

# PROGRAMAÇÃO ORIENTADA PELOS OBJECTOS

Desenvolvimento Orientado pelos Objectos

# Metodologia de trabalho

2

- Vamos estabelecer algumas regras para a resolução de problemas, usando a Programação Orientada pelos Objectos

**StupidFriendsBook**

## Abordagem ao desenvolvimento de projectos em POO

- Documento de partida
- Fase 1: Modelação
- Fase 2: Construção e Validação
- Entregáveis

# Abordagem ao desenvolvimento de projectos em POO

4

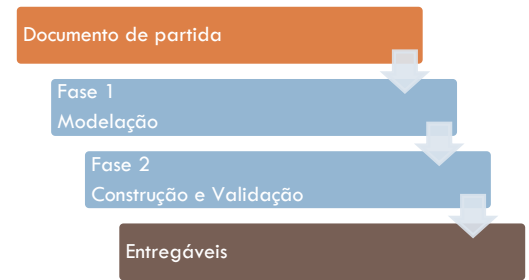
Documento de partida

Fase 1  
Modelação

Fase 2  
Construção e Validação

Entregáveis

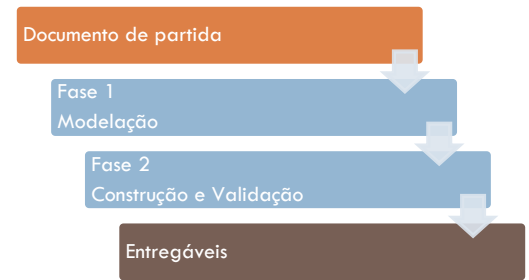
# Documento de partida



5

- **Descrição informal** do sistema / aplicação, numa perspectiva de **alto nível**
  - Descrição dos **objectivos gerais** do sistema/aplicação
  - Descrição mais detalhada das **funcionalidades**
  - Descrição de **cenários** de utilização/interacção

# Modelação



6

- Análise e modelação do sistema a construir
  - A aplicação deve realizar **um conjunto de funcionalidades**. Tais funcionalidades têm que ser **identificadas e nomeadas**
    - Quais são?
  - Cada funcionalidade tem que ser **caracterizada**
    - O que faz?
  - O conjunto das funcionalidades permite definir uma **interface de topo**
    - Esta interface poderá mais tarde ser **refinada**

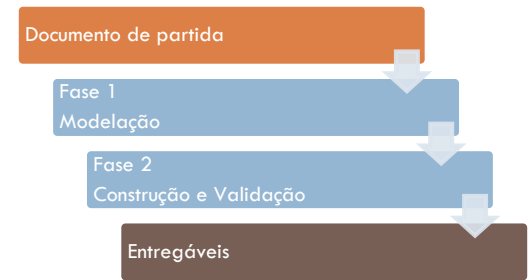
# Modelação



7

- Análise e modelação do sistema a construir (cont.)
  - Deve-se **assegurar** (explicar / justificar) que as operações dessa **interface** de topo permitem realizar os cenários de utilização / interacção
  - Análise da realização de cada operação / funcionalidade de topo com a introdução de:
    - entidades (que encapsulam dados e expõem operações)
    - as operações das entidades
  - Por enquanto, preocupamo-nos sobretudo com o **encapsulamento** e a **ocultação de informação**

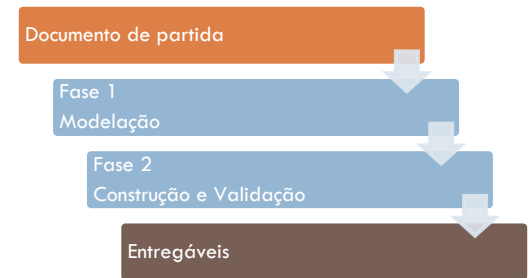
# Modelação



8

- Output da fase de modelação:
  - Identificação da **interface da aplicação**
    - explicação informal da funcionalidade de cada operação
  - **Identificação das entidades do modelo**
    - Nome
    - Lista de operações
    - Explicação informal do efeito de cada operação de cada entidade

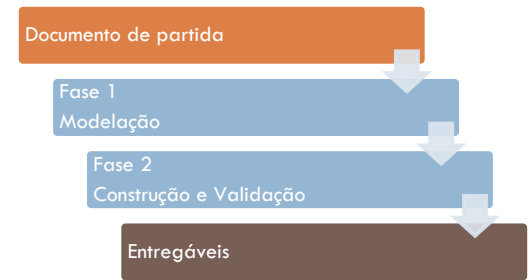
# Modelação



9

- Output da fase de modelação (cont.):
  - **Explicação** de como as interfaces, operações e entidades identificadas contribuem para realizar as funcionalidades caracterizadas anteriormente
  - **Visão organizada da estrutura do sistema / aplicação** a desenvolver em termos de
    - uma unidade principal (interface de topo)
    - um conjunto de entidades e suas operações
      - Independente da representação de dados
      - Independente de algoritmos utilizados, na medida do possível.

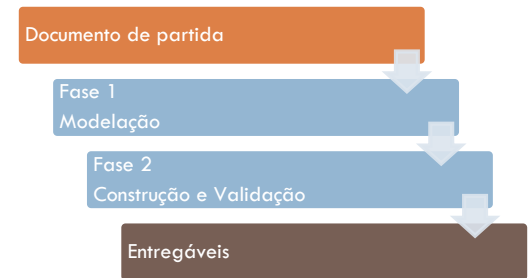
# Construção e Validação



10

- Começar pelo “esqueleto da aplicação”
- As interfaces, entidades e operações desenhadas na fase anterior devem ser implementadas
  - Identificar as classes que as suportam
  - Escolher, para cada classe, uma representação adequada
    - Operações de interface são públicas
    - Operações auxiliares devem ficar escondidas (privadas)
    - Variáveis e constantes para representar o estado
      - Para cada uma indicar o seu tipo e finalidade
      - Podemos usar objectos de um tipo como valores de variáveis usadas dentro de outros objectos

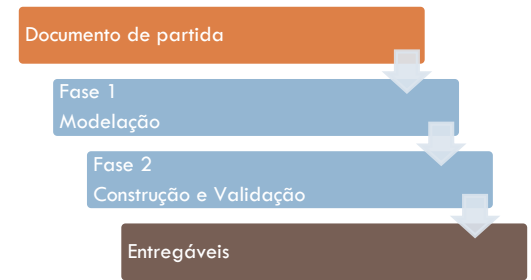
# Construção e Validação



11

- Nesta altura, deverá ter um esqueleto da sua aplicação
  - Interfaces
  - Classes
    - Métodos (por enquanto, vazios)
    - Constantes e variáveis
    - Documentação
- Programa ainda não passa no compilador

# Construção e Validação



12

## ○ Construção do sistema

- Não tentar fazer tudo ao mesmo tempo!
- Definir sequência de implementação que permita ir validando incrementalmente o seu programa

## ○ Exemplo:

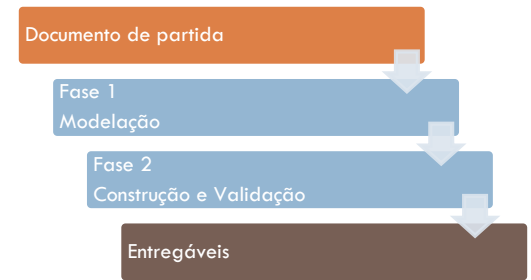
### ○ Interpretador de comandos

- Programa aceita comandos e dados associados, mas (ainda) não faz nada com eles

### ○ Implementação de comandos

- Escolher uma sequência para a implementação de comandos que permita ir testando os comandos, à medida que os implementa

# Construção e Validação

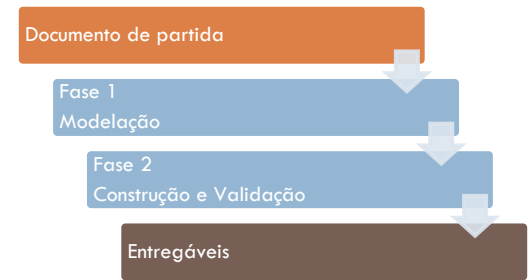


13

## ○ Indicações gerais

- Tente ter sempre uma versão estável do seu programa
  - Compila sem erros
  - Implementa algumas das funcionalidades do seu programa
  - Está bem testada
    - Teste as suas classes, método a método
    - Se necessário, construa pequenos programas de teste
- Não avance para novas fases do desenvolvimento, sem as anteriores estarem sólidas
- Use sempre o melhor estilo de programação e documentação possível

# Entregáveis



14

- No final do desenvolvimento, deve disponibilizar
  - Código fonte do seu programa, devidamente comentado
  - Documentação completa
    - Javadoc
    - Diagramas de classes e interfaces
    - Outros que lhe sejam solicitados
      - Manual de instalação e utilização
      - ...

15

## The StupidFriendsBook

Uma rede social para descobrir a essência dos seus amigos. Serão eles inteligentes, estúpidos, anjinhos, ou bandidos?

**StupidFriendsBook**

# ○ que é o StupidFriendsBook ?

Documento de partida

Fase 1  
Modelação

Fase 2  
Construção e Validação

Entregáveis

16

- Imagine uma rede social onde pode votar, para cada acção de cada um dos seus amigos, se considera que ela é **boa, ou má, para o seu amigo e boa, ou má, para os outros** que sentem as consequências dessa acção
- Nesta rede social, todos podem votar nas acções de todos
- Os votos acumulados permitem construir o perfil das pessoas

# O que é o StupidFriendsBook ?

Documento de partida

Fase 1

Modelação

Fase 2

Construção e Validação

Entregáveis

17

- Cada pessoa no StupidFriendsBook pode ter várias acções, cada uma delas votadas individualmente
- A média das votações em cada acção determina se o perfil de uma pessoa é visto como **inteligente**, **estúpido**, **anjinho** ou **bandido**; é possível consultar essa informação sobre qualquer pessoa
- A pessoa **mais aborrecida** é a pessoa cujas acções são o mais neutras possível
  - Ou seja, cujo efeito médio das acções, para si própria e para os outros, seja o mais próximo de 0 (nem positiva, nem negativa)
- Podemos listar pessoas com determinado perfil

# Construção da personalidade

18

- Pessoa faz acções
- Cada vez que alguém vota numa dessas acções, indica se considera se a acção foi
  - Boa, ou má, para quem a praticou
  - Boa, ou má, para os outros
- Os votos vão-se acumulando
  - Se houver mais votos bons que maus para o próprio, ou pelo menos igual número, considera-se a acção boa para o próprio
  - Raciocínio semelhante, nos votos para os outros
- A média das votações determina a personalidade

# Fulano de tal é inteligente?

19

## ○ Fulano de tal fez frito e cozido

○ Bom para ele?



like



dislike

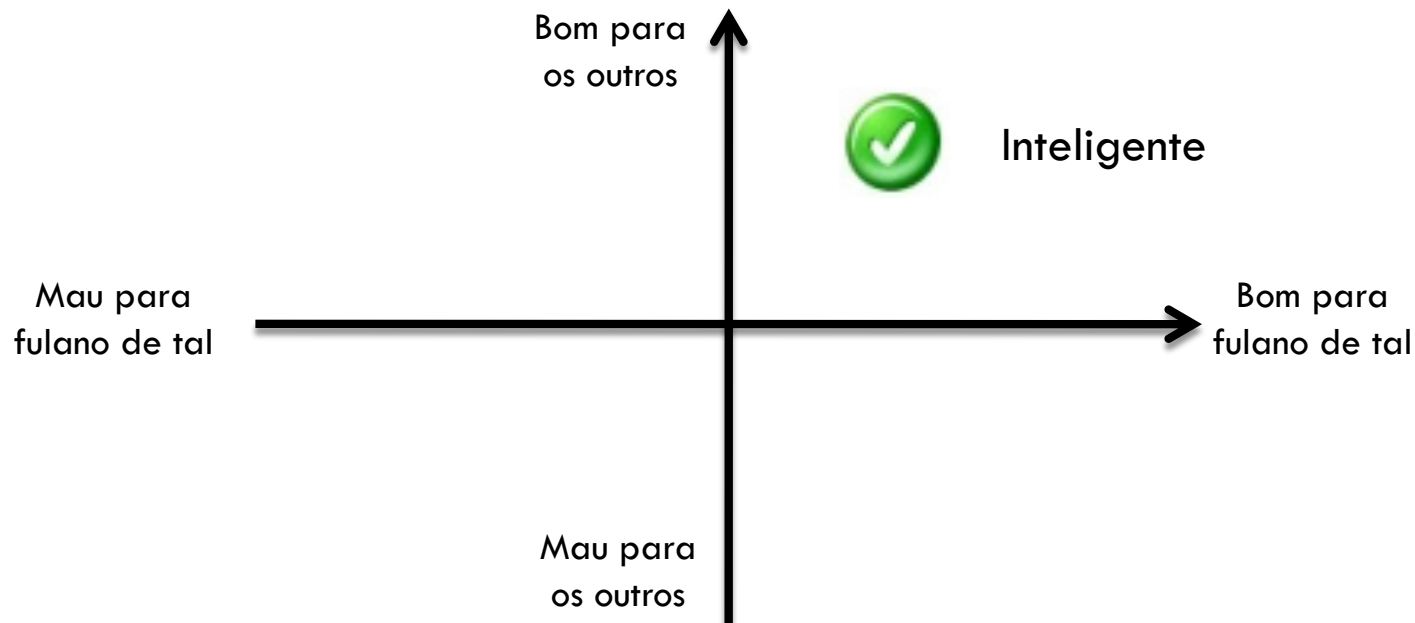
○ Bom para os outros?



like



dislike



# Fulano de tal é inteligente?

20

## ○ Fulano de tal fez frito e cozido

○ Bom para ele?



like



dislike

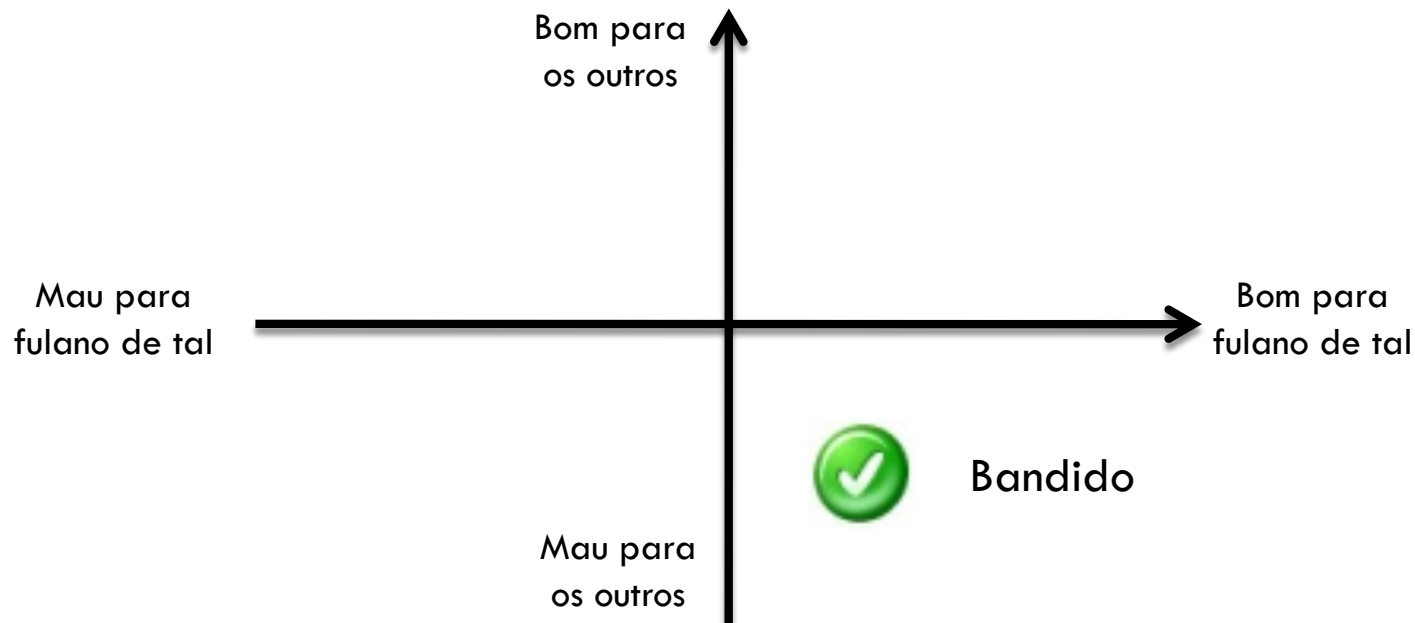
○ Bom para os outros?



like



dislike



# Fulano de tal é inteligente?

21

## ○ Fulano de tal fez frito e cozido

○ Bom para ele?



like



dislike

○ Bom para os outros?



like



dislike



# Fulano de tal é inteligente?

22

## ○ Fulano de tal fez frito e cozido

○ Bom para ele?



like



dislike

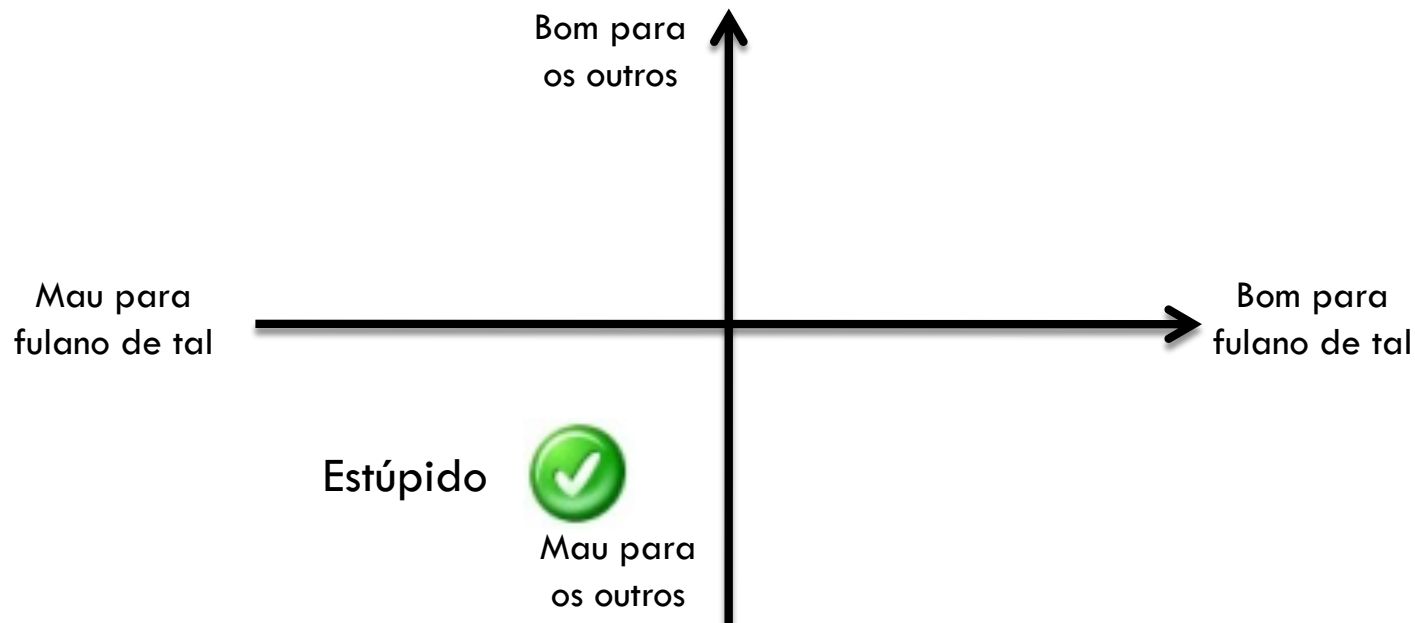
○ Bom para os outros?



like



dislike



# Traço de execução

Documento de partida

Fase 1  
Modelação

Fase 2  
Construção e Validação

Entregáveis

23

Novo  
Adiciona  
Fulano  
Ok  
Adiciona  
Beltrano  
Ok  
Adiciona  
Cicrano  
Ok  
Adiciona  
Beltrano  
Beltrano já existe  
Faz  
Fulano  
Lava pratos  
Ok  
Faz  
Fulano  
Jardinagem  
Ok

Adiciona  
Mais Um  
Ok  
Remove  
Mais Um  
Ok  
Faz  
Fulano  
Lava pratos  
Fulano já Lava pratos  
Faz  
Beltrano  
Pinta grafitti  
Ok  
Faz  
Cicrano  
Pinta grafitti  
Ok  
Vota  
Fulano  
Lava pratos

Bem  
Mal  
Ok  
Vota  
Fulano  
Lava pratos  
Bem  
Bem  
Ok  
Vota  
Fulano  
Lava pratos  
Bem  
Bem  
Ok  
Vota  
Fulano  
Jardinagem  
Bem  
Bem  
Ok

Vota  
Beltrano  
Lava pratos  
Bem  
Bem  
Beltrano não Lava pratos  
Vota  
Beltrano  
Pinta grafitti  
Mal  
Mal  
Ok  
Lista  
Inteligente  
Fulano; Cicrano  
Personalidade  
Beltrano  
Estúpido  
Aborrecido  
Cicrano  
Sair  
Adeus

DI FCT UNL

# Identificando as acções do sistema

24

- Imagine uma rede social onde pode **votar**, para cada acção de cada um dos seus amigos, se considera que ela é **boa, ou má, para o seu amigo e boa, ou má, para os outros** que sentem as consequências dessa acção
- Nesta rede social, todos podem votar nas acções de todos
- Os votos acumulados permitem construir o perfil das pessoas
- Cada pessoa no StupidFriendsBook pode ter várias acções, cada uma delas votadas individualmente
- A média das votações em cada acção determina se o perfil de uma pessoa é visto como **inteligente, estúpido, anjinho ou bandido**; é possível consultar essa informação sobre qualquer pessoa
- A pessoa **mais aborrecida** é pessoa cujas acções são o mais neutras possível
  - Ou seja, cujo efeito médio das acções, para si própria e para os outros, seja o mais próximo de 0 (nem positiva, nem negativa)
- Podemos listar pessoas com determinado perfil

## Acções do sistema:

Criar rede social  
Criar pessoa  
Remover pessoa  
Criar acção  
Votar em acção  
Consultar personalidade  
Descobrir mais aborrecido  
Listar pessoas por perfil  
Sair

# Comandos a disponibilizar

Documento de partida

Fase 1  
Modelação

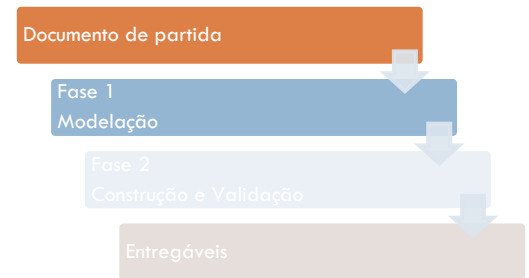
Fase 2  
Construção e Validação

Entregáveis

25

- Novo (Criar rede social)
  - Cria um novo StupidFriendsBook, que começa por estar vazio
- Adiciona amigo (Criar pessoa)
  - Acrescenta um novo amigo ao StupidFriendsBook; se já existir, não faz nada
    - Informação que caracteriza o amigo: nome
- Remove amigo (Remover pessoa)
  - Remove um amigo do StupidFriendsBook; se não existir, não faz nada
    - Informação que caracteriza o amigo: nome

# Comandos a disponibilizar



26

- Acção (Criar acção)
  - Adiciona uma nova acção ao amigo
    - Informação que caracteriza o amigo: nome
    - Informação que caracteriza a acção: descrição
- Vota (Votar em acção)
  - Vota numa determinada acção de um amigo
    - Informação que caracteriza o amigo: nome
    - Informação que caracteriza a acção: descrição, impacto no próprio e impacto nos outros
      - Ao votar, ou se considera uma acção benéfica, ou prejudicial

# Comandos a disponibilizar

Documento de partida

Fase 1  
Modelação

Fase 2  
Construção e Validação

Entregáveis

27

- Personalidade (Consultar personalidade)
  - Consultar a personalidade de uma pessoa
    - Informação que caracteriza a consulta: nome da pessoa
- Aborrecido (Descobrir mais aborrecido)
  - Lista a pessoa “mais aborrecida” no StupidFriendsBook
- Lista amigos (Listar pessoas por perfil)
  - Lista todos os amigos com um determinado traço de personalidade dominante (por exemplo, lista todos os anjinhos)
    - Informação que caracteriza a listagem: traço de personalidade comum às pessoas listadas
- Sair
  - Termina o programa

# Traço de execução

Documento de partida

Fase 1  
Modelação

Fase 2  
Construção e Validação

Entregáveis

28

Novo  
Adiciona  
Fulano  
Ok  
Adiciona  
Beltrano  
Ok  
Adiciona  
Cicrano  
Ok  
Adiciona  
Beltrano  
Beltrano já existe  
Faz  
Fulano  
Lava pratos  
Ok  
Faz  
Fulano  
Jardinagem  
Ok

Adiciona  
Mais Um  
Ok  
Remove  
Mais Um  
Ok  
Faz  
Fulano  
Lava pratos  
Fulano já Lava  
pratos  
Faz  
Beltrano  
Pinta grafitti  
Ok  
Faz  
Cicrano  
Pinta grafitti  
Ok  
Vota  
Fulano  
Lava pratos

Bem  
Mal  
Ok  
Vota  
Fulano  
Lava pratos  
Bem  
Bem  
Ok  
Vota  
Fulano  
Lava pratos  
Bem  
Bem  
Ok  
Vota  
Fulano  
Jardinagem  
Bem  
Bem  
Ok

Vota  
Beltrano  
Lava pratos  
Bem  
Bem  
Beltrano não Lava pratos  
Vota  
Beltrano  
Pinta grafitti  
Mal  
Mal  
Ok  
Lista  
Inteligente  
Fulano; Cicrano  
Personalidade  
Beltrano  
Estúpido  
Aborrecido  
Cicrano  
Sair  
Adeus

## Acções do sistema:

- Criar rede social
- Criar pessoa
- Remover pessoa
- Criar acção
- Votar em accção
- Consultar personalidade
- Descobrir mais aborrecido
- Listar pessoas por perfil
- Sair

# Entidades

Documento de partida

Fase 1  
Modelação

Fase 2  
Construção e Validação

Entregáveis

29

