Instituto de Matemática e Estatística - USP
MAC0218 - Técnicas de Programação II

MONOLITO

Bruno Carneiro Cunha - NUSP
Daniel da Silva Nunes - NUSP 10297612
Eduardo Rocha Laurentino - NUSP 8988212
Pedro Henrique Barbosa de Almeida - NUSP 10258793

Estrutura completa

Nesta fase, focamos na implementação da base de dados e dos métodos presentes nas classes de controle que gerenciarão o fluxo de informações nas diversas funcionalidades do sistema. Disponibilizamos alguns recursos, como a possibilidade de realizar upload de arquivo de imagem no momento da submissão de um novo projeto, e removemos outros como a dashboard, página do site anteriormente descrita.

Como indicado no enunciado desta entrega do projeto, os testes de unidade foram realizados em diversos casos envolvendo as entidades especificadas no modelo. Com eles, aprimoramos a corretude das funcionalidades, corrigindo eventuais problemas de validação e inconsistências menores nas relações entre as classes e os métodos anteriormente implementados.

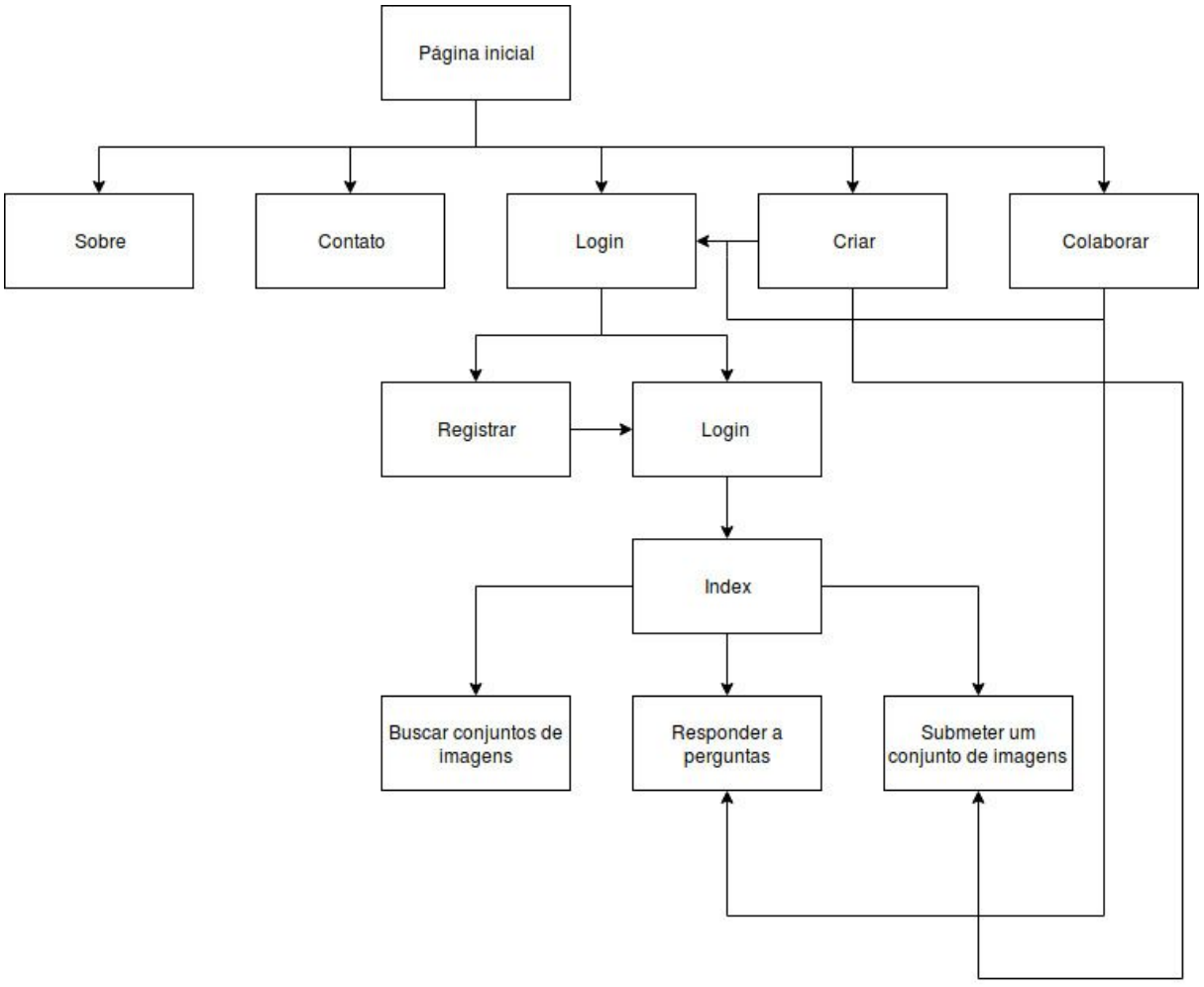
Além disso, para cobrir alterações feitas em sessões como a dashboard, alguns aspectos da parte visual do site foram alterados.

Por fim, funcionalidades como o cálculo de estatísticas referentes às respostas dadas pelos usuários enquanto colaboram para determinado projeto ainda não estão implementadas, por possuírem maior dependência da base de dados.

Documentação da Arquitetura

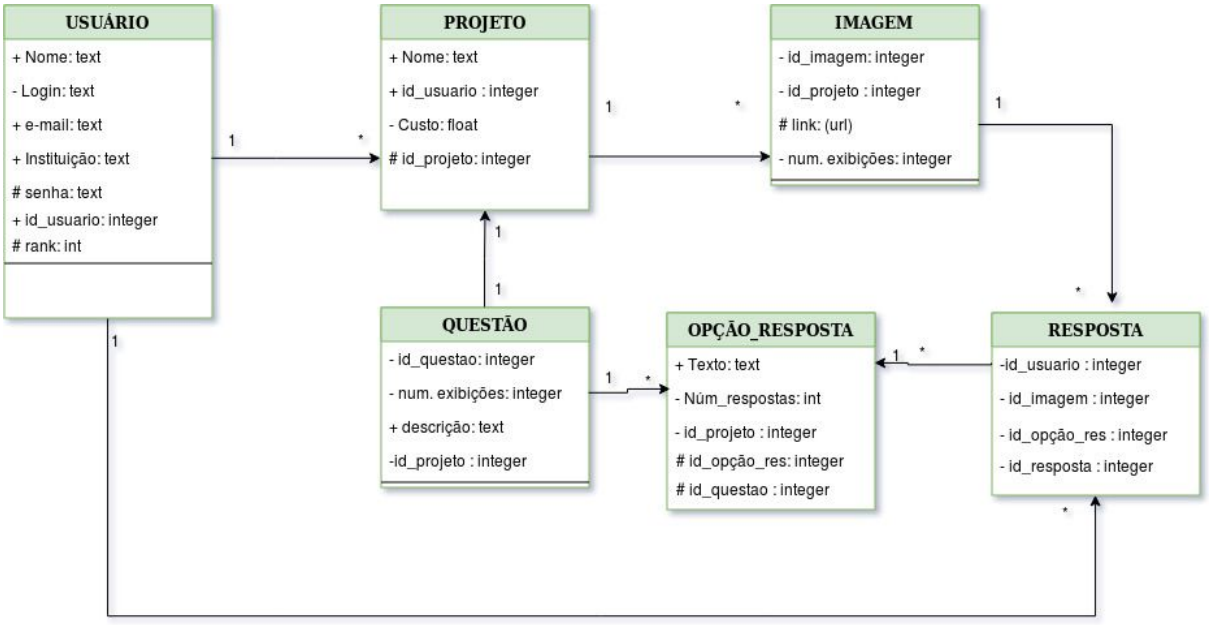
Fluxo do site

Alguns aspectos do fluxo do site foram alterados nesta fase. No fluxo inicial, o usuário era direcionado após o login a uma sessão que chamamos de dashboard, onde seriam encontrados projetos aos quais ele poderia contribuir. No entanto, o grupo chegou à conclusão de que seria mais interessante remover a dashboard e em seu lugar exibir uma página onde o usuário logado escolhe se vai rotular imagens ou criar um novo projeto que será submetido a rotulações de outros usuários. Tal página foi chamada de *index*, e para o usuário logado, ela é exibida com uma breve apresentação e os botões direcionam para a página de login.



Estrutura do banco de dados

A estrutura do banco de dados foi implementada com as devidas associações entre entidades. Chaves estrangeiras foram criadas e modificamos alguns campos de tabelas em relação à fase anterior.

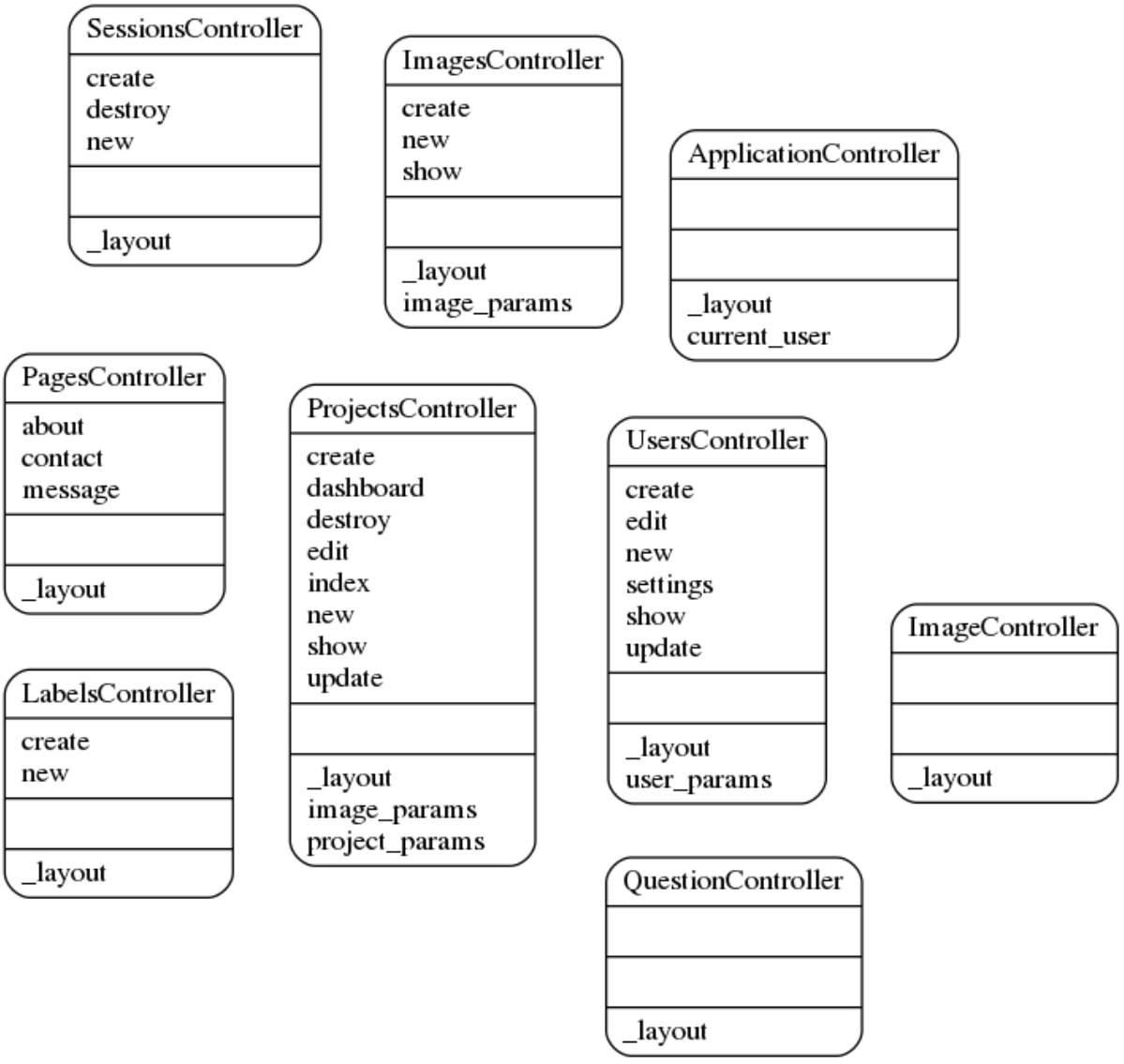


A arquitetura acima implementada, somada ao uso de queries nas classes de controle, mostra-se suficiente para atender as funcionalidades pretendidas no sistema. No entanto, ainda está sujeita à eventuais modificações caso estas se mostrem necessárias.

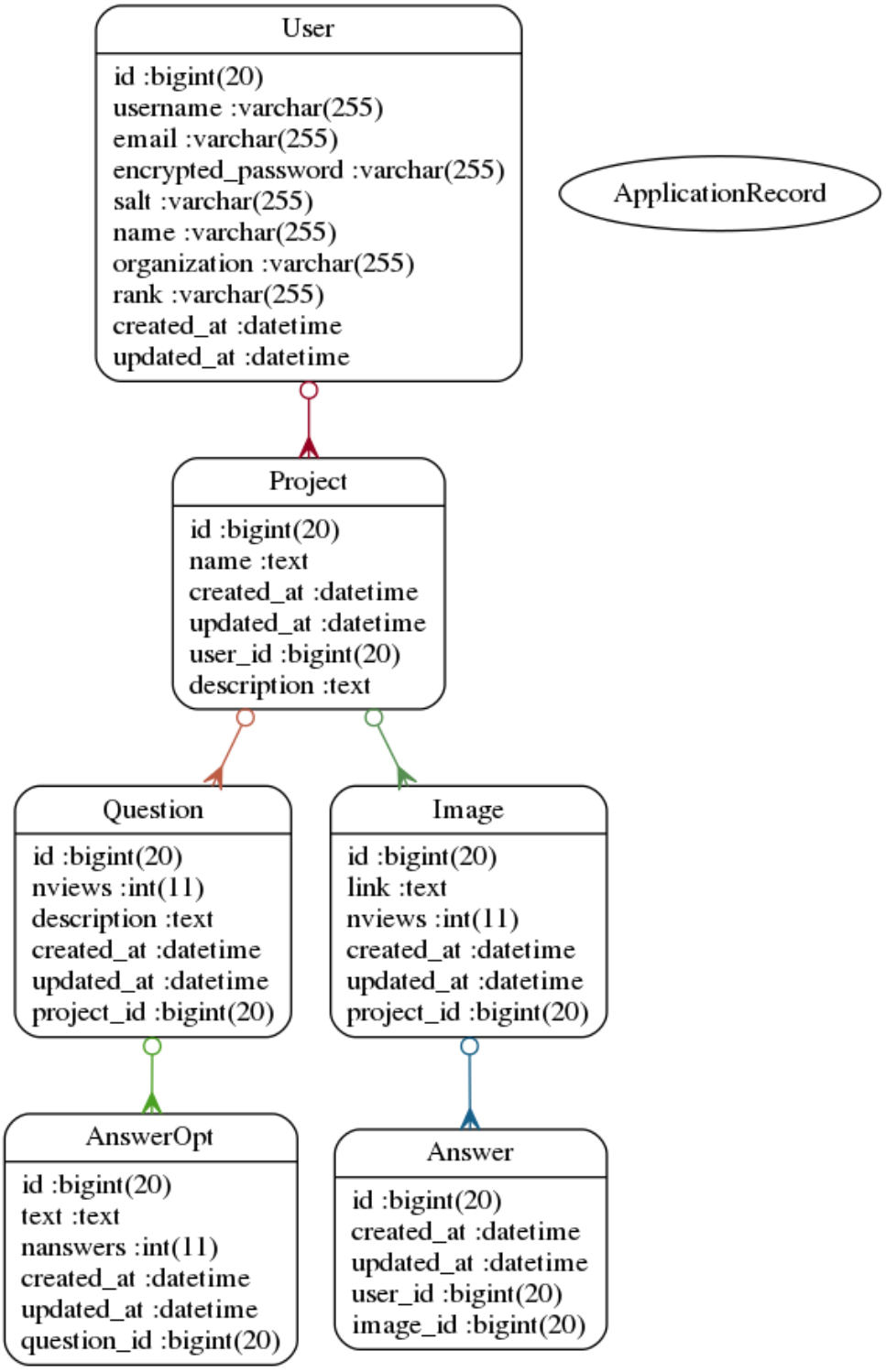
Como na fase anterior, para indicar em que pé a arquitetura do sistema está, em termo de classes, models e banco de dados, a cada fase, utilizamos duas ferramentas: **Rails ERD** (“is a gem that allows you to easily generate a diagram based on your application's Active Record models”, ver https://github.com/voormedia/rails-erd) e **Rails Roady** (“generates Rails 3/4/5 model (ActiveRecord, Mongoid, Datamapper) and controller UML diagrams as cross-platform .svg files, as well as in the DOT language.”, ver https://github.com/preston/railroady).

Na fase atual, assim como na anterior, tais ferramentas nos fornecem às seguintes informações sobre o estágio atual do sistema (todos os arquivos originais estão na pasta docs do projeto):

Controllers



Modelos


User
id :bigint(20)
username :varchar(255)
email :varchar(255)
encrypted_password :varchar(255)
salt :varchar(255)
name :varchar(255)
organization :varchar(255)
rank :varchar(255)
created_at :datetime
updated_at :datetime
ApplicationRecord
Project
id :bigint(20)
name :text
created_at :datetime
updated_at :datetime
user_id :bigint(20)
description :text
Question
id :bigint(20)
nviews :int(11)
description :text
created_at :datetime
updated_at :datetime
project_id :bigint(20)
Image
id :bigint(20)
link :text
nviews :int(11)
created_at :datetime
updated_at :datetime
project_id :bigint(20)
AnswerOpt
id :bigint(20)
text :text
nanswers :int(11)
created_at :datetime
updated_at :datetime
question_id :bigint(20)
Answer
id :bigint(20)
created_at :datetime
updated_at :datetime
user_id :bigint(20)
image_id :bigint(20)

O que foi feito

Nossa aplicação agora tem a base de dados implementada e suas funcionalidades bem definidas e testadas.

- O banco de dados foi implementado com as devidas associações entre entidades feitas.
- O site já pode receber imagens quando um novo projeto está sendo submetido. As imagens são armazenadas com Active Storage.
- Testes para as diversas funcionalidades foram realizados e possíveis erros contornados. (Ver mais no relatório de testes).

Dificuldades encontradas

O primeiro passo foi estruturar o banco de dados de acordo com o que havíamos indicado na fase anterior, e aqui as dificuldades estiveram no campo de estabelecer as relações entre as entidades, isto é, criar migrações que definissem chaves estrangeiras corretamente no banco de dados utilizado. Além disso, o planejamento esquemático da base de dados, antecipando funcionalidades a serem implementadas futuramente também demandou tempo adicional.

Outro problema com o qual tivemos que lidar foi o gerenciamento de diferentes versões de gemas e do próprio Rails utilizado pelos membros do grupo, o que, inicialmente, era fonte de conflitos e de erros locais. Posteriormente, tomamos maiores precauções com o Gemfile e conseguimos contornar esse problema.

Em continuidade, foi bastante desafiador entender o mecanismo de ActiveStorage (que é uma funcionalidade integrada ao Rails desde a versão 5.2. Trata-se de um recurso central para os nossos propósitos e com muita consulta conseguimos implementar uma versão inicial no nosso programa, como dito anteriormente, possibilitando já a criação de novos projetos com arquivos (de imagens) associados, que ficam armazenados localmente. O interessante é que o ActiveStorage é bastante poderoso, e permite que as imagens sejam adquiridas e mantidas em nuvem, sem necessidade de armazenamento local, o que é bastante interessante para a nossa proposta.

Além disso, entender o mecanismo de testes automatizados ao mesmo tempo que requereu bastante esforço para criar testes que cobrisse uma quantidade razoável das nossas funcionalidades, permitiu também que o trabalho daqui pra frente se torne mais agilizado, por permitir constante monitoração das interferências do que está sendo feito com relação ao que já estava implementado.

Próximos passos

Para a próxima fase, que é a entrega final, definimos que trabalharemos para que o site tenha todas as funcionalidades implementadas, como o sistema de recompensas ao colaborar com um projeto e disponibilidade de estatísticas a partir de respostas dadas.

- O site mostrará uma imagem por vez junto com as opções de rótulo para o usuário, já simulando a interação pretendida para classificação do conteúdo;
- Calculará as estatísticas a partir da interação de diferentes usuários com as imagens (isto é, as respostas fornecidas) e gerará relatório para o desenvolvedor;
- Consertar alguns bugs que permanecem no visual;

Relatório de testes

Os testes foram feitos utilizando RSpec. Ao todo, 27 exemplos são testados e todos os testes têm resultado válido, abrangendo amplamente as funcionalidades implementadas até então.

Para executar o testador e verificar os resultados, basta executar no diretório do projeto o seguinte comando:

bundle exec rspec

Os testes verificam se os formulários realizam as validações corretamente, evitando dados duplicados, incoerentes ou ausentes.