

# SGI FV - Plano de Testes

---

## Multi-Tenant & RLS

---

### Pré-requisitos

---

1. **Executar migrações no Supabase:**

- Acesse o Supabase Dashboard
- Vá em SQL Editor
- Execute `supabase/migrations/001_multiempresa.sql`
- Execute `supabase/migrations/002_rls_policies.sql`

2. **Verificar organização default:**

sql

```
SELECT * FROM organizations WHERE slug = 'default';
```

Se não existir, o script de migração criará automaticamente.

3. **Variáveis de ambiente:**

env

```
VITE_SUPABASE_URL=https://...
```

```
VITE_SUPABASE_ANON_KEY=...
```

```
VITE_ORG_SLUG=default # Opcional
```

---

## Testes de Registro

### TC01: Registro de novo usuário

- [ ] Acessar `/register`
- [ ] Preencher todos os campos obrigatórios
- [ ] Clicar em “Confirmar Registro”
- [ ] **Esperado:** Mensagem de sucesso, redirecionar para `/login`

### TC02: Validação de senha

- [ ] Tentar senha sem maiúscula → Erro
- [ ] Tentar senha sem número → Erro
- [ ] Tentar senha sem caractere especial → Erro
- [ ] Tentar senha < 8 caracteres → Erro
- [ ] Senha válida → Sucesso

### TC03: Email duplicado

- [ ] Tentar registrar com email já existente
  - [ ] **Esperado:** Mensagem “Este email já está cadastrado”
-

## Testes de Login

---

### TC04: Login válido

- ☐ Usar credenciais de usuário registrado
- ☐ **Esperado:** Redirecionar para `/dashboard`

### TC05: Login inválido

- ☐ Usar email inexistente
- ☐ **Esperado:** “Email ou senha inválidos”

### TC06: Restauração de sessão

- ☐ Fazer login
- ☐ Fechar e reabrir navegador
- ☐ **Esperado:** Continuar logado (se sessão Supabase válida)

---

## Testes de Isolamento (RLS)

---

### TC07: Criar duas organizações

```
INSERT INTO organizations (slug, name) VALUES ('org-a', 'Organização A');  
INSERT INTO organizations (slug, name) VALUES ('org-b', 'Organização B');
```

### TC08: Usuário de Org A não vê Org B

- ☐ Criar usuário em Org A
- ☐ Fazer login
- ☐ Tentar acessar dados de Org B via console
- ☐ **Esperado:** Nenhum dado retornado (RLS bloqueia)

### TC09: Admin pode ver membros da org

- ☐ Criar usuário admin em Org A
- ☐ Listar membros
- ☐ **Esperado:** Apenas membros de Org A visíveis

---

## Testes de Dashboard

---

### TC10: Dashboard de cliente

- ☐ Login como client
- ☐ **Esperado:** UserDashboard exibido

### TC11: Dashboard de admin

- ☐ Login como admin/owner
- ☐ **Esperado:** AdminDashboard exibido

## TC12: Logout

- [ ] Clicar em “Sair”
- [ ] **Esperado:** Sessão encerrada, redirecionar para `/login`

## Verificação de Build

### TC13: Build de produção

```
npm run build
```

- [ ] **Esperado:** Build sem erros
- [ ] Verificar `dist/` gerado

### TC14: Preview local

```
npm run preview
```

- [ ] Acessar `http://localhost:4173`
- [ ] Testar fluxo completo

## Queries de Verificação (Supabase)

```
-- Verificar organizações
SELECT * FROM organizations;

-- Verificar memberships
SELECT
  om.user_id,
  om.role,
  o.name as org_name,
  p.nome_completo
FROM org_members om
JOIN organizations o ON o.id = om.org_id
LEFT JOIN profiles p ON p.id = om.user_id;

-- Verificar perfis
SELECT id, email, nome_completo, org_id FROM profiles;

-- Verificar RLS está ativo
SELECT tablename, rowsecurity
FROM pg_tables
WHERE schemaname = 'public'
AND tablename IN ('organizations', 'org_members', 'profiles');
```

## Problemas Conhecidos

---

1. **Dashboard ainda usa localStorage:** AdminDashboard lista usuários do array mock. A migração completa para Supabase está no roadmap v1.2.0.
2. **RLS pode bloquear operações:** Se encontrar erros de permissão, verificar policies e usar `service_role` para debug.
3. **Email de confirmação:** Supabase pode exigir confirmação de email. Desabilitar em Auth > Settings se necessário para dev.