

## **BiblioTCC - Biblioteca e Gerenciador de Bancas de TCC**

**Leonardo Sorgetz Monteiro<sup>1</sup>, Cassio Huggentobler da Costa<sup>2</sup>**

<sup>1</sup>Acadêmico do Curso de Curso de Sistemas de Informação/Análise e Desenvolvimento de Sistemas – Universidade Luterana do Brasil (ULBRA) Campus Torres - RS

<sup>2</sup>Professor(a) dos Cursos de Computação – Universidade Luterana do Brasil (ULBRA) Campus Torres – RS

leosorgetz123@gmail.com, cassio.huggentobler@ulbra.inf.br

**Abstract.** *In this paper will be presented the software BiblioTCC, to assist the teacher responsible for the organization of the examination board in the management of the stands of the TCC and also serve as an open library of final papers and works, so that the students can study the projects posted. The system was developed in PHP language, version 7.1, using Symfony Framework 3.4.4 as the basis of the project. The methodology used was Scrum, an agile methodology for planning and managing highly utilized projects.*

**Resumo.** *Neste artigo será apresentado o BiblioTCC, software, para auxiliar o professor responsável pela organização de bancas no gerenciamento das bancas de TCC e também servir como uma biblioteca aberta de trabalhos de conclusão, para que alunos possam estudar os projetos postados. O sistema foi desenvolvido com a linguagem PHP, na versão 7.1, utilizando o Framework Symfony 3.4.4 como base do projeto. A metodologia utilizada foi o Scrum, uma metodologia ágil para gestão e planejamento de projeto altamente utilizada.*

### **1. Introdução**

Quando um aluno chega ao período final de um curso superior, vem o grande momento do TCC, onde algum estudo deve ser apresentado para uma banca de professores avaliadores e que varia entre os vários tipos de curso. No de curso de Análise e Desenvolvimento de Sistema da Ulbra Torres, o aluno necessita apresentar o PDS (Projeto de Desenvolvimento de *Software*), onde é desenvolvido um *software* para o aluno comprovar o aprendizado adquirido no curso. A montagem da banca é um processo demorado e cansativo, sem contar a forma de armazenamento das informações desses projetos, que por não ter uma ferramenta apropriada para o armazenamento, acaba dificultando a consulta desses projetos por parte dos professores e alunos de outros semestres. O BiblioTCC vem para auxiliar no armazenamento das informações da banca e projetos passados, fazendo com o que os mesmos fiquem de fácil acesso a outros alunos novos além de auxiliar a construção e manutenção das bancas de defesa desses projetos.

O artigo segue dividido em partes: na seção 2 é apresentado um referencial teórico explicando detalhadamente o processo das bancas e como o BiblioTCC irá funcionar. Na seção 3 será falado sobre a metodologia utilizada e na 4 sobre as ferramentas utilizadas no *software*. Na seção 5 será descrito sobre o desenvolvimento do projeto mostrando o processo de construção e logo em seguida as dificuldades encontradas ao longo do projeto e a sua conclusão

## **2. Referencial Teórico**

Nesta seção será explicado como que funciona o modelo atual de montagem de banca, desde o processo de definição da data de postagem até a aprovação final do orientador do projeto. Também mostrará como que o *software* irá trabalhar para automatizar estas tarefas e facilitar o contato com o aluno sobre prazos e informações relacionadas a banca. Antes do início do desenvolvimento do projeto, foram feitas pesquisas sobre possíveis *softwares* similares e não foi encontrado um sistema que seja aberto ao público.

### **2.1 Modelo Atual**

Durante o final de cada semestre de um curso de TI, muitos alunos precisam entregar seus trabalhos de conclusão de curso ou projeto de desenvolvimento de *software*. Para cada trabalho, deve-se criar antecipadamente uma banca avaliadora que, geralmente, é formada por três professores, o orientador, que já acompanha o desenvolvimento do *software* juntamente ao aluno, e mais dois professores que avaliarão o trabalho que será apresentado e que conheçam o assunto no qual o projeto em questão irá abordar.

Para organizar as bancas do final de semestre, o professor encarregado da tarefa precisa ir atrás de informações dos trabalhos de cada aluno para que possa montar a banca baseado na disponibilidade e no conhecimento de cada professor nas tecnologias utilizadas pelos alunos. Logo após reunir as informações dos professores, o professor encarregado cria uma escala com as bancas, baseado nas informações adquiridas anteriormente. Após o processo de montagem, o professor necessita notificar cada professor avaliador sobre qual banca irá avaliar, passando várias informações como quem é o aluno, o tema que será abordado, a data, hora e o local da defesa do trabalho.

Após a definição das bancas, como descrito acima, o aluno deve então enviar o artigo e o código fonte do projeto para o orientador até a data final da primeira entrega, definido pelo professor encarregado por definir as bancas, e aguardar a data da banca para apresentar o mesmo. Após a apresentação, o aluno recebe orientação do que deve ser modificado no artigo e ajustado no *software* para a entrega final que, geralmente, é marcada para uma semana após a apresentação da banca. Após a entrega final, o orientador avalia se o artigo e o código fonte estão de acordo com os ajustes e dá a banca como finalizada.

Outra situação é o acesso dos alunos aos artigos e código fontes postados para fins de estudo. Durante o período de curso, muito alunos demonstram interesse em ver projetos já apresentados em semestres anteriores, porém, estes projetos devem ser selecionados pelo professor encarregado ou por critério de nota. No modelo atual, a pesquisa é feita através do *Classroom*, onde o aluno entra em uma matéria que tem acesso à uma pasta do *Google Drive* com artigos e código fonte divididos por semestre.

### 2.3. Solução proposta

O BiblioTCC vem com a proposta de auxiliar o coordenador do TCC/PDS a tarefa de organização de banca facilitando a comunicação com professores e auxiliando o acesso aos trabalhos postados de uma forma mais customizada. No BiblioTCC, todos os projetos que obtiverem grau acima de nove e uma confirmação do professor orientador estarão disponíveis para acesso dos alunos de forma rápida e fácil, com informações sobre o projeto. Juntamente ao projeto, poderá também ser visto informações da banca, como grau final, professores avaliadores, professor orientador e outros.

Quanto a comunicação com professores, o coordenador do TCC/PDS para criar as bancas terá uma facilidade pois ao cadastrar o professor em uma banca e o mesmo conectar no sistema, será mostrado na *dashboard* quais bancas ele tem para participar e o mesmo pode ver informações gerais com antecedência.

## 3. Metodologia

Nesta seção será apresentada a metodologia *Scrum*, uma metodologia ágil muito utilizada nos dias atuais. Também será explicado como que ela foi utilizada no projeto uma vez que o *Scrum* deve ser utilizado em equipes e este projeto foi desenvolvido por apenas um indivíduo.

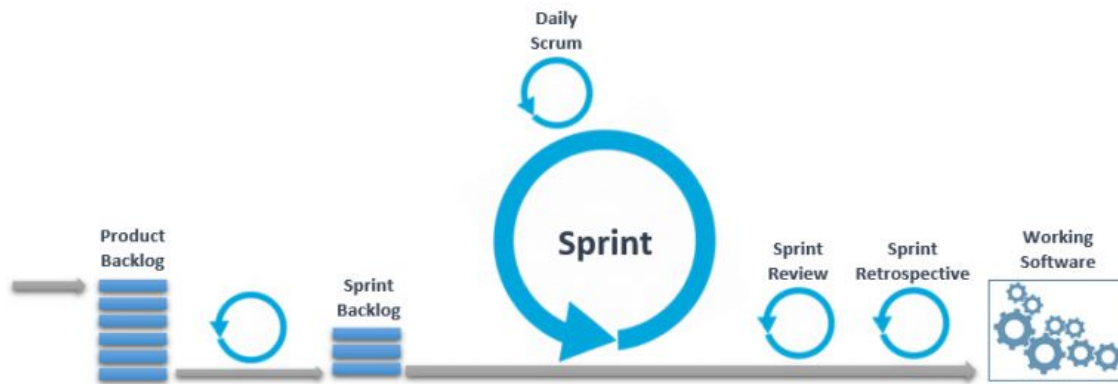
### 3.1 Scrum

*Scrum* é uma metodologia ágil para planejamento de projetos de *software*. O *Scrum* propõe entrega contínua durante o projeto com melhorias e novas funcionalidades a cada entrega, projetos de alta qualidade pois em qualquer etapa os *bugs* encontrados podem ser corrigidos.

As *sprints* são ciclos iterativos que duram de 15 a 30 dias. Cada ciclo funciona como um pequeno projeto dentro do projeto principal, com o objetivo de gerar uma parte executável e estável que ao final juntam-se ao projeto principal.

A metodologia permite que cada membro envolvido no projeto assuma determinada responsabilidade, dando-lhe capacidade de melhor controle sobre suas funções. Esses papéis são:

- **Product Owner:** Responsável por definir o que define o que terá no *product backlog*. Geralmente é a parte contratante do serviço.
- **Scrum Master:** Responsável por guiar o *Scrum Team*, estabelecer vínculo entre o *Product Owner* e a equipe e comprometer que todos sigam as diretrizes do *Scrum*.
- **Scrum Team:** Equipe de desenvolvedores, designers, testadores, comprometidos a desenvolver o produto baseado nas tarefas definidas com o *Product Owner*.



**Figura 1. Ciclo de vida do Scrum**

Explicação de cada etapa do ciclo do *Scrum* da figura 1.

- **Product Backlog:** são todos os requisitos que o cliente gostaria de realizar no projeto. Essa lista é sempre atualizada, de acordo com a importância dada pelo cliente.
- **Sprint Backlog:** tarefas que serão desenvolvidas baseadas no Product Backlog para o incremento do produto na Sprint.
- **Daily Scrum:** reunião feita diariamente que proporciona auto-organização do time de desenvolvimento e tem objetivo de proporcionar à equipe uma visibilidade do trabalho realizado e que vai ser realizado futuramente.
- **Sprint Review:** reunião feita em conjunto com o time de desenvolvimento e o cliente para obter um feedback sobre o incremento do produto produzido.
- **Daily Scrum:** reunião diária feita em conjunto com o time de desenvolvimento para verificar o *status* do andamento da *sprint*. O objetivo da reunião é ser rápida, entre 15 minutos e responder às seguintes perguntas:
  - O que você fez ontem?
  - O que você fará hoje?
  - Há algum impedimento no seu caminho?

### 3.1 Adaptação do Scrum

Como comentado no tópico anterior, o *Scrum* baseia-se no uso juntamente de times para o seu funcionamento. Este projeto utilizou de uma adaptação baseada no *framework*, não utilizando de algumas de suas atividades. Não foi utilizado uma equipe, mas sim apenas um indivíduo. Também não ocorreram as reuniões diárias, ou *Daily Scrums* e foram utilizados o *Product Backlog* e *Sprint Backlog* com acompanhamento das atividades pelo usuário, coordenador das bancas de TCC e nomeado como o *Product Owner*.

## 4. Pré-Planejamento

No momento posterior a primeira entrevista com o *Product Owner*, foram definidos as ferramentas e tecnologias a serem utilizadas para o desenvolvimento do *software* BiblioTCC.

## 4.1. Ferramentas e Tecnologias utilizadas

Nesta seção são descritas as tecnologias e conceitos definidos para o desenvolvimento do BiblioTCC.

### 4.1.1. PHP

Segundo o PHP Group, o PHP é uma linguagem de *script*, muito utilizada para desenvolvimento de aplicações *web*. Nas páginas PHP pode conter CSS, *JavaScript* que são linguagens que interagem com o PHP para gerar um resultado esperado. Segundo AGRELA (2018), o PHP é a 9ª linguagem mais utilizada no mundo.

### 4.1.2. JavaScript

Segundo SILVA(2015), o *JavaScript* é uma linguagem de programação que permite implementar funcionalidades mais complexas em páginas *web*. Auxilia em eventos onde o cliente precisa verificar situações em tempo real.

### 4.1.3. Symfony

Segundo SOUZA(2018), o Symfony é um *framework full stack* criado com um conjunto de componentes do PHP, muito famoso e respeitado na comunidade. O *framework* também é base de outros grandes projetos como Drupal e Laravel.

### 4.1.4. MVC

Segundo LAMIM(2009), *Model-View-Controller* (MVC) é um padrão de arquitetura de *software* que separa em camadas os dados e a lógica de negócio entre o *Model* e *View* introduzindo um componente, que comunica as 2 camadas, chamado *Controller*, dessa forma as alterações feitas não afetam a manipulação de dados, assim podendo ser reorganizados sem mudar o *view*. Exemplo do MVC na figura 2.

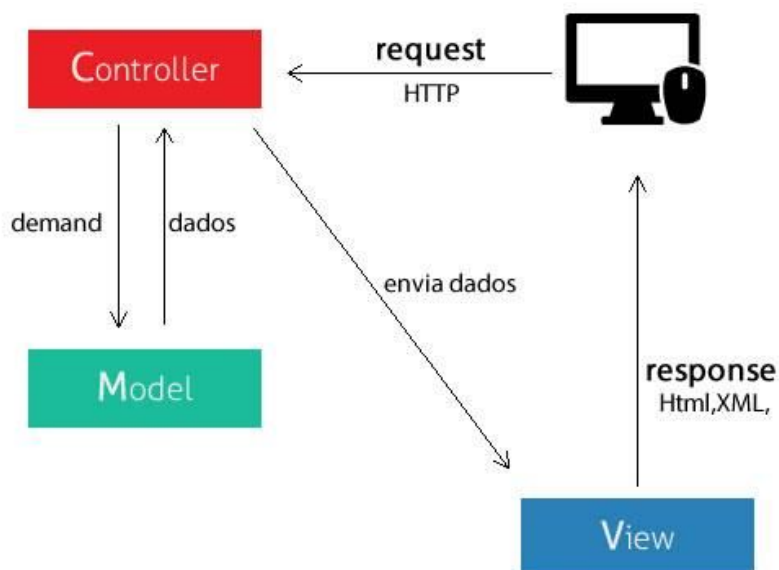


Figura 2. Exemplo de funcionamento do MVC

#### 4.1.5. Composer

Segundo CARDOSO(2014), o Composer é uma ferramenta para gerenciamento de dependências em PHP. Ele permite que você declare as bibliotecas das quais seu projeto depende e as gerenciara (instalará / atualizará) para você. O Composer é uma ferramenta muito importante para o projeto pois é ela quem controla a instalação de auxilia no *deploy* da aplicação para um servidor.

#### 4.1.6. Bootstrap

Segundo LEONE(2018), o *Bootstrap* é um *kit* de ferramentas de código aberto para desenvolvimento com HTML, CSS e JS. O *framework* conta com classes pré prontas para reutilização de maneira rápida com pouco código. De fácil utilização, o Bootstrap auxilia o desenvolvedor em tarefas de *front-end*, assim, agilizando o desenvolvimento de telas do *software*.

#### 4.1.7. MySQL

Segundo PIZA(2012), o MySQL é um sistema gerenciador de banco de dados, que tem por objetivo fazer o armazenamento dos dados, é usado na maioria das aplicações gratuitas para gerir suas bases de dados. O serviço utiliza a linguagem SQL (*Structure Query Language* – Linguagem de Consulta Estruturada), que é a linguagem mais popular para inserir, acessar e gerenciar o conteúdo armazenado num banco de dados.

#### 4.1.8. jQuery

*jQuery* é uma biblioteca *JavaScript* desenvolvida para simplificar os *scripts client-side* que interagem com o HTML. Devida a sua praticidade como seu mantra descreve “*The Write Less, Do More*”, traduzido como “Escreva menos, faça mais”. A sintaxe do *jQuery* foi desenvolvida para tornar mais simples a navegação do documento HTML, a seleção de elementos DOM, criar animações, manipular eventos realizar requisições AJAX. O papel desempenhado pelo *jQuery* é simplificar o desenvolvimento de aplicações *web* dinâmicas de grande complexidade. (BELEM, 2010)

#### 4.1.9. Doctrine

O *Doctrine* é um ORM (*Object Relational Mapping*), que cria uma camada de mapeamento entre os objetos de um *software* e o banco de dados, abstraindo o acesso ao banco, facilitando uma possível migração para outro tipo de banco de dados, por exemplo saindo do MySQL para PostgreSQL. Ele é um grande aliado do desenvolvimento por conta da sua facilidade de uso e também de ser muito poderoso em suas atividades. BRUSCH (2013)

#### 4.1.10. Twig

Segundo BROCANELLI (2016), o *Twig* é um motor de renderização de *front-end* rápido, seguro e flexível, que torna a leitura de código mais limpa, impedindo a abertura de tags PHP em meio ao código HTML. Com uma sintaxe mais direta, facilita o entendimento do desenvolvedor quando o mesmo dar manutenibilidade ao código.

#### 4.1.11. UML

Segundo RIBEIRO(2012), o UML, a Linguagem de Modelagem Unificada, permite exemplificar através de diagramas algumas atividades no sistema, como o diagrama de atividade, que permite explicar um processo de uma determinada ação como por exemplo um processo de compra em uma loja, ou o diagrama de caso de uso, onde explica as ações de um indivíduo em um sistema. Ao total são nove tipos de diagramas que são usados para documentar e modelar um *software*.

Os diagramas utilizados na documentação são:

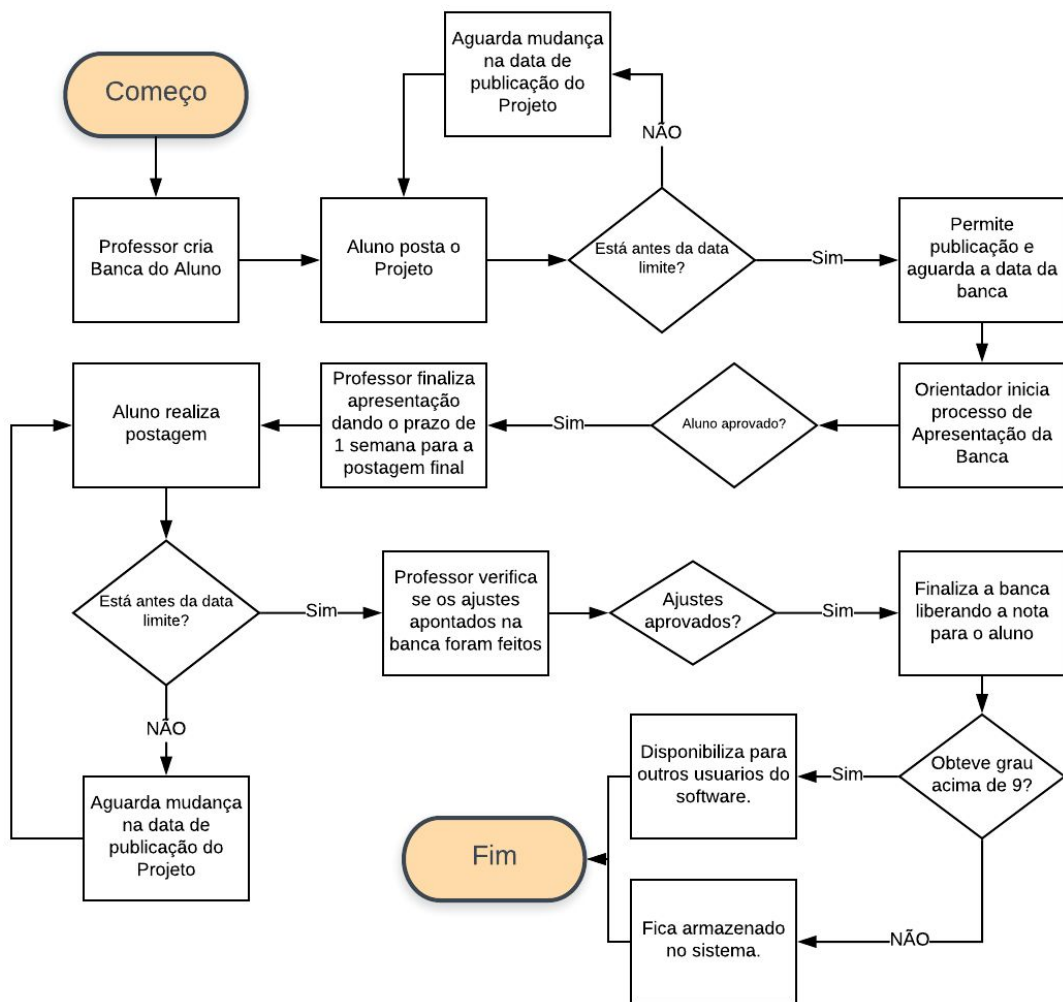
- **Diagrama de Atividade:** diagrama que mostra o fluxo de atividades do sistema, desde o início até o final possibilitando a visão em caso de sucesso ou erro.
- **Diagrama de Caso de Uso:** descreve as funcionalidades do sistema através da interação entre usuário e atividade.

Para criar os diagramas que estão neste artigo foram utilizadas as ferramentas *Lucidchart*, no diagrama de atividades e o *Astah Community* para criar o diagrama de caso de uso.

### 5. Aplicação da Metodologia

Anterior ao início do desenvolvimento do projeto, foi feita uma entrevista com o *Product Owner* para verificar como que está sendo feito no modelo atual e discutir como que isso poderia ser passado para um *software*. Nesta etapa também foram criados o diagrama de atividades e o diagrama de casos de uso do *software*. O diagrama que está na figura 3 demonstra o processo de funcionamento de uma banca no software, do início ao fim, exemplificando seu processo geral.

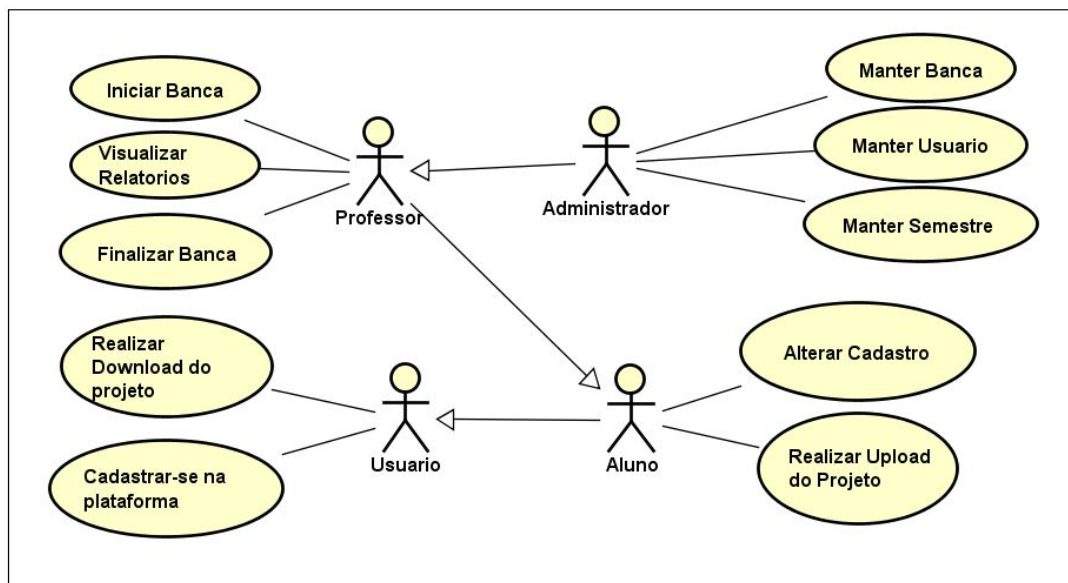
Após a análise de requisitos foi montado o *Product Backlog* com as funcionalidades que o *Product Owner* estava interessado. Baseado nesta lista, foi criado o *Sprint Backlog* com as funcionalidades mais importantes em primeiro momento, como mostra a *sprint 1*.



**Figura 3. Diagrama de Atividade do Gerenciamento de Bancas**

O diagrama de Casos de uso na figura 4 demonstra os papéis atuantes no sistema e quais funções que podem realizar. Um usuário comum, sem estar logado na plataforma, poderá realizar a consulta e o *download* de projeto. O aluno de uma banca, poderá realizar a postagens em sua banca e alterar seus dados. Os Professores poderão gerenciar bancas criadas, porém, não poderão criar bancas e o administrador é responsável por atribuir papéis a outros usuários além da criação do ambiente para que ele outros professores utilizem o *software*.





**Figura 4. Diagrama de Caso de Uso**

## 5.1. Product Backlog

Na tabela abaixo temos o *Product Backlog* do *software* já montado com a ordem de importância. A tabela 1 descreve, de forma breve, os itens que serão desenvolvidos durante o projeto.

**Tabela 1. Product Backlog do Sistema**

ID	Nome	Descrição	Estimativa
001	Integração FOSUserBundle.	Configuração do <i>Bundle</i> FOSUserBundle que realiza as operações de Usuário.	16h
002	Integração AdminLTE	Integração com o <i>template</i> AdminLTE para a parte Administrativa do <i>software</i> .	8h
003	Integração EasyAdmin	Integração com o <i>Bundle</i> EasyAdmin para controle dos usuários cadastrados.	16h
004	Cadastro de Semestre	Cadastro de Semestres.	4h
005	Cadastro de Bancas	Cadastro de Bancas com regras de negócios como validações, permissões de usuário e outros.	24h
006	Cadastro de Projetos	Cadastro de Projetos com regras de negócios como validações, permissões de usuário e outros.	24h

007	Upload de Arquivo	Upload de arquivo para o sistema armazena artigo e código final.	4h
007	Criação de Relatório com Datas da Banca.	Criação de relatório destinado a interessados em assistir a banca,	5h
008	Criação de Relatório com informações sobre a banca.	Criação de relatório com informações passadas para a coordenação.	5h
009	Criação de Relatório de apresentação de banca	Criação de relatório com informações sobre a apresentação da banca como tema do projeto, descrição, notas, observações e assinatura dos professores avaliadores.	5h
010	Criação de Site para Página Inicial	Criação de uma página inicial para apresentação do <i>software</i> .	16h

## 5.2. Primeira *Sprint* - Configuração de Usuário e Cadastro de Semestres

**Tabela 2. Sprint Backlog da Sprint 1**

ID	Nome	Descrição	Estimativa
001	Integração FOSUserBundle.	Configuração do Bundle FOSUserBundle que realiza as operações de Usuário.	16h
002	Integração AdminLTE	Integração com o <i>template</i> AdminLTE para a parte Administrativa do <i>software</i> .	8h
003	Integração EasyAdmin	Integração com o Bundle EasyAdmin para controle dos usuários cadastrados.	16h
004	Cadastro de Semestre	Cadastro de Semestres.	4h

### 5.2.1 Integração FOSUserBundle

Foi iniciado a integração do FOSUserBundle, um pacote utilizado para fazer a configuração de usuários do sistema. Este pacote auxiliou pois já vem implementado toda a parte administrativa necessária para um *software*, como por exemplo login, recuperação de senha, cadastro de usuário, bastando apenas reutilizar o código existente, extender se for necessário e estilizar páginas do *front-end*.

### 5.2.2 Integração AdminLTE

O AdminLTE foi utilizado para a parte do sistema em geral. Por ser um tema muito utilizado e ter uma manutenção contínua, foi optado pelo uso do mesmo para o *software*. A integração consistiu em configurar as views com o HTML do template, armazenar e linkar as dependências ao código. Na figura 5 temos um exemplo de como ficará o sistema com a integração.

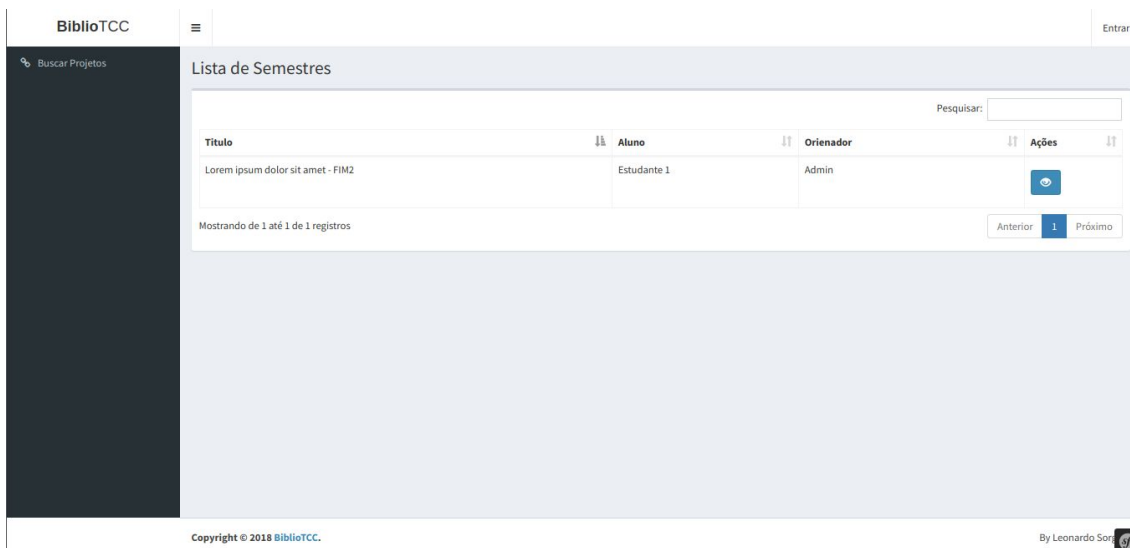


Figura 5. Painel Administrativo com o *template* AdminLTE

### 5.2.3 Integração EasyAdminBundle

O EasyAdminBundle é um pacote que, juntamente ao FOSUserBundle, cria um painel de controle de usuários já pronta, facilitando o controle de usuários cadastrados ou cadastros de novos usuários. A integração foi realizada após a integração com o FOSUserBundle pois ela complementa o pacote com a interface gráfica. Na figura 6 temos um exemplo do painel administrativo do pacote.

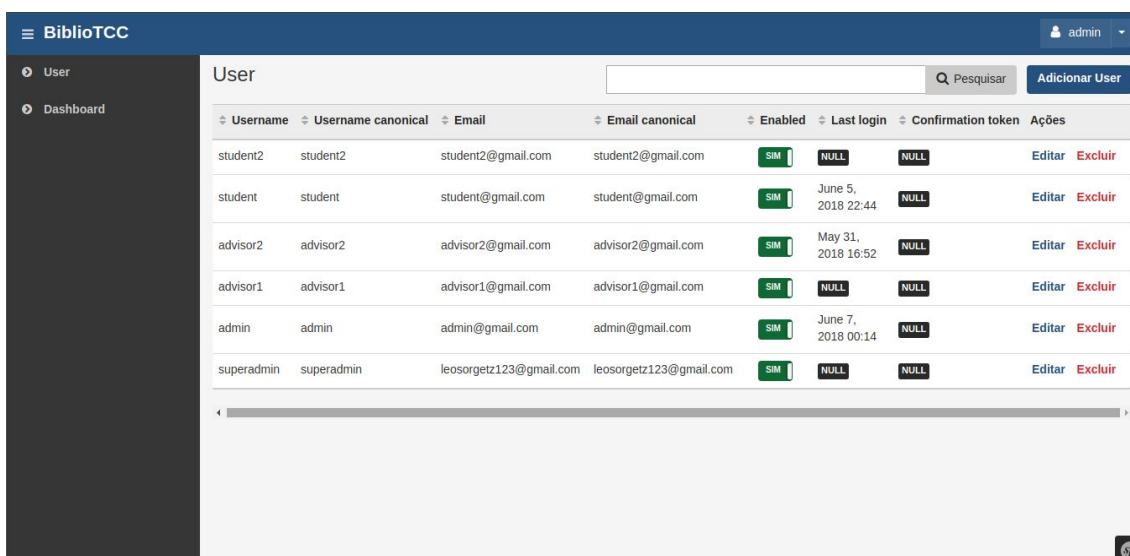


Figura 6. Painel Administrativo do EasyAdminBundle

### 5.2.4 Cadastro de Semestre

O cadastro de semestre servirá para organizar as bancas por semestre, assim, deixando a área, onde o professor encarregado por organizar banca visualiza o que já está criado, mais organizada para a criação de novas bancas, não misturando bancas de um semestre com as de outro.

### 5.3. Segunda Sprint - Gerenciamento de Bancas e Projetos

**Tabela 3. Sprint Backlog da Sprint 2**

005	Cadastro de Bancas	Cadastro de Bancas com regras de negócios como validações, permissões de usuário e outros.	24h
006	Cadastro de Projetos	Cadastro de Projetos com regras de negócios como validações, permissões de usuário e outros.	24h

#### 5.3.1 Cadastro de Bancas

O cadastro de bancas é utilizado para cadastrar as bancas do semestre, onde terá dados da banca que serão listados para que outros usuários acessem esses trabalhos para fins educacionais. As bancas só podem ser visualizadas por usuários caso tenham grau acima de 9 e que já estejam finalizadas com aval do professor orientador. Juntamente com o cadastro de bancas, é possível verificar as informações como título e descrição do projeto que foi avaliado na banca. Na figura 7 temos uma exemplo da página de cadastro de bancas.

**Figura 7. Página de Cadastro de Banca**

#### 5.3.2 Cadastro de Projetos

O cadastro de projeto consiste em permitir a o aluno postar seu PDS baseado nas datas de entrega e *status* da banca. Na prática, o projeto é criado ao mesmo tempo em que a banca é criada, já atribuindo ela aos usuários alunos, professor orientador e professores avaliadores da banca. Um aluno é permitido inicialmente a fazer postagem até a data prevista para a postagem inicial. Após essa postagem e a apresentação da banca, o aluno tem uma semana para realizar a postagem final com os ajustes apontados na banca. O aluno tem acesso a essas observações através do sistema, porém o grau só é liberado após o aval final do professor orientador. Na figura 8 temos um exemplo da página de inserção do arquivo do PDS.

**BiblioTCC**

Dashboard

Buscar Projetos

## Primeira Postagem do Projeto - BiblioTCC - Biblioteca e Gerenciador de Bancas de TCC

**Nome**

BiblioTCC - Biblioteca e Gerenciador de Bancas de TCC

**Descrição**

Neste artigo será apresentado o BiblioTCC, software, para auxiliar o professor responsável pela organização de bancas no gerenciamento das bancas de TCC e também servir como uma biblioteca aberta de trabalhos de conclusão, para que alunos possam estudar os projetos postados. O sistema foi desenvolvido com a linguagem PHP, na versão 7.1, utilizando o Framework Symfony 3.4.4 como base do projeto. A metodologia utilizada foi o Scrum, uma metodologia ágil para gestão e planejamento de projeto altamente utilizada.

**Arquivo**

Escolher arquivo | Nenhum arquivo selecionado

Postar Voltar

**Figura 8. Página de inserção do arquivo do PDS.**

## 5.4 Terceira *Sprint* - Geração de Relatórios

**Tabela 4. *Sprint Backlog* da Sprint 3**

<b>007</b>	Criação de Relatório com Datas da Banca.	Criação de relatório destinado a interessados em assistir a banca,	5h
<b>008</b>	Criação de Relatório com informações sobre a banca.	Criação de relatório com informações passadas para a coordenação.	5h
<b>009</b>	Criação de Relatório de apresentação de banca	Criação de relatório com informações sobre a apresentação da banca como tema do projeto, descrição, notas, observações e assinatura dos professores avaliadores.	5h
<b>010</b>	Criação de Site para Página Inicial	Criação de uma página inicial para apresentação do <i>software</i> .	16h

### 5.4.1 Criação de Relatório com Datas de Banca

O relatório com as datas de bancas é utilizado para que interessados na apresentação da banca saibam o horário e a data que a mesma irá ocorrer. Na figura 9 temos um exemplo de como será o relatório com data de bancas.

## Calendario de Trabalhos do Semestre - 2018/01

Data	30/06/2018 20:00
Aluno	Estudante 1
Orientador	Admin
Banca	<ul style="list-style-type: none"><li>• Avaliador 1</li><li>• Avaliador 2</li></ul>
Título do Trabalho	Lorem ipsum dolor sit amet - FIM2

Figura 9. Exemplo do calendário de bancas do semestre.

### 5.4.2 Criação de Relatório com informações sobre as Bancas.

O relatório de informações sobre a banca é um documento destinado a coordenação do curso com informações referentes às bancas do semestre como título, descrição, aluno e professor orientador. Na figura 10 temos um exemplo de como ficará o relatório com informações sobre as bancas.

## Relatório de Trabalhos do Semestre - 2018/01

Aluno	Estudante 1
Orientador	Admin
Título do Projeto	Lorem ipsum dolor sit amet - FIM2
Resumo do Projeto	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce eu feugiat tellus, non condimentum eros. Ut in eros risus. Donec rhoncus eros sit amet felis venenatis, vel pulvinar odio pulvinar. Praesent a gravida quam. Phasellus tortor leo, euismod vitae interdum sed, vehicula sit amet felis. Praesent interdum mattis nulla, facilisis eleifend mi pretium vel. FIM2

Figura 10. Exemplo do relatório com informação de bancas do semestre.

### 5.4.3 Criação de Relatório de apresentação de Banca

Ao final da apresentação da banca de TCC, os professores avaliadores da banca precisam documentar as observações a serem modificadas no sistema e artigo, juntamente da nota, título e descrição do *software*, nota de cada avaliador e por fim a assinatura do documento por esses professores. Ao final desta tarefa temos o ER final do *software*, como mostra a figura 9.



Figura 9. Diagrama de Entidade e Relacionamento

#### 5.4.4 Criação de Site para Página Inicial

Apesar de existir um template para o *backend* do sistema, foi necessário, para ficar mais agradável a experiência de usuário, a criação de uma página simples para direcionar o usuário para o *software*.

#### 5.5 Pós Planejamento

Após o desenvolvimento do projeto, iniciou-se a fase de finalização da banca onde foi marcado uma reunião para demonstração do *software* atual e então foi executado os testes para verificar possíveis erros. No anexo 1 temos as tabelas 5, 6, 7 e 8 que mostram os testes executados no sistema.

### 6. Considerações Finais

O BiblioTCC irá auxiliar os alunos do curso de Análise e Desenvolvimento de Sistema no acesso a artigos e códigos de trabalhos passados, na documentação de futuras bancas e na comunicação entre alunos e professores do curso durante o projeto final.

Durante o andamento do desenvolvimento, foram encontradas algumas dificuldades como estudo de um novo *framework* para o desenvolvimento do projeto, o *Symfony 3*, mas que auxiliou na produtividade com as ferramentas que o compõe.

Também foi uma dificuldade no desenvolvimento o fato de ficar a maior parte do tempo sem acesso a internet para a execução das tarefas, uma vez que a API do *framework* encontra-se *online*.

Ao fim do desenvolvimento o projeto será implementado para uso do curso, assim, finalizando o processo de desenvolvimento do *software*. Porém, apesar do projeto ser finalizado neste ponto, há espaço para mais funcionalidades, que poderiam ser implementadas em cima deste mesmo código fonte, como por exemplo um Gerador de Bancas, que geraria as bancas de TCC baseando-se na especialidade e disponibilidade dos professores avaliadores.

## 8. Referências

AGRELA, Lucas. **Estas são as 20 linguagens de programação mais usadas:** Relatório do Stack Overflow indica as linguagens mais populares, com base em respostas de mais de 100 mil programadores. 2018. Disponível em: <<https://exame.abril.com.br/tecnologia/estas-sao-as-20-linguagens-de-programacao-mais-usadas/>>. Acesso em: 05 jun. 2018.

ARAÚJO, Alex de. **Scrum para Iniciantes: o que é, como funciona, benefícios e cargos!:** Conheça a origem por trás do *framework* ágil mais famoso do mundo, veja como o Scrum funciona, descubra os seus benefícios e muito mais!. Disponível em: <<https://becode.com.br/scrum-para-iniciantes/>>. Acesso em: 05 jun. 2018.

BELEM, Thiago. **O que é e como funciona o jQuery.** 2010. Disponível em: <<http://blog.thiagobelem.net/o-que-e-e-como-funciona-o-jquery>>. Acesso em: 05 jun. 2018.

BROCANELLI, Diego. **TWIG – A SUPER ENGINE TEMPLATE PARA PHP.** 2016. Disponível em: <<http://www.diegobrocanelli.com.br/php/twig-a-super-engine-template-para-php/>>. Acesso em: 06 jun. 2018.

BRUSCH, Ricardo. **Doctrine 2- Mapeando seu banco de dados.** 2013. Disponível em: <<http://blog.ricardobrusch.com.br/doctrine-2-mapeando-seu-banco-de-dados/>>. Acesso em: 06 jun. 2018.

CARDOSO, André. **Composer para iniciantes.** 2014. Disponível em: <<https://tableless.com.br/composer-para-iniciantes/>>. Acesso em: 05 jun. 2018.

LAMIM, Jonathan. **MVC - O padrão de arquitetura de software:** Neste tutorial vamos conhecer um pouco sobre o padrão MVC.. 2009. Disponível em: <[https://www.oficinadanet.com.br/artigo/1687/mvc\\_-\\_o\\_padrao\\_de\\_arquitetura\\_de\\_software](https://www.oficinadanet.com.br/artigo/1687/mvc_-_o_padrao_de_arquitetura_de_software)>. Acesso em: 06 jun. 2018.

LEONE, Leonello de. **Bootstrap: o que é, porque usar e como começar com o framework.** 2017. Disponível em: <<https://becode.com.br/bootstrap-o-que-e-porque-usar-e-como-comecar/>>. Acesso em: 06 jun. 2018.

PHP Group. **O QUE o PHP pode fazer?** Disponível em: <[https://secure.php.net/manual/pt\\_BR/intro-whatcando.php](https://secure.php.net/manual/pt_BR/intro-whatcando.php)>. Acesso em: 05 jun. 2018.

PIZA, Pedro. **O que é e como usar o MySQL?** 2012. Disponível em: <<http://www.techtudo.com.br/artigos/noticia/2012/04/o-que-e-e-como-usar-o-mysql.html>>. Acesso em: 05 jun. 2018.

RIBEIRO, Leandro. **O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML.** 2012. Disponível em:



<<https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>>. Acesso em: 06 jun. 2018.

SCRUM. 2014. Disponível em: <<http://www.desenvolvimentoagil.com.br/scrum/>>. Acesso em: 05 jun. 2018.

SILVA, Giancarlo. **O que é e como funciona a linguagem JavaScript?** 2015.

Disponível em:

<<https://canaltech.com.br/internet/O-que-e-e-como-funciona-a-linguagem-JavaScript/>>. Acesso em: 06 jun. 2018.

SOUZA, Candido. **Enviar Iniciando com Symfony 2.** 2015. Disponível em:

<<https://tableless.com.br/iniciando-com-symfony-2/>>. Acesso em: 06 jun. 2018.

VENTURA, Plínio. **Entendendo o Diagrama de Atividades da UML.** 2016.

Disponível em: <<http://www.ateomomento.com.br/uml-diagrama-de-atividades/>>.

Acesso em: 06 jun. 2018.

## Anexos

### Anexo 1

**Tabela 5 - Testes de Banca**

ID	Descrição	Resultado Esperado	Resultado Obtido
1	Tentar cancelar uma banca sem ter permissão de acesso a essa funcionalidade	Sistema retorna um feedback “Acesso Negado”	Ok
2	Apresentar Banca sem ser orientador da banca	Sistema retorna um feedback “Acesso negado”	Ok
3	Finalizar Banca antes da entrega final do projeto	Sistema retorna um feedback “Não permitido finalizar banca sem a entrega final de um projeto”	Ok
4	Visualizar Relatório de Bancas do Semestre	Listar informações das Bancas do Semestre em página em branco	Ok
5	Visualizar Calendário de Bancas do Semestre	Visualizar informações sobre datas das Bancas do Semestre em página em branco	Ok
6	Visualizar Ata de Banca	Visualizar informações sobre a Banca específica em página em branco	Erro
7	Editar Banca sem ter permissão Admin	Sistema retorna um feedback “Acesso negado”	Ok

Referente ao item 6 da tabela 5, foi retornado para manutenção uma vez para a correção do erro.

**Tabela 6 - Testes de Projeto**

ID	Descrição	Resultado Esperado	Resultado Obtido
1	Realizar Primeira Postagem do Projeto após a data final	Sistema retorna um feedback “O prazo para postagem já está estourado.”	Ok
2	Realizar Ultima Postagem do Projeto após a data final	Sistema retorna um feedback “O prazo para postagem já está estourado.”	Ok

3	Tentar postar projeto sem ser o responsável por ele	Sistema retorna um feedback “Acesso negado”	Ok
4	Tentar realizar <i>download</i> de um projeto não finalizado	Sistema retorna um feedback “Acesso negado”	Ok
5	Tentar realizar <i>download</i> de um projeto com nota abaixo de 9.	Sistema retorna um feedback “Acesso negado”	Ok

**Tabela 7 - Testes de FOSUserBundle**

ID	Descrição	Resultado Esperado	Resultado Obtido
1	Cadastrar Usuário	Cadastro realizado porém desativado.	Ok
2	Recuperar Senha	Envio de email com link para mudança de senha	Ok
3	Mudar Senha com <i>link</i> recebido por Email	Mudança de Senha Realizada	Ok
4	Lembrar Me ao Logar no Sistema	Permanecer logado sempre ao entrar no sistema.	Erro

Referente ao item 4 da tabela 7, foi retornado para manutenção uma vez para a correção do erro.

**Tabela 8 - Testes de EasyAdmin Bundle**

ID	Descrição	Resultado Esperado	Resultado Obtido
1	Tentativa de acesso sem ser Admin	Acesso Negado	Ok
2	Cadastrar Usuário	Usuário Cadastrado	Ok
3	Editar Usuário	Usuário Editado	Ok
4	Inativar Usuário	Usuário Inativado	Ok