



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PIAUÍ
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS
Curso de Sistemas de Informação
Prof. Francisco das Chagas Imperes Filho
Alunos: Diego Fernando de Sousa Lima
Antonio Ferreira de Carvalho



FASE IV
DETALHAMENTO DO BANCO DE DADOS E TESTE DE USABILIDADE APLICADO A
PLATAFORMA FIXCODE

1. INTRODUÇÃO

FixCode é uma plataforma desenvolvida pensada nos desenvolvedores de softwares. O intuito inicial é a interação entre os vários desenvolvedores e programadores que existem mundo afora, e para isso a comunidade leva consigo a ideia do auxílio aos programadores. O projeto está parcialmente concluído em (<https://fixcode.herokuapp.com>) e é livre no repositório (<https://github.com/diegofsousa/FixCodeProject>).

Este documento trata do detalhamento do banco de dados e teste unitário dos princípios de IHC realizado com base na ferramenta **ErgoList** (<http://www.labiutil.inf.ufsc.br/ergolist/>).

2. DETALHAMENTO DO ESQUEMA DE DADOS

Ao todo, o esquema de banco de dados da plataforma **FixCode** tem 26 entidades somando as escritas e migradas através do arquivo *models.py* como também as pré-definidas pela framework **Django**.

O sistema de banco de dados escolhido foi o **MySQL**. A representação gráfica do modelo relacional está no anexo que vai junto a este documento (banco.png).

Dentre as tabelas da aplicação, as que estão em maior relevância são:

- `auth_user`;
- `core_profile`;
- `core_followers`;
- `core_fixies`;
- `core_comentfixies`;
- `core_post`;
- `core_comentpost`;
- `core_participations`;
- `core_favorites`;
- `core_areas`;
- `core_message`;
- `core_blocked`.

2.1. USUÁRIOS E PERFIS

As tabelas **auth_user** e **core_profile** estão estritamente relacionadas pela relação *OneToOne*. A entidade `core_profile` funciona como uma extensão de `auth_user` e isso acontece por que `auth_user` é uma entidade gerada pela framework Django já com seus atributos definidos e de difícil alteração. Já a `core_profile` foram adicionados novos atributos para o usuário:

- `id`;
- `bio` (Breve descrição para o usuário);
- `git` (Link para perfil do *GitHub*);
- `imagem_perfil` (Diretório para a imagem de perfil);
- `data_cadastro` (Tempo exato em que o usuário se cadastrou no sistema);
- `visto_por_último` (Tempo exato em que usuário usou a aplicação por último);
- `ativo` (Verificação de aptidão para uso da plataforma);
- `expert` (Verificação de primeiro acesso do usuário);
- `user_id` (Chave estrangeira de `auth_user`).

2.2. POLÍTICA DE SEGUIDORES

É adotado na plataforma FixCode um esquema de seguidores, semelhante ao implantado na rede social *Twitter*. Neste caso, um cada usuário pode seguir e ser seguido por várias pessoas, constituindo assim, uma relação *ManyToMany*.

A entidade responsável pela relação entre os usuários é a **core_followers**. Esta armazena, além chave estrangeira dos dois usuários, um campo para a data de quando aconteceu o evento.

2.3. FIXIES E POSTS

Fixies (*core_fixies*) e **Posts** (*core_post*) são abstrações para, respectivamente, questionamento e partilha de conhecimento.

Os atributos da entidade *core_fixies* são:

- id;
- titulo;
- descrição;
- data;
- resolvido (Verificação de quando o criador não o considera como resolvido);
- tem_melhor_resposta (Verificação de quando o criador seleciona uma melhor resposta);
- notificacao (ver item 2.4);
- ativa_notificacao (ver item 2.4);
- user_id (Chave estrangeira de *auth_user*).

Os atributos da entidade *core_post* são:

- id;
- titulo;
- post (Texto da postagem);
- data;
- notificacao (ver item 2.4);
- ativa_notificacao (ver item 2.4);
- exibir_perfil (Verificação de quando o criador desejar que o post seja exibido no perfil);
- anexo (Diretório para a arquivo enviado junto a postagem);
- user_id (Chave estrangeira de *auth_user*).

2.3.1 RESPOSTAS DE FIXIES E COMENTÁRIOS EM POSTS

Cada *Fix* pode ter várias repostas, então há uma relação *OneToMany* entre *auth_user* e **core_comentfixies**, que é a entidade responsável pelas repostas dos *fixies*. Os atributos de *core_comentfixies* são:

- id;
- coment (Texto da resposta);
- data;
- melhor_resposta (Verificação para saber se o criador do *fix* classificou esta resposta como sendo a melhor);
- fixie_id (Chave estrangeira de *core_fixies*);

- `user_id` (Chave estrangeira de `auth_user`).

Assim como nos *fixies*, os posts também tem uma entidade que fica responsável pelo *feedback* de cada post, esta tem o nome de **`core_comentpost`** e tem menos atributos que `core_comentfixies`:

- `id`;
- `coment` (Texto do comentário);
- `data`;
- `post_id` (Chave estrangeira de `core_post`);
- `user_id` (Chave estrangeira de `auth_user`).

2.3.2. PARTICIPAÇÕES E FAVORITOS

Toda vez que um usuário comenta um *fix*, criará um registro na tabela **`core_participations`**. Por sua vez, `core_participations` tem a seguinte estrutura:

- `id`;
- `notificacao` (Quantas respostas foram dadas no mesmo *fix* após o usuário ter comentado);
- `ativa_notificacao` (Verificação se usuário deseja ser notificado pelo *fix*);
- `fixie_id` (Chave estrangeira de `core_fixies`);
- `user_id` (Chave estrangeira de `auth_user`).

Há também a possibilidade de um usuário favoritar um *fix* através de um botão na página. A cada vez que um usuário favoritar um *fix* um novo registro na tabela **`core_favorites`** é criado. Os atributos de `core_favorites` são os mesmos de `core_participations`.

Ambas, as entidades `core_participations` e `core_favorites` servem justamente para o controle de notificações da plataforma. Mais detalhes no item 2.4.

2.4. NOTIFICAÇÕES PARA OS USUÁRIOS

A possibilidade de notificações se dá para o usuário criador pelos atributos `notificacao` das entidades `core_fixies` e `core_post`. Quando esses atributos têm valores maiores que 0 (zero), a aplicação, quando consultada, devolve a mensagem para o autor alertando-o sobre o *fix* ou *post* notificado.

Outra maneira de notificar o usuário se dá quando ele participa ou favorita algum *fix*. Neste caso, quem se armazena o valor-chave das notificações é o atributo `notificacao` de `core_participations` e `core_favorites`.

2.5. ÁREAS/HABILIDADES E SUA IMPORTÂNCIA

As entidades `core_profile`, `core_fixies` e `core_post` tem um relacionamento ManyToMany com `core_areas`. Acontece que, boa parte do que o usuário vê na página principal se dá pelo conjunto de afinidades escolhidas (**`core_areas`**), por exemplo, o post que tem as áreas '*C*', '*Python*' e '*Arduíno*' será exibido na *timeline* do usuário que escolheu '*Java*', '*Android*' e '*Arduíno*' justamente pela habilidade em comum.

2.6. SISTEMA DE MENSAGENS INBOX

A entidade **core_message** é responsável por armazenar mensagens de um usuário ao outro. Sua estrutura detalhada é:

- id;
- data (Tempo em que a mensagem foi enviada);
- texto (Conteúdo da mensagem);
- visualizada (Verificação para se a mensagem foi vista pelo destinatário);
- emissor_id (Chave estrangeira do auth_user que emitiu a mensagem);
- receptor_id (Chave estrangeira do auth_user que recebeu a mensagem).

core_message recebe chave estrangeira de dois usuários distintos. Na aplicação, é feito uma seleção entre as mensagens que estão relacionadas entre estes dois usuários, mas isso já é outra história.

2.7 SISTEMA DE BLOQUEIO DE USUÁRIOS

Por questão de privacidade e segurança do usuário, também foi desenvolvida uma entidade para bloqueio de usuários chamada **core_blocked**. Funciona assim: qualquer usuário poderá bloquear qualquer usuário e após feita esta ação, o usuário bloqueado não poderá interagir com o usuário que o bloqueou em seus *fixies*, *posts* e mensagens inbox.

A estrutura de **core_blocked** foi montada assim:

- id;
- data (Tempo em que houve o bloqueio);
- user_id (Chave estrangeira do auth_user realiza o bloqueio);
- userblocked_id (Chave estrangeira do auth_user bloqueado).

3. TESTE DE USABILIDADE

Foram feitos testes unitários dos princípios de IHC para a plataforma **FixCode**. O teste foi efetuado com base na ferramenta **ErgoList** (<http://www.labiutil.inf.ufsc.br/ergolist/>).

3.1. ITENS AVALIADOS

Cada um dos itens abaixo é especializado em um aspecto ou critério que determina a ergonomia de uma interface homem.

3.1.1. PRESTEZA

Verifique se o sistema informa e conduz o usuário durante a interação.

Resultado:

Total de Questões: 17

Respondidas: 17

Não respondidas: 0

Questões conformes: 7

Questões não conformes: 4

Questões Não Aplicáveis: 6

Questões Adiadas: 0

3.1.2. AGRUPAMENTO POR LOCALIZAÇÃO

Verifique se a distribuição espacial dos itens traduz as relações entre as informações.

Resultado:

Total de Questões: 11

Respondidas: 11

Não respondidas: 0

Questões conformes: 9

Questões não conformes: 1

Questões Não Aplicáveis: 1

Questões Adiadas: 0

3.1.3. AGRUPAMENTO POR FORMATO

Verifique os formatos dos itens como meio de transmitir associações e diferenças.

Resultado:

Total de Questões: 17

Respondidas: 17

Não respondidas: 0

Questões conformes: 8

Questões não conformes: 4

Questões Não Aplicáveis: 5

Questões Adiadas: 0

3.1.4. FEEDBACK

Avalie a qualidade do *feedback* imediato às ações do usuário.

Resultado:

Total de Questões: 12

Respondidas: 12

Não respondidas: 0

Questões conformes: 3

Questões não conformes: 5

Questões Não Aplicáveis: 4

Questões Adiadas: 0

3.1.5. LEGIBILIDADE

Verifique a legibilidade das informações apresentadas nas telas do sistema.

Resultado:

Total de Questões: 27

Respondidas: 27

Não respondidas: 0

Questões conformes: 15

Questões não conformes: 2

Questões Não Aplicáveis: 10

Questões Adiadas: 0

3.1.6. CONCISÃO

Verifique o tamanho dos códigos e termos apresentados e introduzidos no sistema.

Resultado:

Total de Questões: 14

Respondidas: 14

Não respondidas: 0

Questões conformes: 6

Questões não conformes: 3

Questões Não Aplicáveis: 5

Questões Adiadas: 0

3.1.7. AÇÕES MÍNIMAS

Verifique a extensão dos diálogos estabelecidos para a realização dos objetivos do usuário.

Resultado:

Total de Questões: 5

Respondidas: 5

Não respondidas: 0

Questões conformes: 2

Questões não conformes: 2

Questões Não Aplicáveis: 1

Questões Adiadas: 0

3.1.8. DENSIDADE INFORMACIONAL

Avalie a densidade informacional das telas apresentadas pelo sistema.

Resultado:

Total de Questões: 9

Respondidas: 9

Não respondidas: 0

Questões conformes: 6

Questões não conformes: 1

Questões Não Aplicáveis: 2

Questões Adiadas: 0

3.1.9. AÇÕES EXPLÍCITAS

Verifique se é o usuário quem comanda explicitamente as ações do sistema.

Resultado:

Total de Questões: 4

Respondidas: 4

Não respondidas: 0

Questões conformes: 3

Questões não conformes: 0

Questões Não Aplicáveis: 1

Questões Adiadas: 0

3.1.10. CONTROLE DO USUÁRIO

Avalie as possibilidades de o usuário controlar o encadeamento e a realização das ações.

Resultado:

Total de Questões: 4

Respondidas: 4

Não respondidas: 0

Questões conformes: 4

Questões não conformes: 0

Questões Não Aplicáveis: 0

Questões Adiadas: 0

3.1.11. FLEXIBILIDADE

Verifique se o sistema permite personalizar as apresentações e os diálogos.

Resultado:

Total de Questões: 3

Respondidas: 3

Não respondidas: 0

Questões conformes: 0

Questões não conformes: 1

Questões Não Aplicáveis: 2

Questões Adiadas: 0

3.1.12. EXPERIÊNCIA DO USUÁRIO

Avalie se usuários com diferentes níveis de experiência têm iguais possibilidades de obter sucesso em seus objetivos.

Resultado:

Experiência do Usuário

Total de Questões: 6

Respondidas: 6

Não respondidas: 0

Questões conformes: 4

Questões não conformes: 0

Questões Não Aplicáveis: 2

Questões Adiadas: 0

3.1.13. PROTEÇÃO CONTRA ERROS

Verifique se o sistema oferece as oportunidades para o usuário prevenir eventuais erros.

Resultado:

Total de Questões: 7

Respondidas: 7

Não respondidas: 0

Questões conformes: 1

Questões não conformes: 1

Questões Não Aplicáveis: 5

Questões Adiadas: 0

3.1.14. MENSAGENS DE ERRO

Avalie a qualidade das mensagens de erro enviadas aos usuários em dificuldades.

Resultado:

Total de Questões: 9

Respondidas: 9

Não respondidas: 0

Questões conformes: 6

Questões não conformes: 1

15. Questões Não Aplicáveis: 2

15.15. Questões Adiadas: 0

3.1.16. CORREÇÃO DE ERROS

Verifique as facilidades oferecidas para que o usuário possa corrigir os erros cometidos.

Resultado:

Total de Questões: 5

Respondidas: 5

Não respondidas: 0

Questões conformes: 5

Questões não conformes: 0

Questões Não Aplicáveis: 0

Questões Adiadas: 0

3.1.17. CONSISTÊNCIA

Avalie se é mantida uma coerência no projeto de códigos, telas e diálogos com o usuário.

Resultado:

Total de Questões: 11

Respondidas: 11

Não respondidas: 0

Questões conformes: 10

Questões não conformes: 1

Questões Não Aplicáveis: 0

Questões Adiadas: 0

3.1.18. SIGNIFICADOS

Avalie se os códigos e denominações são claros e significativos para os usuários do sistema.

Resultado:

Total de Questões: 12

Respondidas: 12

Não respondidas: 0

Questões conformes: 6

Questões não conformes: 0

Questões Não Aplicáveis: 6

Questões Adiadas: 0

3.1.19. COMPATIBILIDADE

Verifique a compatibilidade do sistema com as expectativas e necessidades do usuário em sua tarefa.

Resultado:

Total de Questões: 21

Respondidas: 21

Não respondidas: 0

Questões conformes: 6

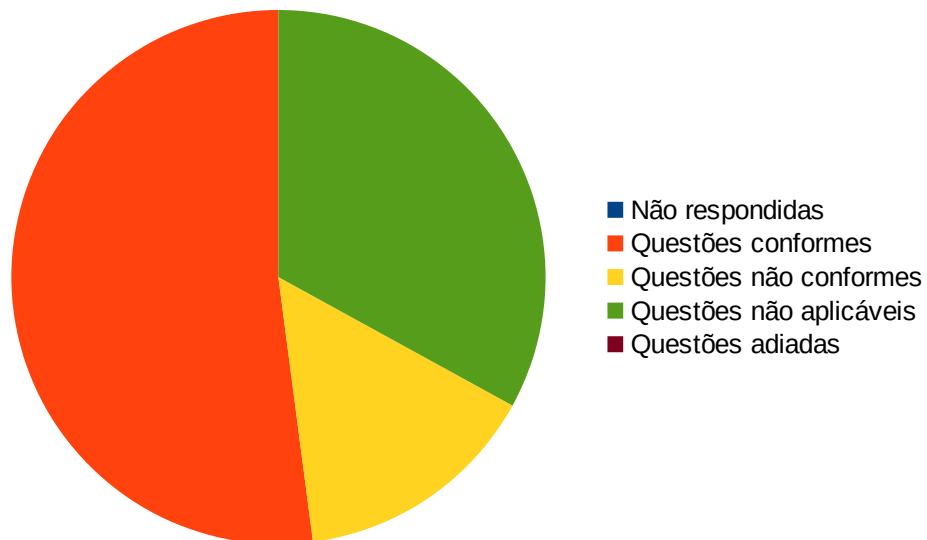
Questões não conformes: 3

Questões Não Aplicáveis: 12

Questões Adiadas: 0

3.2. RESULTADO FINAL

Abaixo está o gráfico representativo do resultado do teste de usabilidade da plataforma **FixCode**. O cálculo efetuado é a soma de todos os parâmetros das questões respondidas.



Resultado do teste de usabilidade ErgoList

O total numérico exato fica:

Total de Questões: 194 – 100%

Respondidas: 194 – 100%

Não respondidas: 0 – 0%

Questões conformes: 101 – ~52,88%

Questões não conformes: 29 – ~15,18%

Questões Não Aplicáveis: 64 – ~33,51%

Questões Adiadas: 0 – 0%