por máquinas. (Abiteboul; Buneman; Suciú, 2000,[s/p]).

O padrão XML é uma linguagem de marcação extensível, capaz de descrever diversos tipos de dados, se propondo a facilitar o compartilhamento de informação através da internet.

O XML é um formato que não depende das plataformas de hardware ou software. Pela sua portabilidade, um banco de dados pode, através de uma aplicação, escrever em um arquivo XML, e outro banco distinto pode ler então estes mesmo dados.

A filosofia deste padrão se baseia em vários princípios, tais como:

- Separação do conteúdo da formatação;
- Simplicidade em ler e escrever;
- Possibilidade de criação de *tags* sem limitação;
- Criação de arquivos para validação de estrutura;
- Interligação de banco de dados distintos;
- Concentração na estrutura da informação e na sua aparência.

#### **5 SIMPLE OBJECT ACCESS PROTOCOL (SOAP)**

SOAP é um protocolo utilizado para troca de informações estruturadas em uma plataforma descentralizada e distribuída, baseado na linguagem de marcação extensível (XML).

Segundo Mack Hendricks (2002, p.12), “SOAP independe do sistema operacional, do ambiente de programação e do modelo do objeto”.

Este protocolo consiste em três partes:

- Envelope: que define um *framework* para a descrição da composição da mensagem e como ela deve ser processada;
- Header: que pode ser usado efetivamente pelos usuários para fornecer todos os dados adicionais necessários;
- Body: que contém a carga útil da XML, codifica a chamada do procedimento, as respostas da chamada do procedimento, ou o relatório de falhas.

## 6 METODOLOGIA

O desenvolvimento da aplicação web para auxiliar na formação de bancas avaliadoras tem como base a metodologia ágil, Scrum, utilizada para gestão e planejamento de projetos. O desenvolvimento será dividido em ciclos (tipicamente mensais) chamados de *Sprints*, que representam um *Time Box* dentro do qual um conjunto de atividades deve ser executado, fornecendo um processo conveniente para sua realização.

No Scrum, existem papéis definidos para os processos a serem realizados, dentre eles, os principais são:

- Scrum Master: mantém os processos (normalmente no lugar de um gerente de projeto), atuando como facilitador, responsável por remover quaisquer obstáculos que possam dificultar no desenvolvimento do projeto;
- *Product Owner*: o proprietário do produto, que representa os *stakeholders* (Ex.: um grupo de investidores) e o negócio;
- *Team*: é a equipe, um grupo multifuncional que fazem a análise, projeto, implementação e teste.

A utilização desta metodologia é interessante porque fornece um mecanismo de informação de status que é atualizado continuamente, muito ágil. “Agilidade é a habilidade de criar e responder a mudanças com respeito ao resultado financeiro do

projeto em um turbulento ambiente de negócios. Agilidade é a habilidade de balancear a flexibilidade com estabilidade”. (Highsmith, Jim. Agile Project Management, 2002).

As ferramentas a serem utilizadas para o desenvolvimento do projeto são:

- Java: linguagem de programação que será utilizada para a codificação da aplicação.
- NetBeans IDE: ambiente de desenvolvimento para a codificação com a linguagem Java;
- Android Studio: ambiente de desenvolvimento para a codificação da aplicação mobile;
- MySQL WorkBench: sistema gerenciador de banco de dados.

A seguir estão descritas as etapas a serem realizadas para a implementação da aplicação web e mobile.

Primeiramente, junto ao coordenador de banca e com as demais pessoas que irão usufruir do sistema, deve-se fazer um levantamento e dos requisitos sobre o assunto a ser tratado, a fim de procurar suas aplicações e vantagens. A coleta de requisitos é uma fase muito importante, pois nela se entende qual a real necessidade do desenvolvimento do projeto.

Em seguida, a partir dos requisitos colhidos, é necessário fazer sua análise, de forma a atender as observações dos responsáveis das atividades realizadas em relação a formação da banca.

Após, será iniciada a fase de implementação do projeto, utilizando dos ambientes de desenvolvimento *NetBeans IDE* e *Android Studio*, no qual, será feito, respectivamente, a codificação das aplicações web e *mobile*, usando a linguagem Java.

Ao longo de todo o processo, serão realizadas reuniões para disseminar conhecimento sobre o já está feito, identificando impedimentos e priorizando os próximos passos a serem realizados, de acordo com as funcionalidades necessitadas em relação do projeto.

Por último, depois da aplicação finalizada, ele será implantado e, assim, realizado uma bateria de testes para verificar sua consistência, comprovando sua utilidade mediante ao que foi proposto inicialmente. O sistema será disponibilizado apenas para a formação de banca do curso de Sistema de Informação do Centro

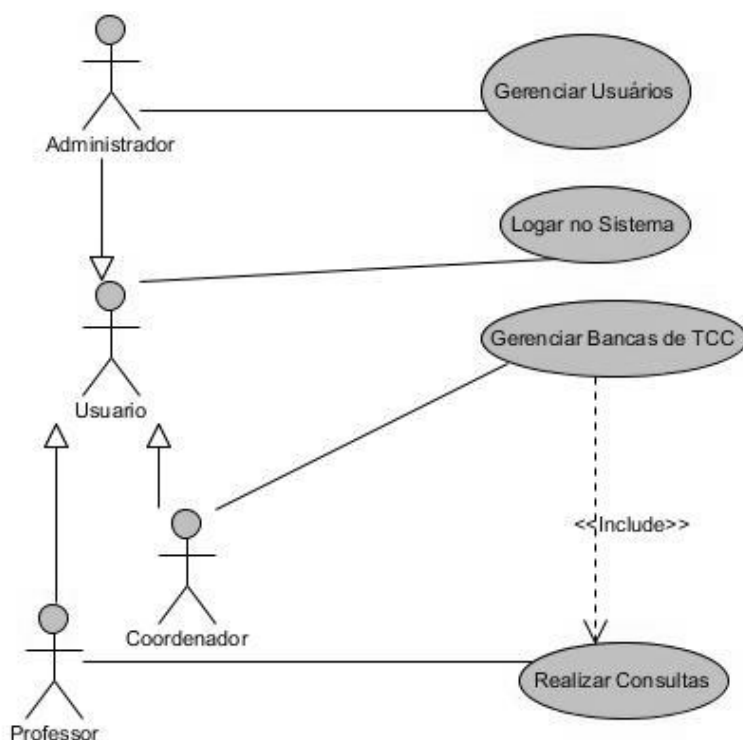
Universitário de Patos de Minas, podendo futuramente ser expandido para outros cursos e/ou até mesmo para outras instituições de ensino.

## 7 DESENVOLVIMENTO

A aplicação é dividida em módulos que são gerenciados por usuários do sistema. Cada usuário tem suas permissões definidas. O usuário Administrador tem responsabilidade geral, mantém os dados necessários para o funcionamento do sistema para outros usuários, ou seja, todo sistema depende das informações adicionadas por ele. O usuário Coordenador é quem faz o gerenciamento da banca de TCC, ele informará os dados para a formação das bancas, tal como aluno orientando, professor orientador, professores que comporão as bancas, horário, data e outros dados necessários.

A Figura 1, abaixo, representa o Diagrama de Caso de Uso que Modela o Contexto do sistema, que contém as funcionalidades essenciais realizadas pelos usuários.

**Figura 1:** Diagrama de Caso de Uso de Modela o Contexto.



**Fonte:** Elaborado pelo autor.

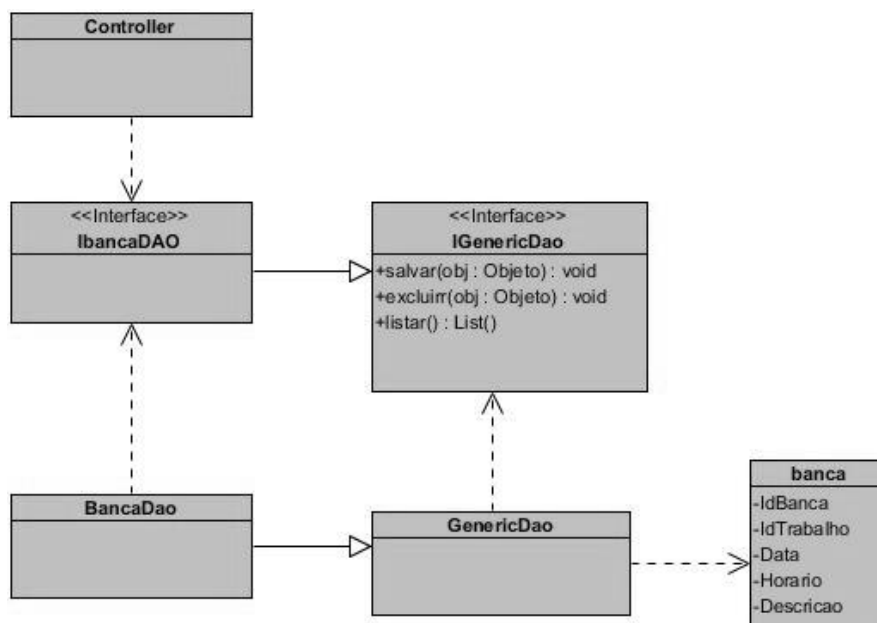
## 7.1 ARQUITETURA

No desenvolvimento foi utilizada a arquitetura MVC (*Model, View, Controller*), que permite fazer a divisão da aplicação em camadas, possibilitando melhor organização do código, a fim de garantir qualidade no sistema. Esse tipo de arquitetura abre espaço para a utilização do padrão DAO (*Data Access Object*), uma estrutura que permite criar classes de dados independentes do tipo da fonte dos dados, ou seja, aumenta a flexibilidade em utilizar diversos SGBDs (Sistema Gerenciador de Banco de Dados). Utilizando esse padrão, as operações de CRUD (Create, Read, Update, Delete) são de inteira responsabilidade do objeto DAO, pois toda lógica de acesso e execução ao banco de dados é colocada dentro dele.

Outra solução utilizada é o framework *Primefaces*, uma biblioteca de componentes de interface gráfica para aplicações web baseada em JSF. Esse framework torna menos árduo criar uma aplicação que seja exibida de forma correta na maioria dos dispositivos, isso o torna muito vantajoso e bem aceito pela comunidade. O JSF é uma tecnologia que permite utilizar componentes visuais pré-prontos, sem necessidade de se preocupar com *JavaScript* e *Html*.

O diagrama de classes, na Figura 2, abaixo, representa o padrão DAO (*Data Access Object*) utilizado na aplicação.

**Figura 2:** diagrama de classes padrão DAO.



**Fonte:** Elaborado pelo autor.



## 7.2. SCRUM E KANBAN

Durante o desenvolvimento da aplicação, não foi possível utilizar a metodologia Scrum, pois as necessidades do desenvolvimento e ambientes externos não se comportaram da forma esperada, como foi proposto no planejamento do projeto.

Como modo de gerenciamento de produção, foi utilizado o quadro *Kanban*. Ele é uma forma de ajudar a assimilar e controlar o progresso das tarefas em um projeto, de forma visual. Normalmente é utilizado um quadro branco com pequenos papéis (*Post-it*) colados. Os papéis representam as tarefas, que são puxados por etapas de progresso até que ela seja finalizada. Para demonstrar esse quadro, foi usado uma função do Portal Acadêmico da faculdade, chamada de Grupo de Trabalho, na qual permite o acompanhamento online do desenvolvimento das tarefas.

A Figura 03, abaixo, mostra o quadro *Kanban* do projeto até o momento.

**Figura 3:** desmonstração quadro *Kanban*.

A Fazer	Em Desenvolvimento	Feito	
Mapear classe PeríodoTrabalho Dt Prevista: 01/09/2016	Schedule Editable (VAGNER) Dt.Início: 18/08/2016	Base de dados (VAGNER) Dt.Início: 18/08/2016 Dt.Fim: 18/08/2016	Mapeamento das entidades de heranças (VAGNER) Dt.Início: 18/08/2016 Dt.Fim: 18/08/2016
		Implementar um template (VAGNER) Dt.Início: 18/08/2016 Dt.Fim: 18/08/2016	Método de Sessão pro Id da pessoa pra cadastro de Usuário (VAGNER) Dt.Início: 18/08/2016 Dt.Fim: 18/08/2016
		Tela Cadastro Usuarios (VAGNER) Dt.Início: 18/08/2016 Dt.Fim: 18/08/2016	Classe SessãoHelper (VAGNER) Dt.Início: 18/08/2016 Dt.Fim: 18/08/2016
		Resolver Links do template (VAGNER) Dt.Início: 18/08/2016 Dt.Fim: 18/08/2016	Método Vincular Usuario e Curso (VAGNER) Dt.Início: 18/08/2016 Dt.Fim: 18/08/2016
		Implementar classe de Sessão (VAGNER) Dt.Início: 18/08/2016 Dt.Fim: 18/08/2016	Mapeamento Professor Trabalho (VAGNER) Dt.Início: 18/08/2016 Dt.Fim: 18/08/2016
		Implementar classe Banca (VAGNER) Dt.Início: 18/08/2016 Dt.Fim: 18/08/2016	Tela listar Cursos de Usuario (VAGNER) Dt.Início: 18/08/2016 Dt.Fim: 18/08/2016
		Cadastro de Horário da Banca (VAGNER) Dt.Início: 18/08/2016 Dt.Fim: 18/08/2016	

**Fonte:** Grupo de Trabalho, Portal Acadêmico, UNIPAM.

## 8. RESULTADO

O sistema tem como objetivo principal atender as necessidades do coordenador de banca, que tem como função gerenciar os dados para a formação dela. Para facilitar o trabalho, foi utilizado um calendário como forma de organização visual, que mostra as bancas em seus respectivos dias e horários.

A forma utilizada para demonstrar o calendário foi um componente da biblioteca do *Primefaces*, a *Schedule*. Ela tem a funcionalidade de organizar seus dados em ordem de Data como ano, mês e dia.

A Figura 4, abaixo, mostra o componente *Schedule* sendo utilizado para organizar as bancas em seus respectivos dias, mostrando o horário e o nome do trabalho.

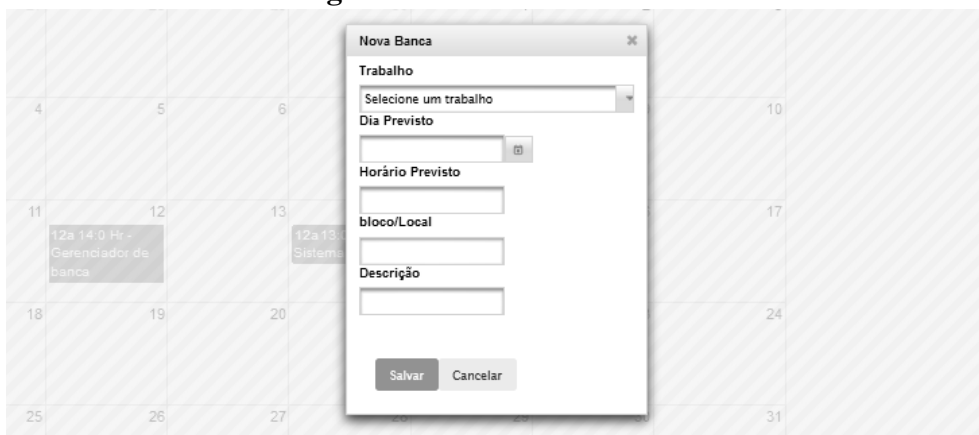
**Figura 4:** demonstração do componente *Schedule* na aplicação.

Dezembro 2016						
Dom	Seg	Ter	Qua	Qui	Sex	Sáb
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12 12a 14:0 Hr - Gerenciador de banca	13	14 12a 13:0 Hr - Sistema teste	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

**Fonte:** Autor do trabalho.

Através do calendário mostrado na Figura 02, é possível fazer o cadastro de novas bancas, como mostra na Figura 5, tendo disponível a escolha de trabalho de que será apresentado na defesa, o dia, horário e descrições caso necessário.

**Figura 5:** tela de cadastro de bancas.



**Fonte:** Autor do projeto.

Outra função do calendário, é gerar relatórios com informações referentes aos participantes do evento. O relatório denominado como Ata de Defesa de Estágio Supervisionado discrimina a avaliação do aluno. Nele contém as notas da apresentação do trabalho, obtidas através da avaliação dos professores que compõem a banca. Esse relatório é muito importante para o coordenador de bancas, pois ele comprova a aprovação ou não do aluno em questão. Outro relatório gerado é o Comprovante de Participação dos componentes da banca. Ele tem a função de discriminar todas as bancas que os participantes (Avaliadores) avaliaram trabalhos, confirmando o comparecimento dele no dia e horário previsto do evento.

## 9 CONCLUSÃO

Foi desenvolvido uma aplicação web para auxiliar no gerenciamento e formação de bancas avaliadoras de trabalho de conclusão de curso, utilizando de todas as tecnologias disponíveis para isso.

Com a implantação e utilização da aplicação, pretende-se que o mesmo contribua para o melhor desempenho das atividades realizadas, impactando de forma

positiva a rotina de trabalho e levando praticidade aos seus usuários e demais dependentes do sistema.

Pretende-se ainda, com o avanço deste projeto, abrangê-lo para outras instituições educacionais que utilizem da metodologia de gerenciamento de bancas baseado no modelo desenvolvido ou se necessário executar adaptações para outras metodologias.

## REFERÊNCIAS

- ABITEBOUL, Serge; BUNEMAN, Peter; SUCIU, Dan. **Gerenciando dados na Web**. Rio de Janeiro: Campus, 2000.
- BRAY, T; PAOLI, J; SPERBERG-MCQUEEN, C. M; YERGEAU, F. **Extensible Markup Language (XML) 1.0**. <http://www.w3.org/TR/2004/REC-xml-20040204/>. Acesso em 27/02/2016.
- GAMMA, Erich. **Padrões de Projeto**: Soluções reutilizáveis de software orientado a objetos. São Paulo: Bookman, 2000.
- HENDRICKS, Mack; GALBRAITH, Ben; IRANI, Romin. **Profissional Java Web Service**. Rio de Janeiro: Alta Books, 2002.
- LECHETA, Ricardo R. **Web Services RESTful**: Aprenda a criar web services RESTful em Java na nuvem do Google. 2015.

