

# PROCESSOS DE SOFTWARE

RAFAEL PRIKLADNICKI

2023/1 - Aula 3

# Mudanças

- A mudança é inevitável em todos os grandes projetos de software
  - Os requisitos do sistema mudam ao mesmo tempo que o negócio responde a pressões externas
  - A disponibilidade de novas tecnologias e aumento de desempenho
- Portanto, qualquer que seja o modelo do software de processo, é essencial que possa acomodar mudanças no software em desenvolvimento
- A mudança aumenta os custos de desenvolvimento de software: retrabalho
  - Por exemplo, se os relacionamentos entre os requisitos do sistema foram analisados e novos requisitos foram identificados, alguma ou toda análise de requisitos deve ser repetida
- Pode ser necessário reprojeter o sistema de acordo com os novos requisitos, mudando programas fazendo novos testes

Adaptado do Prof. Dilnei Venturini

# Custo do Retrabalho

- Prevenção de mudanças: o processo de software inclui atividades capazes de antecipar as mudanças
  - Por ex., um protótipo de sistema pode ser desenvolvido para mostrar algumas características-chave do sistema para os clientes, antes de se comprometer com novos esforços.
- Tolerância a mudanças: o processo foi projetado para que as mudanças possam ser acomodadas com um custo relativamente baixo.
  - Isso normalmente envolve alguma forma de desenvolvimento incremental.
  - As alterações propostas podem ser aplicadas em incrementos que ainda não foram desenvolvidos. Se isso for impossível, então apenas um incremento deve ser alterado

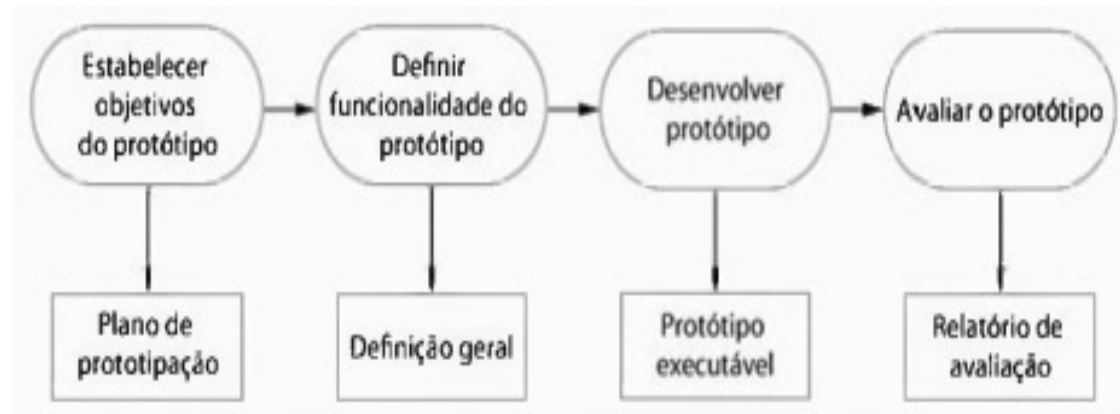
# Prototipação

- Uma versão do sistema ou de parte dele é desenvolvida rapidamente para verificar as necessidades do cliente e a viabilidade de algumas decisões de projeto
- Processo que previne mudanças, já que permite aos usuários experimentarem o sistema antes da entrega e, então, refinarem seus requisitos
- O número de propostas de mudanças de requisitos a ser feito após a entrega é, portanto, suscetível de ser reduzido.
- Pode ser utilizada em:
  - Na processo de engenharia de requisitos para auxiliar na elicitação e validação dos requisitos
  - No processo de projeto para avaliar opções e apoiar o projeto de interface de usuário

# Prototipação

- Aumentam a usabilidade dos sistemas
- Permitem um melhor entendimento das reais necessidades dos usuários
- Aumentam a qualidade do projeto
- Aumentam a manutibilidade
- Reduzem o esforço de desenvolvimento

# Prototipação



# Prototipação

- Deve ser baseado em linguagens de prototipação ou outras ferramentas
- Pode envolver não tratar algumas funcionalidades
  - Protótipos devem focar em áreas onde o negócio não é bem compreendido
  - Teste de erro e recuperação normalmente não são tratados
  - O foco deve ser requisitos funcionais antes de requisitos não funcionais como confiabilidade e segurança

## Descarte de Protótipos

- Protótipos devem ser descartados após o uso
- Eles não são uma base para o sistema em produção
  - Pode ser impossível ajustar o protótipo para atender aos requisitos não funcionais
  - Protótipos normalmente não são documentados
  - As mudanças durante o desenvolvimento do protótipo provavelmente terão degradado a sua estrutura
  - Padrões de qualidade organizacional geralmente são relaxados para o desenvolvimento do protótipo



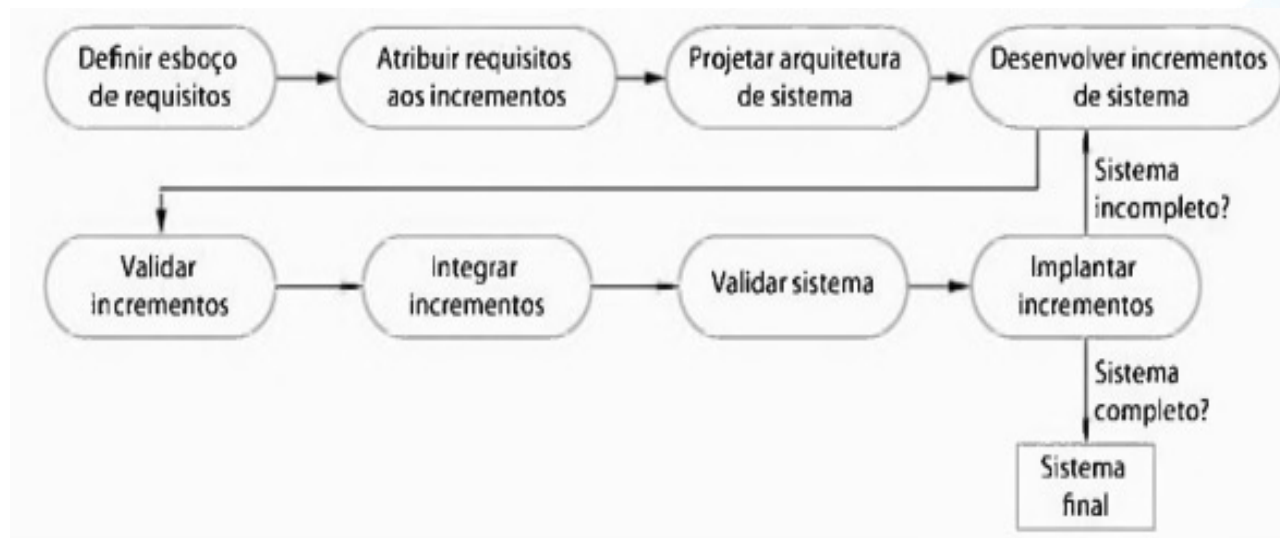
# Entrega Incremental

- Ao invés de uma entrega única, o desenvolvimento e entregas são divididos em incrementos que implementam um subconjunto de uma determinada funcionalidade
- Os clientes identificam, em linhas gerais, os serviços a serem fornecidos pelo sistema e os priorizam. Os requisitos de maior prioridade são implementados e entregues em primeiro lugar
- Durante o desenvolvimento podem ocorrer mais análises de requisitos para incrementos posteriores, mas mudanças nos requisitos do incremento atual não são aceitas

# Entrega Incremental

- Desenvolvimento incremental
  - Desenvolvimento do sistema em incrementos com a avaliação de cada incremento antes de desenvolvimento do próximo
  - Abordagem tradicional dos métodos ágeis
  - Avaliação realizada pelo usuário/cliente
- Entrega incremental
  - Entrega de um incremento para utilização pelo usuários final
  - Gera uma avaliação mais real em função do uso do software
  - Dificuldade de implementação: sistema entregue possui inicialmente menos funcionalidades do que o atual

# Entrega Incremental



# Entrega Incremental

- Os incrementos iniciais servem como protótipos e gera experiência para os requisitos dos incrementos posteriores
- Os clientes não necessitam esperar até que todo o sistema seja entregue para obter ganhos a partir dele
- Facilita a incorporação das mudanças no sistema
- Os requisitos mais importantes recebem a maioria dos testes
- Diminui o risco de fracasso do projeto

## Desafios

- A maioria dos sistemas exige um conjunto de recursos básicos, usados por diferentes partes do sistema
  - Como os requisitos não são definidos em detalhes até que um incremento possa ser implementado, pode ser difícil identificar recursos comuns, necessários a todos os incrementos
- Pode ser difícil quando um sistema substituto está sendo desenvolvido
- A essência do processo iterativo é a especificação ser desenvolvida em conjunto com o software. Não há especificação completa do sistema até que o último incremento seja especificado
  - Isso pode causar conflitos com o modelo de compras de muitas organizações

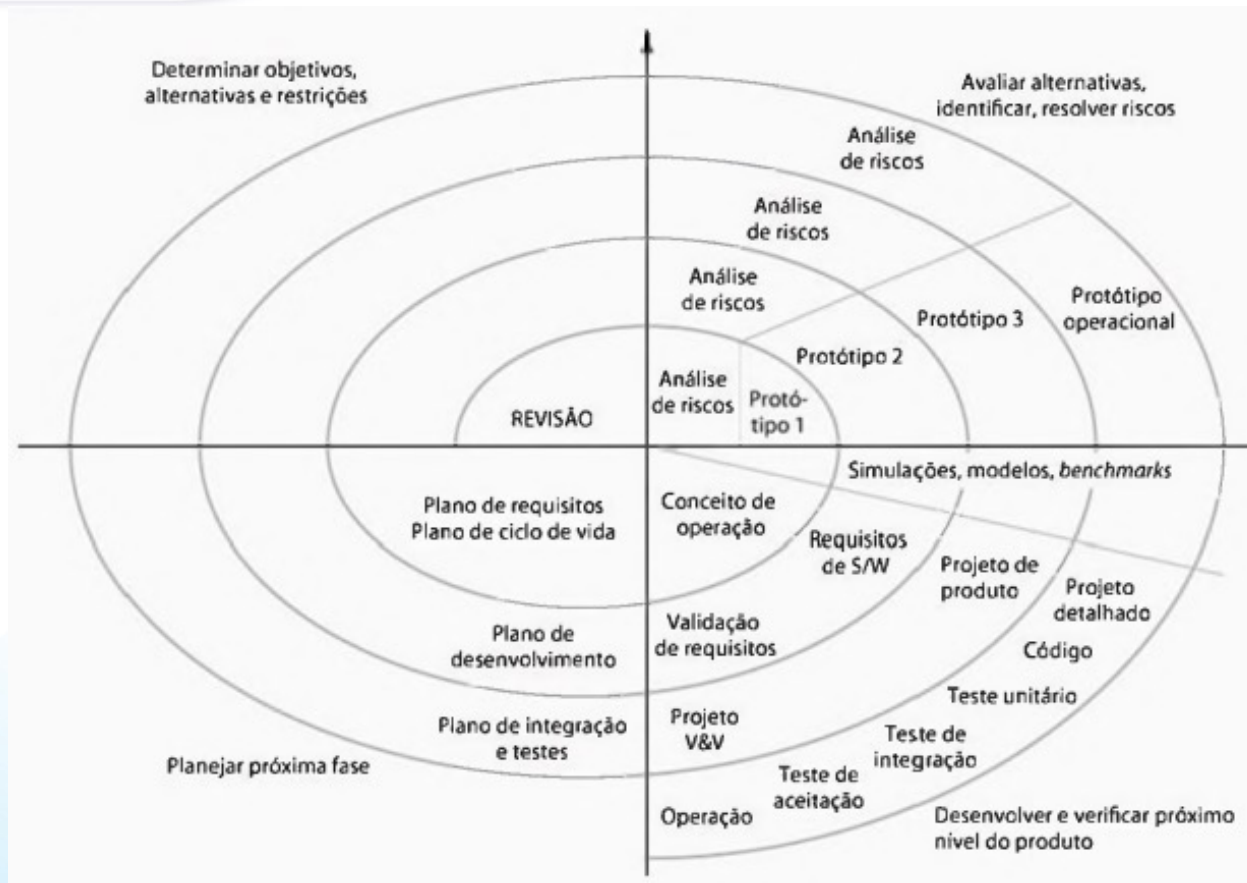
## Desafios

- Sistemas muito grandes com equipes distribuídas
- Sistemas embutidos, em que o software depende do desenvolvimento de hardware
- Sistemas críticos, em que todos os requisitos devem ser analisados na busca por interações capazes de comprometer a proteção ou a segurança do sistema
- Alternativa: desenvolvimento de um protótipo de forma iterativa e usado como uma plataforma para experimentos com os requisitos e projeto do sistema

# Espiral - Boehm

- Processo de software dirigido a riscos
- O processo de software é representado como uma espiral, e não como uma seqüência de atividades, com alguns retornos de uma para outra
- Cada volta na espiral representa uma fase do processo de software.
- Dessa forma, a volta mais interna pode preocupar-se com a viabilidade do sistema; o ciclo seguinte, com definição de requisitos; o seguinte, com o projeto do sistema, e assim por diante.
- O modelo em espiral combina prevenção e tolerância a mudanças
  - As mudanças são um resultado de riscos de projeto e inclui atividades explícitas de gerenciamento de riscos para sua redução

# Espiral - Boehm





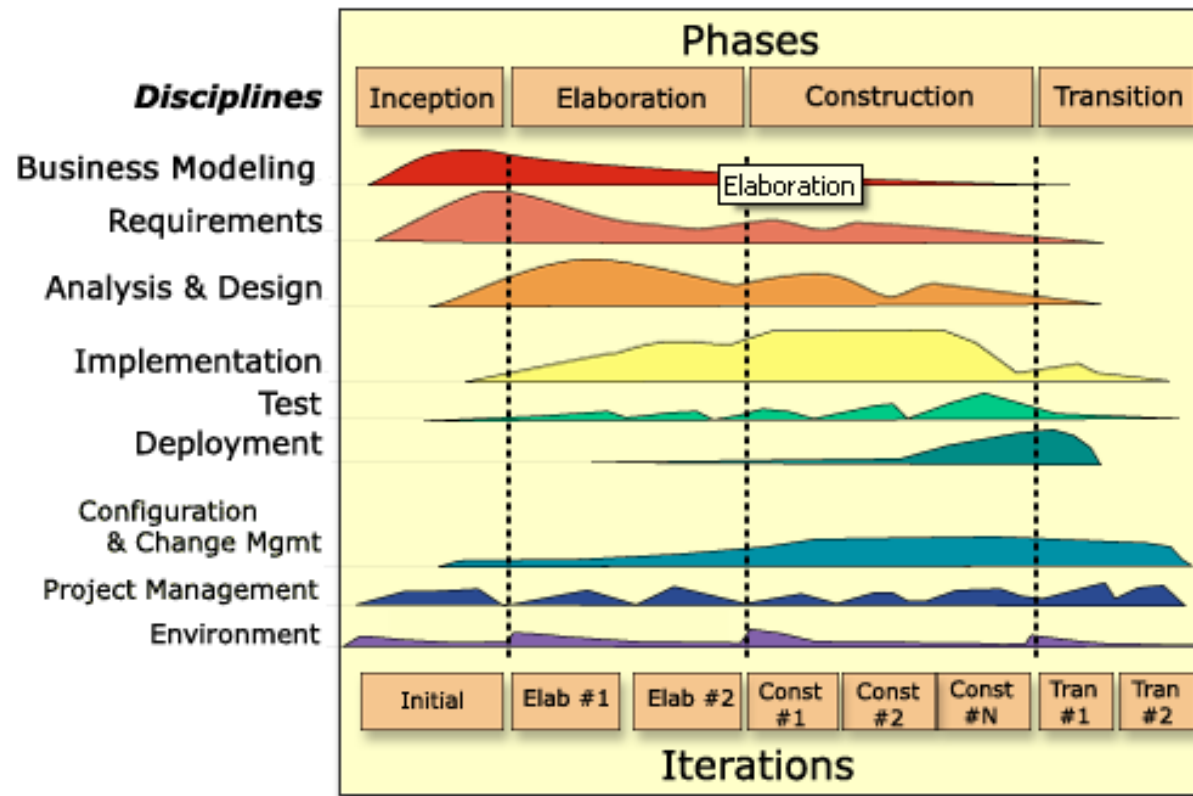
## Espiral - Boehm

- O modelo em espiral combina prevenção e tolerância a mudanças
- As mudanças são um resultado de riscos de projeto e inclui atividades explícitas de gerenciamento de riscos para sua redução
- Riscos levam a mudanças no software e problemas de projeto, como estouro de prazos e custos
- Por exemplo
  - Se a intenção é usar uma nova linguagem de programação, um risco é o de os compiladores disponíveis não serem confiáveis ou não produzirem um código-objeto eficiente o bastante

## Espiral - Boehm

- Definição de objetivos
- Avaliação e redução de riscos
- Desenvolvimento e validação
- Planejamento

# Processo Unificado



# Resumo

- Os processos de software são as atividades envolvidas na produção de um sistema de software
- Modelos de processos de software são representações abstratas desses processos
- Modelos gerais de processo descrevem a organização dos processos de software
- Exemplos desses modelos gerais incluem o modelo em cascata, o desenvolvimento incremental e o desenvolvimento orientado a reuso
- Engenharia de requisitos é o processo de desenvolvimento de uma especificação de software
  - As especificações destinam-se a comunicar as necessidades de sistema dos clientes para os desenvolvedores do sistema

# Processo de Software

- Modelo genérico de processo de software
- Atividades
- Artefatos
- Papéis

# Processo de Software

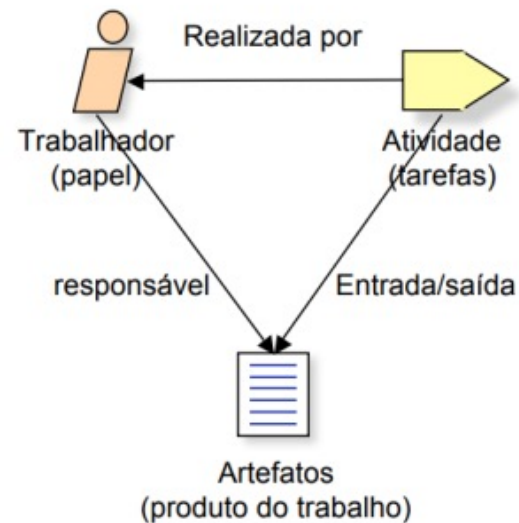
- Processos de software são complexos e dependem de vários fatores como:
  - Produto a ser desenvolvido
  - Equipe
  - Recursos disponíveis
- Não existe um processo ideal, ele pode ser adaptado de acordo com o contexto, podendo ser mais formal ou mais flexível

# Processo de Software

- O que deve ser considerado para definir um processo de software?
  - Características da aplicação (domínio do problema, tamanho, complexidade, etc.)
  - Tecnologia a ser adotada na sua construção (paradigma de desenvolvimento, linguagem de programação, mecanismo de persistência, etc.)
  - Organização onde o produto será desenvolvido
  - Características da equipe
  - Estabilidade dos requisitos
  - Outros
- Modelos de Ciclo de Vida fornecem o arcabouço para a definição do processo

# Processo de Software

- Um processo / fluxo de trabalho é organizado por um conjunto de atividades
- Atividades são tarefas de responsabilidade de um membro da equipe com uma função específica, um papel específico
- Um artefato é resultante de uma atividade





# Atividades

- Processos de software envolvem diversas atividades que podem ser classificadas quanto ao seu propósito:
  - Atividades de Desenvolvimento: atividades diretamente relacionadas ao processo de desenvolvimento do software, ou seja, que contribuem diretamente para o desenvolvimento do produto
    - Ex.: especificação e análise de requisitos, projeto e implementação
  - Atividades de Gerência de Projeto: aquelas relacionadas ao planejamento e acompanhamento gerencial do projeto
    - Ex.: Realização de estimativas, elaboração de cronogramas, análise dos riscos do projeto, etc.
  - Atividades de Apoio: atividades relacionadas principalmente com a garantia da qualidade do produto em desenvolvimento e do processo de software utilizado
    - Ex.: revisões e inspeções de produtos (intermediários ou finais) do desenvolvimento



# Atividades

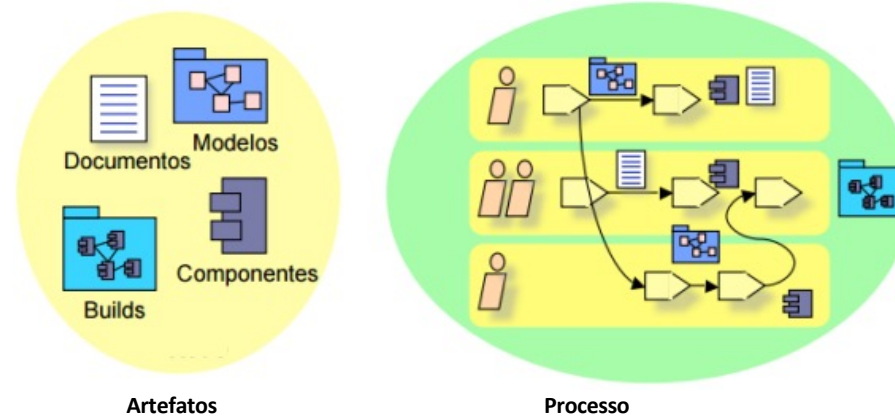
- Outros exemplos
  - Sprint Planning
  - Daily Scrum
  - Sprint Retrospective
  - Planning Poker
  - Inspiração
  - Ideação

## Artefatos

- Do ponto de vista do usuário, é um conjunto de informações que o auxiliem na operação do sistema, como frames de ajuda, documentação do sistema, documentação do usuário, ajuda online, etc.

# Artefatos

- Em um processo, as atividades e papéis consomem artefatos
- Um processo define quem produz e/ou consome os artefatos



# Artefatos

- Outros exemplos
  - Documento de Visão
  - Diagrama de Casos de Uso
  - Diagrama de Clases
  - Product backlog
  - Sprint Backlog
  - Mapa de Empatia
  - Wireframes
  - Storyboards

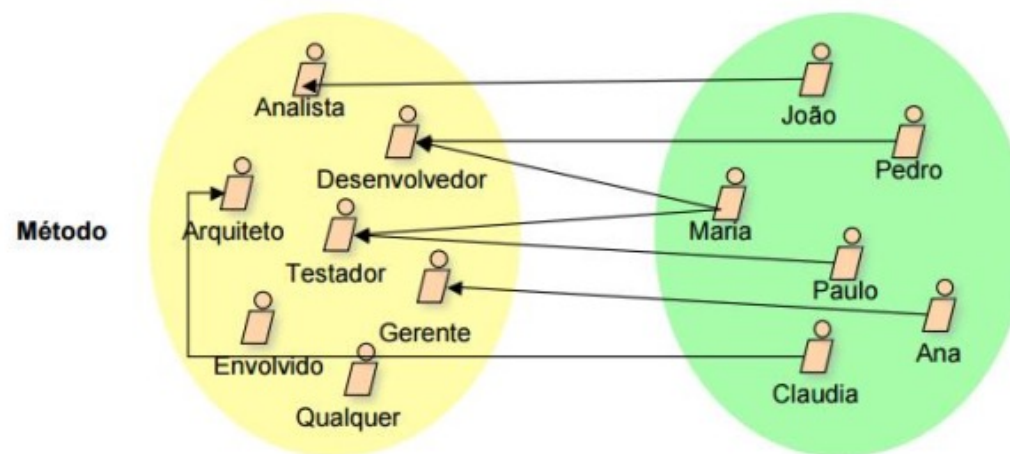
# Papéis

- Em um processo de software deve-se definir quais recursos humanos assumirão quais papéis
- Os papéis podem variar de um processo para outro
- Alguns exemplos:
  - Analista
  - Arquiteto
  - Desenvolvedor
  - Testador
  - Gerente

# Papéis

- Em um processo de software deve-se definir quais recursos humanos assumirão quais papéis
- Os papéis podem variar de um processo para outro
- Alguns exemplos:
  - Analista
  - Arquiteto
  - Desenvolvedor
  - Testador
  - Gerente
- Outros exemplos:
  - Software Architect
  - Tech Lead
  - Product Owner
  - Scrum Master
  - Data Scientist
  - QA
  - UX Designer
  - Business Analyst
  - Agile Coach

# Papéis





## Exercício 1

Baseado no conteúdo visto até o momento, sugira, e justifique, o modelo genérico de processo de software mais adequado para ser usado como base para a gerência do desenvolvimento dos sistemas a seguir:

1. Um sistema para controlar o antibloqueio de frenagem de um carro
2. Um sistema de realidade virtual para dar apoio à manutenção de software
3. Um sistema de contabilidade para uma universidade, que substitua um sistema já existente
4. Um sistema interativo de planejamento de viagens que ajude os usuários a planejar viagens com menor impacto ambiental

Entrega até o dia 22/3 no moodle.

## Referências

