

Software para Gerenciamento de Atividades Complementares

Maria Carolina Silva Vanuchi¹, Frank Willian Cardoso de Oliveira¹, Daniela Eloise Flôr¹, Késsia Rita da Costa Marchi¹

¹Curso de Tecnologia em Análise e Desenvolvimento de Sistemas - Instituto Federal do Paraná – Campus Paranavaí

carolinavanuchi21@gmail.com, frank.willian@ifpr.edu.br,
daniela.flor@ifpr.edu.br, kessia.marchi@ifpr.edu.br

Abstract. *This work presents the development of software to assist in managing for complementary activities that are necessary in formation of students in some educational institutions. For the preparation of that was done a survey of requirements in the context of the Instituto Federal do Paraná, campus Paranavaí. Between the functionalities implemented are manages registration data, hours in calculation of certificates, authentication and validation, each user being able to follow the status of certificates on your restricted area, thus always obtaining feedback from the current situation. Among the benefits acquired through is use are the optimization of service and economy of the material resources.*

Resumo. *Este trabalho apresenta o desenvolvimento de um software para auxiliar no gerenciamento de atividades complementares, que são necessárias na formação de alunos em algumas instituições de ensino. Para sua implementação foi realizado um levantamento de requisitos no contexto do Instituto Federal do Paraná, campus Paranavaí. Entre as funcionalidades implementadas estão a gerencia de dados cadastrais, cálculo de horas dos certificados, autenticação e validação, sendo possível cada usuário acompanhar a situação dos certificados em sua área restrita, obtendo sempre um feedback da situação atual. Entre os benefícios adquiridos com seu uso estão a otimização do serviço e economia de recursos materiais.*

1 Introdução

As atividades complementares são um requisito obrigatório para que alunos de algumas instituições concluam seus cursos. Assim precisam participar de palestras, minicursos entre outros, dentro ou fora da instituição, para cumprir a carga horária mínima estipulada pelo colegiado do curso.

No Instituto Federal do Paraná, campus Paranavaí, uma instituição de ensino pública vinculada ao Ministério da Educação (MEC) por meio da Secretaria de Educação Profissional e Tecnológica (Setec), o gerenciamento dessas atividades são feitas manualmente com auxílio de planilhas. Devido o aumento quantitativo de discentes esse trabalho tornou-se moroso, e sujeito a perdas de informações, além de possíveis erros de contabilização das horas dos certificados.

Para sanar o problema encontrada na instituição foi proposto e desenvolvido um *software* intitulado *Cronos*, o mesmo auxilia nesse gerenciamento e possibilita a substituição do que anteriormente era realizado com planilhas e cópias físicas de

certificados, por cadastros, *upload* e áreas que podem ser realizados acompanhamento das horas contabilizadas, das situações em que os certificados se encontram e emissão de relatórios. Entre as funcionalidades apresentadas no *Cronos* estão o gerenciamento dos certificados, autenticação e validação, além da contabilização automática das horas conforme a equivalências das atividades.

No decorrer das próximas seções e subseções serão apresentados o ciclo de informação implementado, tecnologias e diagramas utilizados, como também partes do próprio *Cronos*.

2 Atividades Complementares

As atividades complementares (AC) permitem a formação interdisciplinar, possibilitando que o aluno adquira conhecimentos e competências que aprimore sua formação. Para uma melhor abrangência e organização as AC são categorizadas em grupos e subdivididos em atividades, são nessas em que os certificados são relacionados. Na Figura 1 é exibida de genericamente essa categorização.



Figura 1. Visão genérica de grupo, atividade e certificado

Como apresentado na Figura 1 os grupos englobam as atividades e nesses estão contidos os certificados. Entretanto cada grupo e atividades possuem quantidades de horas diferenciadas. As equivalências nas atividades também possuem diferenças, sendo essas utilizadas no cálculo de horas de contabilização dos certificados.

Como os cursos possuem características distintas, logo possuem horas e equivalências diferenciadas, mas no mesmo curso pode ocorrer diferenças de valores entre as turmas, para tratar essa possibilidade o *Cronos* possui áreas denominada de “Grupo Turma” e “Atividade Turma”, na Figura 2 é apresentada uma breve ideia dessas funcionalidades.



Figura 2. Visão genérica de grupo turma e atividade turma

A Figura 2 exibe as áreas mencionadas, podendo ser compreendido que “Grupo Turma” é uma interseção entre grupo e turma; as “Atividade Turma” são uma interseção entre atividade e turma, nesses são cadastrados hora mínima, máxima e no último exposto as equivalências, ambos conforme a turma selecionada.

2.1 As Atividades Complementares no Campus

No IFPR campus Paranavaí as atividades complementares são categorizadas em três grupos, segundo o regulamento “IIP N° 02 INSTRUÇÃO INTERNA DE PROCEDIMENTOS DO CÂMPUS IIPC N° 02, DE 31 DE MARÇO\ DE 2015– DIREÇÃO GERAL DO IFPR CÂMPUS PARANAVAÍ”, sendo essas:

- Atividades de ensino: são realizadas no âmbito do próprio curso;
- Atividades de pesquisa, extensão e inovação: se referem à produção do conhecimento, por meio de estudos específicos, visando desenvolver no estudante a vocação, o interesse e a aptidão para a investigação, com vistas ao avanço científico, cultural, tecnológico e socioeconômico da região e do país
- Atividades de formação social, humana e cultural: referem a formação do estudante no caráter social, humana e cultural.

Na Tabela 1 é mostrado alguns exemplos de atividades e equivalências presentes no curso superior em Análise e Desenvolvimento de Sistemas, na turma de 2014.

Tabela 1. Exemplos de atividades turmas

Grupo	Atividade	Máximo de Horas	Equivalência
Atividades de Ensino	Cursos de qualificação em áreas correlatas ao curso	40	1:2
Atividades de Pesquisa, Extensão e Inovação	Publicações em revistas técnicas	40	1:1
Atividades de Formação Social, Humana e Cultural	Participação como expositor em exposição artística ou cultural	15	1:1

Como pode ser observado na Tabela 1 cada atividade está referenciada a um grupo, possuindo horas máximas e equivalências, podendo essas quantidades modificar-se nas turmas.

2.2 Ciclo de Informação Aplicado

Com o objetivo de otimizar o processo de gerenciamento das atividades complementares do IFPR campus Paranavaí, o *Cronos*, implementou regras existentes de autenticação e validação de certificados, para compreensão desse ciclo de informação é apresentado um fluxograma na Figura 3.

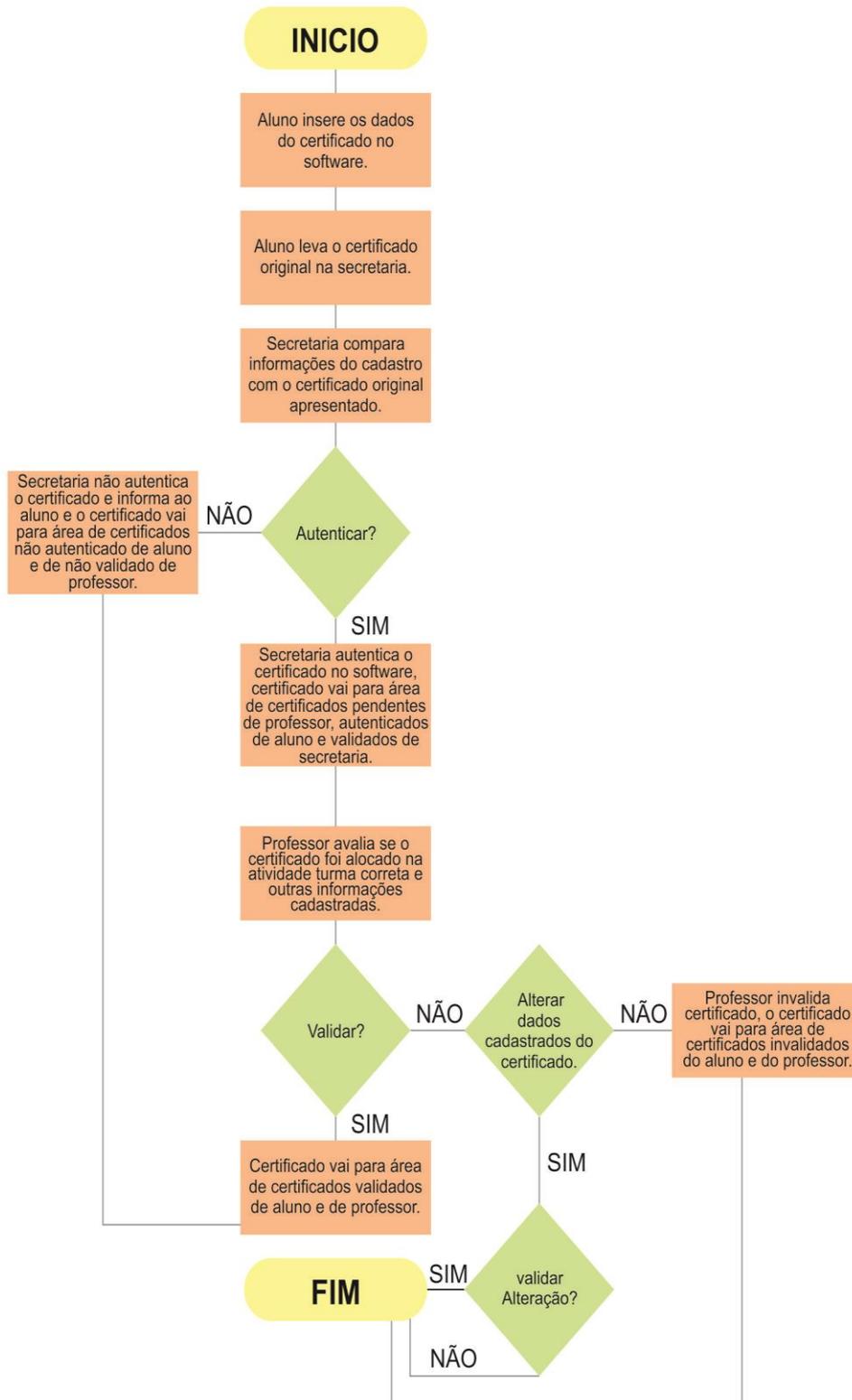


Figura 3. Fluxograma de validação de certificado no Cronos

Para que o processo inicie como apresentado no fluxograma da Figura 3, o aluno precisa cadastrar algum certificado e realizar o *upload* do arquivo em *portable document format* (pdf), após isso deverá ir no dia definido pela secretária apresentar o certificado original, nesse momento o servidor da secretaria deverá verificar sua autenticidade, comparando com as informações do cadastro. Na sequência o certificado ou declaração é encaminhado para o professor responsável, que verificará se as informações inseridas pelo aluno estão corretas, nessa etapa o professor avaliará o mérito da atividade desenvolvida, podendo nesse momento validar ou não o certificado ou declaração.

O aluno pode verificar em qual etapa do processo o certificado está, utilizando funcionalidades de certificados autenticados e validados, podendo também verificar quais foram invalidados tanto pela secretaria quanto pelos professores.

É importante ressaltar algumas regras, apresentadas abaixo, que foram implementadas no *Cronos* para que estivesse adequado ao contexto da instituição:

- Alteração das horas computadas conforme a mudança de Projeto Pedagógico do Curso (PPC);
- Cada turma possui quantidade mínima e máxima de horas tanto para grupos quanto para atividades e equivalência de horas para atividades;
- Realiza o ciclo de autenticação na secretária e validação pelo professor responsável.
- Validação de hora máxima para atividade e grupo.

3 Metodologia

No decorrer do processo de desenvolvimento de *software* são empregadas metodologias, podendo ser tanto ágeis quanto tradicional, padrões arquiteturais e tecnologias para o desenvolvimento.

Para o gerenciamento desse projeto foram utilizados princípios da metodologia ágil e práticas do *Extreme Programming*, e para a definição da arquitetura do *software* foi implementado o *model-view-controller* (MVC). No decorrer dessa seção serão apresentados os princípios, tecnologias e padrão utilizado como também a forma implementada.

3.1 Metodologias Ágeis

As metodologias ágeis são uma forma de realizar a gestão do desenvolvimento de *software* de forma iterativa e incremental. Segundo o manifesto para o desenvolvimento ágil de *software* deve ser mais valorizados os “Indivíduos e interação entre eles mais que processos e ferramentas”, “*Software* em funcionamento mais que documentação abrangente”, “Colaboração com o cliente mais que negociações de contratos” e “Responder a mudanças mais que seguir um plano”.

Para seguir o princípio das metodologias ágeis foram realizados alguns *feedbacks* com alguns *stakeholders*, também analisadas e aplicadas sugestões de adaptações e evoluções dentro do *Cronos*. Além desses pontos foram utilizados princípios de passos de bebês e melhorias presentes na metodologia *Extreme*

Programming (XP), na Tabela 2 abaixo são explicadas esses princípios e momentos utilizados.

Tabela 2. Princípios do *Extreme Programming* utilizados

Princípio	Definição	Momento Utilizado
Passos de bebês	Inferir que realizar pequenas mudanças de cada vez, reduz o tempo de correção e diminuem o impacto dessa modificação caso a mesma esteja incorreta.	Utilizado durante o processo de desenvolvimento, sendo assim realizado pequenas modificações e testes nas mesmas.
Melhorias	“Software não é ouro é alface: um bem perecível. Se não for aprimorado ao longo do tempo, acaba estragando. Essa frase, foi atribuída a Brian Behlendorf no livro <i>The World is Flat</i> ” (Manhães Teles Vinícius, Melhoria, 2006).	Para seguir esse princípio foram aplicadas refatorações durante o processo de desenvolvimento do <i>Cronos</i> .

3.2 Padrão MVC

No padrão arquitetural do projeto foi utilizado o MVC (*model-view-controller* em sua tradução modelo-visão-controle), sendo que a definição dessas camadas segundo Décio Heinzelmann Luckow e Alexandre Altair de Melo (2010, p.181) são:

- *Model*: Camada do MVC que representa o acesso a dados e regra de negócio(...);
- *View*: Representa tudo que compõe a interface gráfica de um sistema. É na *View* que as informações geradas pela *Model* serão exibidas, e é por meio da camada *View* que o usuário pode inserir os dados para utilização dos sistema(...);
- *Controller*: conecta *View* a *Model*, responsável por buscar as informações da camada *Model* para gerar a *View* e por receber as informações da *View* e enviar para *Model*(...).

Na Figura 4 é ilustrado esse padrão arquitetural, mostrando as relações entre as camadas.

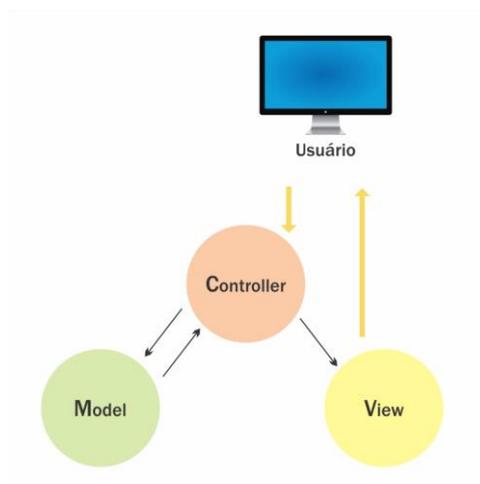


Figura 4. Modelo MVC

Como apresentado na Figura 4, o usuário realiza uma solicitação a camada *view*, essa comunica-se com *controller* o mesmo realiza o processamento e solicita informações a camada de *model*, essa envia para o *controller* que encaminha para a *view*.

3.3 Tecnologias Aplicadas

No desenvolvimento foram utilizadas tecnologias livres e presentes no mercado de *software* atual, essas são mencionadas na Tabela 3 que segue abaixo.

Tabela 3. Tecnologias utilizadas

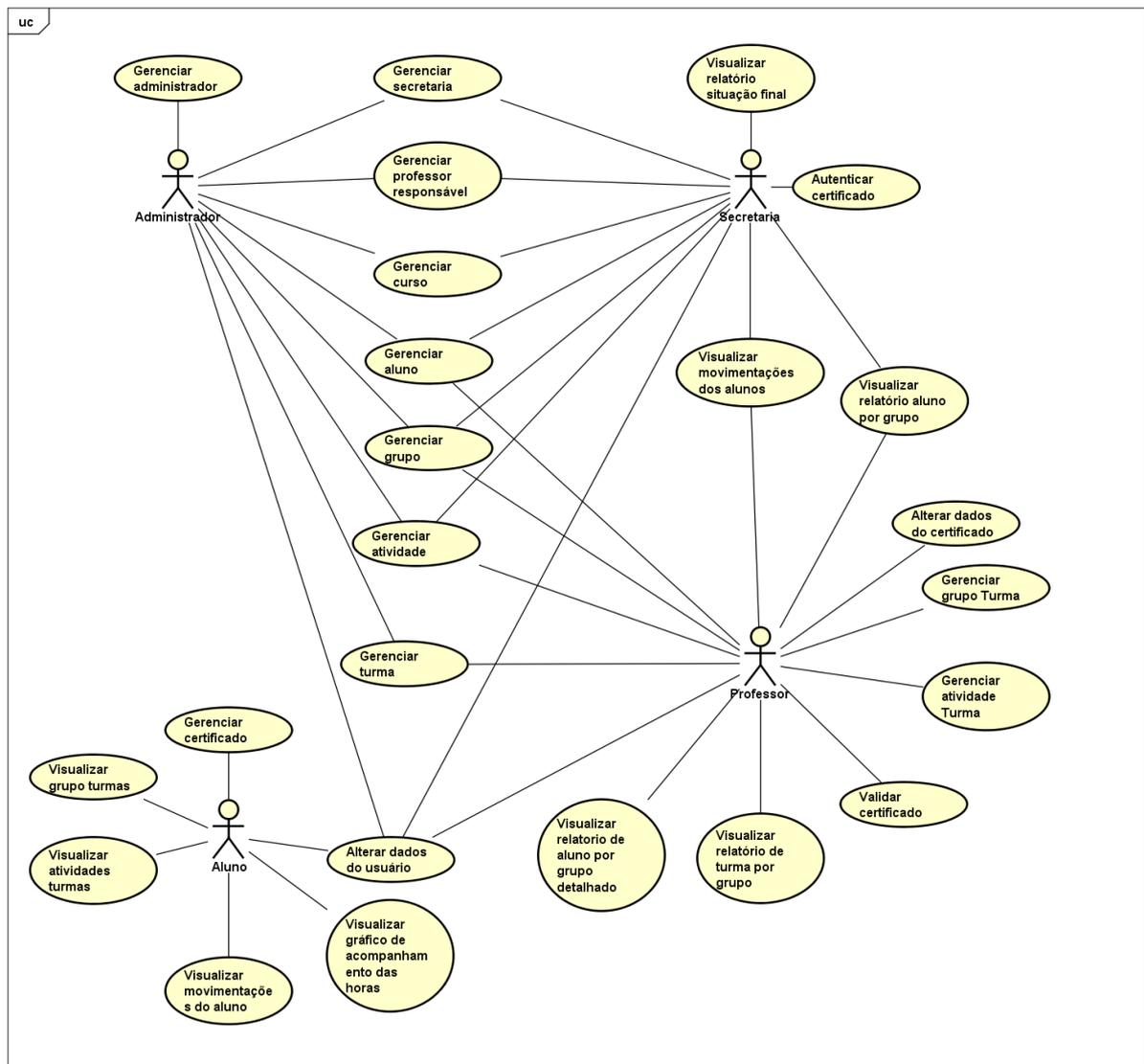
Tecnologia	Descrição	Utilização
<i>Beans Validation</i>	Possibilita a validação dos valores fornecidos com uso de anotações na camada de modelo, o JSR 303 retorna uma mensagem conforme a validação.	Essa especificação foi usada no sistema, com o objetivo quando o usuário burlar as validações das camadas anteriores, retornar uma mensagem para o mesmo.
<i>Bootstrap</i>	<i>Framework fron-end</i> de código livre, que possui classes CSS responsivas, além de componentes.	Esse <i>framework</i> foi utilizado no <i>template</i> do sistema.
<i>Java Persistence API e Hibernate</i>	O JPA segundo Natália de Lima do Nascimento (2009, p. 14) “disponibiliza um conjunto de interfaces de programação que permitem o armazenamento de objetos Java chamados de entidades do BD”, ou seja, serve para persistência dos dados para o banco de dados. Já o <i>Hibernate</i> segundo Christin Bauer Gavin King (2007, p. 31) “é uma ferramenta de mapeamento objeto/relacional completa que fornece todos os benefícios de um ORM”.	Essas foram utilizadas no mapeamento e persistência dos dados.
<i>JavaServer Faces e Primefaces</i>	O JSF segundo Décio Heinzelmann Luckow e Alexandre Altair de Melo (2010, p.72) “é uma especificação para um <i>framework</i> de componentes para desenvolvimento <i>web</i> em Java.”, enquanto o <i>Primefaces</i> é uma biblioteca de componentes para JSF.	Na elaboração do <i>software</i> ambos foram utilizados nas páginas, mas com maior ênfase nos componentes do <i>Primefaces</i> juntamente com tema nativo do <i>Bootstrap</i> .
<i>Maven</i>	<i>Maven</i> segundo o site oficial desse é “ <i>a software project management and comprehension tool</i> ”, sendo assim é uma ferramenta que auxilia no desenvolvimentos de projetos na linguagem de programação <i>Java</i> , algumas de suas funções é o gerenciamento de dependências, controle de versões de artefatos, simplificar o processo de <i>build</i> .	Tecnologia utilizada para o gerenciamento de dependências.
<i>Spring Security</i>	<i>Framework</i> de segurança para sistema desenvolvidos em <i>Java</i> .	Utilizado na aplicação para verificação da autenticação e autorização dos usuários.

<i>Bitbucket</i>	Serviço de versionamento de projetos.	Utilizado para realizar o versionamento do <i>Cronos</i> .
<i>Eclipse Mars 2</i>	IDE (<i>Integrated Development Environmet</i> , em sua tradução Ambiente de Desenvolvimento Integrado).	Ambiente utilizado para o desenvolvimento.
<i>MySQL</i>	Gerenciador de Bancos de Dados (SGBD).	Utilizado para realizar o armazenamento de dados.
<i>TomCat</i>	Servidor de aplicação.	Utilizado para execução da aplicação <i>web</i> .
<i>Java</i>	Linguagem de programação orientada a objetos.	Utilização realizada para a programação do <i>Cronos</i> .
<i>Astah Community</i>	<i>Software</i> de modelagem UML (Linguagem de Modelagem Unificada).	Seu uso foi na modelagem dos diagramas de caso de uso, classe e sequência.

4 Modelagem Implementada

Os diagramas elaborados para a modelagem do *Cronos* foram os diagramas de classe, caso de uso e sequência, esses serão apresentados na seção atual.

No diagrama de caso de uso estão presentes atores, funcionalidades do sistema e interações entre os mesmos, podendo ser afirmado por Eduardo Bezerra (2007 p. 53) “O modelo de caso de uso é uma representação das *funcionalidades* externamente observáveis do sistema e dos elementos *externos* ao sistema que interagem com ele.”. Na figura 5 é apresentado o diagrama de caso de uso utilizado no *Cronos*.



powered by Astah

Figura 5. Diagrama de Caso de Uso do Software Cronos

Como apresentada na figura 5 os atores no *software* são administrador, secretaria, professor e alunos, todos possuem funções e permissões específicas, entretanto algumas delas são iguais para alguns atores como cadastro de grupo, atividade, curso e turma.

O próximo diagrama a ser apresentado é o de classe, nesse estão especificadas as classes do mesmo e seus relacionamentos, podendo essa definição ser afirmada por Gilleanes T.A. Guedes (2011 p. 31) “Como o próprio nome diz, define a estrutura das classes utilizadas pelo sistema, determinando os atributos que cada classe e métodos tem, além de estabelecer como as classes se relacionam e trocam informações entre si.”, para a modelagem desse *software* foi optado não especificar os métodos, pois no diagrama é exibida as classes de entidade onde estão os dados que serão persistidos e os métodos de *get* e *set*, que sua função é permitir acessar e inserir dados nos atributos das classes. Na Figura 6 é apresentado o diagrama de classe utilizado.

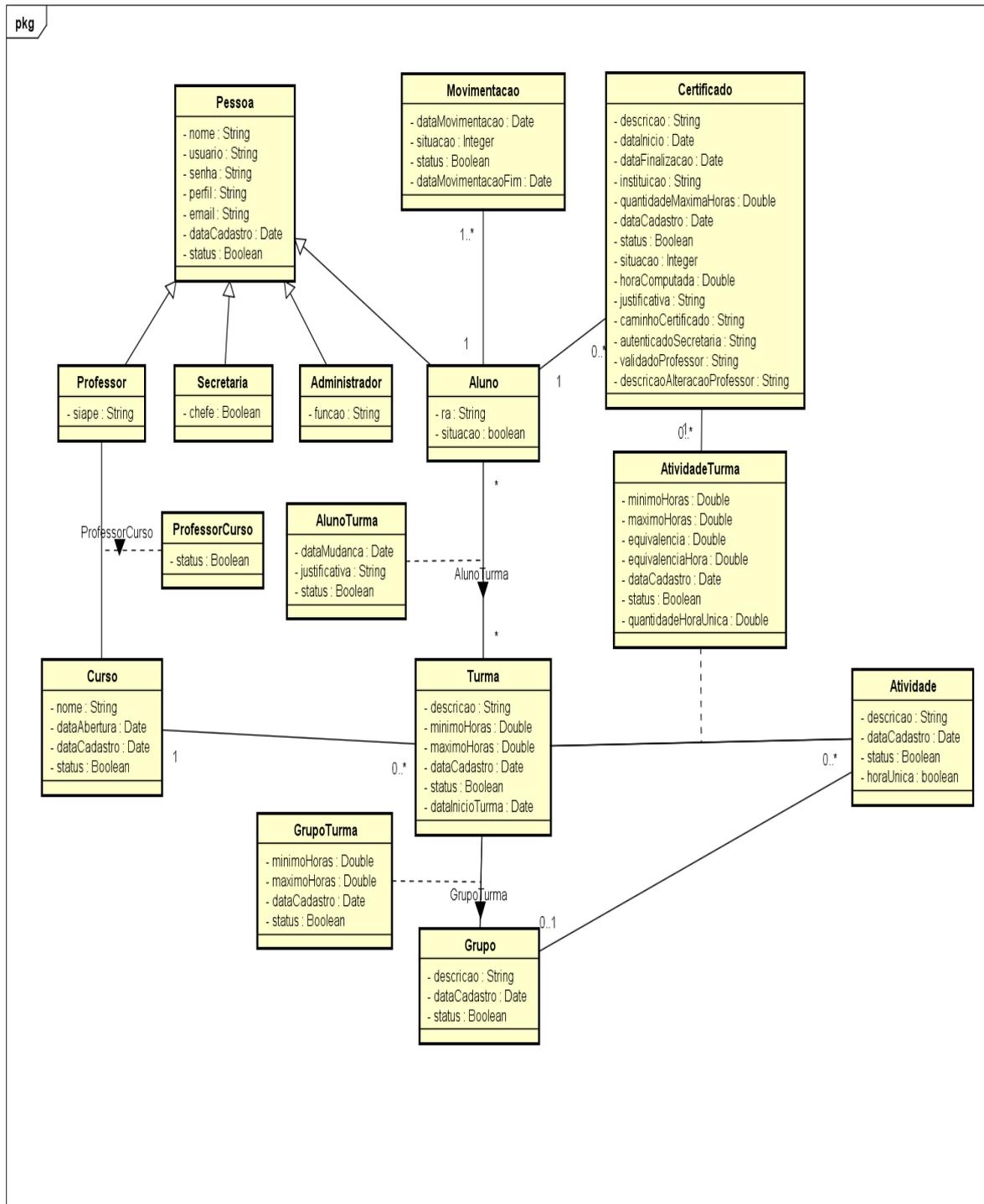


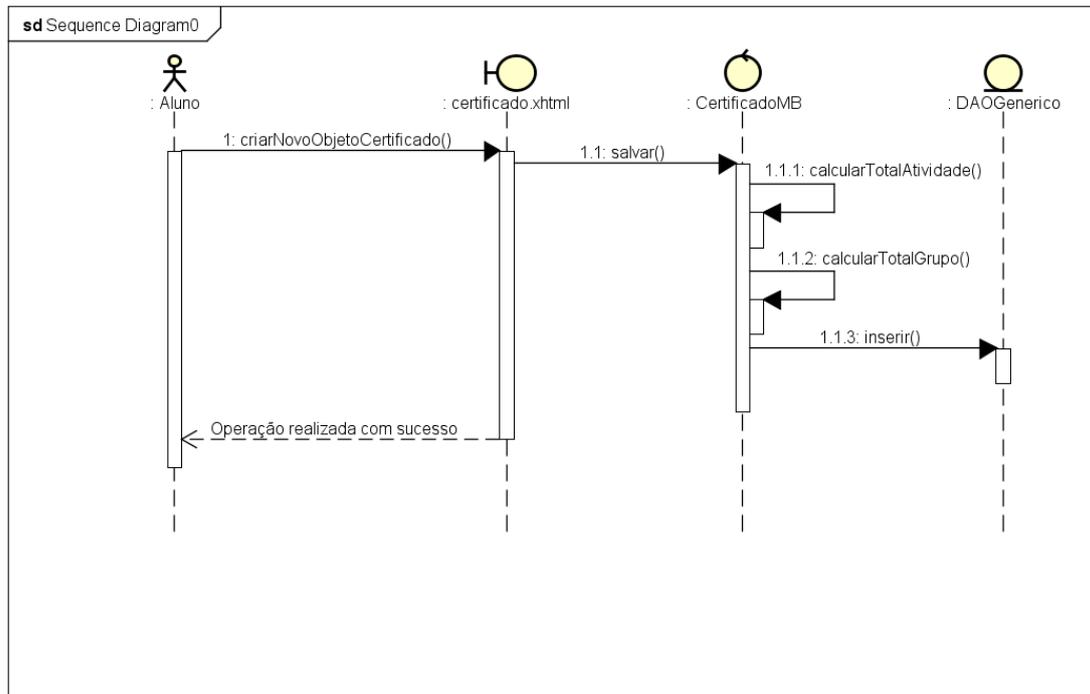
Figura 6. Diagrama de Classe do Software Cronos

Na Figura 6 observa-se classe como “ProfessorCurso”, a mesma foi modelada para situações em que um professor responsável possuir mais de um curso e um curso possuir mais de um professor responsável, surgindo um relacionamento muitos para muitos. Na Tabela 4 serão explicadas as funções de cada classe de entidade apresentadas.

Tabela 4. Descrição das classes de entidade

Classe	Descrição
Administrador	Classe onde estão os atributos referentes ao usuário administrador.
Aluno	Classe onde estão os atributos referentes ao usuário aluno.
AlunoTurma	Classe onde estão os atributos referentes a mudança de turma do aluno, mantendo um histórico dessas movimentações.
Atividade	Classe onde estão os atributos referentes as atividades dos grupos.
AtividadeTurma	Classe onde estão os atributos referentes as diferenças de informações que podem ter de um atividade referente a uma turma.
Certificado	Classe onde estão os atributos referentes aos dados do certificado.
Curso	Classe onde estão os atributos referentes ao curso.
Grupo	Classe onde estão os atributos referentes aos grupos.
GrupoTurma	Classe onde estão os atributos referentes as diferenças de informações que podem ter de um grupo referente a uma turma.
Movimentação	Classe que estão as informações de trancamento e destrancamento do aluno, mantendo um histórico dessas movimentações.
Pessoa	Classe onde estão todos os atributos iguais dos usuários.
Professor	Classe onde estão os atributos do usuário professor.
ProfessorCurso	Classe que permite que um professor seja responsável por muitos cursos e que para o mesmo tenha mais de um professor.
Secretaria	Classe onde estão os atributos do usuário secretaria.
Turma	Classe onde estão os atributos da turma.

Para definir comportamento das classes foi utilizado o diagrama de sequência, que possui a finalidade de exibir o comportamento da passagem de informações entre as classes, segundo Gilleanes T.A. Guedes (2011 p. 33) “O diagrama de sequência é um diagrama comportamental que preocupa-se com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em um determinado processo.” Para o *Cronos* desenvolveu-se um diagrama de sequência o mesmo supria as necessidades lógicas de compreender o comportamento dos dados nas classes, sendo esse diagrama exibido na Figura 7, exemplificando o cadastro de certificados.



powered by Astah

Figura 7. Diagrama de Sequência de cadastro de certificados

Na Figura 7 o usuário aluno requisita o cadastro de um novo certificado, para isso é solicitado a partir da página certificado.xhtml o método criarNovoObjetoCertificado(), após o preenchimento dos dados, requeresse salvar as informações, antes que as mesmas sejam persistidas, é verificado se a quantidade de horas em atividade e grupo já estão completados para o aluno.

Na seção 5, serão apresentadas algumas funcionalidades do *software* e explicadas as mesmas.

5 Solução Implementada

O *Cronos* utilizado no gerenciamento das horas dos certificados das atividades foi desenvolvido para o ambiente *web*, expandindo assim seu uso para qualquer dispositivos que possua algum *browser* e que esteja conectado à *internet*. Com a implementação de um *design* responsivo possibilita a visualização em diversas resoluções.

Para que os usuários utilizem apenas suas funções no *software* foram desenvolvidas quatro áreas distintas, sendo elas área do administrador, aluno, secretaria e professor, no decorrer dessa seção serão apresentadas algumas das funcionalidades.

Primeiramente foi optado que na tela inicial seja apresentado quatro ícones referentes aos módulos, como é exibido na Figura 8.

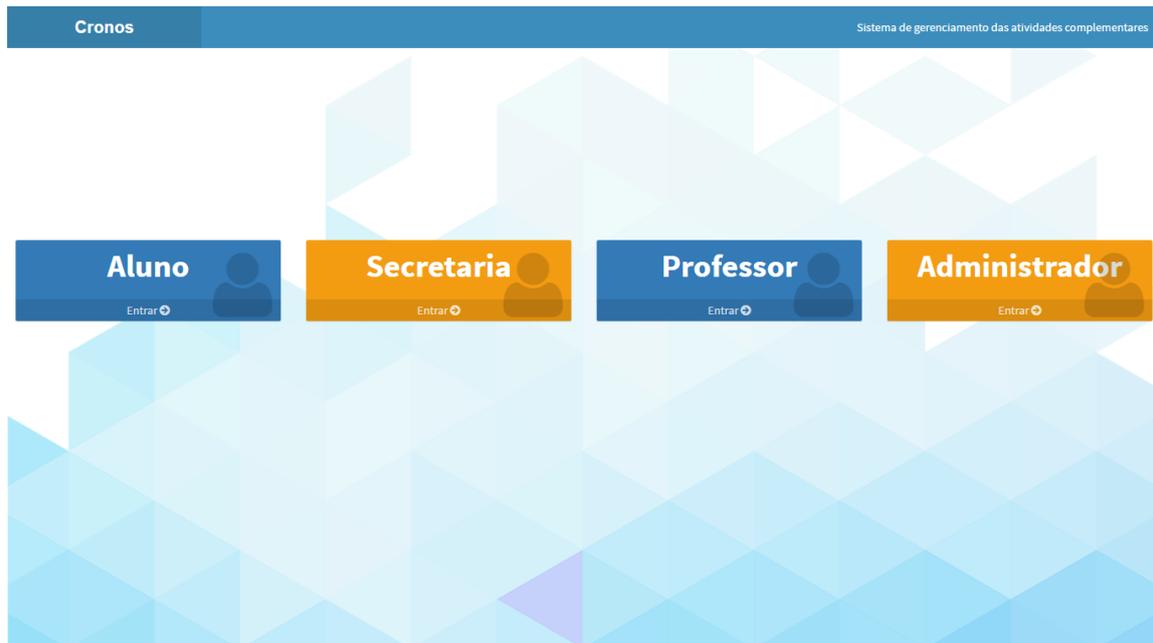


Figura 8. Tela inicial do software Cronos

Posteriormente a ação do usuário selecionar o módulo que deseja acessar, será redirecionado para a tela de *login*, nessa deverá inserir seu usuário e senha, essa tela é apresentada na Figura 9. O controle de acesso é realizado utilizando o *framework Spring Security*, as senhas salvas foram criptografadas com uso do algoritmo *bcrypt*.

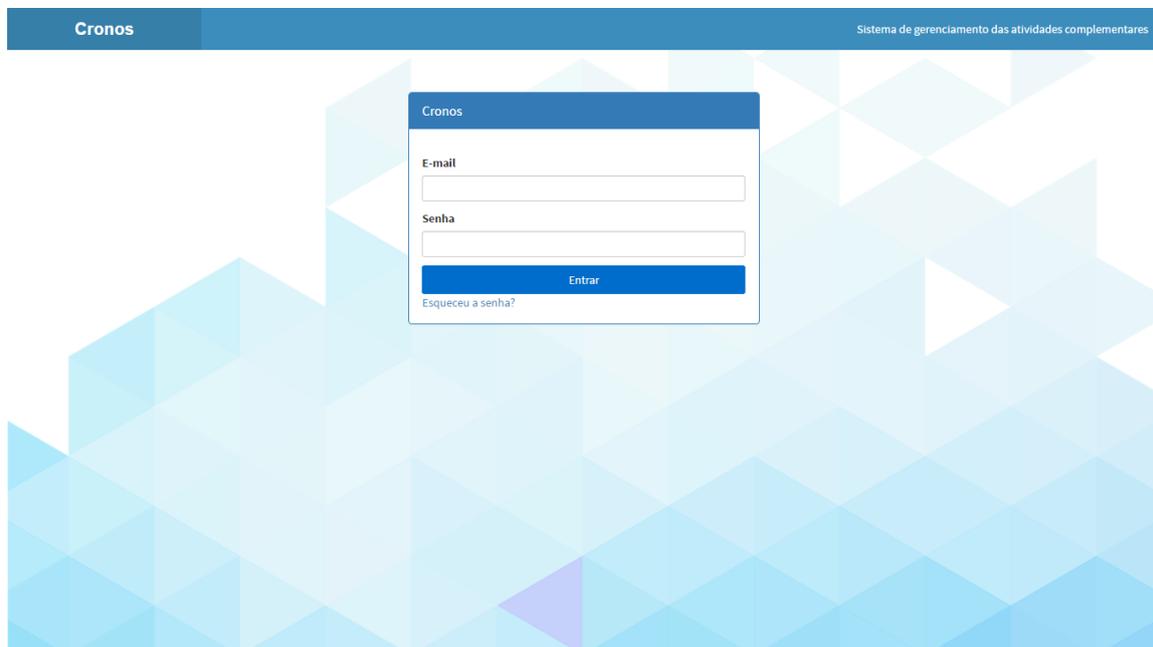


Figura 9. Tela de login do software Cronos

Após serem informadas as credenciais do usuário, serão verificadas a autenticação e autorização, caso estejam corretas, ele será redirecionado para a área de seu usuário. Na Figura 10 é exibida a página inicial, sendo que para todos os tipos de usuários é a mesma.

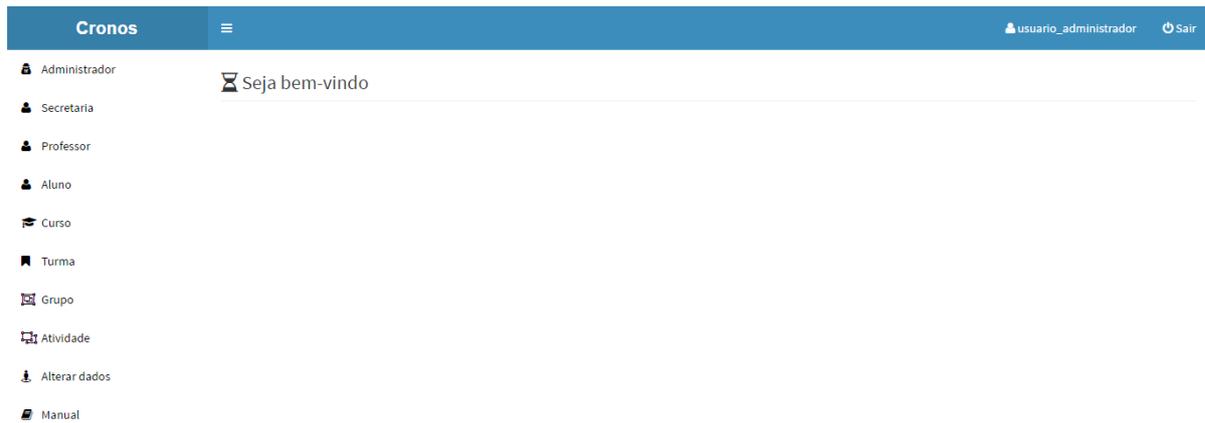


Figura 10. Tela inicial do software Cronos, na área administrador

Mesmo que seja a mesma tela para todos os usuários o menu lateral divergem conforme o tipo de usuário no *Cronos*.

Na Figura 11 é possível visualizar a funcionalidade de autenticação de certificados da secretária, já na Figura 12 exibe-se a mesma tela, mas em uma resolução de um *smartphone*.

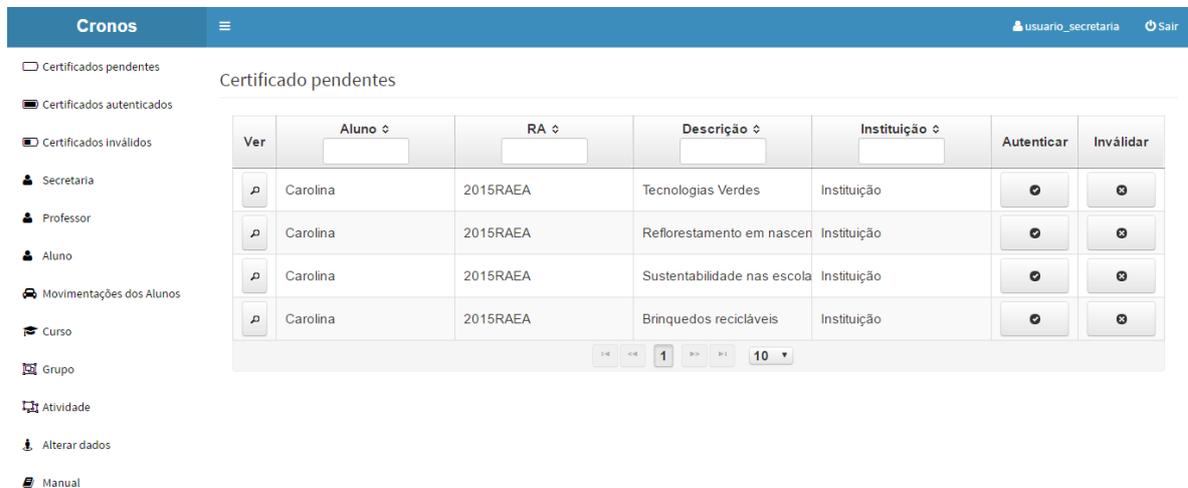


Figura 11. Tela de certificados pendentes do software Cronos, na área secretaria



Figura 12. Tela de certificados pendentes do software Cronos, na área secretaria, em resolução de um smartphone

Seguindo o ciclo de informação o professor pode validar os certificados, na Figura 13 é exibida a tela em que o mesmo pode realizar essa ação; os certificados só irão para esse local após a autenticação da secretária, sendo apenas das turmas dos curso(s) responsável pelo mesmo.

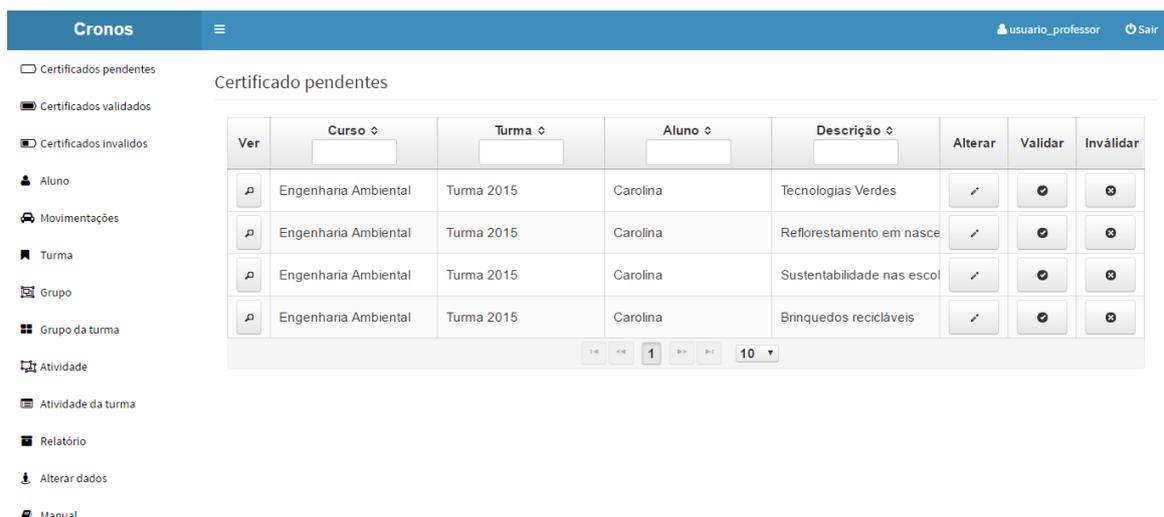


Figura 13. Tela de certificados pendentes do software Cronos, na área professor

Os alunos podem realizar o acompanhamento dos certificados autenticados e validados utilizando o *Cronos*, por meio da área de “Certificados autenticados” e “Certificados validados” que podem ser visualizadas na Figura 14, na mesma representação é exibido o acompanhamento de horas, onde o aluno pode verificar as horas validadas e contabilizadas com o auxílio de um gráfico e um relatório.



Figura 14. Tela de acompanhamento das horas dos certificados validados software Cronos área aluno.

Para que os professores acompanhem as horas contabilizadas de seus alunos foram desenvolvidos três relatórios, um exibindo horas totais dos certificados validados de cada aluno, outro de cada grupo para as turmas e por fim um que exibe os certificados cadastrados dos alunos em cada grupo. Na figura 15 é exibido *layout* dos relatórios, exibindo um pequeno exemplo do primeiro mencionado.

Cronos					15/10/2016 14:56
Sistema de Gerenciamento de Atividades Complementares					
Nome do aluno:	Carolina				
Grupo:	Atividades de Ensino				
Atividade	Certificado	Instituicao	Horas Cadastradas	Horas computadas	
Monitoria	Tecnologias Sustentaveis	Instituição	100.0	50.0	

Figura 15. Horas validados de cada aluno por grupo

6 Conclusões e Trabalhos Futuros

Tendo em vista os requisitos levantados no contexto do Instituto Federal do Paraná, campus Paranavaí, o *software Cronos* obedece as regras de negócio apresentadas até o ano de 2016, sendo assim seu uso pode auxiliar e melhorar o gerenciamento das atividades complementares, possibilitando um melhor acompanhamento e reduzindo o uso de materiais como cópias dos certificados.

Os próximos passos a serem realizados no *software Cronos* dentro da instituição de ensino seriam realizar teste de aceitação, podendo assim verificar a opiniões por parte dos usuários com a utilização do mesmo e se o *Cronos* atende as necessidades requisitadas, como também a integração desse com diferentes *software* que serão desenvolvidos no campus.

7 Referências

Bezerra E. (2007) “Princípios de ANÁLISE e PROJETO de SISTEMAS com UML”, segunda edição, editora Elsevier.

Gilleanes T. A. Guedes (2011) “UML 2 UMA ABORDAGEM PRÁTICA”, segunda edição, editora Novatec.

Por dentro do PrimeFaces 2.2 – Artigo Revista Java Magazine 93. Disponível em: <http://www.devmedia.com.br/por-dentro-do-primefaces-2-2-artigo-revista-java-magazine-93/21651>. Acessado em: 7 de maio de 2016.

Java EE 6: Começando com Bean Validation. Disponível em: <http://blog.caelum.com.br/java-ee-6-comecando-com-bean-validation/>. Acessado em: 11 de maio de 2016.

Spring Security. Disponível em: <http://projects.spring.io/spring-security/>. Acessado em: 11 de maio de 2016.

Luckow Heinzelmann Décio e Melo Altair. Alexandre. (2010) “PROGRAMAÇÃO JAVA PARA A WEB”, primeira edição, editora novatec.

Welcome to Apache Maven. Disponível em: <https://maven.apache.org/>. Acessado em: 11 de maio de 2016.

Conceito básicos sobre Metodologias Ágeis para Desenvolvimento de Software (Metodologias Clássicas x Extreme Programming). Disponível em: <http://www.devmedia.com.br/conceitos-basicos-sobre-metodologias-ageis-para-desenvolvimento-de-software-metodologias-classicas-x-extreme-programming/10596>. Acessado em: 11 de maio de 2016.

Bauer C. e King G. (2007) “JAVA PERSISTENCE COM HIBERNATE”, primeira edição, editora Ciência Moderna.

Nascimento Lima N. (2009) “PERSISTÊNCIA EM BANCO DE DADOS: UM ESTUDO PRÁTICO SOBRE AS API JPA E JDO”.

Manhães Teles Vinícius, MELHORIA. Disponível em: <http://www.desenvolvimentoagil.com.br/xp/principios/melhoria>. Acessado em: 12 de outubro de 2016.

Manhães Teles Vinícius, PASSOS DE BEBÊ. Disponível em: http://www.desenvolvimentoagil.com.br/xp/principios/passos_bebe. Acessado em: 12 de outubro de 2016.

Princípios por trás do manifesto ágil. Disponível em: <http://www.manifestoagil.com.br/principios.html>.