FACULDADE DO VALE DO ITAJAÍ - MIRIM – FAVIM

UNIASSELVI - ASSEVIM

CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

**SIHO – SISTEMA DE HORÁRIOS**

Projeto de iniciação científica desenvolvido através de bolsa de pesquisa pelo curso de Sistemas de Informação. Centro Universitário Leonardo da Vinci – UNIASSELVI/ ASSEVIM.

Orientador: Prof. Ricardo Ferraro de Souza

BRUSQUE

2016

# Resumo

<ESCREVER RESUMO>

# Introdução

Sistema de Horários será uma aplicação produzida para plataforma Web, onde as principais linguagens de desenvolvimento utilizadas serão PHP (Hypertext Preprocessor) e Java Script. O banco de dados adotado para este projeto será o banco de dados Postgresql.

Este sistema poderá ser utilizado no gerenciamento de horários, tais como: cadastrar grades de horário, bem como inserir cursos e permitir que o usuário selecione as disciplinas que deseja cursar durante o semestre; e ao final enviar um e-mail contendo as disciplinas que o aluno pretende frequentar.

Neste projeto, com o objetivo de organizar o desenvolvimento e o código do sistema, será utilizado o padrão de arquitetura de software MVC (Model – View - Controller), que tem por objetivo a separação das tarefas, facilitando futuras manutenções no código; e o padrão de persistência de dados DAO (Data Access Object), que permite separar as regras de negócio das regras de acesso ao banco de dados.

# Justificativa

O sistema de horários será desenvolvido com o objetivo inicial de minimizar e dinamizar o esforço dos coordenadores de curso no ajuste de disciplinas da grade do aluno no sistema de gestão da IES, bem como diminuir a quantidade de alunos que aguarda para conversar com o coordenador para arrumar seus horários, possibilitando este aluno estar dentro de sala de aula.

# Objetivos

O objetivo principal deste projeto é apresentar uma ferramenta que tenha a funcionalidade de acesso online de estudantes para um ambiente, onde ele poderá selecionar disciplinas do seu curso, bem como de outros cursos para montar sua grade de horários e ao final enviar para o coordenador, por e-mail, suas escolhas para o semestre.

# Desenvolvimento

O sistema de horário fará toda parte administrativa de horários da faculdade envolvendo organização das disciplinas concentrando em apenas um sistema.

O sistema deve permitir o gerenciamento (Cadastro, alteração, Consulta e Exclusão) de usuários.

O sistema deve permitir o gerenciamento (Cadastro, alteração, Consulta e Exclusão) de alunos e professores.

O sistema deve permitir o gerenciamento (Cadastro, alteração, Consulta e Exclusão) de Disciplinas.

O sistema deve permitir o gerenciamento (Cadastro, alteração, Consulta e Exclusão) de Cursos.

O sistema deve permitir o gerenciamento (Cadastro, alteração, Consulta e Exclusão) de Grades.

O SIHO terá uma tela de login e senha para que os usuários possam acessar ao sistema, a entrada e saída serão registradas, em logs no banco de dados, inclusive as operações realizadas em todo o sistema, tanto Cadastro, alteração, Consulta e Exclusão.

Será possível após os cadastros das grades serem efetuados, cursos e disciplinas, o controle em um painel de gerenciamento de disciplinas, que poderão ser escolhidos as disciplinas dinamicamente, portanto quando marcadas serão transferidas para uma tela de envio de email as disciplinas escolhidas, que poderão ser enviadas por email as disciplinas para o coordenador do curso, para então ele poder fazer a validação da grade semestral.

As disciplinas terão seus status, de disponíveis ou não disponíveis em uma determinada grade, exemplo de escolha de uma disciplina que não faça parte da grade daquele semestre, será validado no sistema se ela está disponível para envio ou não.

## Requisitos de funcionais

## Requisitos Não Funcionais

A aplicação será em plataforma Web;

A linguagem principal a ser desenvolvida será PHP;

O código será documentado;

A instituição de ensino ASSEVIM – Brusque torna-se proprietária dos direitos de uso do presente sistema a ser desenvolvido

## Regras de Negócio

## Modelo de Casos de Uso

## Diagrama de Classes

## SQL do projeto

#Tabela classeusuario

CREATE TABLE classeusuario (

classeid INTEGER AUTO\_INCREMENT,

classenome VARCHAR(45) NULL,

PRIMARY KEY(classeid)

);

#Tabela Turma

CREATE TABLE turma (

turmaid INTEGER AUTO\_INCREMENT,

turmanome VARCHAR(60) NULL,

PRIMARY KEY(turmaid)

);

#Tabela Modalidade

CREATE TABLE modalidade (

modalidade INTEGER NULL AUTO\_INCREMENT,

modalidadenome INTEGER NULL,

PRIMARY KEY(modalidade)

);

#Tabela Login

CREATE TABLE Login (

usuarioid INTEGER NULL AUTO\_INCREMENT ,

usuario VARCHAR NULL,

senha VARCHAR NULL,

PRIMARY KEY(usuarioid)

);

#Tabela Log

CREATE TABLE log (

idlog INTEGER NULL AUTO\_INCREMENT,

usuarioid INTEGER NULL,

loghora TIME NULL,

logdata DATE NULL,

logtexto VARCHAR(500) NULL,

PRIMARY KEY(idlog)

);

#Tabela Curso

CREATE TABLE curso (

cursoid INTEGER NOT NULL AUTO\_INCREMENT,

cursonome VARCHAR(50) NULL,

cargahoraria VARCHAR(20) NULL,

ementa VARCHAR(150) NULL,

bibliografia VARCHAR(200) NULL,

modocurso VARCHAR(20) NULL,

origemcurso VARCHAR(50) NULL,

situacao BOOLEAN NULL,

PRIMARY KEY(cursoid)

);

#Tabela disciplina

CREATE TABLE disciplina (

disciplinaid INTEGER AUTO\_INCREMENT,

turmaid INTEGER NULL,

modalidadeid INTEGER NULL,

nome VARCHAR(80) NULL,

professor VARCHAR(60) NULL,

cargahoraria VARCHAR(20) NULL,

datacadastro DATE NULL,

conceito VARCHAR(100) NULL,

ementa VARCHAR(80) NULL,

datainicio DATE NULL,

situacaodisciplina BOOLEAN NULL,

universidade VARCHAR(50) NULL,

PRIMARY KEY(disciplinaid),

CONSTRAINT disciplina\_fkey\_modalidade FOREIGN KEY(modalidadeid) REFERENCES modalidade(modalidade),

CONSTRAINT disciplina\_fkey\_turma FOREIGN KEY(turmaid) REFERENCES turma(turmaid)

);

#Tabela Grade

CREATE TABLE Grade (

gradeid INTEGER NULL AUTO\_INCREMENT,

turmaid INTEGER NULL,

usuarioid INTEGER NULL,

alunoid INTEGER NULL,

cursoid INTEGER NULL,

semestreano VARCHAR(50) NULL,

cargahoraria VARCHAR(20) NULL,

diasemana VARCHAR(20) NULL,

disciplinaid INTEGER NULL,

datavalidade DATE NULL,

datacadastro DATE NULL,

PRIMARY KEY(gradeid),

CONSTRAINT grade\_fkey\_usuario FOREIGN KEY(usuarioid) REFERENCES login(usuarioid),

CONSTRAINT grade\_fkey\_turma FOREIGN KEY(turmaid) REFERENCES turma(turmaid)

);

#Tabela gradecurso

CREATE TABLE gradecurso (

gradeid INTEGER NULL AUTO\_INCREMENT,

cursoid INTEGER NULL AUTO\_INCREMENT,

PRIMARY KEY(gradeid, cursoid),

CONSTRAINT gradecurso\_fkey\_grade FOREIGN KEY(gradeid) REFERENCES grade(gradeid),

CONSTRAINT gradecurso\_fkey\_curso FOREIGN KEY(cursoid) REFERENCES curso(cursoid)

);

#Tabela Disciplina

CREATE TABLE gradedisciplina (

gradeid INTEGER NULL AUTO\_INCREMENT,

disciplinaid INTEGER NULL AUTO\_INCREMENT,

PRIMARY KEY(gradeid, disciplinaid),

CONSTRAINT gradedisciplina\_fkey\_grade FOREIGN KEY(gradeid) REFERENCES disciplina(disciplinaid),

CONSTRAINT gradedisciplina\_fkey\_disciplina FOREIGN KEY(disciplinaid) REFERENCES disciplina(disciplinaid)

);

#Tabela Aluno

CREATE TABLE aluno (

Alunoid INTEGER NULL AUTO\_INCREMENT,

turmaid INTEGER NULL,

usuarioid INTEGER NULL,

classeid INTEGER NULL,

nome VARCHAR(60) NULL,

cpf VARCHAR(20) NULL,

rg VARCHAR(14) NULL,

endereco VARCHAR(70) NULL,

email VARCHAR(70) NULL,

cidade VARCHAR(35) NULL,

bairro VARCHAR(45) NULL,

estado VARCHAR(2) NULL,

telefone VARCHAR(20) NULL,

celular VARCHAR(20) NULL,

cep VARCHAR(10) NULL,

estadocivil VARCHAR(20) NULL,

datacadastro DATE NULL,

grauescolaridade VARCHAR(50) NULL,

disciplinaextra BOOLEAN NULL,

naturalidade VARCHAR(40) NULL,

PRIMARY KEY(Alunoid, turmaid),

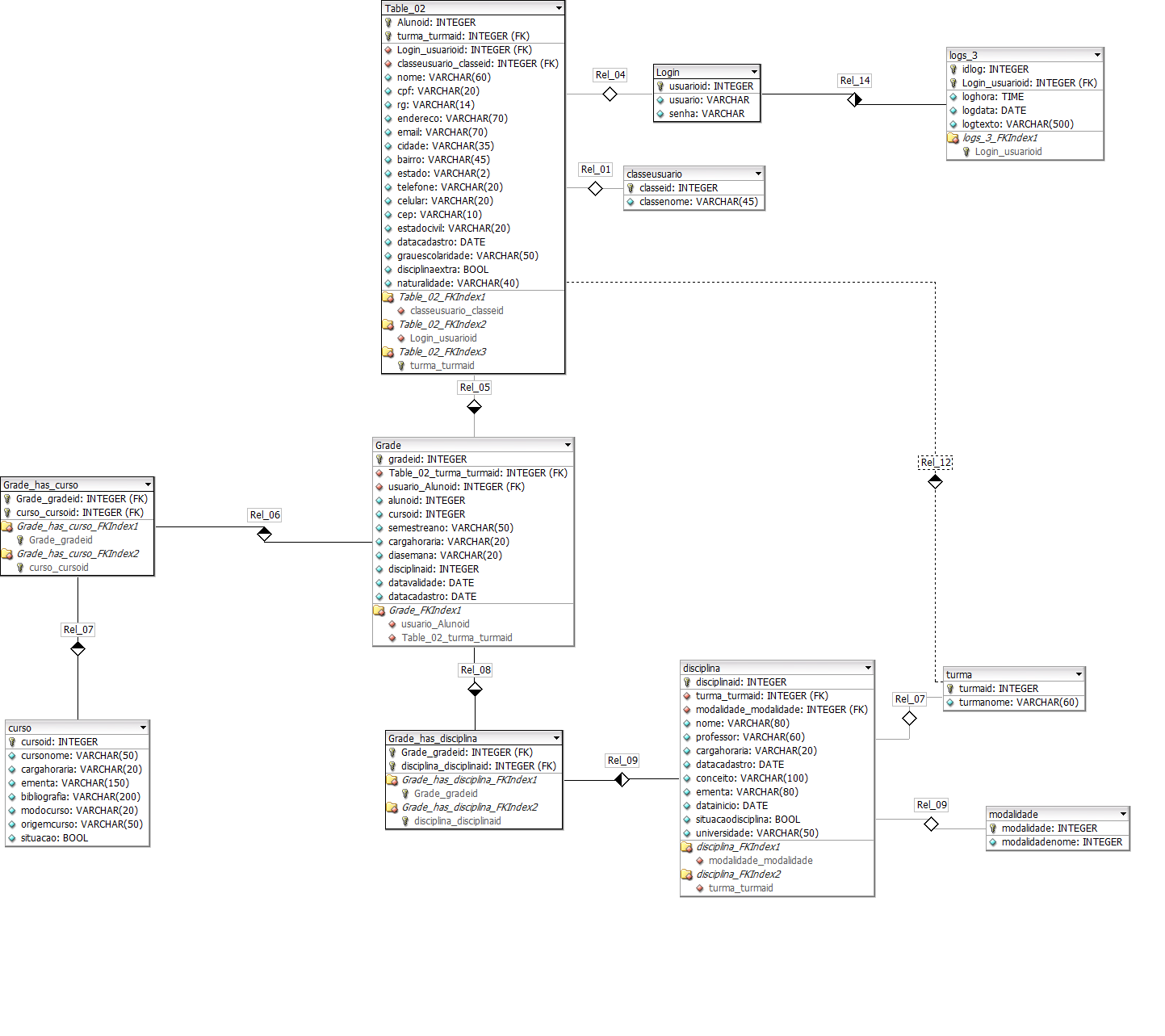
CONSTRAINT aluno\_fkey\_login FOREIGN KEY(usuarioid) REFERENCES login(usuarioid),

CONSTRAINT aluno\_fkey\_classe FOREIGN KEY(classeid) REFERENCES classeusuario(classeid),

CONSTRAINT aluno\_fkey\_turma FOREIGN KEY(turmaid) REFERENCES turma(turmaid)

);

## Modelo Entidade Relacionamento

O modelo entidade relacionamento representa a estrutura da base de dados do sistema.

# Forma de Análise dos Resultados

# Plano de Trabalho e Cronograma

O levantamento bibliográfico, analise de requisitos, desenvolvimento, implementação, implantação e testes utilizaram o modelo interativo de desenvolvimento e seguiram o seguinte cronograma:

A IDE que está sendo utilizada para desenvolver o projeto é o NETBEANS 8.0.2. A linguagem de programação padrão escolhida para o projeto foi o PHP (Hypertext Preprocessor). O banco de dados que está sendo utilizado é o MYSQL.

Foram utilizadas 40 horas para estudar o paradigma de programação Orientado a Objetos com PHP, através de vídeo aulas e apostilas. W3 SCHOOLS. Disponível em: < http://www.w3schools.com/php/default.asp >. Acesso em 01 de fevereiro até 30 de julho.

Foram utilizadas 30 horas para o estudo da linguagem SQL. W3 SCHOOLS. Disponível em: < http://www.w3schools.com/sql/default.asp >. Acesso em 01 de fevereiro até 30 de julho.

Foram Utilizadas 10 horas para o estudo do Framework Botstrap, para melhorar o designer do projeto. W3 SCHOOLS. Disponível em: <http://www.w3schools.com/bootstrap/default.asp>. Acesso em 01 de fevereiro até 30 de julho.

Foram gastos 10 horas para realizar a análise do projeto em geral, como reunião com o coordenador, pesquisa em sites sobre o assunto, escolha da IDE e também da ferramenta CASE para desenvolver o MER, entre outras atividades diversas.

Utilizado 10 horas para a realização do Modelo de entidade e relacionamento, do sistema SIHO, para definição das entidades do projeto, e também os relacionamentos que terão entre si, em sua totalidade.

# Referências Bibliográficas

# Fonte: W3 Schools <http://www.w3schools.com/php/default.asp>

# Fonte: SchoolOfNet

# http://schoolofnet.com