

Portal Bruxa Academica - P2

Nome: Luna Leão de Maria, Lucas Agüena Gatto, Victor Trindade dos Santos

Código das Classes

Classe Usuario

```
public class Usuario {  
  
    // Atributo protegido: nome de usuário (acessível pela própria  
    // classe e subclasses).  
    protected String username;  
  
    // Atributo protegido: senha do usuário.  
    protected String senha;  
  
    // Construtor da classe que inicializa os atributos username e  
    // senha.  
    public Usuario(String username, String senha) {  
        this.username = username;  
        this.senha = senha;  
    }  
  
    // Método público que retorna o nome de usuário.  
    public String getUsername() {  
        return username;  
    }  
  
    // Método público que retorna a senha.  
    public String getSenha() {  
        return senha;  
    }  
  
    // Método público que indica se o usuário é administrador.  
    // Neste caso, sempre retorna false, pois é um usuário comum.  
    public boolean isAdmin() {  
        return false;  
    }  
}
```

```
}  
  
}
```

Classe Admin

```
public class Admin extends Usuario {  
  
    // Construtor da classe Admin que chama o construtor da classe pai  
    (Usuario).  
    public Admin(String username, String senha) {  
        super(username, senha); // Passa os dados para o construtor da  
        classe Usuario  
    }  
  
    // Sobrescreve o método isAdmin da classe Usuario.  
    // Indica que este usuário é um administrador.  
    @Override  
    public boolean isAdmin() {  
        return true;  
    }  
  
}
```

Classe Artigo

```
public class Artigo {  
  
    // Atributo privado que armazena o título do artigo.  
    private String titulo;  
  
    // Atributo privado que armazena o conteúdo do artigo.  
    private String conteudo;  
  
    // Construtor que inicializa título e conteúdo do artigo.  
    public Artigo(String titulo, String conteudo) {  
        this.titulo = titulo;  
        this.conteudo = conteudo;  
    }  
}
```

```
}

// Getter para obter o título do artigo.
public String getTitulo() {
    return titulo;
}

// Getter para obter o conteúdo do artigo.
public String getConteudo() {
    return conteudo;
}

}
```

Classe BancoFake

```
public class BancoFake {

    // Lista pública e estática de usuários (compartilhada por toda a
    // aplicação)
    public static ArrayList<Usuario> usuarios = new ArrayList<>();

    // Lista pública e estática de artigos
    public static ArrayList<Artigo> artigos = new ArrayList<>();

    // Representa o usuário atualmente logado (inicialmente null)
    public static Usuario usuarioLogado = null;

    // Bloco estático executado uma vez ao carregar a classe
    static {
        // Adiciona um usuário administrador padrão à lista
        usuarios.add(new Admin("admin", "123"));
    }

}
```

Classe PortalBruxa

```
public class PortalBruxa {

    public static void main(String[] args) {
        // Executa o código da interface gráfica na thread correta (EDT
        - Event Dispatch Thread)
        javax.swing.SwingUtilities.invokeLater(() -> {
            // Cria e exibe a tela de login
            new TelaLogin().setVisible(true);
        });
    }

}
```

Classe TelaCadastrarArtigo

```
// Classe principal da tela de cadastro de artigos public class
TelaCadastrarArtigo extends javax.swing.JFrame { // Construtor da
tela

    public TelaCadastrarArtigo() {
        initComponents(); // Método que monta a interface gráfica
    }

    // Método gerado automaticamente para inicializar os componentes da
    interface gráfica
    @SuppressWarnings("unchecked")
    private void initComponents() {
        java.awt.GridBagConstraints gridBagConstraints;

        // Componente inválido gerado pelo NetBeans (ignorar ou
        substituir)
        navbar1 = new org.netbeans.modules.form.InvalidComponent();

        // Painei principal com cor de fundo e layout
        jPanel1 = new javax.swing.JPanel();
        jPanel2 = new javax.swing.JPanel(); // Painei interno branco
        onde ficam os campos
    }
}
```

```

// Labels e campos de entrada
jLabel1 = new javax.swing.JLabel(); // Título da tela "Cadastrar
Artigos"
jLabel2 = new javax.swing.JLabel(); // Label "Título"
txtTitulo = new javax.swing.JTextField(); // Campo de texto para
título
jLabel3 = new javax.swing.JLabel(); // Label "Conteúdo"
jScrollPane1 = new javax.swing.JScrollPane(); // Scroll para o
conteúdo
txtaConteudo = new javax.swing.JTextArea(); // Campo de texto
para o conteúdo (textarea)

// Botão "Criar"
btnCriarArtigo = new javax.swing.JButton();

// Configurações básicas da janela
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setMaximumSize(new java.awt.Dimension(1000, 1000));
setMinimumSize(new java.awt.Dimension(612, 100));
setPreferredSize(new java.awt.Dimension(1000, 550));
setResizable(false);

// Adiciona o navbar (componente inválido) ao topo
getContentPane().add(navbar1, java.awt.BorderLayout.PAGE_START);

// Configurações do painel principal
jPanel1.setBackground(new java.awt.Color(102, 0, 102));
jPanel1.setMaximumSize(new java.awt.Dimension(1000, 400));
jPanel1.setLayout(new java.awt.GridBagLayout());

// Configurações do painel interno
jPanel2.setBackground(new java.awt.Color(255, 255, 255));
jPanel2.setMaximumSize(new java.awt.Dimension(1000, 1000));
jPanel2.setPreferredSize(new java.awt.Dimension(700, 396));

// Título "Cadastrar Artigos"
jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 24));
jLabel1.setForeground(new java.awt.Color(204, 51, 255));
jLabel1.setText("Cadastrar Artigos");

// Label e campo para "Título"

```

```

jLabel2.setFont(new java.awt.Font("Segoe UI", 1, 18));
jLabel2.setForeground(new java.awt.Color(204, 51, 255));
jLabel2.setText("Título");

// Label e campo para "Conteúdo"
jLabel3.setFont(new java.awt.Font("Segoe UI", 1, 18));
jLabel3.setForeground(new java.awt.Color(204, 51, 255));
jLabel3.setText("Conteúdo");

// Área de texto configurada
txtaConteudo.setColumns(20);
txtaConteudo.setRows(5);
jScrollPane1.setViewportView(txtaConteudo); // Scroll vinculado
à textarea

// Botão "Criar" com ação definida
btnCriarArtigo.setBackground(new java.awt.Color(204, 51, 255));
btnCriarArtigo.setFont(new java.awt.Font("Segoe UI", 0, 14));
btnCriarArtigo.setForeground(new java.awt.Color(122, 0, 203));
btnCriarArtigo.setText("Criar");
btnCriarArtigo.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
{
        btnCriarArtigoActionPerformed(evt); // Método chamado ao
clicar no botão
    }
});

// Layout horizontal do JPanel2
javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()

```

```

        .addGap(153, 153, 153)

        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING, false)
            .addComponent(jLabel2)
            .addComponent(jLabel3)
            .addComponent(txtTitulo)
            .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 289, Short.MAX_VALUE)))
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(274, 274, 274)
            .addComponent(btnCriarArtigo))
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(198, 198, 198)
            .addComponent(jLabel1)))
        .addGap(310, 310, 310))
    );

    // Layout vertical do jPanel2
    jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(22, 22, 22)
            .addComponent(jLabel1)
            .addGap(43, 43, 43)
            .addComponent(jLabel2)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtTitulo,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(jLabel3)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addGap(40, 40, 40)
        .addComponent(btnCriarArtigo)
        .addContainerGap(44, Short.MAX_VALUE))
    );

    // Adiciona jPanel2 ao jPanel1 com GridBagLayout
    gridBagConstraints = new java.awt.GridBagConstraints();
    gridBagConstraints.gridx = 0;
    gridBagConstraints.gridy = 0;
    gridBagConstraints.gridheight = 14;
    gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
    gridBagConstraints.weightx = 1.0;
    gridBagConstraints.weighty = 1.0;
    gridBagConstraints.insets = new java.awt.Insets(0, 200, 0, 200);
    jPanel1.add(jPanel2, gridBagConstraints);

    // Adiciona o painel principal à janela
    getContentPane().add(jPanel1, java.awt.BorderLayout.CENTER);

    // Ajusta tudo ao tamanho definido
    pack();
}

// Ação do botão "Criar"
private void
btnCriarArtigoActionPerformed(java.awt.event.ActionEvent evt) {
    // Captura os dados dos campos
    String titulo = txtTitulo.getText();
    String conteudo = txtaConteudo.getText();

    // Valida se os campos estão preenchidos
    if (titulo.isEmpty() || conteudo.isEmpty()) {
        JOptionPane.showMessageDialog(this, "Preencha todos os
campos!");
        return;
    }

    // Cria novo artigo
    Artigo artigo = new Artigo(titulo, conteudo);

    // Salva o artigo no banco fake
    BancoFake.artigos.add(artigo);
}

```



```

        // Mostra mensagem de sucesso
        JOptionPane.showMessageDialog(this, "Artigo criado com
sucesso!");

        // Limpa os campos
        txtTitulo.setText("");
        txtaConteudo.setText("");
    }

    // Método principal para iniciar a interface
    public static void main(String args[]) {
        // Configura o "look and feel" da interface para Nimbus, se
disponível
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (Exception ex) {

java.util.logging.Logger.getLogger(TelaCadastrarArtigo.class.getName
()).log(java.util.logging.Level.SEVERE, null, ex);
        }

        // Exibe a interface na tela
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new TelaCadastrarArtigo().setVisible(true);
            }
        });
    }
}

```

Classe Navbar

```
public class Navbar extends javax.swing.JPanel {/**
 * Construtor da barra de navegação.
 */
public Navbar() {
    // Define o layout com espaçamento horizontal de 20px e vertical
    de 10px
    this.setLayout(new FlowLayout(FlowLayout.CENTER, 20, 10));

    // Inicializa os componentes da interface (gerado pelo NetBeans)
    initComponents();

    // Define a altura preferida da navbar
    setPreferredSize(new Dimension(Short.MAX_VALUE, 30));

    // Oculta o submenu inicialmente
    pSubMenu.setVisible(false);

    // Redimensiona o ícone do logo
    ImageIcon originalIcon = new
    ImageIcon(getClass().getResource("/logo_portal_bruxa.png"));
    Image img = originalIcon.getImage();
    Image resizedImg = img.getScaledInstance(80, 80,
    Image.SCALE_SMOOTH);
    ImageIcon resizedIcon = new ImageIcon(resizedImg);
    jLabel8.setIcon(resizedIcon);
    jLabel8.setText(""); // Remove texto do label
    jLabel8.setPreferredSize(new Dimension(60, 60)); // Define o
    tamanho do ícone

    // Exibe ou oculta botões com base no tipo de usuário logado
    if (BancoFake.usuarioLogado != null &&
    BancoFake.usuarioLogado.isAdmin()) {
        btCriar.setVisible(true); // Admin pode criar artigos
    } else {
        btCriar.setVisible(false); // Usuário comum não pode
        btVisualizar.setVisible(false); // Também não pode
        visualizar
        pSubMenu.setVisible(false); // Submenu escondido
        btArtigos.setForeground(new Color(204,51,255)); // Cor do
        botão
    }
```

```

    }
}

/**
 * Inicialização dos componentes da navbar (gerado automaticamente)
 */
@SuppressWarnings("unchecked")
private void initComponents() {

    // Labels de seções e logo
    jLabel1 = new javax.swing.JLabel();
    jLabel8 = new javax.swing.JLabel(); // Logo
    jLabel2 = new javax.swing.JLabel(); // Home
    jLabel3 = new javax.swing.JLabel(); // Magia
    jLabel4 = new javax.swing.JLabel(); // Numerologia
    jLabel5 = new javax.swing.JLabel(); // Tarot
    jLabel6 = new javax.swing.JLabel(); // Astrologia
    btArtigos = new javax.swing.JButton(); // Botão de artigos
    pSubMenu = new javax.swing.JPanel(); // Submenu com
criar/visualizar
    btVisualizar = new javax.swing.JButton();
    btCriar = new javax.swing.JButton();
    jLabel7 = new javax.swing.JLabel(); // Cursos
    btLogout = new javax.swing.JButton(); // Botão de logout

    jLabel1.setText("jLabel1"); // Placeholder

    setPreferredSize(new java.awt.Dimension(1000, 50)); // Tamanho
da navbar

    // Ícone do logo
    jLabel8.setText("jLabel8");
    add(jLabel8);

    // Adiciona os labels como se fossem links
    jLabel2.setForeground(new java.awt.Color(204, 51, 255));
    jLabel2.setText("Home");
    add(jLabel2);

    jLabel3.setForeground(new java.awt.Color(204, 51, 255));
    jLabel3.setText("Magia");
    add(jLabel3);

```

```

jLabel4.setForeground(new java.awt.Color(204, 51, 255));
jLabel4.setText("Numerologia");
add(jLabel4);

jLabel5.setForeground(new java.awt.Color(204, 51, 255));
jLabel5.setText("Tarot");
add(jLabel5);

jLabel6.setForeground(new java.awt.Color(204, 51, 255));
jLabel6.setText("Astrologia");
add(jLabel6);

// Botão de "Artigos" com ação para abrir submenu
btArtigos.setForeground(new java.awt.Color(153, 0, 204));
btArtigos.setText("Artigos ");
btArtigos.setBorder(null);
btArtigos.setBorderPainted(false);
btArtigos.setContentAreaFilled(false);
btArtigos.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
        btArtigosActionPerformed(evt);
    }
});
add(btArtigos);

// Submenu com botões de visualizar e criar artigos
pSubMenu.setBackground(new java.awt.Color(255, 255, 255));
pSubMenu.setPreferredSize(new java.awt.Dimension(100, 78));

btVisualizar.setForeground(new java.awt.Color(153, 0, 204));
btVisualizar.setText("Visualizar");
btVisualizar.setBorder(null);
btVisualizar.setContentAreaFilled(false);
btVisualizar.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
        btVisualizarActionPerformed(evt);
    }
}

```

```

});

btCriar.setForeground(new java.awt.Color(153, 0, 204));
btCriar.setText("Criar");
btCriar.setBorder(null);
btCriar.setContentAreaFilled(false);
btCriar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
{
        btCriarActionPerformed(evt);
    }
});

// Layout do submenu
javax.swing.GroupLayout pSubmenuLayout = new
javax.swing.GroupLayout(pSubmenu);
pSubmenu.setLayout(pSubmenuLayout);
pSubmenuLayout.setHorizontalGroup(

pSubmenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
    .addGroup(pSubmenuLayout.createSequentialGroup())
        .addContainerGap()

.addGroup(pSubmenuLayout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)
    .addComponent(btCriar,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(btVisualizar,
javax.swing.GroupLayout.DEFAULT_SIZE, 88, Short.MAX_VALUE))
        .addContainerGap())
);
pSubmenuLayout.setVerticalGroup(

pSubmenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
    .addGroup(pSubmenuLayout.createSequentialGroup())
        .addContainerGap()
        .addComponent(btVisualizar)
        .addGap(18, 18, 18)
        .addComponent(btCriar)

```

```

        .addContainerGap(22, Short.MAX_VALUE))
    );

    add(pSubmenu);

    // Adiciona botão "Cursos"
    jLabel7.setForeground(new java.awt.Color(204, 51, 255));
    jLabel7.setText("Cursos");
    add(jLabel7);

    // Botão de logout
    btLogout.setForeground(new java.awt.Color(153, 0, 204));
    btLogout.setText("Logout");
    btLogout.setBorder(null);
    btLogout.setContentAreaFilled(false);
    btLogout.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt)
        {
            btLogoutActionPerformed(evt);
        }
    });
    add(btLogout);
}

// Ação ao clicar em "Artigos"
private void btArtigosActionPerformed(java.awt.event.ActionEvent
evt) {
    if (BancoFake.usuarioLogado != null &&
BancoFake.usuarioLogado.isAdmin()) {
        // Alterna visibilidade do submenu se for admin
        pSubmenu.setVisible(!pSubmenu.isVisible());
    } else {
        // Esconde o submenu para usuários comuns
        pSubmenu.setVisible(false);
    }
}

// Ação ao clicar em "Logout"
private void btLogoutActionPerformed(java.awt.event.ActionEvent evt)
{
    TelaLogin login = new TelaLogin(); // Cria nova tela de login
    login.setVisible(true); // Exibe tela de login
}

```

```

        SwingUtilities.getWindowAncestor(this).dispose(); // Fecha
janela atual
    }

    // Ação ao clicar em "Visualizar"
    private void btVisualizarActionPerformed(java.awt.event.ActionEvent
    evt) {
        TelaVisualizarArtigo telaVisualizar = new
TelaVisualizarArtigo(); // Cria nova tela de visualização
        telaVisualizar.setVisible(true); // Abre a tela
        SwingUtilities.getWindowAncestor(this).dispose(); // Fecha atual
    }

    // Ação ao clicar em "Criar"
    private void btCriarActionPerformed(java.awt.event.ActionEvent evt)
    {
        TelaCadastrarArtigo telaCriar = new TelaCadastrarArtigo(); //
Cria nova tela de cadastro
        telaCriar.setVisible(true); // Abre a tela
        SwingUtilities.getWindowAncestor(this).dispose(); // Fecha atual
    }
}

```

Classe TelaCastro

```

// A classe TelaCadastro herda de JFrame, ou seja, ela É uma janela
gráfica do Swing. public class TelaCadastro extends
javax.swing.JFrame {

    // Construtor da classe que inicializa os componentes gráficos da
janela.
    public TelaCadastro() {
        initComponents(); // Método gerado automaticamente que configura
os botões, textos, etc.
    }

    // Esse método configura toda a interface gráfica (textos, botões,
painéis etc.).
    // É código gerado automaticamente pela IDE (NetBeans) quando você
desenha a tela no modo visual.
    @SuppressWarnings("unchecked")

```

```

private void initComponents() {
    // Declaração e inicialização dos componentes da interface.
    jPanel1 = new javax.swing.JPanel(); // Painei principal (roxo)
    jPanel2 = new javax.swing.JPanel(); // Painei branco central com
os campos
    jLabel1 = new javax.swing.JLabel(); // Título "Cadastro"
    jLabel2 = new javax.swing.JLabel(); // Rótulo "Nome de Usuário"
    txtNomeCadastro = new javax.swing.JTextField(); // Campo de
entrada do nome
    txtSenhaCadastro = new javax.swing.JPasswordField(); // Campo de
entrada da senha
    jLabel3 = new javax.swing.JLabel(); // Rótulo "Senha"
    btnCadastrar = new javax.swing.JButton(); // Botão "Cadastrar"
    jLabel4 = new javax.swing.JLabel(); // Texto "Já tem uma conta?"
    btnToLogin = new javax.swing.JButton(); // Botão para voltar à
tela de login

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
// Fecha o app ao clicar no X

    // Configurações visuais dos painéis
    jPanel1.setBackground(new java.awt.Color(153, 0, 153));
    jPanel1.setLayout(new java.awt.GridBagLayout()); // Usa um
layout flexível centralizado

    jPanel2.setBackground(new java.awt.Color(255, 255, 255));
    jPanel2.setPreferredSize(new java.awt.Dimension(500, 450));

    // Título "Cadastro"
    jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 24));
    jLabel1.setForeground(new java.awt.Color(204, 51, 255));
    jLabel1.setText("Cadastro");

    // Rótulo "Nome de Usuário"
    jLabel2.setFont(new java.awt.Font("Segoe UI", 1, 18));
    jLabel2.setForeground(new java.awt.Color(204, 51, 255));
    jLabel2.setText("Nome de Usuário");

    // Campos com fundo cinza claro
    txtNomeCadastro.setBackground(new java.awt.Color(239, 239,
239));

```



```
txtSenhaCadastro.setBackground(new java.awt.Color(239, 239, 239));

// Rótulo "Senha"
jLabel3.setFont(new java.awt.Font("Segoe UI", 1, 18));
jLabel3.setForeground(new java.awt.Color(204, 51, 255));
jLabel3.setText("Senha");

// Botão "Cadastrar"
btnCadastrar.setBackground(new java.awt.Color(204, 0, 204));
btnCadastrar.setForeground(new java.awt.Color(255, 255, 255));
btnCadastrar.setText("Cadastrar");
// Define o que acontece quando o botão for clicado
btnCadastrar.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
        btnCadastrarActionPerformed(evt); // Chama método abaixo
    }
});

// Texto para quem já tem conta
jLabel4.setFont(new java.awt.Font("Segoe UI", 0, 14));
jLabel4.setForeground(new java.awt.Color(204, 51, 255));
jLabel4.setText("Já tem uma conta?");

// Botão para ir para a tela de login
btnToLogin.setFont(new java.awt.Font("Segoe UI", 2, 14));
btnToLogin.setForeground(new java.awt.Color(204, 51, 255));
btnToLogin.setText("<html><u>Entre</u></html>"); // Sublinhado
btnToLogin.setBorder(null);
btnToLogin.setContentAreaFilled(false); // Botão transparente
btnToLogin.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
        btnToLoginActionPerformed(evt); // Chama método abaixo
    }
});

// Layouts dos componentes (gerado automaticamente)
// Aqui são posicionados os textos, campos e botões no painel
```

```

// Adiciona jPanel2 dentro de jPanel1 (roxo central)
jPanel1.add(jPanel2, new java.awt.GridBagConstraints());

// Define o layout final da janela
javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
// Define o tamanho da janela
layout.setHorizontalGroup(...);
layout.setVerticalGroup(...);

pack(); // Ajusta o tamanho da janela conforme os componentes
}

// Ação executada ao clicar em "Cadastrar"
private void btnCadastrarActionPerformed(java.awt.event.ActionEvent
evt) {
    // Obtém os dados dos campos
    String username = txtNomeCadastro.getText();
    String senha = new String(txtSenhaCadastro.getPassword());

    // Verifica se os campos estão vazios
    if (username.isEmpty() || senha.isEmpty()) {
        JOptionPane.showMessageDialog(null, "Preencha todos os
campos!");
        return;
    }

    // Verifica se já existe um usuário com esse nome
    for (Usuario u : BancoFake.usuarios) {
        if (u.getUsername().equals(username)) {
            JOptionPane.showMessageDialog(null, "Usuário já
cadastrado!");
            return;
        }
    }

    // Se não existir, cria novo usuário e adiciona no banco "fake"
    BancoFake.usuarios.add(new Usuario(username, senha));
    JOptionPane.showMessageDialog(null, "Cadastro realizado com
sucesso!");
}

```

```

        // Abre a tela de login e fecha a de cadastro
        TelaLogin login = new TelaLogin();
        login.setVisible(true);
        SwingUtilities.getWindowAncestor(btnCadastrar).dispose();
    }

    // Ação ao clicar em "Entre" (voltar para tela de login)
    private void btnToLoginActionPerformed(java.awt.event.ActionEvent
    evt) {
        TelaLogin login = new TelaLogin();
        login.setVisible(true);
        SwingUtilities.getWindowAncestor(btnToLogin).dispose(); // Fecha
a tela atual
    }

    // Método principal - executa a janela
    public static void main(String args[]) {
        // Define visual padrão da interface (Nimbus se disponível)
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {

                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (...) {
            // Se não conseguir aplicar Nimbus, mantém o padrão
        }

        // Cria e exibe a janela de cadastro
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new TelaCadastro().setVisible(true);
            }
        });
    }
}

```

Classe TelaLogin

```
// Define a classe TelaLogin como uma subclasse de JFrame (janela)
public class TelaLogin extends javax.swing.JFrame {

    // Construtor da classe, que inicializa os componentes da interface
    public TelaLogin() {
        initComponents();
    }

    // Método gerado automaticamente pelo editor para montar a interface
    gráfica
    @SuppressWarnings("unchecked")
    private void initComponents() {
        // Criação e configuração dos componentes da tela
        jPanel1 = new javax.swing.JPanel(); // Painei principal (roxo)
        jPanel2 = new javax.swing.JPanel(); // Painei interno (branco)
        jLabel1 = new javax.swing.JLabel(); // Label "Login"
        jLabel2 = new javax.swing.JLabel(); // Label "Nome de Usuário"
        txtNomeLogin = new javax.swing.JTextField(); // Campo de texto
        para nome
        txtSenhaLogin = new javax.swing.JPasswordField(); // Campo de
        senha
        jLabel3 = new javax.swing.JLabel(); // Label "Senha"
        btnEntrar = new javax.swing.JButton(); // Botão "Entrar"
        jLabel4 = new javax.swing.JLabel(); // Label "Não tem uma
        conta?"
        btnToCadastro = new javax.swing.JButton(); // Botão "Cadastre-
        se"

        // Configura a ação de fechar a janela

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        // Configuração do layout e cor de fundo do painei principal
        jPanel1.setBackground(new java.awt.Color(153, 0, 153));
        jPanel1.setLayout(new java.awt.GridBagLayout());

        // Painei secundário com fundo branco e tamanho fixo
        jPanel2.setBackground(new java.awt.Color(255, 255, 255));
        jPanel2.setPreferredSize(new java.awt.Dimension(500, 450));
```

```

// Configura o título "Login"
jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 24));
jLabel1.setForeground(new java.awt.Color(204, 51, 255));
jLabel1.setText("Login");

jLabel1.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER)
;

// Label do campo de nome
jLabel2.setFont(new java.awt.Font("Segoe UI", 1, 18));
jLabel2.setForeground(new java.awt.Color(204, 51, 255));
jLabel2.setText("Nome de Usuário");

// Estética dos campos de texto
txtNomeLogin.setBackground(new java.awt.Color(239, 239, 239));
txtSenhaLogin.setBackground(new java.awt.Color(239, 239, 239));

// Label do campo de senha
jLabel3.setFont(new java.awt.Font("Segoe UI", 1, 18));
jLabel3.setForeground(new java.awt.Color(204, 51, 255));
jLabel3.setText("Senha");

// Configuração do botão "Entrar"
btnEntrar.setBackground(new java.awt.Color(204, 0, 204));
btnEntrar.setForeground(new java.awt.Color(255, 255, 255));
btnEntrar.setText("Entrar");
btnEntrar.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
        btnEntrarActionPerformed(evt); // Ação ao clicar
    }
});

// Texto informativo
jLabel4.setFont(new java.awt.Font("Segoe UI", 0, 14));
jLabel4.setForeground(new java.awt.Color(204, 51, 255));
jLabel4.setText("Não tem uma conta?");

// Botão "Cadastre-se" com aparência de link
btnToCadastro.setFont(new java.awt.Font("Segoe UI", 2, 14));

```

```

        btnToCadastro.setForeground(new java.awt.Color(204, 51, 255));
        btnToCadastro.setText("<html><u>Cadastre-se</u></html>");
        btnToCadastro.setBorder(null);
        btnToCadastro.setBorderPainted(false);
        btnToCadastro.setContentAreaFilled(false);
        btnToCadastro.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt)
{
                btnToCadastroActionPerformed(evt); // Ação ao clicar
            }
        });

        // Layout do painel branco (jPanel2)
        // Define o posicionamento dos componentes dentro do painel
branco
        javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);

        jPanel2Layout.setHorizontalGroup( /* Define posição horizontal
dos elementos */ );
        jPanel2Layout.setVerticalGroup( /* Define posição vertical dos
elementos */ );

        // Adiciona o painel branco ao painel principal (roxo)
        jPanel1.add(jPanel2, new java.awt.GridBagConstraints());

        // Define o layout geral da janela principal
        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup( /* Layout horizontal */ );
        layout.setVerticalGroup( /* Layout vertical */ );

        pack(); // Ajusta o tamanho da janela
    }

    // Ação quando o botão "Entrar" é clicado
    private void btnEntrarActionPerformed(java.awt.event.ActionEvent
evt) {
        // Obtém os valores digitados

```

```

String username = txtNomeLogin.getText();
String senha = new String(txtSenhaLogin.getPassword());

boolean encontrou = false;

// Verifica se o usuário existe na lista (BancoFake é um
repositório simulado)
for (Usuario u : BancoFake.usuarios) {
    if (u.getUsername().equals(username) &&
u.getSenha().equals(senha)) {
        BancoFake.usuarioLogado = u; // Define o usuário logado
        encontrou = true;
        break;
    }
}

if (encontrou) {
    // Login bem-sucedido
    JOptionPane.showMessageDialog(null, "Login bem-sucedido!
Bem-vindo, " + username);
    TelaVisualizarArtigo home = new TelaVisualizarArtigo(); //
Abre próxima tela
    home.setVisible(true);
    SwingUtilities.getWindowAncestor(btnEntrar).dispose(); //
Fecha a tela atual
} else {
    // Login falhou
    JOptionPane.showMessageDialog(null, "Usuário ou senha
incorretos.");
}
}

// Ação quando o botão "Cadastre-se" é clicado
private void btnToCadastroActionPerformed(java.awt.event.ActionEvent
evt) {
    TelaCadastro cadastro = new TelaCadastro(); // Abre a tela de
cadastro
    cadastro.setVisible(true);
    SwingUtilities.getWindowAncestor(btnToCadastro).dispose(); //
Fecha a tela atual
}

```

```

// Método main para iniciar o programa
public static void main(String args[]) {
    // Define o tema visual Nimbus, se disponível
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (Exception ex) {

java.util.logging.Logger.getLogger(TelaLogin.class.getName()).log(ja
va.util.logging.Level.SEVERE, null, ex);
    }

    // Cria e exibe a tela
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new TelaLogin().setVisible(true);
        }
    });
}

}

```

Classe TelaVisualizarArtigo

```

public class TelaVisualizarArtigo extends javax.swing.JFrame {

/**
 * Construtor da classe TelaVisualizarArtigo.
 * Inicializa os componentes da tela e carrega os artigos no painel.
 */
public TelaVisualizarArtigo() {
    initComponents();
}

```



```

        // Define o layout do painel onde os artigos serão exibidos como
        uma lista vertical (Y_AXIS).
        pArtigos.setLayout(new javax.swing.BoxLayout(pArtigos,
        javax.swing.BoxLayout.Y_AXIS));

        // Chama o método que carrega e exibe os artigos cadastrados.
        carregarArtigos();
    }

    /**
     * Método responsável por carregar os artigos do "banco de dados
     fake" (BancoFake)
     * e exibi-los no painel da interface usando o componente
     personalizado "Card".
     */
    private void carregarArtigos() {
        pArtigos.removeAll(); // Remove todos os componentes existentes
        no painel

        // Percorre todos os artigos existentes na lista simulada
        for (Artigo artigo : BancoFake.artigos) {
            Card card = new Card(); // Cria um novo componente Card para
            exibir o artigo

            // Define os dados do artigo no componente Card
            card.setTitulo(artigo.getTitulo());
            card.setDescricao(artigo.getConteudo());

            // Adiciona o card no painel de artigos
            pArtigos.add(card);
        }

        // Atualiza o painel para refletir as mudanças visuais
        pArtigos.revalidate();
        pArtigos.repaint();
    }

    // Abaixo está o método gerado automaticamente pelo editor para
    inicializar os componentes da tela
    @SuppressWarnings("unchecked")
    private void initComponents() {

```

```

        navbar1 = new portalbruxa.view.Navbar(); // Componente de barra
de navegação
        pArtigos = new javax.swing.JPanel();      // Paine1 onde os
cards de artigo serão exibidos

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
// Fecha a aplicação ao fechar a janela

        // Configuração visual da navbar
        navbar1.setBackground(new java.awt.Color(255, 255, 255));
        navbar1.setPreferredSize(new java.awt.Dimension(900, 90));
        getContentPane().add(navbar1, java.awt.BorderLayout.PAGE_START);
// Adiciona a navbar no topo da janela

        // Configuração visual do painel de artigos
        pArtigos.setBackground(new java.awt.Color(102, 0, 102));
        pArtigos.setPreferredSize(new java.awt.Dimension(1000, 800));

        // Layout padrão gerado para o painel (não tem componentes
filhos inicialmente)
        javax.swing.GroupLayout pArtigosLayout = new
javax.swing.GroupLayout(pArtigos);
        pArtigos.setLayout(pArtigosLayout);
        pArtigosLayout.setHorizontalGroup(

pArtigosLayout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
                .addGap(0, 1000, Short.MAX_VALUE)
        );
        pArtigosLayout.setVerticalGroup(

pArtigosLayout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
                .addGap(0, 800, Short.MAX_VALUE)
        );

        getContentPane().add(pArtigos, java.awt.BorderLayout.CENTER); //
Adiciona o painel de artigos ao centro da janela

        pack(); // Ajusta o tamanho da janela automaticamente de acordo

```

```

com os componentes
}

/**
 * Método principal da aplicação.
 * Define o tema visual Nimbus (se disponível) e exibe a tela.
 */
public static void main(String args[]) {
    try {
        // Define o "Look and Feel" Nimbus, caso esteja disponível
no sistema
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (Exception ex) {
        // Tratamento de erro caso o Look and Feel não esteja
disponível

java.util.logging.Logger.getLogger(TelaVisualizarArtigo.class.getNam
e()).log(java.util.logging.Level.SEVERE, null, ex);
    }

    // Cria e exibe a janela de forma segura na thread da interface
gráfica (EDT)
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new TelaVisualizarArtigo().setVisible(true);
        }
    });
}

// Declaração dos componentes utilizados na tela
private portalbruxa.view.Navbar navbar1;
private javax.swing.JPanel pArtigos;

}

```

Explicação Simples dos Conceitos

Classe Usuário

Encapsulamento

- **Como está aplicado:**
 - Os atributos username e senha são protected, não public.
 - O acesso é feito por meio de **métodos públicos** (`getUsername()` e `getSenha()`), os chamados **getters**.

Classe Admin

Herança

- **Como está aplicado:**
 - A classe Admin declara `extends Usuario`, indicando que herda de Usuario.
 - Herda atributos (username, senha) e métodos (`getUsername()`, `getSenha()`, etc.) da classe Usuario.
- **Explicação:**
 - Herança permite reaproveitar código e criar especializações.
 - Admin é uma especialização de Usuario que mantém a estrutura básica, mas pode adicionar ou modificar comportamentos.

Polimorfismo (sobrescrita de método)

- **Como está aplicado:**
 - A classe Admin sobrescreve o método `isAdmin()` da classe pai com `@Override`.
 - Enquanto `Usuario.isAdmin()` retorna false, `Admin.isAdmin()` retorna true.
- **Explicação:**

- Isso possibilita tratar objetos de diferentes tipos (aqui, `Usuario` e `Admin`) de forma uniforme, mas com comportamentos específicos.

Classe Artigo

Encapsulamento

- **Como está aplicado:**
 - Os atributos `titulo` e `conteudo` são declarados como `private`, ou seja, **não acessíveis diretamente fora da classe**.
 - O acesso a esses atributos é feito por meio de métodos públicos chamados **getters** (`getTitulo()`, `getConteudo()`).
- **Explicação:**
 - O encapsulamento protege os dados internos da classe, controlando como eles são acessados e modificados.
 - Isso aumenta a segurança e a integridade dos dados, além de permitir alterar a implementação interna sem afetar o código externo

Classe BancoFake

Abstração

- **Como está aplicado:**
 - A classe `BancoFake` **representa uma abstração de um banco de dados**, mas sem acesso real a um banco externo.
 - Ela oferece apenas o necessário: listas de objetos e um usuário logado.
- **Explicação:**
 - A abstração permite **esconder detalhes técnicos de persistência** e modelar o conceito de um "repositório de dados" com simplicidade.
 - Isso facilita o uso da classe em outras partes do sistema, como se fosse uma fonte real de dados.

Classe PortalBruxa

Abstração

- **Como está aplicado:**
 - A classe PortalBruxa representa a **entrada principal da aplicação**, sem expor os detalhes da interface gráfica ou da lógica de negócio.
 - A criação da tela (TelaLogin) e a manipulação da interface ficam escondidas dentro do `invokeLater()`.
- **Explicação:**
 - A abstração está em **separar o ponto de entrada (main)** da lógica da aplicação.
 - O restante do sistema (ex.: autenticação, dados) está oculto, e só o necessário para iniciar a aplicação está exposto.

Classe Navbar

Encapsulamento

- A classe Navbar encapsula o comportamento de uma barra de navegação — ela tem seus atributos (`btCriar`, `btLogout`, etc.) e métodos (`btLogoutActionPerformed`, etc.) organizados dentro de si.

Classe TelaVisualizarArtigo

Encapsulamento

- A tela (`TelaVisualizarArtigo`) encapsula seus próprios componentes (`navbar1`, `pArtigos`) e comportamentos (`carregarArtigos`).
- Usa métodos privados para manter a responsabilidade interna da classe, como `carregarArtigos()`.

Composição

- Usa **composição** ao incluir objetos de outras classes: Navbar, Card, Artigo.