

On Saba – Software para Associação do Bairro São Francisco de Torres

Denner Evaldt Machado¹, Cassio Huggentobler de Costa²

¹Análise e Desenvolvimento de Sistemas – Universidade Luterana do Brasil (ULBRA)
Campus Torres - RS

²Professor(a) dos Cursos de Computação – Universidade Luterana do Brasil (ULBRA)
Campus Torres – RS

dnrevaldt@gmail.com, professor@cassiocosta.pro.br,

Abstract. *This article presents the project developed for the association of the São Francisco of Torres RS neighborhood. It's an extension project of Ulbra Torres and aims to maintain a register of members and their dependents, generate associated portfolios and reports emission to manage the association. The project was developed using the Scrum methodology and the tools ASP.Net MVC, HTML5, JavaScript, C # and MySQL.*

Resumo. *Esse artigo apresenta o projeto desenvolvido para a associação do bairro São Francisco de Torres RS. É um projeto de extensão da Ulbra Torres e tem por objetivo manter um cadastro de associados e os seus dependentes, gerar carteiras de associado e emissão de relatórios para gerenciar a associação. O Projeto foi desenvolvido utilizando a metodologia Scrum e as ferramentas ASP.Net MVC, HTML5, JavaScript, C# e Mysql.*

1. Informações Gerais – Introdução

Para o projeto de extensão do Curso de Sup. Tec. Em Análise e Desenvolvimento de Sistemas da Ulbra campus Torres foi solicitada uma assistência tecnológica à associação de moradores do Bairros São Francisco também da cidade de Torres, um aplicativo para cadastramentos dos moradores associados bem como os dependentes destes associados.

Todos os requisitos foram levantados em reuniões com o cliente e foi desenvolvido o On Saba, aplicativo para gerenciar a associação de moradores do bairro com funções de controle de associados e dependentes, geração de carteirinhas de associados e relatórios de informações dos moradores para número de eleitores, número de crianças e qual escola estudam, entre outros. Mesmo tendo sido construído para uma associação específica, nada impede que o *On Saba* possa ser utilizado para outras associações.

Nos próximos capítulos será mostrado como foi o desenvolvido, qual metodologia foi adotada e quais ferramentas foram utilizada para a construção e o resultado final do aplicativo *On Saba*.

2. Referencial Teórico

As subseções seguintes irão apresentar as principais referências para implementação deste projeto.

2.1. Associação de Bairros

Uma Associação de bairros é uma união e organização de moradores onde o principal objetivo é servir como uma ferramenta para a mobilização dos moradores, a fim de enfrentar problemas concretos que surgem no dia a dia de uma comunidade. Trata-se de um espaço privilegiado que tenta conscientizar de modo conjunto a população para construir uma sociedade igualitária e justa, onde se possa exercer cidadania.

É papel de uma associação de bairros reivindicar junto ao poder público e órgão competentes os direitos do povo que paga os seus impostos. Neste sentido cabe a associação definir junto com os moradores residentes, qual desses direitos são os mais prioritários.

Uma associação possui uma diretoria composta por um presidente, vice-presidente, tesoureiro e um secretário. Esta equipe tem o dever de gerenciar os moradores, reivindicar melhorias junto ao governo, conhecer bem a comunidade, a região e a cidade. Também deve apresentar propostas e fiscalizar as ações dos poderes públicos e talvez o mais importante, desenvolver projetos comunitários.

Todos esses deveres de uma associação requer um gerenciamento e um controle muito minucioso, para que se tenha a informação organizada e rápida quando solicitada. Assim a tecnologia não pode ficar de fora. Um software para controlar a gestão de uma associação se faz necessário e deve comportar todas as funcionalidades para gerir esse tipo de negócio.

2.3. Metodologia

A metodologia usada para o desenvolvimento do projeto é o *Scrum*, que é um conjunto de instruções e ferramentas que podem auxiliar em um projeto de uma forma ágil. As vantagens de usá-la em relação às outras metodologias ágeis é que ele é simples de implementar e prover problemas futuros (SABBAGH, 2014).

O *Scrum* consiste em dividir o desenvolvimento em ciclos (*sprints*), que é o período de tempo determinado para realizar uma tarefa. Essas *Sprints* são divididas em fases menores: reunião de planejamento, reunião diária, desenvolvimento, revisão da Sprint, retrospectiva da *Sprint*. Para haver Sprint terão que ter um *product backlog*, é basicamente uma lista de requisitos, itens, que o cliente deseja (SABBAGH, 2014).

O ciclo de vida da *SCRUM* é baseado em três fases principais (SABBAGH, 2014):

- Pré – planejamento: nessa etapa são definidos os requisitos do projeto onde é armazenado em um documento chamado *product backlog*. Posteriormente são

realizadas avaliações para o desenvolvimento de cada requisito. E aonde são definidos os *stakeholders* do projeto e ferramenta que irão utilizar.

- **Desenvolvimento:** nessa etapa o *software* é desenvolvido em ciclos com suas subfases: análise, projeto, implementação e testes. Cada um desses ciclos tem duração de duas semanas a um mês.
- **Pós – planejamento:** essa é a etapa final da fase onde são feitas reuniões para analisar o que foi desenvolvido e mostra o *software* atual para o cliente, assim tendo um aval do mesmo.

A figura 1 demonstra como funciona o ciclo da *Scrum* (SABBAGH, 2013).

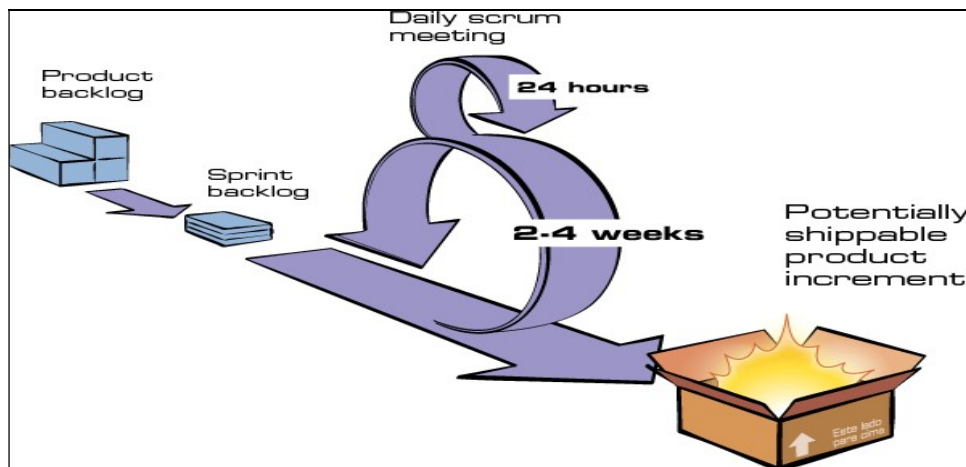


Figura 1 - Ciclo da Scrum

Explicação de cada etapa do ciclo do *scrum* da figura 1.

- **Product Backlog:** No *product backlog* serão levantados todos os requisitos que o cliente gostaria de realizar no projeto. Essa lista é sempre atualizada, de acordo com a importância dada pelo cliente.
- **Sprint Backlog:** Na *Sprint Backlog* serão definidas as tarefas a serem desenvolvidas baseadas no *Product Backlog* para o desenvolvimento do incremento do produto no *Sprint*.
- **Daily Scrum:** No *Daily Scrum* será realizada uma reunião diariamente que proporciona à auto-organização do time de desenvolvimento e tem por objetivo proporcionar a equipe uma visibilidade do trabalho realizado e que vai ser realizado futuramente.
- **Sprint Review:** Ao final de cada *Sprint*, será feito uma reunião em conjunto com o time de desenvolvimento e o cliente para obter um *feedback* dessa pessoa sobre o incremento do produto produzido.
- **Increment:** É o incremento do produto a cada ciclo do *Scrum*.

2.4. Tecnologias utilizadas

Para o desenvolvimento do software serão utilizadas algumas tecnologias abordadas a seguir.

2.4.1. ASP.Net MVC

O *Active Server Pages*, ASP.NET, é a plataforma da *Microsoft* para o desenvolvimento de aplicações *WEB*. É um componente do IIS¹ que permite através de uma linguagem de programação C# para criar páginas dinâmicas. (SANCHES, 2014). O *Model-view-controller* (MVC) é um padrão de arquitetura de *software* que separa em camadas os dados e a lógica de negócio entre o *Model* e *View* introduzindo um componente no meio deles chamado *Controller*, dessa forma as alterações feitas não afetam a manipulação de dados, assim podendo ser reorganizados sem mudar o *view* (Lamim, 2009). Na figura 2 demonstra a relação entre o *Model*, *View* e *Controller*.

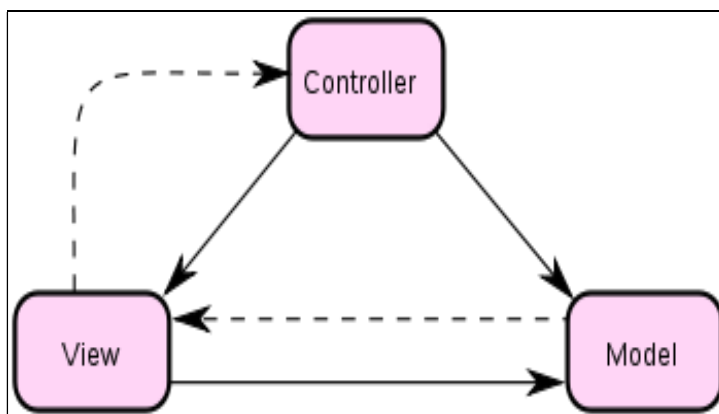


Figura 2 - Relação entre *Model*, *View* e *Controller*

2.4.2. MySQL

Conforme Pisa (2012) é um sistema gerenciador de banco de dados, usado para armazenar informações. O serviço utiliza a linguagem SQL, que permitirá acessar o banco. Nesse projeto é utilizada uma arquitetura *front end* chamada phpmyadmin que facilitará o gerenciamento da base de dados.

2.4.3. Responsive Web Design

De acordo Eis (2011) *responsive web design* é um conceito, que tem objetivo fazer com que um site se adeque a todos os tipos de resoluções de telas. Antigamente as aplicações *web* eram desenvolvidas apenas a plataforma *desktop*, que tinham telas em média de 14, 15 e 17 polegadas. Desta forma as aplicações eram desenvolvidas a esses padrões de telas, o que na época não influenciava muito.

¹ IIS – Internet Information service é um servidor *WEB* criado pela *Microsoft* para seus sistemas operacionais para servidores.

Atualmente destacamos que o cenário está totalmente diferente. Hoje estamos na era da mobilidade onde existe uma variedade grande de tamanho de telas e resoluções. O grande problema que se enfrenta é fazer com o que as aplicações que eram desenvolvidas para os desktops se adequem perfeitamente aos dispositivos móveis, como por exemplo: *smartphones*, *tablets*, que possuem em média entre 2 até 13 polegadas (EIS, 2011). Veja na figura 3 uma aplicação *web* se adequando a diferentes dispositivos.



Figura 1 - Utilização do conceito de responsividade

2.4.4. Bootstrap

É um *framework*² onde se utiliza bibliotecas de códigos e recursos *HTML*, *CSS* e *JavaScript* para desenvolver sites compatíveis com *desktop*, *tablets* e *smartphones* usando o mesmo código fonte (Silotto, 2013). Utilizando-o encontrará classes prontas para criação de botões, menus, tabelas, formulários e entre outros.

3. O Software desenvolvido

O software foi desenvolvido para funcionar em plataforma Web, de forma responsiva, o que dá possibilidade do usuário acessar de qualquer plataforma *desktop* ou *mobile*. As funcionalidades desenvolvidas se aplicam a necessidade do usuário final, como segue:

- Cadastro de Escolas – é necessário cadastrar as escolas pois necessário está informação no cadastramento de dependentes. Isso vai poder dar a informação de quantas pessoas estudam em qual escola;
- Cadastro de Associados – controle de todos os associados do bairro;
- Cadastro de Dependentes – cada associado pode ter dependentes.
- Geração de carteiras de Associados e dependentes – pode ser gerada a carteira de associado. A carteira leva a foto e os dados do associado.
- Relatórios de aniversariantes por período – é possível ver quais associados e/ou dependentes estão de aniversário no mês.

² Framework é uma abstração que une códigos comuns entre vários projetos fornecendo uma funcionalidade genérica (MÜLLER, 2014).

- Relatório de associados e dependentes que são eleitores no bairro – é possível saber que é votante e qual a zona e sessão que vota.

O Software foi desenvolvido para funcionar num navegador de internet e portanto pode ser utilizado de qualquer plataforma ou dispositivo, como celulares, *tablets*, *notebooks*, entre outros. Desta forma está hospedado em nuvem e requer que o utilizador tenha acesso a *internet*.

4. Aplicação da Metodologia

A seguir serão aplicadas as principais fases da metodologia.

4.1. Pré-Planejamento

Fase onde foram levantados os requisitos baseado em pesquisa de *softwares* similares existentes. Veja na Tabela 1 os seguintes requisitos do *product backlog*. Através do *product backlog* baseado em seu grau de prioridade a ser desenvolvido e suas respectivas horas gastas.

Tabela 1 - Product Backlog

ID	REQUISITO	PRIORIDADE	HORAS
01	Cadastramento de Escolas	100	8h
02	Cadastramento de Associados	90	15h
03	Cadastramento de Dependentes	80	8h
04	Geração Carteiras Associados/Dependentes	70	8h
05	Relatório de Associados	65	5h
06	Relatório de Dependentes	65	5h
05	Relatório de Aniversariantes	65	8h
06	Relatório de eleitores	60	6h
07	Relatório de homens e mulheres	50	8h
08	Relatório de Crianças estudantes	40	8h

4.2. Desenvolvimento

Fase onde foi feito o desenvolvimento das funcionalidades do *product backlog* realizado. A implementação foi dividida em dois *Sprints*, com suas respectivas horas gastas por cada tarefa e a sua prioridade para ser desenvolvida. O Diagrama de Casos de Uso foi implementado de forma incremental a cada *Sprint*, porem demonstrado em sua fase final na Figura 4.

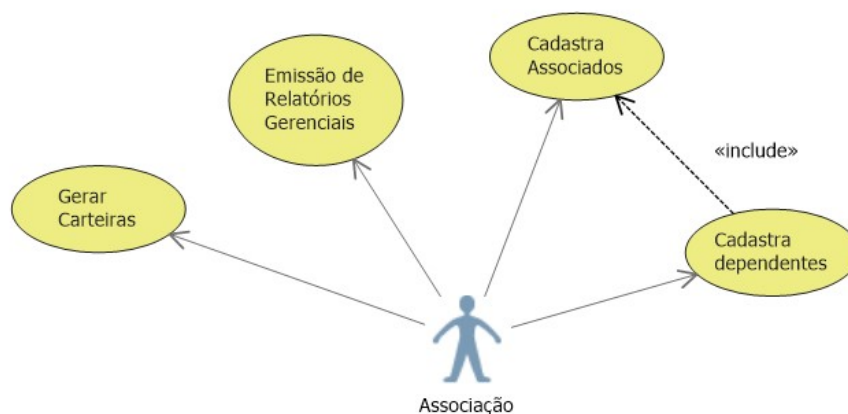


Figura 4 – Diagrama de Casos de Uso

Para melhor visualização do sistema, demonstra-se o DER (Diagrama de Entidade e Relacionamento) na Figura 5, representada em sua fase final.

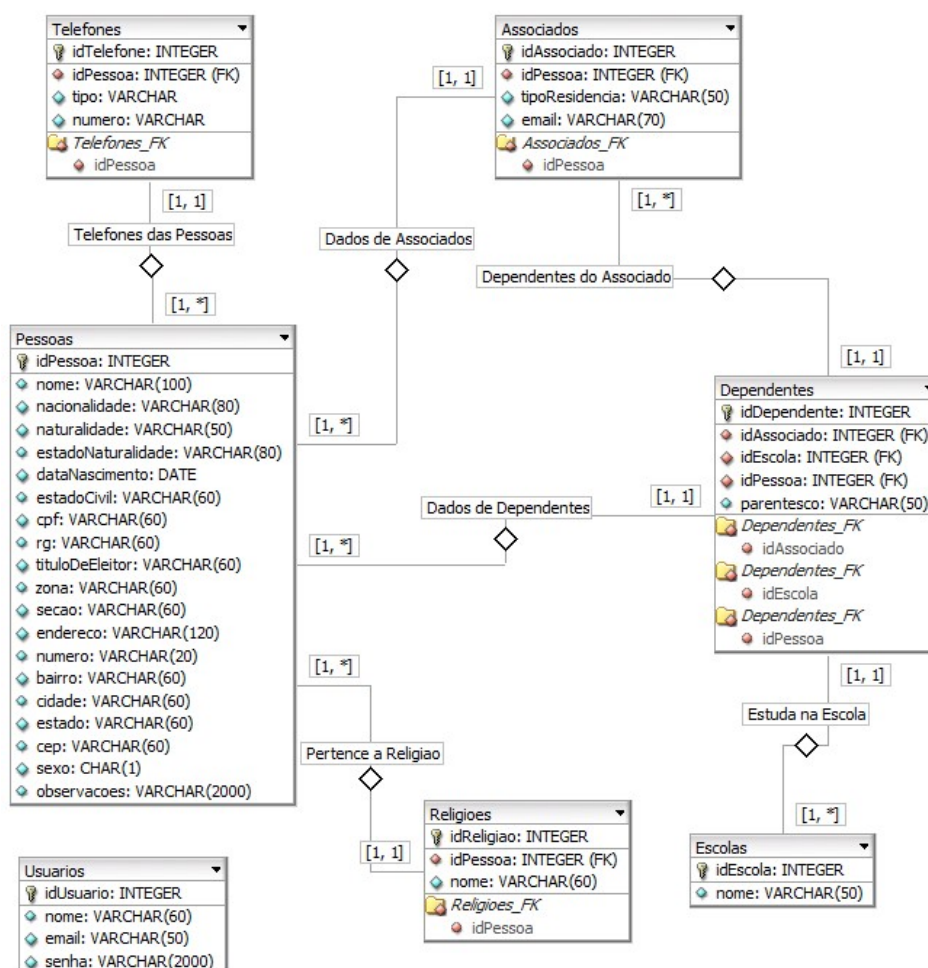


Figura 5 – Diagrama de Entidade e Relacionamento

4.2.1. Sprint 1 – Cadastros

Para o sprint ficou definido os cadastros, onde neste Release a associação já pode iniciar a utilização. Veja a seguir na Tabela 2 de *Sprint Backlog* da *Sprint 1*.

Tabela 2 – Sprint Backlog da Sprint 1

ID	TAREFAS	PRIORIDADE	ESTIMATIVA
01	Modelagem inicial	10	8h
02	Criar projeto Visual Studio	70	2h
03	Criar repositório no git hub	60	3h
04	Criar interface de comunicação C#-Mysql	50	8h
05	Criar Model para Cadastro de Escolas	50	1h
06	Criar Controller para Cadastro de Escolas	50	2h
07	Criar Views para Create, Update, Delete e Get para cadastro de Escolas.	40	4h
08	Criar Model para Cadastro de Associados	40	4h
09	Criar Controller para Cadastro de Associados	5	1h
10	Criar Views Criar Views para Create, Update, Delete e GetOne e GetAll para cadastro de Associados.	35	8h
11	Criar Model para Cadastro de Dependentes	25	8h
12	Criar Controller para Cadastro de Dependentes	15	8h
13	Criar Views Criar Views para Create, Update, Delete e GetOne e GetAll para cadastro de Dependentes.	100	1h
14	Criar Model para Geração de Carteira	90	8h
15	Criar Controller para Geração de Carteira	10	1h
16	Criar Views para Geração de Carteiras.	12	8h

Na Figura 6 demonstra o *screenshot* da tela inicial do ambiente do aplicativo.

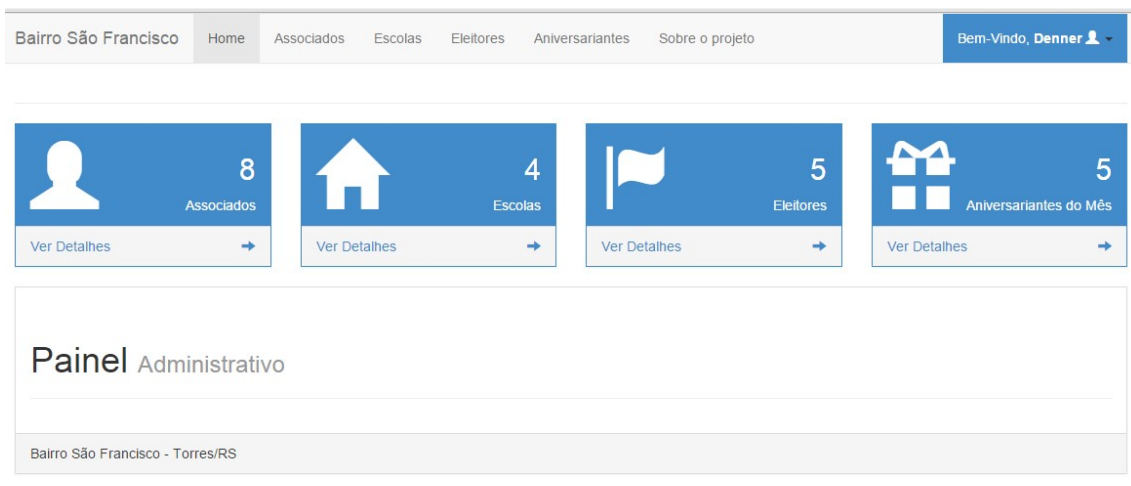


Figura 6 – Tela inicial do aplicativo

Na figura 7 a tela que mostra os associados cadastrados. O aplicativo utiliza paginação mostrando de cinco em cinco associados para dar uma melhor navegabilidade ao utilizador. As outras consultas do aplicativo seguem o mesmo padrão de usabilidade.

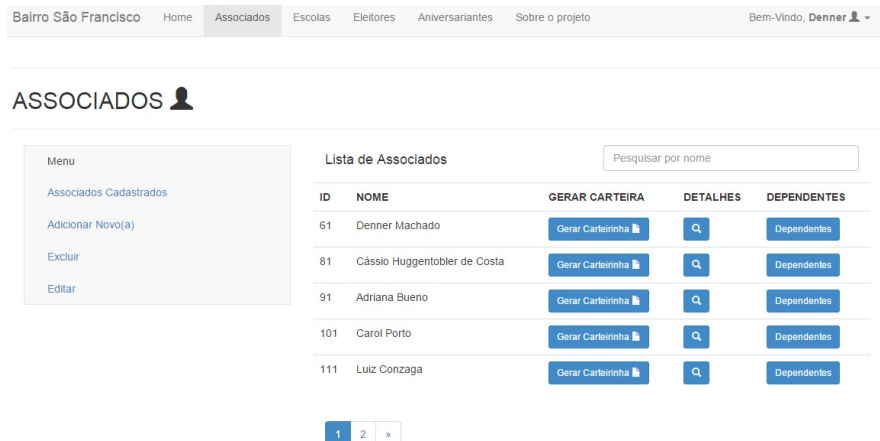


Figura 7 – Tela Lista de Associados

Na figura 8 uma parte da tela de cadastro de associado. Todos os dados coletados junto ao cliente foram considerados nesta tela. As telas de cadastros seguem o mesmo padrão de usabilidade.

Menu

[Associados Cadastrados](#)

[Adicionar Novo\(a\)](#)

[Excluir](#)

[Editar](#)

Adicionar Novo(a)

Nome

Nacionalidade

Naturalidade

Estado Naturalidade

Data de Nascimento

Estado Civil

CPF

RG

Título de Eleitor

Figura 8 – Tela Cadastro de Associado

4.2.2. Sprint 2 – Relatórios

Nessa *Sprint* foi implementados os relatórios do aplicativo. Veja a seguir na Tabela 3 de *Sprint Backlog* da *Sprint 2*.

Tabela 3 – Sprint Backlog da Sprint 2

ID	TAREFAS	PRIORIDADE	ESTIMATIVA
01	Relatório de Associados	10	1h
02	Relatório de pessoas por tipo(crianças, homens, mulheres, idosos)	20	2h
03	Relatório de Aniversariantes	80	3h
04	Relatório de Crianças por escola	80	3h
05	Relatório de número de eleitores por Zona e Sessão	80	3h

Na figura 9 o relatório dos aniversariantes do mês.

Aniversariantes do Mês 🎂

Associados			Dependentes	
Nome	Data de Nascimento	Email	Nome	Data de Nascimento
Denner Machado	11/11/1991	dnrevaldt@gmail.com	Dependente Denner	11/11/1991
Jaqueline de Souza Machado	22/11/1995	jaque@i.com		
Silvana Pereira Costa	27/11/1978	teste@email.com		
Alexandre Gatelli	27/11/1966	cassio.hc@gmail.com		

[Voltar para o menu](#)

Figura 9 – Tela Aniversariantes do mês

A figura 10 ilustra a tela do relatório de eleitores cadastrados.

Eleitores 🇧🇷

Nome	Título de Eleitor	Zona	Secao
Denner Machado	111111111111	233	22
Adriana Bueno	333333333333	87	987
Luiz Conzaga	111111111111	233	22
Jaqueline de Souza Machado	111111111111	233	22
Silvana Pereira Costa	111111111111	233	22

[Voltar para o menu](#)

Figura 10 – Tela de associados que são eleitores

4.3. Pós-Planejamento

Essa é a fase de encerramento da *Scrum*, onde são realizadas reuniões para analisar o progresso do projeto e demonstrar o *software* atual, assim verificando se foi implementado o que realmente foi especificado na análise de requisitos. E ocorre também a integração de todas as funcionalidades do sistema, bateria de testes e finalização de documentações.

5. Conclusão

A participação e o desenvolvimento de um projeto real, ou seja, um projeto que vai além de uma disciplina vista na faculdade é de extrema importância para ensino-aprendizagem para o aluno e também para o professor. Para o aluno dá a possibilidade de por em prática os conhecimentos teóricos adquiridos em sala de aula e o mais importante, dá a possibilidade de entregar um aplicativo funcional para um usuário real. Para o professor que pode divulgar a outros alunos motivando a importância dos trabalhos extra classes.

Participar de um projeto de extensão é de grande satisfação pois dá a possibilidade da universidade colaborar para o desenvolvimento socioeconômico de uma região, gerando valor neste caso com o aplicativo para a associação de moradores do bairro São Francisco.

O aplicativo atendeu todos os requisitos, foi desenvolvido para a plataforma web e já está em produção. Para o próximo semestre, a ideia é dar continuidade neste projeto passando essa continuidade para outros alunos e criando viabilidade para outras associações, não somente de bairros, utilizar o aplicativo de modo *online*.

6. Referências

SANCHES, Fabricio. ASP.NET MVC – Desenvolvimento Web com Asp.Net MVC. São Paulo: Casa do Código, 2014.

LAMIM, Jonathan. **MVC - O padrão de arquitetura de software**. Disponível em: <http://www.oficinadanet.com.br/artigo/1687/mvc_-_o_padrao_de_arquitetura_de_software>. Acesso em: 20 maio 2014.

MÜLLER, Nicolas. **Framework: o que é e para que serve?**. Disponível em: <http://www.oficinadanet.com.br/artigo/1294/framework_o_que_e_e_para_que_serve>. Acesso em: 28 maio 2014.

PISA, Pedro. **O que é e como usar o MySQL?** Disponível em: <<http://www.techtudo.com.br/artigos/noticia/2012/04/o-que-e-e-como-usar-o-mysql.html>>. Acesso em: 19 mar. 2014.

SABBAGH, Rafael. **Scrum: Gestão Ágil para Projetos de Sucesso**. São Paulo: Casa do Código, 2013. 277 p.

SANCHES, Claudio. **Tudo sobre Layout Responsivo #01: Introdução Responsive Design**. Disponível em: <<http://www.ferramentasblog.com/2012/05/tudo-sobre-layout-responsivo-responsive-design-introducao.html>>. Acesso em: 17 maio 2014.

SILOTTO, Reinaldo. **Bootstrap 3 – Conheça a nova versão do Editor Pingendo**. Disponível em: <<http://www.tekzoom.com.br/bootstrap-3-conheca-versao-editor-pingendo/>>. Acesso em: 11 mar. 2014.