

Ambientes padronizados de Desenvolvimento e Monitoramento de Aplicações baseado na cultura DevOps

Chapecó - SC, 2015



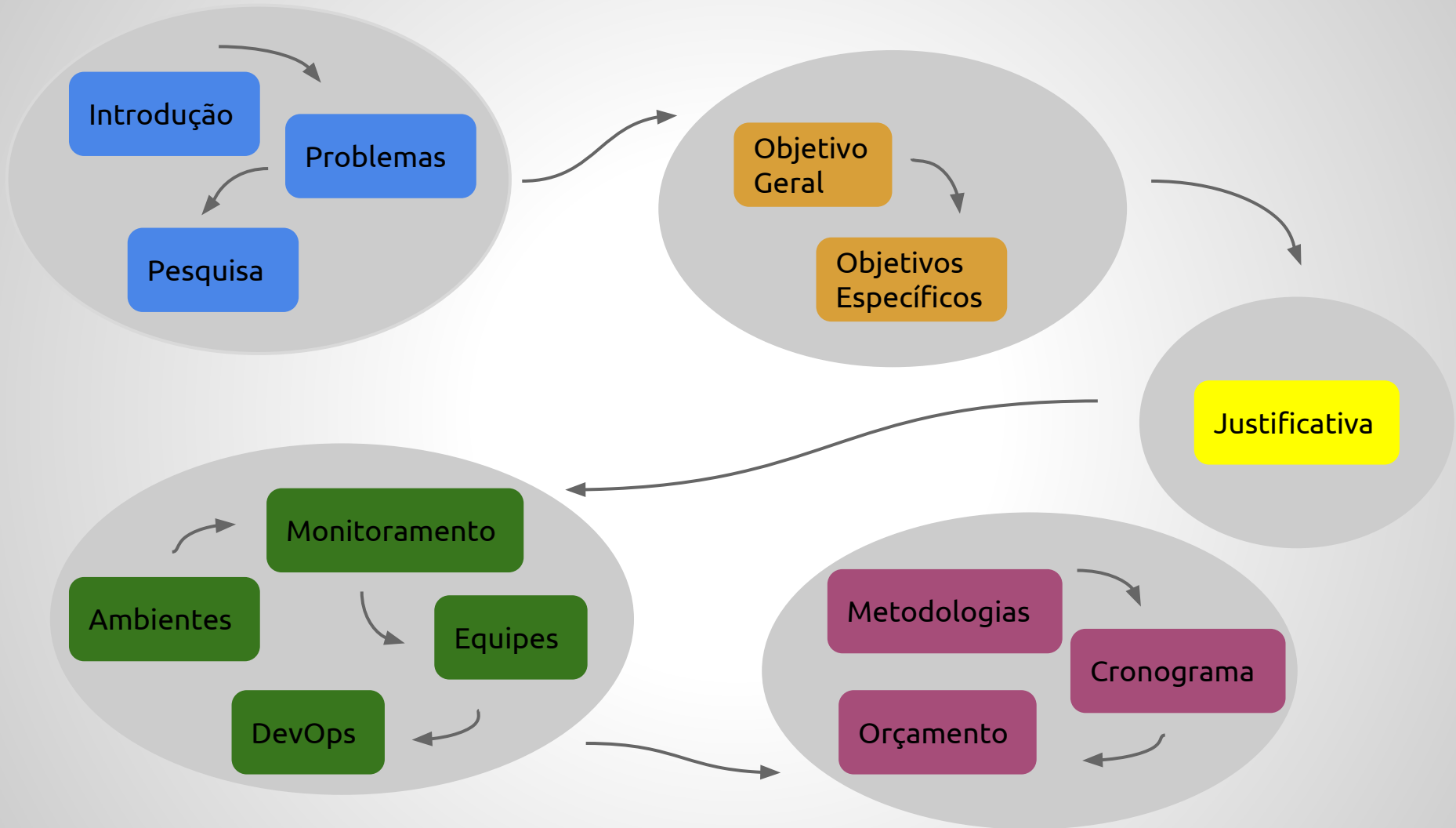
**UNIVERSIDADE COMUNITÁRIA DA REGIÃO DE CHAPECÓ
(UNOCHAPECÓ)**

**Área de Ciências Exatas e Ambientais
Ciência da Computação**

Acadêmico: Alefe Variani

Orientador: Cezar Júnior de Souza

Chapecó - SC, 2015



Introdução

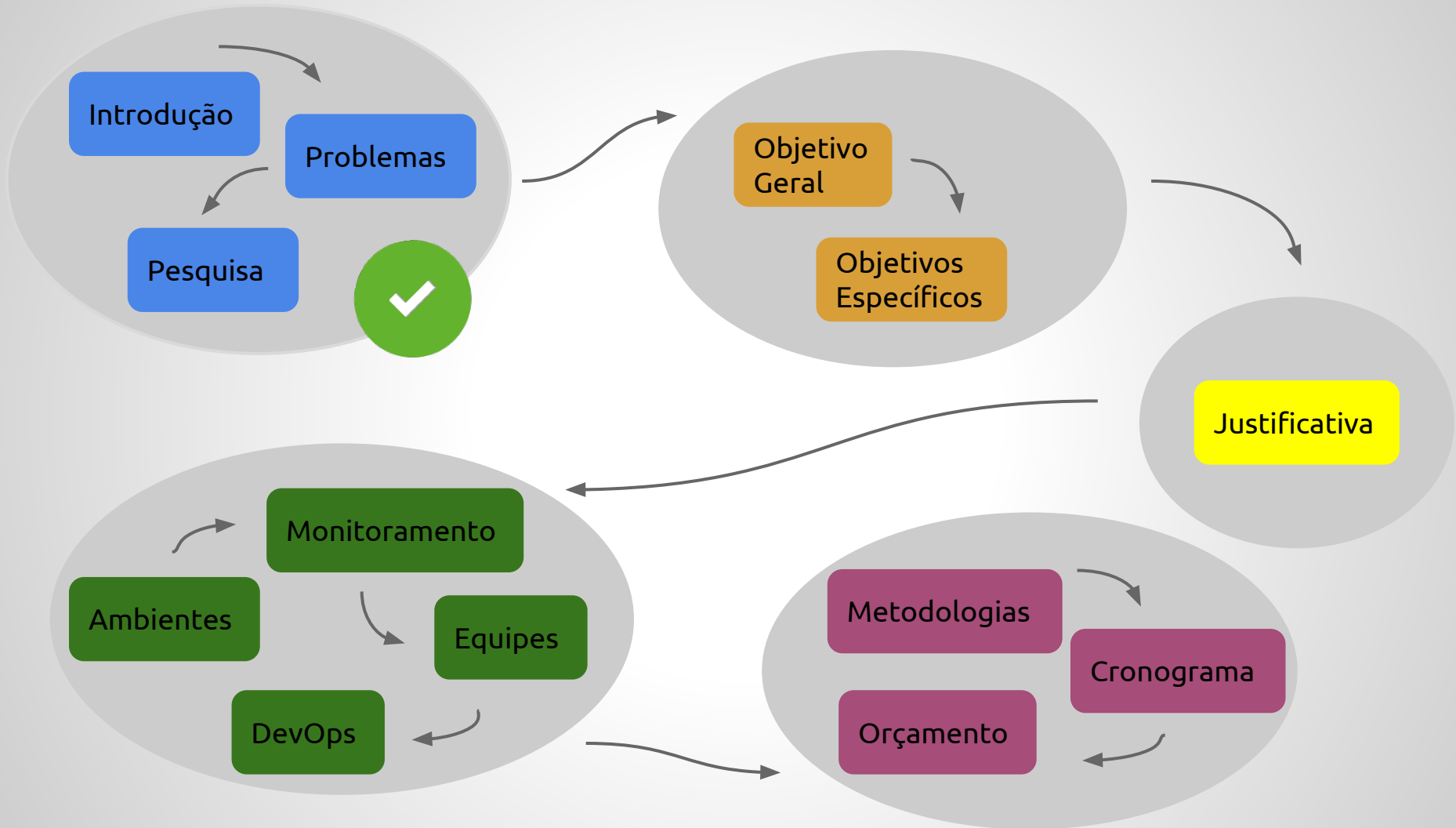
- Avanço tecnológico e importância do *software* para as pessoas;
- Surgimento dos Métodos Ágeis;
- Divisão das responsabilidades;

Problemas

- Muitas demandas para um curto espaço de tempo;
- Processos dos setores de TI com conceitos de sistemas estáticos;
- Área de desenvolvimento e infraestrutura trabalhando separadamente;
- Ambiente de Desenvolvimento diferentes do Ambiente de Produção;
- Mal monitoramento da Aplicação;

Hipóteses e Questões de Pesquisa

- Por que existe diferenças entre ambientes de desenvolvimento e produção?
- Por que o desenvolvimento e infraestrutura trabalham tão distantes?
- É possível utilizar a cultura **DevOps** para melhorar os processos de desenvolvimento de *software* no **CRS** ?
- Implantar a cultura **DevOps** no **CRS**, traz resultados rapidamente?
- Que ganhos e benefícios trará?



Objetivos

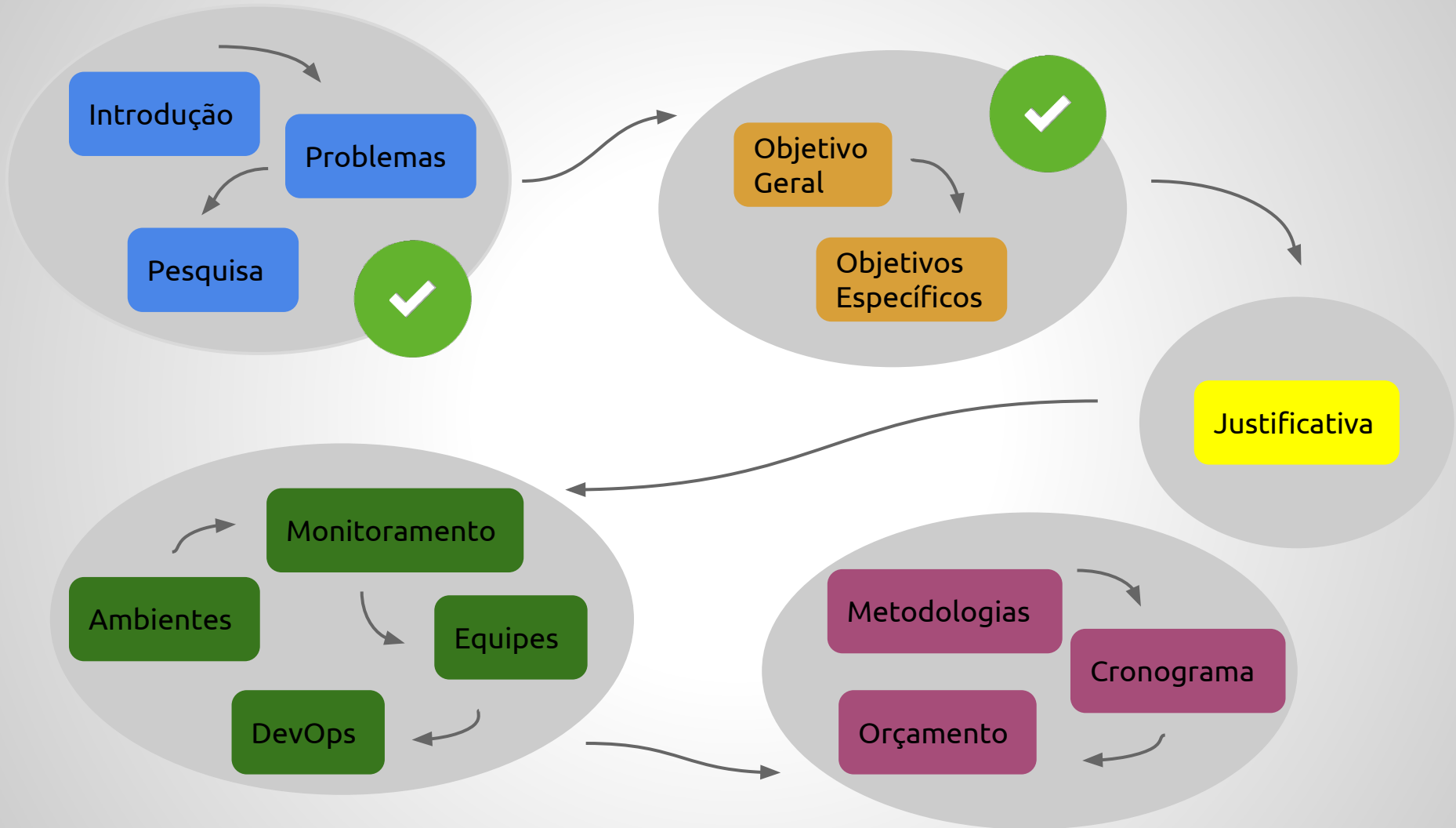
Objetivo Geral

Fazer um levantamento sobre tecnologias e práticas que utilizam os conceitos do DevOps, analisar às melhorias que o DevOps pode trazer para o ambiente de desenvolvimento de *software* e aplicá-la em uma situação real, para verificar se a abordagem realmente traz benefícios aos envolvidos no processo de desenvolvimento de *software*.

Objetivos

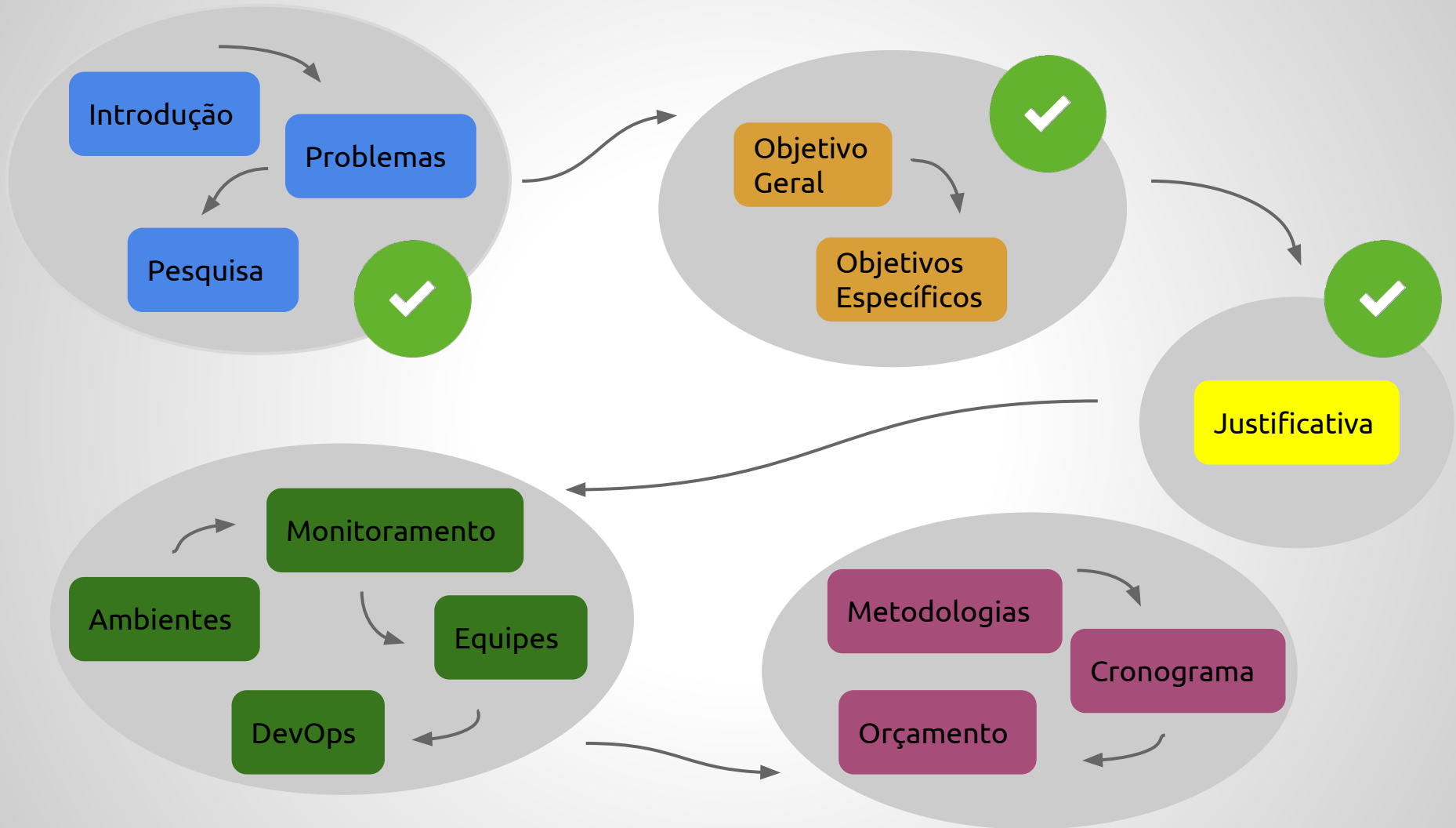
Objetivos Específicos

- Conhecer os processos do desenvolvimento de *software*;
- Entender conceitos da metodologia que se aplicam ao ambiente DevOps;
- Melhorar o ambiente de desenvolvimento e monitoramento da aplicação através do uso da cultura DevOps;
- Aplicar uma proposta em um ambiente desenvolvimento real, utilizando conceitos e práticas do DevOps;
- Analisar o comportamento, vantagens e desvantagens que a proposta traz ao ambiente de desenvolvimento;



Justificativa

- Entregar *software* em produção cada vez mais difícil;
- Métodos ágeis de desenvolvimento não alcançaram todo valor da aplicação;
- Medos de mudanças;
- Liberações de versões (*deploy*) arriscadas;
- “Funciona na minha máquina”;
- Surgimento de Silos;



Revisão Bibliográfica

Ambiente de Desenvolvimento

Ambiente de Operações

Ambientes de desenvolvimento do *Software*

Ambiente de Produção

Ambientes de Desenvolvimento

Necessário para a equipe de desenvolvimento desenvolver o *software*.

Envolve um coleção de ferramentas:

- Sistema Operacional;
- Linguagem de Programação;
- IDE (ambiente de desenvolvimento integrado);
- Controle de Versão;

Ambientes de Produção

Processo de execução de produto, onde os usuários terão acesso ao *software* desenvolvimento.

Mantêm sempre um grau de complexidade alto:

- Gerenciamento de dependências;
- Arquivos de configurações;
- Versões diferentes;

Ambientes de Operações

Necessário para a equipe de infraestrutura manter o *software* funcionando.

Processos envolvidos:

- Instalação/Configuração de servidores;
- *Build* e *Deploy* da aplicação;

Monitoramento da Aplicação

Monitoramento da Aplicação

É observar, analisar e manter o acompanhamento de como a aplicação está se comportando.

Pontos que o monitoramento deve ganhar atenção:

- Notificações;
- Agregação de logs;
- Métricas;
- Visualizações;
- Informações em tempo de execução;
- Disponibilidade;

Monitoramento da Aplicação

Segundo um pesquisa da Aberdeen:

- As empresas podem antecipar em 53% problemas nas aplicações, antes de receber uma reclamação;
- Melhoria de 48% no tempo de correções nas falhas de desempenho;
- Redução de 15% no número total de reclamações dos usuários;

Equipes de Desenvolvimento

Equipes de Operações

Equipes

Problemas

Equipes de Desenvolvimento

Responsável por desenvolver novos produtos, funcionalidades e dar manutenção para possíveis problemas que ocorram no *software*.

Dentre as principais responsabilidades são:

- *Design do Software*;
- Protótipo;
- Programação;
- Testes;
- Validação;

Equipes de Operações

Responsável por manter os sistemas funcionando. Monitorando a performance, avaliando e propondo melhorias.

Principais responsabilidades:

- Gestão de ambiente de teste;
- Gestão de error e incidentes;
- *FeedBack* Contínuo;

Problemas entre as Equipes

Alguns fatores que geram estes conflitos entre as equipes

- Surgimento de metodologias de desenvolvimento ágil para equipes de Desenvolvimento;
- Demora em fazer *Deploys* para a produção;
- Falta de *FeedBack* aos desenvolvedores sobre suas implementações;
- Ambiente dos desenvolvedores diferente do ambiente de produção;
- Equipe de operações com culturas arcaicas de administração;

Introdução

Como surgiu

DevOps

Conceitos

Ferramentas

DevOps



Introdução

O movimento/cultura DevOps foca em aperfeiçoamento da comunicação, colaboração e integração entre desenvolvedores de *software* e administradores da infraestrutura de TI.

DevOps



Como surgiu

- Por volta de 2008 começa a utilizar o termo infraestrutura ágil;
- Em 2009 várias metodologias e pensamentos surgiram;
- O termo DevOps foi criado durante a conferência *Velocity*, foi apresentaram o trabalho “10+ Deploys Per Day: Dev and Ops Cooperation at Flickr”;

DevOps



Como surgiu

- No final de 2009 na Bélgica aconteceu o primeiro encontro chamado DevOpsDay, criado por Patrick Debois;
- Voltado inicialmente pra *startups*, depois passou a ser aplicado em ambientes corporativos;

DevOps

Conceitos

DevOps se mantêm em quatro pilares principais, conhecidos pelas siglas C.A.M.S:

- Cultura;
- Automação;
- Medição/Avaliação;
- Compartilhamento;

DevOps

Ferramentas

- Gerenciamento do Ambiente de Desenvolvimento;
- Gerenciamento das Configurações da Aplicação;



DevOps

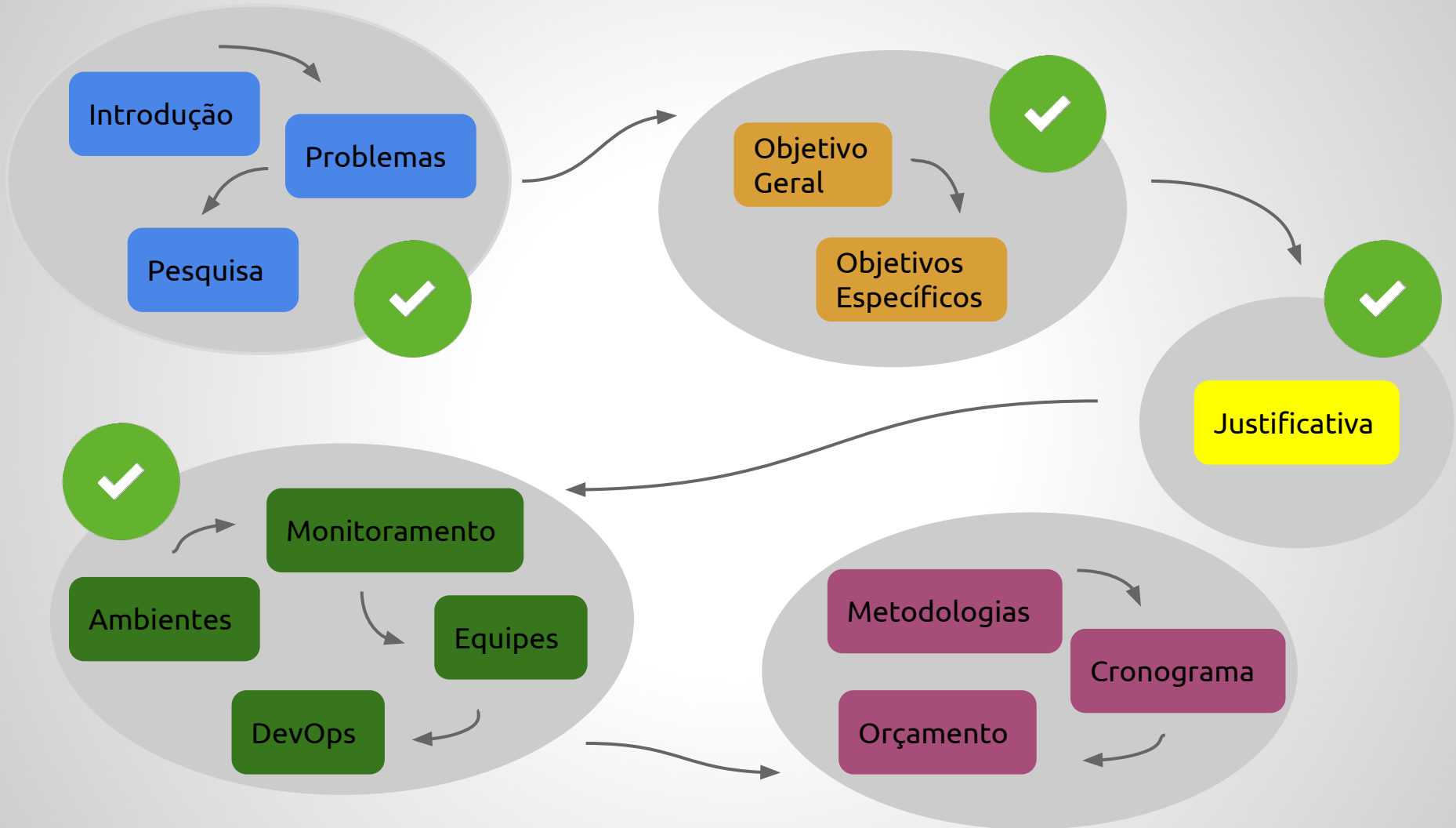
Ferramentas

- Gerenciamento de Configurações;



- Monitoramento da Aplicação;





Procedimentos Metodológicos

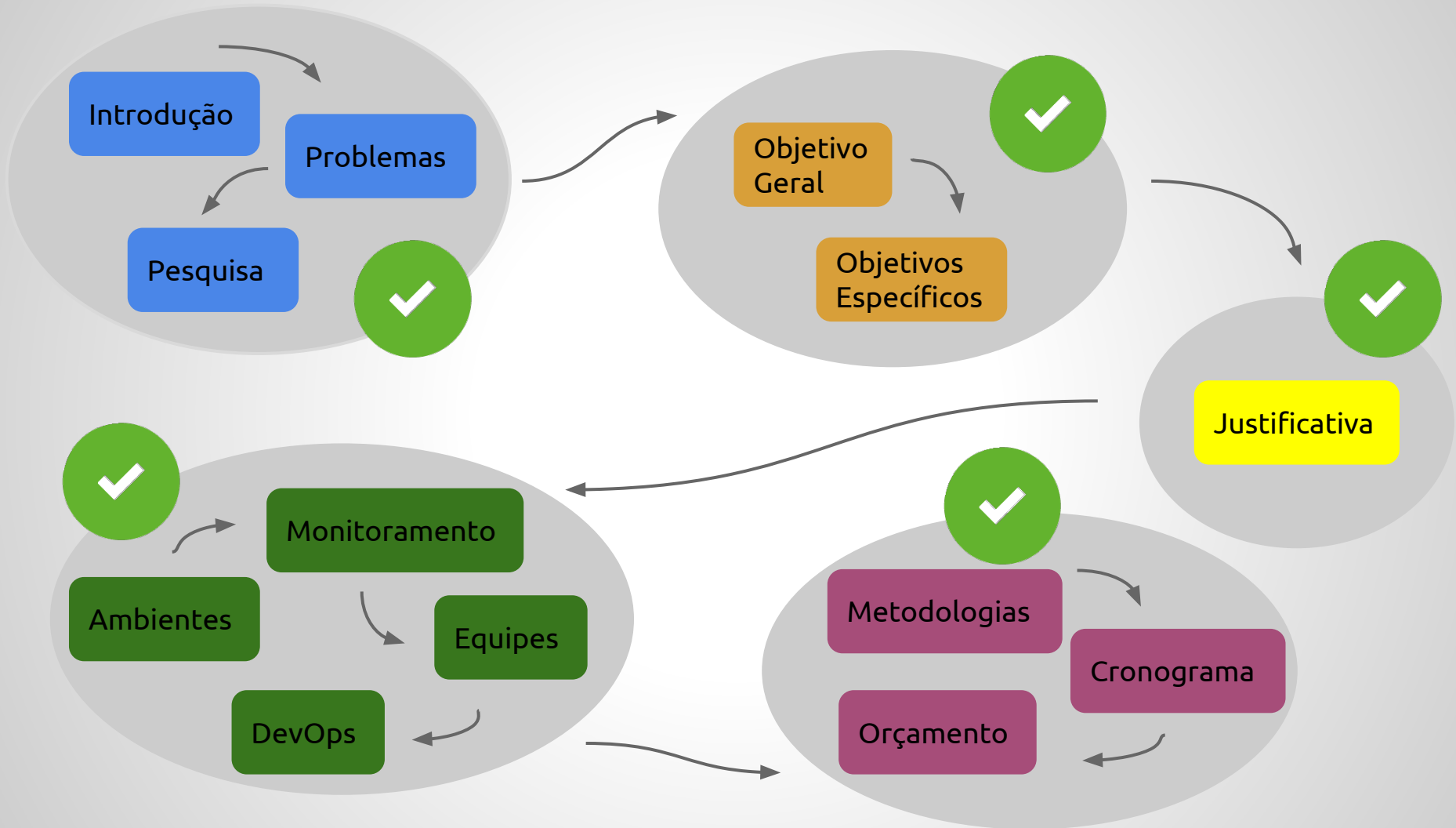
- Pesquisa bibliográfica;
- Pesquisa de levantamento de dados, através de uma questionário sobre o atual funcionamento;
- Analise e interpretação dos problemas;
- Elaborações de etapas;
- Relatando os dados conquistados;

Cronograma

[illegible]

Orçamento

Nenhum gasto será necessário para este projeto.



Referências

- 4LINUX. O que é Monitoramento? Disponível em: <<http://www.4linux.com.br/o-que-e-monitoramento>>. Acesso em: 15 abr. 2015.
- CARVALHO, Guto. O que é DevOps afinal?. Disponível em: <<http://gutocarvalho.net/octopress/2013/03/16/o-que-e-um-devops-afinal/>>. Acesso em: 16 mar. 2015.
- DUVALL, Paul. Agile DevOps: Quebrando os silos. 2013. Disponível em: <<http://www.ibm.com/developerworks/br/library/a-devops9/#ibm-pcon>>. Acesso em: 27 abr. 2015.

Referências

- LEITE, Jair C. O Processo de Desenvolvimento de Software. 2000. Disponível em: <<https://www.dimap.ufrn.br/~jair/ES/c2.html>>. Acesso em: 27 abr. 2015.
- RELIC, New. O que é DevOps? Metodologia, benefícios e ferramentas. Disponível em: <<http://newrelic.com/devops/what-is-devops>>. Acesso em: 01 maio 2015.
- SATO, Danilo. DevOps na prática: entrega de software confiável e automatizada. São Paulo: Casa do Código, 2013. 248 p.

Muito obrigado, perguntas?

Chapecó - SC, 2015

