UNIVERSIDADE TECNOLOGICA FEDERAL DO PARANÁ

CAMPUS PONTA GROSSA

ADS - Tecnologia em Análise e Desenvolvimento de Sistemas

Lucas Gabriel Gonsalves - RA 1590294

Luciano Santana dos Santos - RA 1589490

PROJETO DE ARQUITETURA DE SOFTWARE

SUMÁRIO

Luciano
Lucas

1. Referências

Como referências para o desenvolvimento de um programa para a SATI, utilizamos:

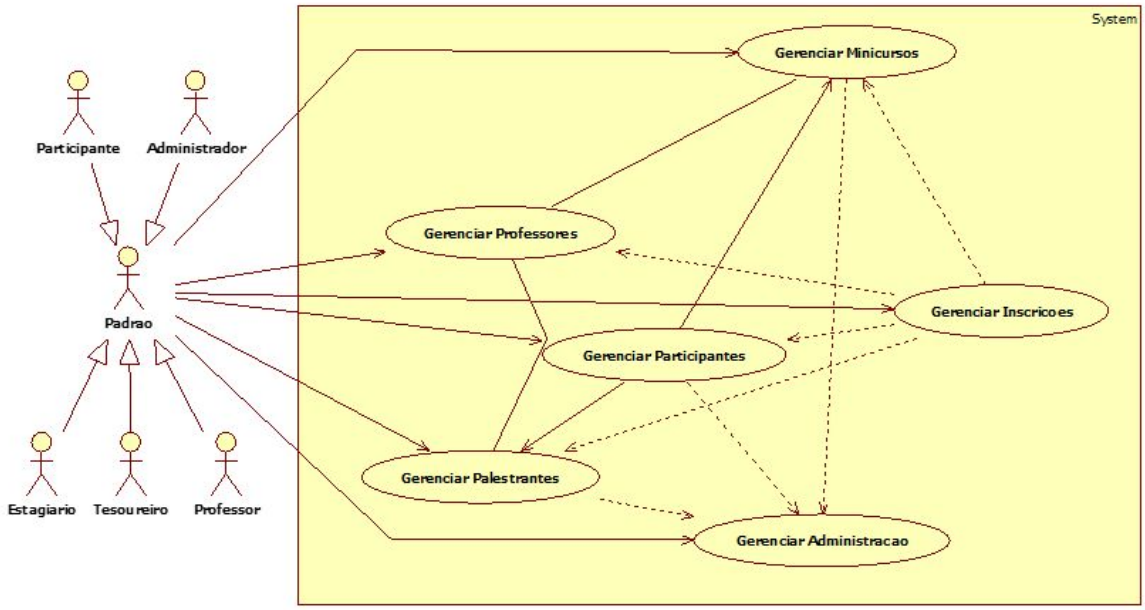
- O site atual da SATI: http://sati.pg.utfpr.edu.br
- As histórias desenvolvidas no Easy Backlog <http://easybacklog.com> e disponíveis aqui como o ANEXO I (AnexoI-SATI.pdf), disponível também em <https://drive.google.com/file/d/0B8ha40i4MQL6b3JFZGEzdzZJRnc/view?usp=sharing>.
- http://www.guj.com.br/java/97185-projeto-de-software-documento-de-exemplo

2. Metas e Restrições da Arquitetura

Poderão haver mais de um evento ao mesmo tempo, os participantes utilizarão o RA para mandar para o sistema o comprovante de presença através de um leitor. Os participantes que não forem da universidade precisarão utilizar do CPF, feito de forma mais manual, para poder comprovar a presença nos eventos. No caso dos estudantes será necessário mostrar a carteirinha de RA para algum responsável no evento para passar pelo leitor e enviar para o sistema. No caso de participantes de fora da universidade, eles deverão mostrar novamente o CPF ao final do evento para o responsável marcar sua presença.

- O software deverá ser programa em linguagem Java, utilizando os recursos de Orientação a Objeto desta linguagem, facilitando a reutilizaçao de código.

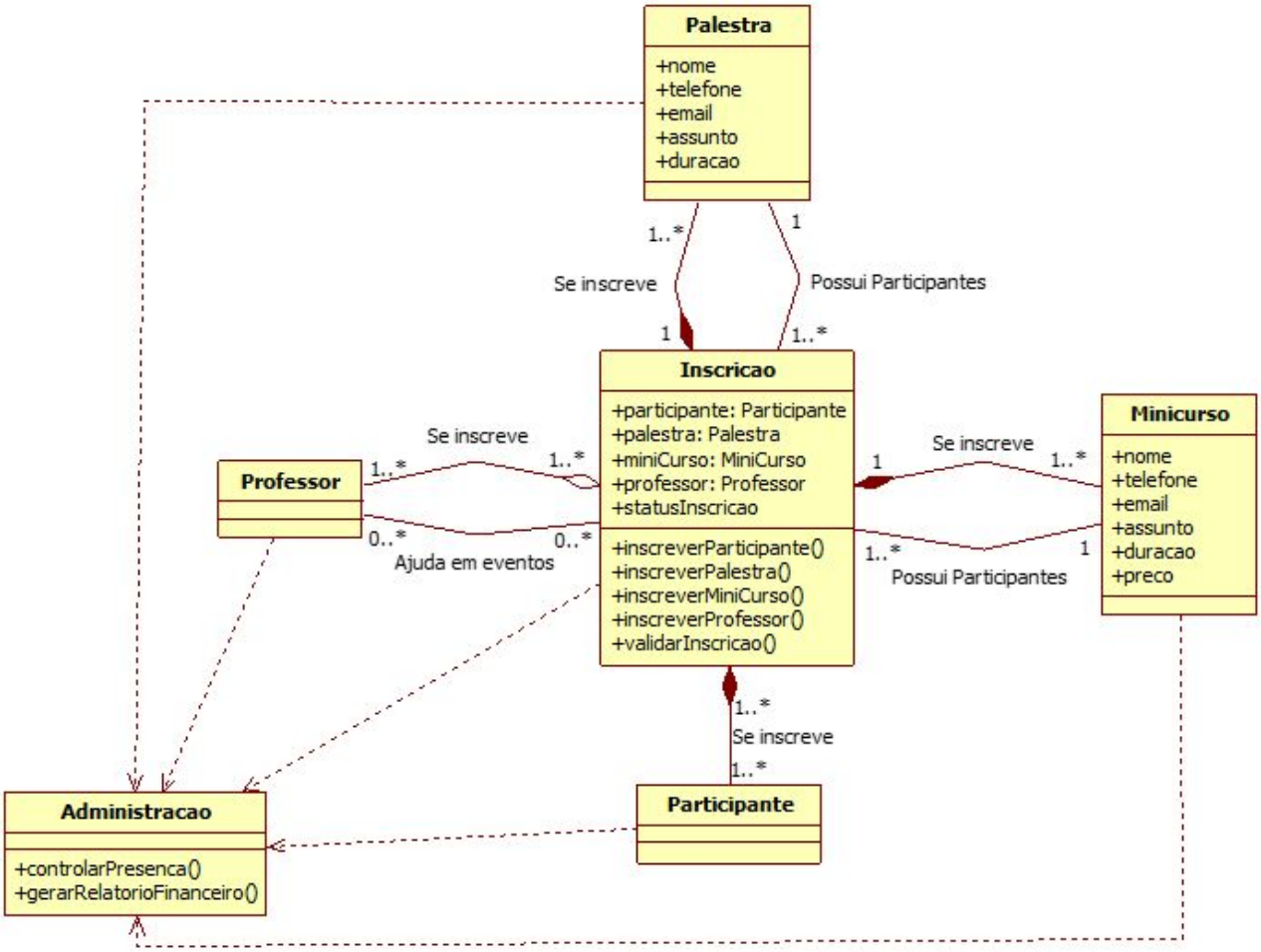
3. Visão de Casos de Uso



Neste diagrama foram usados 6 atores, 5 deles interagem com as diversas partes do sistema. Foi criado um novo ator, Padrao, para poder facilitar na modelagem do diagrama. O ator Padrao possui acesso à todas as funções do sistema, ele possui todos os privilégios de acesso de todos os outros atores do sistema.

O caso de uso “Gerenciar Inscrição” gera tantas dependências pois todas as inscrições serão tratadas nele, isso será mais explorado na visão lógica.

4. Visão Lógica



A parte administrativa necessita da existência de cada uma das outras classes, Minicursos, Participantes, Palestra e Professores pois caso não houvesse essas classes não teria o porquê de se criar um setor administrativo para o evento.

O setor administrativo ficará responsável pelo controle de presença dos usuários e pelas finanças ao final do evento.

Quando alguém entrar na tela de inscrição da SATI aparecerá na tela as opções:

-Se inscrever

-Visualizar minha inscrição

-Entrar como administrador

Ao clicar em se inscrever, o usuário deverá entrar com as informações pessoais e escolher os eventos que deseja participar. Ao clicar em finalizar, será criado um participante e automaticamente uma inscrição para o participante.

Ao clicar em Visualizar minha inscrição, o usuário entrará com seus dados, CPF ou RA e senha e terá acesso à quais eventos ele está inscrito e aos boleto

s de pagamento, caso tenho escolhido mini curso.

Ao clicar em Entrar como administrador, o usuário deverá entrar com login e senha, após isso terá acesso à inscrever nova palestra e novo minicurso, caso alguém tenha interesse em dar uma palestra ou minicurso, esse deverá estar presente com um responsável da SATI para criar um novo evento.

O statusInscrição do Inscrição diz respeito se uma inscrição foi validada, caso um participante escolha um minicurso, a validação será feita por meio do pagamento. Quando relacionado à palestras ou minicursos, dirá a respeito da situação dos documentos (para comprovar certificado em determinada área) caso seja necessário.

No Inscrição, foi declarado o tipo de atributo para mostrar que cada atributo é um objeto, como por exemplo, Participante ou Minicurso.

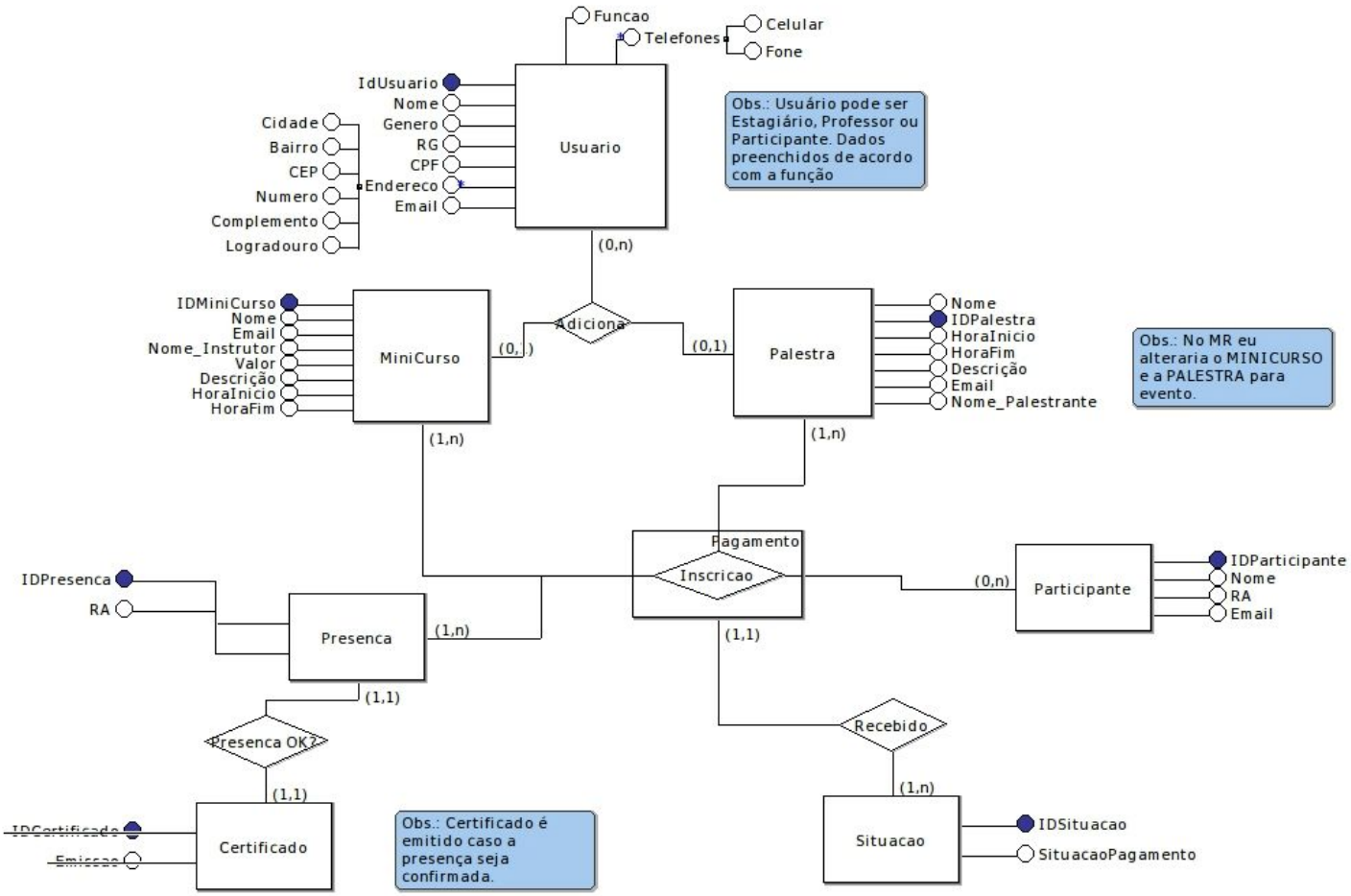
Não foi observado uma necessidade de se criar uma classe para palestrante ou para o aplicador de um minicurso, as informações de quem está dando a palestra ou administrando um minicurso ficarão inclusas dentro das respectivas classes, tais como Nome, E-mail e Telefone. As outras informações dizem respeito ao evento, assunto, duração e preço, caso seja minicurso.

8. Visão de Dados

A Visão de Dados, explicita aqui através da visão Modelo Entidade Relacionamento (MER), demostra a visão básica do Banco de Dados, a partir dos seus relacionamentos.

A Entidade **Usuário**, pode ser um Estagiário, Professor ou Participante, e os dados sao preenchidos de acordo com sua função. Já o **Minicurso** e **Palestra**, que ainda estão separados para permitir uma melhor visibilidade, tem como diferença o custo do MiniCurso. Em uma visão Modelo Relacional (MR) estas duas entidades seriam uma só, denominada **Evento**.

O **Participante** pode fazer a inscrição em um dos eventos, e a partir disso configura a **Situação**. A **Presença**, que controlará a Emissão de Certificados, será ligada através do relacionamento **Inscrição**.




projetodesoftware.participante
idparticipante : int(11)
nome : varchar(255)
ra : varchar(40)
email : varchar(255)
projetodesoftware.inscricao
idinscricao : int(11)
statusinscricrao : varchar(255)
tipo : varchar(255)
palestra_idpalestra : int(11)
minicurso_idminicurso : int(11)
participante_idparticipante : int(11)
projetodesoftware.minicurso
idminicurso : int(11)
assunto : varchar(255)
duracao : varchar(255)
preco : double

9. Tamanho e Desempenho

O sistema deve ser capaz de atualizar os dados simultaneamente de acordo com a necessidade, já que é possível haver mais de 1 evento por vez.

O sistema deve ser capaz de aguentar todos os usuários logados para ver a agenda.

O tempo de resposta para inscrição de eventos não necessita de total prioridade já que estes serão feitos com antecedência, porém, para atualizar as informações sobre a presença dos participantes o tempo deve ser mínimo para não gerar muitas filas.

10. Qualidade

O software não deve mostrar falhas graves para não interferir no funcionamento da SATI, porém, falhas não constantes e de baixa gravidade são toleráveis.

O software não precisará receber atualizações durante sua utilização, apenas o banco de dados deverá ser atualizado e este precisa ser capaz de atualizar mais de 1 informação ao mesmo tempo.

O software deverá ser compatível com dispositivos mobile, para acesso à agenda para visualizar a programação, e para computadores para tanto a visualização da programação por parte dos participantes quanto para os administradores do evento poderem cadastrar novos eventos ou usuários.

11. Codificação

Luciano
Codificar classes dos eventos

```
/*
 * Lucas G. Gonsalves - RA 1590294
 * Luciano Santos - RA 1589490
 * lucianosds@gmail.com
 *
 *
 */
package eventos;

/**
 *
 * @author Luciano Santos <lucianosds@gmail.com>
 */
public class Eventos {

	protected int idEventos;
	protected String nomeEventos;
	protected String descEventos; // Descrição do conteúdo dos eventos
	protected String tipoEventos;
	protected String dataInicioEventos;
	protected String duracaoEventos;

	public Eventos(int ide, String ne, String de, String te, String die, String
due){
	idEventos = ide;
	nomeEventos = ne;
	descEventos = de;
	tipoEventos = te;
	dataInicioEventos = die;
	duracaoEventos = due;
	}

	public Eventos(){

	}


}
```

Lucas
Codificar classe inscrição + telas