

Wisegress: Desenvolvimento de um aplicativo de produtividade para mitigar os efeitos da produtividade tóxica

Gusthawo Junkes¹, Nathan S. Antunes²

¹Universidade Católica de Santa Catarina
Jaraguá do Sul – SC – Brasil

{gusthawo.junkes,nathan.antunes}@catolicasc.edu.br

Abstract. *One of the main challenges faced during the professional and academic development of human beings comes from the difficulty of coordinating the workload imposed today. The frequent increase in this workload itself ends up generating excessive stress and can also cause wear and tear on an individual's health. The present project proposes, through the use of technology and motivational techniques, the development of software aimed at combating excessive self-productivity, commonly referred to as "toxic productivity."*

Resumo. *Um dos principais desafios enfrentados durante a construção profissional e acadêmica do ser humano vem da dificuldade de coordenar a carga de trabalho imposta atualmente. O aumento frequente dessa carga de trabalho por si só acaba gerando excesso de estresse e também pode causar um desgaste na saúde de um indivíduo. O presente projeto propõe, através do uso da tecnologia e de técnicas motivacionais, o desenvolvimento de um software que visa combater a produtividade excessiva contra si mesmo, que é popularmente denominado de "produtividade tóxica".*

1. Introdução

Existem duas perspectivas acerca do termo produtividade, [Allen 2015] determina: "O termo produtividade, infelizmente, costuma ter duas conotações: realização (o ato de empreender) e ocupação (o ato de preencher o tempo)". Essas duas perspectivas juntas costumam se encaixar na forma que o mercado enxerga a produtividade no trabalhador, ou seja, realizando e/ou produzindo uma grande carga de trabalho no menor tempo possível e criando pressão para a entrega, dessa forma, cria-se uma falsa crença que o valor de um indivíduo está diretamente ligado à produção [Kilaru 2022]. Profissionais que atuam diariamente sob pressão e constante responsabilidade tendem a sofrer com alguns sintomas do cansaço extremo, que pode levar até o esgotamento ou *burnout* [PEREIRA 2021]. Este esgotamento nem sempre é visível, tanto que "A violência da positividade não é privativa, mas saturante; não excludente, mas exaustiva. Por isso é inacessível a uma percepção direta.", conforme comenta [Han 2015].

No aspecto pessoal, a produtividade tóxica é criada através de um ciclo interminável de tarefas, da necessidade de provar seu valor e receber aprovação de terceiros na tomada de decisão. Quando se pratica uma produtividade tóxica, o indivíduo não se preocupa sobre o motivo da realização de determinada tarefa, apenas na execução. Ao entrar nesse ciclo o indivíduo passa a olhar a sua própria produtividade como apenas um

número a ser somado e não mais como a qualidade da execução de uma determinada tarefa. Esse tipo de atitude acarreta com consequências tanto físicas quanto neurológicas por negligenciar o próprio descanso. [Kilaru 2022]

Uma forma de evitar a produtividade tóxica, é controlando o tempo de trabalho a partir de modelo de produção (por exemplo, o método de *Pomodoro*). Esse controle pode ser feito de forma manual, ou por meio de uma aplicação computacional, que visa mitigar os efeitos da produtividade tóxica e surge da preocupação sobre o bem-estar de quem sofre com o problema. Como objetivo deste trabalho, destaca-se a análise e identificação do modelo de produção individual de cada usuário para oferecer recursos personalizados e promover uma melhor orientação sobre como coordenar a carga de trabalho e/ou estudo praticado. Através de uma coleta de dados de feedback do próprio usuário sobre quais modelos/técnicas melhor se encaixam para cada realidade será possível avaliar os principais impactos na produtividade individual produzida por cada modelo de realização de tarefas, qual tem a maior eficácia e produz a sensação mais atraente de bem-estar. Sendo abordados no presente artigo os tópicos de revisão bibliográfica, trabalhos correlatos, desenvolvimento, resultados, conclusão e trabalhos futuros.

2. Revisão Bibliográfica

Neste tópico será abordado conceitos e fundamentos necessários para o entendimento do tema, sendo ele a produtividade tóxica, junto a tecnologia da informação. Para tal, serão exploradas as referências entorno dos tópicos mencionados, com a identificação da base necessária para o desenvolvimento de uma solução de software que mitigue os efeitos da produtividade tóxica.

2.1. Produtividade tóxica

Visar o aumento da produtividade e do crescimento profissional têm se tornado cada vez mais comum, afinal, o mercado de trabalho também está ficando cada vez mais competitivo. Porém, realizar tarefas de maneira descontrolada e sem pensar no próprio desgaste, tanto emocional como físico, além de outros fatores emocionais, pode trazer à tona alguns sintomas da produtividade tóxica como por exemplo o estresse, cita [Lipp 1998] em: “A sobrecarga de trabalho e na família, o relacionamento com a chefia, a autocobrança, a falta de união e cooperação na equipe, o salário insuficiente, a falta de expectativa de melhoria profissional e também o meio social podem ser causadores de stress”. A autocobrança, é um dos termos que melhor explica a produtividade tóxica por se tratar de ser uma palavra que define como o indivíduo pode ser prejudicial a si próprio.

“A motivação e positividade induzem ao sujeito o sentimento de confiança que passa a ser o explorador de si mesmo, pois quer e “pode” atingir seus objetivos, não há então exploração, em que se obriga a trabalhar, pois o próprio sujeito quer se autorrealizar” [Mateus 2021]. Quando um sujeito se dispõe de múltiplas tarefas a serem feitas o seu psicológico pode não identificar que ele está se sabotando, justamente por acreditar que ao cumprir com seus afazeres ele estará sendo recompensado. Entretanto, esse comportamento do indivíduo se torna um ponto delicado, pois à vista de algumas pessoas ele está produzindo, desempenhando bem e fazendo o seu trabalho como deve ser, mas as custas de seu desgaste.

A produtividade tóxica é atrelada ao indivíduo que se cobra em excesso indiferentemente do motivo pela qual a autocobrança se originou, e, não é exclusiva do

meio empresarial, podendo também surgir no meio acadêmico, assim como comenta [Cristo et al. 2019] em “A autocobrança em relação ao rendimento acadêmico, por sua vez, refere-se à cobrança pessoal para atingir um rendimento acadêmico considerado satisfatório pelo aluno, bem como a crença que ele tem do que a família espera dele”. Sendo assim, é notado que a produtividade tóxica pode vir a surgir de dois principais meios: O empresarial e o acadêmico.

2.1.1. Métodos de produtividade

Uma das formas de melhorar a produtividade e a saúde mental, utilizando técnicas de aprendizado e foco como o método de *Pomodoro* e o método GTD (Getting Things Done). O *Pomodoro* consiste em definir tempos de produção, com intervalos definidos para descanso, propiciando que a pessoa trabalhe de forma focada sem chegar num desgaste [ANDRADE 2022]. Já o método GTD [Allen 2015] tem como objetivo separar a organização de tarefas em diferentes etapas, que iniciam na divisão de cada tarefa em partes menores, priorizando a urgência dos afazeres, proporcionando uma melhor organização. Uma alternativa para atingir uma melhora no processo de trabalho, é ajudar a pessoa a escolher a melhor técnica de concentração/estudo conforme sua necessidade.

2.2. Tecnologia da Informação

Diante desse cenário, podem ocorrer mudanças que alteram totalmente o dia a dia, como é o caso da pandemia de covid-19, que levou as empresas a adotarem medidas para impedir a proliferação do vírus e proteger seus colaboradores. Tais medidas estavam ligadas a aplicação do home office, trabalho realizado a partir da própria residência, possibilitando horários flexíveis e maior segurança durante a pandemia. No entanto, conforme pesquisa realizada por [PEREIRA 2021] com profissionais da Tecnologia da Informação (TI), identificou que estes profissionais tiveram suas rotinas e mentes prejudicadas pelo *burnout* que [Harrison 1999] definiu como “um tipo de estresse de caráter persistente vinculado a situações de trabalho, resultante da constante e repetitiva pressão emocional associada com intenso envolvimento com pessoas, por longos períodos de tempo”. Esse estresse foi acarretado por alterações e horas de trabalho fora do adequado, devido a necessidade da entrega de resultado e de suas implementações. Conforme sua pesquisa, 21% de um total de 72 pessoas tiveram que exercer atividades extras de 6 horas a 10 horas por semana, as quais apresentaram os efeitos do burnout.

Tais problemas psicológicos agravam as entregas produzidas pelos colaboradores de uma empresa ou até mesmo a efetividade do estudo de um acadêmico, pelo fato do esgotamento físico e psicológico diminuírem sua eficiência. Para combater o estresse causado pela superprodução e o desejo da produtividade crescente em cada pessoa, [Allen 2015] determina uma metodologia para auxiliar na organização dos processos, onde atrela o registro das tarefas a serem feitas a organização em partes menores de cada tarefa maior, para determinar quão acionável uma tarefa deve ser, proporcionando a ela uma prioridade de execução que deve ser corriqueiramente refletida. Por fim, a execução das tarefas em um conjunto organizado com prioridade e foco do que realmente deve ser feito e quando deve ser feito.

Organizar a rotina de trabalho ou de estudo é um passo importante para a mitigar os efeitos do stress e de outras doenças relacionadas a produtividade tóxica, como a

depressão. Tal processo varia de pessoa a pessoa, assim como menciona [Allen 2015], onde não existe uma forma única de aperfeiçoar a organização pessoal e a produtividade, existem coisas que você pode fazer para facilitar o processo. Aos quais normalmente são processos isolados ou distintos, que não estão centralizados numa única aplicação e que muito menos se adaptam as diferentes necessidades de cada pessoa. Portanto, a construção de uma solução que atinja tais requerimentos fomentando um dia a dia proveitoso, com a entrega de valor no lugar da quantidade tende a minimizar os efeitos da produtividade tóxica, juntamente com a potencialização dos resultados entregues por cada pessoa.

3. Trabalhos correlatos

Fonte: Gusthawa e Nathan

Característica	Notion	Trello	Todoist	Evernote	Plug-in Outlook	Wisegress
Lista de tarefas	Sim	Sim	Sim	Sim	Sim	Sim
Timer Pomodoro	-	-	-	-	-	Sim
Modelo Kanban	Sim	Sim	Sim	-	-	Sim
Agenda	Sim	-	-	-	Sim	Sim
Copiloto de Produtividade	-	-	-	-	-	Sim

Tabela 1. Trabalhos correlatos

Ao realizar a análise de possíveis trabalhos correlatos ao tema, foi notado que o meio comercial é extremamente predominante sobre o meio acadêmico, sendo que foram encontrados poucos trabalhos que se referissem à produtividade tóxica relacionado com aplicativos de produtividade. Para apresentar as diferentes características das aplicações identificadas, foi desenvolvida a (Tabela 1) para comparar as diferentes soluções.

A característica mais predominante nos trabalhos correlatos é o modelo de listas de tarefas identificada nos aplicativos Notion, Trello, Todoist, Evernote e Plug-in Outlook de [Matsumoto 2012]. Com o trabalho de [Martino et al. 2020] referenciando-se ao Todoist, Trello e Evernote. Ao contrário do *Pomodoro* que dentre estes aplicativos não foi identificado a existência dessa funcionalidade dentro de suas aplicações. O modelo kanban foi localizado no Notion, Trello e Todoist. Já a agenda os trabalhos que a introduzem são o Notion e o Plug-in Outlook. Com o Wisegress sendo a aplicação que contempla todas as funcionalidades mencionadas, de forma que não só se diferencia, mas contribui por meio do copiloto de produtividade para mitigar a produtividade tóxica no âmbito acadêmico e empresarial.

4. Desenvolvimento

A solução proposta está baseada na colaboração de múltiplas funcionalidades que proporcionam o gerenciamento da produtividade de um indivíduo, assim como auxiliam a organizar o seu tempo, tarefas e entregas. Juntamente com uma ferramenta que caminha com o usuário para aplicar recomendações dos modelos de produtividade a ele utilizados.

4.1. Modelos de Produtividade ao Wisegress

Os modelos de produtividade estão baseados em características distintas, para proporcionar amplas possibilidades para cada usuário, dessa forma, sucederam dois conceitos utilizados na aplicação. Começando pelas “Tarefas”, que seriam partes de projetos maiores do usuário, onde necessitariam de uma divisão do andamento daquele afazer, separando-os em situações que variam de “A Fazer”, indo para “Em Progresso” e finalizando com “Concluído”. Podendo ser definida uma data limite para a execução daquele afazer e uma prioridade de execução, variando de Baixíssima, Baixa, Média, Alta e Urgente. Já o segundo conceito seria voltado à uma “Pendência” do usuário no seu dia a dia, tendo apenas a prioridade como um parâmetro da sua execução.

Portanto, estabeleceu-se quatro modelos diferentes de produtividade baseados nas “Tarefas” e “Pendências” do usuário, que tem como objetivo auxiliar de diferentes formas de se trabalhar com os seus afazeres com a finalidade de aumentar a sua produtividade. Tais modelos estão baseados em Lista de Tarefas, *Kanban*, *Pomodoro* e Pendências.

O modelo primário está baseado em uma Lista de Tarefas, onde o usuário pode nomear cada afazer, definindo aos seus entregáveis uma prioridade e definindo a eles datas de finalização. As tarefas criadas pelo usuário podem ser transformadas em agendamentos e visualizadas em um calendário, sendo que o formato dessa visualização pode ser alternado entre dia, semana e mês. A lista de tarefas pode ser facilmente alterada para um modelo *Kanban*, que se trata de um “método de gestão de mudanças”, conforme [Boeg 2018]. Nesse caso, a organização das tarefas está disposta num quadro separado por colunas que representam as situações das tarefas, sendo elas “A Fazer”, “Em Progresso” e “Concluído”. Dessa maneira, é viável a exibição clara das tarefas separadas pela situação em que se encontra, com a possibilidade da alternância de sua situação conforme o andamento da sua execução.

Para controlar o tempo e as tarefas planejadas, foi desenvolvido uma opção de cronômetro *Pomodoro* que viabiliza ao usuário horários de foco e pequenos intervalos de tempo, para a manutenção do seu desempenho. Tal funcionalidade é mantida de forma global na aplicação proporcionando liberdade ao usuário, que pode estar em outras funcionalidades e funções, sendo notificado ao passo que o período de foco termina ou se inicia.

As tarefas são um dos possíveis meios encontrados para a realização dos afazeres de um indivíduo, pois na aplicação é viável criar as suas próprias “Pendências”, que figuram os pequenos afazeres que o indivíduo encontra diariamente. Neste caso, é possível descrever a pendência, assim como definir uma prioridade de execução, sem uma data de finalização ou possíveis prioridades como nas tarefas. As pendências têm uma visualização única na aplicação, proporcionando uma facilidade a mais para o usuário na identificação dos seus afazeres, com atalhos rápidos para a visualização e manipulação das pendências, denominada também de “caixa de entrada”.

Ao passo que o usuário utiliza os modelos de produtividade, existe uma funcionalidade de copiloto que trabalha em conjunto, sendo ela um provedor de otimização da produtividade do usuário. O copiloto funciona como um assistente pessoal do usuário, que entende o que funciona melhor para ele e faz recomendações com base no uso e feedbacks, além de avisar ao usuário de momentos e funções importantes para manter a

organização das tarefas do usuário, como por exemplo, o “dia da limpeza” que consiste em um lembrete para manter o ambiente organizado em determinados dias da semana, que pode ser posteriormente configurado conforme a necessidade. A sua base de funcionamento está atrelada aos feedbacks que o usuário provém ao finalizar uma Tarefa ou Pendência, sendo uma demonstração de quão agradável está sendo utilizar tal modelo de produtividade. A identificação de qual modelo está sendo utilizado é com base em qual deles o usuário está finalizando a Tarefa ou Pendência. Este feedback segue a seguinte escala: Ruim, Razoável, Bom, Muito Bom e Excelente, onde o usuário deve selecionar uma das opções e caso seja escolhida a pior, sendo a Ruim, ele deve mencionar via mensagem de texto o que levou a tomar tal decisão. Com o feedback coletado será definido um *ranking* de pontos para o usuário, ao qual ele será informado de forma visual para cada modelo de produtividade, proporcionando a ele uma visão clara do que melhor se enquadra a ele. As sugestões que o sistema aplicará ao usuário estão diretamente ligadas aos modelos com piores avaliações, sendo sugerido ao usuário outros modelos para sua utilização. Da mesma forma, os modelos que tiverem mais pontos o sistema alertará para o usuário priorizar a utilização destes modelos.

4.2. Diagrama de Fluxo

Fonte: Gusthawo e Nathan

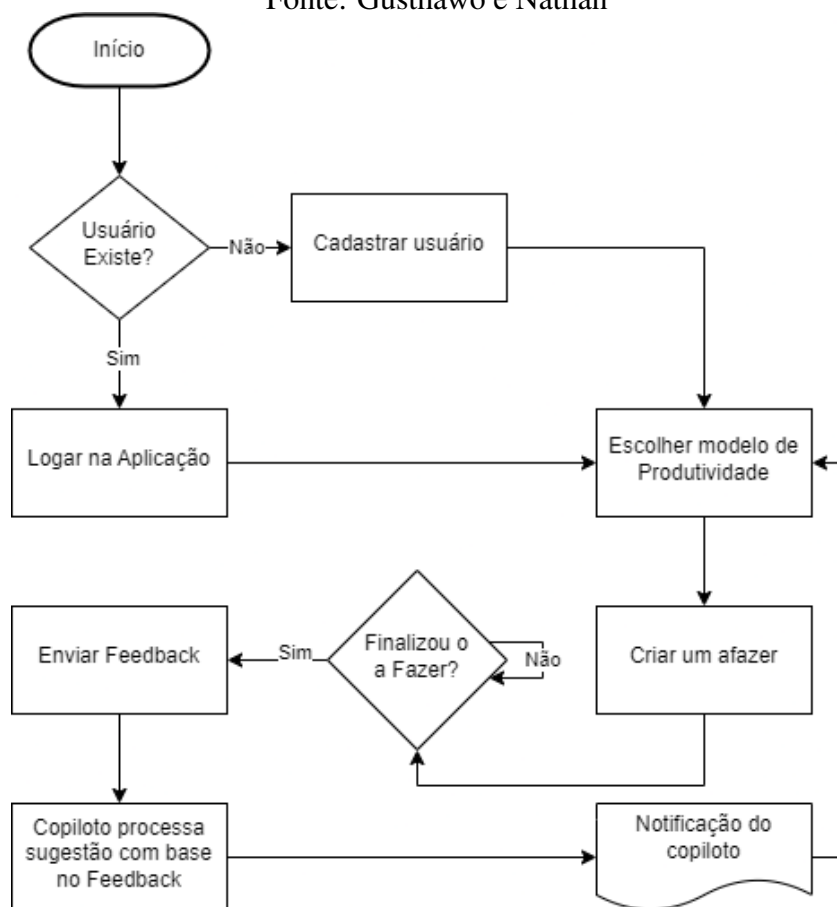


Figura 1. Diagrama de Fluxo

4.3. Modelagem de relacionamento

A aplicação terá uma entidade principal sendo ela o usuário, que tem algumas possibilidades de modelos de produtividade a sua disposição. Tais modelos foram nomeados como “Preferences” e classificados como *todo*, *kanban*, agenda e *pomodoro*. O *todo* está relacionado as pendências do usuário, sendo estes os pequenos afazeres que ele tem como responsabilidade. Por outro lado, as tarefas se referem as “*tasks*” estas que estão relacionadas ao *kanban*, agenda e *pomodoro*, onde estes três modelos implicam em diferentes formas de organização e gerenciamento das tarefas.

Conforme o usuário finaliza as pendências ou as tarefas será solicitado ao usuário que ele informe um índice de satisfação com aquele modelo de produtividade utilizado, gerando assim um *feedback*. A satisfação do usuário pode variar de “Excelente”, “Muito Bom”, “Bom”, “Razoável” e “Ruim”, sendo que no caso da seleção do “Ruim”, será solicitado uma justificativa de texto ao usuário para informar o motivo de tal desagrado. Para todo *feedback* passado pelo usuário será contabilizado na pontuação de cada modelo de produtividade.

Fonte: Gusthawo e Nathan

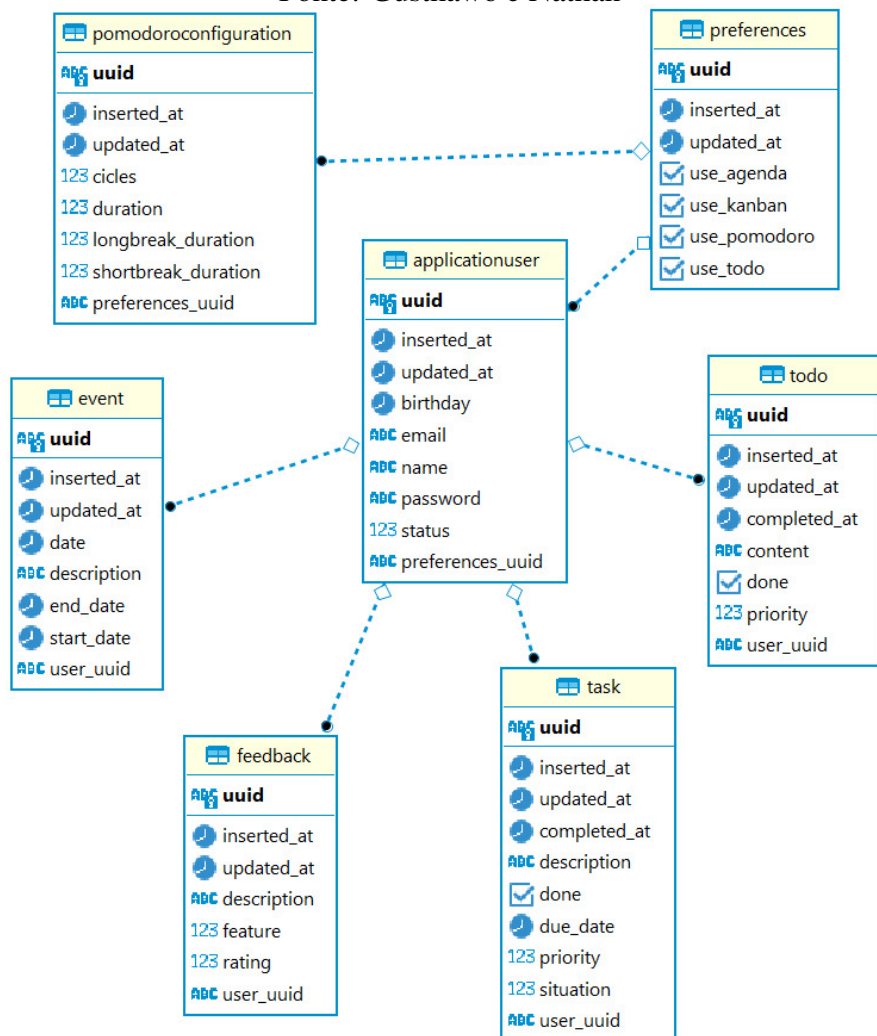


Figura 2. Modelagem de Relacionamento

A modelagem de relacionamento (Figura 2) acima demonstra que o usuário (*applicationuser*) é a informação principal da aplicação e se relaciona com as demais tabelas diretamente, com exceção da tabela de configuração do *pomodoro* (*pomodoroconfiguraction*) onde a relação é indireta.

4.4. Arquitetura do projeto

O projeto está separado em dois repositórios, sendo eles o *wisegress* focado no front-end e o *wisegress-api* com o back-end da aplicação. Dessa maneira, foi possível organizar mais eficientemente a responsabilidade de cada parte do sistema, facilitando o desenvolvimento e manutenção. Os repositórios foram empregados de maneira que o *wisegress-api* está na parte do servidor e é responsável por manter as API's e todas as regras de negócio operacionais, e o *wisegress* está no lado do cliente consumindo as API's e apresentando as telas ao usuário. Para a organização do desenvolvimento do corrente projeto foi utilizada a metodologia *SCRUM*, que segundo [da Silva and Lovato 2016] “oferece um conjunto de práticas que tem como objetivo manter o gerenciamento do projeto visível aos usuários do modelo”. A aplicação utilizada para o gerenciamento do *SCRUM* foi o *Trello* que promove a visualização das tarefas no modelo *Kanban*.

4.5. Ferramentas

O back-end é desenvolvido com a linguagem de programação *Java* [Oracle 2010], que teve o desenvolvimento baseado no ecossistema *Quarkus* [Hat 2019] para organizar os contêineres e a escalabilidade da aplicação conforme a sua necessidade e usufruir de seus outros frameworks. Foi utilizado o banco relacional *PostgreSQL* [Group 1986]. Para o gerenciamento das operações do banco de dados houve a utilização do *Hibernate* [Hat 2001] (para mapeamento objeto-relacional) e foi contado com o *Panache* [Hat 2019], outro framework *Java* para a simplificação dos modelos de persistência. Para a construção das API's foi utilizado do framework *Jakarta* [Foundation 2010], que implementa de forma facilitada especificações e padrões para o desenvolvimento simplificado, o *JUnit* (utilizado para testes) [Erich Gamma 2019], além da utilização de autenticação *JWT* (Json Web Token) na parte da segurança do tráfego de dados da aplicação.

Para o front-end foi utilizado da linguagem de programação *JavaScript* [Eich 1995] para o desenvolvimento, com o *Node.js* gerenciando o ambiente do lado do servidor da aplicação. Juntamente com o framework *VueJS* [You 2014] que proporciona a criação de componentes e organização do front-end, em paralelo a outros frameworks como o *Vuetify* [Leider 2016] para a obtenção de componentes prontos e o *Vite* [Vite 2021] para a realização de testes no front-end, aos quais garantem a conformidade e o funcionamento da solução.

A implementação da solução a fim de mitigar os efeitos da produtividade tóxica tem como base tecnologias separadas em front-end, como define [L. C. B. Rocha 2018], “parte da aplicação na qual o usuário interage diretamente com a interface”, e o back-end que [L. C. B. Rocha 2018] “está implementado toda a regra de negócios do sistema, além de operações de acesso ao banco de dados e tratamento de questões relacionadas à segurança”.

4.6. Funcionalidades da aplicação

Foi desenvolvido uma página para a manipulação das pendências do usuário (Figura 3), possibilitando a criação, edição, exclusão, listagem e finalização das pendências. Na

mesma figura é possível visualizar o *Pomodoro*, ferramenta que possibilita a utilização conjunta com demais funcionalidades da aplicação, sendo possível de ser acessado via um atalho na barra de menu superior da aplicação.

Fonte: Gusthawo e Nathan



Figura 3. Pendências junto ao Pomodoro

Outra funcionalidade desenvolvida foram as tarefas (Figura 4), que possuem as opções de criação, edição, e listagem, com a possibilidade de finalização. Na mesma tela pode-se observar as pendências que foram cadastradas pelo usuário, por meio de um atalho rápido, ao qual possibilita uma fácil visualização e manipulação.

Fonte: Gusthawo e Nathan

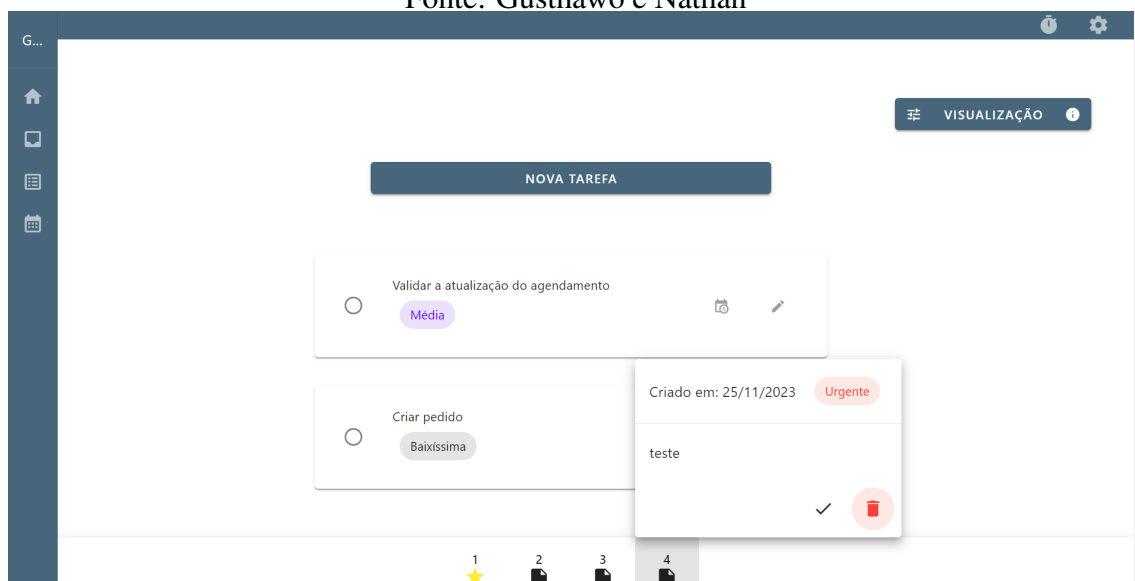


Figura 4. Lista de Tarefas

Na mesma tela de tarefas existe um botão de “Visualização” que possibilita a al-

ternância entre duas exibições das tarefas do usuário, sendo a primeira exibição a Lista (Figura 5) e a segunda exibição o modelo de *Kanban* (Figura 6). Neste segundo modelo foi separado as situações das tarefas em colunas diferentes, possibilitando ao usuário alterar arrastando e soltando as tarefas de acordo com a necessidade.

Fonte: Gusthawo e Nathan



Figura 5. Kanban

De modo a otimizar a organização do usuário, foi desenvolvida uma agenda para exibição das tarefas, as quais tem uma data de limite para a sua execução. Tal informação é a base para a organização das tarefas no calendário de modo a exibir e possibilitar que haja a edição das tarefas criadas.

Fonte: Gusthawo e Nathan

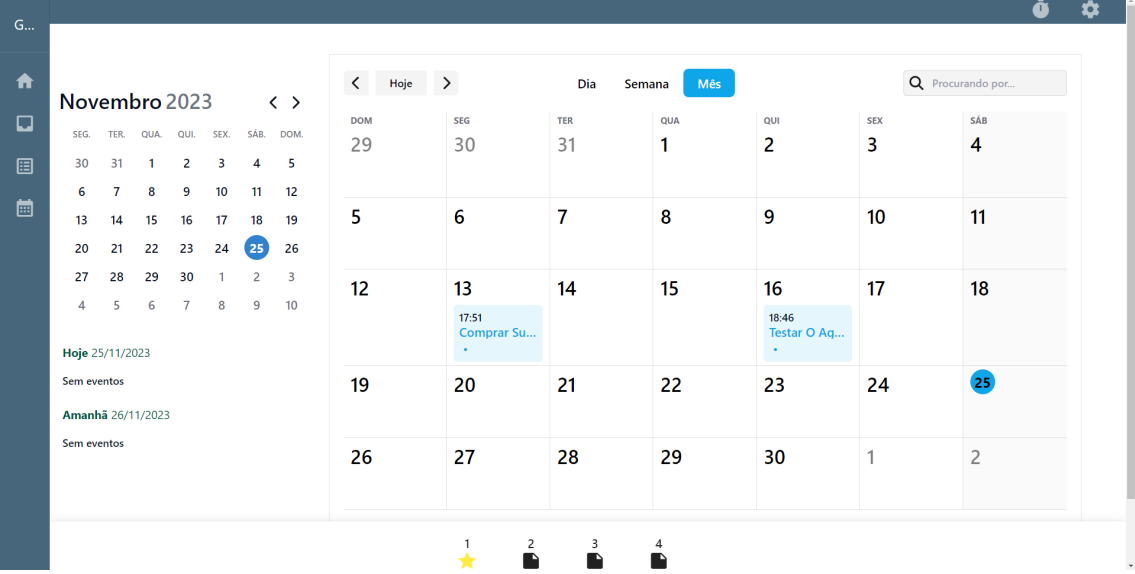


Figura 6. Agenda

5. Resultados

O copiloto foi baseado em feedbacks providos pelo usuário, estes que serão coletados a cada finalização de uma tarefa ou pendência por meio de uma seleção de quão satisfeito está o usuário com o modelo de produtividade que ele havia escolhido. Sendo que o modelo a ser considerado para avaliação será aquele que o usuário estiver utilizando no momento da finalização da sua tarefa ou pendência. Ao clicar no botão de finalização a esquerda da descrição da tarefa (Figura 7), é exibido uma nova tela com uma seleção de satisfação que varia em cinco estágios representados por expressões de emojis. As expressões variam de “Excelente”, “Muito Bom”, “Bom”, “Razoável” e “Ruim”. Caso seja selecionado a pior expressão facial, sendo ela a “Ruim”, uma caixa de texto será exibida para o usuário informar o que o levou a tomar tal decisão.



Figura 7. Obtenção dos Feedbacks do Usuário

Ao ser selecionado um *feedback* o copiloto realizará uma qualificação dos modelos de produtividade para o usuário, de forma a contabilizar os pontos do *feedback* passado. Os pontos são a forma de qualificar os diferentes modelos e possibilitar assim a identificação dos que o usuário melhor se adaptou. O modelo de lista foi utilizado pelo usuário no momento da finalização da Tarefa, sendo passado o *feedback* mais negativo possível para o copiloto, que ajustou os pontos dos modelos de produtividade e exibiu ao usuário o *ranking* atualizado, de acordo com as estrelas pontuadas em tela.

Fonte: Gusthawo e Nathan

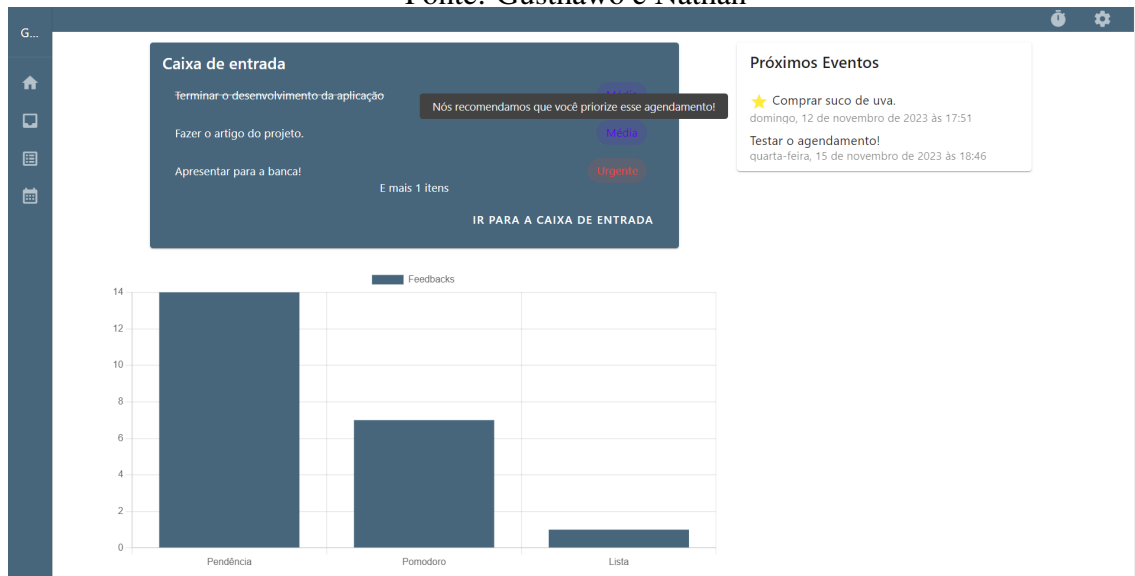


Figura 8. Resultado do Feedback do Usuário com o Modelo de Produtividade de Lista

Realizando o processo em outro modelo de produtividade, desta vez foi realizado a finalização de uma Tarefa com o usuário utilizando o modelo de *Kanban*, sendo passado ao feedback o cenário contrário onde a utilização daquele modelo seria a melhor possível, com o usuário tendo uma excelente experiência.

Fonte: Gusthawo e Nathan

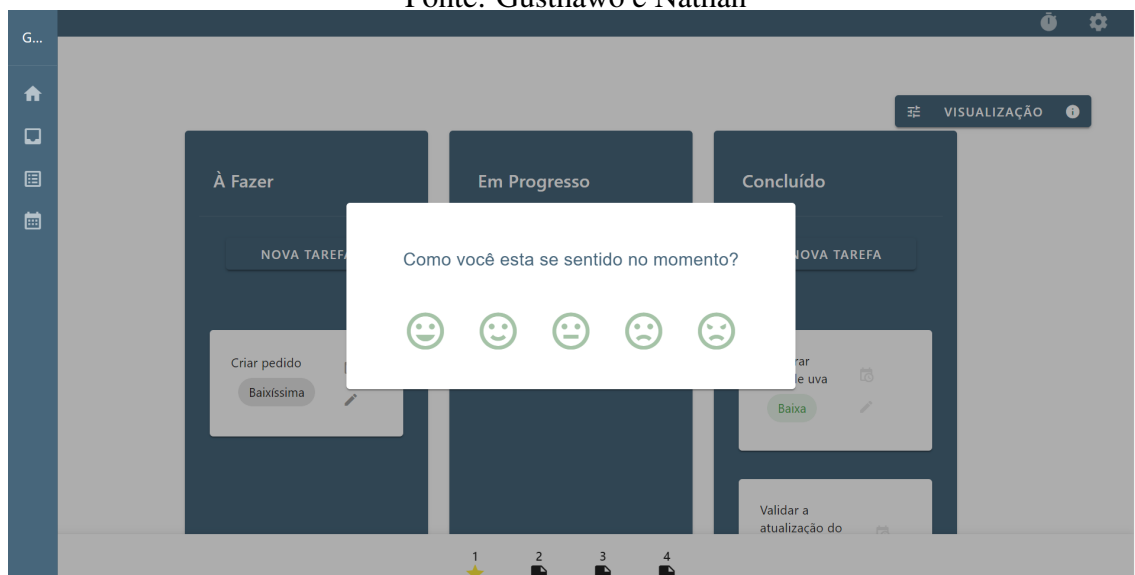


Figura 9. Feedback do Usuário com o Modelo de Produtividade de *Kanban*

Dessa forma, o copiloto determinou o modelo de *Kanban* sendo o ideal para o usuário após realizar o cálculo entre os modelos possíveis e encontrar aquele com a melhor nota. Conforme os feedbacks, *Kanban* é o modelo que o usuário melhor se adaptou, portanto ele deve optar por este modelo de produtividade. A ferramenta possibilitou

a determinação dos diferentes modelos de acordo com os feedbacks registrados, o que mostra que está apta passar ao usuário um retorno visual de acordo com a utilização e a preferência do usuário. Dessa forma, o sistema irá sinalizar ao usuário com uma estrela amarela ao lado da opção de visualização *Kanban*, além de trazer uma mensagem indicando para que se mantenha utilizando este modelo.

Fonte: Gusthawo e Nathan



Figura 10. Resultado do Feedback do Usuário com o Modelo de Produtividade de Kanban

6. Conclusão e Trabalhos futuros

Para se obter um grande rendimento tanto no âmbito acadêmico como no mercado de trabalho, a perspectiva de volume é sobreposta a de qualidade, que ocasiona em prejuízos físicos e mentais para diferentes indivíduos nas mais diversas áreas. A sobrecarga de tarefas é ocasionada pela visão de que para ter resultado profissional ou para construir uma base sólida de conhecimento, elas devem se submeter a uma entrega constante de uma grande quantidade de afazeres. Entretanto, a quantidade de entregas não necessariamente é sinônimo de qualidade, sendo o ponto mais prejudicado pelo alto volume de entregas. Portanto, uma tarefa bem feita e produzida de forma que entregue um valor agregado coincidente com o solicitado é mais valiosa que várias tarefas entregues mas executadas com baixo nível de qualidade.

De tal forma, verificou-se a necessidade da construção de uma aplicação que reunisse num único lugar diferentes maneiras que auxiliassem na entrega de valor por um usuário. Para realizar esse objetivo foi orquestrado de maneira simplificada e objetiva diferentes modelos de produtividade, focados na organização dos afazeres de um indivíduo. Tais processos, são monitorados por um copiloto que recebe feedbacks do usuário e sugere alterações de uso dos modelos de produtividade, alterando ao usuário as formas que ele melhor se adapta pois dessa forma conseguimos trazer à tona uma experiência única para cada indivíduo que venha a utilizar.

A fim de validar a aplicação em cenário real, ao qual diferentes pessoas pudessem

usufruir da solução desenvolvida, seria necessário implantar em um time dentro de uma empresa para a realização de testes e coleta de feedbacks sobre o uso da solução. Tanto que, num cenário empresarial possibilitaria o entendimento de se a aplicação estaria auxiliando no andamento dos projetos e no próprio dia a dia dos colaboradores. Sendo que, para realizar a coleta de dados substancial seria necessário um tempo de ao menos 2 meses para que cada pessoa primeiramente possa se habituar com o sistema, em seguida que ela consiga adequar a rotina dela ao uso da aplicação e que principalmente ela possa utilizar dos diferentes modelos de produtividade e se beneficiar do copiloto. Para a ferramenta se adequar aos diferentes usuários e aos seus estilos próprios, cada pessoa deve usufruir de cada modelo, para assim o copiloto começar a sugerir adequações na sua utilização.

Para aprimorar a ferramenta do copiloto e melhorar as suas sugestões, a implementação de uma Inteligência Artificial generativa seria adequada para compreender os feedbacks textuais passados pelo usuário, ao selecionar a pior avaliação do modelo de produtividade. Assim, poderá ser feito uma análise do retorno do usuário para aprimorar as sugestões e até mesmo aplicar diferentes configurações na aplicação, como no uso do *Pomodoro* e na organização das tarefas da agenda, possibilitando mais personalizações do uso da aplicação aos usuários.

Com o objetivo de proporcionar mais opções de produtividade para o usuário, uma funcionalidade para criar páginas totalmente customizáveis poderia ser implementada a fim de que o usuário possa fazer anotações livres, adicionar imagens, links e outros elementos úteis para o desenvolvimento de uma ideia, que poderá ser transformada em uma tarefa posteriormente.

Referências

- Allen, D. (2015). *A arte de fazer acontecer: O método GTD - Getting Things Done: Estratégias para aumentar a produtividade e reduzir o estresse*. Editora Sextante.
- ANDRADE, B. D. S. C. G. C. (2022). *Aplicação do pomodoro, estimativas e uso do burndown chart em uma fábrica de software acadêmica*. Disponível em: <http://anais.unievangelica.edu.br/index.php/adalovelace/article/view/8293>. Acesso em: 15 out. 2023. Revista Ada Lovelace.
- Boeg, J. (2018). Kanban em 10 passos. Disponível em: <https://gianfratti.com/wp-content/uploads/2018/04/InfoQBrasil-Kanban10Passos.pdf>. Acesso em 15 de nov. 2023.
- Cristo, F. d., Farias, I. M. S. U. d., Cavalcante, A. C., Medeiros, A. L. G. d., Lima, G. D. O., and Diogo, W. F. Q. (2019). O ensino superior e suas exigências: Consequências na saúde mental dos graduandos. *Trabalho (En)Cena*, 4(2):485–505.
- da Silva, E. C. and Lovato, L. A. (2016). Framework scrum: Eficiência em projetos de software. *Revista de Gestão e Projetos*, 7(2):01–15.
- Eich, B. (1995). Javascript. Disponível em: <https://ecma-international.org/> Acesso em 20 de nov. 2023.
- Erich Gamma, K. B. (2019). Junit5. Disponível em: <https://junit.org/junit5/> Acesso em 20 de nov. 2023.

- Foundation, E. (2010). Jakarta. *Disponível em: <https://jakarta.ee/>* Acesso em 20 de nov. 2023.
- Group, P. G. D. (1986). PostgreSQL. *Disponível em: <https://www.postgresql.org/>* Acesso em 24 de nov. 2023.
- Han, B.-C. (2015). *Sociedade do cansaço*. Editora Vozes.
- Harrison, B. J. (1999). *Are you to burn out? Fund Raising Management*.
- Hat, R. (2001). Hibernate. *Disponível em: <https://hibernate.org/>* Acesso em 20 de nov. 2023.
- Hat, R. (2019). Panache. *Disponível em: <https://pt.quarkus.io/guides/hibernate-orm-panache>* Acesso em 20 de nov. 2023.
- Kilaru, L. (2022). The ugly truth about toxic productivity. *Disponível em: Medium <https://medium.com/simplish/the-ugly-truth-about-toxic-productivity-413839080a6b>* Acesso em: 04 out. 2023.
- L. C. B. Rocha, V. C. Calazans, V. C. C. T. H. F. V. (2018). Índice de popularidade das linguagens de programação e frameworks front-end e back-end nas fábricas de software da região de belo horizonte. *Disponível em: <http://revista.fumec.br/index.php/computacaoesociedade/article/view/7308>*. Acesso em 15 nov. 2023.
- Leider, J. (2016). Vuetifyjs. *Disponível em: <https://vuetifyjs.com/en/>* Acesso em 20 de nov. 2023.
- Lipp, M. E. N. (1998). *Como enfrentar o stress*. Icone, 5ª edição.
- Martino, L. M. S., Goulart, A., and Godinho, T. (2020). Disciplina pessoal, dispositivos e precarização do trabalho em três aplicativos de produtividade. *Ação Midiática–Estudos em Comunicação, Sociedade e Cultura.*, pages 33–54.
- Mateus, A. P. Z. (2021). Sociedade do cansaço: a auto exploração do indivíduo associada ao excesso de positividade na sociedade do desempenho.
- Matsumoto, S. (2012). Um framework baseado em plug-ins para raciocínio em ontologias pr-owl 2.
- Oracle (2010). Java. *Disponível em: <https://www.java.com/pt-BR/>* Acesso em 24 de nov. 2023.
- PEREIRA, S. D. A. M. (2021). A síndrome de burnout em profissionais de tecnologia durante a pandemia do covid-19.
- Vite (2021). Vitest. *Disponível em: <https://github.com/vitest-dev/vitest>* Acesso em 25 de nov. 2023.
- You, E. (2014). Vuejs. *Disponível em: <https://vuejs.org/>* Acesso em 20 de nov. 2023.