

DESENVOLVIMENTO DE SOFTWARE:

SISTEMA DE GESTÃO HOTELEIRA

BRUNO PADOVEZ TAVANTE
DIEGO BARBOSA ROCHA
FABIANA IVO SOUZA
MATHEUS DURÃES PIRES
MAYARA KRYSTINA DOS S OLIVEIRA

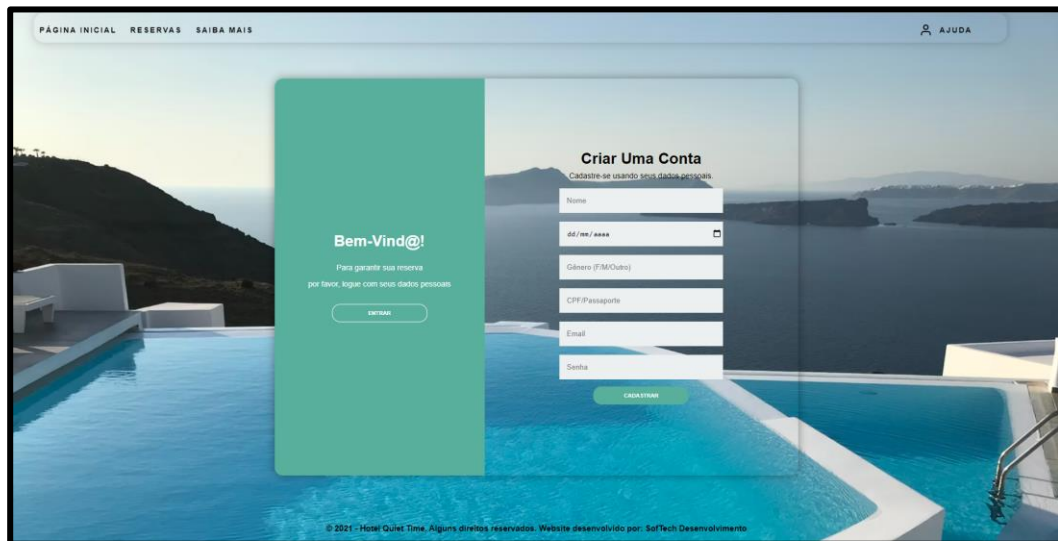
PROJETO INTEGRADO MULTIDISCIPLINAR

UNIVERSIDADE PAULISTA - UNIP
SEMESTRE 4
TURMA: DS4A13
GRADE: 2020/1
CAMPUS: MARQUÊS

Aplicação Desktop completa

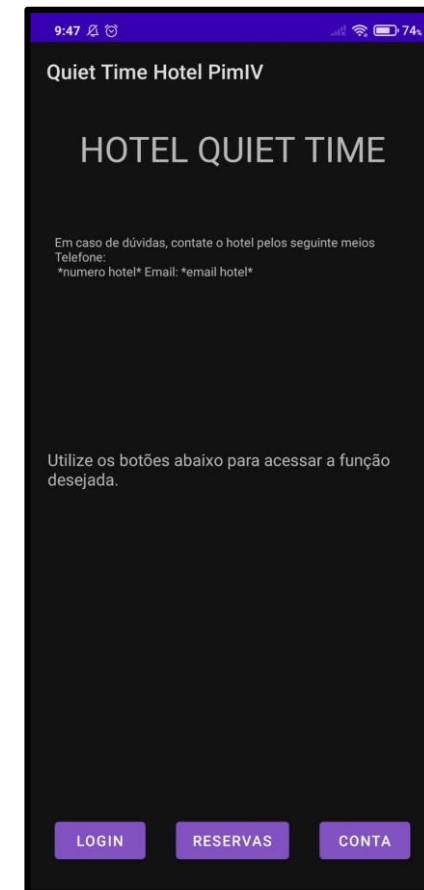


Aplicação Web para testes



**Três softwares.
Três interfaces diferentes**

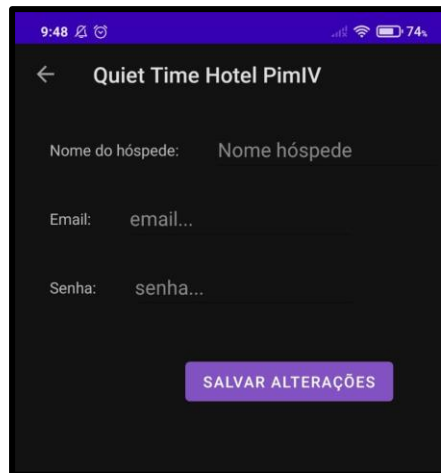
Protótipos mobile (Android)



Iniciando pelo software mais básico, foram feitos protótipos de interface mobile para dispositivos android.

Os usuários seriam hóspedes recorrentes no hotel
O objetivo desse software seria implementar as funções:
Busca por reservas; Login em conta do usuário; Edição de dados da conta.

Interfaces responsivas



9:48 74%

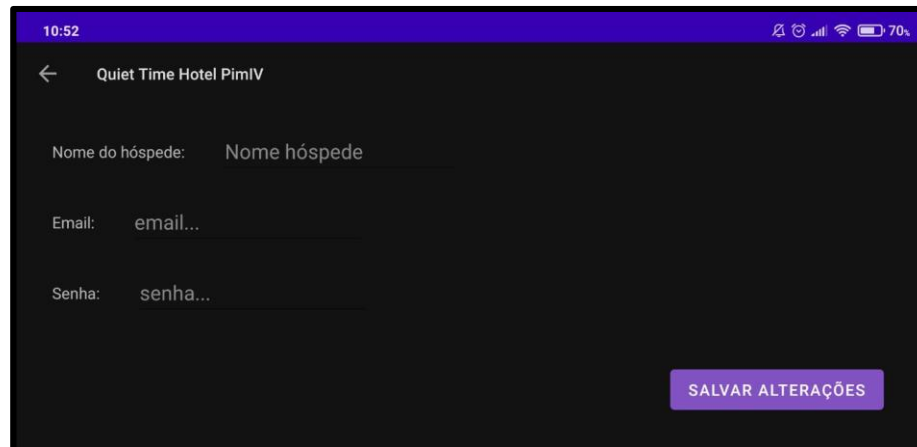
← Quiet Time Hotel PimIV

Nome do hóspede: Nome hóspede

Email: email...

Senha: senha...

SALVAR ALTERAÇÕES



10:52 70%

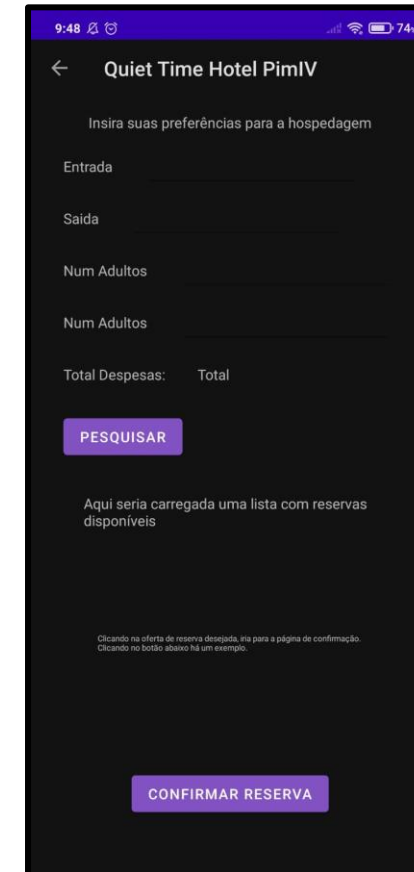
← Quiet Time Hotel PimIV

Nome do hóspede: Nome hóspede

Email: email...

Senha: senha...

SALVAR ALTERAÇÕES



9:48 74%

← Quiet Time Hotel PimIV

Insira suas preferências para a hospedagem

Entrada

Saida

Num Adultos

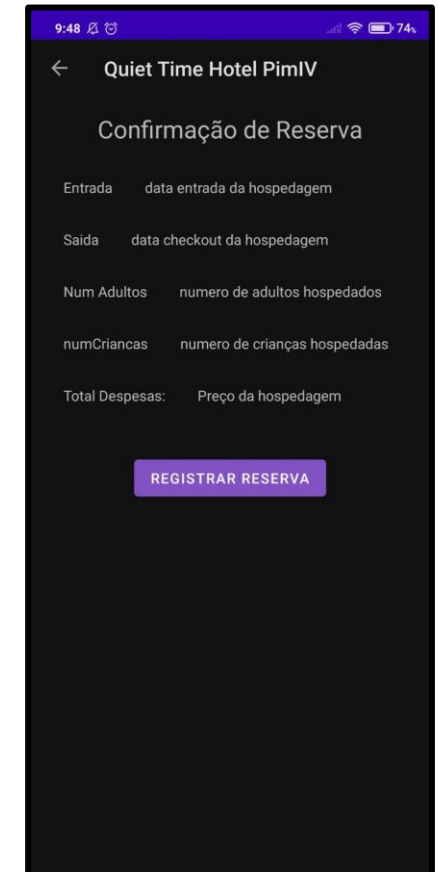
Num Adultos

Total Despesas: Total

PESQUISAR

Aqui seria carregada uma lista com reservas disponíveis

CONFIRMAR RESERVA



9:48 74%

← Quiet Time Hotel PimIV

Confirmação de Reserva

Entrada data entrada da hospedagem

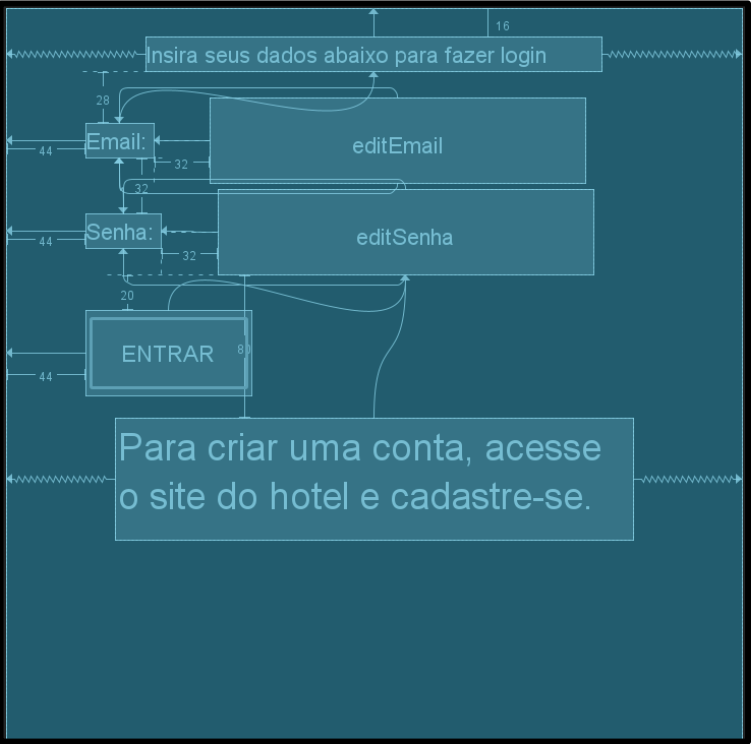
Saida data checkout da hospedagem

Num Adultos numero de adultos hospedados

numCrianças numero de crianças hospedadas

Total Despesas: Preço da hospedagem

REGISTRAR RESERVA



Para **organizar** os elementos (botões, textos) na tela, foi utilizado o mecanismo de “**constraints**”. Onde é feita uma “amarra” entre elementos, de forma que essa conexão se mantém em **qualquer resolução** de tela.

Código em Java da tela inicial, cujos botões executam respectivos métodos, seguindo para outra tela

```
package br.unip.pimivandroid;

import ...

public class MainActivity extends AppCompatActivity {

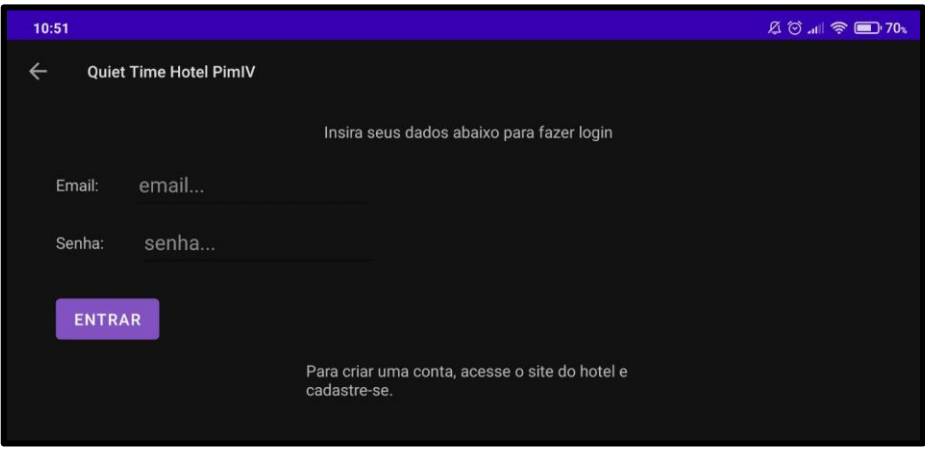
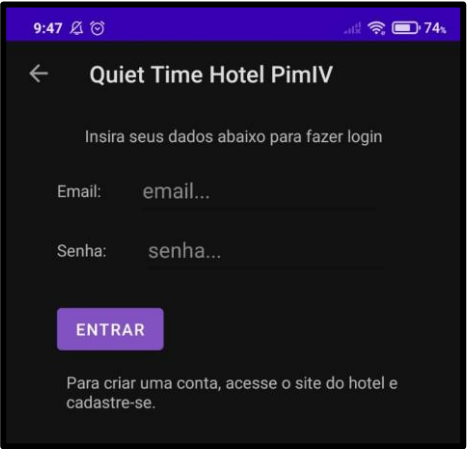
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void buscarReservas(View view) {
        Intent intent = new Intent( packageContext: this, ReservasActivity.class);
        startActivity(intent);
    }

    public void fazerLogin(View view) {
        Intent intent = new Intent( packageContext: this, LoginActivity.class);
        startActivity(intent);
    }

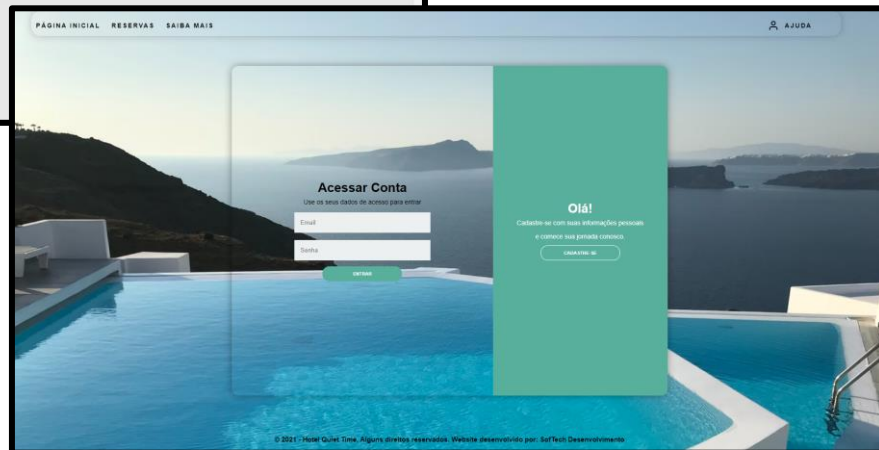
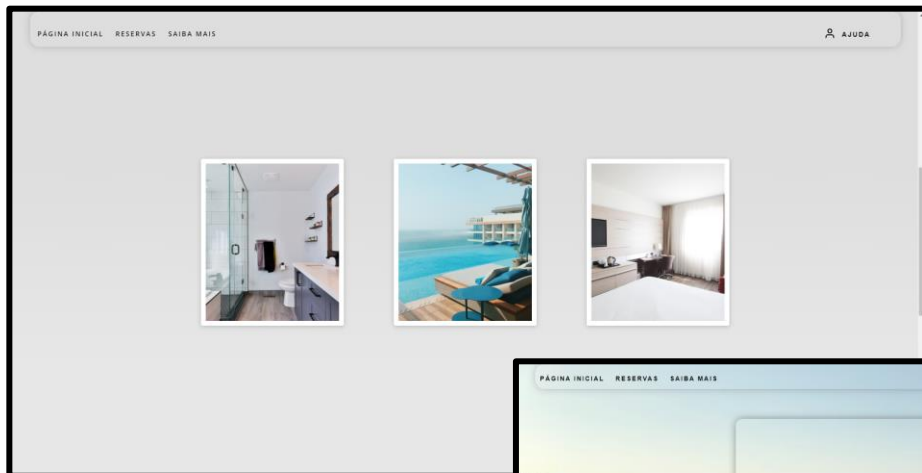
    public void acessarConta(View view) {
        Intent intent = new Intent( packageContext: this, ContaActivity.class);
        startActivity(intent);
    }

}
```

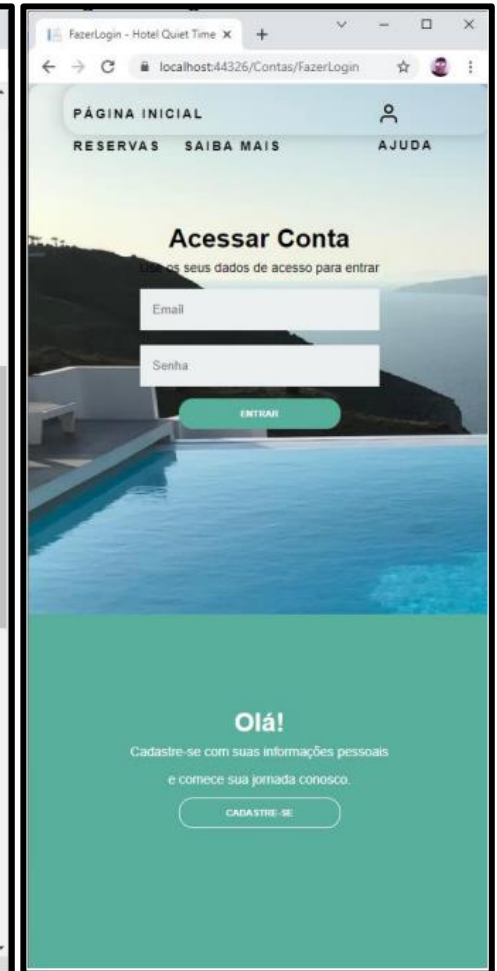
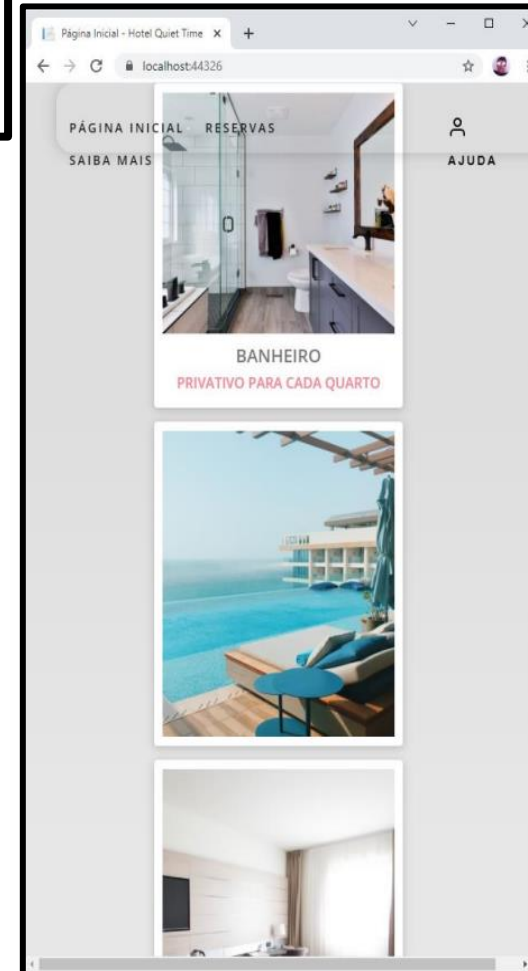


Os usuários seriam hóspedes em geral.

O objetivo desse software seria implementar as funções: Busca por reservas; Login em conta do usuário; Edição de dados pessoais e de conta; Registro de reservas.



Páginas responsivas



O software **faz conexão com banco de dados**.
A partir dos dados no formulário de busca por reservas, são listados as reservas possíveis, com pagamento total já calculado

PÁGINA INICIALRESERVASSAIBA MAIS

AJUDA

Camas do quarto: 1 solteiro | Quarto nº 5

Check-In	Check-Out	Nº Adultos	Nº Crianças
24/11/2021	26/11/2021	1	0

Total = R\$ 80

Reservar

Camas do quarto: 2 solteiro | Quarto nº 9

Check-In	Check-Out	Nº Adultos	Nº Crianças
24/11/2021	26/11/2021	1	0

Total = R\$ 70

Reservar

PÁGINA INICIALRESERVASSAIBA MAIS

AJUDA

Confirmar

Camas do quarto: 1 solteiro | Quarto nº 5

Check-In	Check-Out	Nº Adultos	Nº Crianças
24/11/2021	26/11/2021	1	0

Total = R\$ 80

Confirmar Reserva

© 2021 - Hotel Quiet Time. Alguns direitos reservados. Website desenvolvido por: SofTech Desenvolvimento

Camas do quarto: 1 solteiro | Quarto nº 5

Check-In	Check-Out	Nº Adultos	Nº Crianças
24/11/2021	26/11/2021	1	0

Total = R\$ 80

Reservar

Estando com login efetuado. O usuário pode seguir para **confirmação** da reserva, e devido **registro** no banco de dados.

O modelo de projeto para a interface web contém suporte à páginas com extensão cshtml. Isso significa que é possível utilizar C# para incrementar o conteúdo das páginas HTML.

Exemplo de código controlador, que utiliza C# para tratar requisições via métodos Http e trabalhar respostas.

```
public class ContasController : Controller
{
    ContaDAO cDAO = new ContaDAO();
    HospedeDAO hDAO = new HospedeDAO();
    // GET: Contas
    public ActionResult FazerLogin()
    {
        if(SessaoSite.Conta != null)
        switch (SessaoSite.Mensagem)
        {
            case "ErroLogin":
                SessaoSite.Mensagem = 0;
                return View();
        }
    }

    [HttpPost]
    public ActionResult ValidarHosp(string email, string senha)

    [HttpPost]
    public ActionResult CadastrarHospede(string nome, string email,
        DateTime nasc, string sexo, string cpfPass, string senha)
```

Interfaces web foram desenvolvidas com C#,HTML,CSS e Javascript, no ambiente Visual Studio.

Exemplo de código de uma página web em HTML. É possível identificar a importação de arquivos Javascript e CSS, bem como a utilização de C# para apresentar conteúdos.

```
<script src="~/DesignPaginas/Scripts/FazerLogin.js"></script>
<link href="~/DesignPaginas/Style/FazerLogin.css" rel="stylesheet" />
<div class="containerLogin">

    <div class="content first-content">

        <div class="first-column">
            <h2 class="title title-primary"> Bem-vind@@!</h2><br>
            <p class="description description-primary"> Para garantir sua reserva</p>
            <p class="description description-primary">por favor, logue com seus dados</p>
            <p><span>@ViewBag.LoginInvalido</span></p>
            <p><span>@ViewBag.ErroCadastro</span></p>
            <button id="signin" class="btn btn-primary">Entrar</button>
        </div>

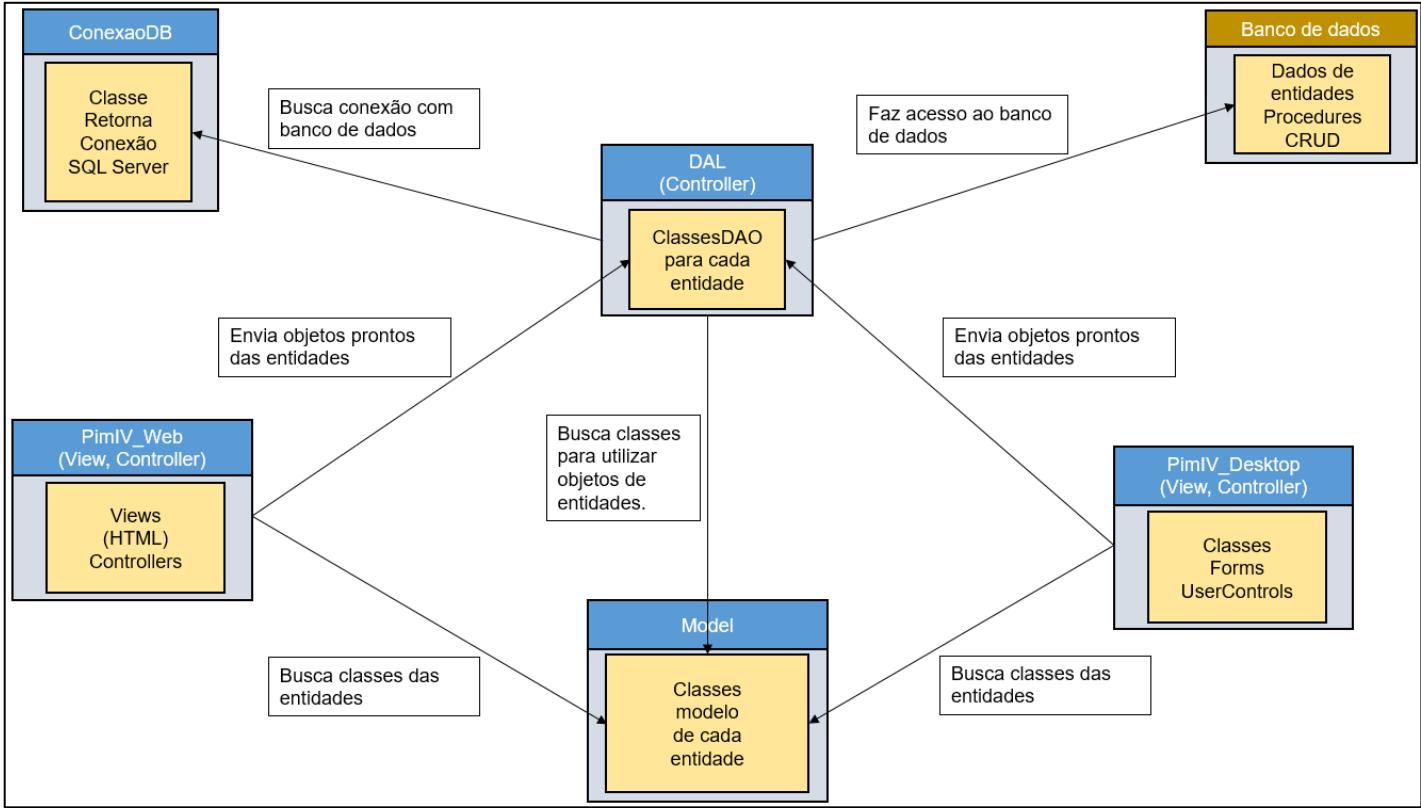
        <div class="second-column">
            <h2 class="title title-second"> Criar uma conta </h2>
            <p class="description description-second">Cadastre-se usando seus dados pe</p>

            <form class="form" method="post" action="@Url.Action("CadastrarHospede","Conta")">
```

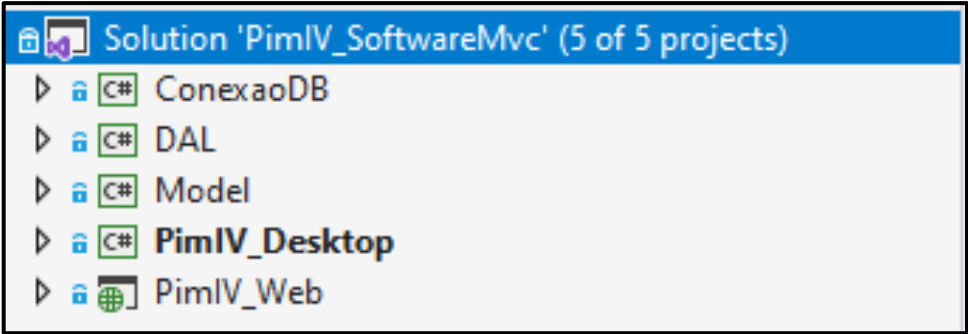

Desenvolvimento

Os projetos **Web** e **Mobile** seguem o padrão arquitetural **MVC** (Model, View, Controller). Isso significa separar os códigos escritos em camadas lógicas. Resumidamente:
Model contém as regras de negócio;
View é responsável por apresentação de dados;
Controller trata requisições do usuário, utilizando conteúdos das camadas View e Model.

Lógica MVC dos projetos Web e Desktop:



A figura ao lado representa a solução do ambiente de desenvolvimento Visual Studio, com os cinco projetos desenvolvidos. 2 são as interfaces web e mobile, enquanto os outros 3 são códigos com a lógica do sistema, que ambas interfaces podem utilizar.



Desenvolvimento

Exemplo de código da camada Model, definindo os atributos e métodos da entidade.

```
namespace Model
{
    public class Hospede
    {
        private int _Id;
        private string _Nome;
        private DateTime _DataNascimento;
        private char _Sexo;
        private string _Email;
        private string _Telefone;
        private string _CPF;
        private string _Passaporte;
        private string _Status;
        private DateTime _DataAtualizacao;
        private Conta _ContaHospede;
        public void setarCPF(string cpfPass)...
```

Exemplo de código controlador (Controller) da camada DAL. Recebe dados vindos da interface (View), utiliza classes Model, e acessa o banco de dados para persistir os dados.

```
public class HospedeDAO : IListaPorStatus
{
    readonly ConexaoSQLServer _conexao = new ConexaoSQLServer();

    public string Inserir_Att(Hospede hospede)
    {
        try
        {
            using (SqlConnection conexao = _conexao.AbrirConexao())
            {
                SqlCommand procedure = new SqlCommand("HOSP_Inserir", conexao);
                procedure.CommandType = CommandType.StoredProcedure;

                procedure.Parameters.Add("@Nome", SqlDbType.VarChar)
                    .Value = hospede.Nome;
```

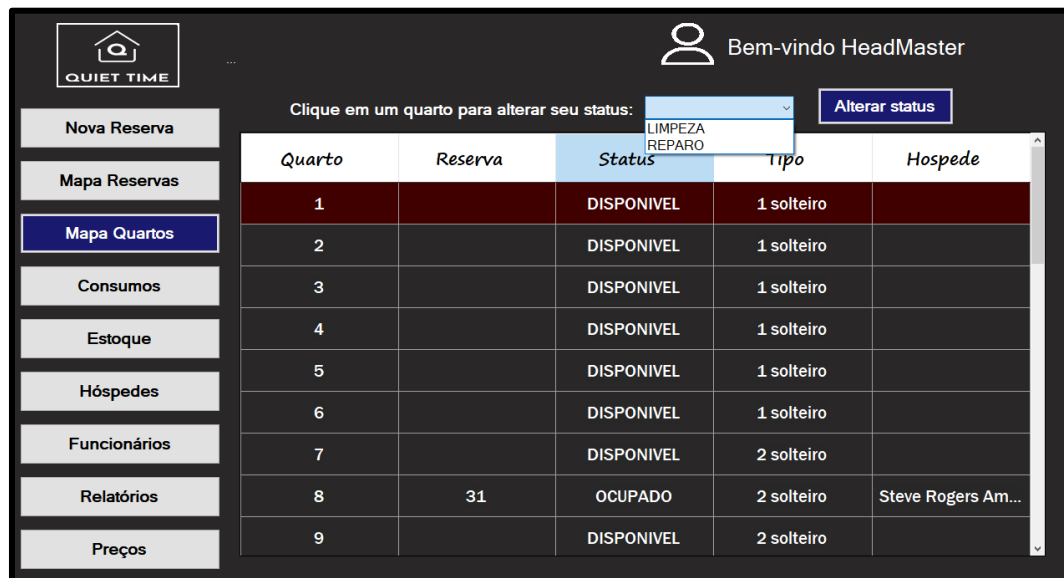
Exemplo de código da camada View no projeto desktop, onde é utilizado um objeto da classe Hospede, para agrupar os dados inseridos pelo usuário, relacionados à essa entidade.

```
private void btnCadastrar_Click(object sender, EventArgs e)
{
    ValidacoesForms.ValidarFormulario(groupDadosPessoais.Controls,
        out string mensagem);
    MessageBox.Show(mensagem);
    if (mensagem == "")
    {
        Hospede hosp = new Hospede();
        hosp.Nome = txtNome.Text;
        hosp.DataNascimento = dtPickerNasc.Value;
        hosp.Sexo = dudSexo.Text[0];
        hosp.Email = txtEmail.Text;
        hosp.Telefone = txtTelefone.Text;
        hosp.setarCPF(txtCPF.Text);
        hosp.Passaporte = txtPassaporte.Text;
        hosp.Status = comboStatus.SelectedItem.ToString();
        string msg = _hospedeDAO.Inserir_Att(hosp);
        ValidacoesForms.ExibirMensagem(msg);
        LimparDadosHospede();
    }
}
```

O software com interface desktop foi desenvolvido para ser completo.

Os usuários seriam apenas funcionários do hotel.

O objetivo desse software seria implementar todas as funções acordadas, desde registro e manipulação de reservas quanto gestão da hospedagem e controle de dados sensíveis

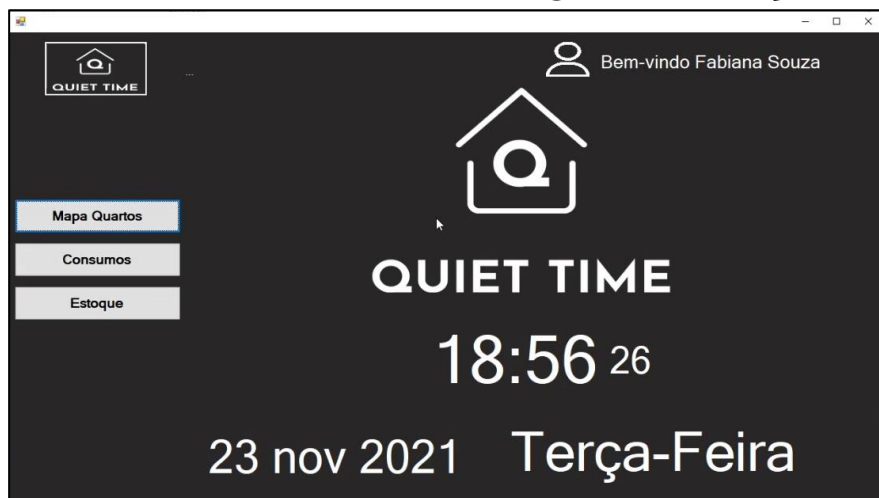


Interfaces simples com foco em usabilidade

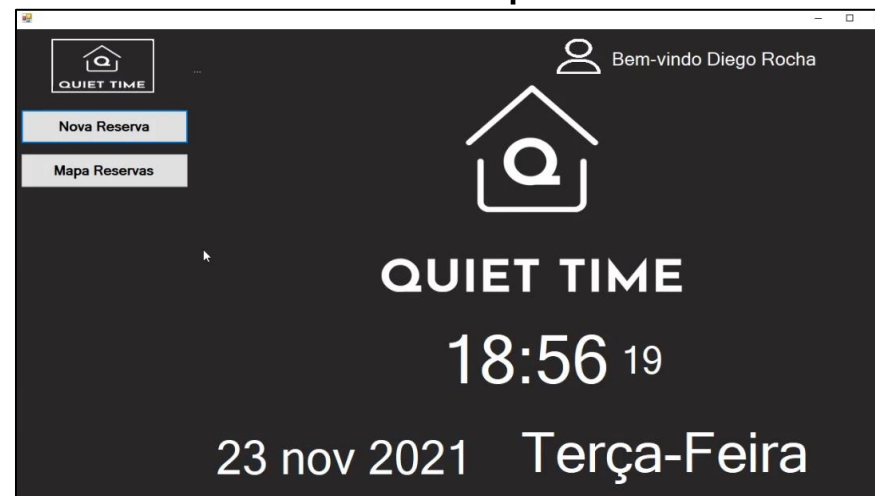
O projeto aderiu à Lei Geral de Proteção de Dados (LGPD).
O principal foco foi em proteger dados pessoais.

Para isso, no momento de login, é identificado o cargo do funcionário, limitando seu acesso a funcionalidades.

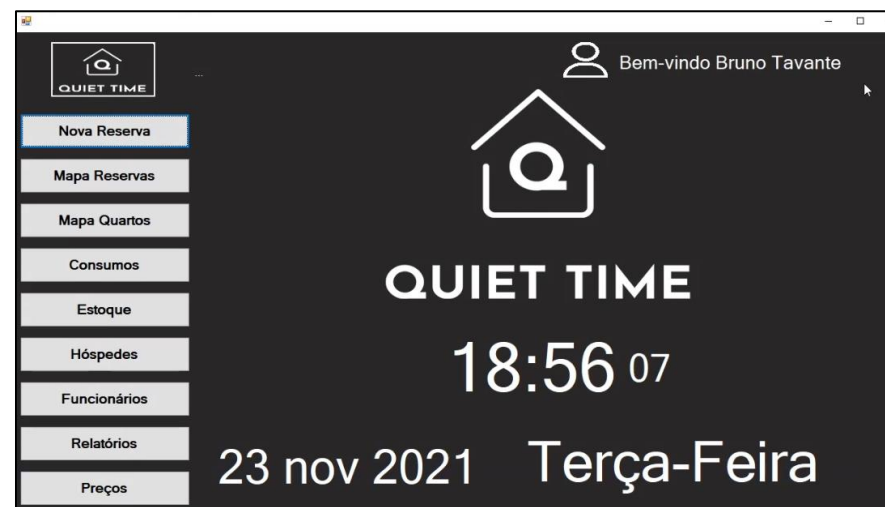
Acesso Gerente de governança



Acesso Recepcionista



Acesso Gerente Geral



No momento de registrar reserva, recepcionistas apenas inserem **dados que os hóspedes fornecerem.**

Gerentes gerais por sua vez, tem acesso direto aos dados pessoais de hóspedes e funcionários. O que implica maior **responsabilidade.**

NOME	SEXO	DATANASC	CPF
Bruno Tavante	M	14/09/1995	
Fabiana Souza	F	12/02/1995	98798798755
Diego Rocha	M	18/05/1995	
Matheus Pires	M	11/07/1995	34534534522
Mayara Oliveira	F	19/03/1995	12312312345
Pao de forma	F	19/12/1999	19823423499
Novo hospede	M	10/10/1999	89999922245
Hagrid	M	24/10/1977	99883334944
Tomiooka	M	24/11/1998	39458588599

Dados pessoais:

Nome: Steve Rogers America

Telefone: (11) 92434-4545 Nascimento: 29/10/1920 Sexo: M

E-mail: steve@america

CPF: 111.222.333-44 Passaporte: tesseract

Dados atualizados em: 23/11/2021 14:11:04 Status: ATIVO

Dados conta hóspede:

Nome de Usuário: Rogers Escudo Senha: escudo

Login atualizado em: 23/11/2021 14:11:04

Interfaces desktop foram desenvolvidas em C#, no ambiente Visual Studio.

Para criar interfaces desktop, foi utilizada a ferramenta Windows Forms. Nela, um Form é uma interface com quantos elementos forem necessários. Também foram utilizados User Controls, que podem ser entendidos como um agrupador de elementos semelhante a um Form, mas que é também tratado como elemento de um Form.

A organização de elementos como botões e textos é feita de forma prática, com “arrastar e soltar”.

A figura abaixo ilustra o desenvolvimento de uma interface do projeto por meio de Windows Forms. Ela, que é um Form, contém diversos User Controls.



Por fim, a proposta dos softwares é **potencialmente positiva**, pois pode gerar **resultados** como:

Maior produtividade no hotel.
Maior organização na gestão.
Praticidade e velocidade no atendimento.
Fácil treinamento de funcionários.
Melhor contato com hóspede.
Facilidade no registro de reservas online.
Potencial fidelidade como cliente ao hotel.

