Desenvolver Gerenciamento de ideias

Existem dois formulários para ideia, ideia simples e melhor ideia

Ideia simples:

O cadastro de ideia simples deve conter os atributos:

* ID
* Título
* Campanha (melhor ideia ou melhoria simples)
* Grupo
* Status (array ?)
* Descrição
* Benefícios/ganhos
* Onde fazer
* Setor executor
* Investimento
* Anexos

Melhor ideia:

O cadastro de melhor ideia deve conter os atributos:

* ID
* Campanha (melhor ideia ou melhoria simples)
* Título
* Grupo
* Status (array ?)
* Descrição
* Benefícios/ganhos
* Onde fazer
* Setor executor
* Investimento
* Anexos
* PROBLEMA
* COMO ERA ANTES
* CALCULO ANTES
* SUGESTÃO DE MELHORIA
* COMO FICOU
* CALCULO DEPOIS
* GANHO
* GANHOS AMBIENTAIS
* GANHOS EM SEGURANÇA
* ENVOLVIMENTO DO GRUPO
* GLOSSÁRIO

**Desenvolver Gerenciamento de grupos:**

**O cadastro de grupos deve conter os atributos:**

* **Nome do grupo**
* **setor do grupo**
* **e-mail do grupo**
* **usuário**

**foto**

* **ativo**
* **~~gestor do grupo~~** **~~(????????) ESTARA VINCULADO AO SETOR~~**

Desenvolver Gerenciamento de Funcionários

O cadastro de funcionários deve conter os atributos:

* Nome
* Documento
* Setor
* Grupo
* ativo
* ~~Tipo de colaborador (comum, gestor, líder, relator)~~

Desenvolver Gerenciamento de Áreas Executoras (setor)

O cadastro de áreas executoras (setor) deverá conter os atributos:

* Setor – fk com tabela setores
* Gestor – fk com gestor
* É área executora?
* ativo

Desenvolver Gerenciamento de Usuários

O cadastro de usuários deverá conter os atributos:

* Usuário fk com os grupos e com funcionários que são gestores
* Senha
* Ativo
* Tipo (grupo, gestor, admin)

Módulo de aprovação de ideias

* Aprovadores

Gestor

validadeInicial

validadeFinal

* Pedidos de aprovação

Id ideia

Gestor – com quem está pendente

Data solicitação aprovação

Data limite aprovação

Módulo de controle de frequência

* Frequência

Colaborador

dataReuniao

presente - boolean

Módulo de gerenciamento de recompensas

* Recompensas

Grupo

Ideia

Quantidade de libras da ideia

Campanha da ideia

Data da libra

create database ideiassol;

use ideiassol;

create table department (

departmentId integer primary key auto\_increment not null,

name varchar (50) not null,

manager\_id integer not null,

executioner tinyint not null,

active tinyint not null

);

create table groupSol (

groupId integer primary key auto\_increment not null,

name varchar (30),

departmentId integer not null,

email varchar(50) not null,

avatar blob,

active tinyint not null,

foreign key (departmentId) references department (departmentId)

);

create table employee(

employeeId integer primary key auto\_increment not null,

name varchar (100) not null,

document integer(11) not null,

departmentId integer not null,

groupId integer not null,

active tinyint not null,

type varchar (10) not null,

foreign key (departmentId) references deparment (departmentId),

foreign key (groupId) references groupSol(groupId)

);

create table groupsMembers (

memberId integer primary key not null,

groupId integer not null,

type varchar (10),

foreign key (groupId) references groupSol(groupId),

foreign key (memberId) references employee(employeeId)

);

create table users (

userId integer primary key auto\_increment not null,

user varchar (20) not null,

password varchar (200) not null,

active tinyint not null,

type varchar(20) not null,

foreign key (userId) references groupsSol(groupId),

foreign key (userId) references employee(employeeId)

);

create table approvers (

approverId integer primary key auto\_increment not null,

initialValidity date not null,

finalValidity date not null,

foreign key (approverId) references employee(employeeId)

);

create table employeeFrequency (

employeeId integer not null,

meetingDate date not null,

attendance tinyint not null,

primary key (employeeId, meetingDate),

foreign key (employeeId) references employee (employeeId)

);

create table ideias (

ideiaId integer primary key auto\_increment not null,

campaign tinyint not null,

title varchar (250) not null,

group\_id int not null,

status varchar(25) not null,

description varchar (4000) not null,

benefits varchar (4000) not null,

whereToDo varchar (4000) not null,

performer\_id int not null,

investment varchar (2000) not null,

attachments blob,

problem varchar (4000),

howWas varchar (4000),

calculationBefore varchar (4000),

suggestion varchar (4000),

howWasAfter varchar (4000),

calculationAfter varchar (4000),

gains varchar (4000),

hasEnvironmentalGains tinyint not null,

environmentalGains varchar (4000),

hasSecurityGains tinyint not null,

securityGains varchar (4000),

groupInvolvement varchar (4000),

glossary varchar (4000)

);

create table approvalRequests (

ideiaID integer not null,

approverId integer not null,

date date not null,

deadline date not null,

foreign key (ideiaID) references ideias (ideiaID),

foreign key (approverId) references approvers(approverId)

);

/\*

create table campain (

);

create table betterMonthIdeia (

);

create table betterYearIdeia (

); \*/