

Comandos Rápidos - SYNK-IA Agents

Desarrollo

Iniciar el proyecto

```
cd /home/ubuntu/synk-ia
npm install
npm run dev
```

El proyecto estará disponible en: `http://localhost:5173`

Ver logs en tiempo real

```
# Terminal 1: Servidor
npm run dev

# Terminal 2: Logs
tail -f ~/.npm/_logs/*.log
```

Testing Rápido

Test Manual de Agentes

1. CEO Brain Agent

```
# Abrir en navegador
http://localhost:5173/ceo-brain

# Usuario autorizado para testing:
# Email: ruben@loffresco.com
```

Pruebas sugeridas:

1. "Dame un resumen ejecutivo del negocio"
2. "¿Cuántas facturas pendientes tengo?"
3. "Analiza la facturación del último mes"
4. Probar reconocimiento de voz
5. Verificar panel de métricas

2. HR Agent

```
# Abrir en navegador
http://localhost:5173/hr-agent
```

Pruebas sugeridas:

1. "Muéstrame mi última nómina"
2. "¿Qué es el IRPF?"
3. "¿Cuántos días de vacaciones me quedan?"
4. Verificar panel de nómina
5. Verificar privacidad de datos

3. Central Agent

```
# Abrir en navegador
http://localhost:5173/central-agent
```

Pruebas sugeridas:

1. "Analiza todo el sistema"
2. "Busca facturas de enero"
3. "¿Qué oportunidades de ahorro hay?"
4. Subir un archivo de prueba
5. "Dame un resumen del negocio"

4. Biloop Agent

```
# Abrir en navegador
http://localhost:5173/bilooop-agent
```

Pruebas sugeridas:

1. Subir archivo CSV de facturas
2. Subir PDF de factura
3. "Analiza las últimas facturas"
4. "Compara precios de suministros"
5. "Dame un resumen de gastos"



Debugging

Ver console.log en navegador

```
# Abrir Chrome DevTools
F12 o Cmd+Option+I (Mac)

# Filtrar logs por agente
[CE0Brain]
[HRAgent]
[CentralAgent]
[BilooopAgent]
```

Ver errores de Base44

```
// En browser console
localStorage.getItem('base44_debug')
```

Limpiar caché de conversaciones

```
// En browser console
localStorage.clear()
sessionStorage.clear()
location.reload()
```



Build y Deploy

Build de producción

```
cd /home/ubuntu/synk-ia

# Build
npm run build

# Preview del build
npm run preview

# El build estará en: dist/
```

Deploy rápido con rsync

```
# A servidor remoto
rsync -avz --delete dist/ usuario@servidor:/var/www/synk-ia/

# Local a carpeta
cp -r dist/* /var/www/html/synk-ia/
```

Deploy con Docker (opcional)

```
# Build imagen
docker build -t synk-ia-agents .

# Run contenedor
docker run -d -p 80:80 synk-ia-agents

# Ver logs
docker logs -f <container-id>
```



Mantenimiento

Actualizar dependencias

```
npm outdated
npm update
```

Verificar errores de ESLint

```
npm run lint
```

Formatear código

```
# Si tienes prettier configurado
npm run format
```



Verificación de Servicios

Test de CEO Brain Service

```
// En browser console (después de autenticación)
import { CEOBrainService } from '/src/services/agents/ceoBrainService.js';

// Test métricas
const metrics = await CEOBrainService.getBusinessMetrics();
console.log('Metrics:', metrics);

// Test enriquecimiento
const enriched = await CEOBrainService.enrichMessageWithContext('análisis');
console.log('Enriched:', enriched);
```

Test de HR Service

```
import { HRAgentService } from '/src/services/agents/hrAgentService.js';

// Test nómina
const payroll = await HRAgentService.analyzeLatestPayroll('email@ejemplo.com');
console.log('Payroll:', payroll);

// Test anomalías
const anomalies = await HRAgentService.detectPayrollAnomalies('email@ejemplo.com');
console.log('Anomalies:', anomalies);
```

Test de Central Service

```
import { CentralAgentService } from '/src/services/agents/centralAgentService.js';

// Test búsqueda
const results = await CentralAgentService.searchAll('cliente');
console.log('Search:', results);

// Test ahorros
const savings = await CentralAgentService.analyzeSavingsOpportunities();
console.log('Savings:', savings);
```

Test de Biloop Service

```
import { BiloopAgentService } from '/src/services/agents/biloopAgentService.js';

// Test tipo de archivo
const type = BiloopAgentService.detectFileType('factura.pdf');
console.log('File type:', type);

// Test análisis de gastos
const expenses = await BiloopAgentService.analyzeRecentExpenses(30);
console.log('Expenses:', expenses);
```



Troubleshooting Rápido

Error: “Cannot connect to Base44”

```
# Verificar appId en base44Client.js
cat src/api/base44Client.js | grep appId

# Debería ser: 6909eb511f749a49b63df48c
```

Error: “Agent not found”

```
# Verificar nombres de agentes
cat src/services/agents/*.js | grep agentName

# Deberían ser:
# ceo_brain
# hr_assistant
# central_coordinator
# biloop_assistant
```

Error: “User not authorized” (CEO Brain)

```
# Verificar email en CEOBrain.jsx
cat src/pages/CEOBrian.jsx | grep CEO_EMAILS

# Añadir tu email si es necesario
```

Error: “Cannot read metrics”

```
# Verificar permisos de entidades en Base44
# Asegurarse de tener acceso a:
# - Invoice
# - Client
# - Provider
# - Payroll
```

Acceso Rápido a URLs

Desarrollo (localhost)

```
Dashboard:      http://localhost:5173/
CEO Brain:      http://localhost:5173/ceo-brain
HR Agent:       http://localhost:5173/hr-agent
Central Agent:  http://localhost:5173/central-agent
Bilooop Agent:  http://localhost:5173/bilooop-agent
```

Producción (ejemplo)

```
Dashboard:      https://app.synk-ia.com/
CEO Brain:      https://app.synk-ia.com/ceo-brain
HR Agent:       https://app.synk-ia.com/hr-agent
Central Agent:  https://app.synk-ia.com/central-agent
Bilooop Agent:  https://app.synk-ia.com/bilooop-agent
```

Checklist de Testing Completo

Pre-deploy

- ☐ Build sin errores
- ☐ Todos los agentes cargan
- ☐ Autenticación funciona
- ☐ Métricas se cargan
- ☐ Mensajes se envían
- ☐ Respuestas se reciben
- ☐ Upload de archivos funciona
- ☐ No hay errores en console
- ☐ UI responsive en móvil
- ☐ Todos los servicios responden

Post-deploy

- ☐ URLs accesibles
 - ☐ SSL/HTTPS activo
 - ☐ Base44 conectado
 - ☐ Logs sin errores críticos
 - ☐ Performance aceptable
 - ☐ Usuarios pueden autenticarse
 - ☐ Datos se cargan correctamente
 - ☐ Conversaciones persisten
 - ☐ Notificaciones funcionan
 - ☐ WhatsApp conecta (si aplica)
-

Backup y Restore

Backup del proyecto

```
# Backup completo
tar -czf synk-ia-backup-$(date +%Y%m%d).tar.gz /home/ubuntu/synk-ia

# Backup solo src
tar -czf synk-ia-src-$(date +%Y%m%d).tar.gz /home/ubuntu/synk-ia/src

# Backup solo servicios
tar -czf synk-ia-services-$(date +%Y%m%d).tar.gz /home/ubuntu/synk-ia/src/services
```

Restore

```
# Descomprimir backup
tar -xzf synk-ia-backup-20250109.tar.gz -C /home/ubuntu/

# Reinstalar dependencias
cd /home/ubuntu/synk-ia
npm install
```



Logs Útiles

Ver logs de desarrollo

```
# Logs de Vite
npm run dev 2>&1 | tee dev.log

# Ver últimas 50 líneas
tail -50 dev.log

# Seguir logs en tiempo real
tail -f dev.log
```

Buscar errores

```
# Buscar errores en logs
grep -i "error" dev.log

# Buscar warnings
grep -i "warn" dev.log

# Buscar por agente
grep -i "\[CE0Brain\]" dev.log
```

Variables de Entorno

Crear .env.local

```
cat > .env.local << EOF
VITE_BASE44_APP_ID=6909eb511f749a49b63df48c
VITE_ENV=development
VITE_DEBUG=true
EOF
```

Verificar variables

```
cat .env.local
```

Comandos de Emergencia

Reinicio completo

```
# Limpiar todo y reinstalar
rm -rf node_modules package-lock.json
npm install
npm run dev
```

Restaurar configuración original

```
# Deshacer cambios locales (!¡CUIDADO!)
git reset --hard HEAD
git clean -fd
npm install
```

Verificar integridad

```
# Verificar archivos modificados
find src -name "*.jsx" -newer /tmp/timestamp
find src -name "*.js" -newer /tmp/timestamp

# Contar líneas de código
find src -name "*.jsx" -o -name "*.js" | xargs wc -l
```

Documentación Rápida

Ver estructura del proyecto

```
tree -L 3 -I 'node_modules|dist' /home/ubuntu/synk-ia
```


Generar documentación de servicios

```
# Si tienes jsdoc instalado  
npx jsdoc src/services/agents/*.js -d docs/services
```

Buscar TODOs pendientes

```
grep -r "TODO" src/  
grep -r "FIXME" src/
```

✓ Verificación Final

Antes de commit

```
# 1. Lint  
npm run lint  
  
# 2. Build  
npm run build  
  
# 3. Test local  
npm run preview  
  
# 4. Verificar archivos  
git status  
  
# 5. Commit  
git add .  
git commit -m "feat: Implementación completa de 4 agentes de IA"  
git push
```

📞 Soporte

Si necesitas ayuda:

1. Revisa la documentación en `GUIA_AGENTES_IA.md`
2. Consulta el README técnico en `AGENTES_README_TECNICO.md`
3. Verifica el resumen en `IMPLEMENTACION_AGENTES_RESUMEN.md`
4. Contacta a soporte@synk-ia.com

Última actualización: 9 de Enero de 2025

Versión: 1.0.0