



Os reais custos de produção como fator de produtividade na Indústria 4.0

Luís Miguel Neiva Teixeira

Doutor Ricardo André Fernandes Costa

Resumo

Vivemos uma era de grande desenvolvimento humano, tanto numa perspetiva económica como tecnológica onde a competitividade é fundamental para os que querem sobreviver tal azafama de informação e disponibilização de opções que nos absorve através de diferentes canais evoluídos pelo conseqüente desenvolvimento que fomos sujeitos.

Há não muitos anos atrás a elasticidade dos produtos disponíveis no mercado eram quase nulos ou mesmo inelásticos. Atualmente somos invadidos por imensas opções que os mercados nos oferecem e com preços cada vez mais competitivos. Por esta razão é cada vez mais importante identificar-se todos as variáveis que influenciam as nossas escolhas. Inevitavelmente, devido á agressividade dos mercados e competidores os preços dos produtos acompanham esta lógica e é de extrema importância obter-se o máximo de rigor e cuidado quando abordamos as questões de custos e proveitos nos produtos que são colocados no mercado.

Foi com base na perspetiva anteriormente abordada que surgiu a ideia de aprofundar a questão do rigor dos custos de mão de obra humana nas empresas com particular foco na indústria transformadora. Feita uma pesquisa global acerca da disponibilidade de softwares de registo e análise do que é produzido em cada empresa e também através das ideias de empresários relatando tal falta de soluções e controlo, chegou-se á conclusão que existe uma oportunidade para o desenvolvimento de uma plataforma que permita o registo e análise da informação das empresas relativamente ao custo de recursos humanos.

Palavras chave: Produtividade, acessibilidade, gestão, rigor, custos, indústria4.0.

Resume

We live in an age of great human development, both in an economic and technological perspective where competitiveness is fundamental for those who want to survive such a turmoil of information and availability of options that absorb us through different channels evolved by the consequent development that we were subject.

Not many years ago, the elasticity of the products available in the market were almost null or even inelastic.

Today we are invaded by a vast of options that the markets offer us and with increasingly competitive prices.

For this reason, it is increasingly important to identify all the variables that influence our choices.

Inevitably, due to the aggressiveness of markets and competitors, product prices follow this logic and it is extremely important to obtain the utmost accuracy and care when dealing with cost and profit issues in products that are placed on the market. The idea of deepening the issue of the strict costs produced in companies with a particular focus on manufacturing industry, emerged based on the perspective previously discussed.

After/Having conducted a global survey of the availability of software for register and analyzing what is produced in each company and also through the ideas of executives reporting such lack of solutions and control, it has been concluded that there is an opportunity for the development of a platform which allows the registration and analysis of information of companies regarding the cost of human resources.

Key words: Productivity, accessibility, management, accuracy/rigor, costs, industry.

Epígrafe

“Nem tudo o que se enfrenta pode ser modificado, mas nada pode ser modificado até que seja enfrentado.”

- Albert Einstein

Agradecimentos

A realização deste projeto de mestrado sem dúvida que se deve a diversas pessoas que estiveram direta ou indiretamente envolvidas.

Primeiro ao professor Ricardo Costa, diretamente envolvido e que sem a ideia e orientação este trabalho, que gostei de fazer e que me deu perspectivas interessantes acerca de diversas áreas como gestão, custos e produtividade, não seria possível. A sua disponibilidade e prontidão quando pedi ajuda ou orientação foi assinalável, pois caso contrário seria difícil acabar pelo menos para já devido aos tempos apertados e por isso agradeço.

A vários amigos que devido à ajuda e orientação fizeram com que o trabalho fosse mais completo e eficiente. Os seus conhecimentos e “debates” acerca de todo o projeto fizeram com que escolhesse as melhores ferramentas e as melhores abordagens.

Por fim às pessoas que ajudaram indiretamente. Aos meus familiares que sempre estiveram presentes e me deram força, compreensão e ajuda ao longo de todo o percurso.

Índice

1	INTRODUÇÃO.....	7
1.1	CONTEXTUALIZAÇÃO DO PROJETO.....	7
1.1.1	<i>Recursos humanos</i>	7
1.1.2	<i>Contabilidade Analítica</i>	8
1.1.3	<i>Indústria 4.0</i>	9
1.2	OBJETIVOS.....	12
1.2.1	<i>Objetivo geral</i>	12
1.2.2	<i>Outros Objetivos</i>	12
1.3	ESTRUTURA DO RELATÓRIO.....	13
2	SOLUÇÃO PROPOSTA	14
2.1	SOLUÇÕES ATUAIS (ESTADO DA ARTE).....	14
2.2	ABORDAGEM	17
2.2.1	<i>Intervenientes</i>	18
2.3	ESTRUTURA	19
2.3.1	<i>Arquitetura da solução</i>	20
3	CONCRETIZAÇÃO DA SOLUÇÃO PROPOSTA.....	21
3.1	PERSPETIVA GLOBAL	21
3.2	SERVIDORES.....	21
3.3	FERRAMENTAS E TECNOLOGIAS USADAS	22
3.3.1	<i>Node.js</i>	22
3.3.2	<i>Nodemon</i>	23
3.3.3	<i>ESLint</i>	23
3.3.4	<i>Docker</i>	23
3.3.5	<i>Bcrypt</i>	25
3.4	APLICAÇÃO MOBILE (IOS)	25
3.4.1	<i>Swift</i>	25
3.4.2	<i>CocoaPods</i>	26
3.4.3	<i>Alamofire</i>	26
3.5	FRONT-END	26
3.5.1	<i>Webpack</i>	26
3.5.2	<i>ElementUI</i>	27
3.5.3	<i>Vue.js</i>	27
3.5.4	<i>Veux.js</i>	30
3.5.5	<i>Vue i18n</i>	31

3.5.6	<i>Axios</i>	32
3.6	BACKEND	33
3.6.1	<i>Express</i>	33
3.6.2	<i>Mongo</i>	33
3.6.3	<i>Mongoose</i>	34
3.6.4	<i>Passport</i>	34
3.6.5	<i>Jwt</i>	34
4	PRODCLEVER	36
4.1	BASE DE DADOS	36
4.1.1	<i>Coleções e documentos</i>	37
4.2	APLICAÇÃO MOBILE	43
4.3	TEMPO REAL E INFERÊNCIA DA INFORMAÇÃO	45
4.3.1	<i>Websockets</i>	45
4.3.2	<i>Médias</i>	47
4.3.3	<i>Rankings</i>	49
4.3.4	<i>Pesquisas e histórico</i>	49
4.4	VISTAS DOS UTILIZADORES	51
4.4.1	<i>Login</i>	52
4.4.2	<i>Dashboard dos postos de trabalho</i>	52
4.4.3	<i>Posto de trabalho</i>	53
4.4.4	<i>Dispositivos</i>	54
4.4.5	<i>Encomendas</i>	55
4.4.6	<i>Relatório dos operadores</i>	58
4.4.7	<i>Utilizadores</i>	58
5	CONCLUSÕES E TRABALHO FUTURO	66
6	REFERÊNCIAS	69

Lista de Figuras

FIGURA 1 - CHÃO DE FÁBRICA, LINHAS DE PRODUÇÃO E POSTOS DE TRABALHO [5]	14
FIGURA 3 - VISÃO DE NEGÓCIO GLOBAL	17
FIGURA 4 - VISTA GLOBAL DA SOLUÇÃO A DESENVOLVER	20
FIGURA 5 - CONFIGURAÇÕES DO SOFTWARE DOCKER	24
FIGURA 6 - ESTRUTURA VUE.JS	29
FIGURA 7 - VEUX - ESQUEMA EXEMPLO	30
FIGURA 8 - ESTRUTURA PARA OS IDIOMAS SUPORTADOS	31
FIGURA 9 - CÓDIGO QUE COLOCA O TOKEN GERADO NOS HEADERS	35
FIGURA 10 - ESTRUTURA DA BASE DE DADOS NO MLAB [35]	36
FIGURA 11 - DOCUMENTO EXEMPLO DA “COLECTION COMPANIES”	37
FIGURA 12 - DOCUMENTO EXEMPLO DA “COLECTION DEVICES”	38
FIGURA 13 - DOCUMENTO EXEMPLO DA “COLECTION WORKSTATION”	39
FIGURA 14 - DOCUMENTO EXEMPLO DA “COLECTION ORDERS”	40
FIGURA 15 - DOCUMENTO EXEMPLO DA “COLECTION USERS”	42
FIGURA 16 - DOCUMENTO EXEMPLO DA “COLECTION REGISTRIES”	43
FIGURA 17 - ALGORITMO DE AUTENTICAÇÃO PARA O IOS	44
FIGURA 18 - ALGORITMOS QUE ENVIA OS REGISTOS PARA A BASE DE DADOS	45
FIGURE 19 – SOCKET.IO ABORDAGENS	46
FIGURA 20 - ALGORITMO PARA CÁLCULO DO TEMPO DE CONCLUSÃO DE ENCOMENDAS	48
FIGURA 21 - RANKING DE OPERADORES	49
FIGURA 22 - ESTRUTURA PARA PESQUISA DE UTILIZADORES	50
FIGURA 23 - EXEMPLO DE PESQUISA DE DISPOSITIVOS	50
FIGURA 24 - VISTA DE PESQUISAS POR DATA	51
FIGURA 25 - VISTA DE LOGIN	52
FIGURA 26 - DASHBOARD DOS REGISTOS DIÁRIOS DOS POSTOS DE TRABALHO	53
FIGURA 27 - VISTA INDIVIDUAL DO POSTO DE TRABALHO	54
FIGURA 28 - VISTA DA LISTA DOS DISPOSITIVOS MÓVEIS	55
FIGURA 29 - VISTA DA LISTA DE ENCOMENDAS	56
FIGURA 30 – VISTA INDIVIDUAL DA ENCOMENDA	57
FIGURA 31 - VISTA DE CRIAÇÃO DE UTILIZADOR E TODAS AS FUNCIONALIDADES	59
FIGURA 32 - VISTA DE LOGIN PARA O OPERADOR	60
FIGURA 33 - VISTA INICIAL COM OPERADOR AUTENTICADO	61
FIGURA 34 – VISTAS DO OPERADOR EM IOS "PAUSA" E "CONTINUAR"	62
FIGURA 35 - VISTA INICIAL DO UTILIZADOR CONSULTOR	63
FIGURA 36 - VISTA INICIAL DO UTILIZADOR CLIENTE	64
FIGURE 37 - VISTA DO GRÁFICO RELATIVO Á PRODUÇÃO DIÁRIA DA ENCOMENDA	65

Lista de Abreviaturas

API – Aplication Programming Interface

HTTP – HyperText Transfer Protocol

WEB – World Wide Web

VPS – Virtual Private Server

HTTPS – HyperText Transfer Protocol Secure

DNS – Domain name System

IOS – Iphone Operating System

JSON – JavaScript Object Notation

CSS – Cascading Style Sheets

HTML – HyperText Markup Language

XSRF – Cross-Site Request Forgery

ODM – Object Data Modeling

JWT – JSON Web Tokens

UUID – Universally Unique IDentifier

1 Introdução

1.1 Contextualização do projeto

Atualmente, vivemos em sociedades muito competitivas onde existe uma grande necessidade de controle das empresas. A gestão na indústria tem grande importância no rumo que as empresas tomam. Ter uma gestão de excelência é fundamental se é pretendido ter uma vantagem clara relativamente aos principais competidores. Más decisões por parte dos gestores e responsáveis de empresas poderão ter um impacto vital na sua existência futura. Assim, torna-se de extrema importância a criação de condições para que as diversas decisões sejam o mais precisas e assertivas possível. As novas tecnologias poderão ter um papel fundamental.

O departamento de produção de qualquer área da indústria tem custos significativos e é importante que tudo o que acarreta custos seja identificado e repartido por toda a produção de forma lógica para que os custos imputados sejam realmente o que cada produto custa. Existindo informação detalhada sobre tempos e contagens de produção, ou seja, custos exatos de imputação e potenciais quebras em tempo real, permite que ações sejam tomadas e decididas rapidamente podendo tornar as empresas ainda mais competitivas. A precisão da informação é primordial na tomada de decisões e disponibilizar informações detalhadas de quantidades produzidas no chão de fábrica é o foco.

1.1.1 Recursos humanos

Quando se trata de medir trabalho e tempos de recursos humanos a complexidade desta abordagem é exponencial. Calcular o trabalho e custos de máquinas é simples pois as máquinas executam com rigor tudo o que é programado, contudo quando o trabalho tem que ser executado por humanos temos sempre que ter em conta variáveis que nas máquinas não entram, como necessidades de descanso, necessidades fisiológicas, experiência, estado físico e emocional, etc. Por estas razões a precisão de execução e médias variam de sujeito para sujeito e também o mesmo sujeito poderá variar o seu desempenho em diferentes períodos.

1.1.2 Contabilidade Analítica

A contabilidade analítica analisa e apresenta resultados que poderão ser úteis na identificação de custos, proveitos e resultados, mas não é de todo suficiente quando se pretende maior rigor, pormenor e rapidez na obtenção da informação. Contabilidade analítica segundo Carvalho, *et al.* (1999) [1] “*é uma forma de tratamento de dados cujos objetivos essenciais são os seguintes: por um lado conhecer os custos das diferentes funções desenvolvidas pela empresa; determinar as bases de valorimetria de alguns elementos do balanço da empresa; explicar os resultados, comparando com o custo dos produtos (bens e serviços) com os seus correspondentes preços de venda; por outro lado, estabelecer previsões de custos e de proveitos correntes, verificar a sua realização e explicar os desvios das resultantes (e: controlo de custos e de orçamentos)*”. Este tipo de contabilidade, quando bem organizada, permite-nos ter informação sobre capacidades e produção detalhada, através de sistemas de custeio e coeficientes de imputação. Contudo os resultados alcançados são sempre numa perspetiva global e com tempos de apresentação das conclusões muito tardias onde esses mesmos resultados apenas servem para relatórios e dados estatísticos. Os resultados da contabilidade analítica também não têm a capacidade de chegar ao pormenor de identificar a produção e tempos de execução de cada produto em cada posto de trabalho e por trabalhador.

Na generalidade e pelo levantamento que tenho vindo a desenvolver ao longo deste projeto, registar manualmente o que se vai produzindo no chão de fábrica ou até mesmo registar numa folha cálculo é a realidade atual. Esta situação está desfasada das potencialidades atuais e a melhor solução seria ter as novas tecnologias ao serviço destas tarefas. O problema em torno da recolha da informação é apenas um dos vários problemas associados, porque além de ter custos com pessoal para a recolha das contagens de produção, existe problemas de enganos, esquecimento ou perda da informação quando esta se encontra registada em formato físico, originando consequentemente problemas de consistência, integridade e disponibilidade da informação. Os dados registados de modo manual não passam disso mesmo (dados). É necessário que os próprios decisores interessados em utilizar os dados transformem em informação, acarretando trabalho extra e existindo, também, o risco de enganos e falhas nas conclusões e decisões. Ter algoritmos já pensados e devidamente estruturados para que a informação seja automaticamente disponibilizada aquando do registo de dados será uma ajuda preciosa quando o objetivo além de mitigar erros é ganhar tempo.

Outro problema associado aos registos e contagens de produção manual é a falta de existência de histórico. Quando registada informação manualmente existe sempre o problema de arquivo físico que ao fim de algum tempo se perde quer seja por questões de espaço ou outras razões (o mais provável será livrarmo-nos de papeis ou ficheiros isolados que por si só não têm grande significado). Através de toda a informação sobre contagens de produção devidamente arquivada assim como toda a informação útil disponível ao longo do tempo, é possível o cruzamento da informação, analisar evoluções, calcular e decidir com base no historial e experiência passada ao longo do tempo.

Vivemos uma profunda revolução tecnológica onde praticamente tudo o que é desenvolvido inclui computadores, gestão da informação, acessibilidade e usabilidade. Cada vez mais existe uma dependência das novas tecnologias na resolução de problemas, sendo este, precisamente, um problema onde a sua ajuda é crucial.

1.1.3 Indústria 4.0

Este projeto está relacionado e pretende ir ao encontro deste conceito (Indústria 4.0) e estratégia. O objetivo será mesmo aproximar a indústria às diferentes abordagens que a “Indústria 4.0” orienta.

“Indústria 4.0” ou “Quarta Revolução Industrial” são termos que nos orientam e incutem tecnologias para a automação e troca de informação através de conceitos e **plataformas** como “Sistemas cyber-físicos”, “Internet das coisas” e “Computação em nuvem” [2]. Este tipo de abordagem rege-se por diversos princípios fundamentais no sentido de manter as fábricas e suas produções “inteligentes”: Interoperabilidade, Virtualização, Descentralização, Capacidade em Tempo Real, Orientação a serviços e modularidade [3]. Cada princípio mencionado tem a sua função para que os processos de produção [4] tenham o máximo de eficácia e eficiência baseados nas principais **inovações tecnológicas**:

Interoperabilidade → Capacidade dos sistemas cyber-físicos (linhas de produção, postos de reunião e produtos), humanos e fábricas inteligentes se ligarem e comunicarem uns com os outros através da internet das coisas e da computação em nuvem. O chão de fábrica ter a capacidade de, a cada fase de produção de determinado produto comunicar o seu fim em tempo real para uma plataforma (seria o ideal, pois desta forma e com estes dados será possível inferir informação a ser colocada em

nuvem para que fique online e possa ser acessada independentemente da localização de quem consulta);

Virtualização → Este princípio propõe uma cópia virtual das fábricas inteligentes, criada por sensores ou conectores que permitam monitorizar os processos físicos. Existindo forma de se transportar os resultados físicos para a estrutura digital é o grande desafio, pois é neste ponto que se estabelece a ligação entre a parte física e lógica;

Descentralização → As tomadas de decisão poderão ser feitas pelo sistema cyber-físico de acordo com as necessidades da produção em tempo real. As máquinas não só recebem comandos, mas também podem fornecer informações sobre o seu ciclo de trabalho. O objetivo deste projeto é abordar a questão de produtividade dos recursos humanos e por isso não é um princípio que possa ser útil nesta situação;

Capacidade em tempo real → Consiste na aquisição e tratamento de dados de forma instantânea, entregando conhecimento aos responsáveis e permitindo a tomada de decisões em tempo real. Este princípio é o grande foco na exposição do problema aqui mencionado. Resolver e agilizar este princípio será um grande passo para estar em sintonia com o “Indústria 4.0”, e conseguir vantagem para quem quer evoluir e melhorar;

Orientação a serviços → Utilização de arquiteturas de software orientadas a serviços e sua disponibilização através da computação em nuvem para os diferentes intervenientes do sistema é uma solução inteligente, pois agiliza o processo global e facilita o caminho a percorrer para tornar a indústria “mais inteligente”;

Modularidade → Produção e disponibilização de recursos (físicos e humanos), tanto para anexação como desanexação nas linhas de produção. Devem estar de acordo com o necessário, em função da informação disponível, isto é, partindo do pressuposto que a informação disponibilizada sobre o chão de fábrica é detalhada e rigorosa como se pretende, ajustar toda a estrutura para a produção de encomendas será facilitada existindo a possibilidade de orientar os recursos para outras tarefas ou até mesmo reforçar a produção, pois existe previsões de conclusão mais rigorosas e poderá existir a necessidade de antecipar a conclusão das encomendas.

A globalização é um fenómeno que não se pode evitar. Aceitar as evoluções, acompanhando-as, é a melhor alternativa para que indústria e “*stakeholders*”¹ se mantenham competitivos. Ter conhecimento sobre a evolução da produção, desde contagens de produção, tempo de finalização de encomendas e relacionar com o conhecimento e capacidades de produção já conhecidas permite que decisores atuem rapidamente, independentemente da sua localização. Ter toda a informação relacionada com contagens do que se está a produzir acessível em qualquer parte do mundo em tempo real é um objetivo primordial, tanto para ter conhecimento exato de custos, como saber se é necessário intervir de forma a que o que está planeado e acordado com clientes é cumprido. Todos os princípios identificados na indústria 4.0 são pertinentes e a ter em atenção, mas após análise é evidente que existem dois que se destacam de forma a alavancar todos os outros princípios e por consequência a indústria que adote tais princípios. “Interoperabilidade” e “Capacidade em Tempo Real” são os princípios que se destacam quando o que se pretende é ligar o mundo real e o mundo digital. O desafio é precisamente fazer com que estas duas realidades se misturem, pois, a partir do momento em que se consiga esta ponte tudo será mais fácil, relativamente à ponte que se estabelece entre realidade digital e realidade física. Obvio que depois de melhorado esta ponte outros problemas poderão advir dessa solução contudo é esse o caminho de melhoria constante que se pretende. Toda a informação inserida na realidade digital vinda da realidade física poderá ser trabalhada, inferida e disponibilizada através dos recursos disponíveis e já bastante evoluídos, quer seja pelas aplicações web existentes, quer pelas bases de dados disponíveis como pela acessibilidade que hoje em dia não é entrave seja qual for a localização dos interessados.

A disponibilidade de uma solução que preencha os requisitos de controlo, gestão e consulta de informação em tempo real da contabilização da produção no chão de fábrica irá melhorar a capacidade de decisão dos diversos decisores da cadeia de valor, podendo permitir, no limite, um acompanhamento em tempo real da evolução das encomendas pelos próprios *stakeholders*.

O histórico que é criado por uma solução que tenha como objetivo o registo de tudo o que se faz globalmente e também individualmente em chão de fábrica, poderá ser útil pois, por exemplo, numa linha de produção onde existem diversos postos de trabalho e os trabalhadores ocupem os diferentes postos a cada dia, existindo

¹ Partes interessadas em todo o processo abordado.

contagens de produção individual poderá ser possível distribuir recursos humanos de forma a que as linhas de produção sejam o mais produtivas possível. Também histórico relacionado com a produção das encomendas, sendo registado em tempo real, poderá ser disponibilizada parte da informação aos clientes para que eles saibam no estado em que estão as suas encomendas e previsões exatas de entrega.

1.2 Objetivos

1.2.1 Objetivo geral

O objetivo deste projeto é desenvolver uma “API” e um *Front-end*² que permita colmatar as lacunas identificadas ao longo da contextualização do projeto.

Pretende-se uma plataforma que, com rigor, calcule custos de trabalho humanos, tempos de execução e conseqüente cálculos de entrega em função de cada encomenda, registo de histórico relativamente a desempenhos dos humanos e tudo o que poderá resultar em custos de execução.

Para que o parágrafo anterior seja alcançado será desenvolvida uma plataforma online onde as empresas se poderão registar e utilizar como auxiliar na gestão dos recursos humanos e otimização dos mesmo nas linhas de produção das organizações.

1.2.2 Outros Objetivos

Utilizar ferramentas, linguagens e tecnologias eficazes e eficientes que estão a emergir no mundo tecnológico de forma a que os utilizadores da plataforma tenham a melhor experiência na sua utilização e também para que a performance, segurança e usabilidade sejam as melhores.

² Parte de um projeto que está relacionado com a componente visual do utilizador e que estabelece relação com o Backend.

1.3 Estrutura do relatório

Este documento começa por fazer uma contextualização da área que se pretende intervir e ao mesmo tempo explicar o que está desconforme tendo em conta as capacidades existentes atualmente. Aborda também o caminho a tomar para a melhoria na exposição, acesso e inferência da informação relacionada com produtividade.

Depois da abordagem á área de intervenção e do problema identificado é apresentada uma solução para a resolução e melhoria do problema exposto. Esta solução está descrita detalhadamente ao longo deste documento, desde as metodologias tomadas para a orientação no desenvolvimento, tecnologias utilizadas, organização de toda a estrutura da plataforma desenvolvida e identificação/explicação pormenorizada dos algoritmos mais pertinentes para a inferência a disponibilização da informação.

No final do relatório está descrita a experiência e conclusão acerca deste projeto assim como a perspetiva global de toda a solução e melhorias que poderão ser continuadas depois da conclusão desta fase.

2 Solução Proposta

Neste ponto é descrito de forma detalhada e estruturada a solução proposta tanto na perspetiva de negócio como na perspetiva de desenvolvimento.

2.1 Soluções atuais (estado da arte)

O registo manual é a solução mais utilizada pelo nosso tecido empresarial. Ao longo das linhas de produção os produtos trabalhados pelos colaboradores nos postos de trabalho vão-se acumulando até que alguém conte o que foi feito e registe em papel ou utilize um dispositivo móvel para registar as contagens num excel ou alguma ferramenta mais funcional e que permita cruzar informações e concluir algo mais que as simples contagens.

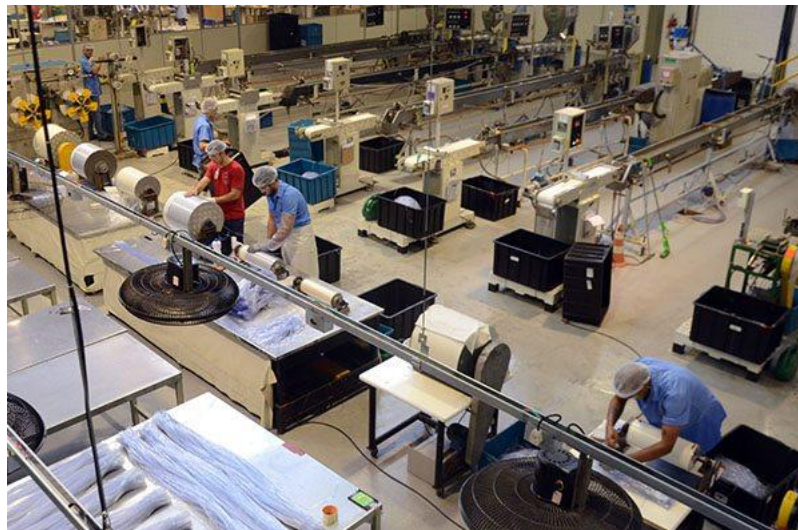


Figura 1 - Chão de fábrica, linhas de produção e postos de trabalho [5]

Após análise das soluções disponíveis no mercado foram identificadas duas que se aproximam e se estruturaram de forma a que exista informação sobre o que se produz, tempos e previsões de entregas de encomendas, “ProdSmart”³ e “Pegasus”⁴. No entanto e como já foi referido antes, também estes softwares não apresentam a informação em tempo real no que toca a tempos sobre contagem da produção numa perspetiva individualizada. As soluções mais evoluídas referem a expressão “tempo

³ <https://prodsmart.com/pt-pt/textil>

⁴ <http://www.pegasussystems.com/web/production.jsp>

real”, mas essa mesma informação será considerada tempo real a partir do momento em que um utilizador do software resolve introduzir os dados produzidos pelos trabalhadores nos postos de trabalho, ou seja, este “tempo real da informação” apenas se refere à disponibilidade a partir do momento que é inserida a informação (conjuntos de informação) e não à execução do trabalho efetuado (registo a cada execução da tarefa concluída).

Como já referido, o grande foco é a indústria de produção e transformação. Como exemplos para acompanhar e simular as funcionalidades da plataforma desenvolvida tive em conta, especialmente, empresas de calçado e têxteis, por serem indústrias fortes na zona onde estou inserido e são flagrantes para aplicar a solução. Uma empresa que fabrique calçado tem que ter o seu departamento de produção devidamente organizado e estruturado devido à complexidade da dinâmica que tem que adotar no sentido de ser o mínimo dispendioso possível e também a melhor capacidade de execução dos seus produtos de forma a ganhar tempo, dinheiro e a satisfação dos clientes. Esta indústria requer diferentes postos de trabalho desde: desenho na matéria prima, corte, preparação ou modelagem dos materiais, costura, montagem (ex: cabedal), preparação da sola, colagem, montagem final (ex: cordões), supervisão e embalagem. Todas estas fases percebem-se que são tarefas cíclicas e com procedimentos iguais em todos os postos de trabalho que asseguram as diferentes fases de construção de calçado. Atualmente, cada posto tem um recipiente onde são colocados os produtos para posteriormente serem contabilizados de forma a que no final do dia se possa ter uma ideia da quantidade produzida em cada posto de trabalho e fazer uma dispersão dos custos que se teve com os recursos humanos e que poderá ser tido em conta no cálculo do custo dos produtos. Ora cada produto varia em função das suas características e dependendo do modelo poderá custar mais ou menos executar as tarefas nos postos de trabalho aumentando ou diminuindo o custo de mão de obra dependendo da complexidade de construção do modelo em causa. Por esta razão é fundamental na indústria transformadora aplicar os reais custos de mão de obra ao produto, pois, por exemplo:

Exemplo: se temos uma encomenda de 5000 sapatos para serem feitos, onde o custo total unitário de tempo por sapato é de 20 minutos e temos uma segunda encomenda de 1000 sapatos onde o custo de tempo é de 5 minutos não faz sentido estar a aplicar coeficientes de imputação, visto que, tanto se pode estar a ter lucro significativo como um prejuízo considerável.

No exemplo apresentado a empresa estaria a ter um grande prejuízo na encomenda dos 5000 sapatos porque a média a calcular através de coeficientes de imputação (para as duas encomendas) estaria a baixar consideravelmente o custo de mão de obra do produto. A encomenda dos 1000 sapatos estaria a dar mais lucro que o esperado, contudo os números de produção são muito diferentes e daí o prejuízo.

As empresas que operam na área da transformação têm esta dificuldade e as empresas com as quais tive contacto transmitiram-me isso mesmo, pessoas experientes na área que expuseram as suas dificuldades e os cuidados que têm que ter para que não tenham prejuízo quando necessitam de calcular o custo de todo o processo produtivo dos seus produtos. Como referido por estes (pessoas experientes na área), o ideal seria encontrar uma solução que permita identificar os reais custos dos produtos, as capacidades produtivas por posto de trabalho e até mesmo um histórico diário que permita saber a capacidade que as empresas dispõem em função do histórico. Também conseguir dispersar os seus recursos pelos postos onde estes são mais produtivos poderá trazer benefícios às empresas e consequentemente satisfação aos clientes pela celeridade que podem ganhar nesta gestão e na entrega das encomendas.

Atualmente, não é fácil encontrar situações onde as novas tecnologias não estejam já pensadas. Contudo, no tema abordado nesta temática específica, é algo que, mesmo após pesquisa, é difícil encontrar uma solução que evidencie e resolva esta problemática. A grande barreira, para que ainda não existam soluções significativas que satisfaçam os princípios mencionados para este projeto será mesmo a dificuldade em juntar duas realidades, a física e a digital (lógica) [6] [7]. Existem aplicações que já foram desenvolvidas a pensar nestes problemas de contabilização da produtividade, contudo ainda há um caminho a percorrer para disponibilizar informação tão rigorosa e precisa como a que se propõe.

2.2 Abordagem

A intenção é disponibilizar uma plataforma que resolva e melhore a perceção da evolução do que se produz nos departamentos de produção das empresas no que se refere a trabalho humano, assim como as capacidades de produção, seus custos e tempos de resposta rigorosos.

A figura 2 pretende dar uma visão global onde a plataforma a desenvolver vai operar e quais os intervenientes que utilizaram a solução. É claro através da imagem identificar quem vai alimentar todo o sistema e quem vai usufruir da informação a registar.

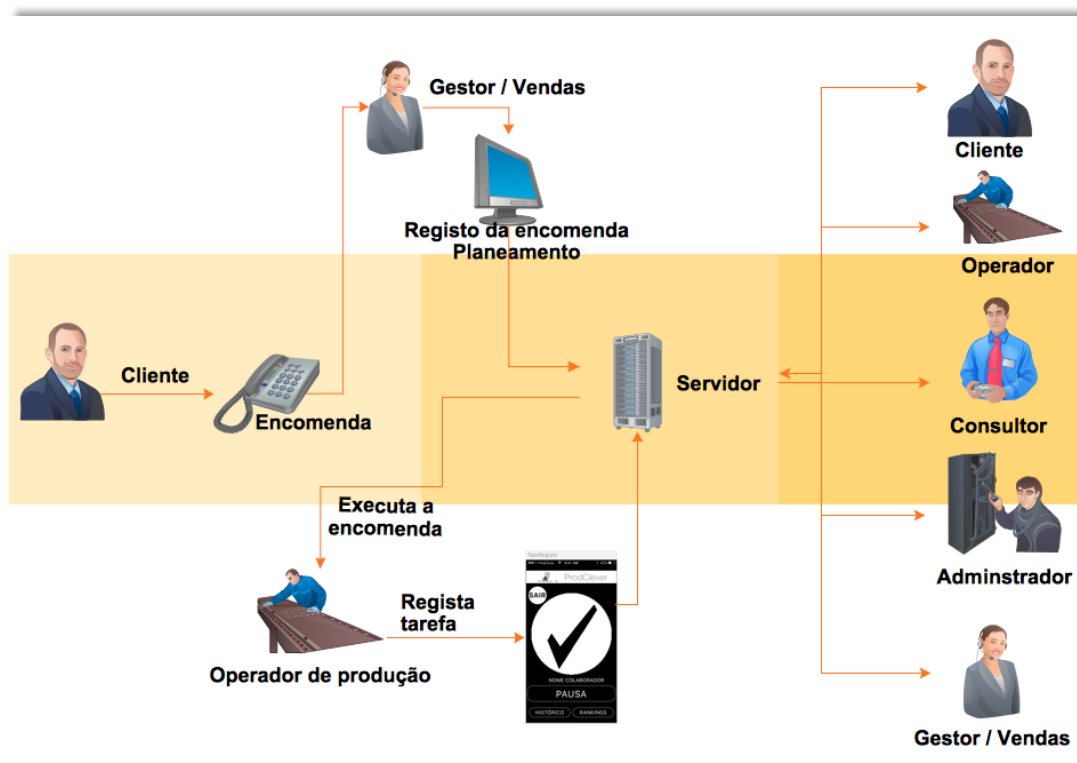


Figura 2 - Visão de negócio global

2.2.1 Intervenientes

A solução que se pretende desenvolver, tem como finalidade encontrar uma forma que cada operador da área de produção registre o que faz a cada execução concluída com o mínimo de interferência possível no seu trabalho. Para isso é disponibilizado um dispositivo *smartphone* ou *tablet* onde está disponível um botão que ocupa grande parte do ecrã para que sempre que seja concluída uma tarefa o operador dê um clique rápido. Sendo um trabalho repetitivo e cíclico á medida que os operadores vão dando os cliques o sistema vai recolhendo esses dados. A cada clique é enviado para a base de dados informação sobre: a hora, operador, dispositivo, empresa, encomenda e posto de trabalho. Com esta informação é possível saber: médias de tarefas, produtividade, capacidade, tempo de conclusão de encomendas e histórico. Toda a informação está disponível para diferentes intervenientes á escolha do administrador do sistema sob diferentes perspetivas.

Os diferentes utilizadores têm disponível online informação útil, que lhes permite estarem informados e que poderão utilizar de forma a tomarem decisões atempadamente, visto que a informação é em tempo real relativamente ao que se passa no mundo real, neste caso dentro de portas da organização e respetivo departamento de produção.

Esta solução prevê cinco tipos de utilizadores que podem usufruir da informação. O **operador de produção** tem disponível os seus tempos de execução, medias e rankings dos postos de trabalho por onde passou. Esta informação permite ao operador saber se realmente está dentro dos padrões e em que postos de trabalho é mais produtivo. O **gestor da aplicação** (encarregado), este interveniente é o responsável por gerir os postos de trabalho, operadores e linhas de produção, assim como controlar a evolução das encomendas. A plataforma disponibiliza o que é necessário para flexibilizar a estrutura física e anexar os recursos e postos de trabalho disponíveis a cada encomenda. O **consultor** (patrão, administrador, gerente, etc...) é o principal visado e para o qual esta ideia se desenvolveu, pois é o interveniente que vai ter disponível toda a informação global da plataforma, tanto numa perspetiva global através de *dashboards*⁵, como individual se assim o entender. O **administrador do sistema** tem acesso á plataforma numa perspetiva global. Este utilizador tem permissões para criar utilizadores, dispositivos e postos de trabalho

⁵ Painel que permite disponibilizar informação estratégica para análise.

que posteriormente são organizados pelo gestor da aplicação. Tem também a responsabilidade de inserir e editar a informação relativa á empresa na plataforma. É o administrador que inicia todo o processo de utilização da plataforma pois é este que regista e insere toda a informação da empresa e sua estrutura física. Por fim temos o utilizador **cliente**. Este utilizador tem vistas que lhe permitem consultar onde informação acerca das suas encomendas e evolução em tempo real, ou seja, com esta funcionalidade consegue saber qual a percentagem já concluída das suas encomendas e a previsão de conclusão das mesmas. Tem também disponível o histórico em que lhe fornece a informação acerca de todas as encomendas, desde a data de início, data de conclusão e número de peças executadas em cada encomenda.

Um aspeto importante a mencionar nesta solução são os tempos de conclusão das encomendas. Como referimos anteriormente a contabilidade analítica e mesmo as contagens que se fazem manualmente não nos permitem saber com rigor o custo de trabalho efetuado em cada modelo e os tempos de conclusão das encomendas. O objetivo será disponibilizar um algoritmo que vai calculando os tempos de conclusão nos diferentes postos de trabalho e respetivas médias á medida que a encomenda vai sendo executada visto que cada encomenda tem as suas características específicas e não podem ser comparadas com outras efetuadas anteriormente a não ser que seja exatamente o mesmo produto e o mesmo modelo.

2.3 Estrutura

A abordagem a seguir foi pensada e planeada de forma a que tudo seja desenvolvido com software, módulos e algoritmos já pensados e devidamente testados para que não haja problemas relacionados com segurança, escalabilidade acessibilidade ou disponibilidade.

O *node.js* é a principal ferramenta utilizada em todo o sistema facilitando o seu desenvolvimento. Esta ferramenta disponibiliza uma panóplia de módulos que torna a sua utilização bastante abrangente e segura, permitindo também a comunicação e interoperabilidade entre os diferentes módulos e consequentemente entre as diferentes abordagens ou tecnologias [8].

2.3.1 Arquitetura da solução

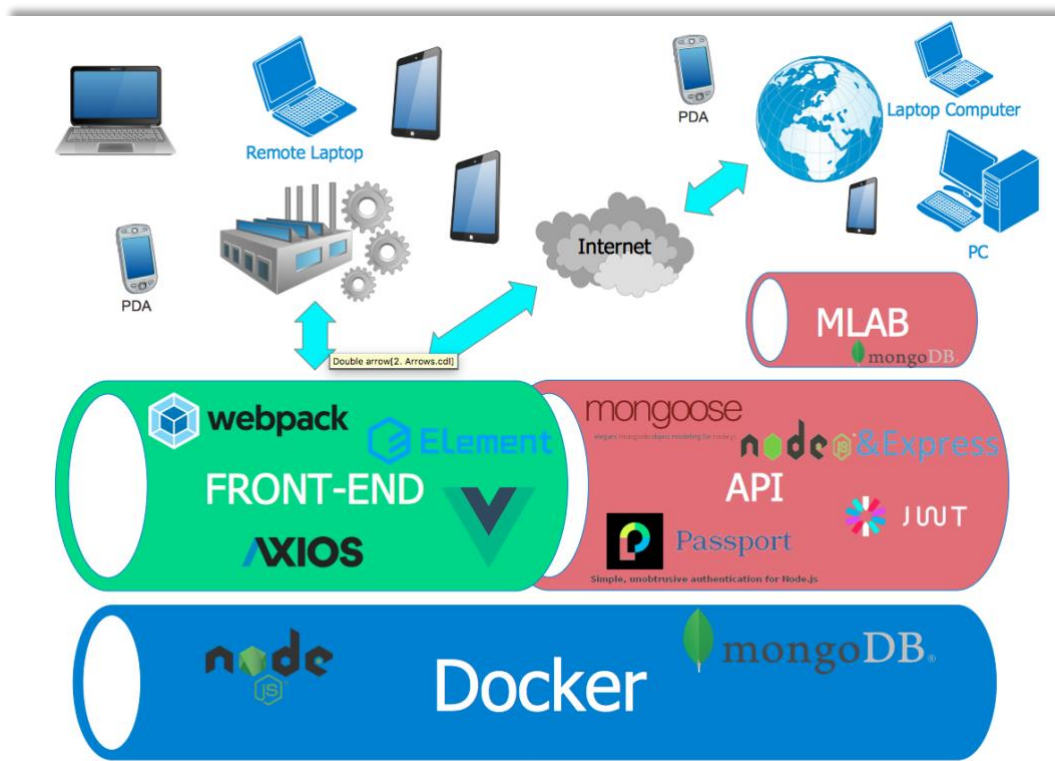


Figura 3 - Vista global da solução a desenvolver

Como a figura 4 demonstra, a solução global é estruturada em diferentes partes, desde a estrutura base que será o **Docker**⁶ e que tem como finalidade permitir e facilitar a utilização de software para a plataforma. A componente lógica fica dividida entre duas partes, a **API** e o **FrontEnd**⁷. A **API**⁸ será a componente responsável por servir de interface entre o **FrontEnd** e as bases de dados assim como inferir toda a informação provenientes das bases de dados. O **FrontEnd**. Numa fase de desenvolvimento a base de dados utilizada estava incluída no **Docker**, mas quando finalizado o projeto a base de dados fica alojada no serviço **MLab**⁹ por questões de performance, fiabilidade e integridade da informação.

⁶ Plataforma para desenvolvimento.

⁷ Interface que permite a ligação entre utilizadores e plataforma

⁸ Interface que estabelece.

⁹ Serviço cloud de base de dados mongoDB.

3 Concretização da solução proposta

3.1 Perspetiva Global

Toda a plataforma está suportada por diferentes serviços, *frameworks*, módulos e softwares. Neste ponto serão abordadas todas as ferramentas que suportam a plataforma. Toda a informação, autenticação e disponibilização em “*realtime*” mereceu todo o cuidado e que através das ferramentas a seguir mencionadas permitiu consistência, integridade e segurança em todos os aspetos e pormenores. A solução está dividida em 3 grandes partes que é importante mencionar e descrever como é pretendido neste ponto. As partes são: Mobile, *frontEnd* e *BackEnd*. Mobile, entenda-se que é tudo o que está relacionado com o desenvolvimento necessário para dispositivos móveis como *smartphones* e *tablets*. *frontEnd*, tudo o que está desenvolvido na parte de cliente, ou seja, a componente mobile apenas executa uma pequena parte da aplicação que é responsável pelos registos necessários para que toda a componente lógica seja alimentada e conseqüentemente o *frontEnd* que engloba toda a interatividade entre utilizadores e respetivas vistas que permitem essa interação. A terceira parte é o *BackEnd*, é onde está desenvolvida toda a lógica da plataforma, desde a base de dados às validações de integridade e algoritmos de análise de informação a disponibilizar no *frontEnd*.

3.2 Servidores

Toda a plataforma fica disponível através de uma VPS¹⁰. Esta solução, adquirida no site da *Scaleway* com o IP: 51.158.64.199. VPS é uma solução económica para que se possa ter um servidor online com excelente desempenho e serviços confiáveis. Além de ser possível ter um IP fixo, este fica disponível “24-7-365”. Este serviço tem como objetivo que a plataforma esteja online e disponível na internet. O serviço apesar de ser acessível financeiramente, garante disponibilidade e fiabilidade aceitável para o contexto atual de forma a que toda a estrutura funcione na rede mundial. Todas as funcionalidades e informação que possa ser acedida pelos diferentes utilizadores provêm ou fluem desta VPS criada para o efeito.

Numa segunda VPS com o IP: 51.15.220.49 está a funcionar como servidor de

¹⁰ Máquina virtual adquirido num fornecedor de hosting virtual.

email. De momento a entrada no cliente de email é o link: “https://51.15.220.49/mail”. Este servidor além de ter disponível o email noreply@prodclever.com para a funcionalidade de recuperação da password, numa segunda fase do projeto pode ser aproveitado por clientes que pretendam e-mails com o domínio “prodclever.com” [9].

3.3 Ferramentas e tecnologias usadas

Foi criado um domínio (<http://www.prodclever.com/>) na “OVH”¹¹ para que o acesso á plataforma **ProdClever** seja facilitado e também com o intuito que funcionalidades como recuperação de senha de acesso funcionem através do email.

Cloudflare foi o serviço escolhido para a resolução de DNS, ou seja, o domínio prodclever.com direciona para o IP: 51.158.64.199 [10].

A VPS que suporta o servidor de email utiliza o software *open source*¹² *iredmail* para fazer a gestão deste serviço. O software *iredmail* funciona em diversos sistemas operativos Linux: Red Hat, CentOS, Debian, Ubuntu, FreeBSD e OpenBSD [11]. Os Sistemas operativos escolhidos para as duas VPS’s foi o Debian Jessie, o que quer dizer que o software escolhido e o sistema operativo são compatíveis, existindo muita e boa documentação para a implementação e configuração do iredmail no sistema operativo Debian Jessie [12].

3.3.1 Node.js



Como já foi referido node.js é um interpretador de código javascript que também funciona do lado do servidor e tem como grande vantagem a escalabilidade, pois permite uma boa otimização de recursos das máquinas e também uma excelente gestão de pedidos/respostas através de uma única *thread*¹³, mas com um componente muito eficiente chamado de “*Event Loop*”. A função de “*Event Loop*” é a de gerir todos pedidos feitos ao servidor sem que estes mesmos pedidos entrem diretamente na execução de IO direto. A perspetiva de “*Non blocking IO*” é uma das razões a que leva à escolha do node.js, pois é uma das principais vantagens na sua utilização. “*Non blocking IO*” é a capacidade de se conseguir executar diferentes

¹¹ Empresa que vende domínios e alojamentos.

¹² Software gratuito.

¹³ É a estrutura que o sistema operativo cria para gerir processos.

tarefas ao mesmo tempo sem que a utilização de recursos por parte de uns processos não interfira na execução necessária para as respostas a dar em outros processos, ou seja, o servidor poderá estar demorado a dar respostas de determinados pedidos, mas isso não irá interferir com outras execuções e respostas a serem dadas pelo servidor [13]. Esta perspetiva é interessante na solução a ser disponibilizada visto que permite garante rapidez e fluidez na solução e com garantias.

Outra das grandes motivações para a escolha desta ferramenta (node.js) é a grande comunidade que contribui para a melhoria e disponibilidade de soluções. Existem muitos “packages” á disposição para o auxílio das funcionalidades que se pretendem implementar e suficientemente maduras para que se possam utilizar profissionalmente [14].

3.3.2 Nodemon



nodemon

Dentro dos diferentes utilitários que temos disponíveis no node.js temos uma ferramenta útil para o desenvolvimento, o Nodemon. Esta ferramenta permite ajudar no desenvolvimento da plataforma pois quando implementada monitoriza as alterações feitas ao código que está a ser desenvolvido e automaticamente reinicia o servidor evitando assim a execução de comandos para que as alterações surta m efeito [15].

3.3.3 ESLint



Esta opção permitiu que fosse possível manter as boas práticas na indentação do código escrito para a solução proposta e também ajudou na redução de possíveis erros que se pudesse incorrer evitando assim perdas de tempo devido a distrações [16].

3.3.4 Docker



Esta opção permite que exista um ambiente pré configurado com as ferramentas necessárias já instaladas e prontas a serem utilizadas por parte de quem desenvolve software. Nesta situação é utilizada uma imagem de um sistema operativo onde já temos instalado e configurado o *mongoDB* e todos os comandos

necessários ao arranque da *API* desenvolvida e respetivos módulos. Desta forma evitamos que exista a celebra afirmação “no meu funciona”, pois, sendo o arranque da imagem, serviços e respetivos comandos da responsabilidade do software *docker*, todos os intervenientes que pretendam interagir com a plataforma em desenvolvimento não terão problemas nem preocupações, pois apenas têm que executar o *docker*.

As figuras abaixo mostram os diferentes ficheiros, comandos e configurações necessárias para que a plataforma em desenvolvimento funcione. Nestas imagens é ilustrativo as aplicações e serviços que o *docker* está a suportar assim como o *container* da plataforma e as portas pelas quais existem comunicações [17].

```
docker-compose.yml x
1 version: '3'
2 services:
3   server:
4     build:
5       context: ./
6       dockerfile: Dev-Dockerfile
7     image: prodclever/prodclever-server:dev
8     environment:
9       NODE_ENV: development
10    env_file:
11      - .env
12    ports:
13      - "8080:8080"
14    depends_on:
15      - mongo
16    volumes:
17      #Para mapearmos o nosso codigo com o container
18      - ./:/code
19      - /code/node_modules
20    mongo:
21      image: mongo
22      ports:
23        - "27017:27017"
24


Dockerfile x
1 FROM node:8.5
2
3 # Create app directory
4 RUN mkdir -p /code
5 WORKDIR /code
6
7 COPY package.json /code
8 RUN npm install
9
10 # Bundle app source
11 COPY . /code
12
13 EXPOSE 8080
14 CMD [ "node", "index.js" ]
15

Dev-Dockerfile x
1 FROM node:8.5
2
3 # Create app directory
4 RUN mkdir -p /code
5 WORKDIR /code
6
7 COPY package.json /code
8 RUN npm install
9
10 # Bundle app source
11 COPY . /code
12
13 EXPOSE 8080
14 CMD [ "npm", "run", "dev" ]
15

docker-compose.prod.yml x
1 version: '3'
2 services:
3   server:
4     restart: always
5     build: .
6     image: prodclever/prodclever-server:prod
7     environment:
8       NODE_ENV: production
9     env_file:
10      - .env
11     ports:
12      - "8080:8080"
13
```

Figura 4 - Configurações do Software Docker

3.3.5 Bcrypt


 Esta ferramenta permite que as senhas dos utilizadores sejam cifradas aumentando assim a segurança da plataforma e a respetiva senha caso o próprio utilizador utilize a sua senha para outros fins. Este método de criptografia é dos mais utilizados e recomendados, pois garante excelente desempenho/proteção em comparação com outros métodos de criptografia [18].

3.4 Aplicação mobile (IOS)

Esta componente do projeto é extremamente importante e, por isso, tem que ser devidamente implementada para que não se perca o foco sobre o objetivo proposto, ou seja, é importante que seja de simples execução, que não contenha demasiadas opções de forma a não dificultar a utilização por quem não tem grandes conhecimentos informáticos, o menos intrusiva possível nos trabalhos dos recursos humanos em questão e que envie de forma simples e rápida a informação pretendida para a API.

O desenvolvimento da aplicação mobile será orientado para dispositivos IOS¹⁴, por nenhuma razão em especial, simplesmente porque de momento tenho equipamentos Apple que me permitem testar e executar a aplicação em diferentes modelos, não sendo necessário investir em equipamentos para dar continuidade ao projeto. Uma melhoria futura será desenvolver a aplicação Android e até mesmo Windows Mobile de forma a haver mais versatilidade de escolha e acesso a equipamentos mais acessíveis financeiramente.


3.4.1 Swift

 A aplicação mobile está desenvolvida em *Swift4*. *Swift* é a linguagem de programação da Apple para desenvolvimento em ambientes: IOS, macOS, watchOS e tvOS. O *swift4* (versão atual) tem uma ferramenta útil para facilitar e organizar o código a aproveitar através de “Packages”, o “*Swift Package Manager*” [19] [20], contudo a ferramenta escolhida para gestão de dependências e

¹⁴ Sistema operativo da Apple.


organização do projeto foi o *Cocoapods*.

3.4.2 CocoaPods

 Permitiu a estruturação e organização de toda a componente mobile. *Cocoapods* gere todas as dependências para projetos em xcode¹⁵.


A distribuição e reutilização de código faz cada vez mais sentido, pois código já escrito e maturado por programadores não tem necessidade de estar constantemente a ser reescrito, daí a importância de um gestor de pacotes para linguagens de programação e nesse sentido o CocoaPods está perfeitamente enquadrado com as novas tendências [21]. O gestor de pacotes permite assim, a capacidade de reutilizar e proporciona a interoperabilidade entre diferentes origens de código, o que para este projeto foi de grande utilidade visto que permitiu a utilização de código já devidamente desenvolvido e testado assim como soluções sob diferentes perspetivas.

3.4.3 Alamofire

 É uma biblioteca que ajuda na comunicação através da rede escrita em Swift. Este módulo foi de grande utilidade, pois caso contrário reescrever código para se conseguir comunicar com a API em node.js de raiz seria algo muito moroso e a dimensão do projeto aumentaria exponencialmente. Os métodos disponibilizados pelo Alamofire de pedidos e respostas, parâmetros em JSON, a serialização das respostas e autenticação foram as principais utilidades na implementação da aplicação mobile [22] para o Prodclever.

3.5 Front-End

3.5.1 Webpack

 Este “agregador de módulos” foi muito útil pois permitiu agregar diversos ficheiros javascript e criar *bundler's* de forma a que haja mais facilidade na implementação e execução de diferentes dependências de código, assim como a

¹⁵ Ambiente de desenvolvimento disponibilizado pela Apple.

simplificação e otimização de ativos (Imagens, ficheiros javascript, css¹⁶, etc). Também permitiu que facilitasse na interpretação e execução de ficheiros que não são compatíveis com os *browser's* onde a utilização desses mesmos tipos de ficheiros permitem a modularização e agregação de diferentes fontes de código, como por exemplo, os ficheiros com extensão “.vue”, “.vuex” “.coffee”, “.less” [23]. A este processo de transformação de ficheiros que o navegador não consegue interpretar para a sua execução por parte do browser dá-se o nome de *transpiling* e é precisamente por este motivo que o *webpack* se torna útil neste projeto.

3.5.2 ElementUI



Neste projeto foi utilizado o módulo *ElementUI* do *Node.js* para auxiliar na aparência e facilitar a implementação de componentes que permitiu que a plataforma desenvolvida se tornasse mais apelativa e funcional. *ElementUI* é uma biblioteca de componentes para Vue. Existem outras bibliotecas para desempenharem esta tarefa, contudo esta parece ser a mais utilizada e mais “madura”, por isso, a mais fiável e completa garantindo assim que todos os componentes que possam ser utilizados no o projeto funcionem sem problemas de bugs ou constrangimentos de limitação dos componentes.

Na plataforma desenvolvida é utilizada a versão 2.0 do *ElementUI*. Inicialmente estava a ser utilizada a versão 1 mas com o evoluir do projeto percebi que a versão ia deixar de ter atualizações e os próprios responsáveis da ferramenta sugeriam a atualização para a versão mais recente. Na nova versão foi perceptível maior flexibilidade nos componentes, por exemplo, o carregamento do conteúdo no componente *Table*, ou o componente *DateTimePicker*, simples de utilizar, aplicar e esteticamente agradável [24]. Também é pertinente destacar a simplicidade e facilidade como são apresentados os componentes e o respetivo código para cada componente por quem disponibiliza esta biblioteca. Um exemplo a seguir!

3.5.3 Vue.js



Este módulo (Vuex) é sem dúvida o módulo mais utilizado e visível neste desafio. Vuex é uma Framework para a construção de interfaces web e neste

¹⁶ Cascading Style sheets – Ficheiro de configuração e formatação de documentos HTML.

momento tem uma comunidade e estrutura considerável ao ponto de ser utilizada por organizações com dimensão mundial. É uma *framework* com uma grande variedade de componentes: acessíveis, versáteis e escaláveis, o que fez com que empresas como a Alibaba ou a Xiaomi a utilizassem, daí a decisão de utilizar neste projeto fosse com mais convicção e sem dúvidas de que seria uma excelente opção.

A reutilização de código e componentes é uma característica que se destaca nas muitas outras vantagens que o *Node.js* oferece. A *Framework Vue.js* reflete os conceitos de reutilização, componentes, acessibilidade, Framework progressiva, documentação simples e de fácil compreensão. Pelo citado acima garantidamente que a camada visual ficará perfeitamente funcional e todo o “ecossistema” com os diferentes ficheiros (.css, .html, .js, etc...) terá uma “harmonia” garantida [25].

```
dashboard.vue — ProdClever
V dashboard.vue x
1 <template>
2   <pcPage
3     title="Dashboard"
4     class="dashboard"
5     full-width>
6     <div class="dashboard-workstations">
7       <WorkstationCard
8         v-for="(workstation, index) in dashboard"
9         class="dashboard-workstations__workstation"
10        :key="index"
11        :workstation="workstation" />
12     </div>
13   </pcPage>
14 </template>
15
16 <script>
17   import { mapGetters, mapActions } from 'vuex'
18
19   import WorkstationCard from './components/workstation-card'
20
21   export default {
22     name: 'Dashboard',
23
24     components: {--
25     },
26
27     computed: {--
28     },
29
30     watch: {--
31     },
32
33     methods: {--
34     },
35
36     sockets: {--
37     },
38
39     created () {--
40     }
41   }
42 </script>
43
44 <style lang="scss" scoped>
45   .dashboard-canvas {--
46   }
47
48   .dashboard-square_element {--
49   }
50
51   .dashboard-workstations__workstation {--
52   }
53 </style>
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
```

Figura 5 - Estrutura Vue.js

Na imagem acima podemos confirmar a estrutura do ficheiro “.vue” da *dashboard* e que é a estrutura para todos os ficheiros *vue.js*. Entre as *tag's*¹⁷ “<template> </template>” fica localizado toda o código respeitante ao *html* a ser apresentado nos *browser's*, entres as *tag's* “<script> </script>” ficará localizada toda a lógica, ou seja, todos os dados e respetivos processamentos serão executados

¹⁷ Linguagem de marcação que identificam instruções.

nesta parte do ficheiro, por fim a última parte do ficheiro entre as tag's “<style></style>” está localizado código responsável pela estética do que é apresentado.

Este é um exemplo onde está a ser utilizado diferente código respeitante e diferentes ficheiros e que agrega e interpreta esses diferentes códigos num só ficheiro.

3.5.4 Vuex.js



O objetivo do Vuex é comportar-se como um *singleton*¹⁸, *Veux* foi inspirado no *Flux* e *Redux(React)* mas adaptado e otimizado para o *Vue*. O *Veux* é um componente que ajuda na gestão de estados e dados que fluem na plataforma através do *Vue.js*. O *Veux* permite uma abstração da aplicação ficando independente da mesma, mas onde todos os componentes desenvolvidos em *Vue.js* tem acesso. Desta forma torna-se mais simples gerir os estados, e componentes que devem reagir às mudanças executadas na estrutura de dados. A imagem abaixo exemplifica de forma clara a função do *Veux*, assim como o *Flux* e *Redux*.

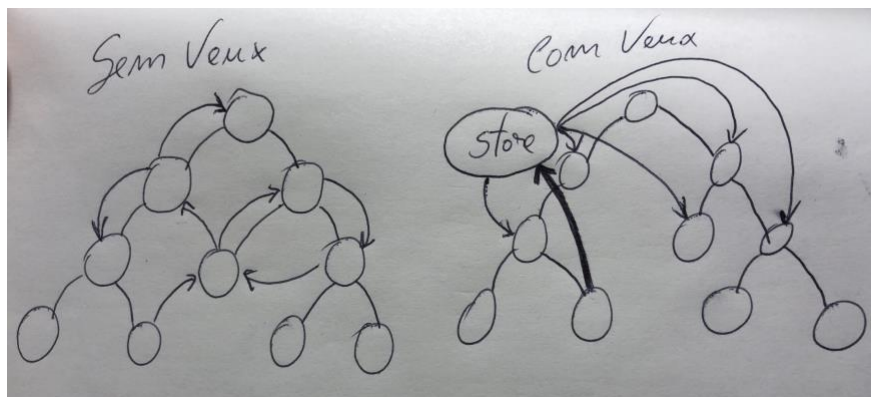


Figura 6 - Vuex - Esquema exemplo

O *Veux* é constituído por três componentes principais: *actions*, *mutations* e a *store*. Na *store* é onde fica localizado todo o código que funcionará como intermediário com a API e dados relacionados com a aplicação, ou seja, sempre que um componente tenha necessidade de obter dados será através da *store* que os consegue obter [26]. Quando se pretende alterar o estado de um objeto utilizamos as *mutations* que de forma síncrona é a responsável por alterar a *store*. As *actions* podem

¹⁸ Classe/objeto instanciada uma única vez e com visibilidade e acessibilidade global.

ser assíncronas e são de grande utilidade quando se pretende aceder à API.

3.5.5 Vue i18n



De forma a complementar e melhorar o projeto, ter uma estrutura que suporte diferentes línguas seria algo a ter em conta. Neste projeto fica o inglês e o português as línguas aplicadas, contudo alargar todo o projeto a outras línguas será fácil visto a estrutura estar já preparada para as acolher.

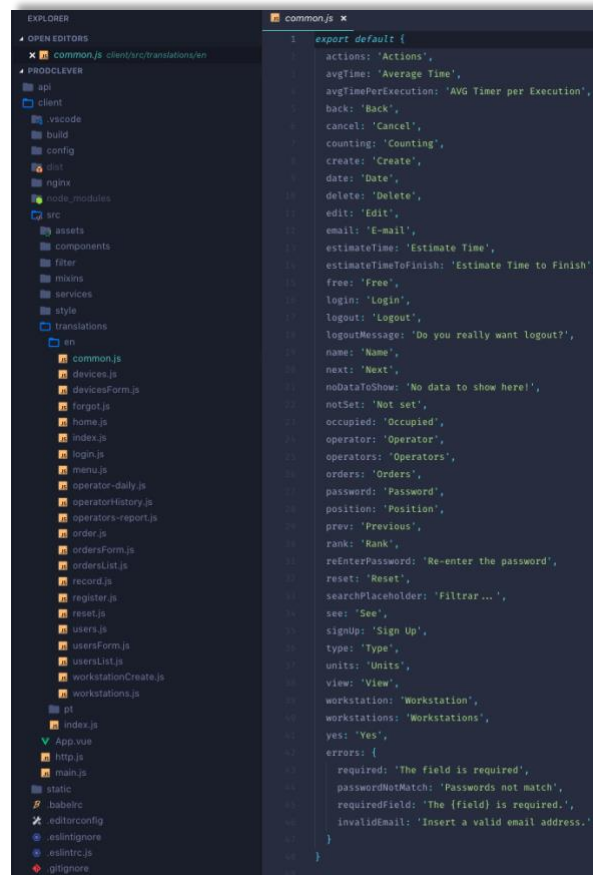



Figura 7 - Estrutura para os idiomas suportados

O módulo *Vue i18n* foi a ferramenta utilizada para facilitar a implementação multi-língua. Ferramenta muito simples de utilizar, como mostra a figura acima, pois como mostra a figura acima, dentro da pasta “*translations*” temos uma pasta para cada língua e dentro de cada pasta temos todas a palavras a serem traduzidas. O utilizador apenas tem que selecionar o idioma que prefere a tudo o resto é executado por esta ferramenta.

Com o módulo do *nodejs Vue i18n*, acrescentar idiomas é muito simples, pois basta adicionar uma pasta igual às existentes (en, pt) (figura acima) dentro da pasta “*translations*” e percorrer todas as pastas substituindo o conteúdo pelo idioma pretendido. Depois disso basta dar a conhecer a nova pasta no ficheiro *index.js* e colocar a opção do novo idioma na vista dos utilizadores. Este módulo é um caso flagrante para que se reutilize código já idealizado e funcional sem que se tenha que estar a criar tudo de raiz. Uma razão pela qual escolhi este módulo foi pelo facto de respeitar a norma internacional *i18n* [27], quanto a mim os módulos estarem em conformidade com as normas internacionais dá-lhe mais prestígio, integridade e consequentemente confiança para a sua utilização.

3.5.6 Axios

 Não existe nenhuma funcionalidade na plataforma **Prodclever** que não tenha necessidade de aceder à base de dados através da API. Este módulo permite fazer pedidos à API de forma simples, garantindo fluidez, rapidez e segurança. O Axios é um cliente “*http*” que tanto funciona no browser como em *node.js*, por isso temos uma ferramenta capaz de interagir com *xmlhttprequest*¹⁹ como com a interface *http* do *node.js* ao mesmo tempo, facilitando o desenvolvimento ajudando na simplicidade do código visto que o mesmo código tanto pode ser utilizado no cliente como no servidor. Além das vantagens de permitir *XMLHttpRequests* a partir do browser e pedidos *http* a partir do *node.js* este módulo suporta, também, *promises API*²⁰, gere os pedidos e respostas, transforma os dados dos pedidos e das respostas (ex: *JSON*) e protege o lado do cliente contra *XSRF*²¹ [28]. Em suma o módulo axios é uma solução segura e pela documentação disponível fiável do ponto de vista de maturidade e flexibilidade, permitindo assim uma abordagem facilitada quanto a preocupações de autenticação e pedidos/respostas à API.

¹⁹ API utilizada para aceder aos dados a partir do cliente

²⁰ Permite código assíncrono e gere os pedidos e respostas à API

²¹ Ataques informático que tem como propósito aproveitar a confiança dada aos websites

3.6 Backend

3.6.1 Express



Express trata-se de uma estrutura para aplicações *web* que facilita o desenvolvimento no `node.js`. Possibilita também a configuração de *middlewares*²² de forma a responder a pedidos *http* sem que haja a preocupação da segurança e interoperabilidade entre diferentes linguagens de código ou aplicações.

A Framework *Express* foi o suporte no desenvolvimento do Backend da plataforma **Prodclever**. Com o apoio desta ferramenta foi mais simples criar a API, a sua estrutura e saber o caminho a seguir para uma API devidamente desenhada e capaz de responder às expectativas. A diversidade de módulos que se podem aplicar no *express* torna a solução interessante ao ponto de ser reconhecida a nível mundial [29].

3.6.2 Mongo



A utilização de uma base de dados *noSQL* foi um desafio ganho, visto que para o contexto do projeto **Prodclever** não havia necessidade de uma base de dados demasiado rígida e carregadas de regras a seguir e dada a previsão da quantidade de informação que a plataforma suportará ao longo do tempo, visto este projeto funcionar como uma única base de dados para todas as empresas, daí e antecipação de problemas no que se refere ao tamanho e capacidade de resposta aos pedidos efetuados através da API. O grande objetivo era encontrar uma base de dados com grande capacidade de escalabilidade e disponibilidade, visto que numa utilização total da plataforma, como por exemplo, uma empresa com várias linhas de produção e dezenas de postos de trabalho em cada linha, a capacidade de receber todos os pedidos e manter a performance era fundamental.

Caso uma empresa adote esta solução, rapidamente retém milhares de registos na base de dados e a rapidez na disponibilidade da informação pedida pelos diferentes utilizadores é fundamental.

²² Software que possibilita a comunicação entre aplicações (Passport).

A pesquisa efetuada no sentido de encontrar uma base de dados que satisfaça o pretendido: escalabilidade, fiabilidade, integridade e segurança da informação, foi encontrada no mongoDB [30] [31]. A grande vantagem é que existem, mais uma vez, soluções que funcionam com o node.js.

3.6.3 Mongoose



Mongoose é um ODM (*Object Data Modeling*) para mongoDB e *node.js*. Este módulo permite a manipulação, interpretação e relações dos dados na base de dados mongoDB, permite também a utilização de esquemas para representação de objetos facilitando o desenvolvimento do código e toda a interação necessária com o mongoDB (base de dados) [32]. A criação de *Schemas*²³ permite a simplificação na utilização dos dados existentes no mongoDB, disponibilizando uma documentação simples com exemplos de forma que o foco do desenvolvimento se preocupe mais com outras perspetivas e não execuções “primárias” no mongoDB.

3.6.4 Passport



Passport é um *middleware*²⁴ para *node.js* já devidamente desenvolvido e testado que tem como objetivo tratar da autenticação numa solução como a **Prodclever** de forma rápida e simples. Este módulo foi utilizado (no *express.js*) com o intuito de gerir as sessões autenticadas na plataforma [33]. O desenvolvimento raiz seria muito trabalhoso e demasiados “pormenores” a ter em conta quando abordamos a questão da segurança, daí, e mais uma vez fica refletido que a reutilização de código já testado e devidamente desenvolvido só proporciona vantagens a quem o reutiliza.

3.6.5 Jwt



JSON Web Token foi utilizado para permitir a autenticação entre a API e a aplicação web, sem que existisse a necessidade de se utilizar cookies²⁵ ou sessões. Jwt é um conjunto de caracteres codificados que, caso o cliente e o servidor

²³ Esquemas das collections para uma correta interpretação da base de dados mongoDB

²⁴ Software que permite a comunicação e segurança entre aplicações

²⁵ Pequeno arquivo que retém informação acerca dos utilizadores de browsers

estejam sob o protocolo *https*, permite a leitura de um *token*²⁶ e assim uma autenticação fácil entre cliente e servidor [34].

```
http.js x user-controller.js
1 import axios from 'axios'
2
3 // configure Axios globals
4 axios.defaults.baseURL = process.env.API_URL
5 axios.defaults.headers.post['Content-Type'] = 'application/x-www-form-urlencoded'
6
7 // when the JWT token exists inject it on the request header
8 axios.interceptors.request.use((config) => {
9   // enable Cross Origin
10  config.withCredentials = true
11  config.headers['Access-Control-Allow-Origin'] = location.origin
12
13  const token = localStorage.getItem('auth-token')
14  if (token !== null) { config.headers['Authorization'] = `Bearer ${token}` }
15  return config
16 }, (error) => Promise.reject(error))
17
18 export default axios
```

Figura 8 - Código que coloca o token gerado nos headers

Como é exemplificado na imagem acima é gerado um *token* pelo servidor. Quando é autenticado e validado alguém do lado do cliente o *token* jwt é enviado nos cabeçalhos dos pedidos e a partir deste momento (se o *token* estiver válido) a API permite o acesso sem necessidade que haja necessidade de autenticação repetidamente.

²⁶ Sistema de segurança através de criação de chaves encriptadas

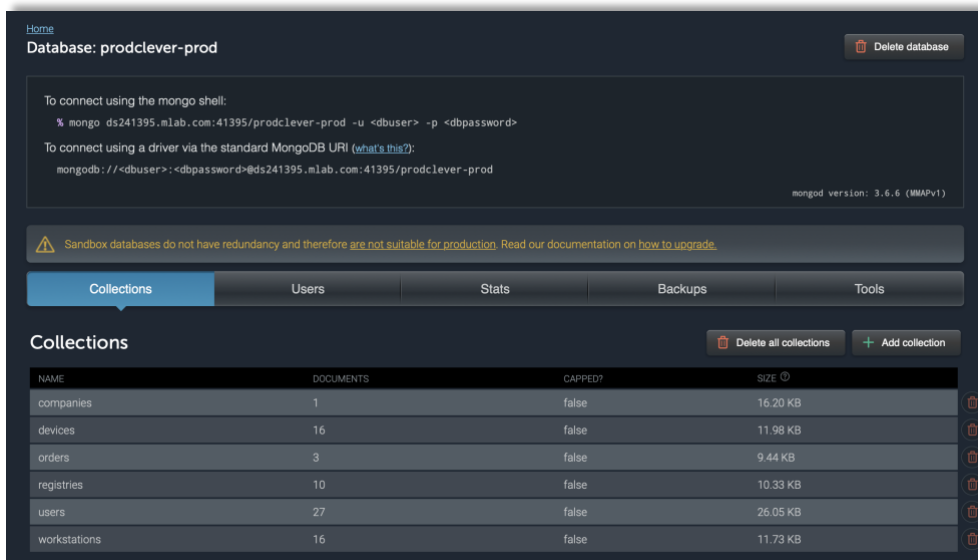
4 ProdClever

Neste ponto fica mais detalhado todos os pontos considerados essenciais para que a plataforma funcione assim como algoritmos e funcionalidades pensadas para tornar esta plataforma diferente e útil no mercado real. Inicialmente é abordada a parte lógica e explicado como os dados estão organizados na base de dados, de seguida são descritos algoritmos que importam referir e ideias que foram implementadas para otimizar e valorizar o trabalho desenvolvido.

Por fim é apresentada a plataforma na ótica do utilizador, mostradas todas as vistas importantes, e descrita toda a informação apresentada assim como estratégias e formas de apresentar a informação.

4.1 Base de dados

A disponibilidade e rapidez de resposta aos pedidos à base de dados são de extrema importância, visto que a utilização desta plataforma por uma empresa tende a ter um aumento de informação muito rápido e a escalabilidade é um problema a ter em conta. Para mitigar este potencial problema escolhi o serviço mlab para alojar a base de dados.



The screenshot displays the Mlab MongoDB interface for a database named 'prodclever-prod'. It includes connection instructions for the mongo shell and a standard MongoDB URI. A warning message states that sandbox databases do not have redundancy and are not suitable for production. Below this, there are navigation tabs for Collections, Users, Stats, Backups, and Tools. The 'Collections' tab is active, showing a table with the following data:

NAME	DOCUMENTS	CAPPED?	SIZE
companies	1	false	16.20 KB
devices	16	false	11.98 KB
orders	3	false	9.44 KB
registries	10	false	10.33 KB
users	27	false	26.05 KB
workstations	16	false	11.73 KB

Figura 9 - Estrutura da base de dados no mlab [35]

Mlab é um gestor de base de dados para mongoDB. Este serviço online garante que a base de dados fica sempre disponível, mantém backups dos dados, rapidez no acesso à informação e gere utilizadores com permissões de acesso ao serviço como é

representado na figura 10.

4.1.1 Coleções e documentos

Na figura 10 estão representadas as coleções de dados, como é denominado no mongoDB, que suportam toda a informação da plataforma **Prodclever**. Dentro das coleções temos os documentos, ou seja, cada registo dentro de cada coleção. Este conceito de bases de dados não relacional foi um desafio, pois esta forma de pensar as relações entre entidades e atualização da informação nos diferentes locais das coleções apesar de ser altamente escalável depois torna-se “complicado” fazer relações, aplicar *triggers*²⁷ e executar comportamentos como o *cascade*²⁸ em bases de dados relacionais.

4.1.1.1 Empresas

Na base de dados a coleção das empresas foi apelidada como “*Companies*”. Esta coleção tem a particularidade de gerir e separar toda a informação respeitante a cada empresa. Como esta plataforma é orientada a serviços para diferentes empresas todas as que se registarem vão partilhar a mesma base de dados e por isso é importante que cada uma apenas aceda à sua informação, daí a importância desta coleção. Cada documento na coleção “*Companies*” representa a informação de cada empresa.

```
1 {
2   "_id": {
3     "$oid": "5bbb3f3d26abac4076a3f75d"
4   },
5   "updatedAt": {
6     "$date": "2018-10-08T11:27:57.892Z"
7   },
8   "createdAt": {
9     "$date": "2018-10-08T11:27:57.892Z"
10  },
11  "name": "Organização 1",
12  "taxNumber": "123456789",
13  "description": "Empresa se produção de calçado",
14  "_v": 0
15 }
```

Figura 10 - Documento exemplo da “collection companies”

Atributos: “_id”, “updatedAt”, “createdAt”, “name”, “taxNumber” e “description”.

“_id” é o identificador único para cada empresa.

“updatedAt” e “createdAt” são atributos automaticamente criados pela base de

²⁷ Muito conhecido em bases de dados relacionais e dispara quando criado para algum evento

²⁸ Comportamento que é adotado na base de dados quando se faz determinada alteração

dados de forma a saber-se quando são criados e alterados os documentos.

“*Name*”, “*taxNumber*” são atributos que nos permite reconhecer a empresa.

“*description*” permite que haja a possibilidade de se poder identificar a empresa e a sua área de negócio ou alguma informação que a empresa pretenda incluir no seu registo.

4.1.1.2 Dispositivos

O nome desta coleção denomina-se de “*Devices*”. Cada empresa tem necessariamente que registar os seus dispositivos para que depois os possa relacionar ao posto de trabalho onde o dispositivo ficará localizado. Por uma questão de organização o dispositivo apenas pode ficar relacionado a um posto de trabalho, pois assim sempre que alguém se autentica nesse dispositivo sabe-se a que posto se refere e qual a empresa.

```
1 {
2   "_id": {
3     "$oid": "5bbb47bd26abac007ba3f789"
4   },
5   "updatedAt": {
6     "$date": "2018-10-08T12:04:13.883Z"
7   },
8   "createdAt": {
9     "$date": "2018-10-08T12:04:13.883Z"
10  },
11  "identifier": "353039094124710",
12  "description": "Linha A --> Posto 2 - Recorte dos modelos",
13  "workstation": {
14    "$oid": "5bbb3fb526abacfe50a3f760"
15  },
16  "company": {
17    "$oid": "5bbb3f3d26abac4076a3f75d"
18  },
19  "__v": 0
20 }
```

Figura 11 - Documento exemplo da “collection devices”

Atributos: “*_id*”, “*updatedAt*”, “*createdAt*”, “*identifier*”, “*description*”, “*workstation*”, “*company*”.

“*Identifier*” é o atributo que guarda o UUID²⁹ de cada dispositivo. Este identificador único disponível em cada dispositivo tem obrigatoriamente que ficar registado no atributo “*identifier*” quando pretendemos adicionar dispositivos á plataforma.

“*workstation*” “*company*” são atributos que permitem relacionar coleções e

²⁹ Identificar único de cada dispositivo.

posteriormente organizar a informação que é recolhida destes dispositivos.

“*description*” é o atributo definido para descrever qual a tarefa que o dispositivo vai registar.

4.1.1.3 Postos de trabalho

Uma das coleções de maior relevo visto ser por causa dos postos de trabalho e todo o trabalho que é executado no mesmo que esta plataforma faz sentido. Esta coleção está identificada como “workstations” na base de dados. Apesar de ter pouca informação e apenas 4 atributos pensados é nesta coleção que se pretende que seja estruturada e organizada o departamento de produção (esquema físico), mapeando-se as linhas de produção e localizando cada posto de trabalho de forma a que se saiba a origem da informação gerada.

```
1 {
2   "_id": {
3     "$oid": "5bbb3fa126abac2d8aa3f75f"
4   },
5   "updatedAt": {
6     "$date": "2018-10-30T11:27:07.324Z"
7   },
8   "createdAt": {
9     "$date": "2018-10-08T11:29:37.024Z"
10  },
11  "identifier": "Linha A --> Posto 1",
12  "task": "Desenho dos modelos na matéria prima",
13  "company": {
14    "$oid": "5bbb3f3d26abac4076a3f75d"
15  },
16  "occupied": true,
17  "__v": 0
18 }
```

Figura 12 - Documento exemplo da “collection workstation”

Atributos: “_id”, “updatedAt”, “createdAt”, “identifier”, “task”, “company”, “occupied”.

“_id” é o atributo que identifica de forma unívoca os documentos na coleção.

“identifier” localiza o posto de trabalho no departamento de produção das empresas. Esta identificação deverá ser efetuada com critério, pois organizações com muitos postos de trabalho poderá gerar desorganização e confusão.

“task” define o que tarefa é desenvolvida neste ponto de execução da linha de produção. Atributo meramente informativo, mas que quando associado a outra informação é útil e dá mais sentido às interpretações por parte de quem analisa a informação.

“company” como referido anteriormente existe a necessidade de relacionar em

toda a base de dados de forma a que a informação gerada seja canalizada para as empresas que a geram.

“*occupied*” tem como finalidade o controlo na associação de postos de trabalho às encomendas. É uma regra fundamental cada posto de trabalho ter definida a encomenda que está a ser executada no momento, caso contrário a informação gerada seria inconsistência e nada rigorosa na inferência que se pretende fazer posteriormente.

4.1.1.4 Encomendas

Esta coleção foi idealizada e pensada já numa fase mais avançada do projeto e que não era parte integrante da proposta inicial. Foi denominada de “*orders*” a coleção na base de dados e tem como principal objetivo facultar toda a informação sobre a evolução e histórico das encomendas.

```
1 {
2   "_id": {
3     "$oid": "5bd8400b272a30233835362d"
4   },
5   "updatedAt": {
6     "$date": "2018-10-30T11:27:07.312Z"
7   },
8   "createdAt": {
9     "$date": "2018-10-30T11:27:07.312Z"
10  },
11  "client": {
12    "$oid": "5bbb461126abac2886a3f77e"
13  },
14  "identifier": "ORD000001",
15  "quantity": 100,
16  "finalWorkstation": {
17    "$oid": "5bbb402226abac3bd3a3f765"
18  },
19  "company": {
20    "$oid": "5bbb3fd26abac4076a3f75d"
21  },
22  "state": "in_progress",
23  "workstations": [
24    {
25      "$oid": "5bbb3fa126abac2d8aa3f75f"
26    },
27    {
28      "$oid": "5bbb3fb526abacfe50a3f760"
29    },
30    {
31      "$oid": "5bbb3fd226abac3312a3f761"
32    },
33    {
34      "$oid": "5bbb3fe726abac395aa3f762"
35    },
36    {
37      "$oid": "5bbb3ffd26abac04c2a3f763"
38    },
39    {
40      "$oid": "5bbb401326abaca36ca3f764"
41    },
42    {
43      "$oid": "5bbb403726abac8565a3f766"
44    },
45    {
46      "$oid": "5bbb402226abac3bd3a3f765"
47    }
48  ],
49  "progress": 0,
50  "_v": 0
51 }
```

Figura 13 - Documento exemplo da “collection orders”

Atributos: “*_id*”, “*updatedAt*”, “*createdAt*”, “*cliente*”, “*identifier*”, “*quantity*”, “*finalWorkstation*”, “*company*”, “*state*”, “*workstations*”.

“*_id*” é o atributo que identifica univocamente um registo apenas na coleção como já referido anteriormente.

“*Cliente*” é o atributo que guarda o nome da empresa que encomendou o que está a ser produzido.

“*identifier*” é o registo interno que a organização atribui a cada encomenda que executa para os seus clientes.

“*quantity*” é o número de produtos que a organização vai fazer em cada encomenda. Este valor é importante pois é através dele que será calculado a progressão da encomenda. É também através deste valor que validamos se a encomenda e calculamos a média global para que se possa projetar no futuro a conclusão da encomenda em execução.

“*finalWorkstation*” este atributo é igualmente importante, pois é através dele que se consegue saber quantos produtos estão finalizados relativamente à encomenda em causa. Sempre que se gera uma encomenda é necessário relacionar todos os postos de trabalho que vão estar envolvidos na execução da encomenda, contudo e para que se saiba onde termina a linha de produção e conseqüentemente onde é finalizado o produto é necessário que se identifique o último posto de trabalho de forma a termos informação rigorosa acerca da evolução da encomenda.

“*state*” tem como finalidade devolver o estado das encomendas, em progresso ou acabadas. Sempre que o atributo “*quantity*” é igual ao número de tarefas executadas no atributo “*finalWorkstation*” o estado muda de “em progresso” para acabado.

“*workstations*” têm que ser sempre relacionadas quando se pretende que o dispositivo registre execuções de um posto de trabalho. Este atributo tem como função relacionar esta coleção à coleção “*workstations*”.

4.1.1.5 Utilizadores

Coleção que faz a gestão de todos os utilizadores. Denominada como “*users*” na base de dados. Esta coleção suporta vários tipos de utilizadores, contudo apenas consegue registar um administrador e é este utilizador que tem todas as permissões disponíveis na plataforma. Depois de registada a empresa e criado o respetivo administrador e só a partir deste momento será possível registar os restantes tipos de utilizador (cliente, consultor, encarregado e operador).

```

1 {
2   "_id": {
3     "$oid": "5bbb3f3d26abacc737a3f75e"
4   },
5   "updatedAt": {
6     "$date": "2018-10-08T11:27:57.928Z"
7   },
8   "createdAt": {
9     "$date": "2018-10-08T11:27:57.928Z"
10  },
11  "name": "admin",
12  "email": "admin@organizacao1.pt",
13  "password": "$2a$10$rbAeTveY1C8F2P76Py3fB01X6ytyHDJuH8f01Na6xepWoR2p1Nu",
14  "type": "admin",
15  "company": {
16    "$oid": "5bbb3f3d26abac4076a3f75d"
17  },
18  "__v": 0
19 }

```

Figura 14 - Documento exemplo da “collection users”

Atributos: “_id”, “updatedAt”, “createdAt”, “name”, “email”, “password”, “type”, “company”.

“_id” é o atributo que identifica de forma unívoca os documentos na coleção.

“name”, “email”, “password” são atributos comuns necessários a qualquer solução que se pretenda autenticação para que seja possível aceder à informação.

“type” é o atributo que permite o nível de acesso de cada utilizador. Diferentes tipos de utilizadores têm acesso a diferente informação.

“company” o atributo que separa a informação por empresas na plataforma e relaciona a coleção “companies” com esta.

4.1.1.6 Registos

A coleção mais crítica da plataforma, tanto pela proporcionalidade que esta coleção pode tomar como pelos pedidos que são feitos utilizando esta coleção. Esta coleção tende a crescer de forma exponencial, porque é nesta coleção que serão efetuados todos os registos de todos os postos de trabalho das diferentes empresas.

```

1 {
2   "_id": {
3     "$oid": "5bc1237caf41e1b69acabf1c"
4   },
5   "updatedAt": {
6     "$date": "2018-10-12T22:43:08.660Z"
7   },
8   "createdAt": {
9     "$date": "2018-10-12T22:43:08.660Z"
10  },
11  "user": {
12    "$oid": "5bc0eb617e15d457f074023b"
13  },
14  "workstation": {
15    "$oid": "5bbb3fa126abac2d8aa3f75f"
16  },
17  "time": 28,
18  "order": {
19    "$oid": "5bc12318af41e1d64acabf1a"
20  },
21  "__v": 0
22 }

```

Figura 15 - Documento exemplo da “collection registries”

Atributos: “_id”, “updatedAt”, “createdAt”, “user”, “workstation”, “time”, “order”.

“_id” é o atributo que identifica de forma unívoca os documentos na coleção.

“user”, “workstation”, “order” são dados importantes a registar pelo utilizador responsável de cada tarefa num determinado posto de trabalho na base de dados e que se encontra autenticado num dispositivo que por sua vez está relacionado com um posto de trabalho e com uma organização, assim como a encomenda correspondente associada.

“time” é um atributo que permitiu dar mais informação á plataforma. Inicialmente pretendia-se fazer contagens diárias e esse objetivo também foi alcançado, mas com o valor do tempo entre execuções de tarefas cíclicas e repetidas conseguimos saber quanto tempo é necessário para efetuar cada tarefa. Com estes tempos é possível saber médias de execução de tarefas em cada posto individual de trabalho e saber, por exemplo, qual o funcionário mais produtivo em cada posto de trabalho, visto existir relatórios que nos permite consultar listas de operadores que têm melhor média ao longo do sua vida como operador.

4.2 Aplicação mobile

O desenvolvimento da aplicação ubíqua foi um grande desafio visto nunca ter contacto com ferramentas *ios*³⁰ e ter que fazer a interoperabilidade entre os

³⁰ Sistema operativo para dispositivos móveis da Apple.

dispositivos apple e sua linguagem swift e a api em *node.js* baseado em *JavaScript*. Desafio esse que foi facilitado com as ferramentas já anteriormente mencionadas e que a seguir são demonstrados os dois algoritmos que importa salientar e que através do módulo *Alamofire* reduziu o código a escrever, pois apenas foi necessário utilizar as funções já existente e devidamente desenvolvidas.

```
func doInternalLogin(username: String, password: String) {
    let dataToSend = {
        "email": username,
        "password": password
    }

    Alamofire.request("https://api.prodclever.com/login", method: .post, parameters: dataToSend)
        .responseObject { response in
            switch response.result {
            case .success(let json):
                let jsonData = json as! [String: AnyObject];

                self.loginButton.isEnabled = true

                if jsonData["error"] != nil {
                    self.onLoginError()
                    return
                }

                if let token = jsonData["token"] as? String {
                    let preferences = UserDefaults.standard
                    preferences.set(token, forKey: "token")

                    DispatchQueue.main.async(execute: self.onLoginDone)
                }
            case .failure(_):
                self.loginButton.isEnabled = true
                self.onLoginError()
            }
        }
}
```

Figura 16 - Algoritmo de autenticação para o IOS

Na figura acima mostra como a livreria *Alamofire* foi útil, pois com poucas linhas de código foi possível estabelecer ligação com a API em *node.js* e receber a resposta e o respetivo *token* gerado pelo *jwt* módulo de gestão da autenticação [36].

A figura seguinte mostra o algoritmo que permite fazer os registos na base de dados e a simplicidade com que se consegue, através de *JSON*³¹, comunicar entre as duas linguagens distintas utilizadas (*swift* e *JavaScript*). *Alamofire*, mais uma vez, na base do desenvolvimento.

³¹ JavaScript Object Notation – Ferramenta que facilita a comunicação entre linguagens

```

@IBAction func onTapMark(_ sender: Any) {
    if self.timer == nil {
        self.markButton.setTitle("REGISTAR", for: [])
        self.initTimer()
        return
    }

    let parametersToSend: Parameters = [
        "identifier": Utils.getDeviceUUID(),
        "time": self.elapsedTime
    ];

    let headers: HTTPHeaders = [
        "Authorization": "Bearer " + Utils.getToken(),
    ]

    Alamofire.request(
        Utils.API_URL + "registry",
        method: .post,
        parameters: parametersToSend,
        headers: headers
    )
    .responseJSON { response in
        if let data = response.data, let utf8Text = String(data: data, encoding: .utf8) {
            print("Data: \(utf8Text)") // original server data as UTF8 string
        }

        switch response.result {
        case .success(_):
            print("Registered")
            self.elapsedTime = 0
            self.resetTimer()
        case .failure(let responseError):
            print("Register Error: ", responseError)
        }
    }
}

```

Figura 17 - Algoritmos que envia os registos para a base de dados

4.3 Tempo Real e Inferência da Informação

Um dos pontos propostos neste projeto seria ter em conta a informação em tempo real. Muitas soluções abordam o tempo real e é um porta estandarte da sua estratégia no mercado, contudo o tempo real que se referem é a partir do momento que a informação é inserida. O mais aproximado do tempo real que foi encontrado nas pesquisas e contactos efetuados foi a recolha de informação em todos os postos de trabalho e o seu registo manual na plataforma. Esta perspetiva já se aproxima do tempo real de execução das tarefas, mas não é de facto tempo real, pois não é no momento que a tarefa é executada que esta entra na informação a disponibilizar.

Nesta parte do documento pretende-se abordar a forma como foi garantida a informação em tempo real assim como toda a informação organizada para que se possa tirar o melhor partido quando é necessário tomar decisões.

4.3.1 Websockets

Obter e disponibilizar a informação em tempo real na plataforma foi um dos grandes desafios a alcançar neste projeto. Sempre que um operador regista as suas tarefas automaticamente toda a informação se ajusta em função do que é registado,

especialmente *dashboard's*³² e encomendas.

Cada utilizador, sempre que entra na plataforma, é apresentada informação que se encontra a ser atualizada em tempo real, especialmente a *dashboard* principal que é apresentada ao administrador, consultor e encarregado. Esta *dashboard* tem como finalidade mostrar a produtividade de cada posto de trabalho assim como quem está a trabalhar nesse mesmo posto ou o último operador que esteve registado. Também as encomendas são importantes que estejam a ser atualizadas em tempo real para que todos os intervenientes que têm acesso a essa informação tomem decisões sem desfasamentos ou ações tardias.

Este desafio foi ultrapassado com recurso á utilização de *websocket's*. Um dos grandes problemas da internet, que tem vindo a ser resolvido, é o dinamismo das páginas *http*, especialmente a atualização da informação em constante mudança onde apenas com a ação do utilizador é que as páginas se atualizavam, ou seja, neste caso seria necessário estar constantemente alguém a atualizar as páginas de forma a termos a informação em tempo real. Para que esta situação fosse ultrapassada recorri ao módulo para “*node.js*”, “*socket.io*”, que permite a gestão e atualização da informação.

```
const socket = SocketManager.getInstance().socket
socket.to(`order-${order._id}`).emit('updateProgress', {
  order: order._id,
  value: order.progress
})

// Initialize Vue Socket
Vue.use(VueSocketIO, io(process.env.API_URL), { store })

async socket_updateProgress ({ commit }, data) {
  commit(ORDER_UPDATE, data)
}

// create socketIO instance
const socket = require('socket.io')(server)
const socketManager = new SocketManager(socket)
socketManager.init()

watch: {
  orders (newVal) {
    const ids = newVal.map(item => item._id)
    this.$socket.emit('listenToOrders', ids)
  }
},

// listening for order updates
this.$socket.emit('listenToOrders', [this.$route.params.id])
}

watch: {
  orders (newValue) {
    const ids = newValue.map(item => item._id)
    this.$socket.emit('listenToOrders', ids)
  }
},

sockets: {
  updateProgress () {
    this.fetchDashboard()
  }
},

created () {
  const ids = this.orders.map(item => item._id)
  this.$socket.emit('listenToOrders', ids)
},
```

Figure 18 – Socket.io abordagens

³² Painel de bordo onde é apresentada a informação mais importante.

Uma das grandes mais valias do `node.js` é o facto de trabalhar de forma assíncrona, com pedidos e promessas evitando bloqueios devido à abordagem baseada em eventos o que torna esta solução (*node.js*) muito apelativa para se utilizar quando a intenção é informação em tempo real. Neste caso e como exemplifica a imagem acima foi instanciado e inicializado um *socket* que sempre que deteta alterações nas encomendas ou *dashboard* dispara (*emit*) um evento para atualizar as vistas em questão [37].

`Socket.io` é uma biblioteca em JavaScript criada para ser aplicada em projetos onde o objetivo é trabalhar com informação em tempo real, proporcionando a comunicação contínua entre cliente e servidor. A utilização desta biblioteca é possível com a existência da linguagem *html5*.

Instanciado e iniciado o *socket.io* no lado do servidor, este fica ativo e à espera de alteração nos conteúdos que estão a ser monitorizados. Existindo essas alterações que foram identificadas dentro do *socket*, neste caso as encomendas e a *dashboard* que suporta todos os postos de trabalho com a contagem diária, então é “disparado” um evento que altera a informação que está a ser vista do lado do cliente.

4.3.2 Médias

As médias calculadas em toda a plataforma são processadas no momento em que são solicitadas e não um valor guardado na base de dados. Devido à especificidade da informação e grande mudança nos registos que por sua vez alteram os resultados. É provável que sempre que se faz um pedido de consulta o valor no próximo pedido tenha mudado, daí a necessidade de existirem algoritmos para responder a esse pedido sempre que solicitado.

Exemplo: quando é calculada a previsão de conclusão de uma encomenda existe a necessidade de calcular diferentes médias. A média de conclusão de uma encomenda está sempre sujeita à evolução da encomenda em tempo real e a todas as médias calculadas em cada posto de trabalho.

```

const orderAvgTimeToFinish = async order => {
  const workstationsAvgTime = await workstationsOrderAvg(order._id)
  const allEstimateTimes = workstationsAvgTime.map(entry => entry.timeToFinish)
  const maxTime = Math.max(...allEstimateTimes)

  return maxTime
}
exports.orderAvgTimeToFinish = orderAvgTimeToFinish

```

Figura 19 - Algoritmo para cálculo do tempo de conclusão de encomendas

O algoritmo acima na figura é um excelente exemplo de que não seria boa prática guardar valores de médias visto os valores serem voláteis.

O cálculo para finalização de uma encomenda começa por criar um *array*³³ com todas as médias dos postos de trabalho que estão associadas à encomenda referida “*workstationsAvgTime*”, ou seja, a função (*workstationsAvgTime*) tem como finalidade agregar todas as médias de todos os registos relacionados com os postos de trabalho e encomenda em causa. De seguida cria um novo *array* que guarda todos os tempos estimados de conclusão “*AllEstimateTime*”. Neste algoritmo (*AllEstimateTime*) verifica-se quantas tarefas já foram executadas, subtrai-se pelo número total da encomenda e obtém-se o que ainda falta executar em cada posto de trabalho. Obtido o número de execuções em falta para cada posto de trabalho multiplica-se por cada média calculada na função anterior e é guardado no array os valores de tempo em falta em cada posto de trabalho. O algoritmo da figura acima devolve o maior valor guardado no *array* “*AllEstimateTime*”, pois é esse tempo que falta para a conclusão da encomenda.

O exemplo acima mencionado é flagrante a preocupação relativamente á questão de agregar toda a informação necessária para que sejam rigorosas as respostas a dar aos diferentes utilizadores.

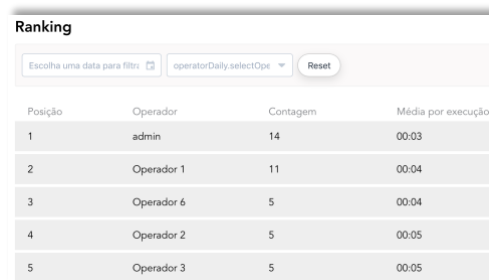
É importante referir que os tempos para todas as médias calculadas como: média dos operadores nos diferentes postos de trabalho, média global dos postos de trabalho, médias para serem apresentadas em rankings, advêm da coleção “*registries*”, pois é nesta coleção que são registadas todas as tarefas e guardados os tempos exatos em cada execução, daí as médias calculadas e os tempos previstos serem rigorosos e reais, facilitando assim o trabalho tanto dos intervenientes que pretendem otimizar a produção e alocar os recursos humanos mais produtivos para os

³³ Estrutura para armazenamento de dados

postos em que estes têm melhor média de execução como os intervenientes administrativos e de gestão que pretendem saber os custos rigorosos dos seus recursos humanos e responsáveis pelos cálculos da contabilidade analítica.

4.3.3 Rankings

Identificar os operadores que são mais produtivos em cada posto de trabalho é algo que poderá beneficiar as organizações em termos de capacidade produtiva, pois quando nos comparamos normalmente ficamos competitivos e queremos sempre os melhores resultados comparativamente aos outros. Além desta competitividade que beneficia as organizações também a organização consegue retirar o máximo partido de cada operário porque pode distribuir os seus recursos humanos tendo em conta os *ranking's* disponibilizados na plataforma otimizando e exponenciando a sua capacidade ao máximo possível, por conta da performance de cada um em cada posto de trabalho.



Posição	Operador	Contagem	Média por execução
1	admin	14	00:03
2	Operador 1	11	00:04
3	Operador 6	5	00:04
4	Operador 2	5	00:05
5	Operador 3	5	00:05

Figura 20 - Ranking de operadores

Como é exemplificado na imagem acima, todos os postos de trabalho têm uma vista individual onde uma das áreas é destinada a mostrar todos os operários que laboraram lá e quem tem melhor média a executar as tarefas.

4.3.4 Pesquisas e histórico

Dado o crescimento previsto para uma plataforma deste tipo é essencial que existam ferramentas disponíveis que nos permita encontrar a informação pretendida de forma rápida e simples. Esta funcionalidade para as pesquisas já tem resolução no *node.js*, pois basta colocar a estrutura abaixo exemplificada para podermos procurar informação na base de dados. Devido ao ficheiro “.vue” já existem os devidos acessos

à informação na área de *scripting*³⁴ respeitante à vista em questão.

```
<PcToolbar>
  <el-input
    v-model="filter"
    class="users-list__search"
    :placeholder="$t('common.searchPlaceholder')" />
</PcToolbar>
```

Figura 21 - Estrutura para pesquisa de utilizadores

Esta funcionalidade de pesquisa foi extremamente útil, pois não houve a necessidade de aprofundar as *query's*³⁵ para pesquisas na base de dados e além disso a forma de pesquisa é muito rápida, permite pesquisar em diferentes colunas das tabelas que são apresentadas e a partir do segundo carácter introduzido no campo de pesquisa o algoritmo de pesquisas procura imediatamente combinações e à medida que é inserida informação os conteúdos vão-se ajustando dinamicamente como demonstra na imagem seguinte.

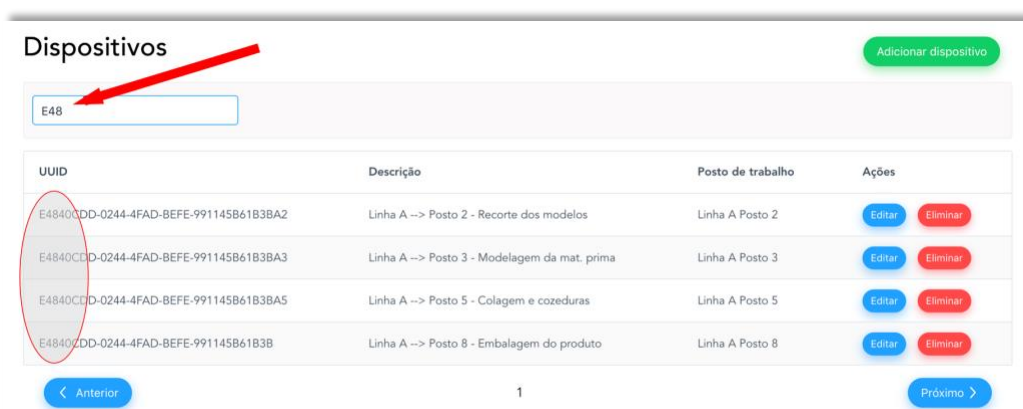


Figura 22 - Exemplo de pesquisa de dispositivos

Nesta plataforma as pesquisas também podem ser feitas por uma data ou entre datas (imagem abaixo). Especialmente quando se trata de obter informação histórica acerca de produtividade ou trabalho feito nos postos de trabalho faz sentido que se possa analisar os comportamentos de forma cronológica e por isso também existem pesquisas disponíveis para datas que se pretendam recuperar (imagem abaixo) o que já foi feito. Esta área de histórico é uma outra funcionalidade que diferencia esta plataforma do que existe de momento no mercado, pois com as ferramentas

³⁴ Tarefas de chamadas à API (neste caso) ou execução de funções lógicas

³⁵ São instruções que se podem executar em bases de dados

disponíveis, atualmente, o histórico está a ser negligenciado no médio longo prazo devido à forma como se regista e gere a informação (através de ficheiros excel ou bloco de notas).

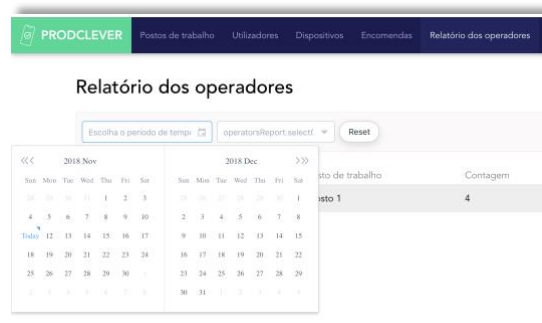


Figura 23 - Vista de pesquisas por data

Tudo o que os operadores registarem na plataforma não será perdido e será apresentado sempre que as pesquisas dos intervenientes sejam no sentido de recuperar o que foi feito no passado. Com isto é possível ter análises muito mais fiáveis e rigorosas sobre os postos de trabalho e respetivos operadores colaboradores das organizações.

4.4 Vistas dos utilizadores

Este ponto do relatório é dedicado a apresentar de forma visual as vistas mais importantes da plataforma e dar uma breve explicação dos pormenores que a tornam apelativa e funcional.

Todas as funcionalidades foram devidamente desenvolvidas e testadas.

4.4.1 Login

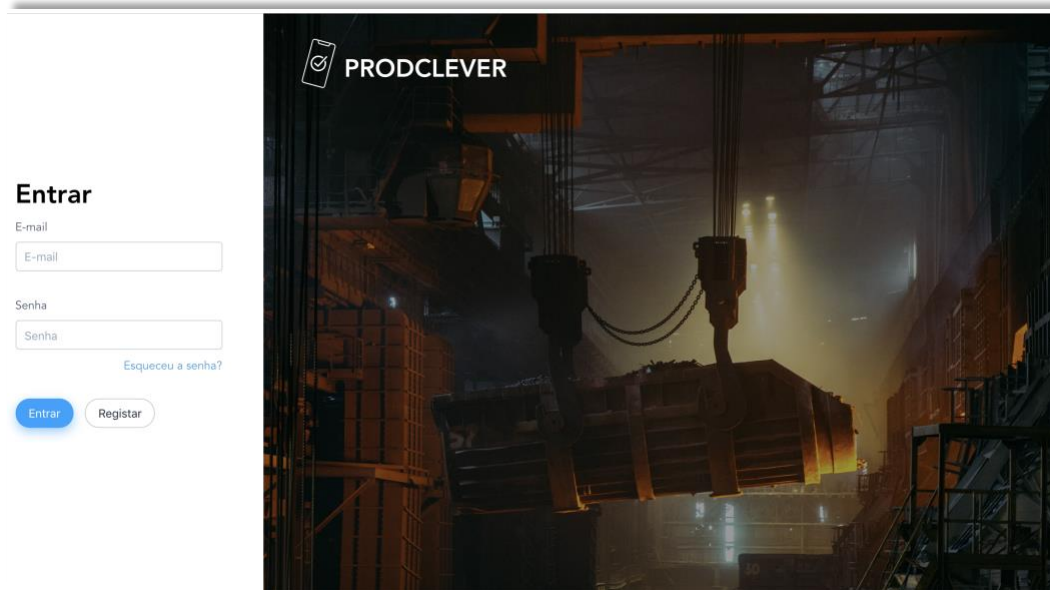


Figura 24 - Vista de Login

A vista acima é a entrada na plataforma. É a primeira vista a ser acedida sempre que se pretende entrar em contas já existentes ou registar nova conta.

Quando um utilizador novo pretende dar início à utilização da plataforma tem que se registar e no mesmo registo terá que criar o administrador de toda a plataforma. Apenas será possível ter um administrador. A partir do administrador criado serão criados todos os outros utilizadores, encomendas, postos de trabalho, dispositivos e encomendas.

Para que a gestão de utilizadores ficasse o mais completa possível também foi desenvolvida e configurada a opção de recuperação da senha de acesso.

4.4.2 Dashboard dos postos de trabalho

Desde o início do projeto a vista mais importante. A ideia central é registar e apresentar a produtividade que se executa em cada posto de trabalho assim como identificar quem está ou esteve nesse posto. A preocupação desta vista era que fosse o mais simples possível e que mostre apenas o essencial.

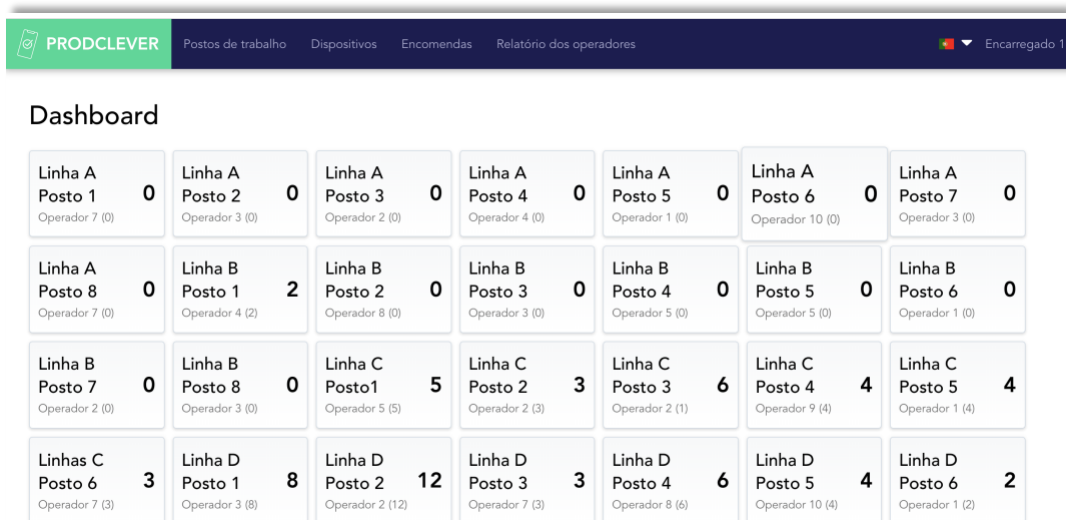


Figura 25 - Dashboard dos registos diários dos postos de trabalho

Nesta vista são apresentados todos os registos diários de todos os postos de trabalho da organização. Cada posto de trabalho é representado por um quadrado que mostra a identificação do posto, o registo diário da produção feita e na parte de baixo em letra mais pequena identifica o operador que está a trabalhar ou o último que esteve. Existem duas contagens a decorrer em cada posto de trabalho, uma (em tamanho maior) que soma a contagens dos diferentes operadores que possam trabalhar nesse local, outra que está entre parêntesis a seguir ao nome do operador que apenas mostra a contagem do mesmo. Todos os utilizadores têm acesso a esta vista exceto o utilizador cliente.

4.4.3 Posto de trabalho

Saber o comportamento e tempos que se praticam nos postos de trabalho é de extrema importância para as organizações e é nesta vista que está apresentada informações mais interessantes e que pode mudar o comportamento e decisões de quem se interessa pela capacidade produtiva das empresas.

Workstation: Linha B Posto 1

Tarefa
Desenho dos modelos na matéria prima

Estado
Ocupado

Ranking

Escolha uma data para filtrar: operatorDaily.selectOpe

Posição	Operador	Contagem	Média por execução	Data
1	Operador 9	9	00:05	11/11/18
2	Operador 5	4	00:06	11/11/18
3	Operador 1	8	00:07	11/11/18
4	Operador 4	2	00:17	11/11/18

Figura 26 - Vista individual do posto de trabalho

Como é demonstrado na vista acima, cada posto tem uma vista individual onde se pode consultar a lista dos operadores com melhor média na sua laboração neste local de trabalho. Também é possível validar quantas execuções um operador já fez no total desde sempre no posto de trabalho selecionado e a respetiva média que precisa para executar cada ciclo individual de trabalho.

Por uma questão de apoio à orientação e organização é possível validar se o posto de trabalho está livre ou não, pois sempre que se associa um posto de trabalho a uma encomenda este fica indisponível para poder ser associado a outra encomenda.

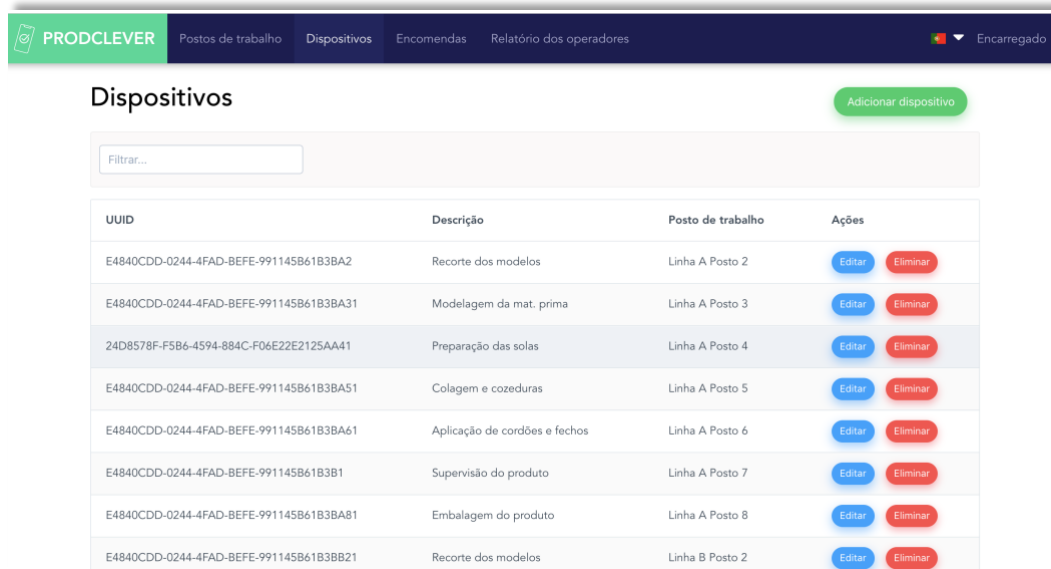
4.4.4 Dispositivos

A única entrada de informação para a plataforma é através dos dispositivos e por isso é importante a gestão destes de forma a que não haja desorganização e confusão. Pode-se registar vários dispositivos e para isso basta registar o UUID do dispositivo, descrever qual o intuito que lhe queremos dar e associa-lhe um posto de trabalho. Os dispositivos não têm necessariamente que ficar desde logo destinados a um posto de trabalho ou com tarefa definida, poderá ser registado na plataforma e ficar em “standby” até que seja necessário.

É de extrema importância que o UUID do dispositivo fique devidamente registado na plataforma, porque é através deste identificador único que os dispositivos móveis possuem que vai permitir que haja proteção e certeza que apenas um posto está relacionado com apenas com um dispositivo. A aplicação móvel envia

sempre o UUID nos registos a gravar na base de dados e caso não esteja devidamente escrito todos os registos enviados serão rejeitados.

É sempre necessário ter um dispositivo com um posto de trabalho associado, pois só assim é que poderá haver contagem.



UUID	Descrição	Posto de trabalho	Ações
E4840CDD-0244-4FAD-BEFE-991145B61B3BA2	Recorte dos modelos	Linha A Posto 2	Editar Eliminar
E4840CDD-0244-4FAD-BEFE-991145B61B3BA31	Modelagem da mat. prima	Linha A Posto 3	Editar Eliminar
24D8578F-F5B6-4594-884C-F06E22E2125AA41	Preparação das solas	Linha A Posto 4	Editar Eliminar
E4840CDD-0244-4FAD-BEFE-991145B61B3BA51	Colagem e cozeduras	Linha A Posto 5	Editar Eliminar
E4840CDD-0244-4FAD-BEFE-991145B61B3BA61	Aplicação de cordões e fechos	Linha A Posto 6	Editar Eliminar
E4840CDD-0244-4FAD-BEFE-991145B61B3B1	Supervisão do produto	Linha A Posto 7	Editar Eliminar
E4840CDD-0244-4FAD-BEFE-991145B61B3BA81	Embalagem do produto	Linha A Posto 8	Editar Eliminar
E4840CDD-0244-4FAD-BEFE-991145B61B3BB21	Recorte dos modelos	Linha B Posto 2	Editar Eliminar

Figura 27 - Vista da lista dos dispositivos móveis

Esta vista permite organizar a lista de dispositivos e identificar qual o posto associado ou se este está livre e poderá ser associado a um posto. A qualquer momento é possível editar o dispositivo e ajustar o UUID, alterar o posto de trabalho ou até mudar a descrição da tarefa a que está destinado.

4.4.5 Encomendas

Esta área de encomendas foi uma das funcionalidades que foram ganhando forma ao longo do desenvolvimento do projeto e sem dúvida acrescenta valor à plataforma dada a informação disponível na vista.

Identificador	Cliente	Progresso	Tempo estimado	Ações
ORD000001	Cliente 1	25 / 100	11:26	Ver
ORD000002	Cliente 2	24 / 50	04:06	Ver
ORD000003	Cliente 1	3 / 30	11:56	Ver
ORD000004	Cliente 1	2 / 50	15:12	Ver

Figura 28 - Vista da lista de encomendas

Nesta vista é possível consultar todas as encomendas, em progresso ou acabadas, quais os clientes relacionados com a encomenda, o progresso onde a encomenda se encontra, caso ainda não esteja concluída e o tempo estimado para conclusão da encomenda. Este tempo refere-se ao tempo útil em trabalho contínuo.

A vista seguinte refere-se à informação individual de cada encomenda e que devolve uma excelente perspetiva para que se possa analisar sob o ponto de vista da produtividade a evolução do trabalho desenvolvido.

Nome da empresa Cliente 1		25%	100/25
Postos de trabalho			
Linha A Posto 1 Desenho dos modelos na matéria prima	Linha A Posto 2 Recorte dos modelos	Linha A Posto 3 Modelagem da matéria prima	Linha A Posto 4 Preparação das solas
Linha A Posto 6 Aplicação de cordões e fechos	Linha A Posto 8 Embalagem do produto	Linha A Posto 7 Supervisão do produto	Linha A Posto 5 Colagem e cozeduras

Operadores		
Posto de trabalho	Operador	Tempo médio
Linha A Posto 2 (Recorte das moldes)	Operador 3	00:03
Linha A Posto 2 (Recorte das moldes)	Operador 4	00:04
Linha A Posto 7 (Supervisão do produto)	Operador 3	00:07
Linha A Posto 7 (Supervisão do produto)	Operador 4	00:02
Linha A Posto 3 (Modelagem da matéria prima)	Operador 2	00:07
Linha A Posto 3 (Recorte das moldes)	admin	00:03
Linha A Posto 5 (Colagem e costuras)	Operador 1	00:07
Linha A Posto 2 (Recorte das moldes)	Operador 1	00:04
Linha A Posto 2 (Recorte das moldes)	Operador 2	00:05
Linha A Posto 4 (Preparação das solas)	Operador 4	00:10
Linha A Posto 8 (Embalagem do produto)	Operador 4	00:07
Linha A Posto 8 (Embalagem do produto)	Operador 1	00:09

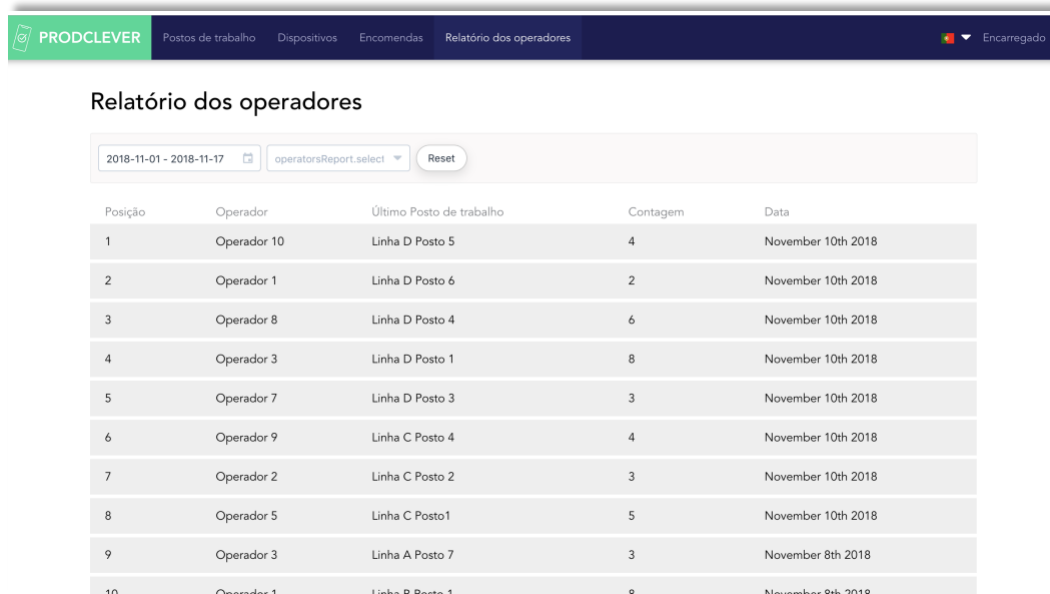
Postos de trabalho		
Posto de trabalho	Média por execução	Tempo estimado para conclusão
Linha A Posto 7 (Supervisão do produto)	00:03	03:03
Linha A Posto 6 (Aplicação de cordões e fechos)	00:06	07:05
Linha A Posto 2 (Recorte dos moldes)	00:04	04:42
Linha A Posto 8 (Embalagem do produto)	00:08	08:51
Linha A Posto 3 (Modelagem da matéria prima)	00:07	08:12
Linha A Posto 5 (Colagem e costuras)	00:07	08:33
Linha A Posto 4 (Preparação das solas)	00:10	11:26

Figura 29 – Vista individual da encomenda

Analisar individualmente e de forma detalhada a informação recolhida ajuda no sentido de tomar decisões e orientar recursos da melhor forma para que se consiga atingir objetivos propostos. A informação retida nesta vista é a seguinte:

- Identificação da encomenda através de uma referência;
- Identificação do cliente para a qual a encomenda está a ser executada;
- Postos de trabalho que estão a ser utilizados para a encomenda;
- Qual o posto final que dá como concluído cada produto;
- Postos de trabalho trabalhadores que estão a trabalhar na encomenda e também a média de cada operador;
- Posto de trabalho com a média global e tempo previsto para conclusão da encomenda em cada posto de trabalho.

4.4.6 Relatório dos operadores



Posição	Operador	Último Posto de trabalho	Contagem	Data
1	Operador 10	Linha D Posto 5	4	November 10th 2018
2	Operador 1	Linha D Posto 6	2	November 10th 2018
3	Operador 8	Linha D Posto 4	6	November 10th 2018
4	Operador 3	Linha D Posto 1	8	November 10th 2018
5	Operador 7	Linha D Posto 3	3	November 10th 2018
6	Operador 9	Linha C Posto 4	4	November 10th 2018
7	Operador 2	Linha C Posto 2	3	November 10th 2018
8	Operador 5	Linha C Posto1	5	November 10th 2018
9	Operador 3	Linha A Posto 7	3	November 8th 2018
10	Operador 1	Linha B Posto 1	8	November 8th 2018

4.4.7 Utilizadores

Na plataforma existem 5 tipos de utilizadores: administrador, encarregado, operador, consultor e cliente. A gestão de permissões é de fácil configuração e por isso as regras implementadas neste momento de acesso às funcionalidades por parte dos utilizadores podem ser alteradas caso se entenda sem que se tenha que fazer grandes alterações.

Administrador

O utilizador administrador refere-se apenas a cada empresa e não à plataforma no seu todo, como referido anteriormente, para que seja criada uma empresa há a obrigatoriedade de incluir os dados do administrador da empresa que terá acesso a todas as vistas e funcionalidades da plataforma para a empresa em questão.

The screenshot shows a web interface for creating a user. At the top, there is a navigation bar with the logo 'PRODCLEVER' and several menu items: 'Postos de trabalho', 'Utilizadores', 'Dispositivos', 'Encomendas', and 'Relatório dos operadores'. The user is logged in as 'admin'. The main heading is 'Criar utilizador' followed by 'Funcionalidades' in a larger, orange font. Below the heading, there is a '<< Back' link. The form consists of four fields: 'Nome', 'E-mail', 'Senha', and 'Tipo de utilizador'. The 'Tipo de utilizador' field is a dropdown menu with a 'Select' button and a list of options: 'Cliente', 'Consultor', 'Encarregado', and 'Operador'. The 'Encarregado' option is currently selected.

Figura 30 - Vista de criação de utilizador e todas as funcionalidades

Relativamente a funcionalidades e às suas restrições no controlo de toda a plataforma, apenas o administrador tem a opção de criar os diferentes tipos de utilizadores de forma a que não haja conflitos e as políticas de inclusão de intervenientes seja o mais controlada possível.

Encarregado

O utilizador “encarregado” é o interveniente mais importante para que a plataforma funcione devidamente. É ele que tem a responsabilidade de organizar e orientar toda a informação que passará pela plataforma. Caso este utilizador não organize e estrutura o departamento de produção na plataforma devidamente esta poderá ser confusa e pode fazer com que a informação gerada possa ter problemas de integridade e confiabilidade.

O encarregado tem a responsabilidade de alocar postos de trabalho a encomendas e dispositivos a postos de trabalho. Este utilizador tem acesso a todas as funcionalidades e vistas da plataforma exceto gestão de utilizadores.

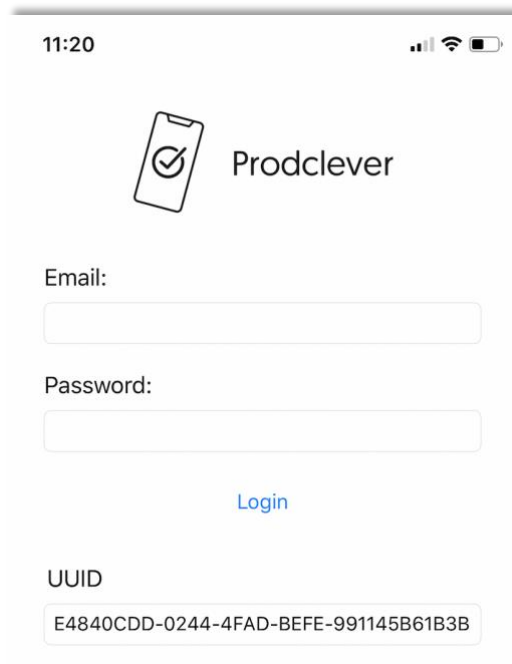
Operador

O operador é o único utilizador que tem funcionalidades tanto na aplicação mobile como em *browser's*. As funcionalidades do operador são poucas, simples e objetivas de modo a que este não se “perca” em burocracias ou funcionalidades que poderão prejudicar o seu foco e produtividade.

Na componente mobile o operador apenas consegue registar execuções de

tarefas como é demonstrador nas imagens e descrições seguintes:

Login:



11:20

Prodclever

Email:

Password:

Login

UUID

E4840CDD-0244-4FAD-BEFE-991145B61B3B

Figura 31 - Vista de login para o operador

Esta é a primeira vista (acima) quando é clicado o ícone da aplicação. Nesta autenticação apenas o operador tem a capacidade de registar para efetuar o registo do seu trabalho nos diferentes postos de trabalho.

Sempre que um operador se autentica a primeira ligação à API é no sentido de validar o operador, enviar o UUID do dispositivo de forma a validar o registo na base de dados do dispositivo e localizar qual o posto de trabalho associado ao mesmo. Toda esta informação validada é devolvida a resposta ao dispositivo com toda esta informação e o respetivo *token* fornecido pela API e gerido pelo módulo *JWT* (Módulo já abordado anteriormente) de forma a que gira e facilite o acesso e autenticação à API sempre que solicitado.

Esta vista também expõe o UUID do dispositivo. Esta foi a forma encontrada de facilitar o administrador sempre que pretenda registar um dispositivo na base de dados, pois assim basta abrir a aplicação e retirar o UUID correspondente.

Utilizador autenticado:



Figura 32 - Vista inicial com operador autenticado

Esta será a vista com o utilizador autenticado e como se pode confirmar o que já antes foi referido, a aplicação não oferece grandes funcionalidades ou opções para que a distração com o dispositivo seja mínima. Neste ponto o operador tem o dispositivo pronto a executar os registos em conformidade com o trabalho que vai fazendo. Sempre que o operador termine a sua tarefa apenas tem que clicar na zona preta para que um registo seja enviado à API identificando-o a si, dispositivo, posto de trabalho associado e tempo de receção do registo. Além dos registos que pode ir executando ao longo do seu trabalho o operador tem a possibilidade de pausar o tempo de execução da tarefa ou sair da aplicação.

Em funcionamento:

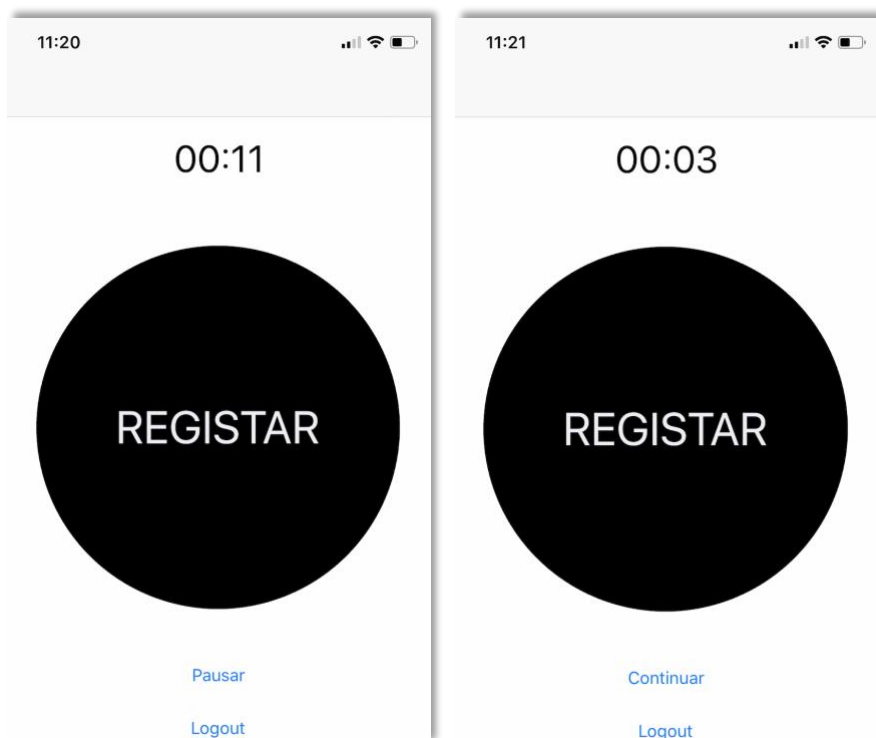


Figura 33 – Vistas do operador em IOS "Pausa" e "Continuar"

Por fim ficam acima demonstradas as vistas em funcionamento. A mudança entre vistas apenas acontece nas palavras que indicam diferentes execuções. Depois da opção *"Iniciar"* logo a seguir à autenticação passa a esperar registos e a opção *"Pausar"* comuta com *"Continuar"* dependendo da opção executada.

Consultor

Este utilizador foi pensado para incluir intervenientes que necessitam da informação na ótica da gestão e da contabilidade, ou seja, pessoas que necessitam da informação para calcular custos como contabilistas ou administrativos e pessoas que têm a pretensão de controlar a produção e respetivas capacidades produtivas das suas empresas.

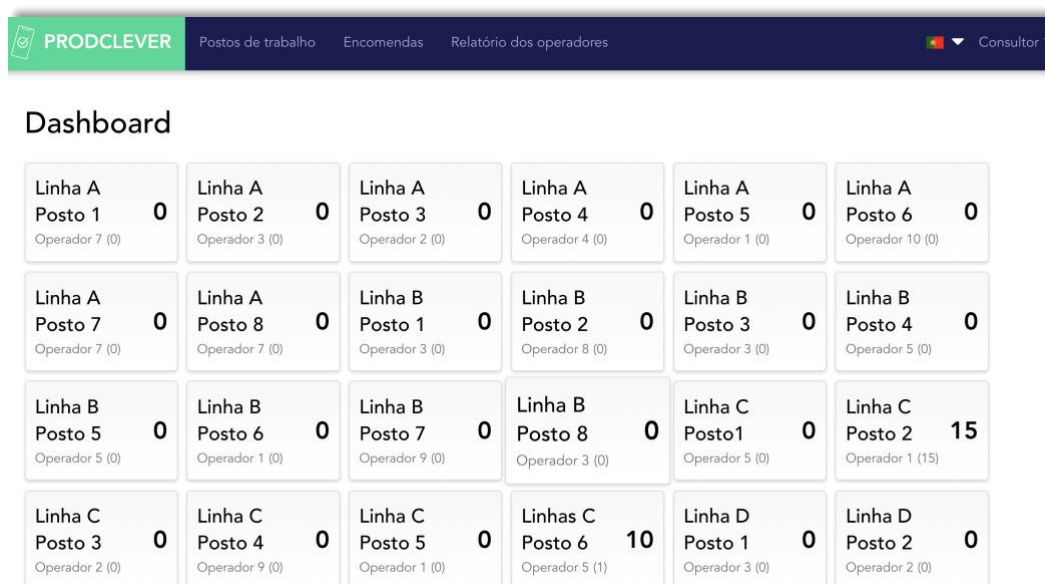
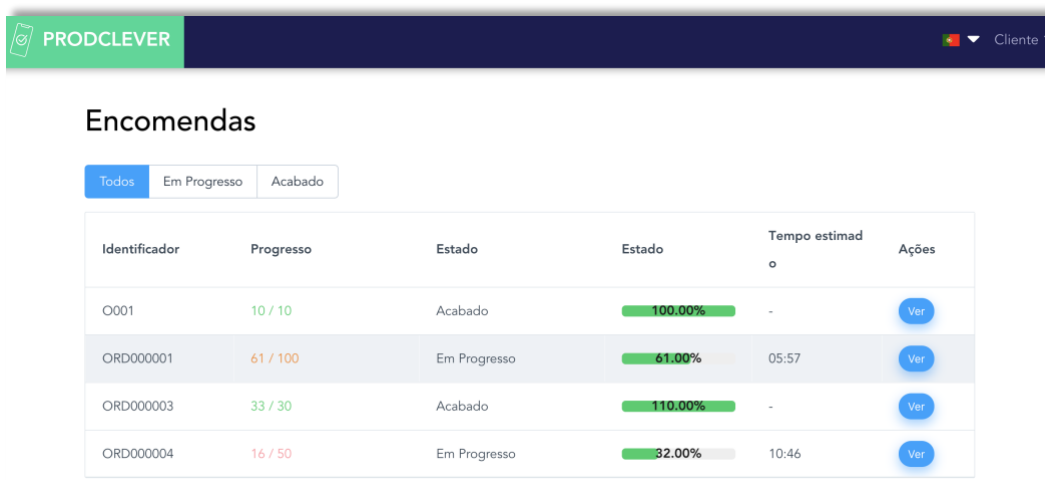


Figura 34 - Vista inicial do utilizador consultor

O consultor tem acesso a toda a informação acerca da produção no chão de fábrica. Este utilizador não tem qualquer opção de criação ou edição da informação, apenas pode consultar a informação disponível. Na sua vista aquando da sua autenticação é apresentada a dashboard principal com os postos de trabalho e nas vistas secundárias tem disponível todas as encomendas, postos de trabalho, trabalho executado pelos operários e relatórios existentes na plataforma.

Cliente

O utilizador “cliente” não estava incluído nos planos iniciais, mas com o evoluir da solução a ideia surgiu e disponibilizar o acesso e respetiva informação não seria tão complexo como inicialmente poderia parecer. Dada a estrutura desenvolvida foi fácil acrescentar o utilizador “cliente” e alocar-lhe a informação necessária de forma a estar atualizado acerca das suas encomendas. Incluir os clientes nos processos da empresa e informar o mesmo em real time através desta opção torna o serviço muito mais interessante para quem contrata os serviços e a qualidade prestada é muito maior, pois manter clientes informados constantemente com certeza que satisfaz.



The screenshot shows the PRODCLEVER client interface. At the top, there is a header with the PRODCLEVER logo and a user profile for 'Cliente 1'. Below the header, the title 'Encomendas' is displayed. There are three tabs: 'Todos', 'Em Progresso', and 'Acabado'. The main content is a table with the following columns: Identificador, Progresso, Estado, Estado, Tempo estimado, and Ações. The table contains four rows of order data.

Identificador	Progresso	Estado	Estado	Tempo estimado	Ações
O001	10 / 10	Acabado	100.00%	-	Ver
ORD000001	61 / 100	Em Progresso	61.00%	05:57	Ver
ORD000003	33 / 30	Acabado	110.00%	-	Ver
ORD000004	16 / 50	Em Progresso	32.00%	10:46	Ver

Figura 35 - Vista inicial do utilizador cliente

Na imagem acima é apresentada a vista do cliente caso a empresa pretenda disponibilizar um acesso ao cliente. Nesta vista o cliente tem sempre acesso, tanto às encomendas que estão a decorrer e que são para ele, como a todo o histórico de encomendas processadas para o cliente.

Além de listadas todas as encomendas e agrupadas por: “Em progresso” e “Acabado”, também é fornecido ao cliente o número de produtos acabados / número total de produtos encomendados, o seu estado, a percentagem de conclusão e o tempo

útil estimado para conclusão de cada encomenda.

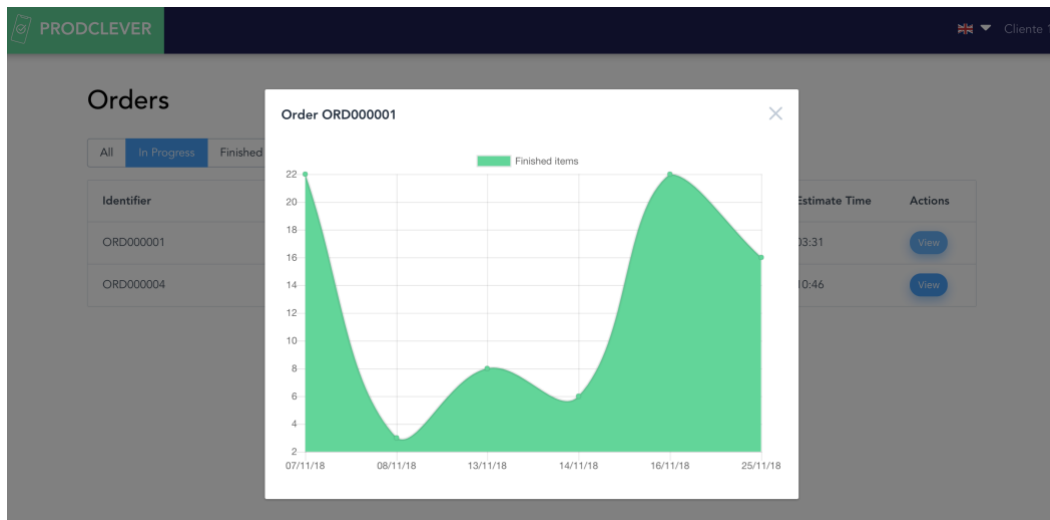


Figure 36 - Vista do gráfico relativo á produção diária da encomenda

A opção “Ver”, é um botão que dá acesso a um gráfico onde é possível verificar o número de peças concluídas diariamente. Esta opção de visualização sumária da encomenda permite validar:

- se a produção e recursos alocadas são contínuos com pequenas oscilações;
- se houve alguma quebra drástica na produção ou se simplesmente pararam de produzir a encomenda.

Poderá ser fornecido ao cliente esta ferramenta para que supervise tudo o que acontece em termos de produtividade com as suas encomendas. Claro que fornecer esta ferramenta ao cliente acarreta riscos e só será aconselhável caso haja confiança nos prazos a cumprir e na continuidade diária de produção das encomendas que ficam sob supervisão pelos clientes caso lhes seja facultada esta opção.

5 Conclusões e Trabalho Futuro

O principal objetivo deste trabalho era colmatar uma lacuna existente na indústria produtiva relativamente ao controlo de custos e capacidade produtiva por parte dos recursos humanos das empresas do setor industrial de transformação e produtivo.

Concluído todo o trabalho estou certo que os objetivos foram atingidos e com a convicção de que resultou algo interessante e útil para se aplicar no mercado.

Esta ideia, proposta inicialmente deixou-me reticente quanto à sua abrangência e capacidade de se tornar em algo que se pudesse transformar em tese de mestrado, contudo com o evoluir das ideias para este trabalho e através de pesquisas feitas percebi que de facto existe um imenso caminho a percorrer acerca desta problemática. Abordar este problema na perspetiva da quarta revolução industrial, ou indústria 4.0 é uma excelente estratégia visto que é precisamente acerca da implementação das novas tecnologias que esta revolução orienta e é neste ponto que reside o problema proposto a resolver.

Com este trabalho tive também a oportunidade de ter contacto com ferramentas de programação mais atuais e que me levaram a concluir que conceitos como reutilização, componentes e interoperabilidade são cada vez mais utilizados e vejo com bons olhos. As comunidades que surgem para a disponibilização de ferramentas úteis com excelente documentação e que nos permitem focar no que realmente importa para o negócio. No meu caso e neste trabalho seria impossível conseguir abordar todos os pormenores como: funcionalidades, segurança, usabilidade, interoperabilidade, tempo real, etc.... não fosse o *node.js*, por exemplo, e todos os seus componentes que permitiram aproveitar o que já foi desenvolvido por programadores e que agora aproveitei focando-me mais nos objetivos iniciais e que deram origem a esta tese. Todas as ferramentas que tive a oportunidade de utilizar são soluções muito maduras e que deram muita segurança ao resultado final, no sentido de saber que os algoritmos adotados foram devidamente desenvolvidos e que a evolução e as diferentes versões já disponíveis colmatam “erros” anteriores e melhorias bastante otimizadas. Caso não fosse possível a adoção de ferramentas já desenvolvidas, a execução deste trabalho teria muitas mais vulnerabilidade e não seria possível executar tudo o que ficará disponível nesta solução. A documentação muito explícita e de fácil orientação, também foi um a ajuda inquestionável ao longo

de todo o trabalho.

Também numa perspetiva contabilística e de gestão foi possível ter mais noção das dificuldades de contabilizar custos de mão de obra humana. Atualmente, o cálculo da mão de obra humana é calculado essencialmente através de coeficientes de imputação que é possível apurar através da analítica, mas não chega porque cada produto tem as suas especificidades assim como cada humano e por isso existe a necessidade do rigor de se identificar o real custo dos produtos.

Quando desenvolvemos um projeto é normal que, ao longo do seu desenvolvimento, surjam ideias e funcionalidades que melhorem o trabalho, contudo a abrangência e o tempo de execução não permite que se esteja constantemente a adotar-se e alterar o foco do que é proposto. Este trabalho não foi exceção e senti que o desenvolvimento de uma APP androide faz todo o sentido por questões de comunidade (a maior a nível mundial). A forma como o operador coloca as credencias no dispositivo que tem no posto de trabalho também deve ser melhorada, visto que neste momento apenas aceita o login por email e password. Aceitar a entrada na plataforma através de impressão digital ou reconhecimento facial tornará autenticação mais fácil e rápida. É uma funcionalidade a desenvolver, contudo a grande dificuldade para que não fosse explorada esta opção prende-se com o facto de haver necessidade de se poder gerir vários utilizadores num dispositivo só. Para que a opção ficasse em funcionamento seria necessário conseguir-se registar várias impressões digitais ou rostos no mesmo dispositivo e no momento o que se consegue é registar apenas uma pessoa e a resposta que sai do dispositivo é apenas um “true” ou “false” a comprovar que é o dono do equipamento e não quem é que se autentica. O desenvolvimento futuro seria gerir vários utilizadores dentro do dispositivo e conseguir-se enviar à API mais dados além de um “true” ou “false”. Outro problema a melhorar será a forma como se calcula a finalização das encomendas, pois se queremos reforçar a produtividade de uma encomenda com mais postos de trabalho estratégicos onde a produção está a ser mais lenta poderá haver a necessidade de dividir ou incluir a produção conjunta de forma a que o calculo seja ajustado com o reforço.

Existem melhorias a fazer-se para que esta solução seja realmente uma mais valia e mais flexibilidade, mas penso que poderá tirar-se muitas conclusões e avançar de forma mais sólida neste tipo de soluções. Quando me foi proposto este trabalho desconhecia a potencialidade que se poderiam retirar do mesmo, mas agora finalizado estou certo de que existe um campo muito fértil para se melhorar e dar mais

ferramentas numa área tão sensível que é o chão de fábrica das empresas.

6 Referências

- [1] J. B. e. a. Carvalho, “Temas de Contabilidade Pública,” *Rei dos Livros*, 1999.
- [2] T. P. B. O. Mario Hermann, “Design Principles for Industrie 4.0 Scenarios,” 2016. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/7427673>.
- [3] C. B. Silveira, “Automação Industrial, Conceitos de Automação, Principios da Indústria 4.0,” 11 10 2016. [Online]. Available: <https://app.uff.br/riuff/handle/1/4868>. [Acedido em 10 7 2017].
- [4] K. Schwab, *A Quarta Revoluyção Industrial*, Levoir, 2018.
- [5] Oregional, Abril 2018. [Online]. Available: <https://oregional.com.br/wp-content/uploads/2018/04/emprego-caged-1.jpg>.
- [6] S. T. Jiafu Wan, “Defined Industrual Internet of Things in the Context of Industry 4.0,” em *IEEE Sensors Journal*, 2016.
- [7] C. Schroth, “Converging Concepts Enabling the Internet of Services,” em *Web 2.0 and SOA*, University of Gallen, 2007.
- [8] L. F. D. Júnior, *Programação web com Node.js*, Kindle, 2017.
- [9] Scaleway, 2018. [Online]. Available: <https://cloud.scaleway.com/#/service>.
- [10] CloudFlare, CloudFlare, 2018. [Online]. Available: <https://developers.cloudflare.com/docs/>.
- [11] [Online]. Available: <https://www.iredmail.org>.
- [12] iRedMail, “Install iRedMail on Debian or Ubuntu Linux,” [Online]. Available: <https://docs.iredmail.org/install.iredmail.on.debian.ubuntu.html>.
- [13] L. F. D. Júnior, *Node.js e Microservices*, eBook Kindle, 2017.

- [14] Node.js, “Node.js v11.2.0 Documentation,” [Online]. Available: <https://nodejs.org/api/documentation.html>.
- [15] L. V. C. S. Isaac Z. Schlueter, npm, Inc., [Online]. Available: <https://www.npmjs.com/package/nodemon>.
- [16] J. Foundation, “ESLint,” npm, inc, 2018. [Online]. Available: <https://www.npmjs.com/package/eslint>.
- [17] S. Hykes, “Docker CE for Debian,” Docker, Inc., 2013. [Online]. Available: <https://docs.docker.com/install/linux/docker-ce/debian/>.
- [18] N. Provos, “node.bcrypt.js,” npm, inc, [Online]. Available: <https://www.npmjs.com/package/bcrypt>.
- [19] C. Lattner, “The powerful programming language that is also easy to learn.,” Apple, 2014. [Online]. Available: <https://developer.apple.com/swift/>.
- [20] “Swift Dependency Management for iOS,” [Online]. Available: <https://medium.com/xcblog/swift-dependency-management-for-ios-3bcfc4771ec0>.
- [21] E. Durán, “Getting Started,” [Online]. Available: <https://guides.cocoapods.org/using/getting-started.html>.
- [22] J. Shier, “Alamofire HTTP networking library,” Alamofire Software Foundation, [Online]. Available: <https://github.com/Alamofire/Alamofire/blob/master/README.md>.
- [23] 2012. [Online]. Available: <https://webpack.js.org/concepts/>.
- [24] “Vue.js 2.0 UI Toolkit for Web,” [Online]. Available: <https://github.com/ElmFE/element>.
- [25] E. Y. (尤雨溪), “Vue.js Guide,” 2014. [Online]. Available: <https://vuejs.org/v2/guide/>.
- [26] “Vaux.js,” 2015. [Online]. Available: <https://vuex.vuejs.org>.

- [27] “Introduction to i18n,” Copyright © 1999-2001 Tomohiro KUBOTA, 2018. [Online]. Available: <https://www.debian.org/doc/manuals/intro-i18n/>.
- [28] “Axios package,” <https://www.sitepoint.com/axios-beginner-guide/>, [Online]. Available: <https://www.npmjs.com/package/axios>.
- [29] T. Holowaychuk, “Expressjs Documentation,” Node.js, 2010. [Online]. Available: <http://expressjs.com/>.
- [30] L. F. D. Júnior, Mongo DB para iniciantes, 2009.
- [31] M. Inc., “Modules anda packages,” 2009. [Online]. Available: <https://www.npmjs.com/package/mongo-express>.
- [32] S. Lyubka, “Mongoose documentation,” 2018. [Online]. Available: https://mongoosejs.com/docs/api.html#mongoose_Mongoose.
- [33] S. Jung, “WEB DEVELOPMENT WITH NODE.JS,” *Journal of Computing Sciences in Colleges*, p. 155, 2018.
- [34] <https://jwt.io/introduction/>, “JSON Web Token (JWT),” 2015. [Online]. Available: <https://www.rfc-editor.org/rfc/pdf/rfc7519.txt.pdf>.
- [35] mLab, “MongoDB Deployments,” mLab, [Online]. Available: <https://mlab.com/>.
- [36] J. Shier, “Alamofire Reference,” Alamofire Software Fondation, [Online]. Available: <http://cocoadoes.org/docsets/Alamofire/4.0.1/>.
- [37] G. Rauch, “Socket.IO library,” [Online]. Available: <https://socket.io/docs/>.

Fim