



Estágio Curricular
Académica da Madeira
2022/2023

Plataforma de Recursos Humanos e de Projetos



Trabalho elaborado por:
Hélvio Nóbrega nº 2077720

Universidade da Madeira, 19 de junho de 2023

Índice

1. Introdução.....	2
2. Solução.....	3
3. Revisão de Literatura.....	7
3.1. Ferramentas de Desenvolvimento.....	7
3.1.1. MySQL Workbench.....	7
3.1.2. Figma.....	7
3.1.3. API Google Calendar.....	8
3.1.4. GitHub.....	8
3.2. Tecnologias Utilizadas.....	8
3.2.1. Laravel.....	8
3.2.2. Symfony.....	9
3.2.3. Docker.....	10
3.3. Tecnologias Alternativas.....	10
3.3.1. Next.js.....	10
3.3.2. Blazor.....	11
4. Implementação.....	12
4.1. Colaboradores.....	14
4.2. Login.....	14
4.3. Projetos.....	14
4.3.1. Geral.....	14
4.3.2. Meus Projetos.....	16
4.4. Calendário.....	16
4.4.1. Meus Turnos.....	16
4.4.2. Validação.....	16
5. Avaliação.....	18
6. Discussão.....	20
7. Conclusão.....	22
8. Referências.....	24
9. Anexo A.....	25

1. Introdução

A gestão de recursos humanos e de projetos são fundamentais em qualquer organização.

A área de recursos humanos é uma área estratégica dentro de qualquer empresa. As empresas reconhecem que o seu sucesso está diretamente relacionado com a qualidade e dedicação dos seus colaboradores, pois lida com processos fundamentais como a formação, recrutamento, planeamento de forma a melhorar as capacidades e habilidades dos colaboradores e, da mesma forma, mantê-los motivados e comprometidos com a empresa. E tudo isso tem que ser feito de acordo com a legislação trabalhista e as melhores práticas do setor. [1]

Por outro lado, a gestão de projetos também é fundamental para o sucesso de uma empresa. No entanto, organizar, planejar, controlar, garantir que sejam concluídos a prazo enquanto satisfazem os requisitos dos stakeholders pode ser uma tarefa complexa, principalmente quando a empresa tem vários projetos em andamento simultaneamente. [2]

A importância de plataformas gratuitas e *opensource* nesta gestão é inegável. Em muitos casos, a falta dessas ferramentas cria um obstáculo considerável para pequenas empresas ou, como é o caso, empresas sem fins lucrativos que não têm recursos financeiros para investir em soluções de software por preços relativamente altos. Além disso, muitas delas têm necessidades únicas que os sistemas de gestão prontos para uso no mercado não conseguem atender.

É neste contexto que surge a necessidade de desenvolver uma plataforma de gestão de recursos humanos e de projetos *opensource* adaptável a qualquer empresa. Desta forma, não preencherá apenas uma lacuna significativa no mercado, mas também permitirá a muitas empresas gerir eficientemente e de forma organizada os seus recursos humanos e projetos, mesmo com recursos limitados.

2. Solução

Existindo uma escassez notável de plataformas *opensource* voltadas para esta área, deixa muitas empresas sem uma solução eficiente e acessível para gerir todas as suas operações.

Neste sentido, foi desenvolvida uma plataforma distribuída pelas duas áreas, ou seja, gestão de recursos humanos e gestão de projetos que no fundo se complementam.

Dentro dos primeiros passos para desenvolver a solução foi pensada e estruturada a base de dados (Figura 1) com muito rigor para que futuramente as alterações necessárias fossem mínimas.

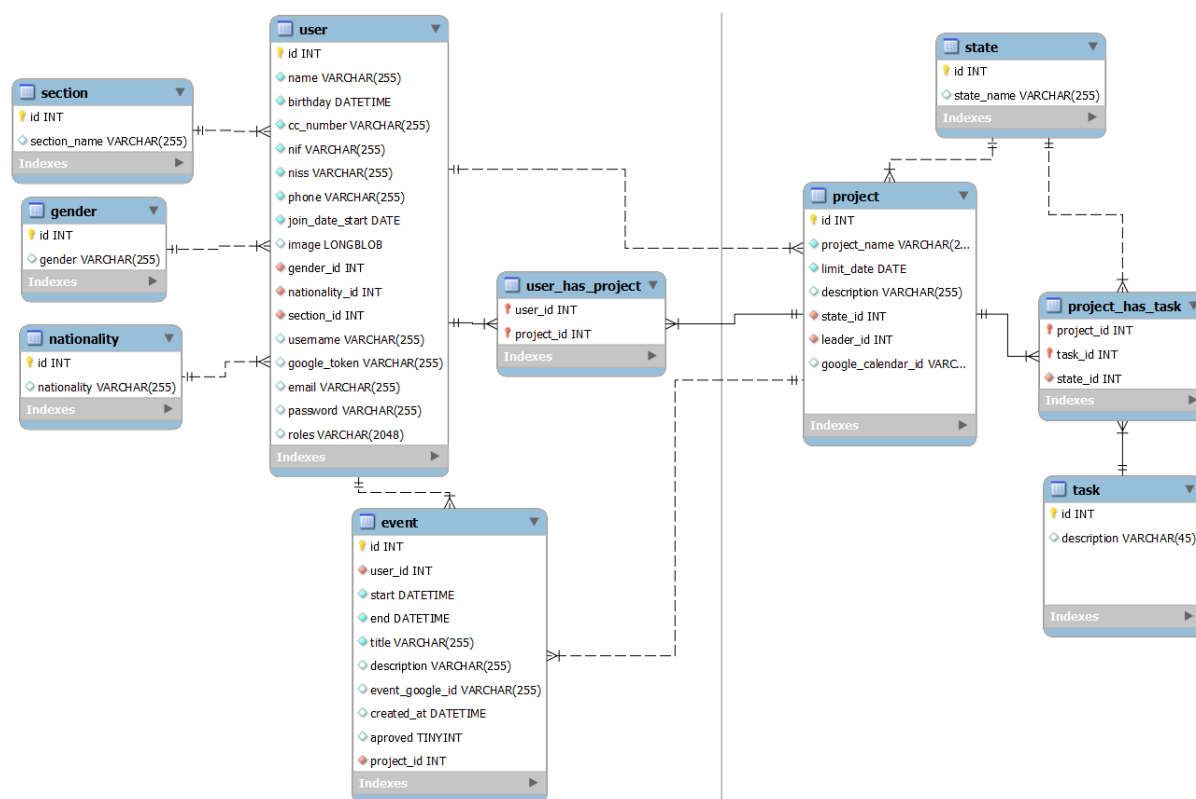


Figura 1 : Diagrama Entidade-Relação

Inicialmente, tivemos em conta a existência de uma interface intuitiva sempre com atenção à usabilidade. Isso garantirá que os utilizadores possam navegar e utilizar a plataforma com facilidade, mesmo aqueles que não possuem habilidades técnicas avançadas, sendo algo que enfrentamos na empresa e, para isso, priorizamos uma template que nos é familiar de modo a manter este problema de parte e continuarmos com o padrão de simplicidade e a clareza na apresentação das informações e nas funcionalidades oferecidas.

Como primeiro passo na implementação foi estruturada a autenticação quer por parte dos utilizadores quer por parte da administração que terá diversos privilégios tal como, criação de conta para novos colaboradores, criação de projetos, criação de turnos de trabalho, entre outros.

A plataforma, de modo a solucionar o problema apresentado é constituída por várias categorias sendo uma delas a “Colaboradores” capaz de armazenar e aceder informações sobre os colaboradores de forma rápida e segura.

Integra-se também as categorias “Projetos” e “Calendário” onde é trabalhado ambas as áreas, sendo a “Projetos” responsável por fazer toda a gestão de projetos desde a criação de um projeto denominando o responsável, colaboradores e tarefas a serem executadas até a data limite de conclusão do mesmo. Além disso é feita uma gestão das tarefas já concluídas e as por concluir de modo a que a equipa de trabalho esteja a par de todo o processo a ser desenvolvido no respetivo projeto resultando numa maior produtividade por parte do grupo que estarão atualizados a todo o momento do estado em que o projeto se encontra.

Complementando com o “Calendário”, é aqui que se encontra a solução de grande valor pois a plataforma tornou-se capaz de tratar problemas que a empresa enfrenta com a marcação de turnos dos colaboradores e principalmente com a sobreposição de turnos, por consequência da falta de um sistema que fosse capaz de melhorar toda a gestão relacionada com os turnos dado à grande quantidade de colaboradores. Tornei todo este processo mais prático, útil por toda a empresa e extremamente eficaz (Figura 2).

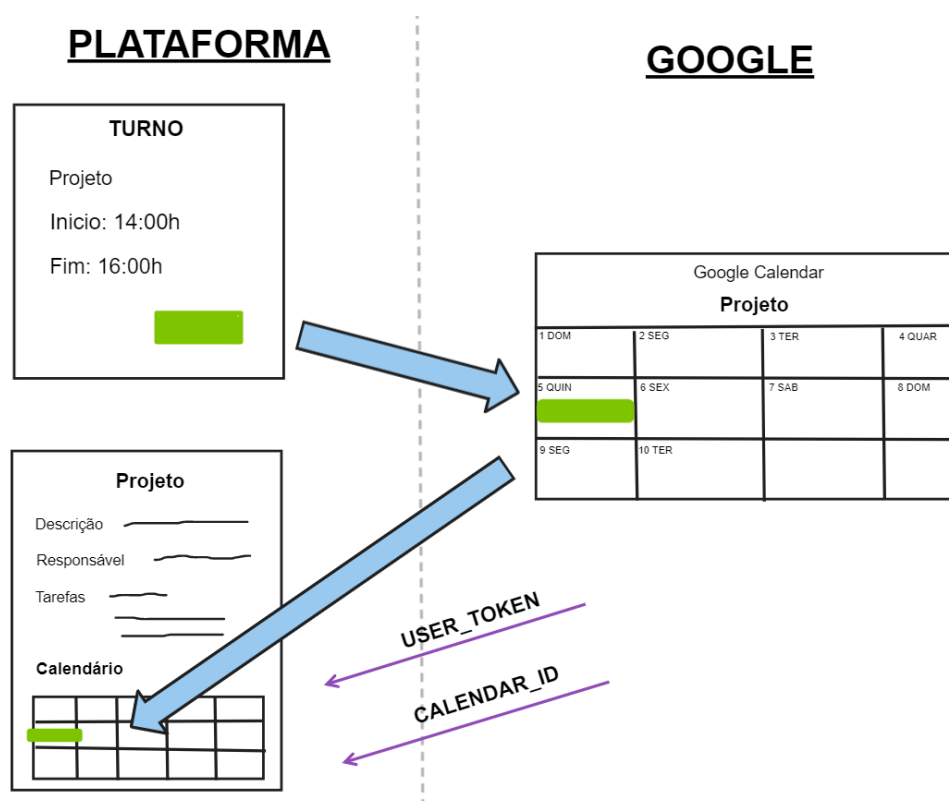


Figura 2: Comunicação entre a plataforma e a API

Para isso, foi utilizada a API da Google, nomeadamente o Google Calendar, ao qual um colaborador terá apenas de aceder à plataforma, pedir um turno de acordo com a sua disponibilidade tendo em conta as horas disponíveis apresentadas no calendário do respetivo projeto que o próprio colaborador irá trabalhar. Após isso será recebido por um administrador ou responsável do projeto que irá analisar e posteriormente validar ou recusar o pedido dependendo se o turno requisitado for de encontro ou não com a disponibilidade do calendário do projeto.

Neste sentido, foi desenvolvido um diagrama use cases para a plataforma. Este diagrama, na Figura 3, ilustra os atores envolvidos no sistema (administradores e colaboradores) e para além disso, apresenta cada use case com o intuito de captar as principais funcionalidades e oferecer uma visão geral clara das capacidades disponíveis para os utilizadores, ou seja, os use cases descrevem uma ação específica que pode ser exercida dentro da plataforma como pedir turno, criar projeto, entre outros. De realçar que um administrador consegue também exercer qualquer use case de um colaborador.

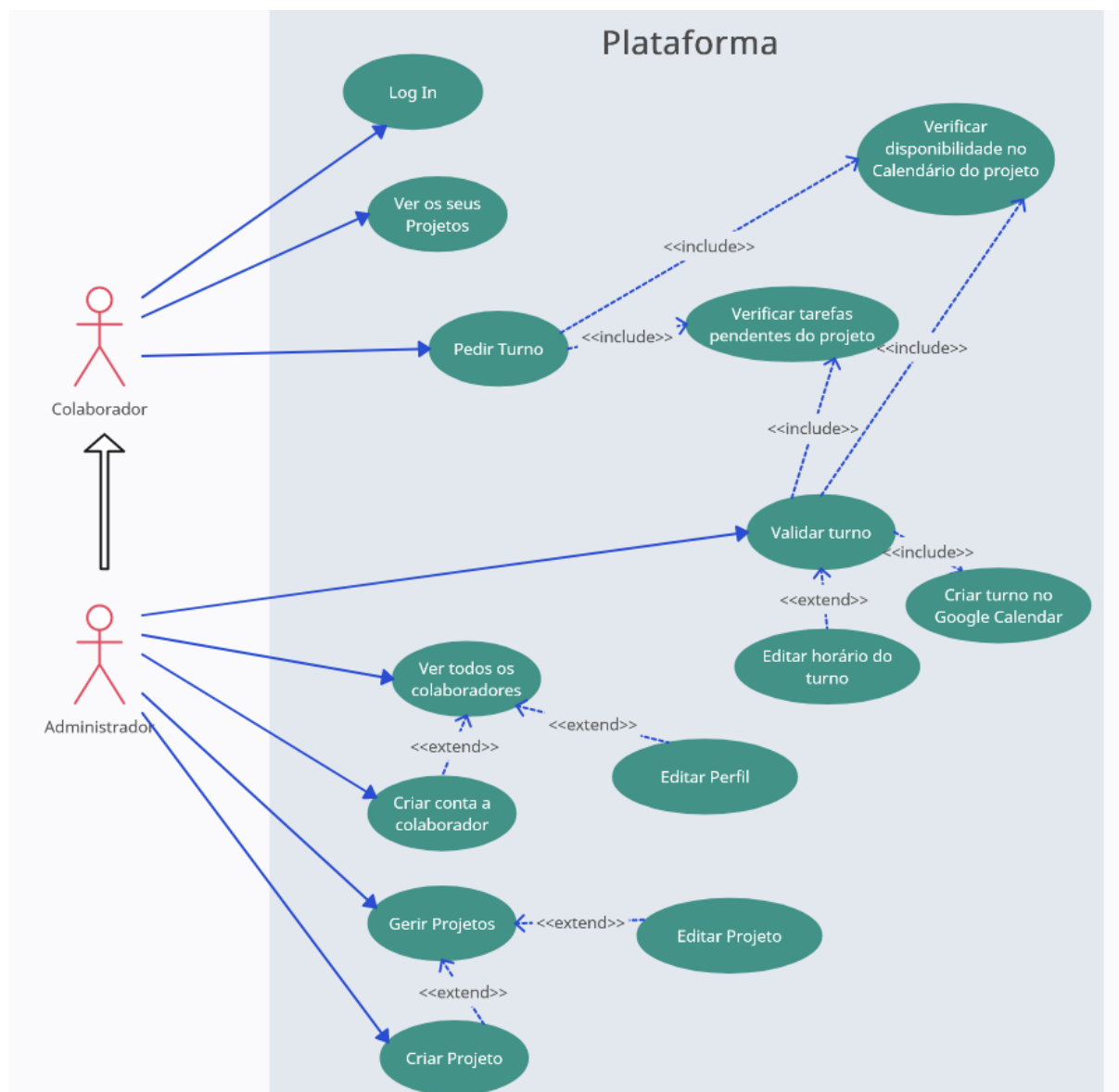


Figura 3: Diagrama Use Cases da plataforma

Ao abordar esta lacuna, a plataforma que desenvolvemos proporciona uma excelente solução para os problemas enfrentados pela a empresa, mas com a grande vantagem de ser acessível e adaptável para todas as empresas, independentemente do seu porte ou recursos disponíveis. E ao adotar tecnologias, como esta plataforma *opensource*, as empresas são capazes de melhorar os seus processos, alcançar maior eficácia e qualidade nos projetos desenvolvidos, além de promover uma gestão mais estratégica dos seus recursos humanos.

3. Revisão de Literatura

A análise e escolha das ferramentas de desenvolvimento e tecnologias utilizadas no projeto de criação da plataforma de gestão é fundamental. A partir desta revisão é possível fornecer uma visão aprofundada das decisões tomadas em relação às ferramentas e tecnologias adotadas, bem como destacar a sua importância para o sucesso do projeto.

3.1. Ferramentas de Desenvolvimento

As ferramentas desempenharam um papel crucial no processo de criação da plataforma de gestão, garantindo a eficiência e qualidade do resultado final.

3.1.1. MySQL Workbench

Antes de qualquer iteração a nível de implementação e após ser feito o levantamento de todos os requisitos e estabelecer os prioritários, utilizamos o MySQL Workbench pois este permite criar e modificar visualmente modelos o que tornou-se útil para estruturar a nossa base de dados durante a fase de design do projeto através de um diagrama ER (Entidade-Relação) onde representamos as tabelas, relacionamentos e restrições. Assim, obtivemos uma visão clara do que será a base do projeto e de que forma irá funcionar como um todo tendo em conta os requisitos elicitados.

Esta ferramenta foi muito importante pois ao longo da implementação do projeto tornou-se um auxiliar quer para a criação de todas as entidades constituintes, quer para dar orientação na criação de todas as funcionalidades.

3.1.2. Figma

Em simultâneo com a estruturação da base de dados, considerando que todos os requisitos prioritários já se encontravam estabelecidos decidimos utilizar o Figma, que é uma ferramenta de design de interface do utilizador (UI), para criar o protótipo interativo.

Esta ferramenta, inicialmente, foi subestimada, no entanto, demonstrou o quão importante foi a sua utilização, pois a partir dela permitiu que os vários stakeholders tivessem uma percepção de como a interface funcionaria antes de ser implementada. O que possibilitou a validação da mesma, obter feedback sobre o design, identificar possíveis melhorias e solucionar problemas de usabilidade antes de perder tempo e recursos na implementação do projeto final.

Também ajudou-nos a obter uma referência para que se tornasse mais fácil ter uma visão e entender melhor o fluxo de trabalho do projeto, alinhando-se assim às expectativas, reduzir erros de interpretação e promover um trabalho mais eficaz entre os stakeholders.

3.1.3. API Google Calendar

Tendo em conta que a empresa já se encontrava dependente de vários serviços da Google, sendo um dos exemplos o facto de na maioria das plataformas pedir a autenticação com a Google para acessos, achamos vantajoso manter este padrão pois futuramente seria possível sincronizar informações entre essas ferramentas, facilitando a gestão. Para além disso, ao utilizar a API cada colaborador caso tenham turnos marcados serão notificados sendo possível fazer toda a gestão dos compromissos de forma simples em qualquer lugar e a qualquer momento.

3.1.4. GitHub

Foi utilizado o GitHub como ferramenta para controle de versões do código, a plataforma possibilita um histórico completo e organizado das alterações realizadas no projeto. Será uma ferramenta fundamental para que várias empresas possam usufruir dos recursos sem a necessidade de investir em soluções pagas.

Além disso, o GitHub permite que a comunidade interessada tenha acesso ao código e possa colaborar com o projeto. Isso possibilita um ambiente de desenvolvimento colaborativo, onde entidades externas podem contribuir com melhorias e novos recursos com o intuito de tornar esta plataforma mais completa.

3.2. Tecnologias Utilizadas

Durante o processo de escolha das tecnologias a serem utilizadas no desenvolvimento deste projeto, foi levado em consideração o uso interno na empresa, propondo manter um padrão e preservar o método de trabalho.

3.2.1. Laravel

A nível de implementação, primeiramente, optamos por utilizar a framework Laravel por ser uma das mais completas pois oferece uma variedade de funcionalidades e recursos tornando o desenvolvimento da plataforma mais simples. Além disso, possui uma estrutura bem definida que auxilia a seguir as melhores práticas de programação e manter um código limpo e organizado, o que para o meu caso sendo um iniciante era excelente. [3]

Porém, mesmo após alguma experiência com o Laravel, inicialmente, houve várias dificuldades em desenvolver a plataforma. Neste sentido, em grupo foi proposto a utilização de um bundle que é um conjunto de funcionalidades “pré-empacotadas” que podem ser adicionadas a um projeto para aumentar as suas capacidades. Neste caso, usamos o bundle simplesmente para devolver uma template de dashboard com a finalidade de auxiliar na questão do front-end e ir de encontro ao âmbito do projeto.

Todavia, tornou-se ainda mais difícil a adaptação pois metade da documentação estava em Inglês e a outra metade encontrava-se em Chines.

Após tudo isto, foi acordado em grupo e optamos por migrar para a framework Symfony. Esta é uma alternativa robusta que atua em um ambiente semelhante ao Laravel, porém é reconhecida por sua eficiência e escalabilidade em projetos complexos e de grande escala.

3.2.2. Symfony

O Symfony adota uma abordagem mais flexível, permitindo um alto nível de adaptação às necessidades específicas do projeto. Ele fornece um conjunto abrangente de componentes reutilizáveis e uma arquitetura bem estruturada, facilitando o desenvolvimento de plataformas web de alta complexidade. [4]

A escolha pelo Symfony foi motivada pela perspectiva de um projeto em constante evolução após o estágio curricular, onde a eficiência e a escalabilidade são fatores essenciais. A framework Symfony oferece recursos avançados, como gestão de cache, segurança, entre outros. Esses recursos permitem um desenvolvimento mais eficiente e facilitam a implementação de funcionalidades complexas no projeto. [5]

No contexto do backend, optamos por utilizar a linguagem PHP, que é a linguagem principal compatível com o Symfony. O PHP será responsável por processar todos os pedidos dos utilizadores, será o ideal para aceder à base de dados, realizar todas as autenticações, no fundo, será responsável por processar as informações que o utilizador pretende ao realizar interações com a plataforma.

Por fim utilizamos o JavaScript, HTML e CSS como linguagens frontend, trabalharão em conjunto para construir todas as páginas web relativas a este projeto e criar uma experiência de utilizador interativa.

Em particular utilizaremos o HTML para ,a partir dos seus elementos, estruturar e definir o conteúdo da página web.

O CSS é uma linguagem de estilo utilizada para definir a aparência visual das páginas web.

O JavaScript terá um papel fundamental na construção da plataforma. Enquanto o HTML e o CSS cuidam da estrutura e da aparência visual, o JavaScript adiciona comportamento dinâmico às páginas. É possível criar funcionalidades interativas, como animações, validações de formulários em tempo real e neste caso, integrações com API.

Utilizamos o JavaScript maioritariamente para integrar a API Google Calendar na plataforma para realizar a criação de turnos, criação de calendários e a partilha dos mesmos.

3.2.3. Docker

O Docker é uma plataforma de virtualização de containers que permite empacotar, distribuir e executar aplicações em diferentes ambientes. Os containers são unidades isoladas que contêm todos os componentes necessários para a execução de uma aplicação, incluindo código, bibliotecas e dependências. [6]

Essa abordagem permite que as aplicações sejam executadas de forma eficiente e segura, independentemente do ambiente em que estão sendo inseridas.

Foi utilizado o Docker na integração com a plataforma, com o objetivo de aproveitar as vantagens que esta tecnologia oferece. A principal vantagem que levou à utilização do Docker é a sua portabilidade, os containers são independentes da plataforma, o que significa que podem ser executados em qualquer ambiente compatível com Docker eliminando assim problemas de compatibilidade. [7]

Outra vantagem do Docker é a facilidade de escalabilidade. Com o Docker, é possível escalar, adicionando ou removendo containers de forma rápida. Isso permite responder de maneira eficiente às grandes quantidades de tráfego e garantir um bom desempenho da aplicação. [8]

No entanto, ao longo da implementação é importante referir uma das desvantagens encontradas que, neste caso, é a complexidade inicial de configuração pois requer um certo nível de conhecimento para aproveitar todos os seus recursos e até mesmo conseguir a integração com a plataforma a 100% e sem problemas de desempenho.

No que diz respeito ao processo de deploy, integramos o Docker dentro do GitHub Actions como parte do workflow para tornar o processo de deploy o mais rápido possível. O GitHub Actions é um recurso do GitHub que permite a criação de workflows automáticos. Ao utilizar o Docker como uma ação dentro do workflow, podemos construir, testar e inserir na plataforma de forma automática e consistente.

Essa abordagem segue a ideologia de Continuous Deployment (CD), onde as alterações de código são automaticamente inseridas na produção. [9]

3.3. Tecnologias Alternativas

É importante realçar que existem outras alternativas excelentes no mercado que poderiam ser consideradas viáveis.

3.3.1. Next.js

Relativamente a tecnologias alternativas temos o Next.js que está cada vez mais popular para o desenvolvimento de aplicações web. [10] Tornando-se uma das boas alternativas mais benéfica por parte do cliente pois este é capaz de lidar com um grande volume de tráfego sem comprometer a experiência do utilizador devido à sua arquitetura de servidor híbrido (*Hybrid Server Side Rendering*), esta permite que as páginas web sejam pré-renderizadas, ou seja, o utilizador recebe uma resposta muito mais rápida do servidor. [11]

Consequentemente, seria utilizado React tendo em conta que esta framework é baseada na mesma. É uma das bibliotecas de JavaScript mais utilizadas hoje em dia, principalmente no frontend que torna mais fácil a criação de interfaces de utilizador e seria outra das alternativas no contexto do projeto, pois permite o desenvolvimento de interfaces reutilizáveis. Isso é útil para criar componentes de interface consistentes em toda a plataforma, facilitando a manutenção e a escalabilidade. [12]

3.3.2. Blazor

Outra alternativa a ser considerada é a Blazor Framework. Esta é uma tecnologia da Microsoft que permite o desenvolvimento de plataformas web utilizando C# e .NET. Com o Blazor, é possível escrever código em C# tanto no lado do cliente quanto no lado do servidor, tornando o desenvolvimento mais eficiente e eliminando a necessidade de utilizar JavaScript para tarefas de frontend. [13]

Ao ser uma framework de linguagem única, facilitaria no caso se a plataforma fosse desenvolvida em equipa pois podem trabalhar com uma linguagem familiar em todas as partes da plataforma facilitando a reutilização de código e a sua manutenção. [14]

Para além disso, teria acesso a todo o ecossistema da .NET como por exemplo, bibliotecas e recursos que proporcionam vastas opções de desenvolvimento.

4. Implementação

Numa primeira abordagem do problema demos prioridade ao levantamento de requisitos a partir de um focus group constituído pelo diretor do departamento académico, coordenador de voluntários, engenheiro informático da empresa e um voluntário.

Foi feita a escolha desta técnica pois era importante recolher todas as informações que estes stakeholders tinham em mente para o desenvolvimento deste projeto e perceber que funcionalidades seriam as mais relevantes a ser desenvolvidas. A composição deste grupo foi bastante importante pois estes representavam diferentes perfis de utilizador com interesses e que atuam em diferentes áreas sendo possível assim obter uma compreensão abrangente das necessidades e expectativas dos utilizadores.

Além disso, o facto da discussão ser guiada levou à exploração de vários temas e ideias, onde cada um partilhou não só as várias experiências com a forma de resolver o problema como também os desafios diários existentes, sempre com o objetivo de obter uma compreensão mais profunda dos requisitos e das expectativas dos utilizadores tornando assim, tudo mais dinâmico no sentido de nós participantes ao trocarmos ideias estamos constantemente a receber feedbacks instantâneos do grupo levando a discussões mais aprofundadas de algo que não possa ter ficado bem esclarecido e a partir destes criar novas sugestões que muitas delas tornam-se requisitos.

Passado uma semana após ter todos os requisitos elicitados relativamente à primeira técnica, sentiu-se a necessidade de realizar mais uma técnica de elicitação com o objetivo de levantar mais requisitos e perceber de forma mais óbvia o que a empresa pretendia com a plataforma. Assim sendo, decidiu-se utilizar a técnica de entrevista ao qual começamos por preparar um modelo de entrevista para manter a discussão focada no objetivo. Para estas entrevistas individuais foram convidados o diretor do departamento académico, o coordenador dos voluntários e o engenheiro informático (orientador).

O resultado destas duas técnicas estão apresentadas no Anexo A.

Após a elicitação, juntamente com o orientador, foi feito o processo de priorização de requisitos tendo em conta o objetivo principal do projeto e tendo em conta o que os stakeholders mais evidenciaram quer seja no focus group como também nas entrevistas.

Resultando nos seguintes 15 requisitos mais prioritários:

1. A plataforma deverá permitir ao administrador criar conta para colaboradores;
- A plataforma deverá notificar o colaborador via e-mail que a conta foi criada;
2. A plataforma deverá permitir a inserção de informações pessoais dos colaboradores;
3. A plataforma deverá permitir os utilizadores editar a conta;
4. A plataforma deverá permitir a visualização de informações pessoais dos colaboradores apenas por administradores.
5. A plataforma deverá permitir ao utilizador realizar o login;

6. A plataforma deverá permitir ao administrador dar cargo aos colaboradores;
8. A plataforma deverá permitir a criação de projetos;
 - A plataforma deverá permitir adicionar o nome do projeto;
 - A plataforma deverá permitir adicionar o responsável pelo projeto;
 - A plataforma deverá permitir adicionar o prazo de entrega;
 - A plataforma deverá permitir adicionar colaboradores ao projeto;
 - A plataforma deverá permitir remover colaboradores ao projeto;
9. A plataforma deverá permitir criar tarefas;
 - A plataforma deverá permitir adicionar data limite;
 - A plataforma deverá permitir adicionar o estado da tarefa;
 - A plataforma deverá permitir a visualização das tarefas por concluir;
 - A plataforma deverá permitir a visualização das tarefas concluídas;
10. O colaborador deverá conseguir visualizar os projetos a quais está associado;
11. A plataforma deverá notificar via e-mail 3 dias antes da data limite de um projeto
13. O administrador deverá conseguir visualizar todos os projetos concluídos;
14. O administrador deverá conseguir visualizar todos os projetos por concluídos;
20. A plataforma deverá monitorizar as horas de trabalho dos colaboradores.
23. A plataforma deverá permitir o colaborador marcar um turno;
 - A plataforma deverá permitir visualizar calendário por projeto;
 - O administrador deverá receber as horas do turno pedidas pelo colaborador;
 - O administrador deverá conseguir aprovar o horário;
 - O administrador deverá conseguir recusar o horário;
 - A plataforma deverá atualizar o calendário sempre que é aprovado um novo horário;
 - A plataforma deverá notificar o utilizador via e-mail sobre o resultado da submissão do turno;
25. A plataforma deverá permitir que o colaborador visualize os seus turnos;

Partindo para a implementação, inicialmente começamos por criar todas as entidades já concordadas e verificadas a partir do diagrama de entidade-relação, tornando assim todo este processo mais simples.

4.1. Colaboradores

Iniciamos pela implementação da CRUD dos colaboradores da empresa onde só os utilizadores com a role “ROLE_ADMIN” são capazes de aceder a esta área para realizar a gestão de todos os colaboradores da empresa, aceder a várias informações como por exemplo, morada, nif, cc, e-mail, número de telemóvel, entre outros e até mesmo dar acesso da plataforma a novos colaboradores.

4.2. Login

Para que esta plataforma pudesse ser privada e apenas ser acedida por membros com credenciais fornecidas pelos administradores foi necessário implementar um sistema de login do absoluto zero (inserção do email e password).

De modo a que um colaborador tenha acesso é necessário que um administrador crie a sua conta com toda a sua informação providenciada tal como as suas credenciais que serão partilhadas para o colaborador via e-mail, de realçar que a password é encriptada logo após a submissão do formulário de criação de conta de maneira a que se mantenha a segurança em todos os processos.

Este sistema de Login é constituído pelo SecurityController que trata de renderizar a página de login.

Pela classe LoginFormAuthenticator que é responsável por toda a autenticação do utilizador a partir das credenciais colocadas no formulário que cria um objeto Passport e verifica se existe ou não este utilizador .

4.3. Projetos

Esta categoria é constituída por duas subcategorias:

4.3.1. Geral

É responsável por toda a gestão de projetos da empresa ao qual, logicamente, só os administradores têm acesso.

Foi implementado um CRUD onde à primeira vista apresenta uma lista com os projetos da empresa ordenados com prioridade os que se encontram em desenvolvimento.

Como referido anteriormente é possível criar e editar cada um deles inserindo dados como o nome, descrição, o responsável, a data limite de entrega, vários colaboradores que farão parte da equipa e as tarefas a serem realizadas até a conclusão, eventualmente todos eles poderão ser editados.

No entanto, para poder estabelecer essas ligações com diferentes entidades foi necessário criar formulários “individuais” para cada uma das entidades com o respectivo(s) input(s) mais convenientes e chamá-los na CRUD dos projetos e assim conseguir fazer ligação com as entidades de interesse.

O calendário apresentado na Figura 5 está associado com a API Google Calendar e é atualizado em tempo real cada vez que um turno é validado pelo administrador na secção “Calendário” de forma a que cada colaborador antes de pedir o seu turno esteja ciente de que horas estão disponíveis evitando assim o grande problema que a empresa enfrenta que é a sobreposição dos turnos e a falta de gestão das tarefas por cada projeto.

4.3.2. Meus Projetos

Esta subcategoria é relativamente parecida com a anterior, com dois únicos pormenores:

- Só é apresentado para cada colaborador os projetos que o mesmo faz parte
- Só é permitido ver os detalhes do projeto

4.4. Calendário

É nesta categoria que é resolvida toda a parte de recursos humanos relativamente à marcação de turnos de todos os colaboradores. Os turnos advêm, neste contexto, sempre de uma tarefa por realizar em um determinado projeto.

4.4.1. Meus Turnos

Todos os colaboradores têm acesso. É nesta subcategoria que cada colaborador irá pedir o seu turno dependendo de que projeto esteja associado, apresentando que tarefa irá desempenhar e que horário o irá fazer. Posteriormente é enviado para os administradores com o estado “Revisão”.

4.4.2. Validação

Aqui é feita toda a gestão de turnos da empresa, após a submissão de turno de um colaborador será o administrador a verificar se haverá disponibilidade para exercer o mesmo dependendo da disponibilidade que será apresentada no calendário associado ao respetivo projeto.

Assim facilitamos o trabalho do administrador, no sentido de não ter que fazer grande esforço para ir de encontro ao calendário do projeto e fazer a verificação.

Após toda a verificação, o administrador simplesmente clica no botão validar que é redirecionado para um formulário que sofreu manipulações com o objetivo de ser aqui implementada a ligação com a API Google Calendar.

É de realçar que existem vários campos “hidden” na página de validação com o intuito dos “values” serem recebidos nos ficheiros javascript nomeadamente o “init.js” e o “app.js”.

O ficheiro “init.js” é responsável pela ligação e as autorizações específicas da plataforma com a API.

Já o ficheiro “app.js” recebe todos os dados providenciados pelo colaborador no pedido de turno e a partir destes é capaz fazer a verificação se o calendário já foi criado, caso sim ele apenas adiciona esse turno ao calendário. Caso contrário criará no Google Calendar o calendário com o nome do projeto (Figura 6) partilhado com todos os administradores da plataforma.

De seguida, em simultâneo, cria o turno e envia um convite para o email do colaborador a avisar que tem que o aceitar.

```
// Check if the calendar already exists
const calendarListResponse = await gapi.client.calendar.calendarList.list({});
console.log(calendarListResponse)

const calendars = calendarListResponse.result.items;
console.log(calendars)
let calendarId;
for (let i = 0; i < calendars.length; i++) {
  if (calendars[i].summary === calendarName) {
    calendarId = calendars[i].id;
    break;
  }
}

// If the calendar doesn't exist, create it
if (!calendarId) {
  const calendar = {
    summary: calendarName,
    timeZone: 'Europe/Lisbon'
  };
  const calendarResponse = await gapi.client.calendar.calendars.insert({
    resource: calendar
  });
  calendarId = calendarResponse.result.id;
}
```

Figura 6 : Verificação da existência do calendário

Após todo este processo é apresentado uma mensagem a indicar que o turno foi criado no Google Calendar, com sucesso, para que o administrador fique ciente que foi feito todas as etapas entre a plataforma e a API.

De realçar que é enviado como campo “hidden” para a base de dados o token do utilizador para que o sistema não apresentasse a cada validação de turno o pedido de acesso e autorização à conta Google do administrador em questão, ou seja com este pormenor o administrador aceita o acesso apenas a primeira vez. Além disso, também é enviado o id do calendário para que fosse possível integrar o calendário da Google com os turnos associados e atualizados em tempo real dentro da nossa plataforma

5. Avaliação

Esta avaliação consiste em verificar se a implementação atingiu os requisitos propostos e se todas as funcionalidades foram implementadas de forma correta e eficiente.

Com base nesta avaliação, será possível identificar pontos fortes e áreas que requerem melhorias ou ajustes adicionais. As conclusões obtidas serão utilizadas como base para futuras iterações do projeto, visando aprimorar a plataforma e garantir um funcionamento eficiente e adequado na gestão de recursos humanos e projetos.

Dessa forma, a avaliação dos requisitos proporcionará uma visão abrangente sobre a realização das funcionalidades propostas, contribuindo para uma análise aprofundada do projeto.

Requisito: A plataforma deverá permitir ao administrador criar conta para colaboradores.

Implementação: 100% concluída

Requisito: A plataforma deverá permitir a inserção de informações pessoais dos colaboradores.

Implementação: 100% concluída

Requisito: A plataforma deverá permitir aos utilizadores editar a conta.

Implementação: 40% concluída

Requisito: A plataforma deverá permitir a visualização de informações pessoais dos colaboradores apenas por administradores.

Implementação: 100% concluída

Requisito: A plataforma deverá permitir ao utilizador realizar o login.

Implementação: 100% concluída

Requisito: A plataforma deverá permitir ao administrador dar cargo aos colaboradores.

Implementação: 100% concluída

Requisito: A plataforma deverá permitir a criação de projetos.

Implementação: 100% concluída

Requisito: A plataforma deverá permitir criar tarefas.

Implementação: 100% concluída

Requisito: O colaborador deverá conseguir visualizar os projetos aos quais está associado.

Implementação: 100% concluída

Requisito: A plataforma deverá notificar via e-mail 3 dias antes da data limite de um projeto.

Implementação: 0% concluída

Requisito: O administrador deverá conseguir visualizar os projetos por estados.

Implementação: 100% concluída

Requisito: A plataforma deverá permitir ao colaborador marcar um turno.

Implementação: 100% concluída

Requisito: A plataforma deverá notificar o utilizador via e-mail sobre o resultado da submissão do turno.

Implementação: 100% concluída

Requisito: O administrador deverá conseguir validar turnos.

Implementação: 100% concluída

Requisito: A plataforma deverá atualizar o calendário sempre que for aprovado um novo turno.

Implementação: 100% concluída

Requisito: A plataforma deverá permitir que o colaborador visualize os seus turnos.

Implementação: 100% concluída

6. Discussão

Durante a avaliação da implementação do projeto, verificamos que a maioria dos requisitos foram concluídos com sucesso. Foram implementados de forma correta e eficiente, garantindo o bom funcionamento da plataforma e fornecendo aos utilizadores a capacidade de gerir projetos e tarefas de recursos humanos de forma eficaz.

No entanto, também foram identificados alguns pontos fracos e requisitos que não foram totalmente implementados. A notificação via e-mail, que deveria ser enviada aos utilizadores três dias antes da data limite de um projeto e a edição do perfil por parte do utilizador que neste caso ficou pelos 40%, ou seja, só foi desenvolvida a edição dos perfis caso o utilizador seja um administrador.

Um dos pontos fracos do projeto foi a ausência de testes unitários, que idealmente seriam implementados. Esta falta de testes pode ter um impacto negativo na manutenção futura do sistema, dificultando a identificação e correção de erros.

Outro ponto que merece destaque é a falta de testes com utilizadores. Idealmente, a plataforma teria sido submetida a um período de teste de duas semanas. Esses testes permitiriam obter feedback valioso sobre a usabilidade da plataforma, identificar possíveis falhas e realizar ajustes necessários.

Estas duas limitações são resultado das prioridades estabelecidas durante o desenvolvimento do projeto, bem como de restrições de recursos e tempo. Embora a falta de testes e validações com utilizadores possa ser considerada um ponto negativo no processo de desenvolvimento, é importante ressaltar que concentramos-nos em priorizar as funcionalidades essenciais e garantir que a plataforma atendesse aos requisitos principais.

Inicialmente houve grandes problemas de adaptação ao qual foi investido muito tempo a perceber como tudo funcionava dentro da framework. No entanto, à medida que progredimos e nos envolvemos mais com o projeto começamos, gradativamente, a ficar mais competentes no desenvolvimento dos requisitos pendentes.

Passei por várias dificuldades a nível de implementação, integração e configuração da API da Google Calendar ao qual abordei esta tarefa sem qualquer experiência de trabalho com API's. Inicialmente tentei integrar a API com PHP, no entanto, sem sucesso pois havia sido retirado da própria documentação da Google, posteriormente surgiu a opção de utilizar JavaScript ao qual tive necessariamente de manipular a própria view para essa integração e para além disso, todo o processo de configuração da API dentro da plataforma foi onde investi pelo facto de aparecerem a maioria dos erros relativamente às várias autorizações necessárias por parte da API com a conta Google. Já, todas as funcionalidades que envolveram o Google Calendar desde a criação de turnos até a adição aos respetivos calendários foi um processo mais agradável tendo em conta que o seu desenvolvimento era mais intuitivo.

Após a conclusão do projeto, foi realizada uma discussão abrangente para avaliar se o problema proposto foi adequadamente resolvido e se todas as metas foram alcançadas. A discussão teve como objetivo compreender os resultados obtidos e identificar possíveis áreas que requerem melhorias ou ajustes adicionais.

Durante a discussão, foram analisados os resultados alcançados em relação aos objetivos iniciais do projeto. Foram considerados aspectos como funcionalidades,

desempenho e usabilidade. Verificou-se que a solução desenvolvida atendeu às expectativas da empresa e que as funcionalidades essenciais foram implementadas de forma correta e eficiente.

Além disso, foram avaliadas áreas que ainda necessitam de ajustes, nomeadamente, pontos que podem ser melhorados na parte de apresentação ao utilizador, tal como aspectos de acessibilidade na página de turnos, no sentido de tornar visível o botão que é responsável por direccionar para a página de validação. Todos estes aspectos serão tratados em futuras iterações do projeto.

7. Conclusão

Dado o projeto como finalizado para esta etapa curricular, concluiu-se que o desenvolvimento do mesmo correu relativamente bem tendo em conta que consegui implementar todos os requisitos prioritários para que a plataforma conseguisse à primeira vista resolver os problemas apresentados.

Todo o trabalho realizado antes da implementação teve um papel importante para que o projeto final corresse da melhor forma. A partir da elicitação de requisitos através de focus group e entrevistas, foi um processo demorado mas crucial pois deu-me a oportunidade de compreender de forma clara os objetivos do projeto, que necessidades teriam que ser tratadas, quais seriam os desafios e a partir deles conseguir um planeamento realista, juntamente com os meus orientadores, do que seria melhor implementar nas diferentes etapas do desenvolvimento do projeto o que acabou por me dar um rumo claro do que deveria ser feito e em que prazos para que consequentemente atingisse o sucesso do projeto.

Outro aspeto a considerar foi a estruturação da base de dados antes da implementação pois foi crucial para garantir a organização, integridade e a eficiência dos dados do sistema que forneceu uma base sólida para o desenvolvimento da plataforma sendo assim um aspeto importante para este projeto.

Ao longo da implementação, preservei o trabalho autónomo com o intuito de aproveitar e aprender ao máximo neste projeto/estágio mantendo o foco nas metas estabelecidas, garantindo que todas fossem alcançadas.

Diante do exposto, ao desenvolver esta plataforma de gestão de recursos humanos e de projetos open-source, as empresas têm a oportunidade de melhorar suas operações e aperfeiçoar soluções flexíveis. Isso não só permite que atendam às necessidades específicas da sua empresa, mas também promove a eficiência e o sucesso a longo prazo, mesmo com recursos limitados. A adoção de tecnologias como o Docker fortalece ainda mais essa abordagem, proporcionando vantagens adicionais de portabilidade, consistência e eficiência no desenvolvimento.

Para além do solucionado neste projeto, existe em mente várias outras features a serem implementadas em breve como a monitorização de horas realizadas por cada colaborador que para isto teremos que implementar um sistema que verifique se realmente o colaborador compareceu ao turno que foi validado. Ou seja, após pedir o turno estas horas manter-se-ão em pré-registo até que o responsável confirme a sua presença e assim essas horas sejam registadas. Será uma funcionalidade extremamente útil pois a empresa a partir das horas de trabalho exercido remunera com ajudas de custo.

Posteriormente, temos como ideia implementar a possibilidade de um responsável de projeto poder criar diretamente um turno para um determinado colaborador a indicar que tarefa irá realizar e para além disso pretendemos que o colaborador após o seu turno ser validado existir a opção do mesmo o cancelar indicando o motivo que será enviado uma notificação para o determinado responsável de projeto a avisar.

Após estes passos migraremos toda esta plataforma para o painel de gestão da empresa.

Por fim, agradeço ao Diogo Freitas e aos meus dois orientadores, o João Vasconcelos e o senhor professor Filipe Quintal por estarem sempre disponíveis para me auxiliar em

qualquer questão e por estarem disponíveis nas reuniões semanais onde discutimos ideias, decidimos funcionalidades e, principalmente, decidimos o melhor rumo para que este projeto fosse desenvolvido com sucesso.

8. Referências

- [1] L. Sands, «Why is human resources (HR) important?»
<https://www.breathehr.com/en-gb/blog/topic/business-process/why-is-human-resources-important>
- [2] Asana, «O que é gestão de projetos e quais são os seus benefícios? [2022] • Asana»,
Asana. <https://asana.com/pt/resources/benefits-project-management>
- [3] «The Laravel PHP Framework – Web App Construction for Everyone», *Kinsta®*, 16 de junho de 2023. <https://kinsta.com/knowledgebase/what-is-laravel/>
- [4] «Symfony: What are the advantages of using this framework?», *Bocasay*, 19 de julho de 2022. <https://www.bocasay.com/benefits-symfony-web-development/>
- [5] Symfony, «Chapter 1 - Introducing Symfony (symfony 1.4 legacy version)».
https://symfony.com/legacy/doc/gentle-introduction/1_4/en/01-Introducing-Symfony
- [6] «Docker | O que é Docker e como ele funciona?»
<https://www.redhat.com/pt-br/topics/containers/what-is-docker>
- [7] D. C, «O Que é Docker e Como Ele Funciona? – Docker Explicado», *Hostinger Tutoriais*, 25 de outubro de 2022. <https://www.hostinger.com.br/tutoriais/o-que-e-docker>
- [8] «Why Use Docker? Advantages and Tools».
<https://www.knowledgehut.com/blog/devops/why-use-docker>
- [9] «Entendendo o GitHub Actions», *GitHub Docs*.
<https://ghdocs-prod.azurewebsites.net/pt/actions/learn-github-actions/understanding-github-actions>
- [10] «Advantages and disadvantages of Next.js», *DEV Community*, 8 de setembro de 2022.
<https://dev.to/richkurtzman/advantages-and-disadvantages-of-nextjs-5hg6>
- [11] «Next.js; what is it and why do we use it?»
<https://www.clock.co.uk/post/next-js-what-is-it-and-why-do-we-use-it>
- [12] «Why Use React?», *TatvaSoft Blog*, 20 de julho de 2022.
<https://www.tatvasoft.com/blog/why-use-react/>
- [13] «What is Blazor Framework (How it Works for Developers Tutorial) | IronPDF».
<https://ironpdf.com/blog/net-help/blazor-framework-tutorial/>
- [14] «The Advantages of Blazor over JavaScript Frameworks», *Telerik Blogs*, 24 de março de 2020.
<https://www.telerik.com/blogs/why-you-should-use-blazor-over-javascript-frameworks-to-build-your-single-page-application>

9. Anexo A

Requisitos de Tecnologia

- A plataforma deverá ser desenvolvida para a web.
- A plataforma deverá ser desenvolvida na framework Symfony.
- A aplicação deverá ser desenvolvida com PHP.
- A aplicação deverá ser desenvolvida com Javascript.
- A plataforma deverá ser desenvolvida com HTML.
- A plataforma deverá ser desenvolvida com CSS.
- A plataforma deverá ser integrada com uma base de dados SQL.
- A plataforma deverá ser integrada num ambiente Docker.

Requisitos Funcionais

1. A plataforma deverá permitir ao administrador criar conta para colaboradores;
 - A plataforma deverá notificar o colaborador via e-mail que a conta foi criada;
2. A plataforma deverá permitir a inserção de informações pessoais dos colaboradores;
3. A plataforma deverá permitir os utilizadores editar a conta;
4. A plataforma deverá permitir a visualização de informações pessoais dos colaboradores apenas por administradores.
5. A plataforma deverá permitir ao utilizador realizar o login;
6. A plataforma deverá permitir ao administrador dar cargo aos colaboradores;
7. A plataforma deverá permitir utilizador criar tarefas individuais;
8. A plataforma deverá permitir a criação de projetos;
 - A plataforma deverá permitir adicionar o nome do projeto;
 - A plataforma deverá permitir adicionar o responsável pelo projeto;
 - A plataforma deverá permitir adicionar o prazo de entrega;
 - A plataforma deverá permitir adicionar colaboradores ao projeto;
 - A plataforma deverá permitir remover colaboradores ao projeto;
9. A plataforma deverá permitir o responsável do projeto criar tarefas;
 - O responsável deverá atribuir a tarefa a um colaborador do projeto;
 - A plataforma deverá permitir adicionar a dificuldade;
 - A plataforma deverá permitir adicionar data limite;
 - A plataforma deverá permitir adicionar o estado da tarefa;
 - A plataforma deverá ser capaz de ordenar as tarefas por data limite;
 - A plataforma deverá permitir a visualização das tarefas por concluir;
 - A plataforma deverá permitir a visualização das tarefas concluídas;

10. O colaborador deverá conseguir visualizar os projetos a quais está associado;
11. A plataforma deverá notificar via e-mail 3 dias antes da data limite de um projeto.
12. A plataforma deverá notificar via e-mail 3 dias antes da data limite de uma tarefa.
13. O administrador deverá conseguir visualizar todos os projetos concluídos;
14. O administrador deverá conseguir visualizar todos os projetos por concluídos;
15. A plataforma deverá permitir adicionar comentários aos projetos;
16. A plataforma deverá permitir aceder a um calendário integrado para visualização do prazo de tarefas;
17. O utilizador deverá conseguir criar lembretes que será notificado via e-mail.
18. A plataforma deve permitir o registo de férias dos colaboradores.
19. A plataforma deverá permitir criar reclamações.
20. A plataforma deverá monitorizar as horas de trabalho dos colaboradores.
21. A plataforma deverá calcular automaticamente os salários dos colaboradores tendo em conta as horas de trabalho.
22. A plataforma deve permitir a realização de inquéritos de satisfação dos colaboradores.
23. A plataforma deverá permitir o colaborador marcar um turno;
 - A plataforma deverá permitir visualizar calendário por projeto;
 - A plataforma deverá permitir visualizar calendário por atividade;
 - A plataforma deverá permitir visualizar calendário por programa;
 - O administrador deverá receber as horas do turno pedidas pelo colaborador;
 - O administrador deverá conseguir aprovar o horário;
 - O administrador deverá conseguir recusar o horário;
 - A plataforma deverá atualizar o calendário sempre que é aprovado um novo turno;
 - A plataforma deverá notificar o utilizador via e-mail sobre o resultado da submissão do turno;
24. A plataforma deverá permitir o administrador visualizar os horários de cada colaborador;
25. A plataforma deverá permitir que o colaborador visualize os seus turno;
26. A plataforma deverá permitir o administrador visualizar calendário por projeto;
27. A plataforma deverá permitir o administrador visualizar calendário por atividade;

28. A plataforma deverá permitir o administrador visualizar calendário por programa;
29. A plataforma deverá notificar o colaborador via e-mail, após cada turno;
- A plataforma deverá apresentar uma secção “Marcar presença”;
 - O colaborador deverá confirmar se esteve presente no turno;
 - O colaborador deverá confirmar se não esteve presente no turno;
 - O utilizador deverá, opcionalmente, explicar a razão de não ter comparecido no turno;
 - O utilizador deverá conseguir editar as horas do turno que realmente fez;