



# Guia de Teste Local - OrganizeZen



## Pré-requisitos

- Node.js 18+
- PostgreSQL
- Yarn package manager
- Git



## Configuração Inicial

### 1. Verificar Variáveis de Ambiente

O arquivo `.env` já está configurado com:

- DATABASE\_URL (PostgreSQL)
- AWS S3 credentials (para upload de ficheiros)
- NextAuth secret
- VAPID keys (notificações push)

**Localização:** `/home/ubuntu/organizezen/nextjs_space/.env`

### 2. Instalar Dependências

```
cd /home/ubuntu/organizezen/nextjs_space
yarn install
```

### 3. Gerar Cliente Prisma

```
yarn prisma generate
```

### 4. Aplicar Migrações (se necessário)

```
yarn prisma db push
```



## Executar em Modo Desenvolvimento

### Iniciar Servidor de Desenvolvimento

```
cd /home/ubuntu/organizezen/nextjs_space
yarn dev
```

O app estará disponível em: **<http://localhost:3000>**

## Funcionalidades para Testar

---

### 1. Autenticação

- Acede a `localhost:3000/login`
- Credenciais de teste (admin):
- **Email:** admin@organizen.cv
- **Password:** [verificar na base de dados]

### 2. Chat com Mídia

- Acede a `/chat`
- Testa pesquisa de utilizadores
- Cria conversas 1:1
- Cria grupos
- Envia mensagens com:
  -  Texto
  -  Imagens (JPEG, PNG, GIF, WebP)
  -  Vídeos (MP4, WebM)
  -  Áudios (MP3, WAV)
  -  Documentos (PDF, DOC)

### 3. Eventos com Mídia

- Acede a `/events`
- Cria um novo evento
- Adiciona imagens ao evento
- Abre o chat do evento
- Envia ficheiros no chat:
  - Clica no ícone 
  - Seleciona ficheiro
  - Vê preview
  - Envia mensagem

### 4. PWA Features

- Abre DevTools (F12)
- Vai para **Application > Service Workers**
- Verifica se `sw.js v2.1.0` está ativo
- Vai para **Application > Cache Storage**
- Verifica cache `organizen-v2.1.0`

### 5. Departamentos

- Acede a `/departments`
- Clica em “Visualizar” num departamento
- Verifica modal com todos os detalhes
- Testa navegação para utilizadores/equipas

### 6. Outras Páginas

- `/dashboard` - Painel principal

- `/tasks` - Gestão de tarefas
  - `/calendar` - Calendário
  - `/users` - Gestão de utilizadores
  - `/settings` - Configurações (admin)
- 

## Comandos Úteis

### Build de Produção (Teste)

```
yarn build  
yarn start
```

### Verificar TypeScript

```
yarn tsc --noEmit
```

### Ver Logs do Prisma

```
yarn prisma studio
```

Abre interface web em `http://localhost:5555`

### Limpar Cache e Reinstalar

```
rm -rf node_modules .next  
yarn install  
yarn prisma generate
```

## Testar PWA em Dispositivo Móvel

### Opção 1: ngrok (Recomendado)

```
# Instalar ngrok (se não tiveres)  
npm install -g ngrok  
  
# Com o servidor dev a correr, noutra terminal:  
ngrok http 3000
```

Usa o URL gerado (ex: `https://abc123.ngrok.io`) no telemóvel.

## Opção 2: IP Local

```
# Descobrir IP local
ip addr show | grep "inet" | grep -v 127.0.0.1

# Aceder via telemóvel (mesmo Wi-Fi)
http://[SEU_IP]:3000
```



## Resolução de Problemas

### Erro: “Cannot find module ‘@prisma/client’“

```
yarn prisma generate
```

### Erro: “Port 3000 is already in use”

```
# Matar processo na porta 3000
lsof -ti:3000 | xargs kill -9

# Ou usar outra porta
PORT=3001 yarn dev
```

### Erro: “Database connection failed”

Verifica se PostgreSQL está a correr:

```
# Status
sudo systemctl status postgresql

# Iniciar
sudo systemctl start postgresql
```

### Erro 500 ao fazer upload

Verifica AWS credentials no `.env`:

- `AWS_S3_ACCESS_KEY`
- `AWS_S3_SECRET_KEY`
- `AWS_BUCKET_NAME`

### Service Worker não atualiza

```
# No Chrome DevTools:
# Application > Service Workers > Unregister
# Depois Ctrl+Shift+R (hard refresh)
```



## Verificar Base de Dados

### Via Prisma Studio

```
yarn prisma studio
```

### Via CLI

```
yarn prisma db pull # Sincronizar schema
yarn prisma db push # Aplicar mudanças
```

## 🎯 Checklist de Testes

- [ ] Login funciona
- [ ] Chat: pesquisa de utilizadores
- [ ] Chat: criar conversa 1:1
- [ ] Chat: criar grupo
- [ ] Chat: enviar imagem
- [ ] Chat: enviar vídeo
- [ ] Chat: enviar documento
- [ ] Eventos: criar evento
- [ ] Eventos: adicionar imagem
- [ ] Eventos: chat com mídia
- [ ] Departamentos: visualizar detalhes
- [ ] PWA: service worker ativo
- [ ] PWA: cache funcionando
- [ ] Notificações push (se configurado)



## Notas Importantes

1. **Base de Dados:** Usa a mesma base de dados de produção (cuidado ao fazer alterações!)
2. **Uploads:** Ficheiros são enviados para S3 de produção
3. **Service Worker:** Pode precisar de hard refresh (Ctrl+Shift+R) após alterações
4. **Hot Reload:** Next.js recarrega automaticamente ao editar ficheiros



## Suporte

Se encontrares problemas:

1. Verifica logs no terminal onde `yarn dev` está a correr
2. Abre DevTools (F12) e verifica Console/Network
3. Verifica se todas as variáveis de ambiente estão corretas

## ✓ Tudo Pronto!

---

O projeto está 100% funcional localmente. Todas as features implementadas:

- ✓ Chat com mídia (imagens, vídeos, áudios, documentos)
- ✓ Eventos com galeria e chat
- ✓ PWA v2.1.0 com auto-update
- ✓ Upload de ficheiros funcionando
- ✓ Signed URLs para S3

**Bom teste!** 🚀