



4ª Mostra de Talentos

Turma: Desenvolvimento C# .NET

Coordenador: Francisco Lucas Sens

Integrantes da Equipe

Adrian Boaventura Silva

Alan Eduardo Alves

Eduardo Valois Franz

Henrique Venâncio M Cunha

Objetivo:

O que ?

Porque?

Como?

Problemas Encontrados

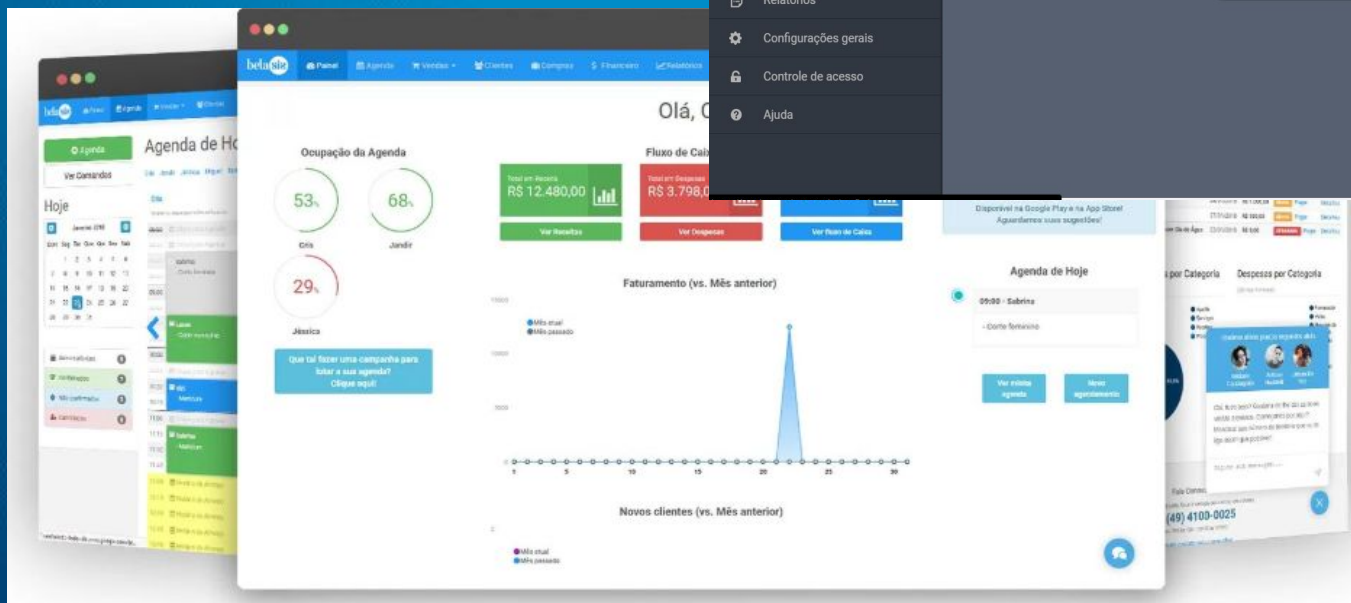
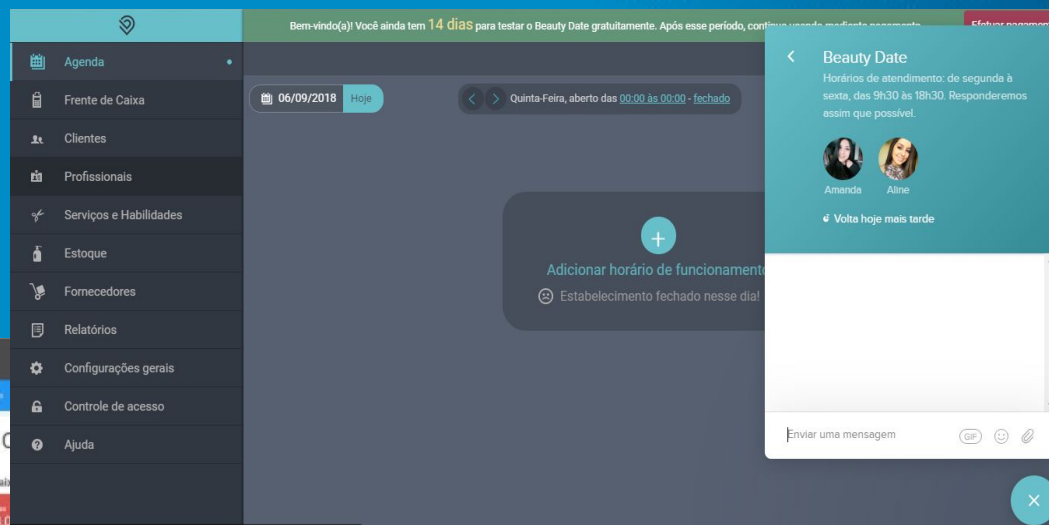
- Demanda
- Agendamentos
- Cadastros
- Segurança das informações

Proposta de Solução

- Desenvolvimento de sistema web
- Otimização
- Agilidade
- Segurança

Soluções Semelhantes

- BelaSis;
- BeautyDate
- Beleza Soft; etc.

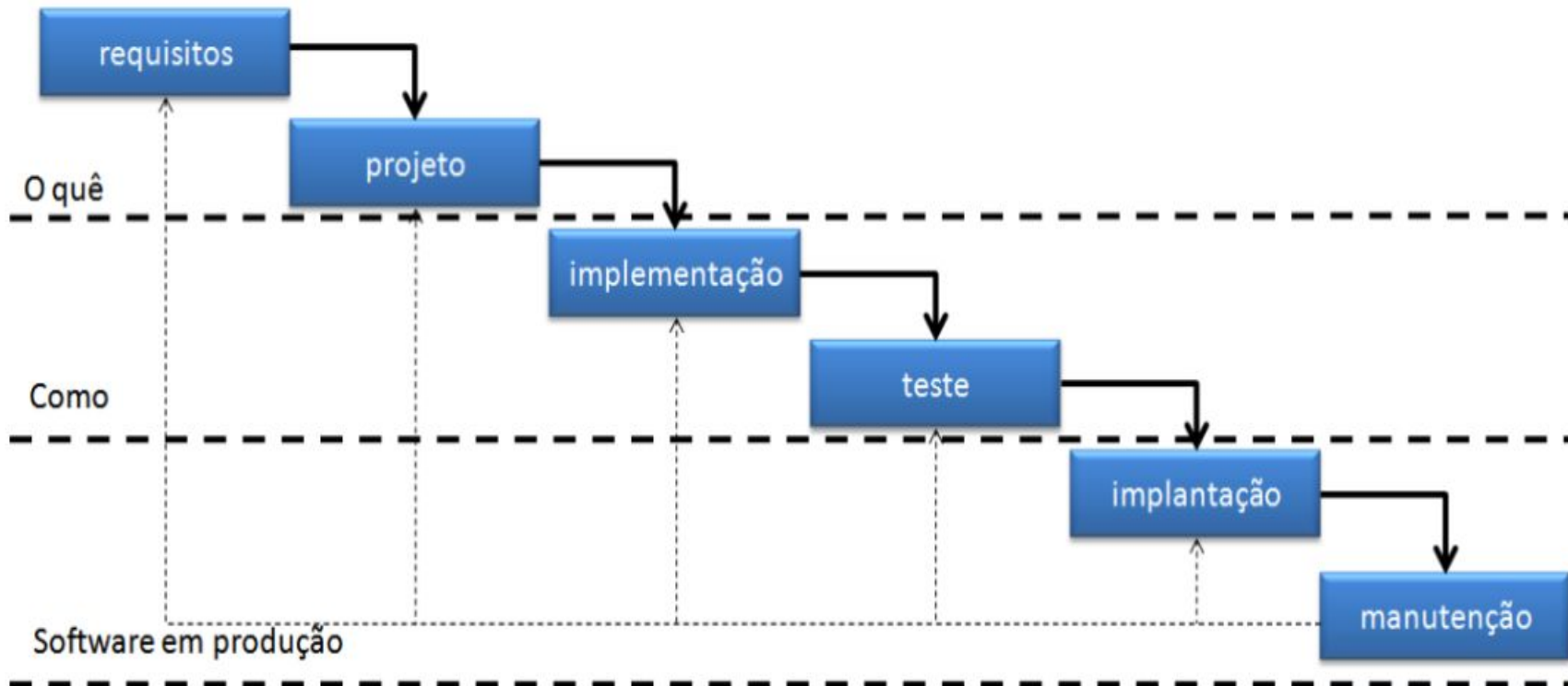


Requisitos Funcionais

- O sistema deve permitir efetuar cadastro de:
 - Clientes;
 - Funcionários;
 - Serviços; etc.
- O Sistema deve permitir realizar o agendamento de serviços
- O sistema deve permitir realizar o pagamento dos agendamentos.
- O sistema deve permitir aos clientes, avaliarem seus agendamentos.
- O sistema deve permitir a exportação dos registros em excel e pdf

Modelo de Ciclo de Vida

Exemplo de gráfico em modelo cascata



Fonte: enghariasoftware.files.wordpress.com

DER - Diagrama de entidade e relacionamento

- Code-First

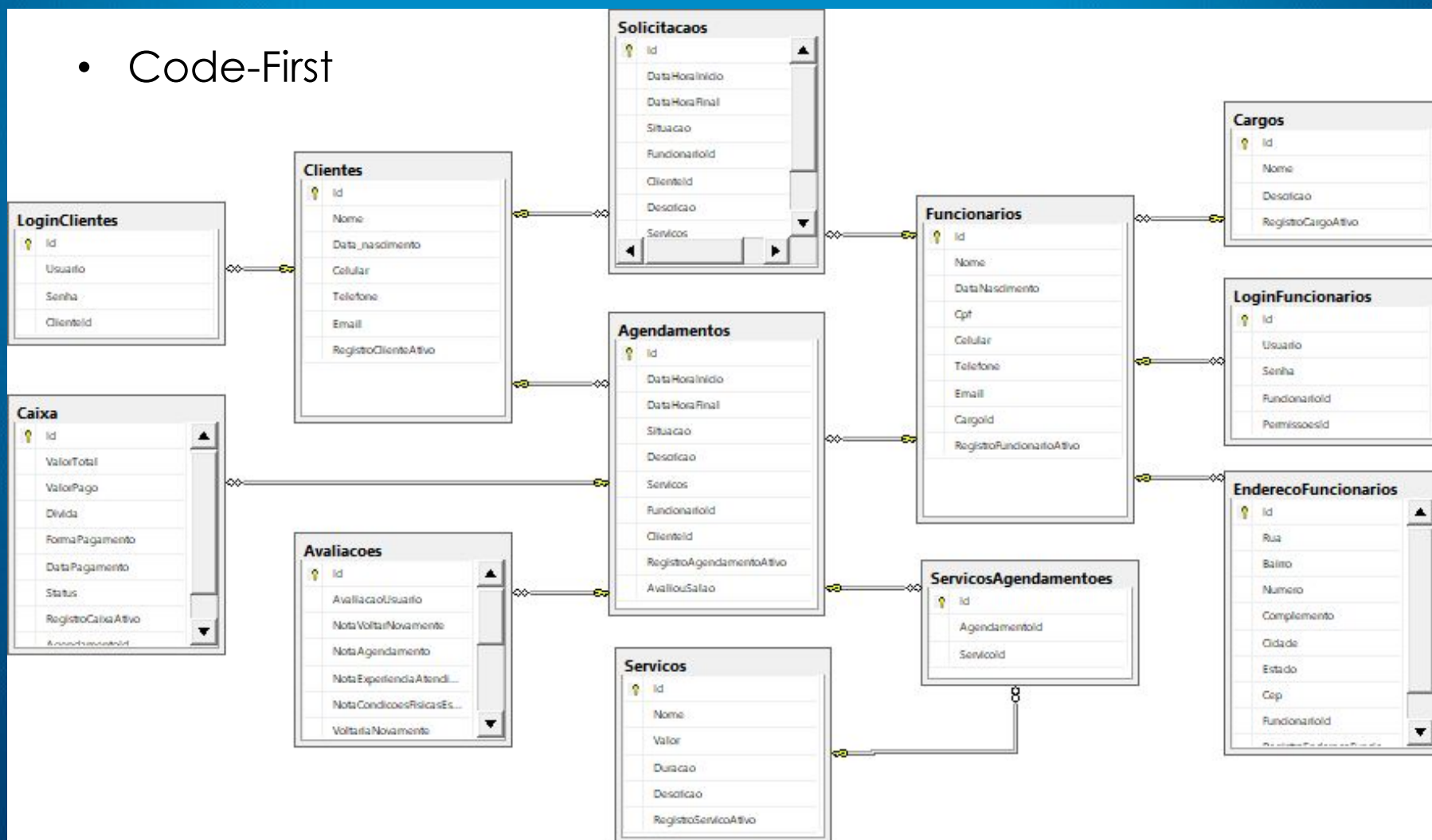
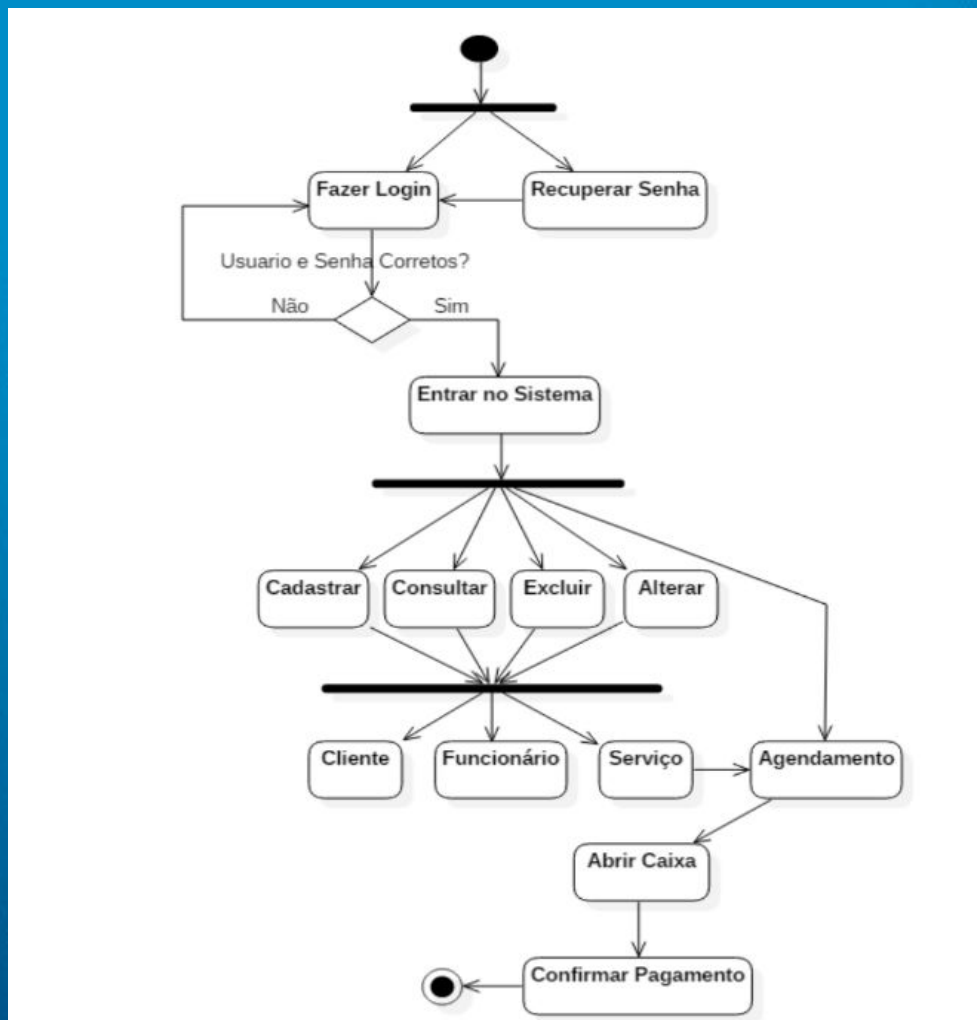


Diagrama de atividades (Funcionários)



Principais tecnologias utilizadas

- IDE: Visual Studio – C#
- SGBD: Microsoft SQL Server – SQL
- Frameworks:
 - Material Design For Bootstrap
 - Bootstrap
 - ASP.NET MVC
 - Entity Framework



Cadastro (Inserindo Permissão)

	Id	TipoPermissao
▶	1	Administrador
	2	Funcionario
⊞	NULL	NULL

	Id	Usuario	Senha	Funcionarioid	PermissoesId
	1	adm	b09c600fddc57...	1	1
	2	fun	77004ea213d5fc...	2	2
⊞	NULL	NULL	NULL	NULL	NULL

Tabelas de LOGINS

Login (Validando Permissão)

```
public ActionResult Login(LoginFuncionario loginF, string returnUrl)
{
    //Valida dados
    if (ModelState.IsValid)
    {
        var usuario = LoginFuncionario.ValidarUsuario(loginF.Usuario, loginF.Senha);

        if (usuario != null)
        {
            int id = usuario.Id;
            int p = db.LoginFuncionarios.Where(end => end.Id == id).SingleOrDefault().PermissoesId;

            permissao = (p == 1) ? "Administrador" : "Funcionario";

            //FormsAuthentication.SetAuthCookie(loginF.Usuario, false);
            var ticket = FormsAuthentication.Encrypt(new FormsAuthenticationTicket
                (1, loginF.Usuario, DateTime.Now, DateTime.Now.AddHours(12), false, permissao));
            var cookie = new HttpCookie(FormsAuthentication.FormsCookieName, ticket);
            Response.Cookies.Add(cookie);
        }
    }
}
```


Acessando método com a permissão

```
// GET: Cliente
[Authorize(Roles = "Administrador, Funcionario")]
public async Task<ActionResult> Index()
{
    return View(await db.Clientes.ToListAsync());
}
```

```
// GET: Cargo
[Authorize(Roles = "Administrador")]
public async Task<ActionResult> Index()
{
    return View(await db.Cargos.ToListAsync());
}
```

Registros Ativos: Ativação e inativação

```
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Cadastrar(Funcionario funcionario, EnderecoFuncionario endereco, int PermissoesId)
{
    if (ModelState.IsValid)
    {
        try
        {
            string aaa = funcionario.Email;
            string verificarCpf = funcionario.Cpf;
            var aa = db.Funcionarios.Where(a => a.Email == aaa || a.Cpf == verificarCpf).SingleOrDefault();

            if (aa != null)
            {
                ViewBag.Endereco = db.EnderecoFuncionarios.Where(end => end.Funcionario.Id == funcionario.Id).SingleOrDefault();
                ViewBag.CargoId = new SelectList(db.Cargos, "Id", "Nome");
                ModelState.AddModelError("", "Esse Cadastro já Existe!");
                return View(funcionario);
            }
            else
            {
                funcionario.RegistroFuncionarioAtivo = true;
                db.Funcionarios.Add(funcionario);
                db.SaveChanges();

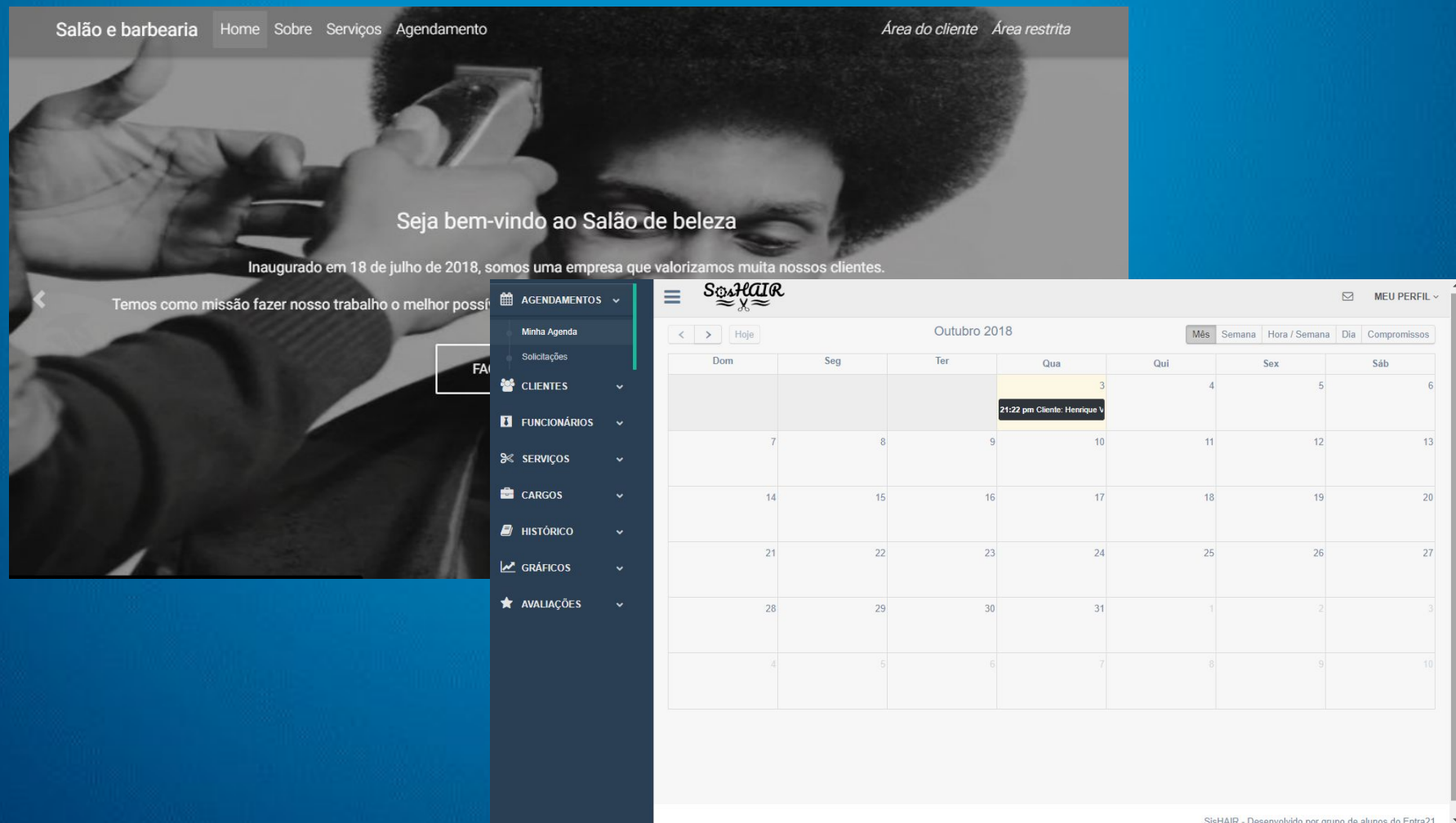
                if (endereco != null)
                {
                    new EnderecoFuncionario().CadastrarEndereco(endereco, funcionario.Id);
                }
                ////
                var loginF = new LoginFuncionario()
                {
                    Usuario = funcionario.Email,
                    Senha = gerarSenha(funcionario),
                    FuncionarioId = funcionario.Id,
                    PermissoesId = PermissoesId
                };
                ////
                db.LoginFuncionarios.Add(loginF);
                db.SaveChanges();
            }
        }
    }
}
```

Registros Ativos: Ativação e inativação

```
[Authorize(Roles = "Administrador")]
[HttpPost, ActionName("InativarRegistro")]
[ValidateAntiForgeryToken]
public ActionResult InativarRegistro(int id)
{
    Funcionario funcionario = db.Funcionarios.Find(id);
    funcionario.RegistroFuncionarioAtivo = false;
    db.Entry(funcionario).State = EntityState.Modified;
    db.SaveChanges();
    return RedirectToAction("Index");
}

[Authorize(Roles = "Administrador")]
[HttpPost, ActionName("AtivarRegistro")]
[ValidateAntiForgeryToken]
public ActionResult AtivarRegistro(int id)
{
    Funcionario funcionario = db.Funcionarios.Find(id);
    funcionario.RegistroFuncionarioAtivo = true;
    db.Entry(funcionario).State = EntityState.Modified;
    db.SaveChanges();
    return RedirectToAction("Index");
}
```


Vamos ver como ficou?







The screenshot displays the SisHAIR web application interface. The top navigation bar includes links for 'Salão e barbearia', 'Home', 'Sobre', 'Serviços', and 'Agendamento'. On the right, there are links for 'Área do cliente' and 'Área restrita'. The main content area features a large image of a person getting a haircut, with the text 'Seja bem-vindo ao Salão de beleza' and a message about the business's inauguration on July 18, 2018. A sidebar menu on the left lists various functions: AGENDAMENTOS (with sub-items 'Minha Agenda' and 'Solicitações'), CLIENTES, FUNCIONÁRIOS, SERVIÇOS, CARGOS, HISTÓRICO, GRÁFICOS, and AVALIAÇÕES. The main content area also displays a calendar for October 2018, showing a booking for '21:22 pm Cliente: Henrique V' on Wednesday, October 3rd. The bottom right corner of the interface includes the text 'SisHAIR - Desenvolvido por grupo de alunos do Entra21'.



Dúvidas?

Obrigado pela atenção!



- Adrian Boaventura Silva
 -  /AdrianBSilva
- Eduardo Valois Franz
 -  /EduardoValoisFranz
- Alan Eduardo Alves
 -  /AlvesAlan
- Henrique Venâncio
 -  /HenriqueVMC

Veja nosso projeto em: github.com/HenriqueVmc/sisHAIR

Só um instante...

