

Rede Vida

Pedro Gomes | Renato Azevedo | Ricardo Rocha

Desenvolvimento Colaborativo de Software – 1º Semestre | CTESP - DWM

Vila Nova de Famalicão, 23 de janeiro de 2025

ÍNDICE DE CONTEÚDOS

INTRODUÇÃO AO TEMA	3
REQUISITOS FUNCIONAIS	4
REQUISITOS NÃO FUNCIONAIS	5
SCRUM	7
EPICS, THEMES E USER STORIES	8
STORYPOINTS.....	25
DEFINIÇÃO DE CADA <i>SPRINT</i>	28
INTEGRAÇÃO DO JIRA	31
PROCESSO DE UTILIZAÇÃO DO GIT E GITHUB NO PROJETO.....	38
CONSIDERAÇÕES FINAIS	45

INTRODUÇÃO AO TEMA

O projeto que se apresenta foi desenvolvido no âmbito da unidade curricular de **Desenvolvimento Colaborativo de Software** no CTeSP em Desenvolvimento *Web* e Multimédia.

O objetivo deste sistema desenvolvido foi criar um sistema digital integrado para a gestão de dadores de sangue no Hospital de Vila Nova de Famalicão, exclusivamente para uso dos profissionais de saúde. Denominado de “**Rede Vida**” solução foca-se na centralização das informações, automatização dos processos e conformidade com as normas legais, assegurando, simultaneamente, segurança, eficiência e escalabilidade. Este sistema de utilização interna, terá acesso à base de dados nacional de dadores de sangue, registando todas as ações realizadas.

Para a sua execução, aplicámos a **framework Scrum**, que nos ajudou a organizar as tarefas de forma iterativa e colaborativa, garantindo um desenvolvimento contínuo, adaptável e focado nas necessidades reais dos utilizadores, neste caso, os profissionais de saúde. Utilizámos o **Jira** para gerir o *backlog*, planejar as sprints e monitorizar o progresso das tarefas, enquanto o **GitHub** foi essencial para o controlo de versões e para facilitar a colaboração no código-fonte.

O planeamento foi estruturado em várias áreas de responsabilidade, como a gestão de dadores, agendamentos, relatórios, segurança e conformidade com o RGPD. Cada membro da equipa, com as suas competências específicas, contribuiu para criar uma solução robusta e alinhada com os objetivos propostos. Através da aplicação do *Scrum*, conseguimos manter o foco na entrega contínua de valor, adaptando-nos aos desafios do projeto e criando um ambiente de trabalho ágil e colaborativo.

Agora, estamos prontos para abrir as portas do banco de recolha de sangue e apresentar todas as funcionalidades do nosso sistema, mostrando como ele pode melhorar a gestão do dia-a-dia dos profissionais de saúde. Damos-lhe as boas-vindas, com a certeza de que **cada um de nós pode fazer a diferença na vida do outro** — seja pela nossa essência de trabalhar em equipa, seja pela doação que, de facto, pode salvar vidas!

REQUISITOS FUNCIONAIS

Gestão de Dadores:

- O sistema deve permitir o registo de novos dadores, incluindo nome, data de nascimento, contacto, morada, NNU, tipo sanguíneo e histórico médico relevante.
- O sistema deve permitir consultar, editar ou eliminar os dados dos dadores e armazenar o histórico de doações, com informação sobre datas, locais e intervalos mínimos para novas doações.
- O sistema deve permitir a pesquisa por nome, tipo sanguíneo ou até histórico de doações. Também pode incluir filtros para exibir apenas os dadores ativos ou inativos.

Gestão de Agendamentos:

- O sistema deve incluir uma agenda centralizada que permita o profissional de saúde ou administrador realizar reservas para horários específicos e alterar ou cancelar reservas feitas anteriormente.
- O sistema deve, automaticamente, notificar os agendamentos, aos seus respetivos dadores, através de e-mail ou SMS.

Relatórios e Estatísticas:

- O sistema deve apresentar, de forma visível, o total de dadores registados no sistema.
- Deve ser possível distinguir entre dadores ativos (com doações recentes ou agendadas) e inativos (sem atividades recentes).
- O sistema deve exibir um resumo com o número de dadores de cada tipo sanguíneo (ex.: A+, O-, etc.), apresentado em formato de lista ou gráfico simples (ex.: barras ou gráfico circular).

REQUISITOS NÃO FUNCIONAIS

Desempenho

- O sistema deve ser capaz de suportar até 100 utilizadores simultâneos sem degradação significativa do desempenho.
- O tempo de resposta para qualquer operação (ex.: pesquisa, registo, edição) não deve exceder 2 segundos em condições normais de carga.

Escalabilidade

- O sistema deve ser escalável horizontalmente para acomodar o crescimento no número de dadores e no volume de dados, suportando até 1 milhão de registos de dadores.
- Deve permitir a integração futura com outros sistemas (ex.: bases de dados nacionais de dadores).

Segurança

- Os dados pessoais e médicos dos dadores devem ser armazenados e transmitidos de forma segura, respeitando o Regulamento Geral de Proteção de Dados (RGPD).
- Validação de dados obrigatórios.
- Todo o acesso ao sistema deve ser autenticado com login e senha, e as permissões de acesso devem ser configuráveis por perfil de utilizador (ex.: administrador, operador).
- As comunicações entre cliente e servidor devem ser encriptadas com protocolo HTTPS.

Usabilidade

- A interface do utilizador deve ser intuitiva, com navegação simples, seguindo os princípios de Don Norman.
- O sistema deve ser responsivo, adaptando-se automaticamente a diferentes dispositivos, incluindo os de baixa resolução.

Confiabilidade

- O sistema deve ter uma disponibilidade mínima de 99,9% ao longo do ano (máximo de 8 horas de indisponibilidade por ano).
- Deve ser implementado um mecanismo de backup automático diário dos dados, com retenção por um período mínimo de 6 meses.

Compatibilidade

- O sistema deve ser compatível com os navegadores modernos (Chrome, Firefox, Edge e Safari) nas suas versões estáveis mais recentes.

Manutenção

- O sistema deve ser modular e documentado, permitindo que novos programadores consigam compreender a sua estrutura em até 2 semanas.
- Atualizações e correções devem poder ser realizadas com interrupção mínima do serviço (idealmente fora do horário de pico).

Auditoria

- Todas as operações críticas (ex.: registo, edição e eliminação de dados ou agendamentos) devem ser registadas em logs de auditoria com as seguintes informações: Utilizador responsável, data e hora e detalhes da operação realizada.
- Os *logs* devem ser acessíveis apenas por administradores e mantidos por um período mínimo de 12 meses.

SCRUM

1. Distribuir Papéis no Scrum

Numa equipa reduzida, é comum acumular funções para assegurar o progresso do projeto. Um dos membros pode assumir o papel de *Product Owner*, ficando responsável por priorizar as tarefas e ser o principal ponto de contacto com os *stakeholders* para alinhar os requisitos. Outro pode atuar como *Scrum Master*, garantindo a aplicação da metodologia *Scrum*, facilitando as práticas e eliminando impedimentos. Paralelamente, todos os elementos da equipa integram o desenvolvimento, dividindo as tarefas conforme as suas competências específicas.

Posto isto, temos uma equipa composta por:

- **Product Owner (PO): Jeremias de Sousa**

Responsável por definir prioridades, gerir o backlog e assegurar a comunicação com os *stakeholders*. O Jeremias também detalha as *user stories* e toma decisões finais sobre o escopo e funcionalidades do projeto.

- **Scrum Master: Gustavo Lima**

Garante a aplicação do *Scrum*, facilita a colaboração na equipa e remove impedimentos que possam afetar o progresso.

- **Equipa de Desenvolvimento:**

Manuel de Vasconcelos: Desenvolvimento das funcionalidades relacionadas à gestão de dados (registo, edição e eliminação).

Susana Ribeiro: Desenvolvimento da interface de utilizador (UI) e filtros de pesquisa.

Gustavo Lima: Implementação das funcionalidades de segurança (encriptação, autenticação e autorização).

Pedro Rocha: Desenvolvimento das funcionalidades de agendamento e notificações automáticas.

Juliana Couto: Criação de relatórios e análise de dados.

Maria Leal: Garantia de conformidade com o RGPD e implementação de medidas de proteção de dados.

Cátia Dias: Implementação do sistema de auditoria.

João Guimarães: Otimização da escalabilidade e desempenho do sistema.

Em suma, temos:

1. **Gestão de Dados:** Manuel de Vasconcelos.
2. **Interface de Utilizador:** Susana Ribeiro.
3. **Segurança e Autenticação:** Gustavo Lima.
4. **Gestão de Agendamentos e Notificações:** Pedro Rocha.
5. **Relatórios e Estatísticas:** Juliana Couto.
6. **Conformidade com o RGPD:** Maria Leal.
7. **Auditoria:** Cátia Dias.
8. **Escalabilidade e Desempenho:** João Guimarães.

EPICS, THEMES E USER STORIES

As *epics* são as principais áreas de trabalho que estruturam o desenvolvimento do projeto, reunindo funcionalidades de elevado valor para profissionais de saúde e administradores. Projetadas para enfrentar desafios como gestão de dados, experiência do utilizador, segurança, escalabilidade e conformidade legal, as *Epics* orientam a equipa no desenvolvimento de soluções alinhadas aos objetivos estratégicos e promovem entregas iterativas e contínuas. Ao serem decompostas em *User Stories*, garantem que cada tarefa do backlog responde às necessidades reais dos *stakeholders*, mantendo o planeamento ágil e adaptável ao longo da evolução do sistema.

EPIC 1: GESTÃO DE DADORES

Theme 1.1: Registo e Atualização de Dadores

- *User Story 1:* Como profissional de saúde, quero registar novos dadores com dados detalhados para manter informações organizadas.
- *User Story 2:* Como profissional de saúde, quero editar informações de dadores existentes para garantir a atualização dos dados.

Theme 1.2: Consulta de Dadores

- *User Story 3:* Como profissional de saúde, quero filtrar a lista de dadores por tipo sanguíneo, intervalo de idades e local de doação para aceder mais rápido às informações.
- *User Story 4:* Como profissional de saúde, quero aplicar filtros para encontrar dadores ativos ou inativos.

Theme 1.3: Histórico de Doações

- *User Story 5:* Como profissional de saúde, quero visualizar o histórico de doações de cada dador para planeamento e análise.

EPIC 2: GESTÃO DE AGENDAMENTOS

Theme 2.1: Gestão de Horários

- *User Story 6:* Como profissional de saúde, quero criar e gerir horários disponíveis para doações para organizar as reservas.
- *User Story 7:* Como profissional de saúde, quero registar agendamentos para os dadores em nome deles.
- *User Story 8:* Como profissional de saúde, quero alterar ou cancelar agendamentos para acomodar mudanças de última hora.

Theme 2.2: Notificações para Dadores

- *User Story 9:* Como profissional de saúde, quero garantir que os dadores recebam lembretes automáticos via e-mail ou SMS sobre os seus agendamentos, para reduzir faltas às marcações.

EPIC 3: RELATÓRIOS E ESTATÍSTICAS

Theme 3.1: Relatórios de Dadores

- *User Story 10:* Como profissional de saúde, quero visualizar relatórios detalhados do número de dadores ativos e inativos.
- *User Story 11:* Como profissional de saúde, quero consultar a distribuição de tipos sanguíneos para planejar campanhas de doação.

Theme 3.2: Visualização Gráfica

- *User Story 12:* Como profissional de saúde, quero visualizar dados em formatos gráficos (barras ou circular) para análise intuitiva.

EPIC 4: AUDITORIA E CONFORMIDADE

Theme 4.1: Logs de Auditoria

- *User Story 13:* Como administrador, quero registar todas as operações críticas para garantir a rastreabilidade.
- *User Story 14:* Como administrador, quero consultar logs de auditoria com informações detalhadas sobre quem realizou cada operação.

Theme 4.2: Conformidade com o RGPD

- *User Story 15:* Como administrador, quero assegurar que todos os dados pessoais sejam encriptados e armazenados em conformidade com o RGPD.
- *User Story 16:* Como administrador, quero implementar controles de acesso por perfil para proteger informações sensíveis.

EPIC 5: ESCALABILIDADE E DESEMPENHO

Theme 5.1: Otimização do Sistema

- *User Story 17:* Como administrador, quero garantir que o desempenho não seja afetado com até 1000 utilizadores simultâneos.
- *User Story 18:* Como administrador, quero criar uma infraestrutura escalável para suportar crescimento futuro.

Theme 5.2: Compatibilidade

- *User Story 19:* Como administrador, quero garantir que o sistema funcione em navegadores modernos para maior acessibilidade.

USER STORIES

De seguida, detalharemos as *User Stories* com os respetivos critérios de aceitação e *mockups*, de forma a permitir uma visão clara do que será desenvolvido neste projeto.

EPIC 1: GESTÃO DE DADORES

Theme 1.1: Registo e Atualização de Dadores

- User Story 1: Registo de Novos Dadores

Como profissional de saúde, quero registar novos dadores com dados detalhados para manter informações organizadas e atualizadas.

Who: Profissional de saúde.

What: Registrar novos dadores com informações detalhadas.

Why: Garantir que os dados dos dadores estão organizados e disponíveis para consulta e análise.

CRITÉRIOS DE ACEITAÇÃO

1- O formulário deve conter os seguintes campos: Nome completo, data de nascimento (com seletor de calendário), contacto, tipo de sangue (*dropdown* com valores pré-definidos) e histórico médico (campo de texto livre), morada e NNU (numérico, número nacional do utente). Todos os campos são obrigatórios.

2- Garantir que as informações são guardadas apenas se todos os campos estão preenchidos e se

foram corretamente preenchidos. No caso de não reunir os critérios, surge um pop-up no ecrã para o profissional de saúde verificar se os dados estão corretamente introduzidos ou se faltam dados.

- 3- Garantir que o número de dador é atribuído automaticamente aquando o registo, seguindo a ordem e as regras recebidas pela base nacional de dadores de sangue.

Banco de recolha de Sangue Nacional - Hospital de Famalicão

LOGO BANCO DE RECOLHA DE SANGUE FAMALICÃO

Bem-vindo(a), João Silva
Terminar sessão

GESTÃO DE DADORES

Dados pessoais *Perfil de doador*

Nome completo: João Carreira

Data de Nascimento: 22/04/1945

Contato: [redacted] rmo, 33 de

Tipo de s [redacted]

Histórico [redacted]

Dados atualizados com sucesso!

Pressione 'ok' para ser encaminhado para a página 'procurar doador' ou 'cancelar' para se manter neste formulário

cancelar OK

Número de dador: 1234343223 NNU: 222345678

Todos os campos são de preenchimento obrigatório

CANCELAR GUARDAR

4- Comportamento dos botões:

- O botão "Guardar" deve guardar as informações introduzidas ou editadas na base nacional de dadores de sangue. Quando introduzidas informações, deve mostrar-se um pop-up ao utilizador de como as informações foram guardadas com sucesso.
- O botão "Cancelar" deve redirecionar para o ecrã anterior sem salvar alterações.

- Aba 'Perfil de Dador': permite consultar informações como o histórico de doações, a data da última doação e o estado (ativo/ inativo). Esta aba é atualizada

automaticamente pela acesso e relação da atividade registada do dador, não permitindo assim a sua edição.

Banco de recolha de Sangue Nacional - Hospital de Famalicão

LOGO BANCO DE RECOLHA DE SANGUE FAMALICÃO

Bem-vindo(a), João Silva
Terminar sessão

GESTÃO DE DADORES

Dados pessoais *Perfil de doador*

Ultima Doação: 19/04/2023

HISTÓRICO DE DOAÇÕES:

LOCAL DE DOAÇÃO	DATA/HORA	CÉDULA PROFISSIONAL DO RESPONSÁVEL PELA COLHEITA	CÓDIGO AMOSTRA BIOLÓGICA
Hospital de São João	19/04/2023 12:13	C12345353-2	#223424323223
Hospital de Santo Tirso	15/02/2020 12:13	C12333333-1	#223424326666

Estado: **ATIVO**

CANCELAR

- User Story 2: Edição de Informações de Dadores

Como profissional de saúde, quero editar informações de dadores existentes para garantir a atualização dos dados.

Who: Profissional de saúde.

What: editar as informações de dadores existentes.

Why: Garantir que os dados estejam sempre atualizados.

CRITÉRIOS DE ACEITAÇÃO

- 1- A partir da página inicial da plataforma, o enfermeiro/médico pode aceder à secção de 'Procurar Dador'. Aí é mostrada a lista de todos os dadores registados, inativos ou ativos.
- 2- Até ao momento, o utilizador tem ao seu dispor 3 formas de procurar pelo dador: deslizar pelo *scroll* da lista, procurar o nome do dador ou pelo seu respetivo número.
- 3- Quando encontra o utilizador, marca o utente e pode clicar no botão "Consultar/Editar dados de dador". Quando clica neste botão, surge um formulário igual ao da mockup da user story anterior com todas as informações do dador. O procedimento de atualização é igual ao descrito na user story anterior a esta.

LOGO BANCO DE RECOLHA DE SANGUE FAMALICÃO

Bem-vindo(a), João Silva
Terminar sessão

PROCURAR DADOR

Número do dador

ou

Nome do dador

Pesquisar

Número	Nome	
12344556754	João Pedro Madeira	<input checked="" type="checkbox"/>
12345678765	Maria Joana Petúnia	<input type="checkbox"/>
35654654656	Pedro Madeira	<input type="checkbox"/>
65445678778	Joana Maria Silva	<input type="checkbox"/>
88767878987	Judite Sousa	<input type="checkbox"/>

Criar Novo Perfil de Dador

Consultar/Editar dados de dador

DESMARCAR SELEÇÃO

cancelar

4- Garantir que a opção de 'desmarcar seleção' apenas surge se tivermos selecionado algum dador da lista.

5- Após a edição, os dados atualizados devem ser refletidos na lista de dadores.

6- Comportamento dos botões:

- Criar Novo Perfil de Dador: Dirige-nos para o formulário da *user story* 1
- Cancelar: Permite-nos voltar à página inicial.

7- Garantir um um histórico de versões, onde se visualiza o antes e depois das

alterações no *log* de auditoria.

Theme 1.2: Consulta de Dadores

- User Story 3: Consulta de Dadores por Tipo Sanguíneo, idade e local de doação

Como profissional de saúde, quero filtrar a lista de dadores por tipo sanguíneo, intervalo de idades e local de doação para aceder mais rápido às informações.

Who: Profissional de saúde.

What: Filtrar a lista de dadores por Tipo Sanguíneo, idade e local de doação

Why: Para economizar tempo e garantir que os dados mais relevantes estejam disponíveis de forma imediata

Banco de recolha de Sangue Nacional - Hospital de Famalicão

LOGO BANCO DE RECOLHA DE SANGUE FAMALICÃO Bem-vindo(a), João Silva Terminar sessão

PROCURAR DADOR

Número do dador ou Nome do dador Pesquisa

Pesquisar por:


- Tipo de sangue: ---
- Intervalo de idade: ---- a ----
- Local de Doação: _____


OK

Criar Novo Perfil de Dador Consultar/Editar dados de dador DESMARCAR SELEÇÃO

CRITÉRIOS DE ACEITAÇÃO

1- Garantir que na lista de dadores, o enfermeiro/médico tem à sua disposição um ícone de filtro.

2- Garantir que o comportamento do ícone está de acordo, se os mesmos forem selecionados ou não.  não estão selecionados filtros;

 estão selecionados filtros de pesquisa.

3- Permitir ao utilizador pesquisar pelo número ou nome de dador, em combinação com os filtros adicionais.

4- Caso a pesquisa demore mais de 3 segundos, surgir uma mensagem a dizer “Estamos a processar o seu pedido. Por favor aguarde...”

Banco de recolha de Sangue Nacional - Hospital de Famalicão

LOGO BANCO DE RECOLHA DE SANGUE FAMALICÃO Bem-vindo(a), João Silva Terminar sessão

PROCURAR DADOR

Número do dador ou Nome do dador Pesquisa

Número	Nome	
12344556754	João Pedro Madeira	<input checked="" type="checkbox"/>
12345678765	Maria Joana Petúnia	<input type="checkbox"/>
35654654656	Pedro Madeira	<input type="checkbox"/>
65445678778	Joana Maria Silva	<input type="checkbox"/>
88767878987	Judite Sousa	<input type="checkbox"/>

Criar Novo Perfil de Dador Consultar/Editar dados de dador cancelar DESMARCAR SELEÇÃO

- User Story 4: Filtro de Dadores Ativos e Inativos

Como profissional de saúde, quero aplicar filtros para encontrar dadores ativos ou inativos.

Who: Profissional de saúde.

What: Aplicar filtros para encontrar dadores ativos ou inativos.

Why: Facilitar a gestão e análise de dadores.

Banco de recolha de Sangue Nacional - Hospital de Famalicão

LOGO BANCO DE RECOLHA DE SANGUE FAMALICÃO Bem-vindo(a), João Silva Terminar sessão

PROCURAR DADOR

Pesquisar por:

- Tipo de sangue: --- ▾
- Intervalo de idade: ---- a ----
- Local de Doação: _____

Filtros Avançados ▾ OK

Criar Novo Perfil de Dador Consultar/Editar dados de dador

cancelar DESMARCAR SELEÇÃO

CRITÉRIOS DE ACEITAÇÃO

1- O sistema deve permitir que o utilizador filtre os registos por tipo de sangue e/ou intervalo de datas, exibindo apenas os resultados que atendam a todos os critérios selecionados.

2- Garantir que os de filtros apareçam com valores padrão (ex.: campos vazios ou a opção "Todos" selecionada) ao abrir o menu, e que o botão "OK" só seja habilitado se ao menos um filtro for preenchido.

Banco de recolha de Sangue Nacional - Hospital de Famalicão

LOGO BANCO DE RECOLHA DE SANGUE FAMALICÃO Bem-vindo(a), João Silva Terminar sessão

PROCURAR DADOR

Pesquisar por:

- Tipo de sangue: --- ▾
- Intervalo de idade: ---- a ----
- Local de Doação: _____

Filtros Avançados ▾

☐ Dador Inativo

☐ Dador Ativo (doação à menos de 6 meses)

OK

Criar Novo Perfil de Dador Consultar/Editar dados de dador

cancelar DESMARCAR SELEÇÃO

3- Garantir uma resposta em menos de 3 segundos ao utilizador, mesmo com todos os filtros selecionados. Caso demore mais, devemos informar o utilizador: "Estamos a processar o seu pedido. Por favor aguarde..."

Theme 1.3: Histórico de Doações

- *User Story 5: Histórico de doações*

Como profissional de saúde, quero visualizar o histórico de doações de cada dador para planeamento e análise.

Who: Profissional de saúde.

What: Consultar a lista completa das doações efetuadas, incluindo informações como data e tipo de sangue.

Why: Para ter uma visão das doações efetuadas num determinado dia ou data específica, facilitar o planeamento e a análise das doações realizadas.

Banco de recolha de Sangue Nacional - Hospital de Famalicão

LOGO BANCO DE RECOLHA DE SANGUE FAMALICÃO Bem-vindo(a), João Silva Terminar sessão

Histórico de doações

Pesquisar

Número	Nome	Data	Tipo sanguíneo
12344556754	João Pedro Madeira	Dia: 22/12/2024 Hora: 14:30	O+
12345678765	Maria Joana Petúnia	Dia: 22/12/2024 Hora: 18:00	AB-
35654654656	Pedro Madeira	Dia: 23/12/2024 Hora: 08:00	AB+
65445678778	Joana Maria Silva	Dia: 24/12/2024 Hora: 15:30	A-

Banco de recolha de Sangue Nacional - Hospital de Famalicão

LOGO BANCO DE RECOLHA DE SANGUE FAMALICÃO Bem-vindo(a), João Silva Terminar sessão

Histórico de doações

Pesquisar

Pesquisar por:

- Tipo de sangue:
- Data: -- / -- / ---- a -- / -- / ----

Número	Nome	Data	Tipo sanguíneo
12344556754			
12345678765			
35654654656			
65445678778	Joana Maria Silva	Dia: 24/12/2024 Hora: 15:30	A-
88767878987	Judite Sousa	Dia: 28/12/2024 Hora: 19:30	B+
65445678778	Elisabete Coelho	Dia: 02/01/2025 Hora: 14:30	O-


CRITÉRIOS DE ACEITAÇÃO

1-O sistema deve apresentar uma lista na qual, em qualquer campo da tabela, pelo menos um, tenha o conteúdo introduzido na caixa de pesquisa.

2-O sistema deve devolver a mensagem "Nenhum resultado encontrado", caso a pesquisa não retorne resultados.

3-Garantir que seja possível organizar os resultados por qualquer coluna da tabela, mas por predefinição, os resultados apresentados sejam ordenados pela data.

4- Garantir que na lista de dadores, o enfermeiro/médico tem à sua disposição um ícone de filtro.

5-Garantir que o comportamento do ícone está de acordo, se os mesmos forem selecionados ou não.  não estão selecionados filtros;



estão selecionados filtros de

pesquisa.

Theme 2.1: Gestão de Horários

- **User Story 6:** Criar e Gerir Horários Disponíveis

Como profissional de saúde, quero criar e gerir horários disponíveis para doações para organizar as reservas.

Who: Profissional de saúde.

What: Criar e gerir horários de doações.

Why: Assegurar uma organização eficiente das reservas e do fluxo de dadores

CRITÉRIOS DE ACEITAÇÃO

- 1- O sistema deve permitir a criação de horários disponíveis, especificando data, hora de início e fim, e número máximo de vagas por horário.
- 2- O profissional de saúde deve conseguir visualizar, editar ou remover horários previamente criados.
- 3- O sistema deve impedir a criação de horários sobrepostos ou inválidos (ex.: fim anterior ao início).

- **User Story 7:** Registo de Agendamentos

Como profissional de saúde, quero registar agendamentos para os dadores em nome deles.

Who: profissional de saúde,

What: registar agendamentos para os dadores,

Why: Para assegurar que todos os horários sejam ocupados

CRITÉRIOS DE ACEITAÇÃO

- 1-Garantir que na página inicial da plataforma existe um botão para adicionar um utente à agenda.
- 2-O sistema deve permitir ao profissional de saúde selecionar um horário disponível e associá-lo a um dador, inserindo os dados necessários.
- 3- Após registar o agendamento, o sistema deve exibir uma mensagem de confirmação e enviar um aviso ao dador, através do sistema de notificações.
- 4-O sistema deve impedir o registo de mais de dois agendamentos para o mesmo horário.

- **User Story 8:** Alteração ou Cancelamento de Agendamentos

Como profissional de saúde, quero alterar ou cancelar agendamentos.

Who profissional de saúde,

What alterar ou cancelar agendamentos para acomodar mudanças de última hora,

Why garantir flexibilidade nas reservas

CRITÉRIOS DE ACEITAÇÃO

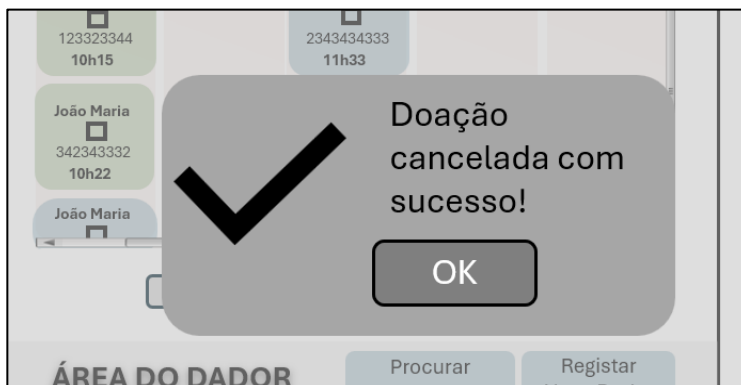
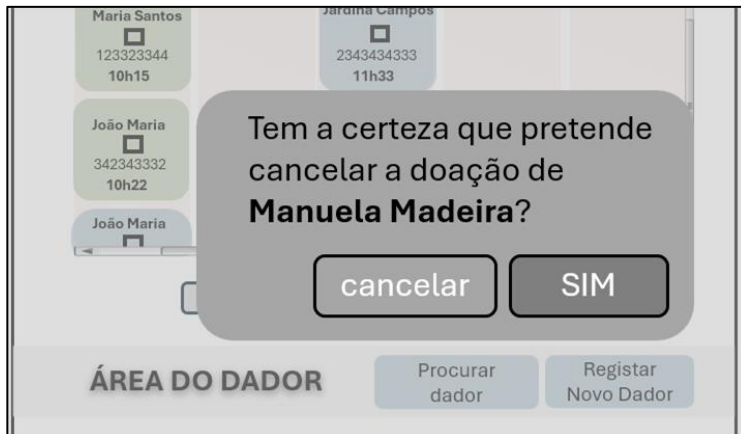
1- O sistema deve permitir ao profissional de saúde modificar os detalhes de um agendamento, como horário ou dados do dador, desde que haja disponibilidade.

2- Deve ser possível cancelar agendamentos, removendo o dador do horário correspondente.

3- O sistema deve guardar um registo das alterações e cancelamentos, incluindo a data e quem realizou a alteração.

4- O profissional pode cancelar um agendamento através da página inicial no botão “cancelar doação”, selecionando previamente a pessoa do calendário do dia em vigor. O profissional deve receber uma mensagem de confirmação no ecrã “Doação cancelada com sucesso!”. Simultaneamente, o sistema envia um email/SMS ao dador a confirmar o cancelamento.

5- Caso o profissional queira cancelar um utente de uma data específica, pode através da lupa no canto superior direito do calendário, aceder ao ‘Histórico de Doações’ já analisado anteriormente e, procurar pelo horário utente pretendido. Depois de encontrar o pretendido, o mesmo clica duas no nome do dador, selecionando-o. Aí temos a opção “Criar Agendamento” e devemos garantir também um botão de “Cancelar Agendamento”. Os



Banco de recolha de Sangue Nacional - Hospital de Famalicão

LOGO BANCO DE RECOLHA DE SANGUE FAMILICÃO Bem-vindo(a), João Silva Terminar sessão

Histórico de doações

Pesquisar

Número	Nome	Data	Tipo sanguíneo
12344556754	João Pedro Madeira	Dia: 22/12/2024 Hora: 14:30	O+
12345678765	Maria Joana Petúnia	Dia: 22/12/2024 Hora: 18:00	AB-
35654654656	Pedro Madeira	Dia: 23/12/2024 Hora: 08:00	AB+
65445678778	Joana Maria Silva	Dia: 24/12/2024 Hora: 15:30	A-
88767878987	Judite Sousa	Dia: 28/12/2024 Hora: 19:30	B+
65445678778	Elisabete Coelho	Dia: 02/01/2025 Hora: 14:30	O-

pop-ups de confirmação de cancelamento são idênticos e o utente também recebe de forma, quase instantânea a confirmação por e-mail/SMS.

Theme 2.2: Notificações para Dadores

- *User Story 9: Lembretes Automáticos*

Como profissional de saúde, quero garantir que os dadores recebam lembretes automáticos via e-mail ou SMS sobre os seus agendamentos, para reduzir faltas às marcações.

Who: Profissional de saúde.

What: Configurar notificações automáticas para lembrar os dadores sobre os seus agendamentos.

Why: Garantir a adesão dos dadores aos horários marcados

CRITÉRIOS DE ACEITAÇÃO

- 1- O sistema deve enviar lembretes automáticos por e-mail ou SMS para os dadores com informações do agendamento (data, hora e local), com antecedência configurada pelo sistema (ex.: 24 ou 48 horas antes).
- 2- O sistema deve permitir ao profissional de saúde verificar se os lembretes foram enviados com sucesso, através de um registo de notificações.
- 3- O lembrete enviado deve conter informações completas e corretas do agendamento, de acordo com o que está registado no sistema.



EPIC 3: RELATÓRIOS E ESTATÍSTICAS**Theme 3.1: Relatórios de Dadores**

- *User Story 10: Visualizar relatórios do número de dadores ativos e inativos*
Como profissional de saúde, quero visualizar relatórios detalhados do número de dadores ativos e inativos.

Who: Como profissional de saúde.

What: Quero visualizar relatórios detalhados do número de dadores ativos e inativos.

Why: Para analisar e melhorar a gestão dos dadores, priorizando ações necessárias para aumentar a participação ativa.

CRITÉRIOS DE ACEITAÇÃO

- 1- O sistema deve exibir o número total de dadores, distinguindo entre ativos e inativos do hospital e a nível nacional, com base na data da última doação.
 - 2- Os dados devem ser apresentados numa tabela simples e legível, exportável para PDF sumariado.
 - 3- A página inicial da plataforma deve conter um link direto para aceder aos relatórios integrados e fazer o respetivo download.
 - 4- Garantir que a data da última atualização de dados é visível a partir da página inicial da plataforma, sendo garantido também um link para forçar a atualização de dados de forma que o profissional tenha acesso aos dados mais recentes.
- *User Story 11: Consultar distribuição de tipos sanguíneos*
Como profissional de saúde, quero consultar a distribuição de tipos sanguíneos para planear campanhas de doação.

Who: Como profissional de saúde.

What: Quero consultar a distribuição de tipos sanguíneos a nível local, em determinado hospital ou panorama a nível nacional.

Why: Para planear campanhas de doação, garantindo um equilíbrio entre os diferentes tipos de sangue.

CRITÉRIOS DE ACEITAÇÃO

- 1- O sistema deve apresentar os dados de distribuição em formato de tabela ou gráfico (barras ou circular).
- 2- Os gráficos devem incluir legendas claras e uma visualização interativa ao passar o cursor.
- 3- Deve ser possível filtrar consultar os dados de outros bancos de recolha de sangue através do link a partir da página inicial da plataforma.
- 4- O sistema deve permitir exportar todos os dados através do link “Gerar PDF integral” presente na página inicial da plataforma atualizados à data atual.

Theme 3.2: Visualização Gráfica

- *User Story 12: Exibir dados em formatos gráficos (barras ou circular)*
Como profissional de saúde, quero visualizar dados em formatos gráficos (barras ou circular) para análise intuitiva.

Who: Como profissional de saúde.

What: Quero visualizar dados em formatos gráficos (barras ou circular).

Why: Para realizar análises intuitivas e facilitar a interpretação de informações.

Segue os **CRITÉRIOS DE ACEITAÇÃO** presentes no US 11.

EPIC 4: AUDITORIA E CONFORMIDADE

Theme 4.1: Logs de Auditoria

- *User Story 13: Registrar operações críticas*
Como administrador, quero registrar todas as operações críticas para garantir a rastreabilidade.

Who: Como administrador.

What: Quero registrar todas as operações críticas realizadas no sistema.

Why: Para garantir a rastreabilidade e conformidade com requisitos legais.

CRITÉRIOS DE ACEITAÇÃO

- 1- Todas as operações críticas (ex.: registro, edição e eliminação de dados) devem ser registradas automaticamente.
- 2- Cada log deve incluir o tipo de operação, utilizador responsável, data e hora.
- 3- Os dados dos logs devem ser armazenados de forma segura, acessíveis apenas a utilizadores com perfil de administrador.
- 4- Garantir uma retenção mínima de 12 meses para os logs.

- *User Story 14: Consultar logs de auditoria com detalhes*
Como administrador, quero consultar logs de auditoria com informações detalhadas sobre quem realizou cada operação.

Who: Como administrador.

What: Quero consultar logs de auditoria com informações detalhadas sobre quem realizou cada operação.

Why: Para monitorizar ações no sistema e identificar problemas ou comportamentos suspeitos.

CRITÉRIOS DE ACEITAÇÃO

- 1- O sistema deve permitir buscas nos logs utilizando filtros como tipo de operação, utilizador e intervalo de datas.
- 2- Os resultados devem ser apresentados em formato de tabela ordenável por qualquer coluna.

- 3- Deve ser possível exportar os logs filtrados em formato CSV para auditorias externas.
- 4- Consultas demoradas devem exibir mensagens de progresso se excederem 5 segundos.

Theme 4.2: Conformidade com o RGPD

- *User Story 15: Garantir encriptação de dados pessoais e conformidade*
Como administrador, quero assegurar que todos os dados pessoais sejam encriptados e armazenados em conformidade com o RGPD.

Who: Como administrador.

What: Quero assegurar que todos os dados pessoais sejam encriptados e armazenados em conformidade com o RGPD.

Why: Para proteger a privacidade dos utilizadores e cumprir com as obrigações legais.

CRITÉRIOS DE ACEITAÇÃO

- 1- Todos os dados pessoais armazenados no sistema devem ser encriptados com algoritmos seguros.
- 2- As comunicações entre cliente e servidor devem ser protegidas por HTTPS.
- 3- Deve ser realizada uma auditoria inicial para verificar a conformidade com o RGPD.
- 4- Mensagens de erro e logs técnicos não devem expor dados sensíveis.

- *User Story 16: Controlos de acesso por perfil para proteger informações sensíveis*
Como administrador, quero implementar controlos de acesso por perfil para proteger informações sensíveis.

Who: Como administrador.

What: Quero implementar controlos de acesso por perfil para proteger informações sensíveis.

Why: Para garantir que cada utilizador apenas acede às funcionalidades e dados permitidos pelo seu perfil.

CRITÉRIOS DE ACEITAÇÃO

1. O sistema deve suportar perfis de utilizador com diferentes permissões (ex.: administrador, operador, leitor).
2. Operações sensíveis (ex.: edição de dados, consulta de logs) devem ser restritas por perfil.
3. Deve ser possível gerir permissões diretamente na interface administrativa por perfil e categoria de utilizador.
4. Todas as tentativas de acesso não autorizadas devem ser registadas nos logs.

EPIC 5: ESCALABILIDADE E DESEMPENHO

Theme 5.1: Otimização do Sistema

- *User Story 17: Suportar até 1000 utilizadores simultâneos*
Como administrador, quero garantir que o desempenho não seja afetado com até 1000 utilizadores simultâneos.
Who: Como administrador.
What: Quero garantir que o desempenho do sistema não seja afetado com até 1000 utilizadores simultâneos.
Why: Para assegurar a estabilidade do sistema durante períodos de alta utilização.

CRITÉRIOS DE ACEITAÇÃO

- 1- O sistema deve demonstrar, em testes de carga, um tempo de resposta inferior a 2 segundos para operações críticas com até 1000 utilizadores simultâneos.
 - 2- Deve ser possível monitorizar métricas de desempenho em tempo real.
 - 3- Testes de carga devem ser realizados com cenários representativos.
- *User Story 18: Criar uma infraestrutura escalável*
Como administrador, quero criar uma infraestrutura escalável para suportar crescimento futuro.
Who: Como administrador.
What: Quero criar uma infraestrutura escalável para suportar crescimento futuro.
Why: Para garantir que o sistema acompanhe o aumento no número de utilizadores e no volume de dados.

CRITÉRIOS DE ACEITAÇÃO

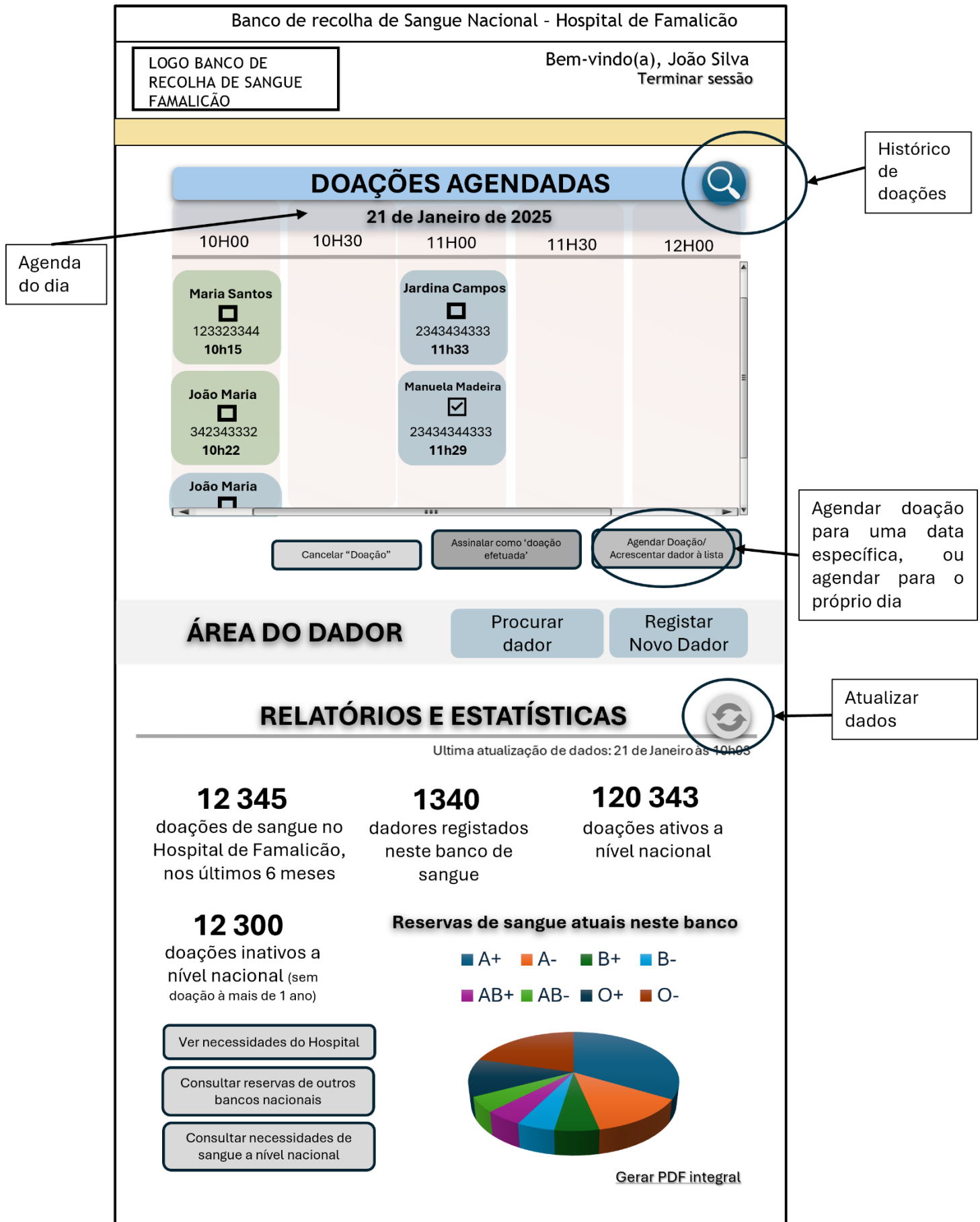
1. A infraestrutura deve permitir a adição de novos servidores ou serviços sem interrupção do sistema.
2. Deve suportar crescimento até 1 milhão de registos sem degradação significativa do desempenho.
3. A arquitetura deve permitir integrações futuras com sistemas externos.
4. Testes de escalabilidade devem ser realizados para validar a capacidade de crescimento.

Theme 5.2: Compatibilidade

- *User Story 19: Garantir compatibilidade com navegadores modernos*
Como administrador, quero garantir que o sistema funcione em navegadores modernos para maior acessibilidade.
Who: Como administrador.
What: Quero garantir que o sistema funcione em navegadores modernos.
Why: Para assegurar acessibilidade e consistência na experiência de utilizador.

CRITÉRIOS DE ACEITAÇÃO

- 1- O sistema deve ser funcional e apresentar corretamente todos os elementos nos navegadores mais padronizados, tais como, Google Chrome, Mozilla Firefox, Microsoft Edge e Safari.
- 2- Todos os recursos e funcionalidades do sistema devem passar em testes de compatibilidade em navegadores modernos, garantindo consistência de design e usabilidade em pelo menos 95% dos cenários de uso simulados.

MOCKUP :: PÁGINA INICIAL DA PLATAFORMA

STORYPOINTS

Os *story points* são uma medida abstrata utilizada para estimar o esforço relativo necessário para completar *user stories*, focando-se na complexidade técnica, esforço da equipa e o impacto no sistema e valor entregue ao utilizador final.

<i>Epic</i>	<i>Theme</i>	<i>User Story</i>	<i>Story Points</i>
Gestão de Dadores	Registo e Atualização de Dadores	US 1: Registo de novos dadores	5
		US 2: Edição de informações de dadores	5
	Consulta de Dadores	US 3: Filtros por tipo sanguíneo, intervalo de idades, e local de doação	5
		US 4: Filtros para dadores ativos/inativos	3
	Histórico de Doações	US 5: Visualizar histórico de doações	8
Gestão de Agendamentos	Gestão de Horários	US 6: Criar e gerir horários disponíveis para doações	8
		US 7: Registrar agendamentos	5
		US 8: Alterar ou cancelar agendamentos	6
	Notificações para Dadores	US 9: Configurar lembretes automáticos via e-mail/SMS	8
Relatórios e Estatísticas	Relatórios de Dadores	US 10: Visualizar relatórios do número de dadores ativos e inativos	8
		US 11: Consultar distribuição de tipos sanguíneos	5
	Visualização Gráfica	US 12: Exibir dados em formatos gráficos (barras ou circular)	8
Auditoria e Conformidade	Logs de Auditoria	US 13: Registrar operações críticas	5
		US 14: Consultar logs de auditoria com detalhes	5
	Conformidade com RGPD	US 15: Garantir encriptação de dados pessoais e conformidade	8
		US 16: Controlos de acesso por perfil para proteger informações sensíveis	8
Escalabilidade e Desempenho	Otimização do Sistema	US 17: Suportar até 1000 utilizadores simultâneos	8
		US 18: Criar uma infraestrutura escalável	5
	Compatibilidade	US 19: Garantir compatibilidade com navegadores modernos	3

Vamos agora analisar os *user points* atribuídos às diferentes *user stories*, de forma sucinta:

US 1: Registo de novos dadores (5 pontos)

Implementação de um formulário com validação de dados e integração com a base nacional de dadores. Esforço moderado e Prioridade Fundamental

US 2: Edição de informações de dadores (5 pontos)

Similar ao registo, mas inclui carregamento de dados e lógica para atualizar registos existentes.
Complexidade equivalente à **US 1**.

US 3: Filtros por tipo sanguíneo, idade e local de doação (5 pontos)

Envolve a aplicação de múltiplos filtros e otimização da pesquisa, necessitando atenção à usabilidade e performance.

US 4: Filtros para dadores ativos/inativos (3 pontos)

Simples extensão do sistema de filtros, com menor esforço técnico em comparação à **US 3**.

US 5: Histórico de doações (8 pontos)

Tarefa complexa que requer:

- Agregação e exibição de grandes volumes de dados históricos.
- Necessário garantir respostas rápidas, mesmo em cenários com muitos registos.
- Mensagens claras para consultas demoradas ou ausência de resultados.
- Integra com funcionalidades existentes (ex.: registo de dadores) e impacta futuras (ex.: relatórios).

US 6: Criar e gerir horários disponíveis para doações (8 pontos)

Criar e gerir horários para doações envolve organizar agendamentos, verificar disponibilidade e evitar conflitos. Prioridade Fundamental.

US 7: Registrar agendamentos (5 pontos)

Envolve a validação e uma interface amigável, mas é tecnicamente menos desafiadora do que a US6, já que o fluxo é mais direto e não envolve tantas interações complexas.

US 8: Alterar ou cancelar agendamentos (6 pontos)

Tarefa complexa que requer:

- Garantir que as alterações não criem conflitos ou inconsistências nos dados.
- Atualizar o estado do agendamento e libertar a vaga para permitir uma nova remarcação naquele horário.
- Enviar notificações claras e precisas para os dadores.
- Exigir mais testes para garantir que todos os fluxos funcionem corretamente.

US 9: Configurar lembretes automáticos via e-mail/SMS (8 pontos)

A tarefa envolve configurar a lógica para enviar lembretes automáticos por e-mail e SMS, integrando esses serviços com o sistema de agendamentos. É necessário garantir que os lembretes sejam enviados no momento certo, sem falhas, o que implica a criação de regras de agendamento e a verificação de que a comunicação externa funciona de forma fiável.

US 10: Visualizar relatórios do número de dadores ativos e inativos (8 pontos)

A visualização de relatórios envolve o processamento de dados para gerar uma visão clara sobre o estado atual dos dadores. Isso pode envolver cálculos e extração de dados de várias fontes, o que adiciona complexidade. A interface precisa ser intuitiva, o que aumenta o esforço de desenvolvimento.

US 11: Consultar distribuição de tipos sanguíneos (5 pontos)

A consulta da distribuição dos tipos sanguíneos é um relatório relativamente simples, onde o maior desafio é apresentar os dados de forma visualmente atraente e compreensível. A complexidade técnica é moderada, pois envolve a consulta a dados e a geração de gráficos.

US 12: Exibir dados em formatos gráficos (barras ou circular) (8 pontos)

A visualização gráfica de dados exige não apenas o cálculo correto das estatísticas, mas também uma implementação de gráficos interativos ou imagens. A integração de bibliotecas de visualização e a garantia de que a experiência do utilizador seja fluída tornam esta tarefa relativamente complexa.

US 13: Registrar operações críticas (5 pontos)

O registo de operações críticas é uma funcionalidade fundamental, mas não excessivamente complexa. Requer apenas o registo de *logs* de atividades e a garantia de que esses *logs* são armazenados de forma segura. A complexidade técnica é média.

US 14: Consultar *logs* de auditoria com detalhes (5 pontos)

Consultar *logs* envolve uma simples interface de pesquisa, mas requer que os dados sejam acedidos rapidamente, mesmo com grandes volumes de informações. Complexidade técnica é idêntica ao US anterior.

US 15: Garantir encriptação de dados pessoais e conformidade (8 pontos)

A encriptação de dados e a conformidade com o RGPD envolvem requisitos de segurança e privacidade, que são críticos para a integridade e conformidade legal do sistema. A implementação de medidas de segurança é complexa e requer atenção especial para garantir que todos os dados pessoais estejam protegidos.

US 16: Controlos de acesso por perfil para proteger informações sensíveis (8 pontos)

A implementação de um controlo de acesso por perfil envolve várias etapas cruciais, como a definição de papéis e permissões, a integração da autenticação, a verificação de acessos e a realização de testes rigorosos para garantir a segurança. Embora não seja uma tarefa excessivamente complexa, a atenção à segurança torna-a crítica.

US 17: Suportar até 1000 utilizadores simultâneos (8 pontos)

Exige uma melhora do desempenho do sistema para que não haja lentidão ou falhas. Isso requer uma análise detalhada da infraestrutura e do desempenho da aplicação.

US 18: Criar uma infraestrutura escalável (5 pontos)

Embora exija um esforço técnico, o foco está mais no planeamento do que na execução intensiva.

US 19: Garantir compatibilidade com navegadores modernos (3 pontos)

Garantir que o sistema funcione em navegadores modernos é uma tarefa simples, especialmente em projetos novos. Como os navegadores mais usados (Chrome, Firefox, Edge e Safari) já são bem suportados, a complexidade técnica é baixa.

DEFINIÇÃO DE CADA SPRINT

- Duração dos *sprints*: **2** semanas
- Reuniões diárias de **15** minutos para manter o alinhamento da equipa e ajustar prioridades
- Número máximo de *story points* por *sprint*: **30** *story points* (pode ser ajustado conforme o progresso do projeto e as avaliações das retrospectivas.)
- As reuniões ao longo de cada *sprint* são essenciais para garantir a comunicação e o progresso contínuo do projeto. A reunião de planeamento do *sprint* ocorre no início de cada *sprint*, onde a equipa define as *user stories* a serem trabalhadas, estima o *esforço* necessário e alinha as expectativas. Nesta reunião, todos os membros da equipa Scrum, incluindo o *Product Owner*, o *Scrum Master* e a equipa de desenvolvimento, participam ativamente.

As reuniões diárias, também chamadas de *daily stand-ups*, realizam-se todos os dias úteis, de segunda a sexta-feira, com uma duração curta de 10 a 15 minutos. Nelas, cada membro da equipa partilha o que fez no dia anterior, o que planeia fazer no dia atual e se há algum bloqueio que possa impedir o progresso. O objetivo é garantir que todos estejam alinhados e identificar rapidamente obstáculos que possam surgir.

No final de cada *sprint*, a equipa realiza a reunião de revisão do *sprint*, onde são apresentadas as funcionalidades desenvolvidas durante o ciclo. Durante esta reunião, também se realizam as demonstrações das funcionalidades, que são momentos importantes para mostrar o que foi desenvolvido e garantir que tudo está a funcionar como esperado. Antes da reunião de revisão, validamos as funcionalidades num ambiente de teste para evitar problemas e garantir que o sistema está estável. As demos, que serão curtas (entre 30 a 45 minutos), vão focar-se exclusivamente no que foi entregue no *sprint*. Os cenários apresentados serão baseados em casos reais, mostrando como o sistema pode ser útil no dia a dia dos profissionais de saúde. O *feedback* dado durante a demo será registado e usado para melhorar o sistema e organizar melhor o *backlog*. Este processo garante que o projeto avança sempre de acordo com as necessidades dos utilizadores, de forma prática e colaborativa.

Após a *Sprint Review*, realiza-se a reunião de retrospectiva do *sprint*, onde a equipa reflete sobre o que correu bem e o que pode ser melhorado. O objetivo é identificar ações concretas para melhorar o processo de trabalho nos próximos *sprints*. Esta reunião é focada na melhoria contínua e deve envolver toda a equipa *Scrum*, para que todos possam contribuir com sugestões e ideias.

Em paralelo, ocorre o refinamento do *backlog*, uma atividade contínua ao longo do sprint, normalmente a meio do ciclo. O objetivo é rever o *Product Backlog*, garantindo que as *user stories* estão bem definidas e prontas para serem trabalhadas no próximo sprint. O *Product Owner* e a equipa de desenvolvimento participam ativamente para esclarecer dúvidas, definir prioridades e preparar o *backlog* para os próximos sprints.

SPRINT 1: FOCO NO PRODUTO MÍNIMO VIÁVEL (MVP)

Objetivo: Implementar funcionalidades básicas para gestão de dadores.

User Stories selecionadas:

- US 1:** Registo de novos dadores (5 pontos).
- US 2:** Edição de informações de dadores (5 pontos).
- US 3:** Filtros por tipo sanguíneo, idade e local (5 pontos).
- US 4:** Filtros para dadores ativos/inativos (3 pontos).
- US** Visualizar histórico de doações de dadores (8 pontos).
- US 15:** Garantir encriptação de dados pessoais e conformidade com o RGPD (8 pontos).

Total: 31 pontos.

Funcionalidades da Demo Prevista:

- Formulário funcional com validação de dados e integração com a base de dadores.
- Capacidade de atualizar registos existentes com validação.
- Pesquisa e filtros por tipo sanguíneo, idade e localização.
- Clareza das mensagens de erro ou sucesso.
- Dados pessoais encriptados e acessíveis de forma segura.

SPRINT 2: INTEGRAÇÃO DE AGENDAMENTOS E SEGURANÇA

Objetivo: Introduzir a gestão de agendamentos e notificações.

User Stories selecionadas:

- US 6:** Criar e gerir horários disponíveis (8 pontos).
- US 7:** Registrar agendamentos (5 pontos).
- US 8:** Alterar ou cancelar agendamentos (6 pontos).
- US 9:** Configurar lembretes automáticos via e-mail/SMS (8 pontos).

Total: 27 pontos.

Funcionalidades da Demo Prevista:

- Criação e edição de horários disponíveis para doação.
- Associação de horários a dadores.
- Envio de lembretes automáticos por e-mail ou SMS.

SPRINT 3: RELATÓRIOS E ESTATÍSTICAS

Objetivo: Implementar relatórios e funcionalidades gráficas.

User Stories selecionadas:

US 10: Visualizar relatórios do número de dadores ativos e inativos (8 pontos).

US 11: Consultar distribuição de tipos sanguíneos (5 pontos).

US 12: Exibir dados em formatos gráficos (8 pontos).

US 13: Registrar logs de auditoria para rastreabilidade (5 pontos).

US 14: Consultar *logs* de auditoria com detalhes (5 pontos).

Total: 31 pontos.

Funcionalidades da Demo Prevista:

- Gerar relatórios com número de dadores ativos/inativos e distribuição de tipos sanguíneos.
- Dados apresentados em gráficos de barras e circulares.
- Possibilidade de exportar para PDF ou Excel

SPRINT 4: ESCALABILIDADE E DESEMPENHO

Objetivo: Preparar o sistema para operações em larga escala.

User Stories selecionadas:

US 16: Implementar controles de acesso por perfil (5 pontos).

US 17: Suportar até 1000 utilizadores simultâneos (8 pontos).

US 18: Criar uma infraestrutura escalável (5 pontos).

US 19: Garantir compatibilidade com navegadores modernos (3 pontos).

Total: 21 pontos

Funcionalidades da Demo Prevista:

- Testes de simulação de carga com 1000 utilizadores simultâneos sem degradação do desempenho onde se verifica a Estabilidade e rapidez do sistema com os utilizadores.
- Tempo de resposta dentro dos limites definidos (máximo 2 segundos).
- Teste de funcionamento adequado nos navegadores Chrome, Firefox, Edge e Safari e verificação da consistência na apresentação e funcionalidade entre diferentes navegadores.
- Mostrar capacidade de escalabilidade sem comprometer a usabilidade.

INTEGRAÇÃO DO JIRA

Para a organização e acompanhamento do desenvolvimento do projeto, utilizámos o *Jira*, uma ferramenta amplamente reconhecida por sua flexibilidade e capacidade de gerir equipas ágeis. O Jira permitiu-nos organizar o backlog, definir prioridades, atribuir tarefas às equipas e acompanhar o progresso do projeto de forma visual e eficiente.

OBJETIVOS DO USO DO JIRA

1. Centralizar a Gestão do Backlog: Todas as tarefas foram organizadas no backlog, com suas prioridades, responsáveis e estimativas de tempo devidamente documentadas.
2. Acompanhar o Progresso do Desenvolvimento: Por meio de relatórios automáticos e acompanhamento do progresso das sprints, conseguimos visualizar o andamento de cada etapa.
3. Facilitar a Comunicação: A ferramenta possibilitou discussões centralizadas nas tarefas e notificações automáticas para toda a equipa.

PREPARAÇÃO PARA A INTEGRAÇÃO

Utilizamos o Jira para gestão da equipa e aplicação da metodologia SCRUM

1. Começamos por criar um projeto na plataforma
 - a. Selecionamos a opção "Create Project"
 - b. Selecionamos o template: "Scrum"
 - c. Adicionamos o tipo de projeto: "Team-managed"
 - d. Escolhemos o nome do projeto (rede-vida)
2. Prosseguimos para a criação product backlog
 - a. Selecionamos o nosso projeto (rede-vida) e acedemos à opção "Backlog" no menu lateral esquerdo
 - b. Na secção "Backlog", utilizamos o botão "+ Create" e colocamos o nome da tarefa.

CRIAÇÃO DE ELEMENTOS NO JIRA

1. Criação de *Epics*

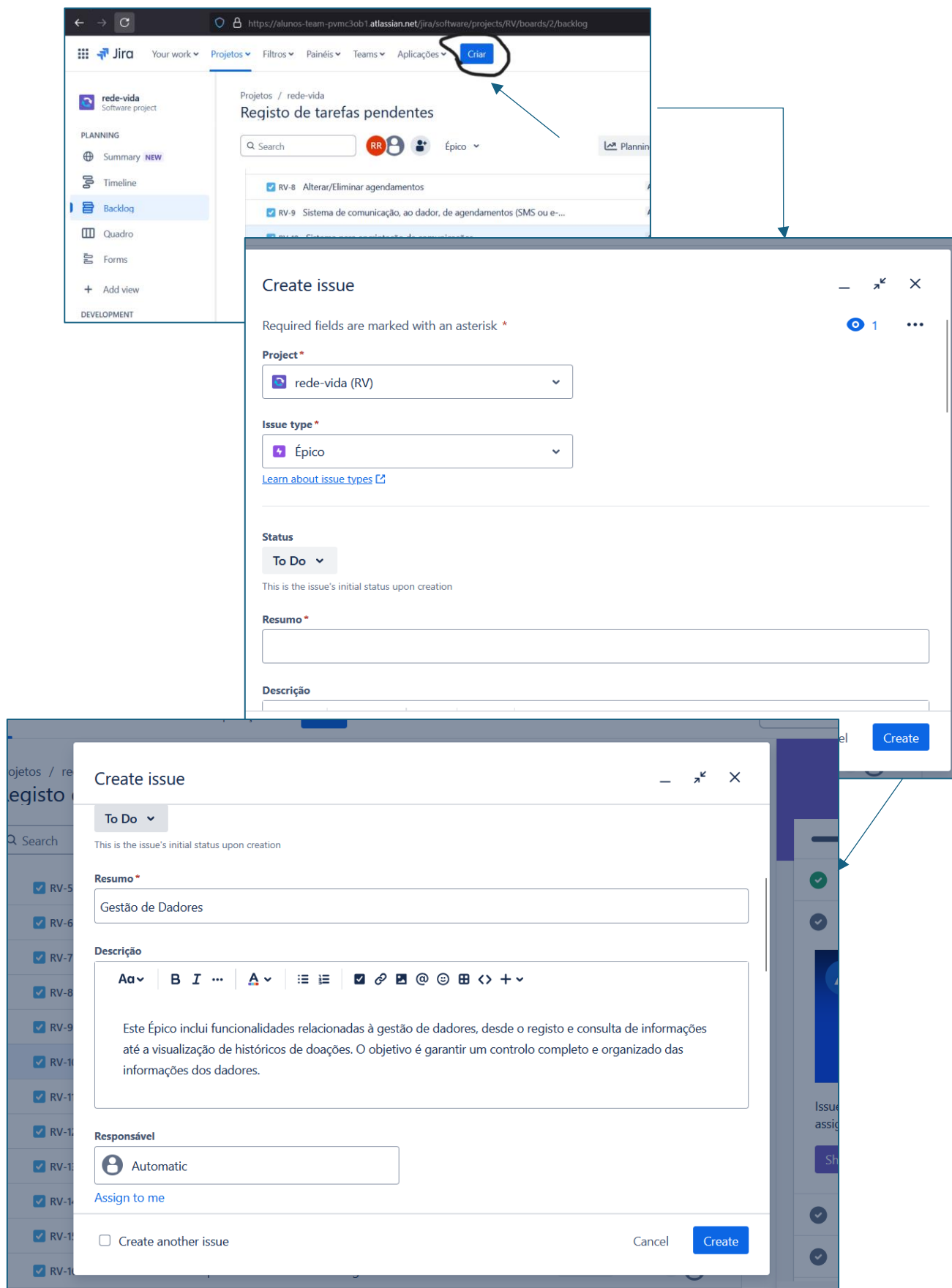
Epics são grandes blocos de trabalho que podem ser divididos em várias tarefas menores.

Detalhando passo-a-passo, temos:

1. Aceder ao projeto no Jira.
<https://alunos-team-pvmc3ob1.atlassian.net/jira/software/projects/RV/boards/2>
2. No menu lateral, seleccionar "Backlog".
3. Clicar em "+ Create issue" e escolher "Epic" como tipo de tarefa.
4. Inserir o nome e descrição do Epic.

5. Salvar e repetir o processo para cada Epic necessário.

As imagens seguintes, mostram estes passos de forma sequencial.



2. Criação de *User Stories*

User Stories representam requisitos ou funcionalidades do ponto de vista do utilizador.

Detalhando passo-a-passo, temos:

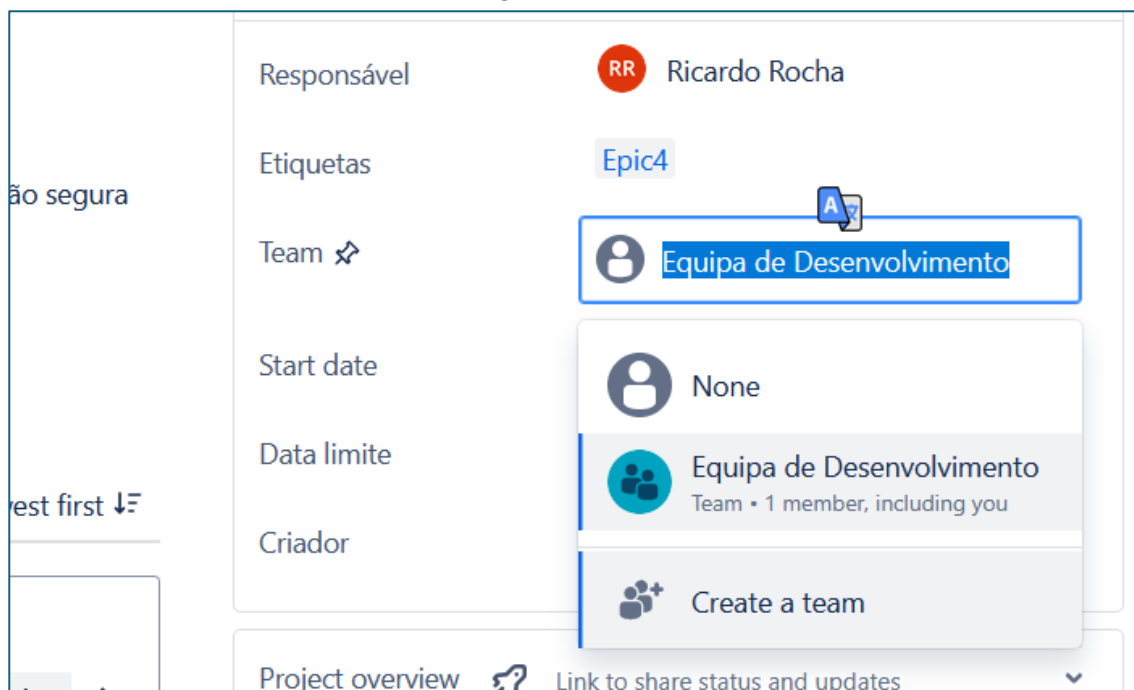
1. No menu "Backlog", clicar em "+ Create issue".
2. Selecionar "Story" como tipo de tarefa.
3. Adicionar um título descritivo para a história.
4. Incluir detalhes no campo de descrição, como objetivos e critérios de aceitação.
5. Relacionar a User Story a um Epic, se aplicável.
6. Salvar.

The image displays two overlapping screenshots of the Jira 'Create issue' dialog. The top screenshot shows the initial form with the following fields: Project (rede-vida (RV)), Issue type (História), Status (To Do), and a summary (Registar novos dadores). The bottom screenshot shows the same form with additional fields: Assign to me, Etiquetas (Theme1.1), Principal (RV-22 Gestão de Dadores), Team (Equipa de Desenvolvimento), and Sprint (Select sprint). The bottom screenshot also includes a 'Create another issue' checkbox and 'Cancel' and 'Create' buttons.

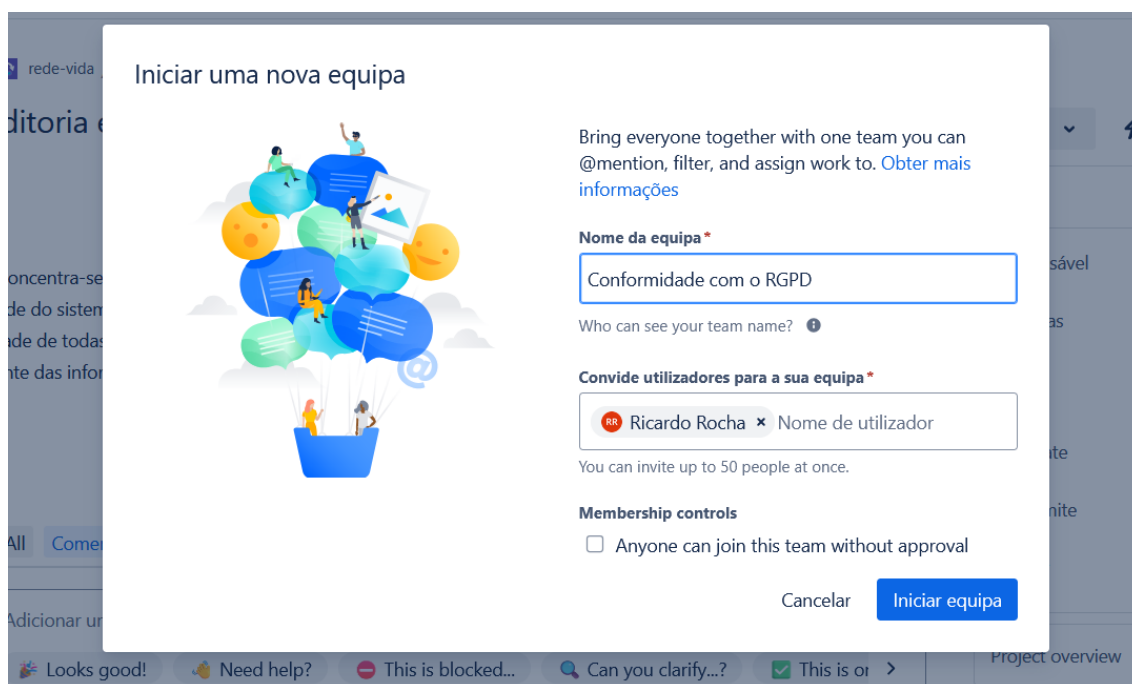
3. Organização de Equipas

Para organizar as equipas e atribuir tarefas, vamos criar equipas de trabalho:

1. Ao criar um issue selecionar a “Team” e “Create a team”





2. Dar um nome à equipa e atribuir membros (foram usadas as contas dos membros do grupo para evitar criar contas para cada elemento das equipas)





3. Adicionar os membros da equipa utilizando os seus endereços de e-mail.
4. Atribuir permissões adequadas a cada membro.

5. Durante a criação de tarefas, definir o "Responsável" para atribuir tarefas específicas a um membro da equipa.

Responsável  Ricardo Rocha


Etiquetas 


Team   Equipa de Desenvolvimento

Start date

Data limite

Criador

Project overview  Link to share status and updates

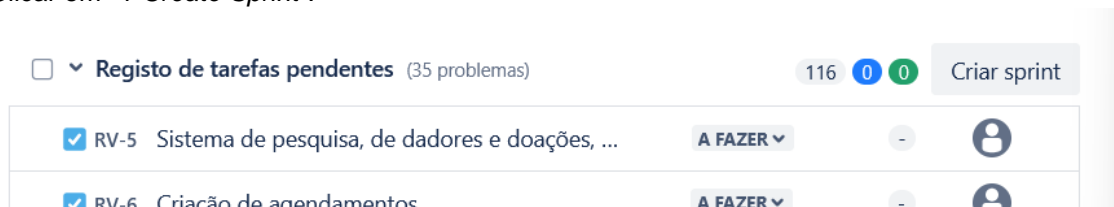
Created há 36 segundos  Configurar

4. Planeamento de Sprints

Sprints são períodos de tempo delimitados para concluir um conjunto de tarefas.

Passo-a-passo:

1. Aceder ao menu "Backlog" do projeto.
2. Clicar em "+ Create Sprint".



3. Arrastar as tarefas do backlog para a secção do sprint.



4. Definir a duração do sprint (geralmente 1-2 semanas).

Edit sprint: RV Sprint 2

Required fields are marked with an asterisk *

Sprint name*

INTEGRAÇÃO DE AGENDAMENTOS

Duration

custom

Start date

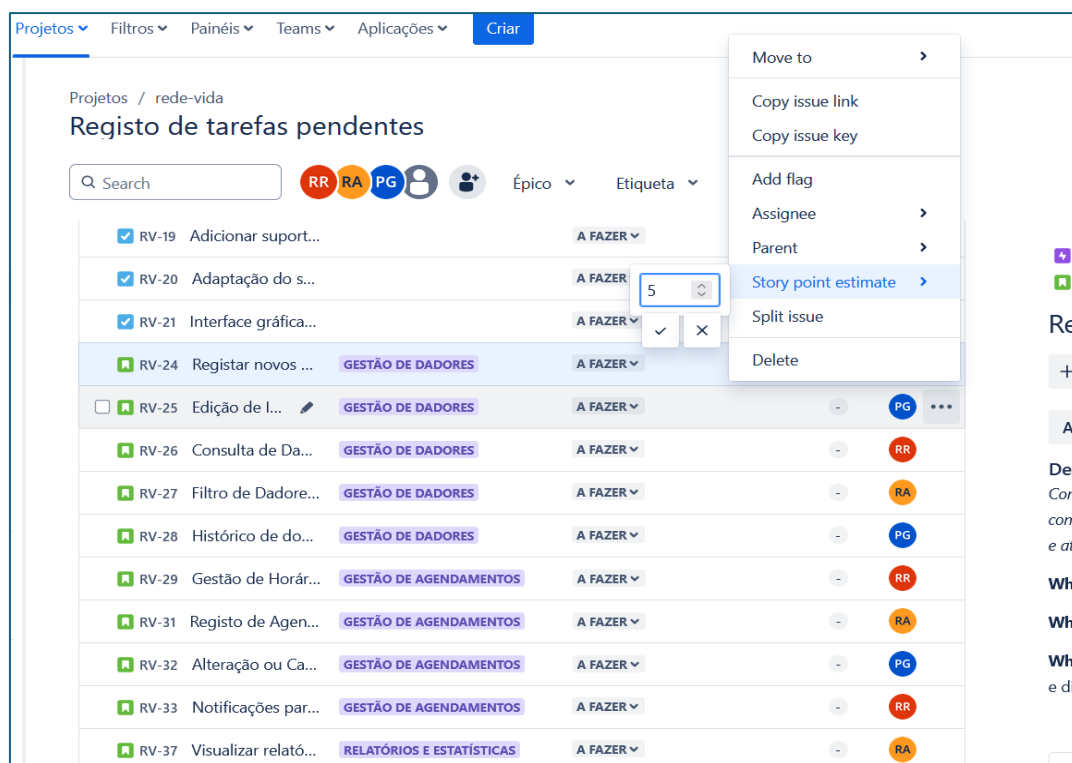
27/01/2025 09:00

End date

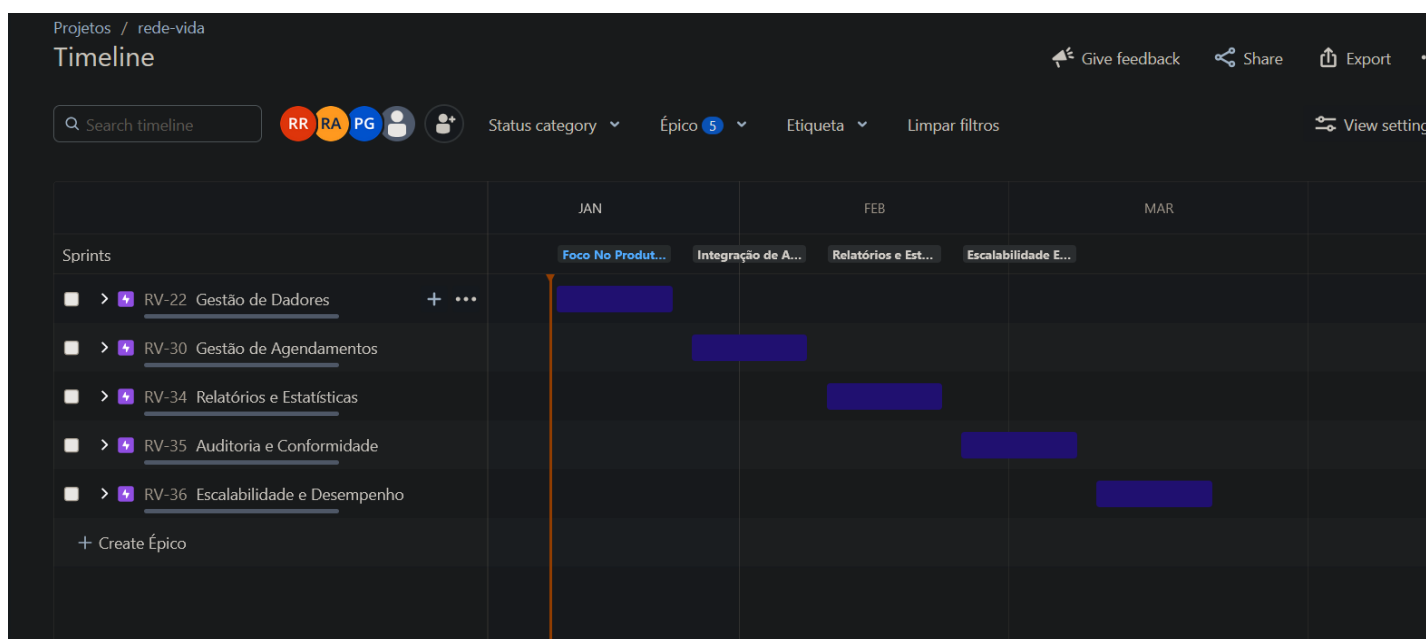
07/02/2025 18:00

Sprint goal

Cancel Update

5. Atribuir *story points* às *user stories*

TIMELINE DO NOSSO PROJETO OBTIDA ATRAVÉS DO JIRA



Podemos então ver o impacto trazido pelo uso do *Jira* ao fornecer organização e visibilidade do projeto para a equipa. A ferramenta facilitou a comunicação entre os membros da equipa, melhorou a eficiência no planeamento e execução das sprints, e deu-nos a capacidade de identificar problemas antes que impactassem os prazos finais.

PROCESSO DE UTILIZAÇÃO DO GIT E GITHUB NO PROJETO

Neste projeto, utilizámos o *Git* como sistema de controlo de versão e o *GitHub* como plataforma para hospedar o repositório remoto. O *Git* permitiu organizar o desenvolvimento de funcionalidades de forma incremental, utilizando *branches* para isolar alterações, enquanto o *GitHub* facilitou a partilha do código e a integração final. Abaixo, descrevo o processo completo desde a criação inicial até à fusão final das alterações.

Porque usamos Git e GitHub?

O *Git* e o *GitHub* são ferramentas fundamentais no desenvolvimento colaborativo de software pelos seguintes motivos:

Histórico de Alterações: O *Git* mantém um registo detalhado de todas as mudanças no código, permitindo reverter alterações problemáticas ou entender o progresso do projeto.

Trabalho em Equipa: O uso de *branches* no *Git* permite que vários programadores trabalhem em diferentes partes do projeto simultaneamente, sem interferirem no trabalho uns dos outros.

Integração Contínua: No *GitHub*, funcionalidades como *Pull Requests* ajudam a rever o código antes da integração no *branch* principal, garantindo maior qualidade.

Backup e Acessibilidade: O *GitHub* funciona como um repositório remoto, armazenando o código na *cloud* e tornando-o acessível de qualquer lugar.

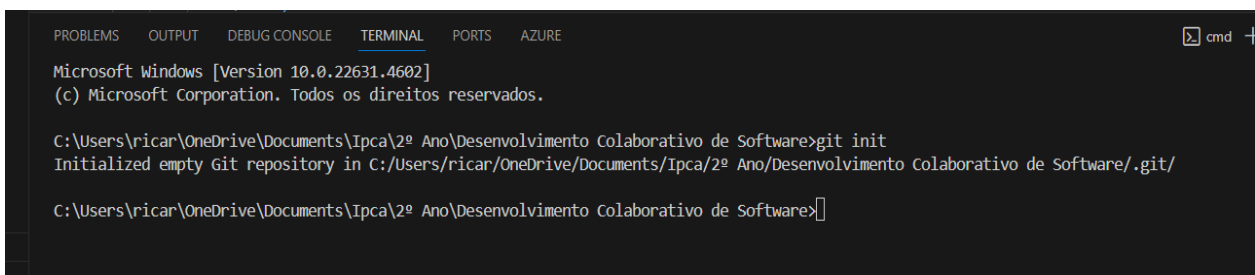
Automação: *GitHub* integra-se com ferramentas de automação, como testes contínuos e *deployment*, para melhorar a eficiência do desenvolvimento.

Etapas do Processo no Projeto

1. INICIALIZAÇÃO DO REPOSITÓRIO

O primeiro passo foi inicializar o repositório localmente e configurar o repositório remoto no GitHub:

- Inicializar o repositório Git local: ***git init***



```
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\ricar\OneDrive\Documents\Ipca\2º Ano\Desenvolvimento Colaborativo de Software>git init
Initialized empty Git repository in C:/Users/ricar/OneDrive/Documents/Ipca/2º Ano/Desenvolvimento Colaborativo de Software/.git/

C:\Users\ricar\OneDrive\Documents\Ipca\2º Ano\Desenvolvimento Colaborativo de Software>
```

- Adicionar todos os ficheiros iniciais: ***git add .***
- Fazer o primeiro commit: ***git commit -m "Primeiro commit: Gestão de Dadores"***

```

C:\Users\ricar\OneDrive\Documents\Ipca\2º Ano\Desenvolvimento Colaborativo de Software>git add .

C:\Users\ricar\OneDrive\Documents\Ipca\2º Ano\Desenvolvimento Colaborativo de Software>git commit -m "Primeiro commit: Gestão de Dadores"
[master (root-commit) 065b2bc] Primeiro commit: Gestão de Dadores
20 files changed, 654 insertions(+)
create mode 100644 05 - Supabase.pdf
create mode 100644 Aula 1DCS.pdf
create mode 100644 ChatGPT - userstories.pdf
create mode 100644 "Estat\303\255stica Descritiva em Excel.pdf"
create mode 100644 FT1_SCRUM - 31out.docx

```

Como o repositório local foi criado manualmente no site do *github* tivemos de usar um comando diferente do normal. Em circunstâncias normais este seria o comando:

git remote add origin https://github.com/Cahves/gestao-dadores.git

Nota: Este comando é utilizado para adicionar um repositório remoto ao repositório local do Git. Ele estabelece uma ligação entre o repositório local e o remoto, permitindo que envie alterações com *push* ou recebas atualizações com *pull*.

Required fields are marked with an asterisk (*).

Owner * Cahves / Repository name * gestão-dadores

✓ Your new repository will be created as **gest-o-dadores**.
The repository name can only contain ASCII letters, digits, and the characters -, ., and _.

Great repository names are short and memorable. Need inspiration? How about **cuddly-octo-tribble** ?

Description (optional)

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:

☐ Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

ⓘ You are creating a public repository in your personal account.

No nosso caso, usamos este comando que altera a url associada ao nosso repositório local

```

C:\Users\ricar\OneDrive\Documents\Ipca\2º Ano\Desenvolvimento Colaborativo de Software\Gestão de dadores>git remote set-url origin https://github.com/cahves/gestao-dadores.git

```

Nota: usamos este comando, uma vez que temos de associar o remoto do GitHub criado ao repositório local.

- Empurrar o *branch* principal (*main*) para o *GitHub*

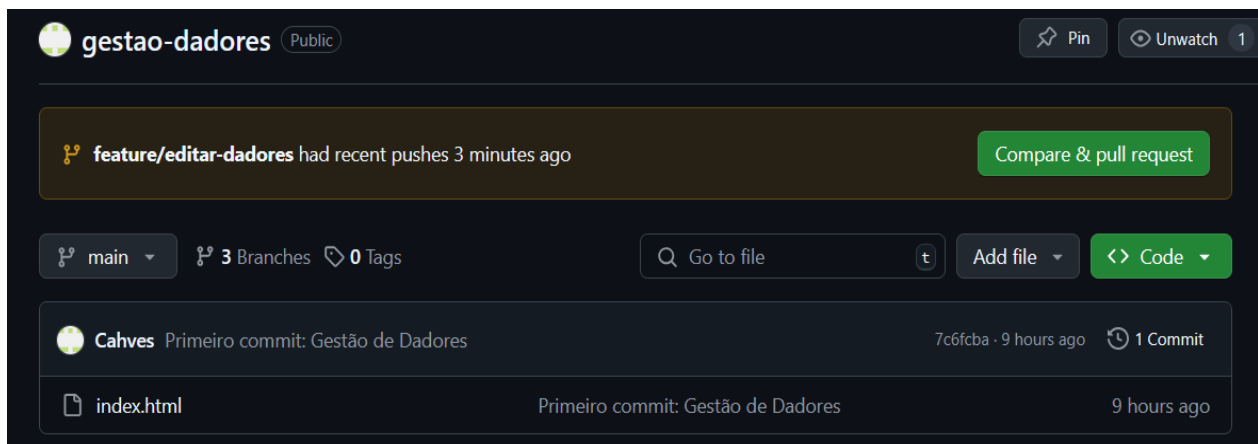
git branch -M main

git push -u origin main

```
C:\Users\ricar\OneDrive\Documents\Ipca\2º Ano\Desenvolvimento Colaborativo de Software\Gestao de dados>git branch -M main

C:\Users\ricar\OneDrive\Documents\Ipca\2º Ano\Desenvolvimento Colaborativo de Software\Gestão de dados>git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 1.13 KiB | 1.13 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
```

E como podemos ver foi criado um *branch*



2. CRIAÇÃO DE *BRANCHES*

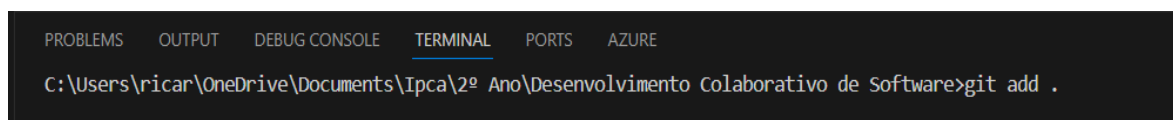
Para cada funcionalidade ou atualização importante, criámos uma *branch* separada. Isso garantiu que cada alteração fosse isolada, reduzindo o risco de conflitos.

- **Exemplo: Criação da Branch para Registo de Dadores:**

git checkout -b feature/registo-dadores

```
C:\Users\ricar\OneDrive\Documents\Ipca\2º Ano\Desenvolvimento Colaborativo de Software\Gestão de dados>git checkout -b feature/registo-dadores
Switched to a new branch 'feature/registo-dadores'
```

git add .



git commit -m "Adicionar funcionalidade de registo de dadores"

```
C:\Users\ricar\OneDrive\Documents\Ipca\2º Ano\Desenvolvimento Colaborativo de Software\Gestão de dados>git commit -m "Adicionar formulário de registo de dados"
[feature/registo-dadores b03a0b1] Adicionar formulário de registo de dados
1 file changed, 41 insertions(+)
create mode 100644 registo.html
```


Por fim, empurrámos a *branch* para o GitHub:

```
C:\Users\ricar\OneDrive\Documents\Ipcal2º Ano\Desenvolvimento Colaborativo de Software\Gestão de dados>git push origin feature/registro-dadores
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 948 bytes | 948.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
```

3. DESENVOLVIMENTO ITERATIVO EM BRANCHES

Outras *branches* foram criadas para funcionalidades específicas:

- **feature/editar-dadores:** Implementámos o *dropdown* para seleccionar dadores e a funcionalidade de edição, incluindo a alteração do estado ativo/inativo.
- **feature/relatorios:** Adicionámos gráficos dinâmicos em relatorios.html para mostrar os dados do *localStorage*.

Cada *branch* seguiu o mesmo fluxo vamos demonstrar:

git checkout -b feature/editar-dadores

```
C:\Users\ricar\OneDrive\Documents\Ipcal2º Ano\Desenvolvimento Colaborativo de Software\Gestão de dados>git checkout -b feature/editar-dadores
Switched to a new branch 'feature/editar-dadores'
```

Após alterações:

git add .

git commit -m "Adicionar formulário para editar informações de dadores"

```
C:\Users\ricar\OneDrive\Documents\Ipcal2º Ano\Desenvolvimento Colaborativo de Software\Gestão de dados>git commit -m "Adicionar formulário para editar informações de dadores"
[feature/editar-dadores 577fe4c] Adicionar formulário para editar informações de dadores
1 file changed, 40 insertions(+)
create mode 100644 editar.html
```

git push origin feature/editar-dadores

```
C:\Users\ricar\OneDrive\Documents\Ipcal2º Ano\Desenvolvimento Colaborativo de Software\Gestão de dados>git push origin feature/editar-dadores
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 996 bytes | 996.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: This repository moved. Please use the new location:
remote: https://github.com/Calves/gestao-dadores.git
remote:
```

E aqui podemos ver o resultado da criação de diferentes branches e o pedido de aprovação.

gestao-dadores Public

feature/editar-dadores had recent pushes 3 minutes ago [Compare & pull request](#)

main 3 Branches 0 Tags

Go to file Add file <> Code

Cahves Primeiro commit: Gestão de Dadores 7c6fcb · 9 hours ago 1 Commit

index.html Primeiro commit: Gestão de Dadores 9 hours ago

Outros *pulls* que também foram realizados e os respetivos branches

feature/index-melhorias had recent pushes 5 minutes ago [Compare & pull request](#)

feature/registo-melhorias had recent pushes 2 minutes ago [Compare & pull request](#)

feature/editar-melhorias had recent pushes 52 seconds ago [Compare & pull request](#)

main 6 Branches 0 Tags

Go to file Add file <> Code

Cahves Adicionar formulário para editar informações de dadores 577fe4c · 36 minutes ago 3 Commits

File	Commit Message	Time
editar.html	Adicionar formulário para editar informações de dadores	36 minutes ago
index.html	Primeiro commit: Gestão de Dadores	9 hours ago
registo.html	Adicionar formulário de registo de dadores	9 hours ago

VISUALIZAÇÃO DO HISTÓRICO

Default

Branch	Updated	Check status	Behind	Ahead	Pull request
main	8 hours ago				<div>Default</div> <div> </div>

Your branches

Branch	Updated	Check status	Behind	Ahead	Pull request
feature/editar-dadores	3 minutes ago		0	2	<div></div> <div> </div>
feature/registo-dadores	8 hours ago		0	1	<div></div> <div> </div>

Active branches

Branch	Updated	Check status	Behind	Ahead	Pull request
feature/editar-dadores	3 minutes ago		0	2	<div></div> <div> </div>
feature/registo-dadores	8 hours ago		0	1	<div></div> <div> </div>

4. Integração de Branches no main

Após desenvolver e testar as funcionalidades, integrámos as branches no branch principal (main).

Mudar para o branch principal:

Nota: Antes de integrar alterações de outro *branch*, devemos estar no *branch* onde queremos que essas alterações sejam fundidas.

git checkout main

```
C:\Users\ricar\OneDrive\Documents\Ipca\2º Ano\Desenvolvimento Colaborativo de Software\Gestão de dados>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

Atualizar o branch principal com o remoto (sincroniza o *branch local main* com a versão mais recente do branch main remoto no GitHub):

git pull origin main

```
C:\Users\ricar\OneDrive\Documents\Ipca\2º Ano\Desenvolvimento Colaborativo de Software\Gestão de dados>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

C:\Users\ricar\OneDrive\Documents\Ipca\2º Ano\Desenvolvimento Colaborativo de Software\Gestão de dados>git pull origin main
From https://github.com/cahves/gestao-dadores
* branch      main       -> FETCH_HEAD
Already up to date.

C:\Users\ricar\OneDrive\Documents\Ipca\2º Ano\Desenvolvimento Colaborativo de Software\Gestão de dados>
```

Fundir a branch de funcionalidade (O Git pega nas alterações que foram feitas na *branch feature/atualizar-páginas* e aplica-as ao *main*.)

git merge feature/atualizar-páginas

```
C:\Users\ricar\OneDrive\Documents\Ipca\2º Ano\Desenvolvimento Colaborativo de Software\Gestão de dados>git merge feature/atualizar-paginas
Updating 338fed7..2cf7de2
Fast-forward
 editar.html      | 114 ++++++++++++++++++++++++++++++++++++++-----
 index.html      |  91 ++++++++++++++++++++++++++++++++++++++-----
 registo.html     |  46 +++++++++++++++++++++++++++++++++-----
 relatorios.html | 141 ++++++++++++++++++++++++++++++++++++++-----
 4 files changed, 344 insertions(+), 48 deletions(-)
```

Empurrar as alterações para o GitHub (após concluirmos as alterações localmente, precisamos de atualizar o repositório remoto para que outros colegas tenham acesso às mudanças).

git push origin main

```
C:\Users\ricar\OneDrive\Documents\Ipca\2º Ano\Desenvolvimento Colaborativo de Software\Gestão de dados>git push origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: This repository moved. Please use the new location:
remote: https://github.com/Cahves/gestao-dadores.git
To https://github.com/cahves/gestao-dadores.git
 338fed7..2cf7de2  main -> main
```

Podemos consultar o repositório do GitHub para analisar com mais detalhe os diferentes *branches* e *updates* feitos em <https://github.com/Cahves/gestao-dadores>.

Os ficheiros são o protótipo das *mockups*, podem ser descarregados e cada *branche* representa uma as diversas iterações que o software passou até chegar ao algo aproximado do objetivo final.

O uso do *Git* e do *GitHub* foi essencial para organizar o desenvolvimento, manter um registro claro das alterações e integrar funcionalidades de forma controlada. O fluxo baseado em *branches* garantiu a separação de tarefas, permitindo adicionar funcionalidades sem comprometer o código existente. Além disso, o *GitHub* facilitou a partilha e revisão do código, contribuindo para a qualidade final do projeto.

CONSIDERAÇÕES FINAIS

O desenvolvimento do projeto "**REDE VIDA**" demonstrou a importância de uma abordagem organizada, com base em metodologias ágeis e ferramentas modernas, para a criação de uma solução robusta e funcional. Através da utilização de *frameworks* como Scrum, aliada ao *Jira* para a gestão de tarefas e ao *Git/GitHub* para o controlo de versões e colaboração, foi possível estruturar o progresso do projeto de forma eficiente e clara.

A divisão em *sprints* e a atribuição de responsabilidades a diferentes membros da equipa foram fatores determinantes para a organização do trabalho. Funcionalidades como o registo e a consulta de dados, a gestão de agendamentos e a geração de relatórios foram desenvolvidas e implementadas com sucesso, alinhadas aos requisitos previamente definidos. Estas entregas asseguraram que o projeto cumprisse os seus objetivos principais, garantindo usabilidade, segurança e escalabilidade.

Além disso, o projeto foi estruturado com atenção à conformidade legal, em especial ao Regulamento Geral de Proteção de Dados (RGPD), através da encriptação de dados sensíveis e da implementação de mecanismos de controlo de acesso. Este foco em segurança reforça a confiança no sistema e no seu potencial para lidar com informações críticas em ambientes reais. Ao longo do desenvolvimento, enfrentámos desafios relacionados à priorização de tarefas e à gestão do tempo, comuns em projetos académicos que simulam o ambiente real de desenvolvimento de software. Estes desafios foram mitigados com revisões regulares do backlog, que permitiram ajustar prioridades e melhorar a comunicação interna da equipa.

A ausência de uma implementação de pipelines de Integração Contínua (CI/CD) foi identificada como uma oportunidade de melhoria. Automatizar testes e deploys em versões futuras do sistema seria uma evolução natural, alinhada às melhores práticas da indústria. Adicionalmente, a integração de feedback de utilizadores finais e a ampliação da documentação técnica também são pontos que podem ser trabalhados em futuras iterações.

Este projeto tem o potencial de ir além do contexto académico, destacando-se como uma solução viável para a gestão eficiente de dados e doações em instituições reais. O impacto social é significativo, uma vez que a plataforma facilita o controlo de informações cruciais e promove a organização de campanhas de doação de sangue de forma segura e intuitiva. Ao continuar a evoluir, este sistema pode tornar-se uma referência em iniciativas semelhantes, promovendo a modernização de processos em instituições de saúde e contribuindo para a melhoria da gestão de recursos humanos e logísticos. A aplicação de novas tecnologias e a exploração contínua de metodologias ágeis fortalecerão ainda mais o projeto, assegurando a sua relevância e eficácia no futuro.

Em suma, este projeto demonstrou que o uso de ferramentas colaborativas, metodologias ágeis e uma visão técnica alinhada às necessidades práticas são cruciais para o desenvolvimento de soluções escaláveis e impactantes. Mais do que um exercício académico, a "Gestão de Dados" representa um exemplo claro de como a tecnologia pode ser utilizada para enfrentar desafios reais, com um impacto social significativo e potencial de aplicação em ambientes profissionais.