

Desenvolvimento de um site para a solução proposta

SoftDesk Sys

Documento de especificação do Projeto Integrador
Módulo 5

Professor: Ivan L. Süptitz

Data: 04/05/2021

Nomes dos participantes do time:

Braian Wegmann Schuster

Darlei Juliano Cardoso

Bruno Dubal Hoffmann

Luiz Flavio de Oliveira

Jocemar Hoppe Pereira

Este documento contém a especificação das atividades relacionadas ao Projeto Integrador – Módulo 5, a serem realizadas pelos times de estudantes.

Trata-se da continuação do desenvolvimento dos projetos já realizados. Desta vez focando nas disciplinas de Programação Back-End I e Font-End I voltada aos estudantes do curso de Análise e Desenvolvimento de Sistemas – modalidade EAD.

Versionamento do documento

| Versão | Data | Responsável | Observações |
|--------|------------|-------------------------|--|
| 1.2 | 05/04/2021 | Bruno, Braian | desenhado os wireframes das telas do sistema |
| 1.1 | 03/04/2021 | Luiz Flavio, Jocemar | decidido nome do sistema, construído paleta de cores e feito logo. |
| 1.0 | 20/03/2021 | braian , Darlei | feito a <i>Atividade 1 (modelo ER, Diagrama de classes e a Contextualização)</i> |

SUMÁRIO

| | | |
|-----|---------------------------|----|
| 1. | ATIVIDADE 1 (10% da nota) | 5 |
| 1.1 | Contextualização: | 5 |
| 1.2 | Modelo de dados: | 6 |
| 1.3 | Diagrama de classes: | 7 |
| 2. | ATIVIDADE 2 (10% da nota) | 8 |
| 2.1 | Nome do site: | 8 |
| 2.2 | Paleta de cores: | 9 |
| 2.3 | Logotipo: | 10 |
| 3. | ATIVIDADE 3 (10% da nota) | 11 |
| 3.1 | Wireframes: | 11 |

Nas disciplinas e projetos integradores dos semestres anteriores, seu grupo propôs um sistema de software para uma situação de uma determinada organização, a qual vocês identificaram os objetivos e aprenderam sobre Gestão do Conhecimento (nas disciplinas de

Inteligência em Processos e Empreendedorismo e Inovação). Também fizeram análise e modelagem deste sistema nas disciplinas de Projeto de Software e Projeto de Banco de Dados.

Neste projeto integrador passaremos para uma fase mais prática na qual vamos iniciar o desenvolvimento de um site para atender algum processo da organização escolhida.

Observe que não precisamos desenvolver o sistema inteiro nesta etapa. Vamos definir um processo que seja atendido sob a forma de um site e cujo escopo seja adequado ao tempo deste módulo.

1. ATIVIDADE 1 (10% da nota)

1.1 Contextualização: Este projeto é para a empresa fictícia SoftDesk, esta empresa atua com abertura de chamados internos para pequenas/médias empresas. O site terá como objetivo gerenciar os chamados abertos tanto pelo usuário que abriu os chamados, quanto pelos técnicos que atenderam os chamados. Exemplo de uso do sistema: Uma empresa contratou a SoftDesk para implementar um sistema de chamados para a TI da mesma. Caso uma impressora estrague no setor administrativo, é aberto um chamado para que algum técnico venha dar uma olhada.

1.2 Modelo de dados:

1.2.1 Usuário: Tabela onde serão guardados os usuários, podendo ser do tipo técnico ou comum.

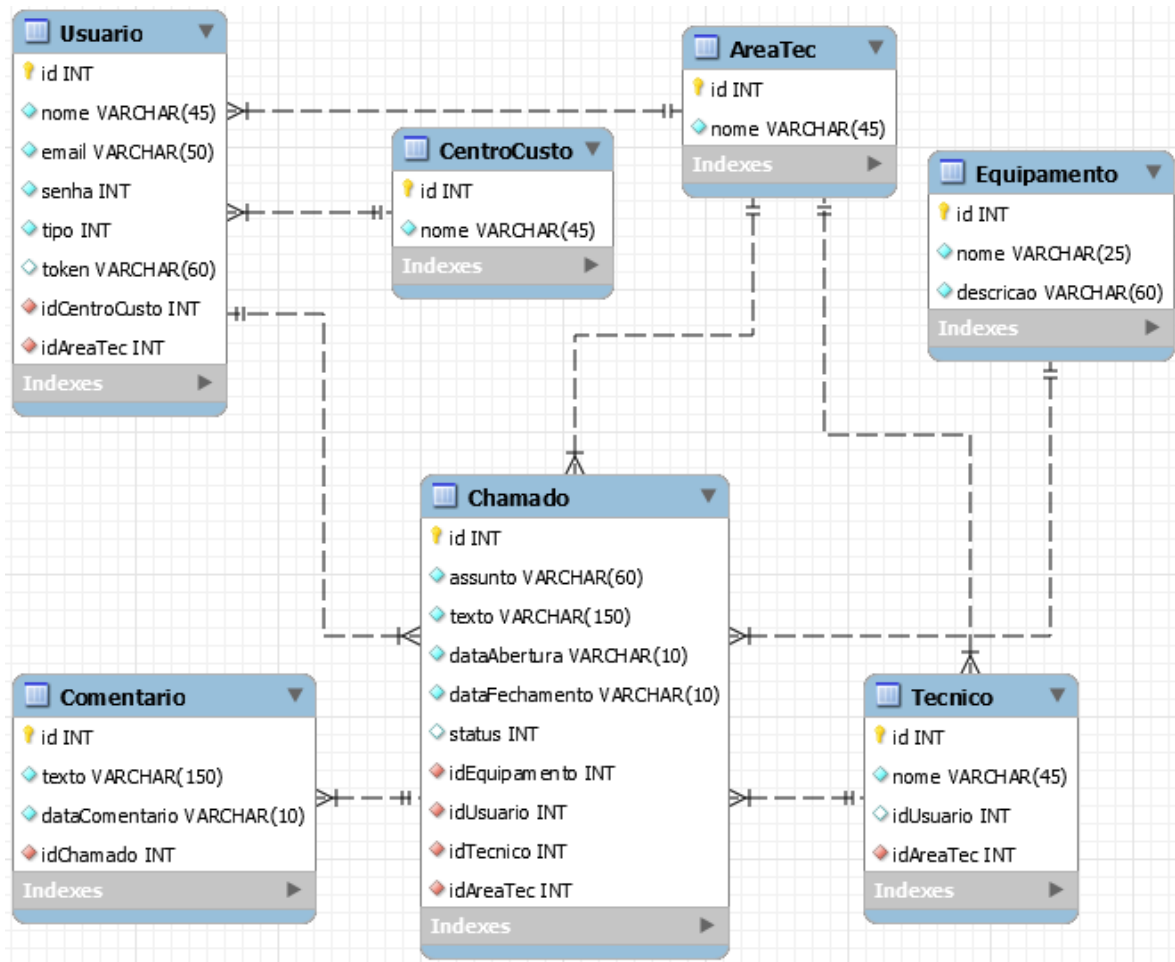
1.2.2 CentroCusto: Armazena os centros de custo de uma empresa Ex.: Financeiro, Almoxarifado, TI.

1.2.3 AreaTec: Armazena as áreas técnica do chamado Ex: Computacional, Limpeza, Administrativo, etc.

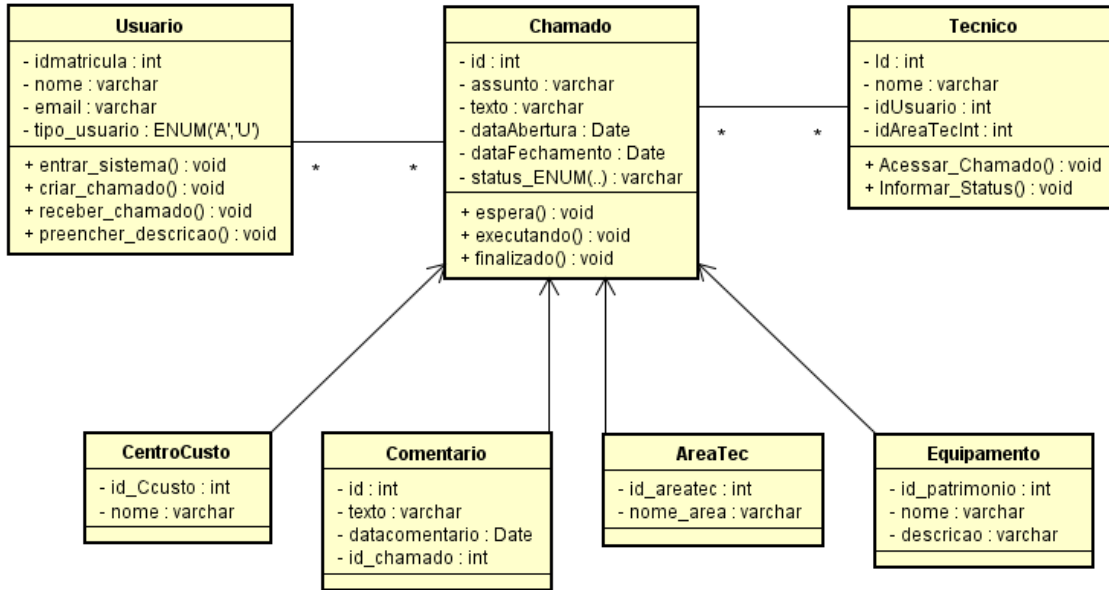
1.2.4 Equipamento: Armazena o possível equipamento que estragou ou que acabou a matéria prima.

1.2.5 Comentário: Armazena os comentários de cada chamado, um histórico do que o técnico e usuário interagiram com o sistema.

1.2.6 Técnico: Armazena a relação do usuário com a sua área técnica



1.3 Diagrama de classes:



1.4 Divisão dos trabalhos:

Atividade 1 : Braian

Atividade 2: Luiz Flávio

Atividade 3: Bruno

Atividade 4: Darlei

Atividade 5: Braian, Jocemar, Luiz, Bruno, Darlei

Atividade 6: Braian, Jocemar, Luiz, Bruno, Darlei

2. ATIVIDADE 2 (10% da nota)

2.1 Nome do site:



2.2 Paleta de cores:

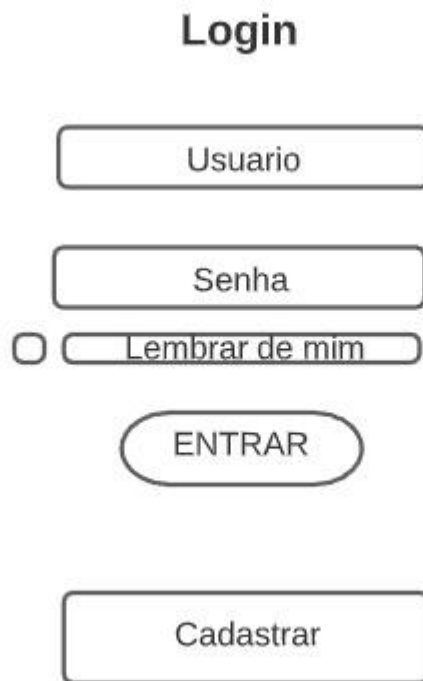


2.3 Logotipo: O logo foi escolhido algo simples e intuitivo, foi usado a imagem para ilustrar um atendente que irá receber o chamado e o nome do sistema.



3. ATIVIDADE 3 (10% da nota)

3.1 Wireframes:



A wireframe for a login form. It features a title 'Login' at the top. Below the title are three input fields: 'Usuario', 'Senha', and 'Lembrar de mim'. The 'Lembrar de mim' field is preceded by a radio button. Below these fields is a button labeled 'ENTRAR'. At the bottom of the form is a button labeled 'Cadastrar'.

Login

Usuario

Senha

☐ Lembrar de mim

ENTRAR

Cadastrar

CADASTRO DE USUARIO

NOME

USUARIO

EMAIL

SENHA

TIPO USUARIO

| Logo | | Nome usuario | |
|--------------------------|---------------|---|----------|
| Menu V | Serviços V | Chamados V | Perfil V |
| V | Assunto | <div>Descrição</div> <div><hr/><hr/><hr/><hr/><hr/></div> | |
| V | Item | | |
| V | Situação item | | |
| <div>Abrir Chamado</div> | | | |
| Rodapé | | | |

| | | | | |
|---------------|------------|--------------|-------------|----------|
| Logo | | Nome usuario | | Logout |
| Menu V | Serviços V | Chamados V | Perfil V | |
| MEUS CHAMADOS | | | | |
| ID | Situação | Descrição | Data inicio | Data fim |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| Rodapé | | | | |

4. ATIVIDADE 4 (20% da nota)

4.1 Desenvolvimento das páginas HTML

Layout geral contendo Menu e rodapé e cabeçalho:

[illegible]

Chamados

SoftDesk Sys Chamados Areas Tecnicas Equipamentos Tecnicos Usuarios Log in

Criar Chamados

| ID | ASUNTO | DATA ABERTURA |
|----|--------|--------------------|
| 1 | fgfdg | 1 de Abril de 2021 |
| 2 | aaaaa | 2 de Abril de 2021 |
| 3 | fgdfgf | 7 de Abril de 2021 |
| 4 | AAAAAA | 8 de Abril de 2021 |

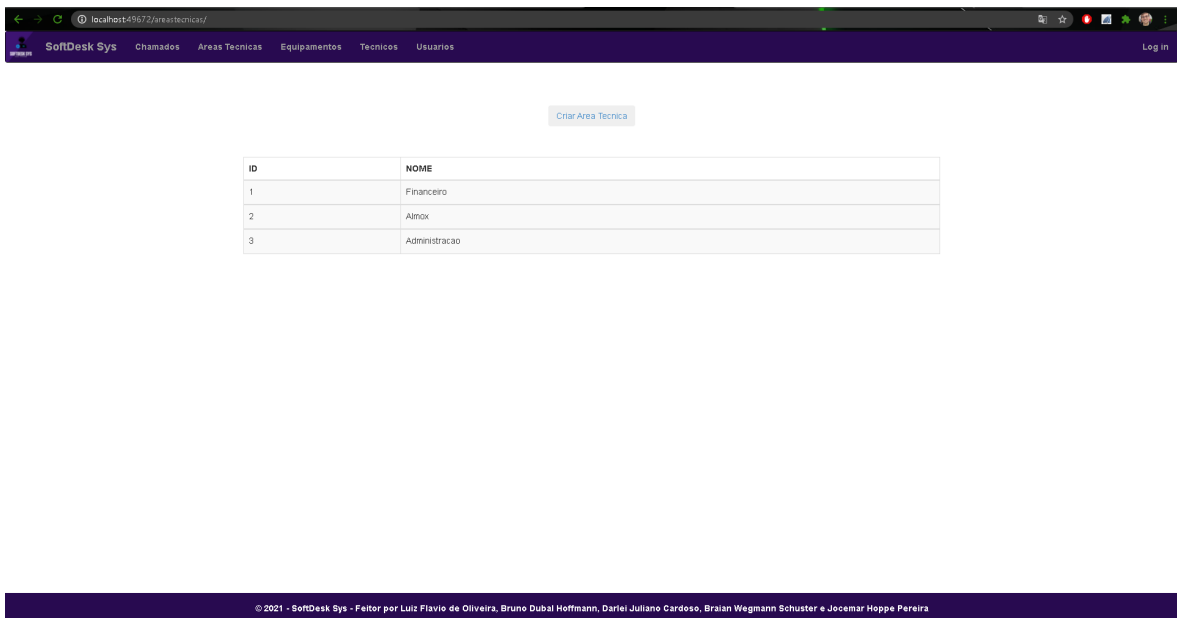
© 2021 - SoftDesk Sys - Feito por Luiz Flavio de Oliveira, Bruno Dubal Hoffmann, Darlei Juliano Cardoso, Braizian Wegmann Schuster e Jocemar Hoppe Pereira

```

{% extends "app/layout.html" %}

{% block content %}
<div style="text-align:center; padding:50px">
  <button class="btn"><a href="{% url 'addchamados' %}">Criar Chamados</a></button>
</div>
<div>
  <table class="table table-striped table-bordered">
    <thead class="thead-dark">
      <tr>
        <th scope="col">ID</th>
        <th scope="col">ASUNTO</th>
        <th scope="col">DATA ABERTURA</th>
      </tr>
    </thead>
    <tbody>
      {% for chamado in chamados %}
      <tr style="height:40px">
        <td>{{ chamado.id_chamado }}</td>
        <td>{{ chamado.assunto }}</td>
        <td>{{ chamado.data_abertura }}</td>
      </tr>
      {% endfor %}
    </tbody>
  </table>
</div>
{% endblock %}

```




```

models.py  forms.py  views.py  urls.py  layout.html  addtecnicos.html  equipamentos.html  tecnicos.html  areastecnicas.html  addareatec
{% extends "app/layout.html" %}

{% block content %}


SoftDesk Sys Chamados Areas Tecnicas Equipamentos Tecnicos Usuarios Log in



Criar Equipamento



| ID | NOME          | PATRIMONIO |
|----|---------------|------------|
| 1  | Equipamento 1 | 23423      |
| 2  | Equipamento 2 | 56446      |

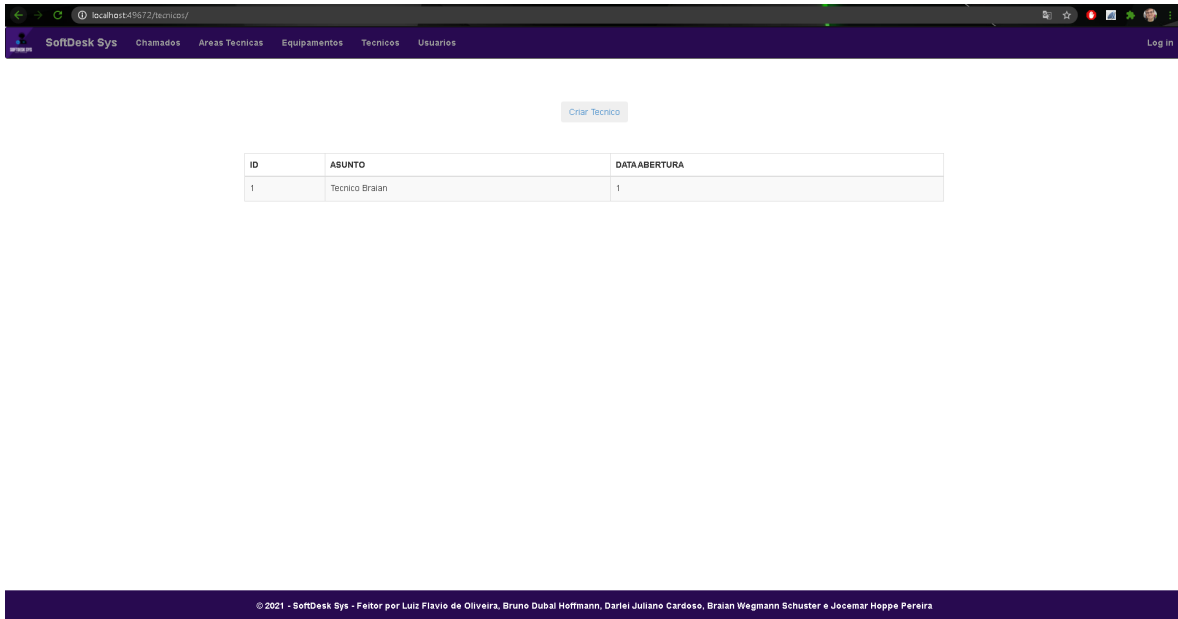


© 2021 - SoftDesk Sys - Feito por Luiz Flavio de Oliveira, Bruno Dubal Hoffmann, Darlei Juliano Cardoso, Braizian Wegmann Schuster e Joceimar Hoppe Pereira



17


```



```

<div style="text-align:center; padding:50px">
  <button class="btn"><a href="{% url 'addtecnicos' %}">Criar Tecnico</a></button>
</div>
<div>
  <table class="table table-striped table-bordered">
    <thead class="thead-dark">
      <tr>
        <th scope="col">ID</th>
        <th scope="col">ASUNTO</th>
        <th scope="col">DATA ABERTURA</th>
      </tr>
    </thead>
    <tbody>
      {% for tecnico in tecnicos %}
      <tr style="height:40px">
        <td>{{ tecnico.id_tecnico }}</td>
        <td>{{ tecnico.nome }}</td>
        <td>{{ tecnico.id_area_tec }}</td>
      </tr>
      </tbody>
    </table>
    {% endfor %}
  </div>

{% endblock %}

```

5. ATIVIDADE 5 (20% da nota)

As atividades 5 e 6 estão relacionadas com a disciplina de Programação Back-End (que trata de tudo que ocorre no lado do servidor). Ao realizá-las faremos o trabalho de um programador web, ou programador back-end.

5.1 Classes de dados

```

# Comentario do Braian
# Criem os model aqui das tabelas que forem necessarias
# primeiro precisa ser descrito como sera a classe/tabela abaixo
# depois rodar abrir um terminal (ctrl + `) e rodar o comando python .\manage.py makemigrations
# rodar o comando python .\manage.py migrate
# e pronto a tabela tera sido criada
# proximo passo criar o form no arquivo forms.py

STATUS_CHOICES = (
    (1, 'Aberto'),
    (2, 'Em processamento'),
    (3, 'Cancelado'),
    (4, 'Concluido'),
)

TYPE_USER_CHOICES = (
    (1, 'Comum'),
    (2, 'Tecnico'),
    (3, 'Admin'),
)

class Chamado(models.Model):
    id_chamado = models.AutoField(primary_key = True)
    assunto = models.CharField(max_length = 255, null=True)
    texto = models.CharField(max_length = 999, null=True)
    data_abertura = models.DateField(default=timezone.now)
    data_fechamento = models.DateField(null=True)
    status = models.IntegerField(choices=STATUS_CHOICES,null=True)
    id_equipamento = models.IntegerField(null=True)
    id_usuario = models.IntegerField(null=True)
    id_tecnico = models.IntegerField(null=True)
    id_area_tec = models.IntegerField(null=True)

    class Meta:
        db_table = 'chamado'

    def __str__(self):
        return self.name

class Usuario(models.Model):
    id_usuario = models.AutoField(primary_key = True)
    nome = models.CharField(max_length = 255)
    email = models.CharField(max_length = 999)
    senha = models.CharField(max_length = 255)
    tipo = models.IntegerField(choices=TYPE_USER_CHOICES)
    token = models.CharField(max_length = 255)
    id_centro_custo = models.IntegerField()
    id_area_tec = models.IntegerField()

    class Meta:
        db_table = 'usuario'

    def __str__(self):
        return self.name

```

```
class AreaTecnica(models.Model):
    id_area_tec = models.AutoField(primary_key = True)
    area_tec = models.CharField(max_length = 45)

    class Meta:
        db_table = 'area_tecnica'

    def __str__(self):
        return self.name

class Tecnico(models.Model):
    id_tecnico = models.AutoField(primary_key = True)
    nome = models.CharField(max_length = 255)
    id_usuario = models.IntegerField(null=True)
    id_area_tec = models.IntegerField(null=True)

    class Meta:
        db_table = 'tecnico'

    def __str__(self):
        return self.name

class Equipamento(models.Model):
    id_equipamento = models.AutoField(primary_key = True)
    nome = models.CharField(max_length = 255)
    patrimonio = models.IntegerField(null=True)
    descricao = models.CharField(max_length = 999)

    class Meta:
        db_table = 'equipamento'

    def __str__(self):
        return self.name
```

6. ATIVIDADE 6 (20% da nota)

6.1 Desenvolvimento dos códigos de interação

```

"""
Definition of forms.
"""

from django import forms
from django.contrib.auth.forms import AuthenticationForm
from app.models import Chamado, Usuario, AreaTecnica, Tecnico, Equipamento
from django.db import models
from django.utils.translation import ugettext_lazy as _

class BootstrapAuthenticationForm(AuthenticationForm):
    """Authentication form which uses bootstrap CSS."""
    username = forms.CharField(max_length=254,
                               widget=forms.TextInput({
                                   'class': 'form-control',
                                   'placeholder': 'User name'
                               }))
    password = forms.CharField(label=_("Password"),
                               widget=forms.PasswordInput({
                                   'class': 'form-control',
                                   'placeholder': 'Password'
                               }))

class ChamadosForm(forms.ModelForm):
    class Meta:
        model = Chamado
        fields = ['assunto', 'texto', 'data_abertura', 'status', 'id_usuario']
        widgets = {
            'assunto': forms.TextInput(attrs={'class': 'form-control'}),
            'texto': forms.Textarea(attrs={'class': 'form-control'}),
            'data_abertura': forms.DateInput(attrs={'type': 'date', 'class': 'form-control'}),
            'status': forms.Select(attrs={'class': 'form-control'}),
            'id_usuario': forms.Select(choices = Usuario.objects.values_list('id_usuario', 'nome'), attrs={'class': 'form-control'})
        }

class AreaTecnicaForm(forms.ModelForm):
    class Meta:
        model = AreaTecnica
        fields = ['area_tec']
        widgets = {
            'area_tec': forms.TextInput(attrs={'class': 'form-control'})
        }

class TecnicoForm(forms.ModelForm):
    class Meta:
        model = Tecnico
        fields = ['nome', 'id_area_tec', 'id_usuario']
        widgets = {
            'nome': forms.TextInput(attrs={'class': 'form-control'}),
            'id_area_tec': forms.Select(choices = AreaTecnica.objects.values_list('id_area_tec', 'area_tec'), attrs={'class': 'form-control'}),
            'id_usuario': forms.Select(choices = Usuario.objects.values_list('id_usuario', 'nome'), attrs={'class': 'form-control'})
        }

class EquipamentosForm(forms.ModelForm):
    class Meta:
        model = Equipamento
        fields = ['nome', 'patrimonio', 'descricao']
        widgets = {
            'nome': forms.TextInput(attrs={'class': 'form-control'}),
            'patrimonio': forms.TextInput(attrs={'class': 'form-control'}),
            'descricao': forms.Textarea(attrs={'class': 'form-control'})
        }

```

```

addusers
"""
Definition of views.
"""

from datetime import datetime
from django.shortcuts import render, redirect
from django.http import HttpRequest
from app.forms import ChamadosForm, AreaTecnicaForm, TecnicoForm, EquipamentosForm
from app.models import Chamado, AreaTecnica, Equipamento, Tecnico

def home(request):
    """Renders the home page."""
    assert isinstance(request, HttpRequest)
    return render(
        request,
        'app/index.html',
        {
            'title': 'Home Page',
            'year': datetime.now().year,
        }
    )

def users(request):
    """Renders the contact page."""
    assert isinstance(request, HttpRequest)
    return render(
        request,
        'app/users.html',
        {
            'title': 'Usuarios',
            'message': 'Listagem de Usuarios',
            'year': datetime.now().year,
        }
    )

def addusers(request):
    """Renders the contact page."""
    assert isinstance(request, HttpRequest)
    return render(
        request,
        'app/addusers.html',
        {
            'title': 'Adicionar Usuario',
            'message': 'Criacao de Usuario',
            'year': datetime.now().year
        }
    )

```

```

def addchamados(request):
    """Renders the contact page."""
    assert isinstance(request, HttpRequest)

    form = ChamadosForm()
    if request.method == 'POST':
        form = ChamadosForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('addchamados')

    return render(
        request,
        'app/addchamados.html',
        {
            'title': 'Criar um Chamado',
            'message': 'Novo Chamado',
            'year': datetime.now().year,
            'form': form
        }
    )

def chamados(request):
    """Renders the about page."""
    assert isinstance(request, HttpRequest)
    chamados = Chamado.objects.all()
    return render(
        request,
        'app/chamados.html',
        {
            'year': datetime.now().year,
            'chamados': chamados
        }
    )

def tecnicos(request):
    """Renders the about page."""
    assert isinstance(request, HttpRequest)
    tecnicos = Tecnico.objects.all()
    return render(
        request,
        'app/tecnicos.html',
        {
            'year': datetime.now().year,
            'tecnicos': tecnicos
        }
    )

def areastecnicas(request):
    """Renders the about page."""
    assert isinstance(request, HttpRequest)
    areastecnicas = AreaTecnica.objects.all()
    return render(
        request,
        'app/areastecnicas.html',
        {
            'year': datetime.now().year,
            'areastecnicas': areastecnicas
        }
    )

```



```

def equipamentos(request):
    """Renders the about page."""
    assert isinstance(request, HttpRequest)
    equipamentos = Equipamento.objects.all()
    return render(
        request,
        'app/equipamentos.html',
        {
            'year': datetime.now().year,
            'equipamentos': equipamentos
        }
    )

def addareatecnica(request):
    """Renders the contact page."""
    assert isinstance(request, HttpRequest)

    form = AreaTecnicaForm()
    if request.method == 'POST':
        form = AreaTecnicaForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('addareatecnica')

    return render(
        request,
        'app/addareatecnica.html',
        {
            'title': 'Criação de area tecnica',
            'year': datetime.now().year,
            'form': form
        }
    )

def addtecnicos(request):
    """Renders the contact page."""
    assert isinstance(request, HttpRequest)

    form = TecnicoForm()
    if request.method == 'POST':
        form = TecnicoForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('addtecnicos')

    return render(
        request,
        'app/addtecnicos.html',
        {
            'title': 'Criar um novo Tecnico',
            'message': 'Novo Tecnico',
            'year': datetime.now().year,
            'form': form
        }
    )

```

```
def addequipamentos(request):  
    """Renders the about page."""  
    assert isinstance(request, HttpRequest)  
  
    form = EquipamentosForm()  
  
    if request.method == 'POST':  
        form = EquipamentosForm(request.POST)  
        if form.is_valid():  
            form.save()  
            return redirect('addequipamentos')  
  
    return render(  
        request,  
        'app/addequipamentos.html',  
        {  
            'title': "Adicionar Equipamento",  
            'message': 'Criação de um novo equipamento',  
            'year': datetime.now().year,  
            'form': form  
        }  
    )
```

```

models.py  forms.py  views.py  urls.py  layout.html  addtecnicos.html

"""
Definition of urls for projetointegrador3a.
"""

from datetime import datetime
from django.urls import path
from django.contrib import admin
from django.contrib.auth.views import LoginView, LogoutView
from app import forms, views

urlpatterns = [
    path('', views.home, name='home'),
    path('users/', views.users, name='users'),
    path('chamados/', views.chamados, name='chamados'),
    path('areastecnicas/', views.areastecnicas, name='areastecnicas'),
    path('equipamentos/', views.equipamentos, name='equipamentos'),
    path('tecnicos/', views.tecnicos, name='tecnicos'),
    path('addchamados/', views.addchamados, name='addchamados'),
    path('addareatecnica/', views.addareatecnica, name='addareatecnica'),
    path('addequipamentos/', views.addequipamentos, name='addequipamentos'),
    path('addtecnicos/', views.addtecnicos, name='addtecnicos'),
    path('addusers/', views.addusers, name='addusers'),
    path('login/',
        LoginView.as_view(
            (
                template_name='app/login.html',
                authentication_form=forms.BootstrapAuthenticationForm,
                extra_context={
                    'title': 'Log in',
                    'year': datetime.now().year,
                }
            ),
            name='login'),
        path('logout/', LogoutView.as_view(next_page='/'), name='logout'),
        path('admin/', admin.site.urls),
    ]

```