

ALEX FIGUR CATANI

Desenvolvimento de uma solução para informatizar a exibição de indicadores em um ambiente fabril.

Coronel Vivida, Paraná

Março de 2022

Resumo

A gestão a vista é uma ferramenta de gestão que se baseia na visualização de dados de indicadores para tomadas de decisões e é crucial que esses dados sejam exibidos em painéis de controle práticos para que possam ser facilmente enxergados por todos os colaboradores. Essa ferramenta permite que os gestores tenham consciência dos resultados, para que possam tomar decisões mais assertivas. Além disso, a visualização dos indicadores ajuda a manter os colaboradores motivados, uma vez que eles podem ver os resultados de seu trabalho. A gestão a vista também facilita a comunicação entre as áreas da empresa, pois todos os membros da organização terão acesso aos mesmos dados. Visando esse contexto este trabalho de conclusão de curso, apresenta um sistema para a exibição de indicadores em um ambiente fabril onde o projeto foi dividido em 4 partes: equipamentos de hardware para TVs, um sistema web, uma solução para atualizar os arquivos no servidor web e uma solução de automação para gerar e atualizar os dados dos indicadores. A metodologia RUP foi escolhida para o desenvolvimento do projeto, pois apresentou a melhor estruturação do processo de desenvolvimento para esse projeto. O resultado foi satisfatório com um sistema capaz de exibir os indicadores de produtividade desejados de forma eficiente e simples, que além de contribuir para a percepção do que acontece na fábrica, ainda possibilita que melhorias contínuas sejam realizadas para aprimorar os processos.

Palavra-Chave: projeto de software; sistema web; gestão a vista.

1 INTRODUÇÃO

A gestão a vista é uma abordagem para a gestão de dados que facilita o acesso, o compartilhamento e o uso de informações entre os diferentes departamentos da empresa. Os benefícios da gestão a vista incluem maior eficiência, redução de custos, maior produtividade dos funcionários e melhora a tomada de decisão. Além disso, a gestão a vista permite que os profissionais de negócios compreendam melhor os dados e usem-nos para tomar decisões informadas.

Este trabalho de conclusão de curso, apresenta a elaboração de um projeto de software para solucionar um problema existente na área de manufatura de uma empresa. O projeto consiste em uma solução não comercial para monitoramento de indicadores de produção em uma fábrica de produtos da linha branca. A solução

possui como objetivo permitir que os funcionários da fábrica possam acompanhar o desempenho da produção em tempo real em diversos pontos da fábrica.

Para o desenvolvimento e gerenciamento será utilizado a metodologia RUP (*Rational Unified Process*) que possui diversas ferramentas para facilitar o gerenciamento do projeto. Além de inclui princípios e ferramentas baseadas em vários modelos de ciclo de vida: incremental, iterativo, evolutivo e estágio-gates. Esta abordagem permite que os gerentes usem múltiplos ciclos de vida de acordo com a complexidade e as necessidades do projeto.

2 ESPECIFICAÇÕES INICIAIS DO SOFTWARE

Para esse desenvolvimento será preciso resolver vários problemas impostos pela empresa que entre eles estão: baixo orçamento, não possuir equipe especializada em desenvolvimento de *software*, flexibilidade no uso de tecnologia de *hardware*, solução com baixa manutenção e a automação dos processos atuais de atualização das informações produtivas.

2.1 Escopo do Produto

A solução foi dividida em 4 partes que estão definidas da seguinte maneira:

- Equipamento de Hardware para TVs.
- Sistema Web.
- Solução para atualização os arquivos a serem exibidos no servidor web.
- Solução de automação para gerar arquivos e dados para publicação.

2.2 Funcionalidade do Produto

Apresentar indicadores de produtividade da fábrica em TVs através de um navegador web, também sendo possível o acesso através do navegador web nos desktops que tiverem acesso a rede local (intranet) da empresa.

2.3 Ambiente Operacional e Tecnologias

Para a funcionalidade do sistema foi preciso dividir a solução em 4 partes: dispositivo de *hardware* para TVs, servidor *web*, servidor de arquivo, servidor para automação.

2.3.1 Dispositivo para TVs

Optei pelo dispositivo “*Raspberry Pi 3 Modelo B+*” com processador Cortex-A53 (ARMv8) SoC de 64 bits com 1.4GHz, SDRAM LPDDR2 LPDDR2 de 1 GB, HDMI em tamanho real, 4 portas USB 2.0, porta Micro SD para carregar o sistema operacional e armazenar dados, entrada de alimentação DC 5V/3A, SD classe 10 de 16 GB, sistema Operacional “*Raspbian*” que é um distribuição baseada no Debian, possui uma licença Open-Source e foi otimizada para o hardware do “*Raspberry Pi*”, “Script de inicialização” que foi desenvolvido pelo autor no projeto. Especificações mais detalhadas do equipamento podem ser encontradas no site oficial do fabricante através do link “<https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus>”.

2.3.2 Servidor Web

VM (computador virtualizado) com as seguintes configurações: Arquitetura 64 bits, 2 CPUs, 4 GB de RAM, 50 GB de espaço em disco, Sistema Operacional Ubuntu 16, Serviço HTTP Apache, Serviço FTP Apache.

2.3.3 Servidor de arquivo.

O servidor de arquivos é utilizado para a atualização dos arquivos a serem exibidos no servidor web, pode ser implementado em qualquer ambiente que tenha permissão de compartilhar arquivos. É preciso fazer a liberação de segurança do protocolo de comunicação FTP no servidor e a instalação do programa “Cobian Backup” versão 11 como serviço.

2.3.4 Servidor para automação

VM (computador virtualizado) com 6 CPUs, 4 GB de RAM, 60 GB de espaço em disco, Sistema Operacional Windows 7 Professional, Microsoft Office 2016 Pro.

Foi identificado que a Microsoft restringe a instalação do pacote MS Office versão Pro no Sistema Operacional Windows com licença “Server”, que é destinada a servidores, por esse motivo foi preciso utilizar a licença do Windows 7 Profissional que é destinada a Desktop.

3 METODOLOGIA

Para o desenvolvimento da solução foi utilizada a metodologia RUP (Rational Unified Process), pois apresenta uma boa estruturação do processo de desenvolvimento, bem como diversas ferramentas que facilitam o gerenciamento do projeto.

Para Pressman o Processo Unificado (UP) é:

[...] uma tentativa de aproveitar os melhores recursos e características dos modelos tradicionais de processo de software, mas caracterizando-os de modo a implementar muitos dos melhores princípios do desenvolvimento ágil de software (PRESSMAN, 2016, pg. 56).

O projeto foi dividido em quatro fases principais: Início, Elaboração, Construção e Transição. Cada uma dessas fases foi dividida em diversas atividades, que foram realizadas de acordo com o planejamento elaborado no início do projeto.

Na fase de Início, foram:

- Realizadas as atividades de planejamento do projeto, bem como a elaboração do protótipo do sistema.
- Identificados requisitos funcionais, não funcionais e de domínio.
- Identificados recursos disponíveis, principais riscos e limitações de infraestrutura.
- Definido cronograma com base nos dados levantados.

Na fase de Elaboração:

- Foram realizadas as atividades de análise e design do sistema.
- Criado modelos de casos de usos

Na fase de Construção:

- Foram realizadas as atividades de desenvolvimento, testes e implantação do sistema.

Na fase de Transição:

- Foram realizadas as atividades de treinamento, acompanhamento de usabilidade e suporte ao sistema.

4 DESENVOLVIMENTO

4.1 Requisitos de Produto de Software

4.1.1 Requisitos Funcionais

- RF001 - O sistema deve permitir acesso aos dados da fábrica através de um navegador web.
- RF002 - O sistema deve permitir que os dados sejam visualizados em tempo real.
- RF003 - O sistema deve permitir que os dados sejam visualizados em gráficos e tabelas.
- RF004 - O sistema deve permitir que os dados sejam atualizados automaticamente.

4.1.2 Requisitos Não funcionais

- RNF001 - O sistema deve ser hospedado em um servidor web.
- RNF002 - O sistema deve ser compatível e acessível a partir dos navegadores Google Chrome, Mozilla Firefox e Microsoft Edge.
- RNF003 - O sistema deve ter um tempo de atualização de no máximo 10 minutos.
- RNF004 - Integração com outras fontes de informações.

4.1.3 Requisitos de Domínio (Negócio)

- RD001 - Flexibilidade na origem das informações.

4.2 PROCESSO DE DESENVOLVIMENTO

4.2.1 Fase de Início

Na fase de Início do Projeto uma análise das soluções existentes no mercado foi realizada. Embora fosse possível utilizar o sistema embarcado das TVs, cada marca e modelo podem usar um sistema embarcado diferente, além dos fabricantes se atualizarem rapidamente, isso poderia gerar problemas de compatibilidade futura.

Outra solução seria a terceirização do desenvolvimento de uma aplicação Android e utilizar equipamento com sistema operacional Android conectados nas TVs, mas foi descartada devido ao alto custo de desenvolvimento e falta de equipe qualificada para dar suporte.

Por fim, nos baseando na análise feita nas soluções pesquisadas e principalmente em uma solução simples já existente na fábrica que consistia em um notebook antigo conectado a uma TV, exibindo uma planilha em Excel e necessitava de atualização por parte dos usuários a cada 30 minutos. Foi decidido a criação de uma solução customizada para o projeto, baseada em 4 partes distintas: i) Equipamento de Hardware para TVs, ii) Sistema Web, iii) Solução para atualizar os arquivos a serem exibidos no servidor web, iv) Automação para gerar e atualizar os dados dos arquivos a serem exibidos, com rapidez e facilidade de manutenção.

4.2.2 Fase de Elaboração

Na fase de Elaboração, foram realizadas as atividades de análise e *design* do sistema. A imagem 1 e Imagem 2 esboçam o *design* inicial das telas do site que apresentaria os indicadores de produção para os gestores e colaboradores nas TVs e pelo navegador web no desktop.

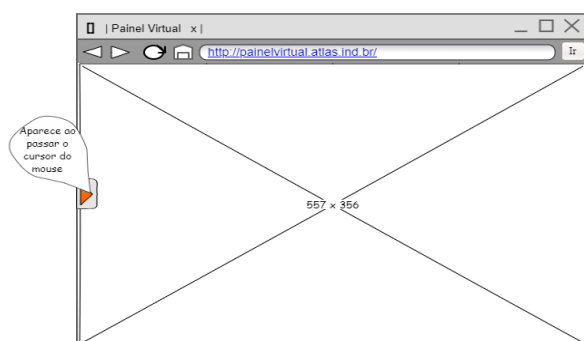


IMAGEM 1: PROTÓTIPO DE TELA INICIAL DO SITE. FONTE: AUTOR

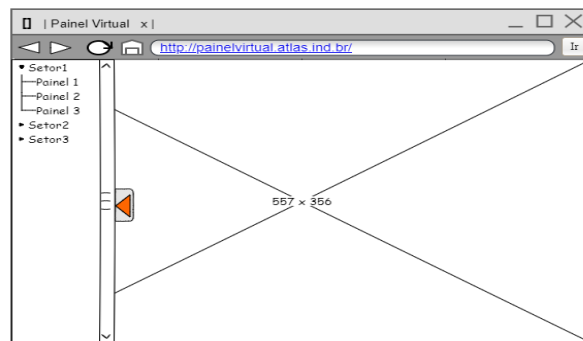


IMAGEM 2: PROTÓTIPO DE TELA INICIAL COM MENU SUSPENSO VISÍVEL. FONTE: AUTOR

4.2.3 Fase de Construção

Na fase de Construção, foram realizadas as atividades de desenvolvimento, testes e implantação do sistema.

4.2.3.1 Equipamento de Hardware para TVs.

Foi escolhido a utilização do *Raspberry Pi* para ser o dispositivo que irá exibir nas TVs os indicadores, pois ele oferece uma solução de baixo custo, possui recursos para rodar aplicações web e é de fácil manuseio. Foi optado por utilizar a versão “*Raspberry Pi 3 Modelo B+*” do equipamento para ser utilizado como interface para as TVs.

No equipamento foi criado um script que é executado quando o equipamento é iniciado. O “*script*” abre a página do Sistema WEB, que foi pré-configurada, no navegador em “modo *Kiosk*”. O modo *Kiosk* (kiosk mode) é um recurso que permite que apenas um aplicativo seja executado dentro de um sistema operacional, limitando a navegação do usuário por uma página web específica. No modo quiosque a página é vista em tela cheia, não é possível acessar o menu do browser ou mudar a URL pela barra de endereços. Também não é possível clicar com o botão direito do mouse e usar o menu. Ou seja, o usuário tem a navegação limitada a uma página específica e só consegue ir para outras páginas caso a página que ele esteja usando possua um link que o direcione para tal.

Foi preciso também criar o serviço “*kiosk.service*” no local “*/lib/systemd/system/*” para iniciar o “*script*” em nodo “*Kiosk*”. Após criado é preciso ativar e iniciar ele para sempre que o equipamento ligue o serviço execute o “*script*” e a página do website seja apresentada. O arquivo do “*scrip*” de inicialização e do serviço estão disponíveis no *GitHub* através do seguinte link “https://github.com/alexcatani/TCC/tree/main/Script_Raspberry”.

4.2.3.2 Sistema Web.

Para essa solução foi dado o nome de “Painel Virtual” e é um website hospedado em um servidor Apache que pode ser acessado apenas dentro da rede interna da empresa. Baseado em páginas HTML, JavaScript e CSS, o site foi desenvolvido para apresentar arquivos em formato de imagem e PDF, mas também aceita outros formatos como vídeos e outros arquivos HTML. O acesso a este website é obtido através do link “<http://painelvirtual.atlas.ind.br>” que é redirecionado via regra de NAT pelo *firewall* da empresa para o Servidor *WEB* interno.

Os arquivos e documentações estão disponíveis no *GitHub* através do seguinte link “<https://github.com/alexcatani/TCC/tree/main/WebSite>”.

4.2.3.3 Solução para atualização os arquivos a serem exibidos no servidor web.

No servidor de arquivo é utilizado o programa “*Cobian Backup*” versão 11, que se encontra em licença “*open source*”, para enviar de 5 em 5 minutos para o servidor *web* via protocolo FTP os arquivos gerados e atualizados no servidor de automação. Caso ocorra alguma indisponibilidade de rede, o software evita travar o processo, ignorando o envio e esperando o tempo definido para nova tentativa.

4.2.3.4 Automação para gerar e atualizar os dados dos arquivos a serem exibidos

Para automação da atualização dos dados foi utilizada uma ferramenta nativa do programa “Excel” que faz parte do pacote “MS Office”. A ferramenta é chamada de “AutoRepublicação” e gera automaticamente páginas HTML estática ao salvar o arquivo utilizado.

O Excel suporta uma linguagem de programação chamada de “VBA” onde podemos automatizar processos. Com esse recurso foi implementado código que acessam as informações no sistema de ERP da empresa (SAP) e traz as informações de produção.

No “servidor para automação” foi criado um “*script*” que é rodado através da ferramenta nativa do Windows “Agendador de Tarefas” para executar o arquivo criado no Excel. A ferramenta “Agendador de Tarefas” também é utilizada para reiniciar o programa do Excel, pois foi identificado que ele trava após um certo tempo de execução devido ao excesso de armazenamento de “cache”.

Os arquivos e documentações estão disponíveis no *GitHub* através do seguinte link “https://github.com/alexcatani/TCC/tree/main/Planilha_de_Automacao”.

4.2.4 Fase de Transição

Na fase de Transição, foram realizadas as atividades de treinamento e suporte ao sistema. Além disso, também foram feitos alguns ajustes no sistema, para garantir o correto funcionamento. A solução foi implementada e os equipamentos de hardware instalados nas TVs disponíveis. A solução ficou em teste para que os gestores e

colaboradores avaliasse o funcionamento da solução e alertassem possíveis erros e falhas. Por fim, foram realizadas as atividades de homologação e implantação, para garantir que o sistema estivesse apto a ser usado pela empresa.

5 CONSIDERAÇÕES FINAIS

Ao final do projeto, foi possível alcançar o objetivo que eram aplicar na prática as técnicas, metodologias e paradigmas abordados no ao longo da graduação cursada, também e principalmente aprofundamento no problema exposto neste artigo onde conseguisse um sistema eficiente, escalonável e de baixo custo.

Após a homologação da solução em produção a solução passou por alguns ajustes fazendo com que a quantidade de TVs utilizadas passasse de 4 para 22 em menos de 4 meses e com isso pode-se afirmar que a solução é válida e aplicável em outros ambientes.

Ainda á muitas melhorias a serem feita as quais posso citar a utilização de uma ferramenta de BI (*Business Intelligence*) que seria mais eficiente que uma planilha rodando no “Excel” para a compilação e geração dos arquivos de indicadores, também uma atualização no website onde pode ser utilizado tecnologias mais modernas como um servidor “NodeJS” com “GraphQL” para a criação das páginas web.

6 REFERÊNCIAS BIBLIOGRÁFICAS

PRESSMAN, R.; MAXIM B. R. **Engenharia de Software – Uma abordagem profissional**. 8. Ed. Porto Alegre: AMGH, 2016.

Sommerville, Ian; **Engenharia de Software** / Ian Sommerville; tradução Ivan Bosnic e Kalinka G. de O. Gonçalves; revisão técnica Kechi Hirma. — 9. ed. — São Paulo: Pearson Prentice Hall, 2011.

CENTRO UNIVERSITÁRIO DE MARINGÁ. Núcleo de Educação a Distância; OSHITA, Marcela Gimenes Bera. **Gestão de Projetos**. Marcela Gimenes Bera Oshita. Maringá-Pr.: UniCesumar, 2019. Reimpresso em 2021.280 p.

CENTRO UNIVERSITÁRIO DE MARINGÁ. Núcleo De Educação A Distância; FREITAS, Janaina Aparecida de. **Gerenciamento de software**. Janaina Aparecida de Freitas. Reimpressão – 2018 Maringá-Pr.: UniCesumar, 2017.181 p.”

6.1 REFERÊNCIA ON-LINE

¹Em < <https://www.cobiansoft.com> >. Acesso em 29 abr. 2022.

²Em < <https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/> >. Acesso em 30 abr. 2022.