



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Giordan Colossi Bianchini

Aplicação de Gamificação em um Sistema MES

Florianópolis
2024

Giordan Colossi Bianchini

Aplicação de Gamificação em um Sistema MES

Relatório final da disciplina DAS5511 (Projeto de Fim de Curso) como Trabalho de Conclusão do Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Santa Catarina em Florianópolis.

Orientador: Prof. Rodrigo Castelan Carlson, Dr.
Supervisor: Samir Fiates, Eng.

Florianópolis
2024

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Bianchini, Giordan Colossi
Aplicação de Gamificação em um Sistema MES / Giordan
Colossi Bianchini ; orientador, Rodrigo Castelan Carlson,
2024.
59 p.

Tese (doutorado) - Universidade Federal de Santa
Catarina, Centro Tecnológico, , Florianópolis, 2024.

Inclui referências.

1. . 2. Sistemas MES. 3. Gamificação. I. Carlson,
Rodrigo Castelan. II. Universidade Federal de Santa
Catarina. . III. Título.

Giordan Colossi Bianchini

Aplicação de Gamificação em um Sistema MES

Esta monografia foi julgada no contexto da disciplina DAS5511 (Projeto de Fim de Curso) e aprovada em sua forma final pelo Curso de Graduação em Engenharia de Controle e Automação

Florianópolis, 27 de Fevereiro de 2024.

Prof. Marcelo de Lellis, Dr.
Coordenador do Curso

Banca Examinadora:

Prof. Rodrigo Castelan Carlson , Dr.
Orientador
UFSC/CTC/DAS

Samir Fiates, Eng.
Supervisor
HarboR

Prof. Max Hering de Queiroz, Dr.
Avaliador
UFSC/CTC/DAS

Prof. Hector Bessa Silveira, Dr.
Presidente da Banca
UFSC/CTC/DAS

Este trabalho é dedicado aos meus colegas de classe e
aos meus queridos pais.

AGRADECIMENTOS

Agradeço a meus queridos pais e irmã, aos meus amigos, meus empregadores e a todos os professores dessa minha caminhada.

DECLARAÇÃO DE PUBLICIDADE

Florianópolis, quinze de Fevereiro de dois mil e vinte e quatro.

Na condição de representante da HarboR na qual o presente trabalho foi realizado, declaro não haver ressalvas quanto ao aspecto de sigilo ou propriedade intelectual sobre as informações contidas neste documento, que impeçam a sua publicação por parte da Universidade Federal de Santa Catarina (UFSC) para acesso pelo público em geral, incluindo a sua disponibilização *online* no Repositório Institucional da Biblioteca Universitária da UFSC. Além disso, declaro ciência de que o autor, na condição de estudante da UFSC, é obrigado a depositar este documento, por se tratar de um Trabalho de Conclusão de Curso, no referido Repositório Institucional, em atendimento à Resolução Normativa n° 126/2019/CUn.

Por estar de acordo com esses termos, subscrevo-me abaixo.



Documento assinado digitalmente
PAULO AFONSO LEAL NARCISO FILHO
Data: 18/02/2024 21:11:57-0300
CPF: ***.381.629-**
Verifique as assinaturas em <https://v.ufsc.br>

Paulo Afonso Leal Narciso Filho
HarboR

RESUMO

Este trabalho aborda a implementação de um Sistema de Execução de Manufatura (MES) em ambientes industriais, visando otimizar a gestão de processos e aprimorar a eficiência operacional. O problema identificado reside na dificuldade de engajar os usuários no pleno aproveitamento do MES, resultando em subutilização e impacto negativo no desempenho global.

A solução proposta consiste na integração de elementos de gamificação no MES, explorando estratégias para motivar e engajar os usuários. A gamificação é aplicada como ferramenta para estimular a participação ativa, promover a competição saudável e elevar o interesse dos colaboradores nas atividades relacionadas ao sistema MES.

A metodologia adotada envolveu uma revisão bibliográfica sobre sistemas MES e gamificação. Além disso, foram analisados casos de sucesso na aplicação de gamificação. E, através das revisões o desenvolvimento se baseou em metodologias ágeis como Scrum e Kanban.

O projeto, embora ainda não tenha apresentado resultados tangíveis devido a limitações de tempo, baseia-se na busca por melhores desempenhos no engajamento dos usuários para otimizar a utilização do sistema. A estratégia adotada prevê ajustes contínuos à medida que a compreensão dos usuários se aprofunda, visando alcançar os objetivos fundamentais da HarboR e de seus clientes. O foco principal está na criação de mecanismos para envolver e fidelizar os clientes, promovendo, por conseguinte, um crescimento sustentável da Harbor e das empresas usuárias do sistema no setor específico.

Palavras-chave: Sistemas MES. Gamificação. Engajamento.

ABSTRACT

This work addresses the implementation of a Manufacturing Execution System (MES) in industrial environments, aiming to optimize process management and enhance operational efficiency. The identified problem lies in the difficulty of engaging users in fully utilizing the MES, resulting in underutilization and a negative impact on overall performance.

The proposed solution consists of integrating gamification elements into the MES, exploring strategies to motivate and engage users. Gamification is applied as a tool to stimulate active participation, promote healthy competition, and increase employees' interest in activities related to the MES.

The adopted methodology involved a literature review on MES systems and gamification. Additionally, successful gamification application cases were analyzed. And through these reviews, the development was based on agile methodologies such as Scrum and Kanban.

The project, although it has not yet shown tangible results due to time limitations, is based on the pursuit of better user engagement performance to optimize system utilization. The adopted strategy foresees continuous adjustments as users' understanding deepens, aiming to achieve Harbor's and its clients' fundamental objectives. The main focus is on creating mechanisms to engage and retain customers, thus promoting sustainable growth for Harbor and the system's user companies in the specific sector.

Keywords: Manufacturing Execution System. Gamification. Engagement.

LISTA DE FIGURAS

Figura 1 – HarboR.	15
Figura 2 – InfinityQS.	15
Figura 3 – LiveMES.	16
Figura 4 – Pirâmide da automação.	17
Figura 5 – Tipos de jogadores de Bartle.	25
Figura 6 – Exemplo de ciclo.	29
Figura 7 – Exemplo de dashboard.	34
Figura 8 – Exemplo de cockpit.	34
Figura 9 – Exemplo de Tabela da Fórmula 1.	36
Figura 10 – Exemplo de desenho de solução.	36
Figura 11 – Diagrama de caso de uso.	42
Figura 12 – Diagrama de sequencia.	42
Figura 13 – Cockpit com acesso ao modal adicionado.	43
Figura 14 – Novo modal aberto no cockpit.	43
Figura 15 – Overlay aberto na frente do gráfico.	44
Figura 16 – Overlay deslocado.	44
Figura 17 – Código html do cockpit.	45
Figura 18 – Código typescript do cockpit.	45
Figura 19 – Código html do modal.	46
Figura 20 – Código typescript do modal parte 1.	46
Figura 21 – Código typescript do modal parte 2.	47
Figura 22 – Código typescript do modal parte 3.	47
Figura 23 – Código typescript do modal parte 4.	48
Figura 24 – Código typescript do serviço.	48
Figura 25 – Código html do overlay parte 1.	49
Figura 26 – Código html do overlay parte 2.	50
Figura 27 – Código html do overlay parte 3.	50
Figura 28 – Código typescript do overlay parte 1.	51
Figura 29 – Código typescript do overlay parte 2.	51
Figura 30 – Código typescript do overlay parte 3.	52
Figura 31 – Código typescript do overlay parte 4.	52
Figura 32 – Código typescript do overlay parte 5.	53
Figura 33 – Código typescript do overlay parte 6.	53
Figura 34 – Código typescript do overlay parte 7.	54
Figura 35 – Código typescript do overlay parte 8.	54
Figura 36 – Código CSS do overlay parte 1.	55
Figura 37 – Código CSS do overlay parte 2.	55

Figura 38 – Código CSS do overlay parte 3.	56
Figura 39 – Código CSS do overlay parte 4.	56

SUMÁRIO

1	INTRODUÇÃO	12
1.1	OBJETIVOS	13
1.2	ESTRUTURA DO DOCUMENTO	14
2	HARBOR	15
3	SISTEMAS MES E GAMIFICAÇÃO	17
3.1	SISTEMAS MES	17
3.1.1	Definição e Propósito	17
3.1.2	Componentes e Funcionalidades Principais	18
3.1.3	Funcionamento e Processo de Implementação	18
3.1.4	Benefícios e Impacto na Indústria	19
3.1.5	Desafios e Considerações	20
3.1.6	Conclusão	20
3.2	GAMIFICAÇÃO	20
3.2.1	Fundamentos da Gamificação	21
3.2.2	Motivação Intrínseca e Experiência do Usuário	22
3.2.2.1	Motivação intrínseca X Motivação extrínseca	23
3.2.2.2	Tipos de Jogadores	24
3.2.3	Mecânicas de Jogo	26
3.2.3.1	Placares	26
3.2.3.2	Privacidade e Placares	27
3.2.3.3	Missões	28
3.2.3.4	Ciclos de Engajamento	29
3.2.3.5	Feedbacks	30
3.2.3.6	Ganhar Status	31
3.2.4	Conclusão	31
4	LIVEMES E GAMIFICAÇÃO	33
4.1	LIVEMES	33
4.2	GAMIFICAÇÃO NO LIVEMES	35
5	METODOLOGIAS E REQUISITOS	38
5.1	METODOLOGIAS	38
5.2	REQUISITOS E FLUXO DE DADOS	39
6	DESENVOLVIMENTO DO PROJETO E RESULTADOS	41
6.1	DESENVOLVIMENTO DO PROJETO	41
6.2	RESULTADOS	53
7	CONCLUSÃO	57
	REFERÊNCIAS	59

1 INTRODUÇÃO

A crescente digitalização e automação nos ambientes industriais têm conduzido as empresas a adotarem Sistemas de Execução de Manufatura (MES) como ferramentas essenciais para otimizar operações, aumentar a eficiência e promover a visibilidade dos processos produtivos. O MES, define-se como um software de monitoramento online de produtividade, que permite o controle em tempo real da produção, a eliminação de apontamentos em papel e a tomada de decisões baseadas em dados (LIVEMES, s.d.). Contudo, mesmo diante dessa revolução tecnológica, observa-se um fenômeno preocupante: muitas organizações investem em sistemas MES de ponta, mas, paradoxalmente, não conseguem extrair todo o potencial que essas plataformas oferecem.

O problema central reside na subutilização desses sistemas, uma lacuna que não se deve à falta de recursos tecnológicos, mas, sim, à inadequada utilização por parte dos usuários e gestores. Empresas que possuem sistemas MES enfrentam desafios significativos ao não explorar integralmente as funcionalidades disponíveis, resultando em ineficiências operacionais, perda de competitividade e, em última instância, insatisfação dos stakeholders.

Na HarboR, onde foi desenvolvido esse trabalho, é desenvolvido um MES, o LiveMES. Portanto esse problema foi identificado e não prejudica apenas a eficiência operacional dos clientes, mas também tem implicações diretas no sucesso financeiro, reputação no mercado, capacidade de inovação e relacionamento com clientes e parceiros estratégicos da HarboR.

Como solução para esse problema, a proposta de incorporar a gamificação no Sistema de Execução de Manufatura (MES) visa redefinir a interação dos usuários com a plataforma, tornando-a mais cativante e motivadora. A gamificação, estratégia que introduz elementos de jogos em contextos não lúdicos, surge como resposta ao desafio de falta de engajamento e aproveitamento inadequado do sistema.

Essa abordagem busca estimular o envolvimento dos usuários ao transformar suas atividades cotidianas em missões desafiadoras, criando um ambiente interativo e atrativo. Mecânicas de jogos, como tutoriais interativos, conquistas e progressão de níveis, incentivam a exploração contínua do sistema MES, promovendo a aprendizagem e aprofundamento no seu uso.

Além disso, a gamificação introduz elementos competitivos, como rankings e pontuações, para promover uma competição saudável entre os usuários. Essa dinâmica não apenas eleva a motivação individual, mas também fomenta uma cultura de excelência operacional, impulsionando os colaboradores a constantemente melhorarem seus desempenhos.

O reconhecimento público das conquistas dos usuários, seja por meio de pla-

cares de líderes ou recompensas simbólicas, reforça o senso de realização pessoal e contribui para a construção de um ambiente de trabalho positivo e colaborativo.

Ademais, os mecanismos de feedback imediato fornecem informações em tempo real sobre o desempenho dos usuários, permitindo ajustes rápidos e correções contínuas. Essa retroalimentação constante estabelece um ciclo de melhoria, contribuindo não apenas para otimizar o aproveitamento do sistema, mas também para criar um ambiente de trabalho mais engajado, produtivo e alinhado aos objetivos estratégicos da empresa.

Para tal desenvolvimento, foram feitas pesquisas sobre gamificação, inicialmente através de um curso on-line da universidade da Pensilvânia disponibilizado na plataforma Coursera e vídeos disponibilizados pela TEDx Talk, e para confirmar o conhecimento obtido foi feita leitura do livro *Gamification by Design*. Dessa forma, foi iniciado o desenvolvimento em si do projeto de software dentro do MES, fazendo uso de metodologias ágeis como o Kanban.

Apesar de ainda não ter obtido resultados conclusivos, os resultados tendem a ser os melhores possíveis, visto que será preciso refinar o projeto a medida que conhecermos melhor os nossos jogadores, ou seja, os usuários do sistema. E dessa forma, atingir o objetivo principal para a HarboR e seus clientes, criando uma forma de engajar cada vez mais os clientes e fidelizando-os, que por consequência os levará a um caminho de crescimento dentro do seu ramo específico.

Dentro do desenvolvimento desse projeto, que foi composto por algumas fases, inicialmente foi apresentada ao autor a ideia para origem desse projeto pelos CTOs e diretores da HarboR. Logo, o autor ficou responsável por pesquisar sobre o “Gamificação” e definir uma possível solução que deveria ser aprovada pelos diretores. Dentro desse processo de pesquisas e definição de solução foram feitas algumas reuniões para apresentar a solução aos diretores e assim que aprovado por eles foi iniciado o desenvolvimento em si. No desenvolvimento, visto que o LiveMES é um aplicativo web, a parte referente ao back-end da solução já estava pronta. Já a parte gráfica (front-end) foi totalmente desenvolvida pelo autor. Por fim, há a fase de testes, na qual a equipe de testes será responsável por aprovar que não há nenhum bug. No entanto, ainda há alguns pontos a serem ajustados pelo autor do projeto para que o software passe para a fase de testes.

1.1 OBJETIVOS

A partir do problema já mencionado, o objetivo geral do projeto consiste em implementar no LiveMES uma funcionalidade que engaje seus usuários no sistema, ao mesmo tempo em que os torne mais competitivos em seus respectivos ramos por meio do sistema.

Dessa forma, podemos desdobrar o desenvolvimento deste projeto em objetivos

específicos. São eles: validação do uso da gamificação dentro do sistema, estruturação de uma possível solução por meio da gamificação, o desenvolvimento da solução no software e o último, validação da funcionalidade a partir de dados do sistema.

1.2 ESTRUTURA DO DOCUMENTO

Concluindo esse capítulo introdutório, será descrito como estão estruturadas os demais capítulos dessa monografia.

No capítulo 2, é apresentada uma descrição detalhada da empresa na qual o Projeto Final de Curso (PFC) foi realizado, incluindo informações sobre seus processos e produtos.

No capítulo 3 é apresentada uma fundamentação teórica sobre sistemas MES e Gamificação.

No capítulo 4 são descritos o LiveMES e a gamificação que pode ser aplicada no sistema. São explicados conceitos e funcionalidades dentro do LiveMES , assim como a maneira como o método pode ser aplicado.

No capítulo 5 são expostos os requisitos gerais, funcionais e não-funcionais, a serem considerados no projeto. Também, é explicada a metodologia utilizada no projeto.

No capítulo 6, é detalhado o desenvolvimento do projeto de maneira técnica, por meio de códigos e interfaces gráficas. Ao final, uma breve análise dos resultados obtidos é realizada.

No capítulo final, encontra-se a conclusão do PFC. Nela, é apresentado um resumo do que foi realizado, retomando a motivação inicial do trabalho, a solução proposta, os resultados alcançados e uma sugestão para trabalhos futuros.

2 HARBOR

A HarboR foi fundada em 1996 em Lages-SC com foco na prestação de serviços de desenvolvimento de software de MES inicialmente para um único cliente. No primeiro ano da empresa ocorreu uma expansão aumentando o escopo de atuação neste cliente e desenvolvendo soluções para outros clientes. Em 1997 a empresa se mudou para Florianópolis onde localiza-se até hoje. Desde então não parou de crescer, com foco em sistemas industriais. Em 2017, deu os primeiros passos com um novo produto para controle de produção na nuvem, o LiveMES. Esse produto segue em constante crescimento e é cada vez mais usado por empresas ao redor do Brasil. No Brasil, em 2018 foi fechado acordo de parceria com o SENAI para levar a Indústria 4.0 a todos os tamanhos de empresas, e em todo o território nacional (HARBOR, s.d.).

Figura 1 – HarboR.



Fonte: Site da HarboR.

Atualmente, a empresa opera em três segmentos distintos. Em 2002, estabeleceu uma parceria estratégica com a InfinityQS International, renomada desenvolvedora de softwares especializados em controle estatístico de processos. Desde então, a HarboR assume a posição exclusiva de distribuidora desses softwares no mercado brasileiro. A colaboração inclui a capacitação de uma equipe da HarboR, focada na prestação de serviços de implantação e treinamento especializado.

Figura 2 – InfinityQS.



Fonte: Site do InfinityQS.

O segundo setor de atuação é o LiveMES, que engloba uma equipe dedicada ao desenvolvimento contínuo do sistema, bem como outra equipe especializada no suporte ao cliente, denominada PMaaS (Production Manager as a Service). A equipe de desenvolvimento é responsável pela manutenção operacional e implementação de inovações ou sugestões dos clientes no sistema. Por outro lado, o PMaaS é uma equipe multidisciplinar composta por engenheiros, colaborando com a inteligência artificial do

LiveMES. Essa equipe analisa os dados gerados pelo LiveMES e desempenha um papel ativo na identificação e resolução das principais perdas de eficiência. O PMaaS é um dos pilares fundamentais para o êxito do LiveMES em indústrias de diversos setores (LIVEMES, s.d.).

Figura 3 – LiveMES.



Fonte: Site do LiveMES.

Por último, temos o setor de projetos, especializado no desenvolvimento de soluções personalizadas para indústrias em geral. Essas soluções são projetadas para abordar problemas específicos que podem não ser completamente abrangidos pelas capacidades do Infinity e do LiveMES. Adicionalmente, quando necessário, essa equipe oferece suporte em questões relacionadas a integrações e personalizações específicas associadas ao Infinity e ao LiveMES.

3 SISTEMAS MES E GAMIFICAÇÃO

No cenário empresarial contemporâneo, a busca incessante por eficiência operacional e engajamento dos colaboradores tem levado as organizações a explorar inovações que transcendem as fronteiras tradicionais. Nesse contexto, dois conceitos vêm ganhando destaque: Sistemas MES (*Manufacturing Execution Systems*) e Gamificação. Este capítulo se propõe a desbravar as intrincadas interseções dessas duas ferramentas, delineando como a integração de um Sistema MES pode potencializar a eficácia operacional, enquanto a aplicação de elementos lúdicos através da Gamificação pode catalisar o envolvimento e desempenho dos colaboradores. Ao mergulharmos nesse universo convergente, vislumbramos uma nova era de gestão industrial, onde a otimização de processos se entrelaça harmoniosamente com a motivação intrínseca, pode promover um ambiente de trabalho dinâmico e eficiente.

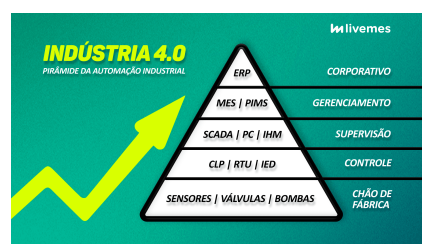
3.1 SISTEMAS MES

Os Sistemas MES (*Manufacturing Execution Systems*) representam uma categoria essencial de tecnologia no contexto da manufatura moderna. Eles desempenham um papel crítico na integração, monitoramento e controle dos processos de produção em tempo real (LIVEMES, s.d.). Neste subcapítulo, vamos explorar os sistemas MES, abrangendo desde sua definição e funcionalidades até sua importância e aplicabilidade na indústria atual.

3.1.1 Definição e Propósito

Os Sistemas MES são sistemas de software projetados para interligar os sistemas de automação da produção (como PLCs e SCADAs) com os sistemas corporativos de nível superior, como os sistemas ERP (*Enterprise Resource Planning*). Eles atuam como uma ponte fundamental entre o chão de fábrica e a gestão empresarial, fornecendo uma camada de controle e visibilidade em tempo real sobre as operações de manufatura.

Figura 4 – Pirâmide da automação.



Fonte: Site do LiveMES.

O propósito principal dos sistemas MES é otimizar a execução dos processos de fabricação, garantindo eficiência, qualidade e conformidade. Eles realizam isso através de uma variedade de funcionalidades, incluindo monitoramento de produção, rastreabilidade de produtos, gestão de qualidade, programação de produção e gestão de ativos.

3.1.2 Componentes e Funcionalidades Principais

Os sistemas MES são compostos por uma variedade de módulos e funcionalidades, cada um desempenhando um papel específico na gestão da produção. Serão citados alguns desses componentes:

1. Controle de Processo:

- Monitoramento em tempo real do fluxo de produção.
- Controle de parâmetros de processo para garantir eficiência e qualidade.

2. Rastreabilidade:

- Registro detalhado do histórico de produção de cada produto.
- Capacidade de rastrear a origem e o destino de componentes individuais ao longo da cadeia de produção.

3. Gestão de Qualidade:

- Implementação e execução de planos de controle de qualidade.
- Monitoramento e análise de dados para garantir conformidade com padrões de qualidade.

4. Integração com outros Sistemas:

- Integração bidirecional com sistemas corporativos, como ERP, para troca de dados e informações.
- Integração com sistemas de automação de chão de fábrica para coleta e controle de dados em tempo real.

3.1.3 Funcionamento e Processo de Implementação

O funcionamento dos sistemas MES envolve várias etapas, desde a coleta de dados até a tomada de decisões baseada em análises em tempo real. O processo típico de implementação inclui

1. Análise de Requisitos:

- Entendimento dos processos de fabricação existentes e identificação de necessidades específicas.
- Definição de objetivos de implementação e métricas de sucesso.

2. Design e Configuração:

- Projeto da arquitetura do sistema MES, incluindo seleção de módulos e funcionalidades relevantes.
- Configuração e personalização do software para atender às necessidades específicas da empresa.

3. Implantação e Integração:

- Instalação do sistema MES e integração com sistemas existentes, como PLCs e ERP.
- Testes de integração e validação para garantir o funcionamento adequado do sistema.

4. Treinamento e Adoção:

- Treinamento de pessoal em todos os níveis da organização para garantir a adoção efetiva do sistema.
- Educação sobre os benefícios e funcionalidades do sistema MES e como utilizá-lo de forma eficaz.

5. Monitoramento e Melhoria Contínua:

- Monitoramento contínuo do desempenho do sistema MES e dos processos de produção.
- Identificação de áreas de melhoria e implementação de ajustes e atualizações conforme necessário.

3.1.4 Benefícios e Impacto na Indústria

Os sistemas MES oferecem uma série de benefícios significativos para as organizações de manufatura, incluindo:

- Aumento da eficiência operacional: Otimização dos processos de produção resulta em maior produtividade e redução de custos.
- Melhoria da qualidade do produto: Monitoramento e controle rigorosos garantem a conformidade com padrões de qualidade e reduzem o desperdício.

- Aumento da visibilidade e rastreabilidade: A capacidade de rastrear produtos ao longo de toda a cadeia de produção aumenta a transparência e simplifica processos de recall, quando necessário.
- Agilidade e capacidade de resposta: Tomada de decisões baseada em dados em tempo real permite ajustes rápidos e adaptação a mudanças nas condições de mercado.
- Redução de riscos e custos de conformidade: Gestão eficaz da qualidade e conformidade com regulamentações reduzem o risco de falhas e multas.

3.1.5 Desafios e Considerações

Apesar dos benefícios, a implementação e operação de sistemas MES também apresentam desafios significativos:

- Complexidade de Implementação: A integração com sistemas existentes e a customização para atender às necessidades específicas da empresa podem ser complexas e demoradas.
- Cultura Organizacional: A adoção bem-sucedida de sistemas MES muitas vezes requer uma mudança na cultura organizacional, com foco em colaboração, transparência e uso efetivo de dados.
- Segurança da Informação: Como os sistemas MES lidam com dados sensíveis e críticos para a operação da empresa, é essencial garantir a segurança da informação contra ameaças internas e externas.

3.1.6 Conclusão

Os sistemas MES representam uma ferramenta poderosa para melhorar a eficiência, qualidade e agilidade das operações de manufatura. No entanto, sua implementação bem-sucedida requer um planejamento cuidadoso, investimento significativo e um compromisso com a mudança organizacional. Ao superar os desafios e aproveitar os benefícios dos sistemas MES, as empresas podem obter uma vantagem competitiva significativa no mercado global.

3.2 GAMIFICAÇÃO

Gamificação, um termo em voga no marketing, pode ter diferentes interpretações. Alguns a veem como a criação de jogos para publicidade, enquanto outros a associam a mundos virtuais 3D que promovem mudanças comportamentais ou treinam usuários em sistemas complexos. A gamificação integra conceitos como jogos sérios,

publicidade em jogos e jogos para mudança, baseando-se em pesquisas em psicologia comportamental e no sucesso de jogos sociais.

A gamificação pode ser definida como um método, que através do uso da psicologia comportamental e mecânicas de jogo envolve usuários e resolve problemas. Este método é flexível, aplicável a qualquer problema solucionável por influência na motivação e comportamento humano.

No contexto do presente projeto de fim de curso (PFC), é explorada a gamificação como ferramenta estratégica, reconhecendo que suas mecânicas de jogo têm limitações específicas na resolução de problemas fundamentais de negócios. No entanto, a gamificação não pode, por si só, resolver problemas estruturais ou abordar problemas críticos de atendimento ao cliente. Além disso, destaca-se que o sucesso da gamificação não necessariamente replicará o impacto viral de jogos específicos, mas pode oferecer soluções eficazes ao se concentrar em simplicidade, recompensa e diversão.

Através do livro *Gamication by Design* (ZICHERMANN; CUNNINGHAM, 2011), foram feitas algumas observações sobre sistemas gamificados, como o Foursquare, demonstrando que, para além dos elementos de jogo, a busca pela conexão impulsiona a experiência do jogador. Nesse sentido, é possível argumentar que a eficácia da gamificação reside na sua simplicidade, destacando que a aplicação não precisa ser complexa para resolver problemas complexos.

Por fim, optou-se por adotar o termo jogador invés de cliente ou usuário, promovendo uma abordagem centrada no engajamento de longo prazo e em uma relação simbiótica. Esta seção proporcionará uma base para a exploração subsequente do processo de desenvolvimento utilizado para o projeto, visando a gamificação de engajamento com clientes.

3.2.1 Fundamentos da Gamificação

Tudo pode ser divertido, inclusive atividades cotidianas. Jogos populares nos últimos anos exploraram temas como plantar culturas, servir mesas e trocar fraldas. A chave do sucesso não se encontra no tema, mas sim nas mecânicas do jogo utilizadas.

Máquinas caça-níqueis em cassinos podem ter marcas diferentes, mas suas mecânicas são praticamente idênticas. A diversão não está no logotipo, mas nas operações subjacentes. A marca atrai, mas são as mecânicas que mantêm os jogadores, por exemplo a imprevisibilidade (COURSERA, s.d.).

Na educação, por exemplo, existe o desafio de combinar diversão e aprendizado. Jogos como “Onde no Mundo Está Carmen Sandiego?” foram sucessos educativos divertidos, porém muitos sistemas falham em equilibrar a combinação. Jogos como *Civilization* e *SimCity* ensinam de forma indireta, usando a diversão como subproduto do aprendizado.

Destaca-se que a pesquisa conduzida pelo Dr. Arne May, da Universidade de Regensburg, Alemanha, na qual oferece uma perspectiva fascinante (ZICHERMANN; CUNNINGHAM, 2011). Seu estudo revelou que aprender uma nova tarefa resulta em um aumento mensurável na matéria cinzenta do cérebro em poucas semanas. Especialistas em neurociência global concordam que o ciclo desafio-realização-recompensa inerente aos jogos estimula a liberação de dopamina no cérebro, fortalecendo à inclinação natural para a participação no jogo.

A pesquisa também destaca que crianças podem aprender por meio de jogos divertidos. No entanto, se a educação é priorizada em detrimento da diversão, a aprendizagem pode não ser tão eficaz. A parte desafiadora é integrar ambos de maneira equilibrada, reconhecendo a importância das mecânicas de jogo e a interação entre diversão e aprendizado.

Portanto, a essência da gamificação reside na compreensão profunda da psicologia humana e dos fatores que impulsionam o seu comportamento. Ao incorporar elementos como desafios, recompensas, rankings e narrativas, a gamificação desencadeia uma série de reações emocionais que transcendem a simples execução de tarefas, transformando o ordinário em extraordinário.

3.2.2 Motivação Intrínseca e Experiência do Usuário

Esta seção busca explicar sobre essa questão psicológica, comportamental e a razão que leva indivíduos aos jogos. Por exemplo, em narrativas mitológicas ou até propagandas de cerveja com mulheres de corpos esculturais, é evidente que o sexo e a violência possuem a capacidade de impulsionar ações extremas, como se espelhar em um Deus ou comprar essa determinada cerveja. No entanto, ao contrário dessas impulsões, os jogos buscam encontrar um equilíbrio único de prazer e diversão (ZICHERMANN; CUNNINGHAM, 2011).

Este fenômeno é explicado pelo conceito central de “fluxo”, derivado dos estudos de Mihaly Csikszentmihalyi sobre felicidade e criatividade. A busca incessante dos desenvolvedores por criar esse estado de “estar na zona” envolve uma interação cuidadosa entre sistema e jogador, encontrando o ponto ideal entre ansiedade e tédio.

Neste contexto, um reforço ao jogador desempenha um papel fundamental. Estudando a conversão de recompensa esperada em ação do jogador, pesquisadores como Pavlov e B. F. Skinner entendem ser crucial estruturar esses sistemas de recompensa de forma muito eficaz.

Ao explorar diferentes cronogramas de reforço, desde o intervalo fixo comum em empregos industriais até a razão variável presente em jogos de caça-níqueis, é compreensível como esses padrões impactam o engajamento do jogador. É essencial planejar a inclusão de recompensas de maneira equilibrada, evitando abusos.

A aplicação desses princípios psicológicos nos jogos apresenta não apenas um

lado negativo, com vícios e negligência de responsabilidades, mas também um lado positivo, melhorando a saúde, a aprendizagem e o estilo de vida dos jogadores.

No âmago desse universo, uma questão persiste: o que impulsiona os indivíduos a se envolverem com os jogos? Essa dinâmica complexa, revela uma interação única de prazer e impulsos que cativam os participantes, transformando os jogos uma experiência tão intrigante e envolvente. Uma teoria subjacente à motivação para jogar sugere quatro razões principais, que podem ser consideradas tanto de forma conjunta quanto separada:

- Maestria.
- Alívio de estresse.
- Diversão.
- Socialização.

Já Nicole Lazzaro, especialista em experiência do jogador, categoriza quatro tipos de diversão em jogos:

- Diversão desafiadora: focada na vitória competitiva.
- Diversão fácil: explorando o sistema do jogo.
- Diversão em estado alterado: a experiência modifica as emoções.
- Diversão social: interação com outros jogadores.

3.2.2.1 Motivação intrínseca X Motivação extrínseca

A análise das motivações dos jogadores na gamificação se desdobra em uma distinção entre origens intrínsecas e extrínsecas, como delineadas pela psicologia. Motivações intrínsecas é a vontade que parte do indivíduo, independentes do entorno, enquanto as motivações extrínsecas são impulsionadas pelo seu arredor, como o desejo de lucrar ou alcançar sucesso em competições de soletração.

No contexto da gamificação, três perspectivas se destacam na dicotomia entre motivações intrínsecas e extrínsecas:

De acordo com Daniel H. Pink, autor de “Drive“, dinheiro é considerado uma recompensa inadequada para estimular jogadores em tarefas complexas. Pesquisas indicam que motivações externas, como dinheiro, não são eficazes em tarefas criativas, inclusive podendo até prejudicar o desempenho criativo. Portanto, Pink argumenta contra o uso de recompensas em dinheiro para impulsionar a criatividade. Já o Dr. John Houston, especialista em competitividade, sugere que pessoas altamente competitivas podem ser autodestrutivas em sua busca incessante pela competição. Sejam

realizadores ou “assassinos” (jogadores agressivos), a alta competitividade persiste mesmo sem ganhos tangíveis e, frequentemente, mesmo quando há desincentivos claros. Outra perspectiva é teoria do viés de superjustificação/substituição, a qual sugere que substituir uma motivação intrínseca por uma recompensa extrínseca é possível, no entanto pode ter efeitos negativos. Introduzir recompensas externas em atividades intrinsecamente motivadoras pode resultar na perda permanente da motivação intrínseca. Uma conclusão dessas perspectivas é que, ao incorporar recompensas, é imperativo manter os jogadores engajados nesse ciclo de recompensa continuamente, impactando o custo total de propriedade na gamificação. Portanto, paradigmas antigos, como a superioridade percebida da motivação intrínseca sobre recompensas extrínsecas, são agora reconsiderados. O novo entendimento destaca a importância de alinhar cuidadosamente motivações intrínsecas e recompensas extrínsecas, reconhecendo e apoiando os jogadores em seus estados motivacionais para promover o alcance eficaz de seus objetivos desejados.

Concluindo, o poder da gamificação reside na ativação da motivação intrínseca. Ao criar experiências que envolvem e ressoam com os desejos naturais de conquista e superação, a gamificação catalisa uma mudança no comportamento do usuário. A busca por recompensas virtuais ou a escalada em um ranking não são apenas indicadores de progresso, são estímulos poderosos que mantêm o jogador imerso na experiência.

A experiência do usuário, nesse contexto, transcende a funcionalidade pura de uma aplicação. Torna-se uma narrativa interativa, onde cada ação e conquista são elementos de uma história mais ampla, conectando-se de forma coesa para formar uma experiência rica e envolvente. Portanto, é importantíssimo conhecer o jogador.

3.2.2.2 Tipos de Jogadores

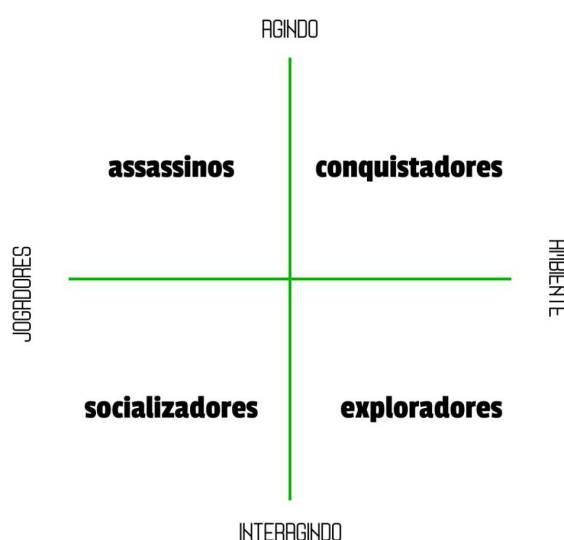
Seguindo a seção anterior, é preciso compreender os jogadores, tanto os atuais quanto os potenciais. Isto é crucial para projetar uma experiência que oriente seu comportamento de maneira desejada. Nesse contexto, a obra de Richard Bartle, focada nos tipos de jogadores, fornece uma estrutura valiosa. Originalmente identificando quatro tipos principais, apesar da possibilidade de separar em oito. Os perfis principais são os mais relevantes e intrigantes para o projeto e são citados a seguir:

- Exploradores: Os jogadores do tipo “Explorador” são motivados pela exploração do mundo do jogo para trazer descobertas significativas à sua comunidade. O foco está na experiência em si, tornando jogos como Super Mario Brothers exemplos notáveis desse tipo.
- Conquistadores: Os “Conquistadores” desempenham um papel vital em jogos competitivos, impulsionando projetos, serviços e marcas. No entanto, projetar

exclusivamente para esse tipo pode ser desafiador, já que nem todos podem alcançar a vitória sem perder o interesse.

- **Socializadores:** Os “Socializadores” jogam em busca de interações sociais significativas, compõem a maior população. Jogos tradicionais, como dominó, bridge e poker, exemplificam essa busca, onde o jogo é um meio para conexões sociais a longo prazo.
- **Assassinos:** Também conhecidos como “griefers”, os “Assassinos” compõem a menor população, buscando vitória e, mais crucialmente, a derrota de outros. Buscam reconhecimento por suas proezas, muitas vezes desafiando outros jogadores.

Figura 5 – Tipos de jogadores de Bartle.



Fonte: Adaptado do Livro Gamification by Desing.

Ao aplicar esses tipos de jogadores a um eixo que varia de ações para interações e de indivíduos para ambientes, obtemos uma compreensão mais rica da dinâmica do jogo, representado na Figura 5. A análise dos resultados do teste de Bartle revela que, embora muitos jogadores possam exibir características de múltiplos tipos, a predominância geral recai sobre os "Socializadores". Esse entendimento aprofundado dos tipos de jogadores fornece uma base essencial para o desenvolvimento eficaz de sistemas gamificados.

Os desenvolvedores de sistemas, predominantemente orientados para conquistas, enfrentam um desafio ao criar sistemas gamificados. Sua tendência natural é acumular pontos, buscar status e competir para vencer, o que não reflete as motivações

do jogador comum. Este, busca a socialização, e embora as conquistas proporcionem uma experiência agradável, não é o combustível principal para sua manutenção no sistema. Projetar com a premissa de que o jogo é principalmente sobre conquistas pode excluir uma parcela significativa da audiência. Em muitos casos, a experiência de jogo é mais valorizada pela comunidade e camaradagem proporcionadas do que pela conquista pura.

3.2.3 Mecânicas de Jogo

Recapitulando uma parte previamente escrita, o desenvolvimento de jogos e sistemas gamificados tem suas raízes profundamente enraizadas na psicologia e no pensamento sistêmico. No contexto da criação de experiências gamificadas, são selecionados e incorporados elementos essenciais de jogos, direcionando a atenção para aqueles que promoverão o maior impacto. Nesse processo, opta-se por negligenciar a estrutura narrativa, focalizando, em vez disso, nas narrativas preexistentes dos jogadores.

As mecânicas de um sistema gamificado são ferramentas que, quando empregadas adequadamente, buscam gerar uma resposta significativa dos jogadores. Para o foco do projeto de fim de curso, são destacados três elementos principais: placares, desafios/missões, integração e ciclos de engajamento, além de dois conceitos extremamente relevantes: feedbacks e ganhar status.

3.2.3.1 Placares

O placar tem o propósito de possibilitar comparações simples, sendo facilmente compreendido sem a necessidade de explicações. Em fliperamas dos anos 80, a presença de placares exibindo altas pontuações muitas vezes representava um desincentivo para jogadores novatos, mesmo quando a escala numérica era pouco clara. Atualmente, são empregados principalmente dois tipos de placares.

Os placares atuais passaram por mudanças significativas desde os dias das máquinas de pinball e fliperamas de ficha, agora focando em incentivos sociais na era do Facebook e do gráfico social. Essa abordagem coloca o jogador no centro das atenções, independentemente de sua posição no ranking, proporcionando uma visão clara de amigos próximos abaixo e indicando o quão próximo estão da próxima pontuação mais alta acima, denominando esse placar de placar de líderes sem desincentivo.

Para os jogadores que se encontram entre os 10 ou 20 melhores, o placar direciona uma abordagem mais direta, revelando sua classificação literal. Essa representação específica tem um significado mais profundo e relevante para esses jogadores, contribuindo para uma experiência mais envolvente e personalizada.

Já nos fliperamas, é desafiador manter os jogadores permanentemente no quadro de líderes de um jogo. Com o tempo, suas pontuações podem ser ultrapassadas,

resultando em sua remoção ou ficar estagnado por semanas até ser superado. Atualmente, estratégias são empregadas para garantir que os jogadores não sejam removidos dos quadros de líderes de forma permanente, denominando essas placares como placares infinitos.

Um exemplo é o Doodle Jump, jogo popular para iPhone, que oferece diferentes visualizações do quadro de líderes, permitindo aos jogadores verem suas classificações localmente, socialmente entre amigos e globalmente em relação a todos os jogadores. Essa customização do quadro de líderes pelos jogadores fornece valiosas informações sobre suas preferências e comportamentos, orientando os designers na criação de experiências mais envolventes. Além disso, jogos como Flight Control implementam quadros de líderes em camadas, exibindo outros jogadores no mesmo nível de proximidade e atividade recente em relação ao jogador principal, gerenciando eficazmente grandes bases de jogadores.

Quadros de líderes sociais, como o Klout e o Yelp, também se destacam ao classificar usuários de acordo com critérios específicos, como o poder social no Twitter ou os check-ins mais frequentes. A capacidade de segmentar o quadro de líderes por amigos ou principais jogadores adiciona uma camada de personalização, proporcionando uma experiência mais dinâmica e atualizada para os participantes.

3.2.3.2 Privacidade e Placares

Apesar desses tipos, a criação de rankings nem sempre é uma tarefa direta, especialmente quando se lida com elementos sensíveis ou desafiadores de quantificar. Isso se torna um dilema, pois o propósito principal dos rankings é realizar comparações públicas, o que pode ser complicado quando se trata de informações que deveriam ser mantidas privadas. Por exemplo, em uma academia, onde se busca promover estilos de vida saudáveis, comparar publicamente o peso de seus membros pode ser constrangedor e afastar potenciais participantes.

A complexidade aumenta quando diferentes participantes têm objetivos distintos, desde treinar para uma maratona até ganhar massa muscular. É crucial que os designers considerem múltiplos tipos de rankings para atender às necessidades variadas dos participantes. Para iniciantes, um ranking que lista a frequência de participação pode ser uma ótima introdução ao sistema, enquanto corredores podem preferir participar de um ranking onde competem com outros membros da academia. Os fisiculturistas podem optar por compartilhar seu peso publicamente e observar seus números aumentarem, mas para aqueles que buscam perder peso, a exposição pública pode ser um desincentivo. Portanto, é essencial equilibrar a exposição pública com o respeito à privacidade e à diversidade de metas de fitness.

3.2.3.3 Missões

Orientações e missões desempenham um papel crucial ao fornecer direcionamento aos jogadores dentro do contexto da experiência gamificada. Imaginem o quão empolgante seria participar de uma caça ao tesouro se a instrução fosse simplesmente sair e encontrar algumas coisas, sem especificações... apenas coisas. Nesse cenário, haveria uma falta significativa de desafio, intrigas e estrutura.

Alguns participantes ingressam no jogo sem compreender completamente os objetivos ou impulsos fundamentais. Mesmo que um desafio não ocupe a posição central da experiência, a inclusão de desafios como uma opção em algum ponto do sistema pode acrescentar profundidade e significado para o jogador. A premissa fundamental é garantir que sempre exista um desafio à disposição dos jogadores, proporcionando-lhes algo interessante e significativo para realizar ou experimentar ao entrar na experiência. Alguns jogadores podem preferir enfrentar desafios sequencialmente, visando superar o máximo possível do jogo, enquanto outros podem escolher abordar apenas o necessário para manter o interesse. Nesse contexto, a responsabilidade do desenvolvedor do sistema é criar uma variedade significativa de opções instigantes.

Existem também as missões cooperativas, colaborar em experiências de busca e missões depende da participação de uma comunidade de jogadores. Esses tipos de missões representam desafios consideráveis ao serem desenvolvidos. Ao organizar, por exemplo, uma partida de futebol, a dificuldade reside não em encontrar postes de gol ou uma bola, mas em garantir que 22 pessoas apareçam ao mesmo tempo para jogar.

Na fase inicial do processo de design de um sistema gamificado, é mais estratégico projetar um jogo para um único jogador, permitindo a evolução para o modo cooperativo à medida que o jogador adquire habilidades e mais participantes se juntam ao jogo. O Gowalla, por exemplo, oferece uma variedade de desafios e missões que um jogador pode explorar em seu próprio tempo. Pode-se completar um desafio após fazer o check-in em um local específico ou até mesmo criar e realizar suas próprias missões para ganhar recompensas. Por exemplo, comprometer-se a caminhar 12 km para arrecadar fundos para o câncer de mama, fazendo check-in no início e no final. Essas realizações são compartilhadas com os amigos, proporcionando orgulho ao jogador.

É evidente que os designs cooperativos possuem maior poder social. O exemplo do futebol mencionado, que envolve uma participação mínima considerável, também oferece uma experiência social e recompensas mais intensas, especialmente quando há um grande número de participantes. Se a comunidade já conta com muitos jogadores ativos, a consideração do design cooperativo é altamente recomendada. Pode-se projetar experiências para jogadores individuais em um contexto de grupo, onde as

conquistas individuais contribuem para o grupo, ou os resultados são compartilhados com o grupo. Algumas vezes, apenas distribuir recompensas em um ambiente de grupo pode ser o suficiente para estimular uma resposta positiva. Exemplos adicionais incluem desafios encadeados oferecidos por redes de fast-food e competições de exercícios físicos em uma empresa de marketing, ilustrando como o design de grupo pode intensificar consideravelmente o envolvimento.

3.2.3.4 Ciclos de Engajamento

Os ciclos de engajamento, embora não sejam exclusivos de jogos, são fortemente influenciados pelo design de loops virais. O designer precisa não apenas entender como o jogador interage inicialmente com o sistema, mas também compreender o processo de saída e, talvez até mais crucialmente, o que o motiva a retornar. Dentro de um ciclo de engajamento social, uma emoção motivadora impulsiona a reintegração do jogador, desencadeando uma chamada social para ação, que se desdobra em progresso visível e/ou recompensas, encerrando o ciclo ao retornar à emoção motivadora.

Figura 6 – Exemplo de ciclo.



Fonte: Livro Gamificaton by Desing.

Integrar a viralidade intrínseca em um sistema gamificado demanda considerar o ciclo de engajamento social em diferentes estágios ao longo da jornada rumo à maestria, conforme discutido anteriormente. Um exemplo disso pode ser observado no Twitter, onde os loops virais operam de maneira eficaz nessa plataforma de comunica-

ção renomada. Para aqueles que estão dando os primeiros passos no Twitter, o ciclo de engajamento compreende:

- Emoção motivadora(*Motivating Emotion*) = Conectar e expressar
- Reengajamento do usuário(*Player Re-Engagement*) = Menções
- Chamada social para ação(*Social Call to Action*) = Tweets
- Progresso visível/recompensa(*Visible Progress/Reward*) = Seguidores

Em resumo, um iniciante no Twitter decide conectar-se e expressar suas opiniões. Após essa ação, ele pode sair do sistema, mas se alguém o menciona em um tweet (menção), ele se reengaja. Isso leva a responder ao tweet e ganhar seguidores, motivando-o a continuar se conectando e expressando. No entanto, o ciclo varia para um especialista, ou seja, uma pessoa que utiliza o twitter a mais tempo e conhece melhor suas funcionalidades. Usando o mesmo ciclo de engajamento, é ilustrado como um especialista deixa e retorna ao jogo:

- Emoção motivadora(*Motivating Emotion*) = Coletar e classificar
- Reengajamento do usuário(*Player Re-Engagement*) = Tweets e retweets
- Chamada social para ação(*Social Call to Action*) = Seguir retweets
- Progresso visível/recompensa(*Visible Progress/Reward*) = Lista de seguidores

Em resumo, um especialista é motivado pela sua posição no sistema, concentrando-se na quantidade de seguidores e em como se compara a outros jogadores. O reengajamento, além das menções, ocorre pelo interesse em estar em listas de outros jogadores, aparecer em placares (como o Klout) e ter seus tweets retweetados, com uma recompensa visível à medida que a base de seguidores e o status crescem.

3.2.3.5 Feedbacks

A mecânica de feedback desempenha um papel crucial na gamificação, fornecendo aos jogadores informações sobre seu progresso. Amplamente definido como o retorno de informações aos jogadores, o feedback é essencial nos jogos, especialmente na interação entre pontuações e níveis. À medida que as pontuações aumentam, oferecem um feedback claro ao jogador, indicando que ela está progredindo na direção desejada. Níveis e outras mecânicas de progresso complementam esse processo, dividindo a jornada em unidades mais gerenciáveis. O papel crucial do feedback é evidente em áreas desafiadoras, como saúde e educação, onde a falta aparente de progresso pode desmotivar os jogadores.

3.2.3.6 Ganhar Status

O status desempenha um papel significativo em nossas ações, moldando nossa percepção de nós mesmos em relação aos outros. Mesmo entre aqueles que rejeitam ativamente o sistema, como anarquistas, entusiastas do punk rock ou mensageiros de bicicleta, a busca por identidade muitas vezes está vinculada à recusa ao status quo. Esse desejo complexo e onipresente pode ser simplificado como um sistema que determina nossa posição em uma hierarquia.

Ao explorar sistemas gamificados complexos, como programas de milhagem, já discutimos como o status impulsiona comportamentos. Neste contexto, é relevante destacar algumas das mecânicas comuns usadas para facilitar a conquista e promoção de status em ambientes gamificados. Todas essas mecânicas compartilham a eficácia de comunicar status quando são públicas e competitivas.

Exemplos de mecânicas de status incluem distintivos, troféus e níveis. Esses indicadores de progresso não necessitam de comparação direta, contanto que a hierarquia relativa seja compreendida. Itens raros e de edição limitada, como carros especiais ou bens virtuais associados a marcas de luxo, também podem representar poderosos símbolos de status. O acesso prioritário, exemplificado por filas especiais e ambientes virtuais exclusivos, é outra mecânica evidente de status, que, para ser eficaz, geralmente deve ser ativada ou entregue publicamente.

A meta é criar um efeito semelhante à área VIP em uma boate, onde a exclusividade é percebida publicamente, diferentemente de uma sala de jantar privativa em um arranha-céu corporativo. Compreender essas mecânicas é essencial para projetar sistemas gamificados que não apenas engajem, mas explorem de maneira impactante a psicologia envolvida na busca por status.

3.2.4 Conclusão

Ao concluir este capítulo sobre as mecânicas de sistemas gamificados, pode-se refletir sobre a complexidade e versatilidade desses elementos na construção de experiências envolventes. Observa-se como as ferramentas, como pontos, placares, desafios, integração e ciclos de engajamento, são peças fundamentais na criação de dinâmicas que cativam e motivam os participantes, se aplicadas de forma a pensar em seus jogadores.

Através da análise dessas mecânicas, fica claro que a gamificação vai muito além do simples entretenimento, ela representa uma abordagem estratégica para envolver, motivar e reter usuários em diversos contextos, sejam eles educacionais, corporativos ou sociais. As explicações referente aos fundamentos da gamificação proporcionam insights valiosos para o desenvolvimento de sistemas mais eficazes e impactantes, principalmente para esse projeto de fim de curso.

É fundamental reconhecer que, embora tenham sido exploradas apenas algumas mecânicas específicas neste capítulo, existem inúmeras mecânicas que poderiam ser aplicadas no projeto e explicadas nesse capítulo, mas a riqueza e a eficácia de um sistema gamificado reside na combinação inteligente e na adaptação dessas ferramentas e outras ao contexto e aos objetivos específicos do projeto.

4 LIVEMES E GAMIFICAÇÃO

No atual cenário empresarial, a busca incessante por eficiência, produtividade e qualidade é uma constante. Nesse contexto, os Sistemas de Execução de Manufatura (MES) emergem como ferramentas essenciais, desempenhando um papel crucial na gestão e otimização dos processos de produção. Entre esses sistemas, destaca-se o LiveMES, uma solução dinâmica e inovadora que integra tecnologia avançada para oferecer controle em tempo real e insights precisos sobre a operação fabril.

Paralelamente, a gamificação surge como uma estratégia promissora para engajar colaboradores, promover aprendizado e impulsionar resultados dentro das organizações. Ao aplicar mecânicas e elementos de jogos em contextos não lúdicos, como ambientes industriais, a gamificação se revela uma abordagem interessante para estimular o desempenho e a participação dos funcionários.

Neste contexto, este capítulo tem como objetivo explorar a integração entre o LiveMES e a gamificação, delineando os fundamentos teóricos e práticos dessa sinergia. Ao combinar a robustez do LiveMES com os princípios motivacionais da gamificação, busca-se não apenas otimizar os processos de produção, mas também potencializar o engajamento e a satisfação dos colaboradores, promovendo uma cultura de melhoria contínua e excelência operacional.

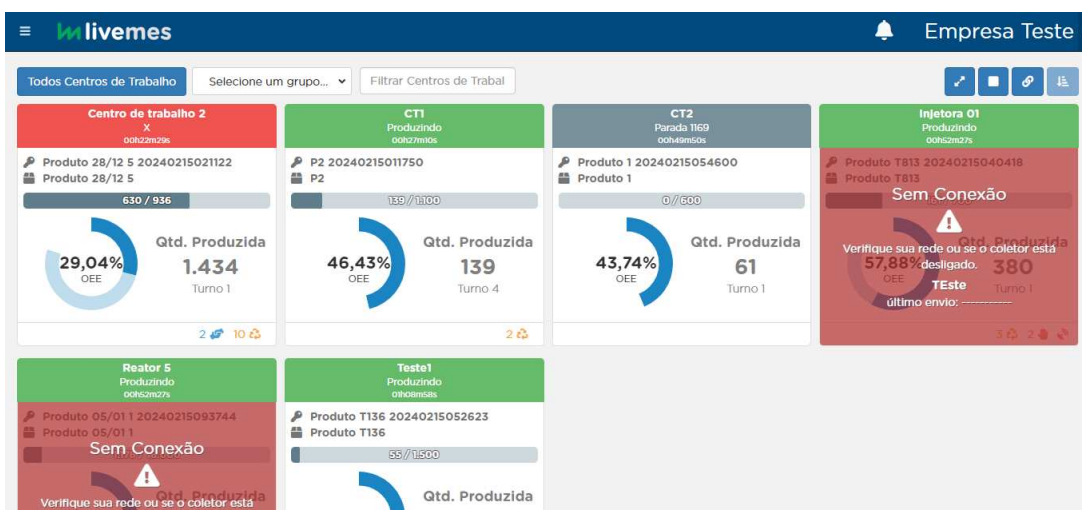
Inicialmente, serão apresentados alguns conceitos essenciais relacionados ao LiveMES, evidenciando sua arquitetura e funcionalidades. A análise desse sistema servirá como base para compreender como a gamificação pode ser integrada de forma sinérgica, agregando valor à operação fabril e estimulando a participação ativa dos colaboradores.

Por fim, serão delineadas as diretrizes e estratégias para a implementação bem-sucedida da gamificação no contexto do LiveMES, considerando os desafios e melhores práticas associadas a essa abordagem inovadora.

4.1 LIVEMES

O LiveMES, como já mencionado, é o sistema MES desenvolvido pela HarboR que apresenta inúmeras funcionalidades como controle de Eficiência Global do Equipamento (OEE, do inglês *Overall Equipment Effectiveness*, é uma métrica fundamental na indústria manufatureira, que avalia a eficiência e a eficácia do equipamento de produção) por máquina/linha de produção, gestão de ordens de produção e previsibilidade sobre finalização das ordens, gráficos analíticos e históricos dos principais indicadores de produção, cadeia de ajuda para chamados de modo geral e avaliação dos atendimentos, entre outras. No entanto, o foco da arquitetura e funcionalidade a ser observada nesse projeto é o dashboard e o cockpit (também conhecido como tela do operador).

Figura 7 – Exemplo de dashboard.



Fonte: App do LiveMES.

Figura 8 – Exemplo de cockpit.



Fonte: App do LiveMES.

Através da Figura 7, é possível observar alguns conceitos iniciais e importantíssimos para o LiveMES. No canto superior direito, é visto o nome da empresa/planta, podendo ser considerada planta pois uma empresa pode ter mais de uma planta no sistema. Por exemplo, uma planta reside em São Paulo e outra em Florianópolis. Já no centro da visualização estão presentes os cards que representam os centros de trabalho e suas informações mais importantes, como OEE e a produção da ordem vigente, para rápida visualização de gerentes e operadores das empresas. Superior a esses cards é tido um seletor de grupo de centro de trabalho para filtrá-los e observar apenas os interessantes ao momento, e bem à direita do seletor, é possível alternar

o tamanho dos cards mantendo as informações originais mas apresentadas de outra forma.

Por outro lado, na Figura 8, observa-se uma visão mais robusta do centro de trabalho, mantendo os números da produção da ordem vigente, além de denominar qual a ordem em produção. No centro, há o gráfico que apresenta a produção atual no turno bem como a funcionalidade de alterar o gráfico e suas variáveis, localizado à direita. Já na parte inferior e na parte superior esquerda e direita são tidas algumas funcionalidades para os operadores, por isso o cockpit também leva o nome de tela do operador. Dentre essas funcionalidades vale destacar o apontamento manual de produção, o botão de “stop” do centro de trabalho que pode parar a produção para uma parada ou manutenção, e quanto à manutenção e parada é possível gerar um chamado pelo operador caso ocorra uma parada com algum problema. No entanto, vale ressaltar que não há nenhuma informação referente aos demais centros de trabalho.

Em síntese, a análise detalhada do Figura 7 e Figura 8 revela a importância e a complexidade do sistema LiveMES na gestão e monitoramento dos processos industriais. Enquanto o Figura 7 oferece uma visão panorâmica e simplificada dos centros de trabalho, permitindo uma rápida visualização das informações essenciais para gestores e operadores, o Figura 8 proporciona uma abordagem mais detalhada e operacional, voltada especialmente para as atividades dos operadores em tempo real. A combinação dessas duas interfaces reflete a abordagem abrangente do LiveMES, fornecendo ferramentas poderosas para otimizar a produção, aumentar a eficiência e garantir a qualidade, enquanto oferece suporte operacional essencial para os funcionários envolvidos nos processos industriais. A integração dessas funcionalidades exemplifica a capacidade do LiveMES de adaptar-se às necessidades específicas de diferentes usuários e contextos de operação, destacando seu papel fundamental na era da Indústria 4.0.

4.2 GAMIFICAÇÃO NO LIVEMES

A partir das explicações sobre o sistema neste capítulo e sobre a gamificação em si no capítulo anterior, o projeto a ser desenvolvido foi pensado pelo graduando e seus diretores. Após várias ideias serem debatidas, foi identificada uma solução para o pontapé inicial da gamificação no sistema: uma tabela de tempos dos centros de trabalho dentro da tela do operador. Essa ideia foi definida visto que abrange vários conceitos da gamificação e, por questão de tempo de desenvolvimento, seria também a mais viável.

A ideia seria algo similar à Figura 9. Portanto, é preciso fazer algumas considerações iniciais. Por exemplo, o cockpit fornece muitas informações vitais aos operadores em tempo real e não podem ser obstruídas. Logo, essa tabela teria de ser pequena e não pode ser ao estilo de modal, que ocupa grande parte da tela. Outro ponto rele-

Figura 9 – Exemplo de Tabela da Fórmula 1.

F1® TEST 2 DAY 2 - CLASSIFICATION		BARCELONA 🇪🇸 #F1Testing			
			Lap Time	Gap	Laps
1	Sebastian Vettel	Ferrari	1:16.841		144
2	Pierre Gasly	AlphaTauri	1:17.066	+0.225	138
3	Lance Stroll	Racing Point	1:17.118	+0.277	130
4	Nicholas Latifi	Williams	1:17.313	+0.472	158
5	Lando Norris	McLaren	1:17.573	+0.732	112
6	Max Verstappen	Red Bull Racing	1:17.738	+0.897	31
7	Valtteri Bottas	Mercedes	1:17.985	+1.144	47
8	Esteban Ocon	Renault	1:18.013	+1.172	37
9	Kevin Magnussen	Haas	1:18.225	+1.384	110
10	Alexander Albon	Red Bull Racing	1:18.393	+1.552	60
11	Daniel Ricciardo	Renault	1:18.395	+1.554	59
12	Antonio Giovinazzi	Alfa Romeo	1:19.670	+2.829	91

Fonte: Pesquisas no google sobre essa tabela.

vante: não faria sentido essa tabela conter todos os centros de trabalho, visto que nem todos operam ao mesmo tempo. Portanto, é preciso separá-los de forma similar ao que existe no dashboard, ou seja, em grupos de centros de trabalho definidos. Assim, o desenho de ideia tido inicialmente é visto em Figura 10, onde é possível ver a tabela pequena em relação ao resto, com um seletor de grupos. O resultado final, assim como os requisitos, serão apresentados nos demais capítulos.

Figura 10 – Exemplo de desenho de solução.



Fonte: Desenho desenvolvido na ferramenta Figma pelo autor.

Dos conceitos da gamificação que valem ser destacados: é preciso conhecer o jogador. Para o LiveMES, é possível identificar ao menos três possíveis jogadores: os operadores, os diretores das plantas e empresas. Por se tratar de um pontapé inicial nesse quesito, a opção escolhida foram os operadores, pois ambos têm acesso ao cockpit e acompanham o centro de trabalho, portanto redefinindo os jogadores para os centros de trabalho. Dessa forma, é possível gerar níveis de socialização entre operadores, gerando uma competição entre eles onde poderão interagir.

Já referente às mecânicas de jogo utilizadas, é visível que uma das que serão utilizadas é o placar/tabela que, a fim de evitar desmotivação ao jogador, será mantida apenas com a visualização de 5 competidores. Este conceito foi obtido através do curso da Universidade da Pensilvânia (COURSERA, s.d.), o qual sugere que, por vezes, placares e tabelas podem ser desmotivadores. Os placares apresentam algumas peculiaridades quanto à sua apresentação aos jogadores e sua motivação, como mencionado no capítulo anterior. No entanto, é a melhor mecânica para conhecer cada vez mais seus jogadores e, por fim, evoluir ainda mais o sistema gamificado, deixando-o mais fluido e bem receptivo ao jogador.

Outros pontos interessantes relacionados aos conceitos de gamificação são ciclo de engajamento e feedbacks, ambos também devem ser compreendidos dentro do placar, a fim de realmente manter os operadores utilizando a funcionalidade, informando seu desempenho ou motivando e recompensando-o como melhor centro de trabalho do turno.

5 METODOLOGIAS E REQUISITOS

No contexto do desenvolvimento de um Projeto de Fim de Curso, a escolha e aplicação de metodologias adequadas desempenham um papel crucial para o sucesso do projeto. Este capítulo visa apresentar as metodologias empregadas durante a condução do presente trabalho, bem como os requisitos que guiaram sua concepção e implementação.

Ao longo deste capítulo, serão exploradas as metodologias utilizadas como estrutura orientadora para a pesquisa, a análise e a implementação do projeto proposto. Essas metodologias forneceram o arcabouço necessário para a realização de um estudo sistemático, bem como para a definição de procedimentos e técnicas que permitiram alcançar os objetivos estabelecidos.

Além disso, serão discutidos os requisitos essenciais que influenciaram a definição e o desenvolvimento do projeto. Esses requisitos, sejam eles funcionais ou não funcionais, delinearam as características e funcionalidades-chave do produto final, atendendo às necessidades identificadas e às expectativas dos potenciais usuários.

Por meio da análise das metodologias adotadas e dos requisitos estabelecidos, este capítulo busca oferecer uma compreensão detalhada do processo de elaboração do PFC, evidenciando as escolhas metodológicas e os critérios que embasaram sua condução.

5.1 METODOLOGIAS

O primeiro passo para o desenvolvimento foi a pesquisa sobre gamificação, que incluiu o estudo e validação do conceito para sua aplicação no sistema. Após concluir essa fase de estudos e validações, deu-se início ao desenvolvimento no sistema com o auxílio de metodologias específicas, que serão mencionadas posteriormente.

É importante ressaltar como foi conduzida a pesquisa. Inicialmente, foram analisados aproximadamente 10 vídeos do TED talk sobre o tema. Os TED talks são eventos globais que reúnem pessoas de diversas áreas, como ciência, tecnologia, entretenimento, design e educação, para compartilhar ideias inspiradoras em apresentações curtas e impactantes. Além disso, foi feita a leitura da obra "Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps", escrita por Gabe Zichermann e Christopher Cunningham. Este guia prático explora como incorporar mecânicas de jogos em aplicativos da web e móveis para melhorar a experiência do usuário e alcançar objetivos de negócios. Também foi realizado um curso sobre Gamificação na plataforma Coursera, oferecido pela Universidade da Pensilvânia.

Por meio desses estudos, foram identificadas diversas abordagens para a aplicação das técnicas de gamificação. Algumas dessas abordagens parecem promissoras, enquanto outras provavelmente não atingirão os resultados desejados. No entanto, o

curso foi particularmente útil nesse aspecto, pois apresentou de forma mais concreta como essas técnicas podem ser aplicadas. Assim, deu-se início ao projeto, fazendo uso de algumas metodologias para projetos.

No processo de desenvolvimento da gamificação, foi adotada uma metodologia ágil denominada Kanban. Essa metodologia é caracterizada por uma abordagem visual na qual as atividades são representadas por “cards” e organizadas em colunas diferentes em um quadro de controle. Cada coluna representa uma etapa distinta do processo de trabalho, como “to test”, “testing”, “rework” e “double check”. Atualmente, o projeto encontra-se em processo de rework, corrigindo código implantado e detalhes visuais.

Essa metodologia ágil foi projetada para facilitar e melhorar o desenvolvimento de projetos, especialmente em equipes de software e desenvolvimento de produtos. Seus objetivos principais são aumentar a eficiência, a transparência, a colaboração e a capacidade de resposta a mudanças no processo de desenvolvimento.

5.2 REQUISITOS E FLUXO DE DADOS

Os requisitos funcionais descrevem as funcionalidades específicas que o sistema deve oferecer, delineando as atividades que ele deve ser capaz de realizar para atender às necessidades dos usuários e aos objetivos do projeto.

Por outro lado, os requisitos não funcionais definem as características de qualidade do sistema, como desempenho, segurança, usabilidade e confiabilidade. Eles descrevem como o sistema deve se comportar em termos de eficiência, escalabilidade e outros atributos importantes que impactam diretamente a experiência do usuário.

Portanto, os requisitos funcionais do projeto são os seguintes:

- O sistema deve ser capaz de buscar todos os grupos de centros de trabalho em que o centro de trabalho do cockpit pertence;
- Dentro de cada centro de trabalho o sistema precisa obter o tempo de trabalho no turno;
- O sistema deve ser capaz de apresentar apenas cinco centros de trabalho na tabela;
- A tabela deve ser pequena e possível de alterá-la de local na tela;
- O sistema deve ser capaz de fornecer feedback, caso haja ultrapassagem do centro de trabalho em destaque de trabalho em relação a outro;
- O sistema deve ser capaz de fornecer feedback, caso possa haver uma ultrapassagem iminente de um centro de trabalho ao centro de trabalho em destaque;

- O sistema deve ser capaz de fornecer feedback, caso o centro de trabalho em destaque seja o melhor colocado ao fim do turno;
- O sistema deve destacar na tabela o centro de trabalho do respectivo cockpit;

Por outro lado, os requisitos não funcionais são os seguintes:

- O sistema deve enviar os dados dos centros de trabalho selecionados num ciclo de no máximo 11 segundos;
- As animações de feedback devem durar menos de 10 segundos;

A partir desses requisitos será exemplificado a seguir o fluxo dos dados. Inicialmente o usuário abrirá um modal para definir o grupo de centros de trabalho. Assim que selecionado, o usuário precisa dar início ao clicar num botão do modal. Dessa forma, o modal é fechado e um overlay representando a tabela é aberto. Esse overlay deverá ficar atualizando os dados de tempo em tempo. A depender dos tempos dos centros de trabalho, a tabela deve ser estendida com o feedback adequado ao momento. Caso o usuário não queira mais ver a tabela, basta fechá-la no botão específico e todo fluxo pode ser reiniciado.

6 DESENVOLVIMENTO DO PROJETO E RESULTADOS

Este capítulo apresenta o desenvolvimento técnico do projeto, focando especificamente na implementação do front-end utilizando Angular. Serão discutidos os aspectos técnicos envolvidos na construção da interface do usuário, destacando a estruturação do código, a arquitetura de componentes, o gerenciamento de estado e a integração com o back-end.

É importante ressaltar que a parte de busca de dados, realizada pelo back-end, já estava pronta e desenvolvida pela equipe correspondente. Portanto, o autor deste trabalho não precisou desenvolver essa funcionalidade, concentrando-se exclusivamente no desenvolvimento do front-end.

Ao longo deste capítulo, serão apresentados os desafios enfrentados durante o processo de desenvolvimento, as soluções adotadas para superá-los e os resultados obtidos. Também será abordada a integração entre o front-end e o back-end, destacando como essa colaboração foi fundamental para o funcionamento adequado do sistema como um todo.

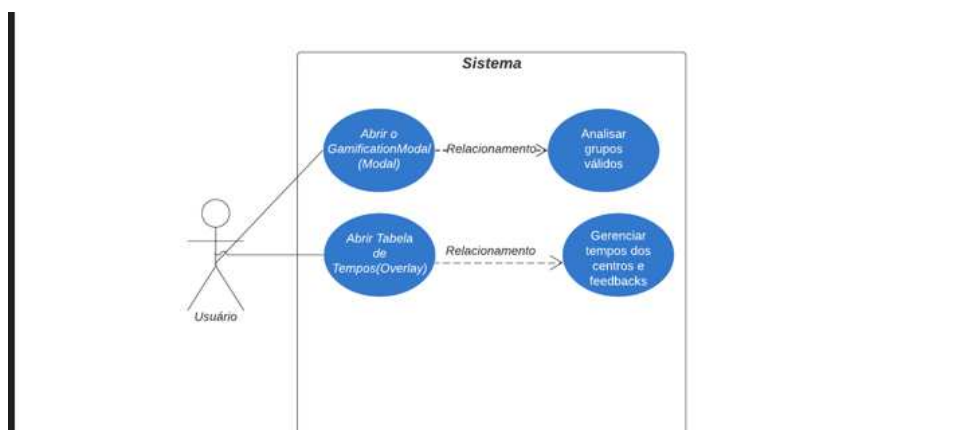
Por meio dessa análise técnica, busca-se oferecer uma visão detalhada do trabalho realizado, demonstrando as habilidades e conhecimentos aplicados na implementação eficaz do front-end do projeto, bem como os resultados alcançados em termos de funcionalidade, desempenho e usabilidade.

6.1 DESENVOLVIMENTO DO PROJETO

Para explicação do desenvolvimento do projeto, vale relembrar o fluxo de dados apresentado no capítulo anterior. Para começar, o usuário abrirá um modal onde poderá selecionar o grupo de centros de trabalho desejado. Após fazer a seleção, ele deve iniciar o processo clicando em um botão dentro do modal. Em seguida, o modal será fechado e um overlay representando a tabela será exibido. Esse overlay será responsável por manter os dados de tempo atualizados em intervalos regulares. A tabela será dinamicamente ajustada de acordo com os tempos registrados nos centros de trabalho, proporcionando feedback em tempo real conforme necessário. Se o usuário decidir não visualizar mais a tabela, poderá simplesmente fechá-la usando o botão específico, reiniciando assim todo o fluxo de interação. A fim de auxiliar a compreensão do fluxo de dados, o autor desenvolveu os diagramas de caso de uso e de sequência presentes nas Figura 11 e Figura 12.

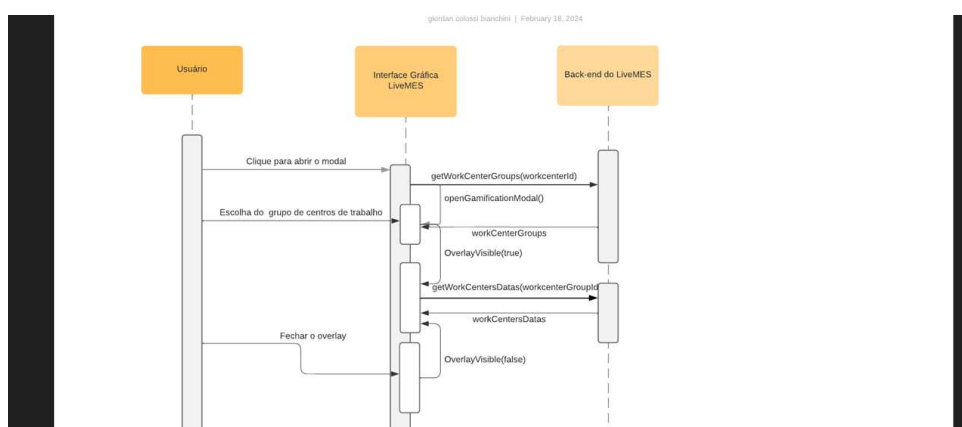
Portanto, partindo do princípio, o usuário precisa abrir um modal para selecionar o grupo dentro do cockpit, conforme ilustrado na Figura 8. Como descrito anteriormente, essa funcionalidade de abrir modais e outras funcionalidades importantes para o operador é encontrada na parte inferior da tela, precisamente na parte inferior direita, onde estão localizados três pontinhos. Ao clicar nesse ícone, um menu reduzido de

Figura 11 – Diagrama de caso de uso.



Fonte: App lucid.

Figura 12 – Diagrama de sequencia.



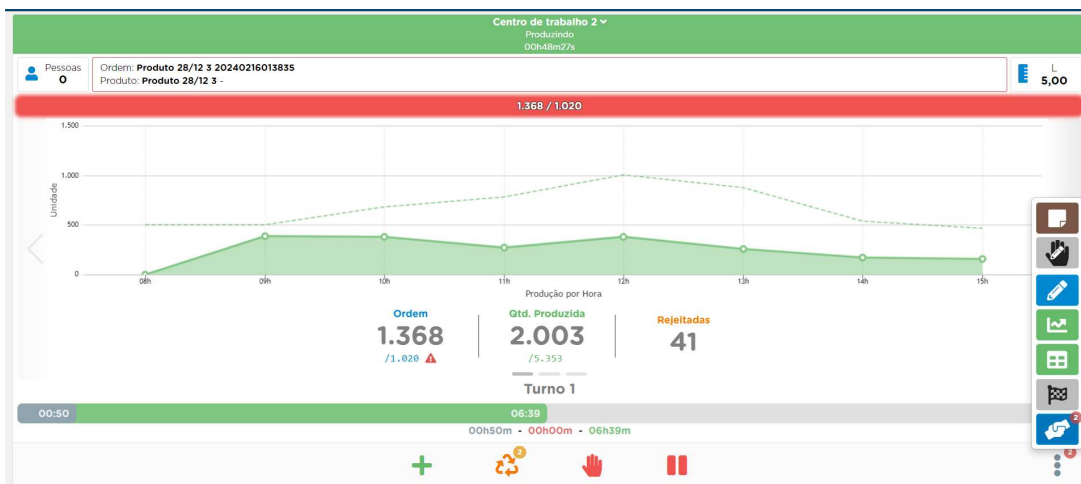
Fonte: App lucid.

opções é exibido, como pode ser visto na Figura 13. Dentre essas opções, já é possível observar o botão para abrir o modal criado, representado pela bandeira de corrida.

Após clicar nesse botão, o modal é aberto, como ilustrado na Figura 14. Nele, é possível observar o seletor de grupos de centro de trabalho e o botão "Vamos Começar!", que fecha o modal e abre o overlay referente à tabela de tempos no cockpit. A Figura 15 apresenta o overlay recém-aberto, que claramente obstrui o gráfico. Assim, o usuário pode simplesmente arrastar o overlay para o local desejado no cockpit, seja usando o mouse ou a tela de toque, como demonstrado na Figura 16.

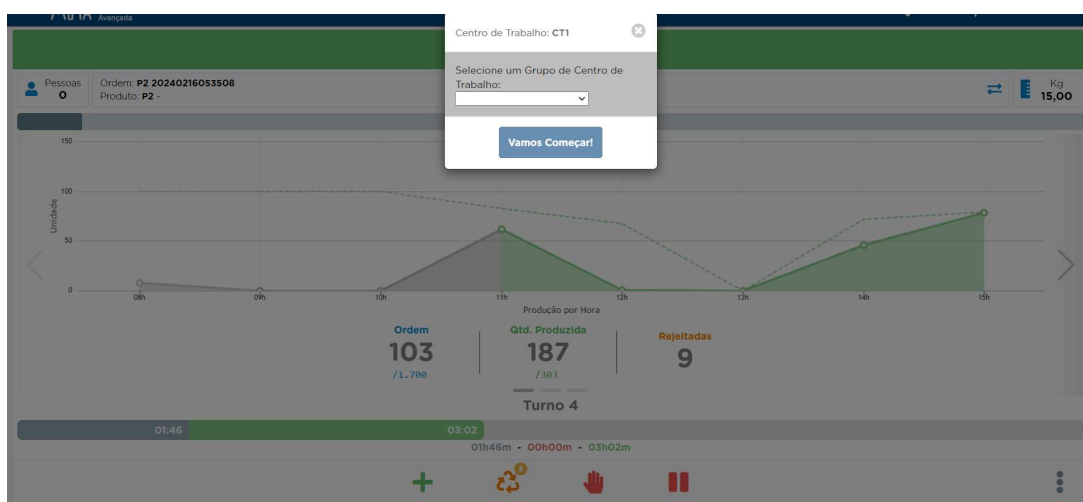
Além da mobilidade do overlay, observa-se o cumprimento de um requisito importante para o sistema em relação à gamificação: apenas cinco competidores são visualizados. Neste caso específico, o grupo da empresa de teste com dados simulados apresenta apenas cinco centros de trabalho no grupo. No entanto, caso

Figura 13 – Cockpit com acesso ao modal adicionado.



Fonte: App do LiveMES.

Figura 14 – Novo modal aberto no cockpit.



Fonte: App do LiveMES.

houvesse um sexto centro de trabalho e o centro de trabalho referente ao cockpit fosse o quinto, a visualização seria do segundo ao sexto. Essa explicação fará mais sentido ao apresentar o código de desenvolvimento da aplicação a seguir, assim como a questão dos feedbacks, que é um dos motivos para o projeto encontrar-se em “rework” no Kanban. Isso ocorre porque eles estão sendo gerados no código, mas é preciso melhorar a forma como são apresentados ao usuário. Para completar o fluxo, no canto superior direito, é disponibilizado o botão para fechar o overlay. Caso o usuário queira trocar de grupo, ele pode simplesmente abrir o modal novamente e refazer os passos.

Na implementação de código utilizando o framework Angular, o desenvolvimento

Figura 15 – Overlay aberto na frente do gráfico.



Fonte: App do LiveMES.

Figura 16 – Overlay deslocado.



Fonte: App do LiveMES.

é realizado em TypeScript, HTML e CSS. O TypeScript é empregado para desenvolver a lógica de funcionamento da tela, como por exemplo, as interações ao clicar em botões. O HTML é utilizado para definir os elementos visuais necessários, como botões e seletores, enquanto o CSS é responsável por estilizar a aparência desses elementos.

No início do fluxo de dados, a primeira etapa foi adicionar o botão ao menu e implementar sua funcionalidade de clique. Para isso, o botão foi incorporado ao componente do cockpit, conforme demonstrado na Figura 17. Esse botão é configurado para executar uma função específica quando clicado, como mostrado no código TypeScript do cockpit na Figura 18. Essa função utiliza um serviço do Bootstrap para abrir modais, denominado "modalService", que se refere ao NgbModal e abre o componente

denominado GamificationModal, como evidenciado na Figura 14.

Figura 17 – Código html do cockpit.

```
467 <button class="lm-btn lm-btn-full lm-btn-grey"  
468     (click)="openGamificationModal()"  
469     title="{{ languageTagService.getTitle('gamificationModal') }}">  
470     <i class="fas fa-2x fa-flag-checkered"></i>  
471 </button>
```

Fonte: VS Code.

Figura 18 – Código typescript do cockpit.

```
300 public openGamificationModal(): void {  
301     const modalRef = this.modalService.open(GamificationModalComponent, { size: 'sm' });  
302     modalRef.componentInstance.workCenterCockpit = this.workCenterCockpitFlat;  
303 }  
304
```

Fonte: VS Code.

Ao abrir o GamificationModal, os textos referentes aos centros de trabalho, ao seletor de grupos e ao botão são exibidos. Todos esses elementos foram implementados de forma simples e podem ser vistos no trecho de código apresentado na Figura 19. O cabeçalho foi construído usando um componente compartilhado por todos os modais, onde é necessário apenas fornecer o nome do centro de trabalho. Abaixo, a variável “selectedGroup” é utilizada em [(ngModel)] para receber o valor do grupo de centros de trabalho, que é apresentado pela variável “workCenterGroups”. Já no “modal-footer”, encontra-se o botão que inicialmente fica desabilitado até que algum grupo seja selecionado. Quando um grupo é selecionado, o botão é habilitado e, ao ser clicado, executa a função “openRaceTable”.

Nas Figura 20, Figura 21, Figura 22 e Figura 23, é apresentada a implementação de um controlador para a visão HTML, começando com as importações necessárias e seguido pela declaração do componente para utilizar o arquivo HTML apresentado na Figura 19. Em seguida, ocorre a declaração da classe GamificationModal e suas variáveis. Após isso, é definido o construtor com os serviços necessários, sendo relevante destacar o “raceTableService”, criado especificamente para o projeto e que será explicado posteriormente. Na função “getWorkCenterGroups”, é realizada a chamada à API que contém as informações dos grupos de centro de trabalho, adicionando apenas os grupos aos quais o centro de trabalho atual faz parte à variável “workCenterGroups”. Por fim, a função “openRaceTable”, mencionada anteriormente, utiliza o serviço desenvolvido para abrir o overlay, passando informações importantes como o ID do centro

Figura 19 – Código html do modal.

```
Go to component
1 <div class="gamification-modal">
2   <div class="modal-header">
3     <app-modal-header [workCenterName]="workCenterCockpit.workCenterName" (closeModal)="activeModal.dismiss()">
4     </app-modal-header>
5   </div>
6
7   <div class="modal-body gamification-body">
8     <span>Selecione um Grupo de Centro de Trabalho:</span>
9     <div class="select-group">
10
11       <select [(ngModel)]="selectedGroup">
12         <option *ngFor="let wcGroup of workCenterGroups" [value]="wcGroup.id">{{wcGroup.workCenterGroupName}}</option>
13       </select>
14
15     </div>
16   </div>
17
18   <div class="modal-footer">
19     <button class="lm-btn button-race" [disabled]="!selectedGroup" (click)="openRaceTable()">Vamos começar!</button>
20   </div>
21
22 </div>
```

Fonte: VS Code.

de trabalho e o grupo selecionado. No método “ngOnInit”, que é chamado quando o modal é aberto pela primeira vez, a função “getWorkCenterGroups” é executada.

Figura 20 – Código typescript do modal parte 1.

```
1 import { Component, OnInit, Input, Inject } from '@angular/core';
2 import moment, { Moment } from 'moment';
3 import { Subject, BehaviorSubject } from 'rxjs';
4 import { takeUntil, tap } from 'rxjs/operators';
5
6 import { NgbActiveModal, NgbModal } from '@ng-bootstrap/ng-bootstrap';
7
8 //models
9 import { IWorkCenterCockpit, IWorkCenterGroup } from 'src/factories/dashboard-data.model';
10
11
12 import { DashboardDataService } from 'src/factories/dashboard-data.service';
13 import { LanguageTagService } from 'src/services/language-tag.service';
14 import { PageSetupService } from 'src/services/page-setup.service';
15 import { FormatterService } from 'src/services/formatter.service';
16 import { RaceTableService } from '../race-table/race-table.service';
17
18
19 @Component({
20   templateUrl: './gamification.modal.html',
21   styleUrls: ['./gamification.modal.scss'],
22 })
23
24 export class GamificationModalComponent implements OnInit{
```

Fonte: VS Code.

Sobre o serviço destacado no último parágrafo, o código referente é visto na Figura 24, o serviço começa com uma série de imports. Entre eles, estão o Injectable, que é um decorador Angular usado para marcar uma classe como um provedor de serviços, e “BehaviorSubject” e “Observable” que são importados de “rxjs”, uma biblioteca

Figura 21 – Código typescript do modal parte 2.

```
23
24 ✓ export class GamificationModalComponent implements OnInit{
25
26     public workCenterGroups: IWorkCenterGroup[];
27     public selectedGroup: number;
28     public loadingScreen: boolean = false;
29
30
31     public selectedDate$: BehaviorSubject<moment.Moment>;
32     private destroy$ = new Subject();
33
34 ✓     constructor(
35         private readonly dashboardDataService: DashboardDataService,
36         public readonly languageTagService: LanguageTagService,
37         public readonly pageSetup: PageSetupService,
38         public readonly activeModal: NgbActiveModal,
39         public readonly modalService: NgbModal,
40         public readonly formatter: FormatterService,
41         private raceTableService: RaceTableService,
42     ){}
43
44     @Input() workCenterCockpit: IWorkCenterCockpit = null;
45
46
```

Fonte: VS Code.

Figura 22 – Código typescript do modal parte 3.

```
48     private getWorkCenterGroups(): void {
49         this.loadingScreen = true;
50         const workcenterId = this.workCenterCockpit.workCenterId;
51         this.dashboardDataService.getWorkCenterGroups()
52             .pipe(takeUntil(this.destroy$))
53             .subscribe(data => {
54                 const permittedGroups = data.filter(wcGroups => wcGroups.linkedWorkCenters.some(wc => wc.workCenterId ===
55                     workcenterId));
56                 this.workCenterGroups = permittedGroups.sort((a, b) => a.workCenterGroupName.localeCompare(b.workCenterGroupName));
57                 console.log(this.workCenterGroups);
58                 this.loadingScreen = false;
59             });
60
61     public openRaceTable(): void {
62         this.raceTableService.toggleOverlay(this.workCenterCockpit.workCenterId, this.selectedGroup);
63         this.activeModal.dismiss();
64     }
65
66     public cancel(): void{
67         this.activeModal.dismiss();
68     }

```

Fonte: VS Code.

usada para programação reativa em JavaScript.

Em seguida, temos a declaração da classe “RaceTableService”. Esta classe é decorada com “@Injectable”, indicando que é um serviço Angular injetável. Ela possui várias propriedades e métodos.

Uma das propriedades é “overlayVisible”, que é um “BehaviorSubject<boolean>”. Esta propriedade será usada para controlar a visibilidade de um overlay na interface do usuário. Também há uma propriedade pública chamada “overlayVisible\$”, que expõe

Figura 23 – Código typescript do modal parte 4.

```
69
70     ngOnInit(): void{
71         |   this.getWorkCenterGroups();
72         | }
73     ngOnDestroy(): void{
74         |   this.destroy$.next();
75         |   this.destroy$.complete();
76         | }
77 }
78
```

Fonte: VS Code.

Figura 24 – Código typescript do serviço.

```
1 import { Injectable } from '@angular/core';
2 import { BehaviorSubject, Observable } from 'rxjs';
3
4 @Injectable({
5   | providedIn: 'root'
6   | })
7 export class RaceTableService {
8   | private overlayVisible = new BehaviorSubject<boolean>(false);
9   | overlayVisible$ = this.overlayVisible.asObservable();
10  | public workCenterGroupId: number;
11  | public workCenterId: number;
12  |
13  | toggleOverlay(workcenterId: number, workCenterGroup:number) {
14  |   | this.workCenterGroupId = workCenterGroup;
15  |   | this.workCenterId = workcenterId;
16  |   | this.overlayVisible.next(true);
17  |   | }
18  |
19  | closeOverlay() {
20  |   | this.overlayVisible.next(false);
21  |   | }
22  | }
23
```

Fonte: VS Code.

um “observable” para permitir que outros componentes ou serviços se inscrevam e sejam notificados sobre alterações na visibilidade do overlay.

Além disso, existem duas variáveis públicas, “workCenterGroupId” e “workCenterId”, que armazenam identificadores relacionados ao centro de trabalho.

O método “toggleOverlay” é responsável por alternar a visibilidade do overlay. Ele recebe dois parâmetros, “workcenterId” e “workCenterGroup”, que são usados para

atualizar as variáveis “workCenterGroupld e workCenterId”. Em seguida, ele emite “true” através do “BehaviorSubject”, indicando que o overlay deve ser exibido.

Por fim, o método “closeOverlay” fecha o overlay, emitindo “false” através do “BehaviorSubject”, indicando que o overlay deve ser ocultado.

Nas Figura 25, Figura 26 e Figura 27 é apresentado o código referente ao overlay controlado pelo serviço “RaceTableService”, garantindo que ele seja exibido apenas quando essa propriedade é definida como verdadeira. Dentro do overlay, há uma tabela que exibe dados relacionados aos centros de trabalho. Esses dados são iterados por meio de um loop “*ngFor”, permitindo que informações como nome, posição e tempo de produção de cada centro de trabalho sejam exibidas de forma dinâmica.

Figura 25 – Código html do overlay parte 1.

```
Go to component
1 <div *ngIf="overlayVisible" class="overlay" cdkDrag>
2   <div>
3     <button type="button" class="close" (click)="clickOutsideOverlay()">
4       <span aria-hidden="true">
5         <i class="fas fa-times-circle"></i>
6       </span>
7     </button>
8     <table>
9       <thead>
10        <tr>
11          <th>CT</th>
12          <th class="time">Tempo(hh:mm:ss)</th>
13        </tr>
14      </thead>
15      <tbody>
16        <tr [ngClass]="{'identity': wc.workCenterId == workCenterId}" *ngFor="let wc of dataTable">
17          <td>{{wc.position}} - {{ wc.workCenterName }}</td>
18          <td class="time">{{ wc.shiftProduction.runTime | secondsToFormat:'HH:mm:ss' || '--:--:--'}}</td>
19        </tr>
20      </tbody>
21    </table>
22  </div>
23
```

Fonte: VS Code.

Além da tabela, há uma seção que exibe mensagens de feedback, que são exibidas condicionalmente com base em mensagens recebidas do serviço. Essas mensagens podem incluir informações sobre ultrapassagem de produção, melhorias de desempenho ou advertências. A aparência visual das mensagens de feedback é controlada por classes CSS como “scene” e “expanded”, que são aplicadas de forma condicional para garantir uma exibição adequada e animações quando necessário.

Cada mensagem de feedback é acompanhada por uma representação visual, como emojis de carros ou medalhas, que servem como indicadores visuais do conteúdo da mensagem. O texto real da mensagem é exibido dentro de uma <div> específica, identificada pela classe “message”.

Finalmente, os botões dentro do overlay, como o botão de fechar, fornecem meios de interação com o overlay, permitindo que os usuários realizem ações como fechar o overlay ou executar outras operações relacionadas. Esses botões normalmente

Figura 26 – Código html do overlay parte 2.

```
23 <div *ngIf="feedbackMessage.ultrapassing !== ''" class="scene" [ngClass]="{'expanded': isExpanded}">
24 <div class="car car1">
25 | 🏎️
26 </div>
27 <div class="car car2">
28 | 🏎️
29 </div>
30 <div class="message">{{feedbackMessage.ultrapassing}}</div>
31 </div>
32
33 <div *ngIf="feedbackMessage.best !== ''" class="scene" [ngClass]="{'expanded': isExpanded}">
34 <div class="medalha">
35 <div class="circulo"></div>
36 <div class="ouro">
37 <span class="numero">1</span>
38 </div>
39 </div>
40 <div class="message">{{feedbackMessage.best}}</div>
41 </div>
42
```

Fonte: VS Code.

Figura 27 – Código html do overlay parte 3.

```
43 <div *ngIf="feedbackMessage.warning !== ''" class="scene" [ngClass]="{'expanded': isExpanded}">
44 <div class="car car1">
45 | 🏎️
46 </div>
47 <div class="car car3">
48 | 🏎️
49 </div>
50 <div class="message">{{feedbackMessage.warning}}</div>
51 </div>
52 </div>
53 </div>
54
```

Fonte: VS Code.

estão associados a métodos definidos nos componentes Angular correspondentes, permitindo que as interações do usuário sejam tratadas de forma adequada.

Nas Figura 28, Figura 29, Figura 30, Figura 31, Figura 32, Figura 33, Figura 34 e Figura 35 é visto o código referente ao “RaceTableComponent” que utiliza o “RaceTableService”, que desempenha um papel fundamental na exibição dinâmica de uma tabela na interface do usuário. Ao longo do código, várias propriedades e métodos são definidos para gerenciar diferentes aspectos da funcionalidade deste componente.

Inicialmente, o construtor do componente é configurado para injetar vários serviços importantes, incluindo “DashboardService”, “LanguageTagService”, “FormatterService” e “RaceTableService”, que são utilizados para acessar dados e funcionalidades específicas da aplicação.

Uma das principais funções deste componente é ordenar e exibir os dados da tabela de forma adequada. O método “sortingFivePlayers” é responsável por classificar

Figura 28 – Código typescript do overlay parte 1.

```
1 import { Component, OnInit, Input, Inject, ElementRef, HostListener } from '@angular/core';
2 import moment, { Moment } from 'moment';
3 import { Subject, BehaviorSubject, forkJoin, interval, timer } from 'rxjs';
4 import { switchMap, takeUntil, tap } from 'rxjs/operators';
5 import { LanguageTagService } from 'src/services/language-tag.service';
6 import { FormatterService } from 'src/services/formatter.service';
7 import { RaceTableService } from './race-table.service';
8 import { DashboardService } from '../../services/dashboard.service';
9
10
11
12 @Component({
13   selector: 'app-race-table',
14   templateUrl: './race-table.component.html',
15   styleUrls: ['./race-table.component.scss'],
16 })
17
18 export class RaceTableComponent implements OnInit{
19
20   public loadingScreen: boolean = false;
21   overlayVisible: boolean = false;
22   workCenterId: number = null;
23   dataTable: any;
24   oldTable: any;
25   workCenterGroupId: number[] = [];
26   isExpanded: boolean = false;
```

Fonte: VS Code.

Figura 29 – Código typescript do overlay parte 2.

```
27   feedbackMessage = {
28     best: '',
29     ultrapassing: '',
30     warning: ''
31   };
32
33
34   private destroy$ = new Subject();
35
36   constructor(
37     private readonly dashboardService: DashboardService,
38     public readonly languageTagService: LanguageTagService,
39     public readonly formatter: FormatterService,
40     public raceTableService: RaceTableService,
41   ){}
42
```

Fonte: VS Code.

os dados recebidos, garantindo que o centro de trabalho atual esteja no centro da visualização e selecionando cinco centros de trabalho para exibição.

Além disso, o componente lida com eventos do usuário, como o clique fora do overlay que levam a funcionalidades fora do cockpit, para fechá-lo quando necessário. O método “clickOutsideOverlay” é invocado para essa finalidade.

Figura 30 – Código typescript do overlay parte 3.

```
43
44   sortingFivePlayers(dataTable: any, workCenterId: number): void{
45       this.dataTable = [];
46       this.feedbackMessage.best = '';
47       this.feedbackMessage.warning = '';
48       this.feedbackMessage.ultrapassing = '';
49       let position = 0;
50
51       dataTable.sort((a,b) => b.shiftProduction.runTime - a.shiftProduction.runTime);
52       dataTable.forEach(wc => {
53           position++;
54           wc.position = position;
55       });
56
57       const wcReference = dataTable.findIndex(wc => wc.workCenterId === workCenterId);
58
59       let beginTable = Math.max(0, wcReference - 2);
60       let endTable = Math.min(dataTable.length - 1, wcReference + 2);
61
```

Fonte: VS Code.

Figura 31 – Código typescript do overlay parte 4.

```
61
62   if (endTable - beginTable + 1 < 5) {
63       let missingCompetitors = 5 - (endTable - beginTable + 1); // Número de competidores faltando para atingir 5
64
65       // Expandir a janela de visualização para incluir mais competidores, se possível
66       if (beginTable > 0) {
67           // Se houver espaço para expandir para cima, expandimos para cima
68           const expandUpwards = Math.min(beginTable, missingCompetitors);
69           beginTable -= expandUpwards;
70           missingCompetitors -= expandUpwards;
71       }
72
73       // Expandir para baixo, se ainda houver competidores faltando
74       if (missingCompetitors > 0 && endTable < dataTable.length - 1) {
75           endTable = Math.min(dataTable.length - 1, endTable + missingCompetitors);
76       }
77   }
```

Fonte: VS Code.

Uma parte importante do componente é a geração de feedback dinâmico sobre o desempenho dos centros de trabalho. O método “feedbacks” analisa os dados da tabela e fornece feedback contextual sobre o desempenho atual, destacando áreas de excelência ou preocupações.

No método “ngOnInit”, o componente se inscreve para observar mudanças na visibilidade do overlay por meio do serviço “RaceTableService”. Quando o overlay é exibido, o componente busca dados relevantes por meio do serviço “DashboardService” para atualizar a tabela de acordo, ou seja, busca os dados referentes aos tempos dos centros de trabalho.

Por fim, nas Figura 36, Figura 37, Figura 38 e Figura 39, é apresentada uma parte do código CSS do overlay, referente à animação desenvolvida para o feedback. É importante ressaltar que, devido a esta parte, o projeto encontra-se em “rework” e provavelmente será alterado. No entanto, vale destacar que a funcionalidade estava

Figura 32 – Código typescript do overlay parte 5.

```
78
79     this.dataTable = dataTable.slice(beginTable, endTable + 1);
80     this.feedbacks(this.dataTable, workCenterId, this.oldTable);
81     if(this.overlayVisible === true) {
82         |   this.oldTable = this.dataTable;
83     }
84     if(this.overlayVisible === false) {
85         |   this.oldTable = [];
86     }
87 }
88
89 clickOutsideOverlay() {
90     |   this.raceTableService.closeOverlay();
91 }
92
```

Fonte: VS Code.

Figura 33 – Código typescript do overlay parte 6.

```
93     feedbacks(dataTable: any, workCenterId: number, oldTable?: any): void {
94         |   let workcenter = dataTable.filter(wc => wc.workCenterId == workCenterId);
95         |   let oldWorkcenter = oldTable?.filter(wc => wc.workCenterId == workCenterId);
96         |   let actualTime = moment();
97         |   let shiftEnd = moment(workcenter[0].shiftProduction?.shiftEndTime);
98         |   let differenceInSeconds = shiftEnd.diff(actualTime, 'seconds');
99         |   let rivalUnder = dataTable.filter(wc => wc.position === workcenter[0].position+1);
100        |   let differenceRival = workcenter[0].shiftProduction?.runtime - rivalUnder[0].shiftProduction?.runtime;
101        |   if((differenceInSeconds < 11) && (workcenter[0].position === 1)) {
102            |   |   this.showFeedback('best', `Parabéns, ${workcenter.workCenterName} foi o melhor do turno!`);
103            |   }
104            |   if((differenceRival < 11) && (workcenter[0].currentState.stateReasonTypeCode !== "Uptime")) {
105                |   |   this.showFeedback('warning', `Vamos voltar ao trabalho, estamos quase sendo ultrapassados!`);
106            |   }
107            |   if((oldWorkcenter !== undefined) && (workcenter[0].position < oldWorkcenter[0].position)) {
108                |   |   this.showFeedback('ultrapassing', `Bela ultrapassagem, continue o bom trabalho!`);
109            |   }
110        }
111    }
```

Fonte: VS Code.

operacional, respeitando um dos requisitos atribuídos quanto ao tempo de exibição.

6.2 RESULTADOS

Após todo esse desenvolvimento, embora o projeto não tenha alcançado uma conclusão definitiva em relação ao problema que se propõe a resolver, ainda é possível considerar o resultado atual como positivo se comparado aos objetivos determinados na introdução. Isso se deve também ao cumprimento de todos os requisitos propostos para o sistema e à quase completa funcionalidade do mesmo.

Figura 34 – Código typescript do overlay parte 7.

```
112     private showFeedback(type: string, message: string): void {
113         this.isExpanded = true;
114         this.feedbackMessage[type] = message;
115         setTimeout(() => {
116             this.isExpanded = false;
117         }, 10000);
118     }
119
```

Fonte: VS Code.

Figura 35 – Código typescript do overlay parte 8.

```
120     ngOnInit(): void{
121         this.clickoutsideOverlay();
122         this.workCenterGroupId = [];
123         this.workCenterId = null;
124         this.raceTableService.overlayVisible$.subscribe(visible => {
125             this.workCenterGroupId.pop();
126             this.workCenterId = this.raceTableService.workCenterId;
127             this.overlayVisible = visible;
128             this.workCenterGroupId.push(this.raceTableService.workCenterId);
129             if(this.overlayVisible === true) {
130                 timer(0, 11000)
131                     .pipe(
132                         takeUntil(this.destroy$),
133                         switchMap(() => this.dashboardService.getIndustryDashboardData(this.workCenterGroupId))
134                     )
135                     .subscribe(wcGroupData => {
136                         this.dataTable = wcGroupData.map(group => group.workCenters).reduce((acc, curr) => acc.concat(curr), []);
137                         this.sortingFivePlayers(this.dataTable, this.workCenterId);
138                     });
139             }
140         });
141     }
```

Fonte: VS Code.

Figura 36 – Código CSS do overlay parte 1.

```
68 .scene {
69     width: 100%;
70     height: 0;
71     position: relative;
72     overflow: hidden;
73 }
74
75 .scene.expanded {
76     height: 200px; /* Altura expandida da div */
77 }
78
79 .car {
80     font-size: 50px;
81     position: absolute;
82     bottom: 20px;
83 }
84
85 .car1 {
86     left: -80px; /* Posição inicial do carro 1 */
87     transform: scaleX(-1); /* Inverte o carro 1 horizontalmente */
88     animation: moveCar1 8s linear forwards; /* Animação do carro 1 se movendo */
89     color: #286090;
90 }
```

Fonte: VS Code.

Figura 37 – Código CSS do overlay parte 2.

```
91
92 .car2 {
93     left: -100px; /* Posição inicial do carro 2 */
94     transform: scaleX(-1);
95     animation: moveCar2 10s linear forwards; /* Animação do carro 2 se movendo */
96 }
97
98 .car3 {
99     left: -80px; /* Posição inicial do carro 1 */
100    transform: scaleX(-1); /* Inverte o carro 1 horizontalmente */
101    animation: moveCar1 7s linear forwards; /* Animação do carro 1 se movendo */
102 }
103
104 .message {
105     position: absolute;
106     bottom: 70px; /* Posição ajustada */
107     left: 50%;
108     transform: translateX(-50%);
109     font-size: 16px;
110     color: #fff;
111     animation: showMessage 6s linear forwards; /* Animação da mensagem */
112     opacity: 0; /* Inicialmente transparente */
113 }
114
```

Fonte: VS Code.

Figura 38 – Código CSS do overlay parte 3.

```
114
115 @keyframes moveCar1 {
116   0% { left: -80px; } /* Posição inicial do carro 1 */
117   100% { left: calc(100% + 200px); } /* Posição final do carro 1 (sair da tela) */
118 }
119
120 @keyframes moveCar3 {
121   0% { left: -80px; } /* Posição inicial do carro 1 */
122   100% { left: calc(100% + 200px); } /* Posição final do carro 1 (sair da tela) */
123 }
124
125 @keyframes moveCar2 {
126   0% { left: -100px; } /* Posição inicial do carro 2 */
127   100% { left: calc(100% + 200px); } /* Posição final do carro 2 (sair da tela) */
128 }
129
130 @keyframes showMessage {
131   0% { opacity: 0; } /* Inicialmente transparente */
132   90% { opacity: 0; } /* Mantém transparente após a passagem dos carros */
133   100% { opacity: 1; } /* Torna-se visível após a passagem dos carros */
134 }
135
```

Fonte: VS Code.

Figura 39 – Código CSS do overlay parte 4.

```
135
136 √ .medalha {
137   position: relative;
138   width: 100px;
139   height: 100px;
140   animation: showMedal 6s linear forwards;
141 }
142
143 √ @keyframes showMedal {
144   0% { opacity: 1; }
145   90% { opacity: 0; }
146   100% { opacity: 0; }
147 }
148
```

Fonte: VS Code.

7 CONCLUSÃO

Este capítulo marca o encerramento do Projeto de fim de curso, representando o ápice de um esforço dedicado à pesquisa, análise e desenvolvimento de um projeto de relevância tanto acadêmica quanto organizacional. Retomando o problema abordado por este projeto, é notório que empresas e indústrias buscam incessantemente manter sua competitividade no mercado atual. Nesse contexto, os sistemas MES emergem como ferramentas essenciais, fornecendo uma miríade de informações e insights valiosos capazes de impulsionar a produtividade, reduzir desperdícios e otimizar o tempo de operação das máquinas.

Contudo, apesar do potencial desses sistemas em prover dados e funcionalidades úteis, muitas empresas enfrentam dificuldades em utilizá-los plenamente. Muitos usuários desconhecem completamente as capacidades da ferramenta, enquanto outros, embora estejam cientes, não se engajam o suficiente para explorar seu potencial máximo. Identificando essa lacuna, a empresa HarboR, em colaboração com o autor deste trabalho, decidiu buscar uma solução. A gamificação do LiveMES surgiu como uma possível abordagem para essa questão, com o autor encarregado de pesquisar sobre o assunto e desenvolver uma solução tangível.

Após uma análise detalhada, que incluiu pesquisas em cursos online, vídeos e livros sobre gamificação, foi concebido um projeto inicial que foi submetido à avaliação dos diretores da HarboR. Esse processo demandou tempo, exigindo a elaboração de mais de um modelo até a aprovação final e o início do desenvolvimento. Durante essa fase, a metodologia ágil do Kanban foi adotada para gerenciar o fluxo de trabalho, e atualmente, o projeto está passando por uma fase de “rework” para correções pontuais.

Embora ainda não tenhamos tido a oportunidade de verificar os resultados na prática, espera-se que a implementação da gamificação LiveMES resulte em um aumento no engajamento dos operadores das empresas. Isso, por sua vez, deve traduzir-se em uma maior disponibilidade dos centros de trabalho, potencialmente impulsionando a produtividade e o desempenho geral.

Concluindo, mesmo que um resultado final conclusivo não tenha sido alcançado, ainda há uma grande expectativa de que o método e a solução funcionem conforme o esperado. Além disso, existe um grande potencial para incluir outros elementos de gamificação dentro do sistema.

Partindo do princípio de conhecer o jogador, ao contrário da solução desenvolvida neste projeto, os jogadores desse potencial sistema gamificado não seriam apenas os operadores, mas sim as próprias empresas. Em vez de uma competição interna em cada empresa, a competição será entre todas as empresas do LiveMES através de ranking de pontos entre elas. Onde, suas pontuações serão atribuídas a partir de diferentes ações dentro do sistema. Ações as quais otimizam a produção das

empresas e potencializam o uso do LiveMES, como uma troca de ordem em seguida do encerramento da ordem que estava vigente, evitando perda de tempo como acontece. Além disso, para essa ideia ter sucesso possivelmente será necessário reiniciar o ranqueamento a cada mês, visto que se for desenvolvido uma tabela infinita algumas empresas irão se destacar muito em relação a outras, e poderá causar desmotivação nessas com menor pontuação.

Portanto, seguindo essa ideia de reinicialização dos pontos, todos poderão competir igualmente. Para finalizar, essa abordagem poderá incluir diversos elementos de jogos, como tabelas e pontuações, mas também outros, como ganho de status e fidelidade através da disponibilização de recompensas como medalhas e status ao fim de cada mês, por exemplo, a empresa terminou na primeira colocação então é denominada empresa de ouro, semelhantes às milhas de sistemas de viagens de avião. Resultando, ao final do ano, em uma compensação financeira à uma empresa específica com melhor desempenho durante o ano, no entanto é preciso ter muito cuidado quanto a isso como descrito nos capítulos iniciais.

REFERÊNCIAS

COURSERA. **Gamification Course**. [S.l.: s.n.].

<https://www.coursera.org/learn/gamification>.

HARBOR. **HarboR - Informática Industrial**. [S.l.: s.n.].

<https://www.harbor.com.br/>.

LIVEMES. **LiveMES, uma solução HarboR**. [S.l.: s.n.].

<https://www.livemes.com/sobre/>.

ZICHERMANN, Gabe; CUNNINGHAM, Christopher. **Gamification by Design**. [S.l.]: O'Reilly, 2011.