

# **DOCUMENTO DE PROCESSO, TECNOLOGIAS E FERRAMENTAS**

## **Equipe:**

**508161 – João Paulo Freitas Queiroz**

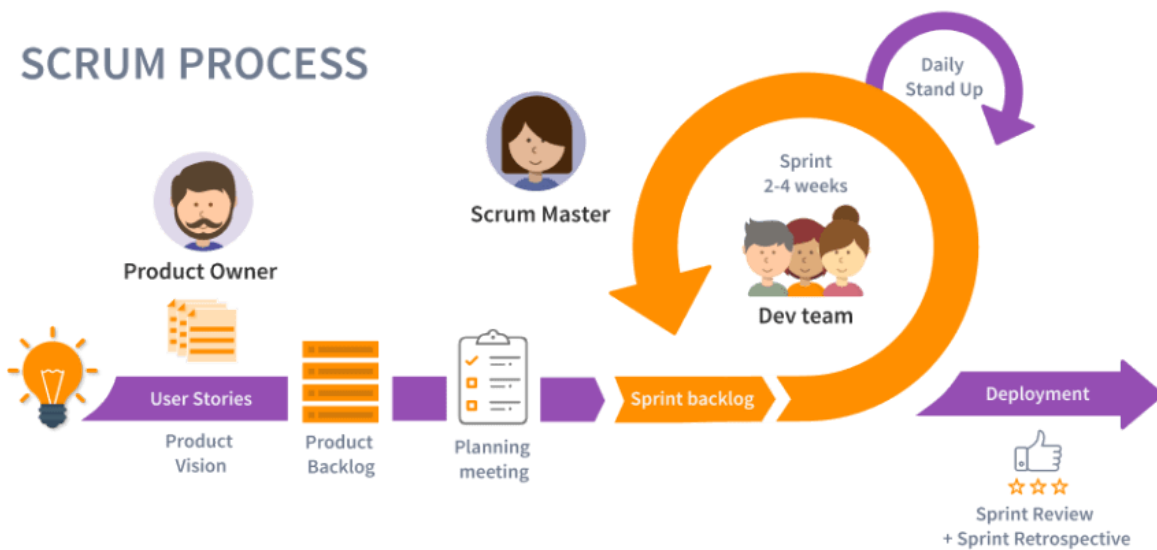
**511790 – Matheus Leandro de Melo Silva**

**473143 - João Lucas de Oliveira Lima**

# PROCESSOS

## Scrum

O Scrum é um framework de gerenciamento de projetos ágil que pode ser aplicado em diferentes campos, dentre eles, o desenvolvimento de softwares. Ideal para gerenciar projetos complexos, tornando-os mais palpáveis a partir de divisões menores e funcionais (incrementos).



## Vantagens

### 1. Maior flexibilidade e adaptabilidade ao projeto e grupo de trabalho:

Por ser uma metodologia de gerenciamento de projetos ágil, o Scrum permite que as equipes respondam rapidamente às mudanças e estejam mais prontas para lidar com os desafios inesperados que podem surgir ao longo do projeto.

### 2. Maior transparência e visibilidade:

A abordagem fornece uma visão clara do progresso de cada uma das etapas do projeto – facilitando imensamente não apenas a continuidade do trabalho, como a própria tomada de decisão pelos gestores de projeto.

### 3. Entrega mais rápida:

Projeto segmentado em ciclos curtos, torna mais palpável a realização das entregas o quanto antes possível, sem deixar de entregar um produto com qualidade, por exemplo. Desta maneira, os prazos são menores, abrindo portas até para uma maior lucratividade, visto que o alcance é aumentado – pensando em negócios, no geral.

#### 4. Qualidade aprimorada:

A melhoria contínua, como sendo um dos principais pilares agregados do Scrum, permite com que os processos que envolvam o projeto, sejam constantemente melhorados – resultando em produtos e serviços de maior qualidade.

#### 5. Uso de recursos mais eficiente:

O método Scrum se direciona ao conceito de "just-in-time". Isto significa que todo o processo de compras, de produção ou de atividades relacionadas, é voltado apenas para aquele momento – sem criar grandes fluxos de estoque ou trabalho desnecessário.

### Etapas que seguimos

1. **User Stories:** Com a elicitação já feita e validada com possíveis usuários, o time se juntou para a criação do User Stories, onde pegamos cada requisito e seguimos o padrão *"Como <posição do usuário>, eu quero <objetivo>, para que <benefício>."* para assim converter os requisitos em User Stories. Onde, mais tarde, foram definidos com um nível de prioridade para começar o desenvolvimento.
2. **Product Backlog:** O product backlog está sendo mantido no Trello da matéria, e está nos servindo para acompanhar o progresso do trabalho, vendo o quanto faltava de user stories e até mesmo o nível de prioridade, que foi definido após o time de desenvolvimento se juntar e realizar um planning poker.
3. **Sprint:** As nossas sprints duram 2 semanas, no começo cada desenvolvedor recebia um User Storie com o maior valor de importância para ser desenvolvido durante a sprint. Porém, com as dificuldades encontradas, nós passamos a dividir mais atividades para os integrantes do grupo e começamos a priorizar telas, invés de user stories, por conta das exigências de um de nossos stakeholders. Contudo, ainda mantemos o backlog das User Stories e vamos movendo para a área concluída à medida que terminamos as telas, para manter um acompanhamento das funcionalidades.
4. **Daily Meeting:** Todos os dias durante a sprint - na maioria das vezes às 10 da manhã - o time de desenvolvimento se encontra para falar sobre seu progresso no desenvolvimento das telas ou componentes que foram feitas, os problemas que foram encontrados e os testes finalizados. Além disso, o time ajuda caso um dos integrantes do time tenha alguma dificuldade com a parte que lhe foi dada.
5. **Sprint Review:** Ao fim de duas semanas, normalmente na sexta-feira, a equipe se encontra para debater sobre os acontecimentos durante a sprint, as principais dificuldades, se for o caso o que ainda falta ser terminado e se acha que é possível continuar com a mesma quantidade de telas para cada um. Normalmente, a gente decide novas telas, porém com o tempo está escasso, focamos em priorizar a qualidade das telas principais, desde os teste dos composables até a padronização do projeto e escolha de melhores métodos para o desenvolvimento e desempenho da aplicação.

## Conclusão

O Scrum está nos ajudando principalmente no acompanhamento do desempenho do time de desenvolvimento durante o projeto, porque por meio das reuniões a gente pode determinar o que está dando certo e o que precisa ser alterado. Além disso, está servindo como um meio mais fácil e otimizado para a gerência de tempo gasto com as principais entregas e telas do projeto. Contudo, por conta dos imprevistos, além do daily meeting, estamos nos comunicando por meio de mensagens assíncronas, tanto pelo Whatsapp quanto pelo Discord.

## Kanban

O quadro Kanban é uma ferramenta para visualização do fluxo de trabalho, projetada para trazer clareza ao processo de trabalho e aumentar a eficiência limitando o trabalho em andamento.



## Vantagens

### 1. Visibilidade do trabalho em andamento:

Fornecer uma visão clara do que está sendo trabalhado atualmente, permitindo que todos os membros da equipe saibam quais tarefas estão em andamento e quem está responsável por elas. Isso ajuda a evitar a sobrecarga de trabalho e a distribuir as tarefas de forma equitativa.

### 2. Priorização de tarefas:

É possível priorizar tarefas com facilidade. As tarefas são frequentemente representadas como cartões ou notas adesivas que podem ser movidas para diferentes colunas ou categorias, indicando seu status e prioridade.

### 3. Previsibilidade:

Ao monitorar o fluxo de trabalho e os tempos médios de conclusão das tarefas, a equipe pode ganhar uma maior previsibilidade em relação aos prazos e estimativas.

## Como usamos Kanban

1 Fase: Inicialmente tínhamos apenas um quadro Kanban, para o acompanhamento do desenvolvimento da aplicação, envolvendo tanto a entrega de documentos como de telas ou funcionalidades de nossa aplicação. Este quadro pode ser encontrado aqui: <https://trello.com/b/vYIRAKKE>.

2. Fase: Vimos que também nos foi necessário manter um acompanhamento mais específico e separado de todos os documentos criados, assim como as respectivas datas da entrega, para isso criamos um outro quadro Kanban, onde tínhamos acesso a todas as necessidades voltadas para os documentos. Este quadro pode ser encontrado aqui: <https://trello.com/b/2VaejWkK>.

3. Fase: Com o surgimento da necessidade de criação de documentos parecidos para uma matéria distinta, resolvemos criar um novo quadro, uma vez que por mais que os documentos fossem os mesmos, as matérias queriam versões diferentes do projeto. Este quadro pode ser encontrado aqui: <https://trello.com/b/MGXZaVau/kanban>.

## Conclusão

Por fim, por mais que o Kanban seja uma ferramenta que nos ajuda a manter o acompanhamento de todo o processo de desenvolvimento, desde a programação até a documentação, percebemos que é difícil manter atualizando todos os links e dados de diversos quadros, o que acabou sendo nossa maior dificuldade. Por fim, também não usamos mais o novo quadro da fase 3, uma vez que a matéria específica dele não passou

mais atividades envolvendo o trabalho de projeto integrado. Além disso, por mais que tenha facilitado a manutenção dos quadros de Kanban atuais, ainda tem sido difícil manter tracking das datas e dos requisitos de cada entrega, mesmo fazendo sempre o acompanhamento e anotações.

# TECNOLOGIAS

## Front-End

### Kotlin

Kotlin é uma linguagem de programação estaticamente tipada e orientada a objetos que é interoperável com a máquina virtual Java (JVM), bibliotecas de classes Java e Android.



### Vantagens

#### **1. Suporte Oficial para Android:**

O Kotlin é a linguagem oficialmente suportada pelo Android desde 2017, substituindo o Java como linguagem nativa.

#### **2. 100% Interoperável com o Java:**

O Kotlin foi projetado para funcionar perfeitamente com o Java. Permitindo utilizar todo o ecossistema Java existente, bibliotecas e frameworks.

#### **3. Conciso e Legível:**

O Kotlin oferece uma sintaxe mais concisa em comparação com o Java, o que resulta em código mais limpo e fácil de ler.

#### **4. Comunidade Ativa:**

O Kotlin tem uma comunidade robusta e ativa de desenvolvedores, tornando mais simples a busca por recursos, tutoriais e demais suportes.

#### **Conclusão**

O Kotlin foi escolhido como a linguagem front-end devido a vantagens essenciais. Sendo a linguagem oficialmente suportada pelo Android, oferece integração total com a plataforma e interoperabilidade com o Java. Sua sintaxe concisa e legível resulta em código mais limpo, e a comunidade ativa de desenvolvedores fornece suporte valioso. Em resumo, o Kotlin se destaca por sua compatibilidade, clareza e apoio da comunidade, tornando-o ideal para o desenvolvimento de software mobile.

### **Android Jetpack Compose**

O Android Jetpack Compose é uma framework de IU para o desenvolvimento de aplicativos Android. Ele foi desenvolvido pelo Google e foi projetado para simplificar e acelerar a criação de interfaces de usuário nativas para aplicativos Android.



#### **Vantagens**

##### **1. Declaração Declarativa de UI:**

O Jetpack Compose utiliza uma abordagem declarativa para criar a interface do usuário, utilizando uma descrição de como a interface do usuário deve ser exibida, ao invés de uma série de manipulações de visualizações. Tornando o código mais claro e legível.

##### **2. Integração Nativa:**

Compose é uma parte nativa do ecossistema Android oferecendo integração com outros componentes do Jetpack, como Navigation, LiveData e ViewModel. Facilitando a criação de aplicativos robustos e bem estruturados.

##### **3. Reatividade:**

Partes da interface do usuário podem ser atualizadas automaticamente quando os dados subjacentes mudam, tornando o desenvolvimento de interfaces de usuário dinâmicas mais fácil de implementar.

#### **4. Compatibilidade com Kotlin:**

O Jetpack Compose é totalmente compatível com o Kotlin. Permitindo o aproveitamento de recursos avançados do Kotlin, com foco no desenvolvimento de aplicativos nativos mais eficientes.

#### **5. Ferramentas de Desenvolvimento:**

O Android Studio, a IDE oficial do Android, oferece suporte total ao Jetpack Compose, incluindo recursos de autocompletar, depuração e visualização de pré-visualização instantânea, ajudando na criação e testes das interfaces de usuário.

### Conclusão

O Android Jetpack Compose foi escolhido para auxiliar na construção de telas de interface em um software mobile em Kotlin devido à sua abordagem declarativa simplificada, integração nativa com componentes do Jetpack, capacidade de reatividade, compatibilidade com Kotlin e suporte total do Android Studio. Esses benefícios combinados tornam o Compose uma escolha eficaz para o desenvolvimento eficiente e intuitivo de interfaces de usuário.

## Back-End

### Firestore

Firestore, Inc. é um conjunto de serviços de computação em nuvem de back-end e plataformas de desenvolvimento de aplicativos fornecidos pelo Google. Ele hospeda bancos de dados, serviços, autenticação e integração para uma variedade de aplicativos, incluindo Android, iOS, JavaScript e dentre outros.



# Firestore

### Vantagens



### **1. Facilidade de Uso:**

Fornecer uma interface simples e uma documentação abrangente, possuindo uma curva de aprendizado relativamente pequena.

### **2. Banco de Dados em Tempo Real:**

O Firebase Realtime Database é uma opção de banco de dados NoSQL em tempo real que permite sincronização instantânea de dados entre clientes e servidor.

### **3. Autenticação:**

Firebase oferece serviços de autenticação robustos, incluindo autenticação por email/senha, autenticação social (como login via Google ou Facebook), autenticação por número de telefone e muito mais.

### **4. Armazenamento de Arquivos:**

O Firebase Storage permite armazenar e gerenciar arquivos, como imagens, vídeos e documentos.

### **5. Analytics e Monitoramento de Desempenho:**

Firebase fornece ferramentas de análise que ajudam a entender o comportamento dos usuários no aplicativo. Permitindo o acompanhamento de eventos, métricas e relatórios em tempo real. Além disso, o Firebase Performance Monitoring ajuda a identificar problemas de desempenho no aplicativo.

### **6. Hosting e Funções na Nuvem:**

Firebase Hosting fornece uma maneira simples de hospedagem de aplicação WEB. Além disso, as Funções na Nuvem do Firebase permitem executar código em resposta a eventos, como acionadores de banco de dados, autenticação, entre outros.

### **7. Ampla Comunidade e Suporte:**

Firebase tem uma comunidade ativa de desenvolvedores e uma equipe de suporte do Google por trás dela. Permitindo o acesso mais facilmente há: documentação, fóruns ou recursos de suporte pagos.

### **8. Gratuito para Começar:**

O Firebase oferece um nível gratuito para começar a construir e testar aplicações.

## Conclusão

O Firebase é a escolha preferencial para o back-end e banco de dados devido à sua facilidade de uso, documentação abrangente e integração simplificada. Oferece um banco de dados em tempo real para sincronização instantânea de dados, além de serviços robustos de autenticação, armazenamento de arquivos e análise de desempenho. A ampla comunidade de desenvolvedores e o suporte do Google garantem recursos prontamente disponíveis, enquanto o nível gratuito facilita o início do desenvolvimento sem custos significativos. Resumindo, o Firebase é uma opção completa e eficaz para o desenvolvimento ágil, eficiente e repleto de recursos.

# Testes unitários

## JUnit 5

JUnit é uma API de código aberto para a criação de testes unitários em Java, bem como outras linguagens compatíveis com a JVM.



## Vantagens

### 1. Testes Automatizados:

Permite escrever testes unitários automatizados para verificação do comportamento de componentes específicos da aplicação.

### 2. Identificação Precoce de Problemas:

Os testes unitários podem ajudar a identificar erros e problemas de lógica de programação mais cedo no ciclo de desenvolvimento, o que torna mais fácil e econômico corrigi-los. Isso evita que bugs se acumulem e se tornem mais difíceis de resolver.

### 3. Integração Contínua:

O JUnit pode ser integrado em pipelines de integração contínua (CI) para garantir que os testes sejam executados automaticamente sempre que for enviado uma alteração de código.

## Conclusão

O JUnit foi escolhido como o framework para testes unitários devido à sua capacidade de automatizar verificações de comportamento, identificar problemas precocemente para evitar acumulação de bugs e integrar-se de forma contínua em pipelines de integração. Resumidamente, o JUnit é essencial para assegurar a qualidade e confiabilidade do software, tornando o desenvolvimento mais eficiente e confiável.

# FERRAMENTAS

## Versionamento de código

### Git

É um sistema de controle de revisão distribuído, rápido, escalável e com um conjunto de comandos incomumente rico que oferece operações de alto nível e acesso completo aos seus recursos.



### Vantagens

#### 1. Controle de Versão:

O Git permite controlar todas as mudanças feitas no código-fonte ao longo do tempo, permitindo acompanhar as alterações feitas em arquivos, quem fez essas alterações e quando elas foram feitas.

#### 2. Ramificação (Branching) e Mesclagem (Merging):

Permite criar ramificações (branches) para desenvolver recursos ou correções de bugs separadamente do código principal (branch principal ou master). Você pode mesclar essas ramificações de volta ao código principal quando estiverem prontas.

#### 3. Rastreamento de Alterações:

Permite voltar a versões anteriores do seu código (reverter) ou verificar o histórico de alterações para entender como o código evoluiu ao longo do tempo.

#### 4. Histórico de Alterações:

Cada alteração é registrada por meio de um commit, que inclui um ou mais comentários explicando a mudança efetuada. Isso ajuda a entender por que uma alteração foi feita e a rastrear problemas ou regressões.

### **5. Colaboração Eficiente:**

Permite que vários desenvolvedores possam trabalhar simultaneamente no mesmo projeto sem conflitos significativos. O Git facilita a fusão de diferentes ramificações do código.

### Conclusão

O Git foi escolhido como a ferramenta de versionamento de código devido às suas funcionalidades abrangentes. Ele oferece controle de versão preciso, facilita branches e oferecendo opções de branch e rebase eficientes, fornece rastreamento detalhado de alterações com commits explicativos e promove a colaboração eficiente entre desenvolvedores. Resumidamente, o Git é a escolha superior para o versionamento de código no desenvolvimento de software, proporcionando controle e colaboração de alta qualidade.

## **Plataforma de hospedagem de código-fonte**

### GitHub

GitHub é uma plataforma e serviço baseado em nuvem para desenvolvimento de software e controle de versão usando Git, permitindo aos desenvolvedores armazenar e gerenciar seu código.



### Vantagens

#### **1. Controle de Versão:**

Controle de Versão Baseado no Git: O GitHub é construído sobre o Git, aproveitando todas as vantagens do controle de versão distribuído, como rastreamento de alterações, ramificação e mesclagem eficientes.

## **2. Hospedagem de Repositórios:**

O GitHub fornece hospedagem de repositórios de código gratuitos e pagos, permitindo o armazenamento e compartilhamento de código.

## **3. Gestão de Versões de Documentação:**

Permite o armazenamento de documentação.

## **4. Integração com Ferramentas de Terceiros:**

Possui integrações com uma ampla variedade de ferramentas de terceiros, como Slack, Trello e JIRA, facilitando a incorporação dessas ferramentas ao fluxo de trabalho de desenvolvimento.

## Conclusão

O GitHub foi escolhido para hospedagem de código-fonte devido à sua base no Git, que proporciona controle de versão eficiente, criação de branches e merge/rebase. Além disso, oferece uma plataforma confiável e flexível para hospedar repositórios de código, gerenciar versões de documentação e integrar-se ao Trello, aprimorando o fluxo de trabalho da equipe. Em resumo, o GitHub é a escolha ideal devido à sua robustez e capacidade de melhorar o desenvolvimento de software.

# **Controle e gestão do projeto**

## Trello

É uma ferramenta visual que possibilita o gerenciamento de projetos, fluxo de trabalho e monitoramento de tarefas.



## Vantagens

### **1. Visualização Clara do Fluxo de Trabalho:**

O Trello utiliza quadros, listas e cartões para representar seu fluxo de trabalho. Fornecendo uma visualização clara de todas as tarefas em andamento.

### **2. Colaboração em Equipe:**

O Trello facilita a colaboração entre membros da equipe. Podendo atribuir tarefas a membros da equipe, adicionar comentários, anexar arquivos e mencionar colegas de equipe para manter todos atualizados.

### **3. Priorização de Tarefas:**

Você pode usar etiquetas, datas de vencimento e prioridades para categorizar e priorizar tarefas. Isso ajuda a garantir que as tarefas mais importantes sejam tratadas primeiro.

### **4. Integrações:**

O Trello se integra facilmente a várias outras ferramentas populares, como Slack, Google Drive, GitHub e muitas outras. Permitindo uma conexão ao restante do seu conjunto de ferramentas de desenvolvimento.

### **5. Acompanhamento do Progresso:**

Oferece uma maneira fácil de rastrear o progresso das tarefas. Você pode mover cartões pelas listas à medida que são concluídos, o que fornece uma visão instantânea do andamento do projeto.

### **6. Flexibilidade para Metodologias Ágeis:**

O Trello é adequado para várias metodologias ágeis, como Scrum e Kanban. Permitindo criar quadros personalizados para atender às necessidades do projeto, seja ele um projeto ágil ou mais tradicional.

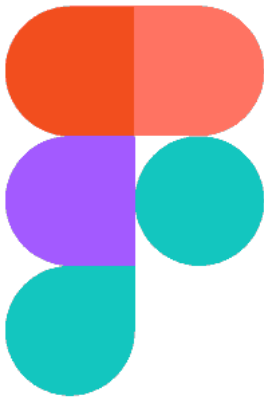
## Conclusão

O Trello foi escolhido para o gerenciamento do processo Scrum devido à sua visualização clara do fluxo de trabalho, capacidade de colaboração em equipe e integração com outras ferramentas populares. Resumidamente, o Trello proporciona um fluxo de trabalho eficiente e uma visão instantânea do progresso do projeto no desenvolvimento de software.

## Ferramenta de prototipagem

### Figma

Figma é um editor gráfico de vetor e prototipagem de projetos de design baseado principalmente no navegador web, com ferramentas offline adicionais para aplicações desktop para GNU/Linux, macOS e Windows.



# Figma

### Vantagens

#### **1. Design Colaborativo em Tempo Real:**

O Figma permite que múltiplos designers e membros da equipe colaborem em um projeto de design ao mesmo tempo, em tempo real.

#### **2. Prototipagem Interativa:**

Você pode criar protótipos interativos diretamente no Figma, permitindo que a equipe teste a usabilidade do design e valide ideias antes de começar a programar.

#### **3. Design Responsivo:**

O Figma permite criar designs responsivos para diferentes tamanhos de tela e dispositivos, ajudando a garantir que o software funcione bem em todas as plataformas.

#### **4. Versionamento:**

O Figma mantém um histórico de versões de cada projeto, o que é útil para rastrear mudanças ao longo do tempo e reverter para versões anteriores, se necessário.

## Conclusão

O Figma se destacou como a ferramenta de prototipagem ideal devido ao design colaborativo em tempo real, criação de protótipos interativos, suporte a designs responsivos e registro de versionamento. Resumidamente, o Figma aprimora o desenvolvimento de software ao facilitar a colaboração, teste e criação de designs responsivos.

## Ambiente de desenvolvimento

### Android Studio

O Android Studio é o ambiente de desenvolvimento integrado (IDE) oficial para o desenvolvimento de apps Android. Com base no editor de código e nas ferramentas para desenvolvedores avançados do IntelliJ IDEA.



### Vantagens

#### **1. Total Integração com o Android:**

O Android Studio é projetado especificamente para desenvolvimento Android, oferecendo suporte total para todas as versões do Android, componentes e bibliotecas. Ele é atualizado regularmente para manter-se compatível com as últimas versões do Android.

#### **2. Emuladores Avançados:**

O Android Studio inclui emuladores de dispositivos Android poderosos e configuráveis, permitindo que você teste seus aplicativos em uma variedade de dispositivos virtuais com diferentes versões do Android e tamanhos de tela.

#### **3. Integração com o Kotlin:**



O Kotlin é a linguagem de programação recomendada para o desenvolvimento Android, e o Android Studio oferece suporte total ao Kotlin, tornando mais fácil e eficiente escrever código Android.

#### **4. Gerenciamento de Dependências Simplificado:**

O Android Studio integra o Gradle como seu sistema de gerenciamento de build, permitindo que você gerencie facilmente as dependências do projeto e as bibliotecas externas.

#### **5. Suporte à Publicação:**

O Android Studio oferece um fluxo de trabalho completo para empacotar e publicar seu aplicativo na Google Play Store, simplificando todo o processo de distribuição.

### Conclusão

O Android Studio é a escolha ideal como IDE para desenvolvimento Android devido à sua integração total com a plataforma, atualizações regulares, emuladores avançados, suporte ao Kotlin, gerenciamento de dependências e fluxo de trabalho completo para publicação. Em resumo, o Android Studio oferece um ambiente robusto e eficiente para o desenvolvimento de software Android.

## Comunicação entre o time

### Discord

É uma plataforma de comunicação online projetada para permitir que pessoas se comuniquem por meio de bate-papo via texto, voz e vídeo.



# Discord

### Vantagens

#### **1. Comunicação em tempo real**

Oferece comunicação via chat em tempo real, permitindo a criação de canais específicos para diferentes aspectos do desenvolvimento de software.

#### **2. Chamadas de voz e vídeo e compartilhamento de arquivos:**

Oferece recursos de chamadas de voz e vídeo de alta qualidade além de permitir o compartilhamento de arquivos.

### **3. Bots:**

Capacidade de criar bots para automação de tarefas comuns, como notificações de eventos.

## Conclusão

O Discord é a escolha ideal como ferramenta de comunicação principal devido à comunicação em tempo real, criação de canais específicos, chamadas de voz/vídeo de alta qualidade e a capacidade de criar bots para automação de tarefas comuns. Resumidamente, o Discord facilita a colaboração e comunicação eficaz no desenvolvimento de software.

## WhatsApp

É um aplicativo de mensagens instantâneas e serviço de comunicação multiplataforma, que permite com que os usuários troquem mensagens de texto, façam chamadas de voz e vídeo, compartilhem fotos, vídeos, documentos e façam chamadas de voz e vídeo em tempo real.



# WhatsApp

## Vantagens

### **1. Comunicação em tempo real**

Oferece comunicação via chat em tempo real, permitindo a criação de canais específicos para diferentes aspectos do desenvolvimento de software

### **2. Facilidade de Acesso:**

A maioria das pessoas está familiarizada com o WhatsApp e o utiliza regularmente em seus dispositivos móveis.

### **3. Grupos de Chat:**

Permite a criação de grupos de chat específicos. Ajudando a manter conversas organizadas e direcionadas para os tópicos relevantes.

## Conclusão

O WhatsApp foi escolhido como ferramenta de comunicação alternativa em relação ao Discord, devido à sua comunicação em tempo real, capacidade de criar canais específicos, facilidade de acesso à informação e ampla familiaridade. Esses recursos permitiram uma colaboração ágil, rápida adoção pela equipe e organização eficiente das conversas.