

## (contexto) Sequenciador

# SEQUENCER

Fill in the blanks  
marked with post-  
its in your color.

①

E \$ <3

Gestão de Contatos

\$\$\$ <3<3  
Gestão de  
Tarefas

E \$ <3  
Registro de  
Despesas e  
Receitas

②

E \$ <3<3  
Catálogo de  
Bandas e  
Projetos

EE \$\$ <3<3  
Agenda de  
Compromissos

EE \$\$ <3<3  
Catálogo de  
Músicas

# MVP

③

Integração com Plataformas de Streaming

EE \$\$ <3<3<3  
Comunicação  
Interna

E \$ <3<3  
Feedback dos  
Clientes

**INCREMENT**

④

EE \$\$\$ <3  
Integração com  
Redes Sociais

EEE \$\$\$ <3<3<3  
Compartilham  
ento de  
Arquivos

EEE \$\$\$ <3<3<3  
Análise de  
Dados

⑤

Get ready, plan, work and conquer!



## Product backlog

**Link:**

<https://ovscaio.atlassian.net/jira/software/projects/SCRUM/boards/1/backlog?epics=visible&issueParent=10002&selectedIssue=SCRUM-24&atlOrigin=eyJpIjoiMjgwOTg0NDc3NDZhYjg2ZjE2ZTc4Njc5NTk2M2MiLCJwIjoiajI9>

Epic

Items sem epic

Gestão de Contatos

Gestão de Tarefas

Registro de despesas e receitas

Catalogo de bandas

Agenda de compromissos

Catalogo de Musicas

Interação com

+ Criar epic

▼ Sprint 1

Adicionar tasks (12 issues)

0 0 0 Iniciar sprint

SCRUM-13 Criar tela de cadastro de contatos.	GESTÃO DE CONTATOS	TAREFAS PENDENTES	-	
SCRUM-14 Desenvolver opção para categorizar contatos por tipo (músicos, fornecedores, parceiros, etc.).	GESTÃO DE CONTATOS	TAREFAS PENDENTES	-	
SCRUM-15 Implementar funcionalidade de adicionar, editar e excluir contatos.	GESTÃO DE CONTATOS	TAREFAS PENDENTES	-	
SCRUM-16 Integrar sistema de busca para facilitar a localização de contatos.	GESTÃO DE CONTATOS	TAREFAS PENDENTES	-	
SCRUM-18 Integrar sistema de notificação para lembrar membros da banda sobre tarefas pendentes.	GESTÃO DE TAREFAS	TAREFAS PENDENTES	-	
SCRUM-17 Desenvolver opção para definir prioridades e prazos para as tarefas.	GESTÃO DE TAREFAS	TAREFAS PENDENTES	-	
SCRUM-19 Implementar funcionalidade de atribuir tarefas a membros da banda.	GESTÃO DE TAREFAS	TAREFAS PENDENTES	-	
SCRUM-20 Criar interface para criar e visualizar tarefas.	GESTÃO DE TAREFAS	TAREFAS PENDENTES	-	
SCRUM-22 Integrar sistema de geração de relatórios para análise financeira.	REGISTRO DE DESPESA...	TAREFAS PENDENTES	-	
SCRUM-21 Implementar funcionalidade de categorizar despesas e receitas por tipo (aluguel de equipamentos, receitas de shows, ...)	REGISTRO DE DESPESA...	TAREFAS PENDENTES	-	
SCRUM-23 Criar formulário para registro de despesas e receitas.	REGISTRO DE DESPESA...	TAREFAS PENDENTES	-	
SCRUM-24 Desenvolver opção para adicionar comprovantes e documentos relacionados às transações.	REGISTRO DE DESPESA...	TAREFAS PENDENTES	-	

+ Criar item.

**Backlog**

Imprimir trabalho Insights Configurações de visualização

Epic

- Agenda de compromissos
- Catálogo de Músicas
- Integração com plataformas de streaming
- Comunicação interna
- Feedback dos clientes
- Integração com redes sociais
- Compartmentamento de arquivos
- Análise de dados
- + Criar epic

Backlog (30 issues)

ID	Título	Status	Atribuído a
SCRUM-38	Pesquisar e identificar plataformas de streaming populares para integração.	Pending Review	[Avatar]
SCRUM-39	Desenvolver conexão com APIs das plataformas selecionadas.	In Progress	[Avatar]
SCRUM-37	Implementar funcionalidade de compartilhar músicas e playlists diretamente do aplicativo.	Pending Review	[Avatar]
SCRUM-40	Implementar funcionalidade de criar grupos de conversa para membros da banda.	Pending Review	[Avatar]
SCRUM-42	Criar sistema de mensagens ou chat interno.	Pending Review	[Avatar]
SCRUM-41	Desenvolver opção para compartilhar arquivos e links durante as conversas.	Pending Review	[Avatar]
SCRUM-45	Integrar opção para os usuários deixarem comentários e sugestões.	Pending Review	[Avatar]
SCRUM-44	Implementar sistema de feedback para os usuários.	Pending Review	[Avatar]
SCRUM-43	Desenvolver formulário de avaliação de satisfação.	Pending Review	[Avatar]
SCRUM-46	Implementar funcionalidade de conectar e autenticar contas de redes sociais dentro do aplicativo.	Pending Review	[Avatar]
SCRUM-47	Desenvolver opção para compartilhar eventos, músicas e atualizações nas redes sociais.	Pending Review	[Avatar]
SCRUM-48	Pesquisar e identificar redes sociais relevantes para integração.	Pending Review	[Avatar]
SCRUM-50	Desenvolver opção para controlar permissões de acesso aos arquivos.	Pending Review	[Avatar]
SCRUM-49	Criar sistema de armazenamento de arquivos.	Pending Review	[Avatar]
SCRUM-51	Implementar funcionalidade de compartilhamento de arquivos entre membros da banda e colaboradores.	Pending Review	[Avatar]
SCRUM-52	Implementar sistema de coleta e armazenamento de dados financeiros.	Pending Review	[Avatar]
SCRUM-53	Desenvolver algoritmos para análise de dados e geração de relatórios.	Pending Review	[Avatar]
SCRUM-54	Integrar ferramentas de visualização de dados para apresentação de relatórios financeiros.	Pending Review	[Avatar]
SCRUM-36	Integrar sistema de busca para facilitar a descoberta de novos bandos.	Pending Review	[Avatar]

Epic			
> Agenda de compromissos	<input checked="" type="checkbox"/> SCRUM-48 Pesquisar e identificar redes sociais relevantes para integração.		
	<input checked="" type="checkbox"/> SCRUM-50 Desenvolver opção para controlar permissões de acesso aos arquivos.		
	<input checked="" type="checkbox"/> SCRUM-49 Criar sistema de armazenamento de arquivos.		
> Catálogo de Músicas	<input checked="" type="checkbox"/> SCRUM-51 Implementar funcionalidade de compartilhamento de arquivos entre membros da banda e colaboradores.		
	<input checked="" type="checkbox"/> SCRUM-52 Implementar sistema de coleta e armazenamento de dados financeiros.		
> Integração com plataformas de streaming	<input checked="" type="checkbox"/> SCRUM-53 Desenvolver algoritmos para análise de dados e geração de relatórios.		
	<input checked="" type="checkbox"/> SCRUM-54 Integrar ferramentas de visualização de dados para apresentação de relatórios financeiros.		
> Comunicação interna	<input checked="" type="checkbox"/> SCRUM-26 Integrar sistema de busca para facilitar a descoberta de novas bandas.		
	<input checked="" type="checkbox"/> SCRUM-25 Criar banco de dados para armazenar informações das bandas.		
> Feedback dos clientes	<input checked="" type="checkbox"/> SCRUM-28 Implementar funcionalidade de adicionar, editar e excluir informações de bandas.		
	<input checked="" type="checkbox"/> SCRUM-27 Desenvolver opção para categorizar bandas por gênero musical, localização, etc.		
> Integração com redes sociais	<input checked="" type="checkbox"/> SCRUM-29 Integrar calendário para visualização em diferentes dispositivos.		
	<input checked="" type="checkbox"/> SCRUM-32 Implementar funcionalidade de adicionar, editar e excluir compromissos.		
> Compartilhamento de arquivos	<input checked="" type="checkbox"/> SCRUM-30 Criar interface para visualização da agenda.		
	<input checked="" type="checkbox"/> SCRUM-31 Desenvolver opção para enviar convites e notificações para membros da banda e outros envolvidos.		
> Análise de dados	<input checked="" type="checkbox"/> SCRUM-34 Desenvolver opção para categorizar músicas por gênero, álbum, etc.		
	<input checked="" type="checkbox"/> SCRUM-33 Criar banco de dados para armazenar informações de músicas.		
+ Criar epic	<input checked="" type="checkbox"/> SCRUM-35 Implementar funcionalidade de adicionar, editar e excluir informações de músicas.		
	<input checked="" type="checkbox"/> SCRUM-36 Integrar sistema de reprodução para ouvir amostras de músicas.		

**Definition of Ready (DoR):**

- Cada item do backlog deve ter uma descrição clara e concisa que explique a funcionalidade desejada e seu propósito dentro do contexto do aplicativo.
- Deve incluir critérios de aceitação claros e mensuráveis que indiquem quando a funcionalidade estiver completa e atender às expectativas dos usuários, incluindo casos de uso específicos e resultados esperados.
- Deve ser estimado o esforço necessário para implementar a funcionalidade, levando em consideração a complexidade técnica, a integração com outras partes do sistema e o tempo necessário para o desenvolvimento e testes.
- Cada item do backlog deve estar claramente priorizado em relação a outros, com base no valor para o usuário, na dependência de outras funcionalidades e no impacto geral no produto.
- Todas as dependências externas, como integrações com APIs de terceiros ou recursos de infraestrutura necessários, devem ser identificadas e documentadas para garantir que não haja impedimentos durante a implementação.
- Deve ser verificado se há recursos disponíveis, como equipe qualificada, ferramentas adequadas e acesso aos ambientes de desenvolvimento necessários, para implementar a funcionalidade dentro do prazo desejado.
- Todos os riscos potenciais associados à implementação da funcionalidade devem ser identificados e documentados, com planos de contingência definidos, se necessário.

**Definition of Done (DoD) para o Aplicativo:**

- Todo o código relacionado à funcionalidade deve ser escrito de acordo com os padrões de codificação estabelecidos pela equipe e revisado por pares para garantir qualidade e consistência.
- Deve haver testes unitários e de integração escritos e executados para garantir que o código funcione conforme o esperado e que não haja regressões em outras partes do sistema.
- A interface de usuário correspondente à funcionalidade deve ser implementada de acordo com o design e as diretrizes de usabilidade estabelecidas.
- Qualquer documentação relacionada à funcionalidade, como manuais de usuário ou descrições de API, deve ser atualizada para refletir as alterações implementadas.
- Os critérios de aceitação da funcionalidade devem ser testados e verificados para garantir que a funcionalidade atenda às expectativas do usuário.
- A funcionalidade deve ser revisada e aprovada pelo Product Owner antes de ser considerada concluída e pronta para ser entregue ao usuário final.

# Detalhamento da sprint 1:

## Épico Gestão de contatos:

1. Criar tela de cadastro de contatos.
2. Implementar funcionalidade de adicionar, editar e excluir contatos.
3. Integrar sistema de busca para facilitar a localização de contatos.
4. Desenvolver opção para categorizar contatos por tipo (músicos, fornecedores, parceiros, etc.).

**História do Usuário:** Como usuário do aplicativo, desejo poder gerenciar meus contatos para manter um registro organizado e acessível de músicos, fornecedores e parceiros relacionados à minha banda.

### Critérios de Aceitação:

1. **Dado que** estou na tela principal do aplicativo, **Quando** eu seleciono a opção "Contatos", **Então** devo ser redirecionado para a tela de gerenciamento de contatos.
2. **Dado que** estou na tela de gerenciamento de contatos, **Quando** eu seleciono a opção "Adicionar Contato", **Então** devo ser direcionado para um formulário onde posso inserir as informações do novo contato.
3. **Dado que** estou na tela de adicionar contato, **Quando** preencho todas as informações obrigatórias do formulário e seleciono "Salvar", **Então** o contato deve ser adicionado à lista de contatos com sucesso.
4. **Dado que** estou na tela de gerenciamento de contatos, **Quando** seleciono um contato existente, **Então** devo ver opções para editar ou excluir o contato.
5. **Dado que** estou visualizando a lista de contatos, **Quando** utilizo o sistema de busca, **Então** devo poder filtrar os contatos por nome, tipo ou outras informações relevantes.
6. **Dado que** estou na tela de adicionar/editar contato, **Quando** seleciono uma categoria para o contato (músico, fornecedor, parceiro, etc.), **Então** o contato deve ser categorizado corretamente para facilitar a organização e busca.

## Épico Gestão de tarefas:

1. Criar interface para criar e visualizar tarefas.
2. Implementar funcionalidade de atribuir tarefas a membros da banda.
3. Desenvolver opção para definir prioridades e prazos para as tarefas.
4. Integrar sistema de notificação para lembrar membros da banda sobre tarefas pendentes.

**História do Usuário:** Como um membro da banda, desejo poder gerenciar minhas tarefas para acompanhar o progresso das atividades e garantir que nada seja esquecido.

### Critérios de Aceitação:

1. **Dado que** estou na tela principal do aplicativo, **Quando** eu selecionar a opção "Gestão de Tarefas", **Então** devo ser direcionado para uma página onde posso visualizar todas as tarefas cadastradas.
2. **Dado que** estou na página de gestão de tarefas, **Quando** eu selecionar a opção "Adicionar Tarefa", **Então** devo ser redirecionado para um formulário onde posso inserir os detalhes da nova tarefa.

3. **Dado que** estou na página de gestão de tarefas, **Quando** eu selecionar uma tarefa existente, **Então** devo ver opções para editar, excluir ou marcar a tarefa como concluída.
4. **Dado que** estou visualizando a lista de tarefas, **Quando** eu selecionar uma tarefa, **Então** devo ver os detalhes da tarefa, incluindo responsável, prioridade e prazo.
5. **Dado que** estou atribuindo uma tarefa a um membro da banda, **Quando** eu selecionar um membro da banda na lista de contatos, **Então** a tarefa deve ser atribuída ao membro selecionado e aparecer em sua lista de tarefas.

## Épico Registro de despesas e receitas:

1. Criar formulário para registro de despesas e receitas.
2. Implementar funcionalidade de categorizar despesas e receitas por tipo (aluguel de equipamentos, receitas de shows, etc.).
3. Desenvolver opção para adicionar comprovantes e documentos relacionados às transações.
4. Integrar sistema de geração de relatórios para análise financeira.

**História do Usuário:** Como o responsável financeiro da banda, desejo poder registrar despesas e receitas para manter um controle preciso das finanças.

### Critérios de Aceitação:

1. **Dado que** estou na tela principal do aplicativo, **Quando** eu selecionar a opção "Registro de Despesas e Receitas", **Então** devo ser direcionado para uma página onde posso visualizar todas as transações financeiras registradas.
2. **Dado que** estou na página de registro de despesas e receitas, **Quando** eu preencher os campos obrigatórios do formulário, como tipo de transação, valor e descrição, **Então** devo poder salvar a transação com sucesso.
3. **Dado que** estou adicionando uma nova transação, **Quando** eu categorizar a transação como despesa ou receita, **Então** a transação deve ser devidamente classificada e aparecer na lista correspondente.
4. **Dado que** estou visualizando a lista de transações financeiras, **Quando** eu selecionar uma transação, **Então** devo ver os detalhes da transação, incluindo data, tipo e valor.

# (Extra) – Timeline -MVP + Incrementos

