# Sistema Lost & Found

Desenvolvedor: Leandro Silva Soares – leandrogamedesigner@gmail.com

## Sobre o sistema

O sistema Lost & Found tem como objetivo ajudar instituições a informatizar o departamento de achados e perdidos.

O sistema permite que o usuário possa fazer buscas na base de dados, cadastrar um item perdido ou achado. Quando alguém acha o item na base, pode solicitar o item com a pessoa que o encontrou.

O sistema tem seis telas:

* Home
* Lista de objetos
* Registrar objeto
* Enviar e-mail
* Login
* Signin

## Planejamento

### Composição das telas

#### Home

Na tela home temos uma chamada ao usuário para registrar sua perda, ou fazer cadastro no site.

#### Lista de objetos

Na tela de lista de objetos o usuário pode fazer uma busca na base, ou procurar pelos objetos listados.

Clicando no botão “Achei este item“ ou “Perdi este item“ de um objeto perdido o usuário é levado a tela de envio que e-mail “Enviar mensagem“.

#### Registrar objeto

Na tela registrar objeto o usuário, logado, pode cadastrar um objeto como perdido, caso ele tenha perdido este objeto, ou achado, caso o tenha achado.

Os campos de cadastro são:

1. Nome do objeto – obrigatório
2. Descrição – obrigatório
3. E-mail – obrigatório
4. Imagem – opcional
5. Status – obrigatório

#### Enviar email

Na tela de enviar email o usuário pode escrever uma mensagem para que possa ser combinada uma ação para a devolução do objeto perdido.

O usuário pode cancelar esta ação.

#### Login

Na tela login temos os campos:

* Username
* Password

O usuário tem a opção de clicar no botão “Sign in” para ir à tela de cadastro.

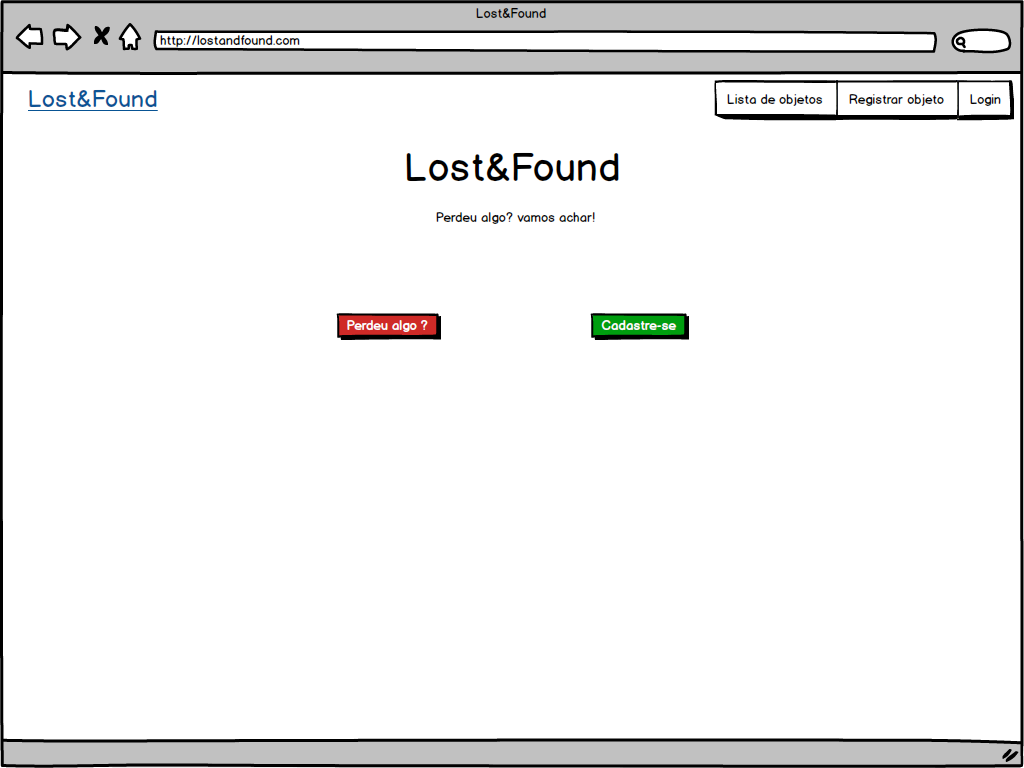
#### Signin

Na tela de signin temos os campos:

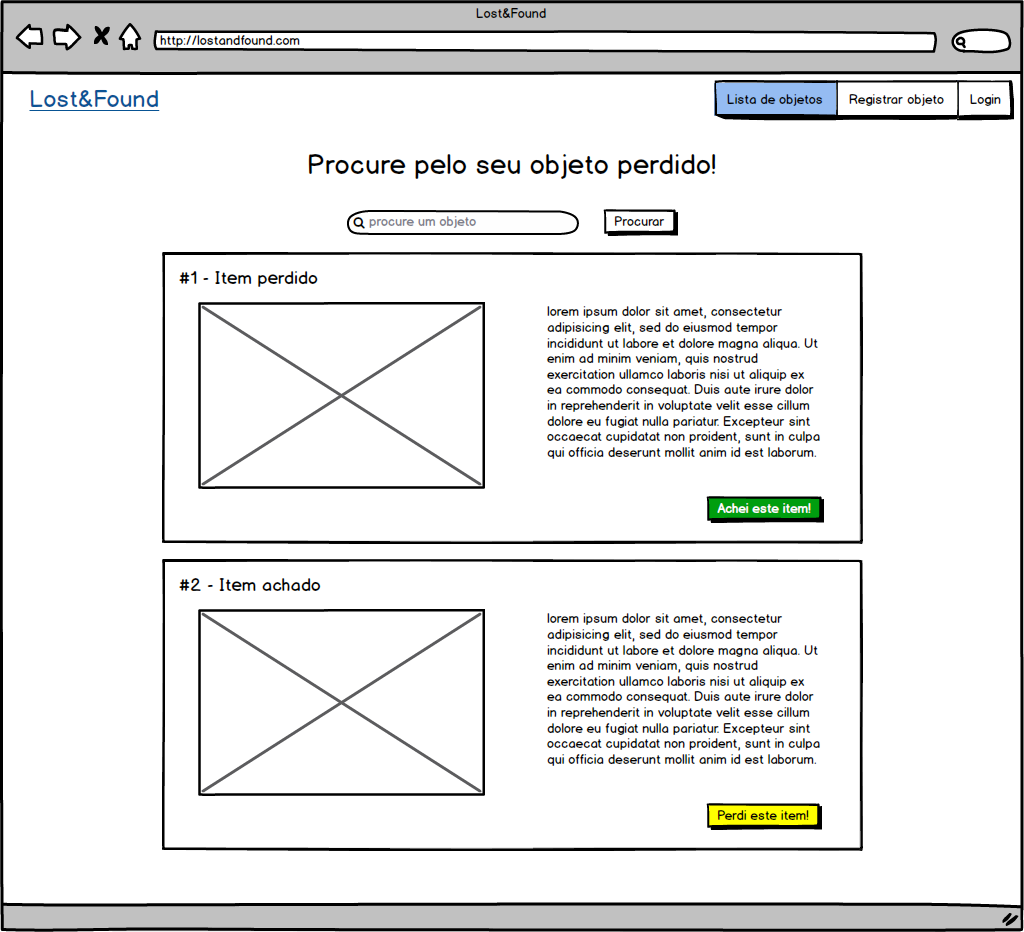
* Username
* Password
* Email
* Nome completo

### Wireframe

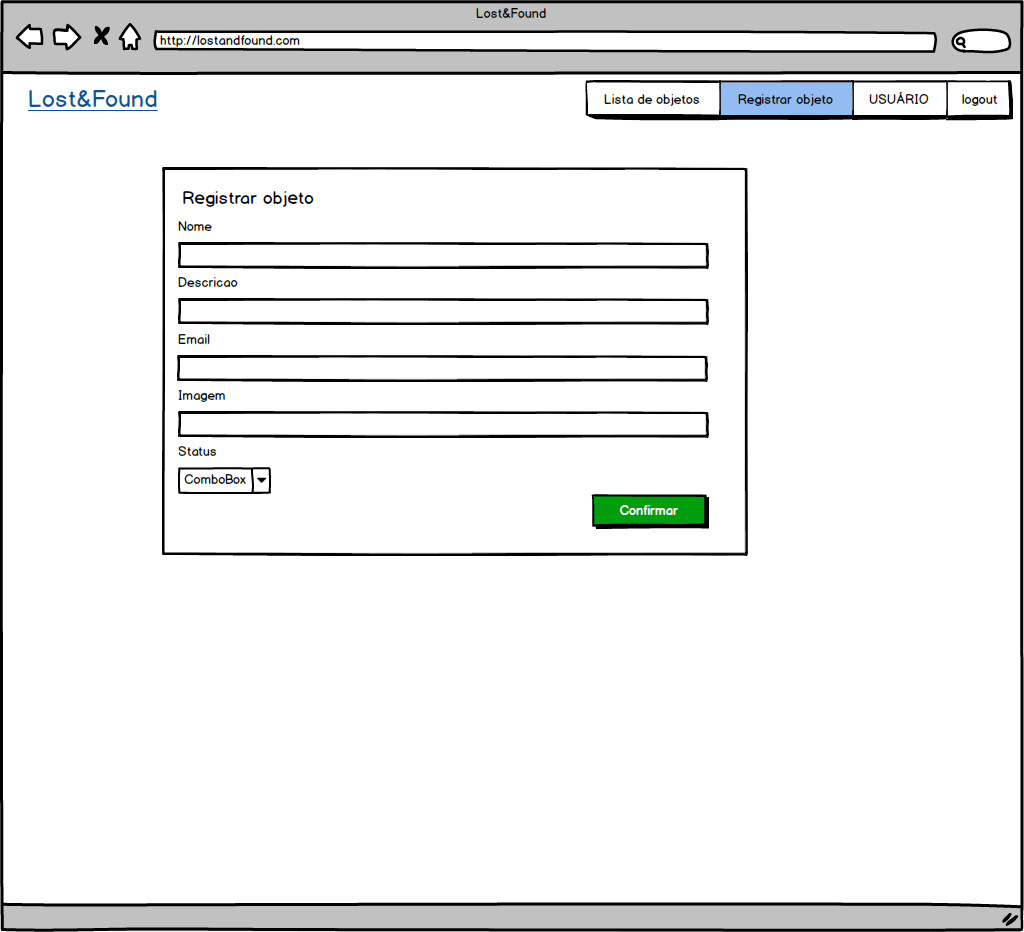
#### Home



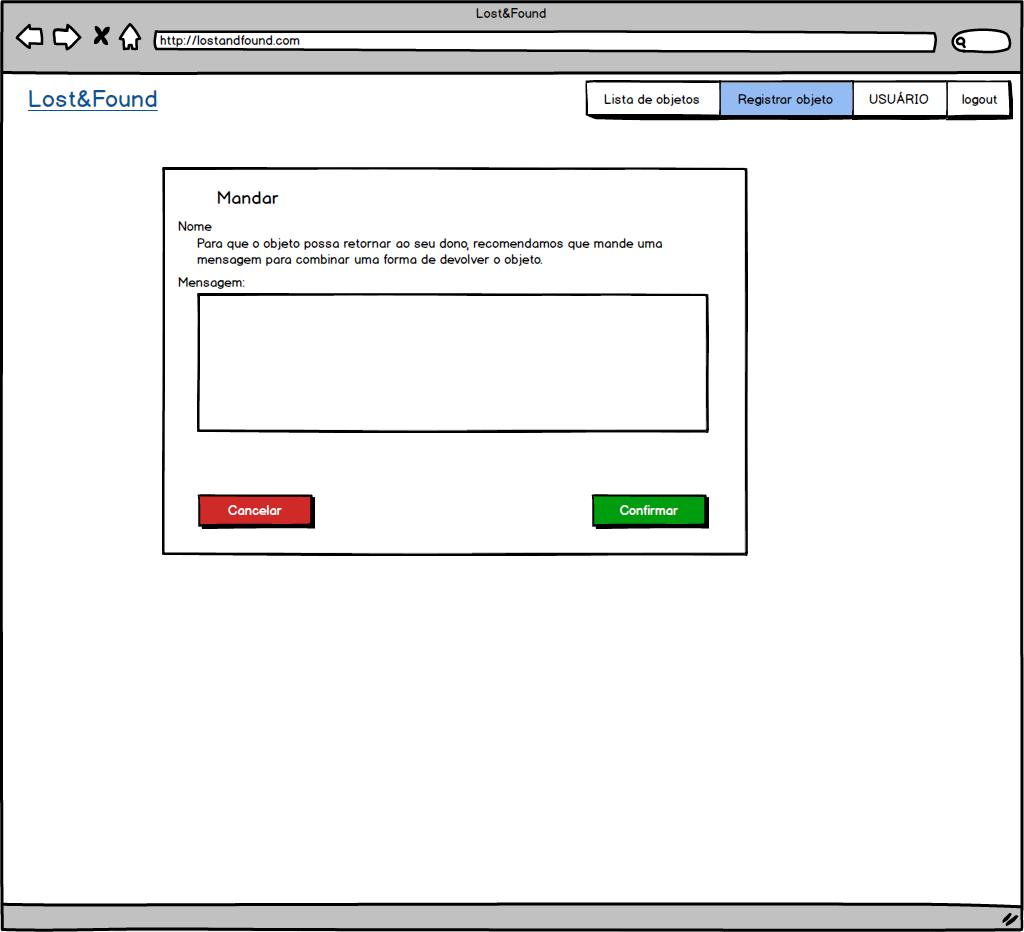
#### Lista de objetos



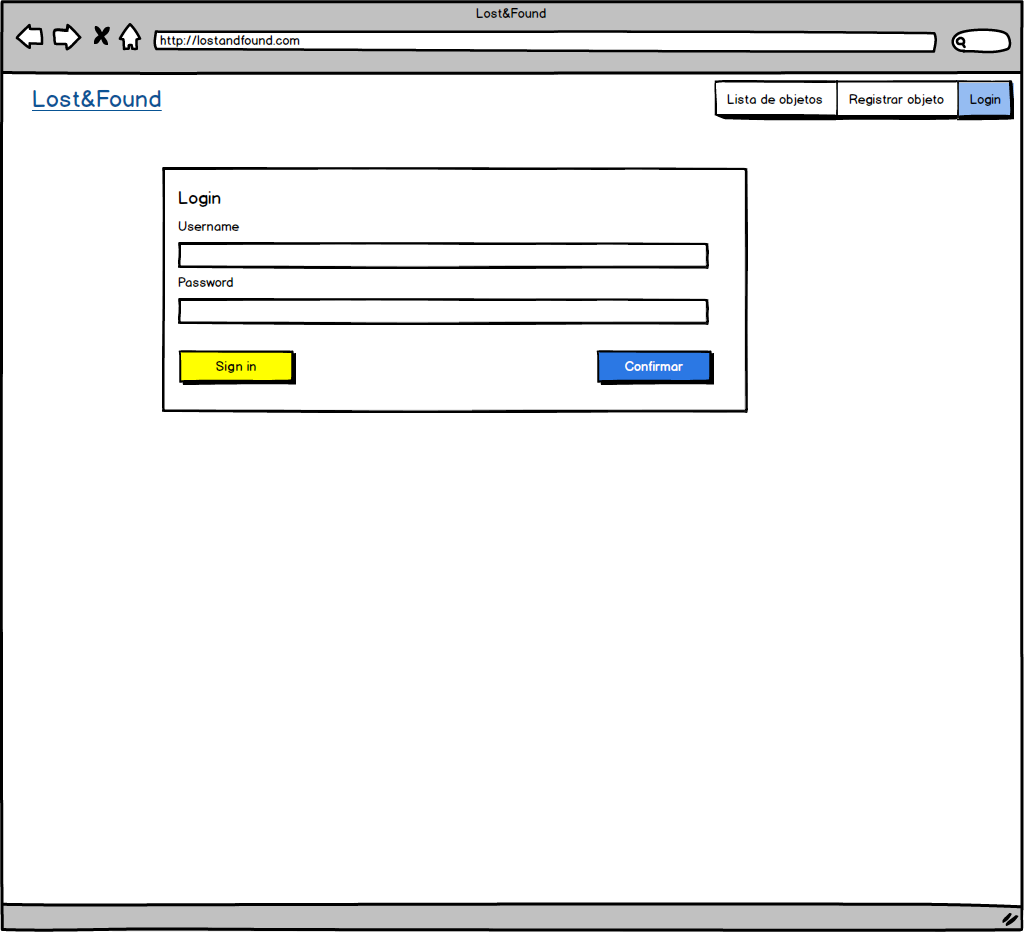
#### Registrar objeto



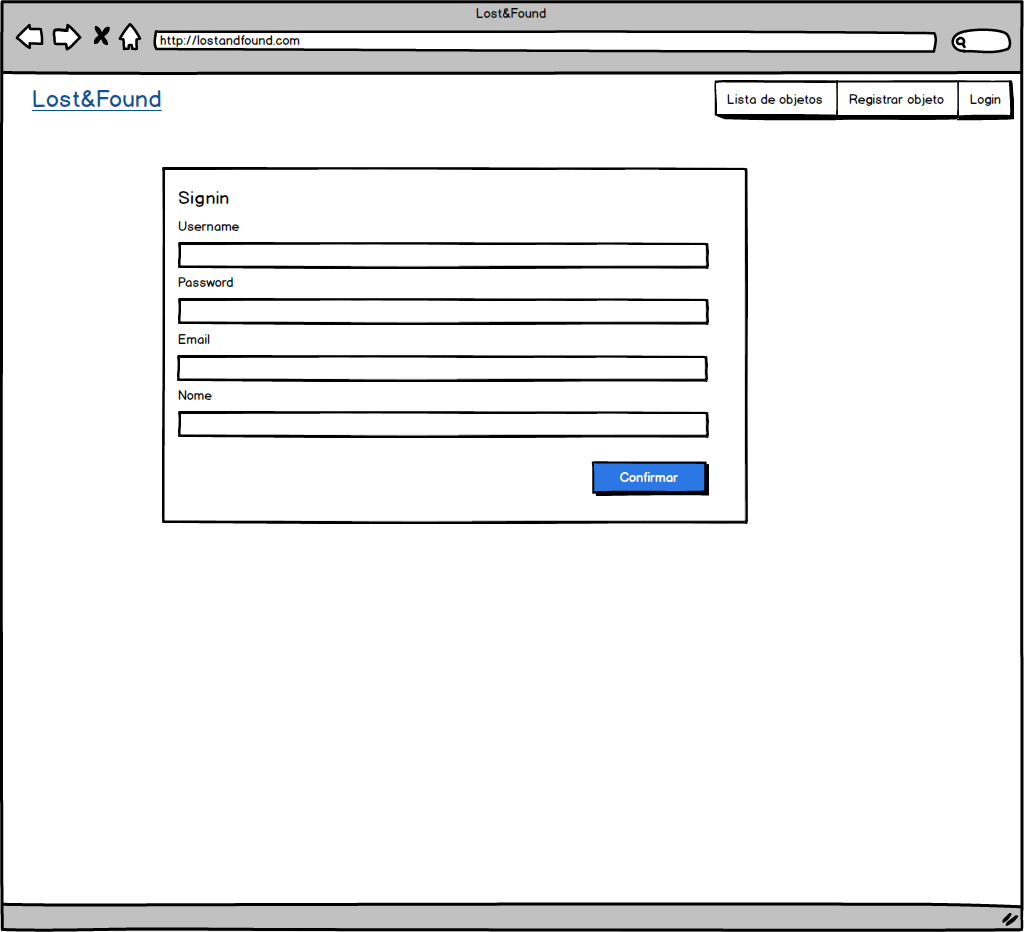
#### Enviar email



#### Login

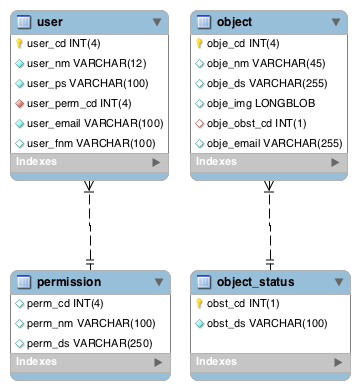


#### Signin



### Banco de dados

No banco de dados planejei as seguintes tabelas:



## Desenvolvimento

### *Views*

Na procura em aplicar o melhor possível a orientação a objetos ao criar as *views*, e acabei desenvolvendo uma pequena *library* de renderização de HTML a vcore, inspirada nas implementações que fizemos em aula. Também utilizei a biblioteca visual twitter bootstrap.

### *Controllers*

Implementei um *controller* para cada view principal seguindo o *wireframe*, e criei uma *controller* base para que meus *controllers* herdassem algumas características que eu desejava.

### *Models*

Implementei uma *model* para cada tabela principal do banco de dados, uma *UserModel* e uma *ObjectModel*.

### *Controller-Model*

Procurei uma melhor a forma de tratar os dados entre *model* e *controller* e criei um objeto de modelo que é passado ao invés dos dados puros, no caso da tabela do banco de dados “objeto”.

### Problemas gerais

Tive alguns problemas com o envio de email, tentei utilizar o outlook para fazer o envio, mas percebi que ele tem uma segurança contra robôs que enviam *spam*, então eu utilizei um servidor de email de uma hospedagem minha para que funcionasse adequadamente.

Os posts do fórum ajudaram bastante na hora de começar a implementar a parte de envio de emails.

### Documentação de programação

Procurei documentar as minhas implementações diretamente no código usando comentários no estilo *JavaDocs*.