

# Preparação dos Acadêmicos dos Cursos de Computação da UniEVANGÉLICA para o Mercado de Trabalho através de uma Fábrica de Software Acadêmica

Fabiana Cristina de Sousa<sup>1</sup>, Leonardo Duarte Amorim<sup>2</sup>, Kleber Silvestre Diogo<sup>3</sup>,  
Walquiria Fernandes Marins<sup>4</sup>

<sup>1-4</sup>Bacharelado em Engenharia de Computação - Centro Universitário de Anápolis  
(UniEVANGÉLICA) - Anápolis - GO

{<sup>1</sup>fabi\_sousa1995, <sup>2</sup>leonardoduarteamorim}@hotmail.com, {<sup>3</sup>kleber.diogo,  
<sup>4</sup>wallmarins}@gmail.com

**Resumo.** Este artigo aborda aspectos da preparação dos acadêmicos dos cursos de Bacharelados em Computação do Centro Universitário de Anápolis - UniEVANGÉLICA para o mercado de trabalho através de sua inserção em projetos reais em uma Fábrica de Software Acadêmica (FSA). O objetivo é apresentar dados sobre projetos, alocação de egressos para o mercado de trabalho, exibir a metodologia e processo de trabalho utilizados na Fábrica de Tecnologias Turing (FTT) e discutir o impacto dessa abordagem no perfil profissional e técnico dos acadêmicos que ingressam na fábrica. Com isso, é evidenciada a importância das Fábricas de Software Acadêmicas na formação do perfil profissional dos acadêmicos nos cursos Bacharelados em Computação bem como potenciais melhorias para esse processo.

**Palavras-Chave:** Computação; Fábrica de Software; Mercado de Trabalho.

## 1. Introdução

Na conferência *Gartner* em outubro de 2017 foi estimado que haverá um aumento mundial, em 2018, de 4,3 % de gastos com TI em relação ao ano de 2017, chegando a um valor total de U\$\$ 3,7 trilhões em 2018 (SOFTEX, 2017).

Embora as previsões para 2018 sejam positivas é previsto que só na América Latina haverá um déficit de 449 mil profissionais em 2019. Em 2015 o maior *déficit* de profissionais, capacitados e empregados, foram brasileiros com uma falta de 195 mil, é previsto que este número reduza para 161 mil até 2019 (COMPUTERWORLD, 2016).

O Brasil passou por um período de mudança nas diretrizes dos seus cursos graduação em computação, visando proporcionar aos acadêmicos uma melhor preparação para o mercado de trabalho. Preocupada com o perfil profissional de seus acadêmicos de computação a UniEVANGÉLICA disponibiliza uma FSA denominada FTT.

A FTT é uma FSA disponível para os acadêmicos dos cursos de computação da UniEVANGÉLICA, com a utilização de técnica, ferramentas, processos e metodologias atuais no mercado, proporciona uma vivência de trabalho em projetos de desenvolvimento e incentivo a pesquisa e inovação. Dedicar-se a projetos reais de desenvolvimento de sistemas internos e externos da instituição. Os membros da FTT apresentam um diferencial em seu perfil profissional, com um grande índice de egressos para o mercado de trabalho. A FTT dispõe aos acadêmicos, dos cursos de computação da UniEVANGÉLICA, de seu laboratório para projetos de pesquisa.

## 2. Fundamentação Teórica

### 2.1. *Cursos de Bacharelados em Computação no Brasil*

Em conformidade com o documento de Referenciais de Formação para os Cursos de Graduação em Computação 2017 da Sociedade Brasileira de Computação (SBC), durante anos abordaram em diversos eventos da SBC sobre as diretrizes curriculares dos cursos de graduação em computação do Brasil, sendo propostos em 2015 que fosse elaborado documentos que contribuíssem na elaboração de projeto pedagógico dos cursos de graduação até 2017 (ZORZO *et al.*, 2018).

No dia 16 de novembro de 2016, o Ministério da Educação (MEC), o Conselho Nacional de Educação (CNE) e a Câmara de Educação Superior (CES) anunciam a Resolução de nº 5, instituindo Diretrizes Curriculares nas áreas de graduação em Computação, incluindo os cursos de licenciatura em Computação e os bacharelados em Ciência da Computação, Sistemas de Informação, Engenharia de Computação e Engenharia de Software, a serem constadas pelas Instituições de Educação Superior do País. Os cursos de graduação na área de Computação precisam conter conteúdo partilhado e distinto referentes as áreas de Computação em cada curso. Institui que as Atividades Complementares devem desenvolver habilidades, conhecimentos, competências e atitudes do aluno de forma a enriquecer e implementar o perfil do formando. Esta Resolução teve como decreto ser implantada nas Instituições de Educação Superior em até 2 (dois) anos, a partir da data em que foi vigorada (CURI, 2016).

### 2.2. *Metodologias Ágeis*

Os *softwares* eram desenvolvidos, em meados da década de 1980 até 1990, por grandes equipes geograficamente dispersas e os projetos ainda duravam longos períodos. No entanto para abordar projetos complexos, porém menores esses métodos não eram rentáveis, então surgiu a necessidade de criar novas metodologias ágeis, as quais têm uma abordagem incremental e aceita diversas mudanças durante o desenvolvimento. Ademais, essas novas estratégias de desenvolvimento focam no *software* e não em sua parte burocrática, embora todas as metodologias reconheçam sua importância (SOMMERVILLE, 2011).

#### 2.2.1. *Scrum*

*Scrum* é um *framework* que possibilita ao usuário tratar de problemas e também entregar ao cliente o mais alto valor de produto (IMPROISSI, 2017).

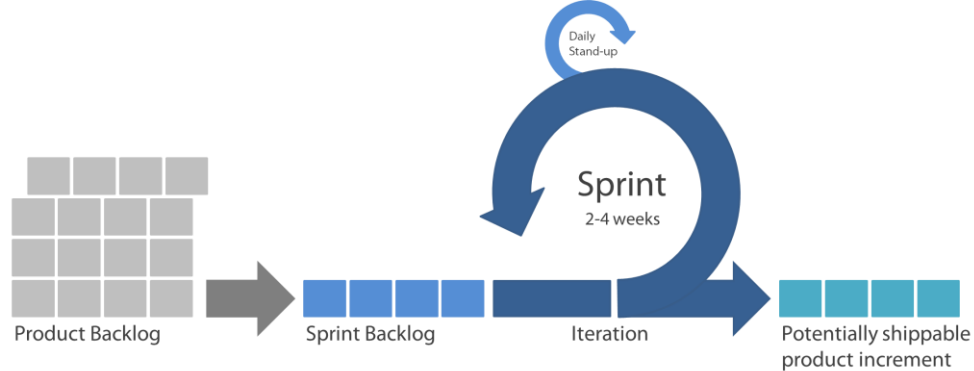
Utilizado a partir de 1990, tem seus valores fundamentados no empirismo, o seu conhecimento é adquirido através da experiência e tomadas de decisão. O *Scrum* é baseado em três pilares: transparência, inspeção e adaptação. E com quatro eventos determinados pelo *framework*: reunião de planejamento, reunião diária, reunião de revisão e reunião de retrospectiva, que visam amparar e garantir que os pilares e o empirismo sejam seguidos (SCHWABER; SUTHERLAND, 2013).

O Time Scrum é composto por Time de Desenvolvimento, P.O (*Product Owner*) e S.M (*Scrum Master*). A equipe deve ser independente, organizável e multifuncional. O Time de Desenvolvimento deve produzir o incremento “pronto”, entregue no final de cada *Sprint*. O *Product Owner* amplia o valor do produto que será desenvolvido, deve definir a priorização do *backlog*, representa o cliente dentro da organização. E o *Scrum Master* deve garantir que todos dentro do Time *Scrum* entendam e sigam o processo de desenvolvimento *Scrum*, resolve os impedimentos e conduz as reuniões (SCHWABER; SUTHERLAND, 2013).

O desenvolvimento *Scrum* é dividido por *sprints* com *time box* de 3 a 4 semanas. Durante a cada *sprint* acontecem os eventos com as: i) reunião de planejamento ao início de cada *sprint*, que é

definido os requisitos do *backlog* que serão entregues ao final do *time box*, ii) as reuniões diárias que acontecem todos os dias e os membros espoem as atividades que realizarão no dia anterior e quais irão realizar no dia, é relatado também os impedimentos se necessário, iii) as reuniões de revisão e retrospectiva acontecem ao final da *sprint*, na revisão é apresentado o produto “pronto” e o *Product Owner* avalia se está de acordo com o solicitado e na retrospectiva, é avaliado o desempenho da *sprint* e exposto por todos quais foram os pontos bons, o que precisa ser melhorado e quais as lições aprendidas (SCHWABER; SUTHERLAND, 2013).

Figura 1. Processo do *Framework Scrum*.

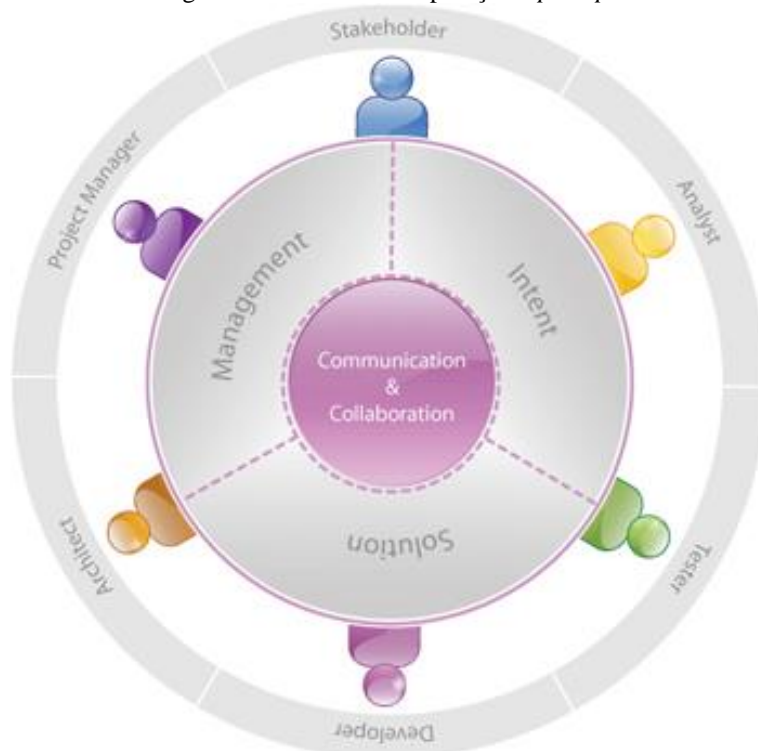


Fonte: <https://www.clarityupgrades.be/en/methodology/scrum/>

### 2.2.2. *OpenUP (Open Unified Process)*

Diferentemente das metodologias ágeis das décadas de 80 e 90 o *OpenUP* “É um processo de desenvolvimento de software de código aberto projetado para equipes pequenas eco-localizadas que querem ter uma abordagem ágil para desenvolvimento.” (Eclipse, 2017).

Figura 2. Modelo de Cooperação *OpenUp*.



Fonte: [https://www.cin.ufpe.br/~rls2/processo\\_tg/openup/rolesets/openup\\_roles\\_5CDDEEDA.html](https://www.cin.ufpe.br/~rls2/processo_tg/openup/rolesets/openup_roles_5CDDEEDA.html)

Considerado uma filosofia de desenvolvimento ágil mínima, pois utiliza o mínimo de recursos necessários para pequenas equipes, completo, pois pode ser aplicado sem nenhuma alteração e extensível, pois pode ser personalizado de acordo com a necessidade do projeto. Possui quatro alicerces que comunicam entre si: i) Colaboração: perfilar interesses e compartilhar fundamentos. ii) Balanceamento: definir prioridades para aumentar o valor dos interessados. iii) Foco: na arquitetura para diminuir riscos, possibilitar melhor colaboração técnica, minimizar o sucateamento e o retrabalho. iv) Evolução: evoluir para obter *feedback* e resultados positivos do cliente (Eclipse, 2017a).

“A Fundação *Eclipse* distribui, como parte da plataforma EPF<sup>1</sup>, o processo *OpenUP/Basic*<sup>2</sup>, o mais básico dos membros da família de processos muito leves, mas ainda com muitas características similares às do RUP<sup>3</sup>.” (PAULA FILHO, 2013), destaca: i) desenvolver *softwares* iterativamente. ii) gerenciar requisitos. iii) usar arquiteturas baseadas em componentes. iv) modelar *software* visualmente. v) verificar a qualidade do *software*. vi) controle de mudanças (SOMMERVILLE, 2011).

### 2.3. *Fábrica de Software Acadêmica*

No final dos anos 1960 foi utilizado pela primeira vez o termo FS (Fábrica de *software*) por RW Bremer de General Electric e em 1968 MD McIlroy da AT & T (Lenz; Wienands, 2006). “As fábricas de softwares usam conhecimento de domínio específico, arquiteturas de solução, ferramentas e outros ativos reutilizáveis para ajudar os usuários a produzir tipos específicos de soluções de softwares.” (Microsoft, 2018). Entre os objetivos da FS lista-se: i) Incorporação de novas arquiteturas e métodos de desenvolvimento. ii) Acrescer competências em maioria dos componentes acadêmicos, os quais serão aplicados de acordo com os métodos e processos de software em um laboratório. iii) Estimular docentes e discentes em várias atividades importantes para o mercado de trabalho (Oliveira; Colenci Neto, 2003).

A abordagem exata do conceito fábrica associa a concepções como “Fordismo”, “Toyotismo” e “Taylorismo”. Embora conceitos diferentes tenham como base o trabalho repetitivo, tratando as pessoas como se fossem máquinas, porém não é essa abordagem visada para as fábricas de software, uma prova disso é que as FSA não têm pautado como características principais: i) Redução de tempo de desenvolvimento. ii) Lucro de negócio. Apesar de serem necessidades do mercado o foco das FSA é o aprendizado a preparação para o mercado de trabalho (POVICI, 2006). Com isso, durante o período de graduação, é estimulado a interdisciplinaridade e simulado um ambiente empresarial, diferenciais que são alcançados com a liberdade de busca de novas metodologias para a realização de projetos de software. Além de provocar o corpo docente a estar sempre atualizado e capacitado, por causa das constantes mudanças no ambiente criado da FSA (Oliveira; Colenci Neto, 2003).

## 3. **Abordagem e Proposta**

### 3.1. *Cursos de Computação na UniEVANGÉLICA*

Os cursos de Computação da UniEVANGÉLICA, possuem ênfase em desenvolvimento de *software*, com o objetivo de formar profissionais qualificados a atuarem no mercado de trabalho como: desenvolvedor de *software*, analista de requisitos, analista de negócios, arquiteto de *softwares*, e demais segmentos da área (UniEVANGÉLICA, 2017a).

<sup>1</sup> *Eclipse Process Framework* (Eclipse, 2018).

<sup>2</sup> OpenUp/Basic is the same thing of OpenUp (Sente, 2010).

<sup>3</sup> *Rational Unified Process* (Sente, 2011).

Com foco em contribuir para a formação de profissionais aptos ao mercado de trabalho da indústria de *software* e preencher o déficit na formação tecnológica. As diretrizes dos cursos contam com a obrigatoriedade do TCC, o qual é desenvolvido pelo acadêmico com orientações e acompanhamento docente e também dispõem de Estágio Curricular obrigatório, a fim de proporcionar ao acadêmico a experiências de convivência no ambiente profissional e contribuir no exercício da profissão (UniEVANGÉLICA, 2017b; 2017c).

Como parte integrante das políticas institucionais do Centro Universitário de Anápolis - UniEVANGÉLICA os acadêmicos são incentivados a participar de programas de pesquisa e inovação. E os cursos de bacharelado em computação possuem um programa que aborda ações como seminários, minicursos, cursos de qualificação, palestras e demais eventos do segmento (UniEVANGÉLICA, 2017d; 2017e).

Os cursos de bacharelado em Computação da UniEVANGÉLICA contêm um diferencial significativo na formação profissional de seus acadêmicos, disponibilizam de uma Fábrica de *Software Acadêmica* nomeada de Fábrica de Tecnologias Turing (FTT) (UniEVANGÉLICA, 2017f).

### 3.2. Fábrica de Tecnologias Turing

Fundada em 2006, a FTT é um laboratório para o desenvolvimento de software vinculada aos Cursos de Bacharelado em Computação da UniEVANGÉLICA, que oferece aos graduandos uma experiência em FS (UniEVANGÉLICA, 2017f).

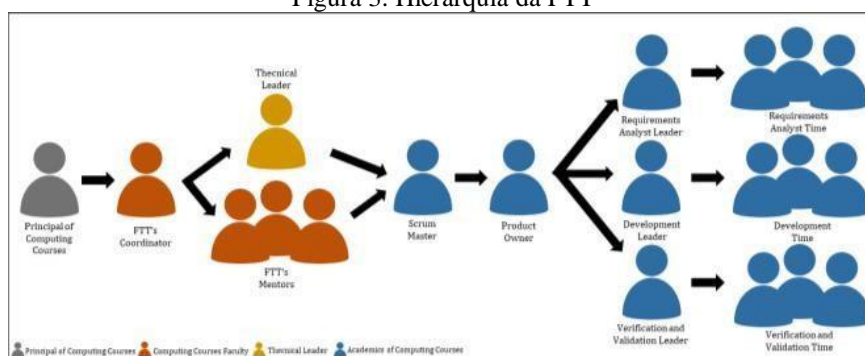
Os alunos membros da FTT são constantemente incentivados a se aperfeiçoarem em novas técnica, ferramentas, processos e metodologias, ganhando destaque em recrutamentos de profissionais por empresas externas. Com uma rotatividade alta da equipe, há a necessidade de diversas seleções e treinamentos para novos membros, dando a oportunidade para mais alunos fazerem parte da FTT e terem uma vivência real antes do mercado de trabalho.

Além dos membros a FTT tem o apoio da coordenação dos cursos de computação, responsáveis também pela coordenação da FTT, dispõe de um diretor designado a administrar e supervisão do desenvolvimento dos projetos, conta com o líder técnico encarregado por manter o ambiente em funcionamento e auxilia no desenvolvimento mantendo a qualidade do projeto durante a integração. A um conjunto de professores orientadores encarregado por instruir os membros de cada equipe.

#### 3.2.1. Os Processos e Metodologias da FTT

A FTT é um ambiente de constantes melhorias, e visa por utilizar técnicas, ferramentas, processos e metodologia atuais no desenvolvimento de seus projetos como o uso de metodologias ágeis nos *frameworks Scrum e OpenUp*.

Figura 3. Hierarquia da FTT



Fonte: Os Autores.



O time de desenvolvimento na FTT é dividido em três equipes: Análise de Requisitos, Desenvolvimento e Verificação e Validação, que são responsáveis por entregar os incrementos do produto “pronto” a cada *Sprint*.

A equipe de Análise de Requisitos é encarregada por documentar o projeto e em termos gerais especifica o que será desenvolvido, por isto suas atividades são realizadas na *Sprint* anterior a equipe de Desenvolvimento. São produzidas as histórias de usuário como forma de documentação da FTT, estas possuem seus cenários e seus critérios de aceitação definidos e foram adaptadas para inclusão do protótipo e definição de campos e comandos. As histórias de usuário são especificadas na modelagem de processo que utiliza a notação *Business Process Model and Notation* (BPMN). Na ferramenta *BizAgi* a notação BPMN é especificada para apresentar a relação dos requisitos e o fluxo de atividades do projeto.

A equipe de Desenvolvimento, realiza a implementação do projeto, baseada nas documentações produzidas pela equipe de Análise Requisitos. É elaborado o banco de dados após a validação da documentação do requisito, a modelagem acontece na *Sprint* anterior, aplica-se a linguagem SQL e utiliza a ferramenta *PostgreSQL*. Na implementação conta com *spring boot* para facilitar a configuração e publicação da aplicação, emprega a linguagem *Java* com *framework Angular 4.X* e *TypeScript*, adotam as ferramentas de implementação *Visual Studio Code* no *front-end* e *IntelliJ IDEA* no *back-end*. No desenvolvimento de cada requisito é implementado primeiramente o teste unitário - TDD e depois produzido a codificação, este então passa pelo teste de unidade, se válido é encaminhado para a equipe de Verificação e Validação, caso não valide o teste unitário e a codificação são refatorados.

A equipe de Verificação e Validação, é responsável por testar o que foi elaborado pelas equipes de Análise de Requisitos e Desenvolvimento. É executado o teste de inspeção nos documentos produzidos pela equipe de Análise de Requisitos, com o uso de um *checklist* para verificação, as inconsistências são relatadas na ferramenta *Mantis*, a seguir é encaminhado para a equipe de Análise de Requisitos que fará as correções. Quando realizado as correções retornam e é executado o reteste de inspeção, ao ser validado, é criado os casos de teste no *TestLink*, que serão utilizados no teste ponta a ponta - e2e, se necessário o ciclo de inspeção se repete. Nos requisitos implementados pela equipe de desenvolvimento é realizado o teste de ponta a ponta, com a aplicação dos casos de teste que foram criados anteriormente, são relatados na ferramenta *Mantis* os erros, falhas e defeitos encontrados e encaminhados para equipe de Desenvolvimento que efetuam as correções e retornam para que seja realizado o reteste de ponta a ponta, em caso de não conformidade o processo se repete, até aprovação. Em casos que são identificadas necessidades de mudança no requisito é encaminhado a equipe de Análise de Requisitos e verificado a viabilidade em conjunto com o P.O., em caso de aceitação é refatorado os documentos, implementações e testes. Ao finalizar o requisito com todos os testes aceitos, é efetuado o teste automatizado, com o uso da ferramenta *Selenium*.

A ferramenta *GitLab* tem a função de controlar as atividades e comunicação entre o time *Scrum*, também realizado o gerenciamento do repositório, representando o *Kanban online*. Além das ferramentas há comunicação direta entre todas as equipes e exerce características cooperativas e comunicativas do *OpenUp*.

Após as etapas realizadas pelo time de desenvolvimento o Líder Técnico é responsável por validar a qualidade do código, se necessário a codificação é refatorada pela equipe de Desenvolvimento, o incremento então é integrado as demais funcionalidades do sistema no *GitLab*, em casos de conflito na integração é identificado a origem e realiza busca por solução.

O P.O. representa o cliente dentro da FTT, é responsável por iniciar o projeto pela definição da visão em que é produzido o escopo e documento de visão do produto. O *backlog* do produto é elaborado e logo o prioriza para o desenvolvimento de acordo com as prioridades do cliente. Sua

missão é fazer com que a time entenda e desenvolva o solicitado por ele. Deve garantir que o *backlog* do produto esteja sempre atualizado e priorizado.

O S.M. é encarregado por garantir que os membros conheçam e sigam os processos e metodologias utilizadas na FTT. É responsável por conduzir as reuniões de planejamento, diárias, de revisão e retrospectiva. Deve garantir o bom andamento do projeto, remover os impedimentos, gerir de riscos e acompanhar as atividades.

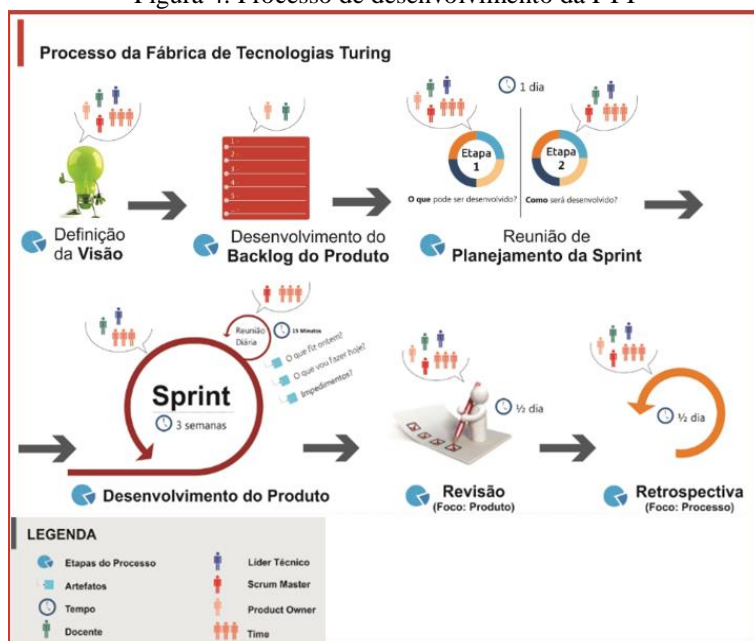
Na FTT o S.M. também realiza atividades de recursos humanos efetuando o recrutamento de novos membros, através de processos seletivos, os quais são lançados por meio de editais e divulgado aos alunos dos cursos de computação, os interessados são avaliados por um conjunto de atividades de caráter classificatório definidas pelos professores orientadores. Os classificados são apresentados ao atual projeto através de *workshop* conduzido pelo S.M., em sequência passam por um processo de capacitação, na área que atuará, junto aos membros da FTT. Além disso o S.M. assegura o bom comportamento dos membros.

A reunião de planejamento acontece antes de cada *Sprint* é conduzida pelo S.M. com a participação do P.O. e do time. O P.O. apresenta as funcionalidades do *backlog* priorizado e o processo de estimativa de esforço de tempo é executado pelo S.M., com a adoção da técnica PERT (*Program Evaluation and Review Technique*) para estimar os requisitos que serão documentados e as técnicas *Planning Poker* e *Ideal Days* para estimativa dos requisitos a serem implementados. Com o tempo definido para execução das atividades determina os requisitos do *backlog* de *Sprint* para o time de desenvolvimento.

Durante as *Sprint* o que foi definido na reunião de planejamento é executado e nas reuniões diárias é acompanhado pelo S.M. e o P.O. os quais os membros relatam as atividades que em progresso e, caso haja, os impedimentos.

A reunião de revisão e retrospectiva acontecem após o *time box* da *Sprint*, no qual é tratado durante a revisão o que foi produzido e é validado pelo P.O. se atende as necessidades solicitadas durante o planejamento e na retrospectiva são abordados os aspectos que foram satisfatórios e os que precisam ser aprimorados na realização da *Sprint*. O processo da FTT segue descrito na Figura 4.

Figura 4. Processo de desenvolvimento da FTT



Fonte: Os Autores.

## 4. Resultados Alcançados

Os processos e metodologias apresentados são resultados de estudos e melhorias adquiridos ao longo de 12 anos da Fábrica de Tecnologias Turing. Apresenta resultados em seu progresso, projetos desenvolvidos, egressos para o mercado de trabalho e como base para projetos de pesquisas.

### 4.1. Progresso e Projetos da FTT

Com 40 membros ativos, entre alunos e professores dos cursos de Computação da UniEVANGÉLICA a FTT está bem distante de seu contexto inicial, em que possuía 3 membros e ocupava um pequeno espaço junto a secretária do curso com disponibilidades de 2 computadores. Devido ao andamento dos projetos durante o tempo a FTT conquistou seu espaço e credibilidade na UniEVANGÉLICA. Aos poucos foi possível proporcionar a maior quantidade de alunos a vivência de uma fábrica de software com trabalho em projetos reais de desenvolvimento. No gráfico a seguir é exposto a quantidade de acadêmicos que estiveram na FTT por ano até 2017.

Gráfico 1. Quantidade de Membros por Ano (2006-2017).



Fonte: Os Autores.

Em 2018 a FTT possuía um laboratório próprio com 30 computadores, o seu funcionamento era matutino e vespertino, continha um grupo de acadêmicos em cada turno, dispunha uma equipe de professores destinados à orientação, desfrutava do auxílio de seis estágios e 11 bolsas de estudos ofertados pela UniEVANGÉLICA, que são distribuídas aos membros e disponibilizava vagas para trabalho voluntário.

Quadro 1. Projetos Desenvolvidos na FTT.

PROJETO	INÍCIO	TÉRMINO
UniFisio	ago/06	dez/14
SISPESQ	fev/09	jun/16
Portinari	fev/10	dez/10
SISLAC	fev/11	jun/15
SigTuring	fev/12	dez/12
Sefaa	fev/12	dez/12
GLI	fev/12	jun/13
Biblioteca Digital	fev/12	dez/13
PronPortal	fev/13	jun/16
UniMaps	fev/14	jun/16
Sites	fev/15	dez/16
CPS	fev/15	jun/15
GPE	fev/16	set/16
VirToo	set/16	Em andamento
SEintegra	ago/18	Em andamento

Fonte: Os Autores.



Um dos principais índices de desempenho no quesito fábrica de software, é expresso pelos projetos desenvolvidos. A FTT apresenta uma série de projetos que foram trabalhados durante sua existência (Quadro 1).

Foram desenvolvidos projetos para controles internos como o SisLac que é um sistema para gerenciar o Laboratório de Análises Clínicas do curso de farmácia da UniEVANGÉLICA. Projetos externos como o Portinari<sup>4</sup>, que faz o controle do acervo do Candido Portinari na PUC.RJ. Atualmente está desenvolvendo o projeto internacional VirToo que é um Sistema Acadêmico e Financeiro destinado as faculdades ISTEEL (Instituto Superior de Teologia Evangélica no Lubando) e ISPEL (Instituto Superior Politécnico Evangélica no Lubando) de Lubando na Angola.

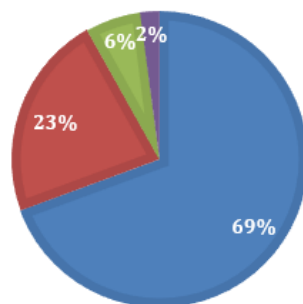
#### 4.2. *Egressos para o Mercado de Trabalho*

A FTT é um diferencial disponível para a formação dos acadêmicos de computação na UniEVANGÉLICA, como um ambiente em que visa preparar os alunos para o mercado de trabalho, a sua rotatividade é grande e constante.

Dos 49 alunos que participaram da FTT no ano de 2017: i) Com 11 contratados para o mercado de trabalho por fábricas de softwares, atuantes como Analistas, Desenvolvedores e Testadores; ii) Permanecendo 34 na FTT no desenvolvimento do projeto VirToo; iii) Foram 3 desistências durante o ano; iv) Houve um desligamento por não obter o perfil adequado ao ambiente. Como demonstrado no Gráfico 2.

Gráfico 1. Rotatividade em 2017.

■ Active ■ Egress ■ Withdrawal ■ Shut off



Fonte: Os Autores.

#### 4.3. *Projetos de Pesquisa*

A FTT contribui como laboratório de pesquisa para projetos de internos e externos, como os TCCs dos acadêmicos de computação da UniEVANGÉLICA.

Os TCCs embasados na FTT, realizam estudos das técnicas, ferramentas, processos e métodos utilizados, para abordagem de proposta de melhorias a serem aplicadas no desenvolvimento de projetos. Contribuindo de forma significativa para a crescente evolução do processo da FTT.

No último ano os TCCs contribuíram para as diversas áreas existentes na FTT, mas especificadamente na equipe de Verificação e Validação com a melhoria do processo de inspeção de requisitos, análise das vulnerabilidades dos sistemas desenvolvidos, implantação da matriz de gestão de riscos, mudança para a modelagem de processos na especificação de requisitos.

<sup>4</sup> Disponível em [www.portinari.org.br](http://www.portinari.org.br).

Estão em processos de pesquisa os TCCs, como por exemplo a intervenção de melhorias na aplicação no Scrum na FTT e adoção de uma técnica de estimativa de esforço de tempo adequada ao desenvolvimento produzido.

## 5. Considerações finais

Com o atual déficit de profissionais de computação capacitados no mercado de trabalho Brasileiro, a disponibilidade da FSA, para os acadêmicos de computação, é fundamental na formação ao longo da graduação, lhes proporciona um grande diferencial na inserção do mercado de trabalho. Por atuarem em projetos reais de desenvolvimento, cumprindo prazos e metas determinado pelas estimativas realizadas em seus processos e métodos.

A FTT prepara os seus membros com constantes treinamentos, workshops e uso das principais ferramentas, técnicas, processos e metodologias presentes no mercado de trabalho brasileiro. E obtêm grande destaque nos processos de recrutamento em fábricas de software.

## Agradecimentos

Este trabalho foi realizado no ambiente do Centro Universitário de Anápolis - UniEVANGÉLICA, em Anápolis, Goiás, Brasil. Com o apoio da direção dos cursos de informática da instituição, disponibilizou-se a Fábrica de Tecnologias de Turing (FTT), para estudo de caso e pesquisa. Agradecemos à Diretora Ma. Viviane Carla Batista Povicci pelo incentivo, nossos assessores por propor o desenvolvimento deste artigo e dar-nos todo o apoio necessário e os membros da FTT por dispor de seu tempo para nos fornecer informações.

## Referências

- COMPUTERWORLD. 2016. *Faltarão 449 mil profissionais de TI na América Latina até 2019*. Retrieved January 17, 2018. From <http://computerworld.com.br/falta-de-profissionais-de-ti-na-america-latina-chegara-32-ate-2019>.
- Curi, Luiz Roberto Liza. 2016. *Resolução nº 5, de 16 de novembro de 2016 (\*)*. Retrieved January 15, 2018 from [http://portal.mec.gov.br/index.php?option=com\\_docman&view=download&alias=52101-rces005-16-pdf&category\\_slug=novembro-2016-pdf&Itemid=30192](http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=52101-rces005-16-pdf&category_slug=novembro-2016-pdf&Itemid=30192)
- Eclipse. 2017a. *Introdução ao OpenUp/Basic OpenUp*. Retrieved January 2, 2018 from <http://epf.eclipse.org/wikis/openuppt/index.htm>
- Eclipse. 2017b. *Introduction to OpenUp*. Retrieved January 2, 2018 from <http://epf.eclipse.org/wikis/openup/index.htm>
- Eclipse. 2018. *Eclipse Process Framework Project (EPF)*. Retrieved January 3, 2018 from <https://www.eclipse.org/epf/>
- Lenz, Gunther; Wienands, Christoph. 2006. *Practical Software Factories in .NET*. Copyright.
- Microsoft. 2018. *Projete e Implemente uma Fábrica de Software*. Retrieved January 11, 2018 from <https://msdn.microsoft.com/library/bb245657.aspx>
- Ministério da Educação. 2016. *Conselho Nacional de Educação - CNE*. Retrieved January 15, 2018 from [http://portal.mec.gov.br/index.php?option=com\\_content&view=article&id=12449&Itemid=753](http://portal.mec.gov.br/index.php?option=com_content&view=article&id=12449&Itemid=753).
- Ministério da Educação. 2016. *História*. Retrieved January 14, 2018 from [http://portal.mec.gov.br/index.php?option=com\\_content&view=article&id=2](http://portal.mec.gov.br/index.php?option=com_content&view=article&id=2).

- Oliveira, Dlogas Hamilton de. Colenci Neto, Alfredo. 2003. *Fábrica de Software: Promovendo a Criação de Empresas Competitivas em Tecnologias de Informação*. Retrieved January 05, 2018 from [http://abepro.org.br/biblioteca/enegep2003\\_tr0707\\_1257.pdf](http://abepro.org.br/biblioteca/enegep2003_tr0707_1257.pdf)
- Paula Filho, Wilson de Pádua. 2013. *Engenharia de Software: Fundamentos, Métodos e Padrões* (3ªed). Rio de Janeiro: LTC.
- Pocivi, Viviane Carla Batista. 2011. *Um Estudo para Melhoria de Ensino e Aprendizagem em Engenharia de Software em Cursos de Graduação*. Mester'sthesis. C.E.S.A.R - Centro de Estudo e Sistemas Avançados de Recife, Recife, Brasil.
- SBC. 2017. *Sobre a SBC*. Retrieved January 18, 2018 from <http://www.sbc.org.br/institucional-3/sobre>
- Schwaber, Ken; Sutherland, Jeff. 2013. *Guia do Scrum*. Retrieved January 13, 2018 from <https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>
- Sente, Rafael Pereira. 2010. *OpenUp uma Visão Geral*. Retrieved January 3, 2018 from <https://www.tiespecialistas.com.br/2010/09/openup-uma-visao-geral/>
- Sente, Rafael Pereira. 2011. *RUP - Primeiros Passos*. Retrieved January 3, 2018 from <https://www.tiespecialistas.com.br/2011/02/rup-primeiros-passos/>
- SOFTEX. 2017. *Gartner afirma que despesas globais com TI atingirão US\$ 3,7 trilhões em 2018*. Retrieved January 17, 2018 from <https://www.softex.br/gartner-afirma-que-despesas-globais-com-ti/>.
- Sommerville, Ian. 2011. *Engenharia de Software* (9ªed). São Paulo: Pearson Prentice
- UniEVANGÉLICA. 2017a. *Engenharia de Computação*. Retrieved January 12, 2018 from <http://www.unievangelica.edu.br/curso.engenharia-de-computacao/>.
- UniEVANGÉLICA. 2017b. *Engenharia de Computação - Trabalho de Conclusão*. Retrieved January 12, 2018 from [http://www.unievangelica.edu.br/curso.engenharia-de-computacao/trabalho\\_de\\_conclusao\\_curso/](http://www.unievangelica.edu.br/curso.engenharia-de-computacao/trabalho_de_conclusao_curso/)
- UniEVANGÉLICA. 2017c. *Engenharia de Computação - Estágio*. Retrieved January 12, 2018 from <http://www.unievangelica.edu.br/curso.engenharia-de-computacao/estagio/>
- UniEVANGÉLICA. 2017d. *Engenharia de Computação - Atividades de Extensão*. Retrieved January 12, 2018 from [http://www.unievangelica.edu.br/curso.engenharia-de-computacao/atividades\\_de\\_extensao/](http://www.unievangelica.edu.br/curso.engenharia-de-computacao/atividades_de_extensao/)
- UniEVANGÉLICA. 2017e. *Engenharia de Computação - Pesquisa*. Retrieved January 12, 2018 from <http://www.unievangelica.edu.br/curso.engenharia-de-computacao/pesquisa/>
- UniEVANGÉLICA. 2017f. *Engenharia de Computação - Diferenciais*. Retrieved January 12, 2018 from <http://www.unievangelica.edu.br/curso.engenharia-de-computacao/diferenciais/>
- Zorzo, Avelino F.; Nunes, Daltro; Matos, Ecivaldo S.; Steinmacher, Igor; Leite, Jair C.; Araujo, Renata; Correia, Ronaldo C. M.; Martins, Simone. 2018. *Referenciais de Formação para os Cursos de Graduação em Computação*. Retrieved January 16, 2018 from <http://www.sbc.org.br/documentos-da-sbc/send/127-educacao/1155-referenciais-de-formacao-para-cursos-de-graduacao-em-computacao-outubro-2018>