

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA

bigLib: sistema Web de compartilhamento de livros

MAC-0499: Trabalho de formatura

Claudivan Ribeiro N° USP: 5968788
Tiago Nicolosi Bomventi N° USP: 3690177
Orientador: Prof. André Fujita

SÃO PAULO
01/12/2011

Agradecimentos

Agradecemos ao Professor André Fujita por acreditar em nosso projeto, pelos incentivos e contribuições, inclusive para participar da Olimpíada USP de inovação, oportunidade em que pudemos moldar melhor o serviço e adicionar funcionalidades. Agradecemos também aos amigos e familiares que contribuíram com ideias, sugestões, críticas e elogios.

“As vezes você toma a iniciativa da mudança. As vezes a vida te obriga a isso. Em ambos os casos você progride.”

Autor desconhecido

“A verdade esta lá fora. Alguém sabe a URL?”

Autor desconhecido

Sumário

Resumo	4
1 Introdução	5
1.1 Problema e motivação	5
1.2 Solução proposta	5
1.3 Pesquisa de serviços semelhantes	7
1.3.1 TrocandoLivros - trocandolivros.com.br	7
1.3.2 LivraLivro - livralivro.com.br	7
1.3.3 Skoob - skoob.com.br	7
1.3.4 BookCrossing - bookcrossing.com	8
1.3.5 BookMooch - bookmooch.com	8
1.4 O nome e o logo	8
2 Funcionalidades e Requisitos	10
2.0.1 Requisitos funcionais	10
2.0.2 Requisitos não-funcionais	10
3 Análise do sistema	11
3.1 Modelagem dos dados	11
3.2 Descrição da arquitetura	15
4 Tecnologias envolvidas	19
4.1 Linguagens utilizadas	19
4.1.1 Python	19
4.1.2 JQuery (Javascript)	20
4.2 Escolha do arcabouço	20
4.2.1 Django	20
4.3 Bibliotecas e aplicações externas	22
4.3.1 Haystack (aplicação)	22
4.3.2 Tagging (aplicação)	22
4.3.3 WebBiblio (biblioteca)	23
5 Metodologia de desenvolvimento	25
6 Resultados obtidos - telas principais	26
7 Conclusões	31
8 Bibliografia	32

Resumo

O empréstimo de livros é uma prática muito utilizada. Isso em geral ocorre dado os interesses em comum em um determinado grupo de convivência e da confiança que depositam entre si a ponto de poder conceder por um período de tempo a utilização de um bem particular. Porém, nem sempre sabemos os livros que nossos amigos possuem. Outro empecilho ao empréstimo é a grande probabilidade de o livro emprestado não ser devolvido por esquecimento das pessoas. Ao observar estes comportamentos e identificar essas necessidades (conhecer livros dos amigos e não esquecer do livros emprestados), propomos facilitar o empréstimo de livros através de uma plataforma *web* que permite aos usuários conhecer os exemplares dos amigos e gerenciar seus empréstimos.

Palavras-chave: livros, compartilhamento, empréstimo, bibliotecas.

1 Introdução

1.1 Problema e motivação

O empréstimo de livros é uma prática amplamente difundida entre amigos¹. Uma pesquisa de 2008 realizada pelo IBOPE a pedido do Instituto Pró-Livro constatou que cerca de 46% dos leitores da classe C adotam esta estratégia. O mesmo índice era de 44% na classe B e de 35% mesmo na classe A, em que a aquisição é esperada, dada a condição financeira privilegiada deste público. Na época, a população de leitores no país era de 95,6 milhões de pessoas [4].

Em bibliotecas públicas e/ou universitárias faltam exemplares para atender toda a demanda, principalmente em períodos próximo a exames escolares, em que os livros de uma certa disciplina se esgotam. Além disso, possuem datas curtas de devolução, burocracia para cadastrar-se e para retirar do acervo.

Como alternativa à carência de livros para todos nestas instituições e por uma questão de comodidade, as pessoas já se organizam para promover o uso compartilhados destes bens. Empréstimo de um amigo em geral é mais rápido e fácil. É o tipo de interação que se beneficia inclusive dos interesses em comum entre o dono e o requisitante², pois estes podem trocar experiências, visões e percepções sobre o material.

A perda de livros em empréstimos é recorrente, mesmo entre amigos, pois nem todos conseguem lembrar que emprestaram nem que pegaram emprestado um material. Alguns leitores percebem a necessidade de manter um controle sobre estas operações e criam planilhas para isso. Mas mesmo nestes casos é comum esquecer de atualizar a planilha, o que torna este método bem intencionado, ineficaz.

Uma outra necessidade que há entre estas pessoas é de conhecer, *a priori*, a lista de livros de um certo amigo. Ou se existe algum amigo que possui uma dada obra. Obter estas respostas pode ser muito trabalhoso.

1.2 Solução proposta

Para facilitar e potencializar este costume propomos um sistema *web* de compartilhamento de bibliotecas pessoais entre amigos (ver ilustração na figura 1).

A ideia principal é que o usuário possa reproduzir nesta rede as interações que mantém no dia-a-dia no tocante a esta prática. Como é de se esperar, apenas os amigos poderão solicitar emprestado uma obra. Mas o dono poderá, a critério próprio, negar o pedido.

¹Amigos - Neste texto entende-se por amigos duas ou mais pessoas que mantêm um certo vínculo, seja ele profissional, familiar ou afetivo

²Requisitante - aquele que pega um livro emprestado

A vantagem de manter um *software* para controlar estas operações é a de que o próprio sistema gerenciará os empréstimos, as datas de devoluções, quem está com qual obra e por quanto tempo, não sendo mais necessários controles manuais suscetíveis a erros e esquecimentos. Também é uma plataforma em que os amantes de livros poderão compartilhar suas experiências de leituras de maneira mais ampla que em uma relação dois-a-dois. Além disso, as buscas por títulos serão feitas com rapidez e o problema do conhecimento *a priori* do acervo de um colega estará resolvido pois todo usuário terá uma página acessível para todos os seus amigos de sua lista de livros.

Os pré-requisitos para usar a plataforma serão os mínimos possíveis. As informações de cada-
 tros serão suficientes para que os usuários se identifiquem. Informações mínimas de localização (apenas cidade e estado) do leitor será pedido, por que espera-se que em uma rede de confiança como a que pretendemos implantar as pessoas ou sabem onde os pares residem, ou não precisam disso para receber o livro.

Outro recurso que acrescenta valor é a disponibilização de histórico de uso dos leitores. Apesar de ser uma rede de amigos, em um primeiro empréstimos não sabemos ainda se o requisitante, sendo este um colega de faculdade por exemplo, costuma devolver ou devolver no prazo ou cuidar bem do material. Dados como total de livros já emprestados e eventuais atrasos serão muito importantes no momento de o dono responder sim ou não a uma requisição.

Notificações de solicitações por *e-mail*, datas de devolução tanto para o dono quanto para o requisitante serão feitas automaticamente com o intuito de evitar atrasos, esquecimentos e aborrecimentos.

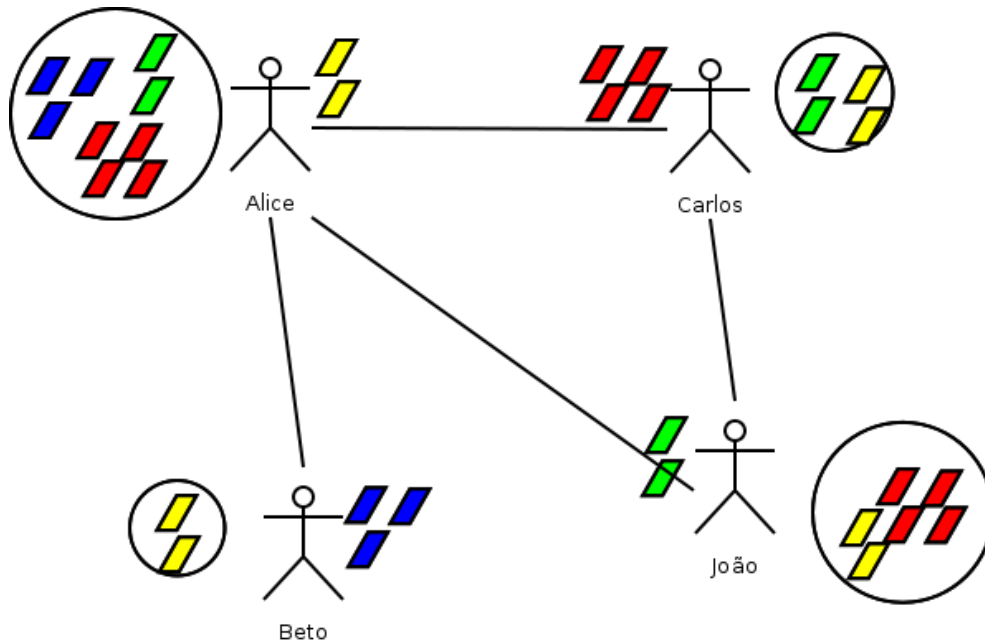


Figura 1: Solução proposta: a construção de bibliotecas pessoais através das relações de amizade dos usuários. Nessa figura, cada boneco representa um usuário do sistema, as setas representam relações de amizade, os retângulos coloridos são livros e os círculos as bibliotecas de cada usuário.

1.3 Pesquisa de serviços semelhantes

A partir da definição da idéia central do projeto, um sistema de compartilhamento de livros, buscamos na internet *sites* que oferecem serviços semelhantes, ou seja, que também utilizam o livro como objeto de troca, compartilhamento, ou apenas *tracking*³ de leitura. Os principais serviços encontrados nesta pesquisa estão listados abaixo:

1.3.1 TrocandoLivros - trocandolivros.com.br

É um *site* destinado a trocas, onde o usuário pode obter um perfil no sistema para cadastrar os exemplares que possui e que gostaria de “doar” para outras pessoas. Os livros ficam disponíveis para todos os perfis cadastrados. Quando um usuário conectado encontra um livro em que tem interesse, ele pode pedi-lo. O dono então envia o material pelo Correio, arcando com os custos. Feito isso, o dono recebe em troca um crédito, onde um crédito equivale a um livro no *site*, ou seja, ele tem direito a requisitar um livro para si após confirmada a troca. O lema *site* é “Não deixe os seus livros parados na estante”. O serviço é gratuito, o usuário só paga o envio postal do livro pelo Correio. Atraso no envio pode cancelar o pedido de troca.

Prós: exige que o usuário envie apenas livros em bom estado de conservação.

Contras: serviço estagnado. Já não é mais possível criar contas.

1.3.2 LivraLivro - livralivro.com.br

A idéia desse *site* é semelhante à do TrocandoLivros. LivraLivro visa estimular a troca de livros entre seus usuários usando a *web* como meio facilitador. O usuário deve se cadastrar no *site*, e disponibilizar para troca os livros que possui para esse fim. Um livro enviado, dá direito a um crédito no *site*. Um crédito equivale a um livro. O serviço é gratuito, o usuário só paga o envio postal do material pelo Correio.

Prós: permite que o usuário adicione uma descrição do estado de conservação do livro. A interface permite troca de mensagem entre os usuários. A possibilidade de criar uma lista de livros desejados e ser notificado por *e-mail* quando eles estiverem disponíveis.

Contras: cadastro burocrático. Dificuldade nas buscas.

1.3.3 Skoob - skoob.com.br

O *site* é uma rede social para leitores. O usuário se cadastra e recebe um perfil onde pode adicionar os livros que está lendo, já leu ou vai ler, criando dessa maneira uma estante virtual. Os usuários também podem incluir avaliações e opiniões sobre os livros, construindo um sistema de avaliação colaborativo dos mesmos.

³Tracking: do inglês, nesse contexto, indica que o dono do livro, após doá-lo, pode acompanhar os locais por onde o livro passa ao trocar de leitor

Prós: implanta a ideia de gamificação⁴ no serviço, com usuários ganhando títulos conforme a quantidade de livros que lê. Isso cria uma atmosfera saudável de competição. Permite acesso ao *site* utilizando a conta do Facebook.

Contras: não possui gerenciamento de empréstimos.

1.3.4 BookCrossing - bookcrossing.com

responsável pelo surgimento da ideia de BookCrossing [7], um movimento que cresce em todo o mundo e incentiva a prática de deixar o livro em algum lugar público para que outras pessoas encontrem-no, leiam e depois façam o mesmo. O objetivo dos membros do BookCrossing é “transformar o mundo inteiro em uma biblioteca”. O conceito básico do BookCrossing se baseia em “Ler, Registrar e Libertar”. O livro deixado em um lugar público por seu dono, ao ser encontrado, pode ser registrado no *site*. Sendo assim, o livro pode ser “rastreado” e o seu antigo dono pode acompanhar os lugares por onde o seu livro passou.

Prós: o BookCrossing é uma forma diferente de compartilhar livros, estimulando a leitura e o acompanhamento.

Contras: não possui gerenciamento de empréstimos.

1.3.5 BookMooch - bookmooch.com

Rede com usuários de todo o mundo e com uma versão em português. Seu funcionamento é semelhante ao LivroLivro e TrocandoLivros, pois também utiliza um sistema de pontos de crédito por trocas mas com uma diferença: trocas internacionais valem três pontos de crédito.

Prós: o usuário pode realizar trocas de livros internacionais. Ganha um décimo de crédito apenas por disponibilizar um livro. Possui versão para *smartphones*.

Contras: não possui gerenciamento de empréstimos.

O único site que se propõe a prestar o mesmo serviço que a bigLib (gerenciamento de empréstimo) é o inglês <http://bookhopper.co.uk>. Porém só atende a usuários na Grã-Bretanha.

1.4 O nome e o logo

Os requisitos eram de um nome curto, mnemônico, que descrevesse o serviço instantaneamente. Pensamos nomes em português como nossoLivro, nossaBiblioteca, meuLivro. Por serem palavras compridas ou por não terem domínio *web* disponível, as desconsideramos. A palavra meLivro por algum momento chegou a ser cogitada mas apesar do bom trocadilho, passava a ideia de se desfazer do material, um nome mais adaptável a serviços semelhantes como o BookCrossing, mas contrário à proposta original.

⁴Gamificação: é um conceito recente, de transformar a vida das pessoas em jogo. Sistemas *web* tem explorado essa idéia, dando prêmios simbólicos e posições de destaque para os usuários que mais usam seus serviços

Para reavivar a noção, até certo ponto utópica, de um lugar em que estivessem todos os livros do mundo, formados justamente pela concatenação da biblioteca privada de cada indivíduo, pensamos em “bigLib”. Um nome curto, que pode ser entendido em muitas culturas, e que apesar de não ser entendido por todo brasileiro, é facilmente pronunciável.

Também pensando na representatividade do serviço, na ideia de totalidade, e considerando principalmente as limitações artísticas dos membros, produzimos um logo.



Figura 2: Logotipo da bigLib

2 Funcionalidades e Requisitos

Partimos do princípio de que os potenciais usuários do site seriam na maioria estudantes, jovens com habilidades básicas de digitação e navegação na *internet*.

2.0.1 Requisitos funcionais

- Convidar um amigo para participar da rede
- Procurar por pessoas conhecidas
- Adicionar amigos
- Procurar por livros
- Adicionar livros
- Cadastrar livros
- Solicitar empréstimo
- Histórico de cada usuário.
- Devolver livro
- Consultar lista de livros de um dado amigo
- Interface simples

2.0.2 Requisitos não-funcionais

- Usabilidade. Não deve ser um sistema de difícil aprendizado. Boa navegação.
- Confiabilidade. Espera-se que não ocorram falhas no sistema. Deverá existir alguma redundância de armazenamento (*backup*).
- Desempenho. O sistema deve funcionar de maneira conveniente ao usuário, ou seja, não pode ficar sobrecarregado gerando lentidão nas respostas.
- Sistema robusto e confiável para garantir que os empréstimos dos livros não se percam.
- Sendo um sistema *web*, onde os usuários se cadastram fornecendo algumas informações pessoais, tais como *e-mail*, espera-se que o sistema não disponibilize de maneira indevida alguns dados dos usuários ainda que estejamos coletando informações mínimas.

Algumas regras de negócio relevantes

- Todo usuário deve se cadastrar com um *e-mail* válido e criar um *login* para usar o sistema.
- Para um usuário pedir um livro emprestado ele deve ter pelo menos um livro cadastrado.
- Cada usuário pode pedir emprestado apenas livros de seus amigos.
- Atrasos de devolução de livros devem ser notificados de alguma forma aos amigos.
- Cada leitor possui uma nota quantitativa de credibilidade.

3 Análise do sistema

Após o levantamento dos principais requisitos, os integrantes do projeto se reuniram para definir, de forma qualitativa, o sistema a ser desenvolvido.

3.1 Modelagem dos dados

O único tipo de usuário no sistema é o leitor. Seus dados mínimos requeridos são nome, sobrenome, *e-mail*, gênero, usuário e senha. O primeiro nome será utilizado quando a identificação precisar de um nome curto, como embaixo da foto de rosto. O endereço de *e-mail*, que será verificado, servirá para enviar notificações correntes como pedido de empréstimo e devoluções. O nome de usuário, além de servir na autenticação juntamente com a senha, tem o papel de criar uma página disponível publicamente em www.biglib.com.br/USUARIO constando a lista de livros que aquele leitor disponibiliza para empréstimo aos amigos. A partir desta página, um visitante pode optar por juntar-se à rede para solicitar alguma obra.

A verificação do *e-mail* fornecido durante o cadastro permitirá a ativação da conta. Uma chave única será gerada, guardada em uma tabela secundária no banco e enviada junto com um *link* ao *e-mail* fornecido. A chave terá um período de validade de sete dias. Se o leitor não ativar sua conta neste período, um *script* que executará em intervalos de tempo a ser determinado apagará tanto a chave quanto o usuário inativo gerado. Isso evita que nomes de usuários sejam inutilizados, já que estes são únicos.

Apenas com a finalidade de conhecer melhor o público da rede, outros atributos facultativos como data de nascimento e profissão serão solicitados. Espera-se que os usuários sejam em sua maioria estudantes (secundaristas e principalmente universitários), dado que a pesquisa IBOPE [4] constatou que metade dos leitores do país são estudantes, e que leem principalmente obras indicadas pela escola/universidade. Um relacionamento importante que surge nesta modelagem é o de simetria na amizade. Se Alice é amiga de Beto, Beto também é amigo de Alice. Ao contrário de relacionamentos unilaterais, como o de seguir (*follow*) em redes como Twitter, a representação de amizade deve ter necessariamente a informação representada duas vezes no banco. Que Alice é amiga de Beto e que Beto é amigo de Alice.

Na modelagem do livro, ao invés de cada leitor que possui, por exemplo, Dom Casmurro de Machado de Assis, cadastrar título, autor e editora, temos um único cadastro e os donos apenas fazem referência a ele para informar que possuem um exemplar. Esta solução, simples e óbvia, evita muita digitação, duplicação e imprecisão de dados.

Os atributos mínimos para um cadastro de livro são ISBN⁵ de 10 ou 13 dígitos, título, autor e editora. Apesar de estar bem distinta os dois ISBN no modelo, ao leitor não precisará fazer esta distinção. Sempre que possível a aplicação produzirá o ISBN restante de acordo com os algoritmos de conversão conhecidos [6]. A título de curiosidade, apenas não será possível a conversão de ISBN-13 para ISBN-10 quando o primeiro não começar por “978”. Também do

⁵ISBN: *International Standard Book Number*, é um número que identifica um livro unicamente

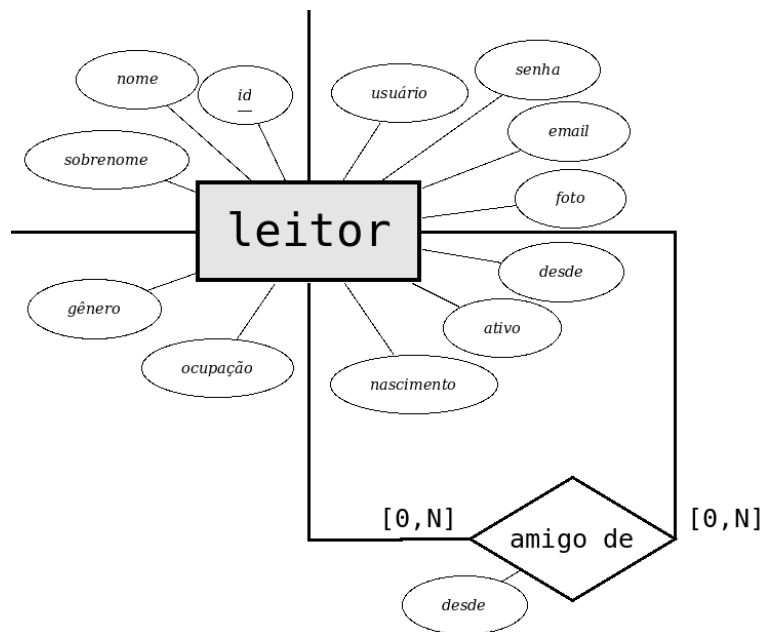


Figura 3: Modelo Leitor

lado da aplicação, semelhante ao que ocorre com o CPF, será feita a validação do número digitado com base no último dígito para se evitar erros de digitação.

Autores e editoras serão tratados como entidades. Assim, no futuro será possível conhecer quais os autores mais lidos ou as editoras mais populares em uma rede de amigos. Isso servirá para fazer recomendações de leituras.

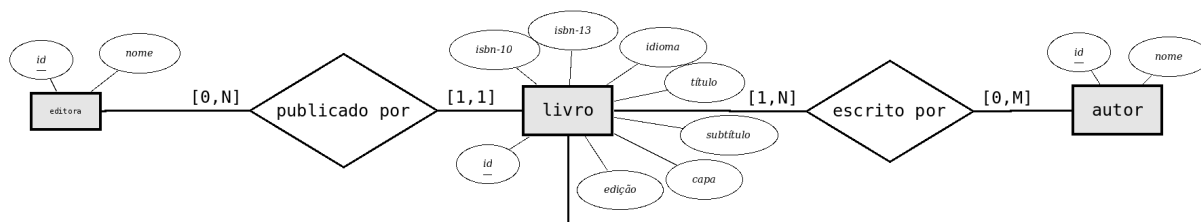


Figura 4: Modelo Livro/Autor/Editora

Já o exemplar, tido como uma instância da obra, guarda apenas quem é o dono, seu estado de conservação, quantidade e data de adição à rede. Também por uma questão de facilidade de uso, se a obra já estiver cadastrada no sistema, adicioná-la à lista particular de livros será feita com apenas um clique. O estado de conservação é um atributo que ajudará o leitor a escolher entre dois exemplares de uma mesma obra pertencentes a amigos diferentes. Há uma lista de possibilidades: novo, bem conservado, conservado e velho. O padrão será “conservado”.

Em um segundo momento, depois de ter adicionado, será possível editar estas informações. Cada leitor poderá organizar sua lista de livros atribuindo *tags* aos mesmos. Isso cria grupos de obras que facilita a procura e melhora a navegação pelos livros. Por exemplo, Beto poderia

atribuir a *tag* “BCC” a todos os livros usados durante o curso de Computação que faz na Universidade de São Paulo. Assim Alice poderá, ao acessar www.biglib.com.br/beto, visualizar o material que Beto utilizou/utiliza e que estaria disposto a emprestar. Para demonstrar mais uma possibilidade de uso, Beto poderia ser mais específico agrupando livros usados em cada semestre da faculdade, o que torna sua ordenação mais relevante aos amigos da faculdade.

Em um texto livre, o dono poderá recomendar a obra ou fazer um resumo. Campo facultativo.

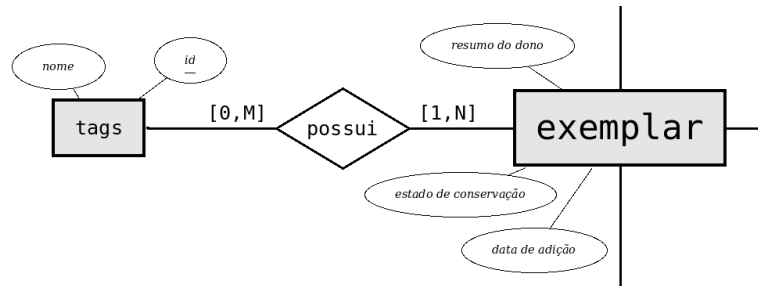


Figura 5: Modelo Exemplar

Por fim e não menos importante está a modelagem dos dados para o empréstimo propriamente. Precisamos de um protocolo que represente o mais fielmente possível a posse dos materiais. Ou seja, se o livro de Alice estiver com Beto, esta informação deve estar no sistema e tão logo Alice recupere o material, a atualização deve ser feita no *site*.

Então, para requisitar um exemplar, um usuário autenticado clica no mesmo e, se o requisitante for amigo do dono e a obra já não estiver emprestada, a pessoa escreve um texto facultativo de pedido e escolhe um período em que ficará com o livro. Um *e-mail* é disparado ao dono informando de que um amigo fez uma requisição com um *link* para que, depois de autenticado, possa responder. A resposta pode ser “aceitar” (por padrão) ou “negar” o empréstimo. Em ambos os casos, haverá um texto livre de resposta. Novamente uma notificação é enviada via *e-mail* para o requisitante informando que o primeiro respondeu, convidando-o a acessar um *link* para consultar a resposta. Se a resposta for negativa, a iteração cessa. O que pode ser feito neste caso é um novo pedido, com período para devolução mais curto se o período tiver sido o motivo da negação. Continuando, se a resposta for positiva, então o sistema já registra o exemplar como de posse do requisitante e registrando-o como indisponível para pedidos. A troca real é realizada entre as partes e independente do sistema.

A partir disso, a qualquer momento o requisitante pode informar a devolução. Esta é a parte mais crítica do protocolo, quando podem haver conflitos. Para que o sistema registre a devolução, o dono deve confirmar o recebimento. Se este não o fizer, a obra continua emprestada para o amigo, mesmo que atrasada. Veja uma representação das trocas na figura 7.

Complementando, após a devolução ser feita com sucesso, ficará disponível por sete dias a possibilidade de o requisitante avaliar o livro que leu/consultou. Tal avaliação consistirá de uma

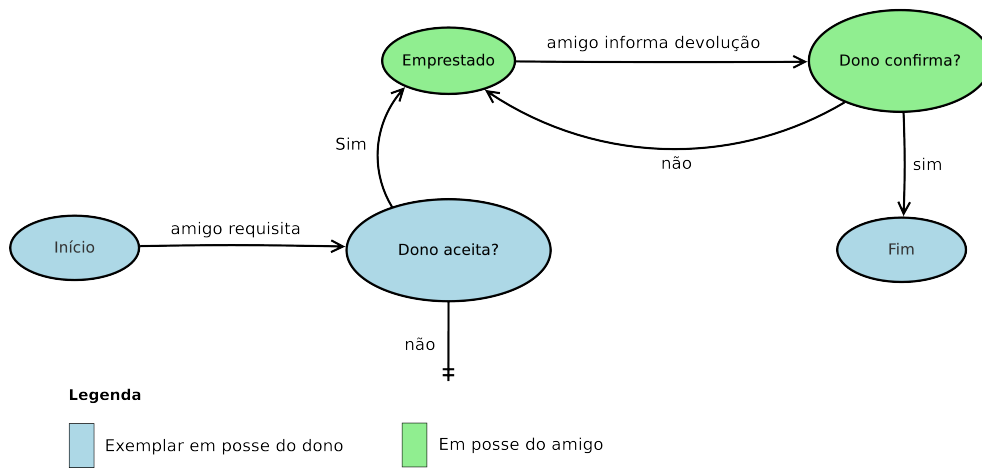


Figura 6: Protocolo de empréstimo

escolha entre recomenda/não recomenda acompanhada de um texto livre, facultativo. A figura 7 complementa a descrição.

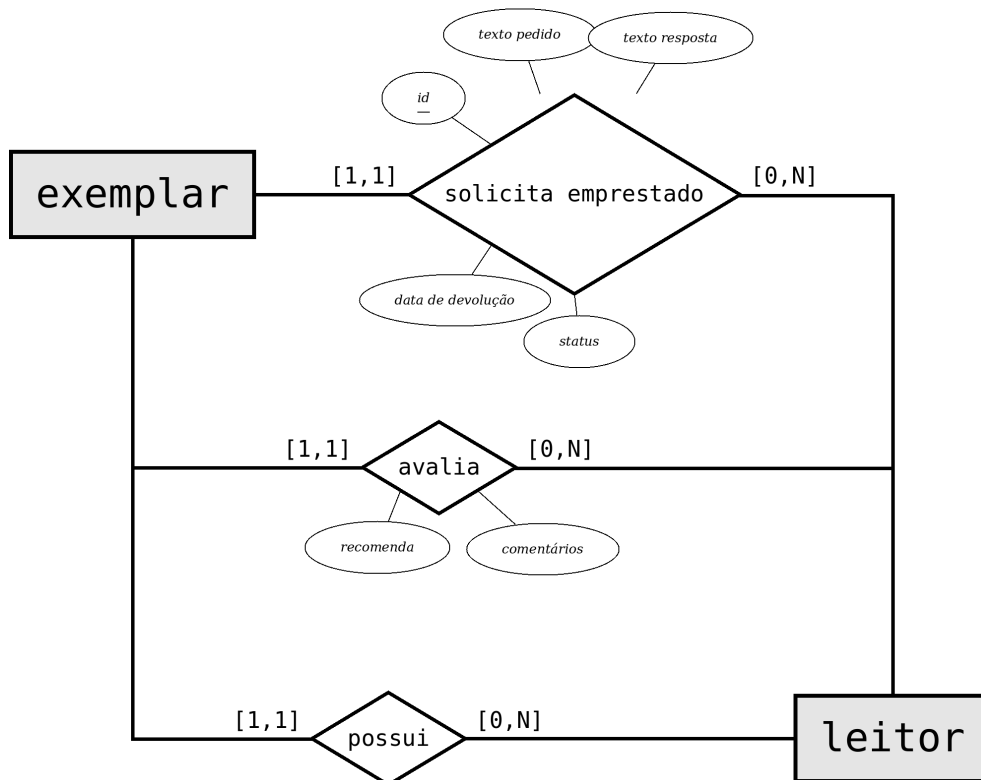


Figura 7: Diagrama Entidade Relacionamento do empréstimo

3.2 Descrição da arquitetura

A definição da arquitetura do sistema está diretamente ligada à escolha do arcabouço de desenvolvimento para implementá-lo. Optamos pelo Django, que assim como outros arcabouços ágeis para desenvolvimento *web*, utiliza o padrão MVC para o desenvolvimento de aplicações. No padrão MVC, podemos dizer que o sistema é separado em três partes, entrada (*Model*), saída (*View*) e processamento (*Controller*), onde cada uma destas partes, conhece o mínimo sobre as outras. Isso diminui o acoplamento⁶ entre as partes descritas, facilitando as modificações ao longo do desenvolvimento e manutenção.

Desenvolver um projeto em Django consiste em dividi-lo em pequenas aplicações, onde cada aplicação, pode ser entendida como um componente desse projeto. É importante que cada componente possua um conjunto bem definido de funcionalidades, lidando com tarefas específicas do sistema, como por exemplo: gerenciar usuários e permissões, gerenciar empréstimos de livros e gerenciar requisições de amizade. Aplicações implementadas dessa forma, maximizam a coesão⁷ do sistema, eliminando redundância e facilitam futuras modificações.

Conforme descrito acima, um sistema desenvolvido em Django, é composto basicamente por essas três camadas do padrão MVC. Mas essa descrição é por demais genérica, e não revela detalhes do funcionamento do sistema. Para entender como o projeto funciona, deve-se olhar para o que cada aplicação faz e como elas interagem entre si.

Sendo assim, após a modelagem dos dados, os integrantes do projeto deram início a implementação das aplicações. Cada aplicação tentou ser desenvolvida seguindo as idéias de coesão e acoplamento mencionadas, buscando ser o mais flexível possível.

Foram implementadas três aplicações, cada uma delas com a finalidade de resolver um problema específico do sistema. Para exemplificar o uso de cada aplicação, foram feitos diagramas de sequência, usando a terminologia do Django⁸ para classificar os objetos que interagem.

⁶Acoplamento é uma medida de quanto uma parte do sistema conhece da implementação de outras partes. Em sistemas fortemente acoplados, quando um componente é modificado, outras partes também devem ser

⁷Coesão é uma medida do quanto as responsabilidades de um componente compõe uma unidade do sistema. Um componente bem focado em um determinado aspecto do sistema, é um componente bem coeso

⁸Para separar a lógica de negócios da exibição dos dados, o arcabouço deixa toda a parte de HTML, JavaScript e CSS dentro dos templates e a parte de processamento dos dados, dentro de funções e *views*(funções que respondem às requisições do usuário)

Nome da aplicação: accounts

Tarefas: fazer login/logout, registrar usuários, recuperar *e-mails*, alterar informações cadastradas do usuário.

Funcionamento geral: A aplicação *accounts*, gerencia as tarefas de manipulação de informações dos usuários. Funciona em conjunto com o módulo *django.contrib.auth*, para o qual delega parte de seu trabalho. Para isso, ela deve formatar os campos corretamente para serem preenchidos pelos usuários e analisar se os dados estão de acordo com o modelo. Essa aplicação é responsável pela primeira página do *site*, onde o usuário faz a autenticação ou o cadastro.

Exemplo de comportamento

Como exemplo de um comportamento dessa aplicação, segue um diagrama de sequência do cadastro de um novo usuário na figura 8.

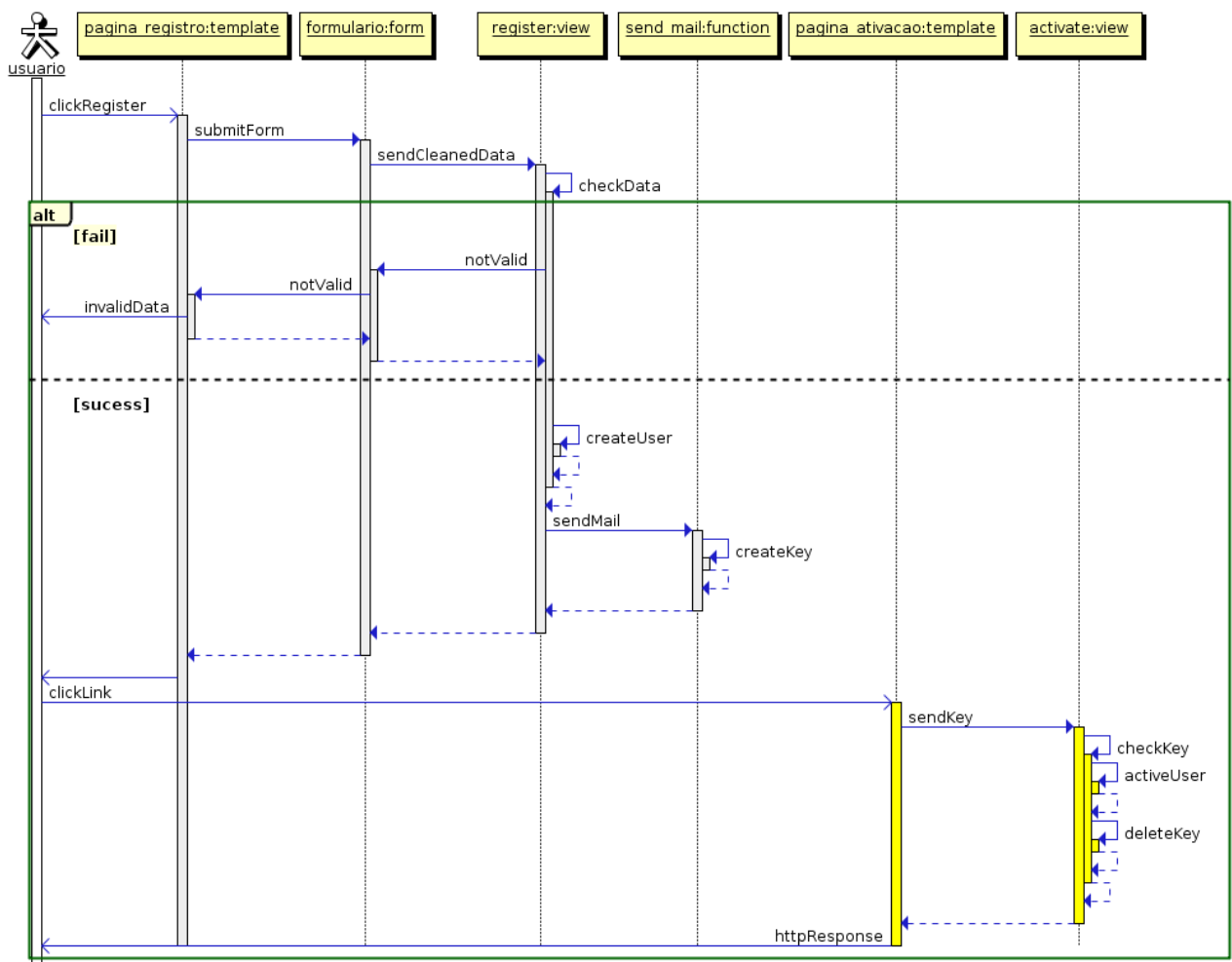


Figura 8: Diagrama de sequência do registro de um usuário

Nome da aplicação: bookLoan

Tarefas: gerenciar empréstimos de livros (solicitação, resposta, devolução e avaliação), cadastrar livros no sistema, gerenciar e visualizar a biblioteca de cada usuário.

Funcionamento geral: a aplicação bookloan é o núcleo do sistema. A maior parte de processamento de dados é delegada a este módulo, através de suas *views* e seus modelos. Ela utiliza uma aplicação *haystack*⁹ para realizar uma busca indexada dos livros na biblioteca de um usuário¹⁰. A maior parte dos dados do sistema estão dentro dessa aplicação. quando um usuário faz login, ela é responsável pela renderização de sua tela inicial.

Exemplo de comportamento

Como exemplo de um comportamento, segue um diagrama de sequência do pedido de empréstimo do livro de um amigo por um usuário na figura 9.

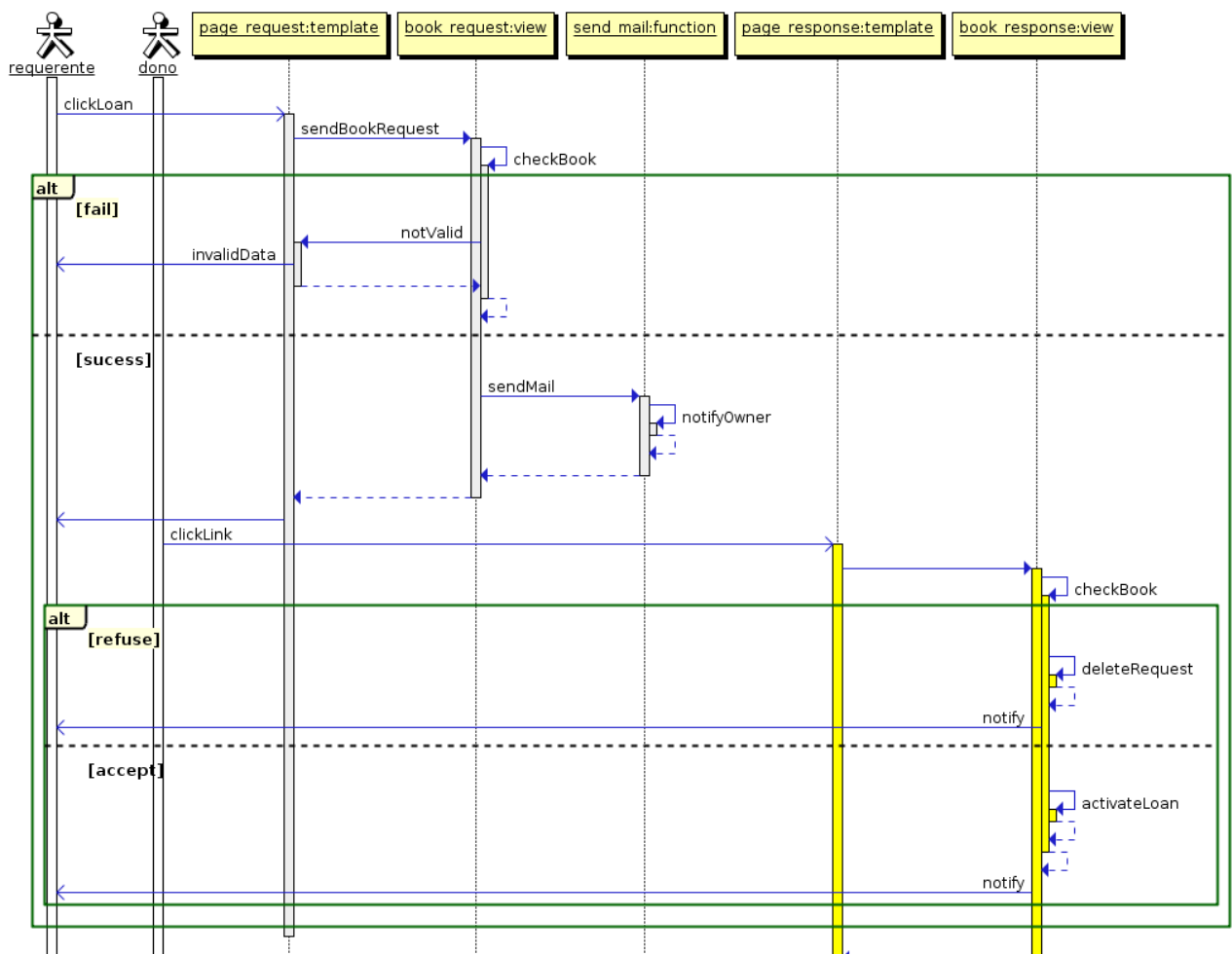


Figura 9: Diagrama de sequência do empréstimo de um livro

⁹ ver adiante a descrição de bibliotecas e aplicações externas

¹⁰a biblioteca de um usuário são todos os livros de todos seus amigos

Nome da aplicação: friendship

Tarefas: gerenciar requisições de amizade (solicitação e resposta) e gerar convites para usuários ainda não cadastrados no sistema.

Funcionamento geral: A aplicação *friendship* é responsável por gerenciar a tabela que representa as amizades no sistema. Essa aplicação pode ser acessada através da aplicação bookLoan, quando um usuário procura e adiciona um amigo.

Exemplo de comportamento

Como exemplo de um comportamento, segue um diagrama de sequência da requisição de amizade entre dois usuários cadastrados no sistema 10.

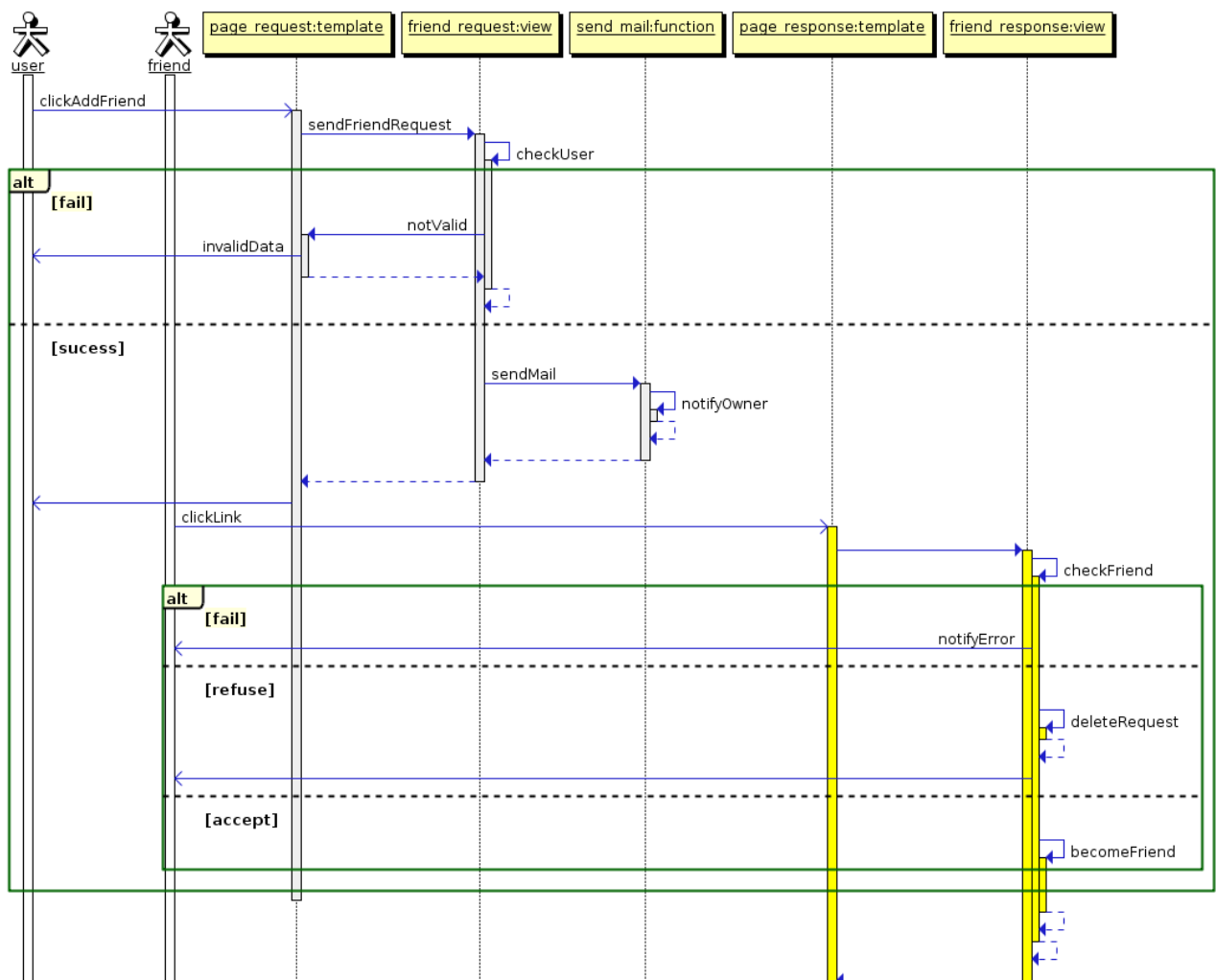


Figura 10: Diagrama de sequência de requisição de amizade

4 Tecnologias envolvidas

4.1 Linguagens utilizadas

Não muitos anos atrás, a maioria dos aplicativos de computadores, se destinava a alguns poucos usuários. Eram programas pequenos e monousuários. O hardware era limitado assim e conseqüentemente os sistemas operacionais também. Grande parte dos desenvolvedores utilizavam e dominavam linguagens de baixo nível¹¹. As linguagens de alto nível¹² eram pouco eficientes.

Com o surgimento da Internet, os aplicativos antes restritos a um pequeno número de usuários, passaram a serem utilizados por centenas e até milhares de pessoas simultaneamente. Com isso, a complexidade dos sistemas aumentou, mas a capacidade de processamento computacional também. Nesse contexto, as linguagens de alto nível evoluíram, se tornaram eficientes e atualmente são ferramentas utilizadas para garantir a flexibilidade e velocidade de sistemas cada vez mais complexos.

Neste contexto, as tecnologias aplicadas no desenvolvimento do projeto foram basicamente as linguagens interpretadas Python (no lado do servidor) e javascript (no lado do navegador). Assim foi utilizado um arcabouço que abstrai grande parte das tarefas repetitivas de aplicações *web* como criação e ativação de contas.

4.1.1 Python

É uma linguagem de alto nível, desenvolvida pelo holandês Guido van Rossum que buscava criar uma linguagem “para preencher o vazio entre a linguagem C e o Bourne Shell (shell script)”.

Algumas de suas principais características são:

interpretada: o código fonte é executado por um programa, o interpretador, antes de ser executado pelo sistema operacional ou processador. Com isso, novos códigos podem ser gerados enquanto o programa já esta em execução.

tipagem dinâmica: ou seja, o tipo das variáveis é inferido durante a execução do programa. Não é necessário declarar o tipo no código fonte.

fortemente tipada: não é permitido mudar implicitamente o tipo de uma variável, uma vez que este tenha sido determinado.

multi-paradigma: Possui suporte para os paradigmas: orientação a objetos, procedural e funcional.

multi-plataforma: Possui uma implementação de referência única para cada uma das grandes plataformas. Seu código funciona nos Sistema Operacionais Windows, Linux e iOS.

¹¹Linguagem de baixo nível é aquela que se utiliza de informações mais próximas ao código de máquina, como por exemplo, instruções do processador e manipulação de memória

¹²Linguagem de alto nível é uma linguagem com maior poder de abstração assemelhando-se mais a linguagem natural

Além disso, algumas decisões de design fazem de Python uma excelente linguagem. Algumas de suas vantagens para uso no projeto são:

gratuita e de código aberto: Pode ser baixada, estudada, alterada e distribuída sem nenhum custo adicional. É liberada sob uma licença compatível com a *GNU Public Licence* (GPL).

concisa: Enfatiza a simplicidade. Possui estruturas, por exemplo, compreensão de listas, que tornam a implementação de listas mais sucinta.

legível: É fácil entender o que o código “deveria fazer”. Por isso tem sido chamado de “pseudo-código executável”.

facilmente extensível: Vem por padrão com muitas bibliotecas. Outras podem ser obtidas gratuitamente na internet.

4.1.2 JQuery (Javascript)

Jquery é uma biblioteca javascript usada para agilizar o desenvolvimento de funcionalidades no lado do cliente (navegador) [1]. Possui compatibilidade com a maioria dos navegadores populares (Internet Explorer, Firefox, Chrome e Safari). Oferece suporte para *Cascade Style Sheets* (CSS) e *Asynchronous Javascript and XML* (Ajax). Foi usado para permitir ao usuário adicionar e devolver livros com apenas um clique. Isso torna a navegação mais rápida e prática, pois evita o recarregamento de toda a página.

A biblioteca também foi utilizada na manipulação da página para inserir e remover campos de autores, realizar consultas para autocompletar editoras e autores e na renderização de mensagens ao usuário.

4.2 Escolha do arcabouço

Devido à crescente complexidade dos sistemas computacionais, é cada vez mais necessário utilizar ferramentas que permitam ao desenvolvedor evitar a repetição de funcionalidades e promover o reuso de código¹³.

Arcabouços são conjuntos de bibliotecas que promovem o reuso de código, são basicamente padrões e componentes [5]. A utilização de um arcabouço pode acelerar muito o desenvolvimento de um projeto.

Uma desvantagem é o tempo gasto estudando e aprendendo o arcabouço.

4.2.1 Django

Como já descrito, Django é um arcabouço de desenvolvimento *web*, escrito em Python. Foi criado pelo grupo editorial “The World Company” para escrever a versão *web* de seus jornais. Em 2005, foi liberado sob a licença *Berkley Software Distribution* (BSD).

¹³afinal de contas, se não fosse o reuso, estaríamos sempre a reinventar a roda!

A principal meta do Django é facilitar a criação de sistemas *web* baseados em banco de dados [2]. Ele enfatiza a reusabilidade de seus componentes e segue o princípio (*Don't Repeat Yourself*) DRY. Este é um princípio fundamental de desenvolvimento de *software*, que visa reduzir a repetição da informação em todos os níveis de um sistema. Quando a aplicação desse princípio é bem sucedida, alterações em um elemento do sistema, não requerem outras alterações em outros elementos.

Django também é baseado no padrão *Model View Controller* (MVC)¹⁴.

MVC é um padrão de arquitetura de *software* que visa a separar a lógica de negócio da lógica de apresentação, permitindo o desenvolvimento, teste e manutenção isolados de ambas[8]. Seus componentes são: o modelo (*model*), responsável por representar a modelagem do sistema; a visão (*view*), responsável por apresentar visualmente os dados requeridos (é importante destacar que não é função da visão realizar nenhum tipo de processamento dos dados) e o controlador (*controller*), um componente entre o modelo e a visão, que recebe as entradas, analisa, processa e devolve uma visão para o usuário.

A principal utilidade do padrão MVC é permitir a diferentes desenvolvedores trabalhar em camadas separadas do sistema de maneira independente, ou seja, alterações no layout não alteram em nada a modelagem dos dados e vice-versa [3].

A implementação do padrão MVC, no Django, consiste em um grupo de componentes que exercem as funções dessas camadas:

A camada Modelo: primeiramente, o coração do arcabouço Django é sua camada *Object Relational Mapper* (ORM), ela é responsável por permitir que o usuário utilize os objetos em Python sem se preocupar com a persistência desses dados no banco de dados. Praticamente nenhum código SQL é necessário (a não ser consultas complexas). O usuário define o seu modelo de dados (*Model*) através de classes e métodos Python e essas classes são mapeadas para um banco de dados relacional através da ORM.

A camada Visão: o Django fornece uma linguagem própria para a criação de templates (podendo utilizar herança para evitar repetição de código) que pode ser usada na geração de páginas dinâmicas.

A camada Controladora: para acessar os dados do Modelo e chamar a Visão correta para visualizar esses dados, o Django utiliza dois componentes, o *URL dispatcher* e as *Views*. O *URL dispatcher* realiza um mapeamento da URL fornecida como entrada ao sistema e através do uso de expressões regulares direciona o fluxo de execução para uma *view*, ou função, que irá cuidar do acesso e manipulação dos dados do Modelo.

¹⁴É possível também encontrar a sigla MTV *Model Template View*, que representa basicamente a mesma coisa, dita com palavras diferentes

Além desses componentes, o Django possui outros, que apesar de não estarem ligados a arquitetura do arcabouço, também merecem ser citados pela importância na implementação do projeto:

Sistema de administração: Interface para gerenciar os objetos do modelo de dados da aplicação.

Essa interface é gerada automaticamente e poupa o desenvolvedor da implementação das operações (CRUD)¹⁵

Formulários: a partir do modelo de dados, os formulários são gerados automaticamente. Além disso, o Django utiliza um padrão de manipulação desses formulários que facilita trabalhar com os dados enviados por eles.

Segurança: essa componente cuida do gerenciamento de autenticação de usuários e controle de permissões.

4.3 Bibliotecas e aplicações externas

4.3.1 Haystack (aplicação)

É uma aplicação que implementa a busca indexada¹⁶ de maneira desacoplada dos principais *backends*¹⁷ disponíveis para Django, entre os quais: Solr, Xapian e Whoosh.

É focada em conceitos de reusabilidade. Através do Haystack, pode-se escrever um código para fazer buscas com o Whoosh e futuramente, se desejar, alterar o *backend* para Solr modificando minimamente o código do seu projeto.

A barra de busca no canto superior esquerdo (ver Figura 13) utiliza o Haystack para realizar a busca tanto por amigos como por livros. As buscas procuram por padrões tanto no início como no meio delas. Por exemplo: buscar por “ana” pode casar com “analisar”, “bananas” ou “juliana”.

4.3.2 Tagging (aplicação)

Basicamente, uma aplicação que permite gerenciamento (inserção, remoção e atualização) de *tags*, ou palavras-chave, para quaisquer entidades do banco de dados, neste caso exemplares de livros.

No sistema desenvolvido, a maneira encontrada para permitir ao usuário classificar os livros da maneira mais flexível possível, ou seja, não categorizá-los previamente e permitir alterar sua classificação, foi utilizar estas palavras-chave.

¹⁵Operação básicas em um sistema: *Create*-criar objeto, *Read*-listar objetos, *Update*-atualizar objeto e *Delete*-deletar objeto

¹⁶Busca indexada é realizada em um índice invertido de documentos. É um índice que dada uma busca de palavras, devolve os documentos nos quais as palavras estão presentes. Isso facilita a recuperação de informação (texto) fazendo-a de maneira rápida e acurada

¹⁷O termo *backend* designa uma camada da aplicação que serve de suporte para executar serviços indiretos pelo usuário

Pelo que foi implementado, o leitor pode aplicar *tags* aos seus livros, procurar e navegar através destas no conjunto de livros que pode pegar emprestado. Esse recurso facilita o agrupamento de livros com as mesmas *tags* e futuramente pode servir como uma métrica de distância a ser utilizada para calcular a similaridade entre livros e projetar um sistema de recomendação para o usuário.

4.3.3 WebBiblio (biblioteca)

Um dos maiores desafios da bigLib, algo que definirá se as pessoas usarão a ferramenta largamente ou se os usuários se restringirão a poucos é o de tornar fácil o cadastro de livros. Um cadastro dificultoso seria a maior barreira de entrada, dado que o registro de usuário foi feito para ser o mais simples e direto.

Pensando em um cenário ideal, o leitor logo após se autenticar procura cada exemplar que possui e os encontra já cadastrados. Então, com apenas um clique adiciona à sua lista. Porém, para que isso ocorra é preciso que a obra já tenha sido catalogada por outro usuário. Em um cenário pessimista, e mais recorrente pelo menos no início, o leitor teria que cadastrar todas as informações de cada um de seus livros.

Para dar uma solução a isso, começou-se pela forma com que seriam preenchidos os autores e editora. Como existirão muitas destas entidades, no banco não seria uma boa opção a seleção em uma lista pois teria a usabilidade prejudicada.

A solução foi deixar campos de preenchimento livre com auto-completar. À medida que o usuário digita, consultas são feitas no banco e devolvidos os nomes semelhantes. Se o autor/editora que o usuário preencher não estiver no banco, estes são criados.

Também pensando em diminuir a digitação, apenas foram requeridos como obrigatórios o ISBN, título, autor(es) e editoras. A imagem da capa não é obrigatória. Além disso, para incentivar o cadastro de mais livros foi dado crédito pra o usuário que o faz. Então cada livro possui o “leitor que primeiro cadastrou”.

Apesar deste esforço, já existem bancos de dados com informações de livros. Alguns dos mais notáveis são o da Amazon.com, da OCLC (Online Computer Library Center) e do Google.com. As três disponibilizam API para trazer informações a partir de um número ISBN. Mas os termos de usos destes serviços são restritivos no caso da Amazon e Google. Por exemplo, não se pode armazenar dados buscados por uma consulta em sua base de dados. E deve-se colocar um *link* para os *sites* destas empresas em toda página que possuir alguma informação obtida com a API. Ou seja, no caso da Amazon, o intuito é gerar tráfego de compradores para seu *site*. Apesar de ser uma das mais completas[9], não é de interesse da bigLib gerar tráfego para lojas. A opção pela OCLC também foi descartada pois só permitem o uso por membros cadastrados e que passam por uma rigorosa seleção, que garante que seu serviço não será um competidor deles.

A opção mais livre encontrada foi a da API da Open Library, projeto do Web Archive cuja missão de longo prazo é ter “uma página na web para cada livro publicado”. Atualmente possuem cerca de 13 milhões de livros catalogados e inclui também a procura por capas. Infelizmente, sua base de dados para livros publicados no país (que possui ISBN-10 começando por “85”) é pequena.

Com isso, usando esta *web service*, conseguiu-se evitar que o usuário preenchesse em alguns casos todos os campos. Após entrar com o ISBN, e clicar no botão “Estou com sorte”, o sistema faz uma consulta pelos dados daquele livro e preenche os campos se encontrar. Veja na figura 11 a tela de cadastro em uso.

Início > Cadastro de livros

Cadastro de livros

ISBN

[O que é isso?](#)

Título

Autor

Editora

Capa

Figura 11: Tela de cadastro de livro

5 Metodologia de desenvolvimento

O processo de desenvolvimento do sistema seguiu uma metodologia iterativa, permitindo que pequenas correções fossem feitas à medida que novas funcionalidades eram implementadas. Pouca divisão de trabalho foi feita, os integrantes optaram por trabalhar juntos.

Essa decisão visou favorecer o comprometimento com as metas do trabalho, pois trabalho em conjunto é mais motivante porque estimula a comunicação e a troca de conhecimento.

6 Resultados obtidos - telas principais

A tela inicial do sistema pode ser vista na figura 12. Nessa tela, está disponibilizado ao usuário um formulário de cadastro e um formulário de login. Depois de fazer o cadastro, o usuário é obrigado a ativar sua conta, através de um *link* enviado por *e-mail*.

A tela de boas-vindas do bigLib apresenta o logo da marca no canto superior esquerdo. No canto superior direito, há campos para 'usuário' e 'senha', seguidos por um botão 'Entrar' e um link 'Esqueceu a senha?'. O corpo da tela é dividido em duas seções principais: uma de texto informativo à esquerda e um formulário de cadastro à direita. O texto informativo descreve a missão do bigLib e inclui uma citação sobre o hábito de leitura no Brasil. O formulário de cadastro, intitulado 'Novo na bigLib? cadastre-se agora!', contém campos para 'primeiro nome', 'sobrenome', 'escolha o gênero' (menu suspenso), 'e-mail', 'usuário', 'senha' e 'confirmação de senha', além de um botão 'Cadastrar'. No rodapé, há links para 'Política de privacidade' e 'Termos de uso'.

Figura 12: Tela de Boas-Vindas

Ao fazer *login*, o usuário é levado para sua tela principal (figura 13). Nesta ele pode visualizar os livros disponíveis em sua biblioteca pessoal, ou seja, os livros de seus amigos, os quais pode pedir emprestado. Também pode ver as requisições de empréstimo e de amizade, bem como uma lista de seus livros cadastrados. No canto esquerdo superior da tela, está posicionado um menu listando as principais funcionalidades disponíveis, tais como: gerenciar empréstimo, adicionar livro, cadastrar livro, convidar amigos e navegar por tags.

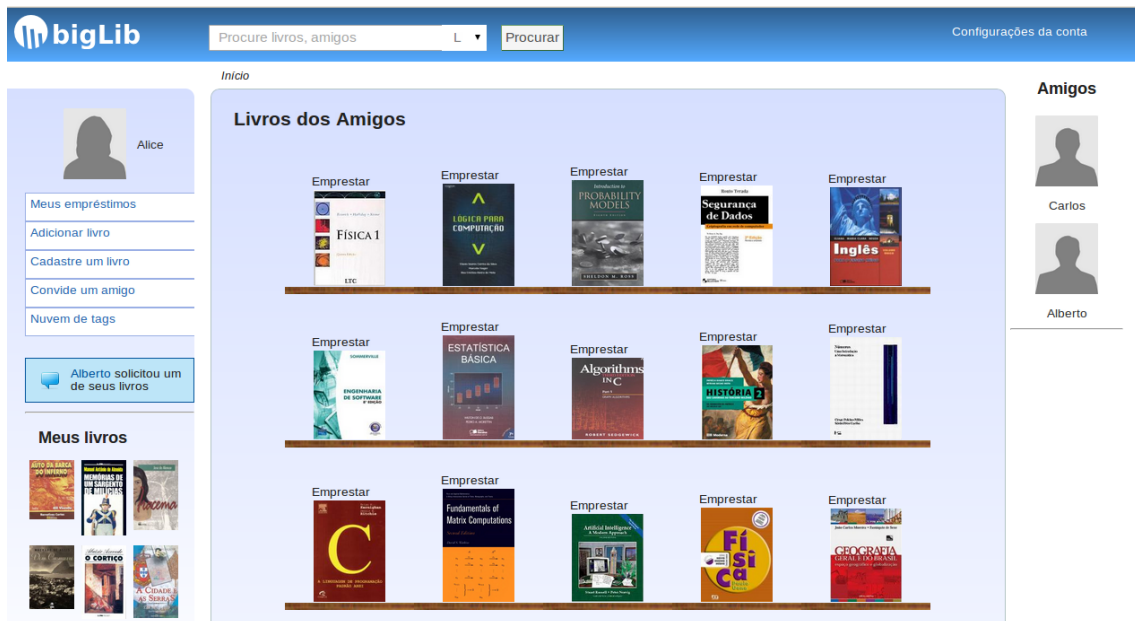


Figura 13: Tela Principal

No canto superior direito é possível acessar um menu para alterar dados pessoais, como *e-mail* e senha. Também é possível editar o perfil e sair do sistema. Ainda nessa tela, na barra superior, pode-se procurar por amigos cadastrados e por exemplares dos amigos.

Para realizar o cadastro de um livro ainda não existente no banco de dados, a tela de cadastro (figura 14) conta com duas facilidades: preenchimento automático de informações dos campos a partir de um ISBN fornecido (ao clicar no botão “Estou com sorte”) e autocompletar de autores e editora, ambas para minimizar o trabalho de digitação.

bigLib Procure livros, amigos L Procurar Configurações da conta

Início > Cadastro de livros

Cadastro de livros

ISBN 9788120321410
O que é isso?

Título Introduction to algorithms

Autor Thomas H. Cormen + -
Charles E. Leiserson
Ronald L. Rivest
ste|

Editora Clifford Stein
Garth Stein
Gertrude Stein
Jeff Stein

Capa

Figura 14: Tela de cadastro de livro

Ao requerer um livro, o usuário é dirigido a tela de solicitação de empréstimo (figura 15). Nesta pode escrever um texto livre, solicitando ou combinando o local para realizar a entrega da obra. A escolha da duração do empréstimo pode abranger um período de até três meses.

bigLib Procure livros, amigos L Procurar Configurações da conta

Início > Solicitação de empréstimo

Solicitação de empréstimo

 Livro **Álgebra linear e aplicações** de seu amigo(a) **Alberto da Silva**.

Pedido

Oi Beto, tudo bem?
Me empresta este livro? acho que estudando por ele consigo passar na matéria. Pego com você amanhã na faculdade.

Bjos,
Alice

Por que quero ler/consultar este livro. Opcional

Duração 2 meses

Solicitar emprestado

Figura 15: Tela de solicitação do empréstimo

Por fim, todos os empréstimos realizados podem ser acompanhados pela tela de controle de empréstimos (figura 16). Nela existem duas abas, a primeira delas, lista os livros do usuário que estão com seus amigos, a segunda, os livros dos amigos que estão com o usuário. Os livros são exibidos ordenados em função de sua data de devolução, ou seja, os livros que devem ser devolvidos mais cedo, aparecem no topo. A lista mostra o status do empréstimo (ativo, em devolução ou devolvido), o nome do livro, com quem ele está ou a quem pertence e a data de devolução do mesmo.

Outras telas complementam o sistema como: perfil público do usuário, nuvem de tags, tela de convite de amigos, tela de edição de livro, tela de busca de amigo e telas secundárias que fazem a transição entre as principais.

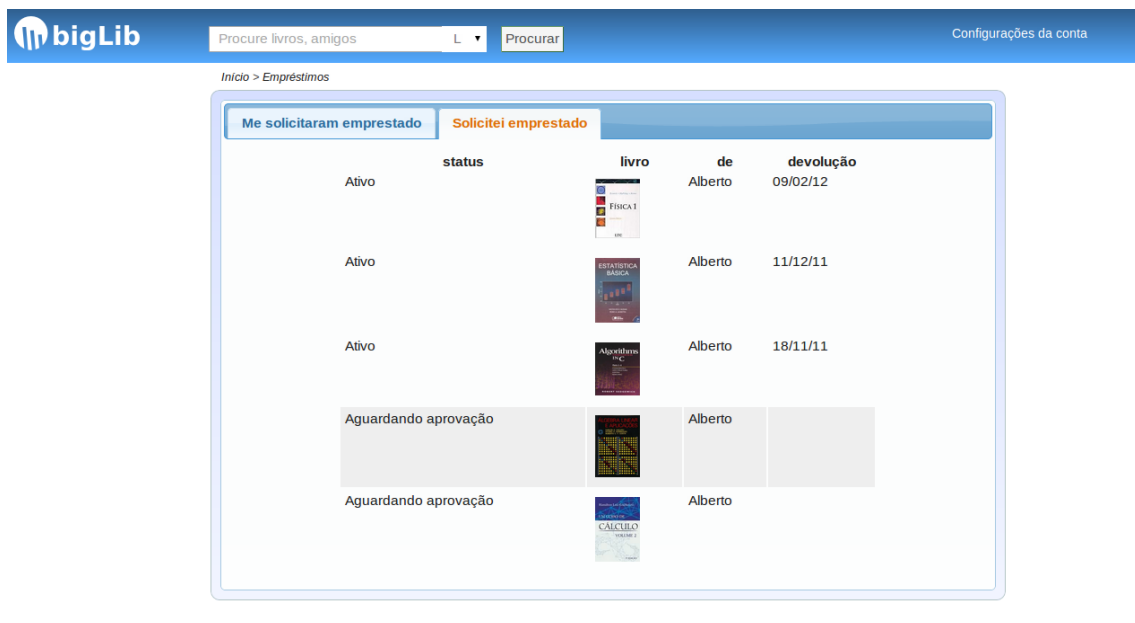


Figura 16: Tela de controle de empréstimos

7 Conclusões

O desenvolvimento de um sistema envolve estimativas que por vezes não se concretizam. Apesar disso, podemos concluir que um conjunto substancial de funcionalidades previstos no início do projeto foram escritas.

Algumas funcionalidades previstas como *login* facilitado com contas do Facebook e histórico de empréstimos não foram desenvolvidas pois limitamos os recursos em um conjunto que conseguíssemos cumprir no tempo dedicado à escrita do código.

Foi muito positiva a utilização do arcabouço Django, que compensou o tempo dedicado ao seu estudo e agilizou muito a implementação.

Dentre todas as dificuldades enfrentadas (disponibilidade de tempo, definição do escopo, criação da interface, desconhecimento prévio do arcabouço, etc), pode-se destacar a criação de um design que deveria ser agradável aos futuros usuários. Uma tarefa de caráter artístico que vai além das competências computacionais dos integrantes.

Atualmente um leitor já conseguiria: criar e ativar uma conta, logar-se, procurar e adicionar amigos, convidar amigos para o *site*, procurar, cadastrar, adicionar e classificar livros, solicitar emprestado e emprestar assim como funções de administração da conta (trocar senha, *e-mail*, editar perfil e etc). Isso é o suficiente para que as pessoas já possam ter seus empréstimos gerenciados pelo sistema, evitando o esquecimento de seus livros emprestados a amigos e tomando conhecimento dos livros que seus amigos possuem, as duas principais motivações do projeto.

Mas apesar de tudo o que foi feito ainda são necessárias algumas modificações, ou mesmo melhorias. Planejamos o seguinte para o futuro do projeto.

- Melhorar o visual
- Login via Facebook, Google e Twitter
- Melhorar a resolução do ISBN (obter a imagem da capa)
- Histórico de uso do leitor (credibilidade do usuário)
- Usuários poderem trocar mensagens entre si
- Adicionar comentários aos livros
- Utilizar servidor de *e-mail* próprio
- Possibilidade de renovação, pedir de volta livro

8 Bibliografia

- [1] JQuery: The write less, do more, javascript library (em inglês), 2011. [Online; accessed 22-novembro-2011].
- [2] J. Bennett. *Practical Django Projects*. Apress, 2008.
- [3] O. S. e Thiago Galesi. *Python e Django. Desenvolvimento ágil de aplicações web*. novatec, 2010.
- [4] v. I. Instituto Pró-Livro. Retratos da leitura no brasil, 2008. <http://www.prolivro.org.br/ipl/publier4.0/dados/anexos/48.pdf>.
- [5] R. E. Johnson. Frameworks = (components + patterns). *Commun. ACM*, 40:39–42, October 1997.
- [6] Wikipedia. International standard book number — wikipedia, the free encyclopedia, 2011. [Online; accessed 19-November-2011].
- [7] Wikipédia. Bookcrossing — wikipédia, a enciclopédia livre, 2011. <http://pt.wikipedia.org/w/index.php?title=BookCrossing&oldid=25399516>.
- [8] Wikipédia. Mvc — wikipédia, a enciclopédia livre, 2011. [Online; accessed 19-novembro-2011].
- [9] B. Wilderness. Alternatives to amazon api, 2009. [Online; accessed 22-October-2011].