# GUÍA DE DESARROLLO RÁPIDO -SPEAKLYPLAN

# **MINICIO RÁPIDO (5 minutos)**

### 1. Ubicación del Proyecto

```
cd /home/ubuntu/speaklyplan/nextjs_space
```

### 2. Comandos Esenciales

```
# Desarrollo local
yarn dev # Puerto 3000

# Build producción
yarn build

# Generar Prisma client (después de cambios en schema)
yarn prisma generate

# Ver base de datos (GUI)
yarn prisma studio # Puerto 5555
```

### 3. Rutas Principales

- Landing: http://localhost:3000
- Login: http://localhost:3000/auth/login
- Dashboard: http://localhost:3000/dashboard
- Tutor IA: http://localhost:3000/tutor
- Vocabulario: http://localhost:3000/vocabulario
- Recursos: http://localhost:3000/recursos
- Guía: http://localhost:3000/guia

# ARCHIVOS CRÍTICOS

### No Modificar (A menos que sea absolutamente necesario)

```
.env  # Credenciales de producción
lib/auth.ts  # Configuración NextAuth
prisma/schema.prisma  # Schema DB (requiere migración)
next.config.js  # Config Next.js
```

### **Modificar con Cuidado**

```
lib/ai/prompts.ts  # Prompts del tutor IA
lib/ai/gamification-service.ts # Lógica de gamificación
app/layout.tsx  # Layout principal
app/providers.tsx  # Providers globales
```

### **Modificar Libremente**

```
app/*/page.tsx  # Páginas
app/*/_components/*.tsx  # Componentes de página
components/ui/*.tsx  # Componentes UI
components/gamification/*.tsx  # Componentes gamificación
```

# **© TAREAS COMUNES**

### Agregar Nueva Página

```
# 1. Crear estructura
mkdir -p app/nueva-seccion/_components
touch app/nueva-seccion/page.tsx
touch app/nueva-seccion/_components/nueva-seccion-client.tsx
# 2. Plantilla básica de página
```

```
// app/nueva-seccion/page.tsx
import { getServerSession } from 'next-auth';
import { authOptions } from '@/lib/auth';
import { redirect } from 'next/navigation';
import { NuevaSeccionClient } from './_components/nueva-seccion-client';

export default async function NuevaSeccionPage() {
  const session = await getServerSession(authOptions);
  if (!session) redirect('/auth/login');

  return <NuevaSeccionClient />;
}
```

# **Agregar Nuevo API Endpoint**

```
# 1. Crear archivo
touch app/api/nueva-ruta/route.ts
```

```
// app/api/nueva-ruta/route.ts
import { NextRequest, NextResponse } from 'next/server';
import { getServerSession } from 'next-auth';
import { authOptions } from '@/lib/auth';
import { prisma } from '@/lib/db';
export async function GET(req: NextRequest) {
  const session = await getServerSession(authOptions);
  if (!session?.user?.id) {
    return NextResponse.json({ error: 'Unauthorized' }, { status: 401 });
  }
 try {
    // Tu lógica aquí
    const data = await prisma.user.findUnique({
     where: { id: session.user.id }
    });
    return NextResponse.json({ data });
  } catch (error) {
    console.error('Error:', error);
    return NextResponse.json({ error: 'Internal error' }, { status: 500 });
 }
}
export async function POST(req: NextRequest) {
  const session = await getServerSession(authOptions);
 if (!session?.user?.id) {
    return NextResponse.json({ error: 'Unauthorized' }, { status: 401 });
 }
 try {
    const body = await req.json();
    // Tu lógica aquí
    return NextResponse.json({ success: true });
  } catch (error) {
    console.error('Error:', error);
    return NextResponse.json({ error: 'Internal error' }, { status: 500 });
}
```

### Agregar Nuevo Modelo a Base de Datos

```
# 1. Editar schema
nano prisma/schema.prisma
```

```
// Agregar al final del archivo
model NuevoModelo {
           String @id @default(cuid())
 id
 userId
           String
                    @relation(fields: [userId], references: [id], onDelete: Cascade)
 user
           User
  campo1
           String
  campo2
           Int
                    @default(0)
  createdAt DateTime @default(now())
  updatedAt DateTime @updatedAt
 @@index([userId])
}
```

```
# 2. Actualizar modelo User
# Agregar en model User:
# nuevosModelos NuevoModelo[]

# 3. Crear y aplicar migración
yarn prisma migrate dev --name agregar_nuevo_modelo

# 4. Generar cliente
yarn prisma generate
```

### **Agregar Nuevo Componente UI**

```
# 1. Crear componente
touch components/ui/nuevo-componente.tsx
```

```
import * as React from 'react';
import { cn } from '@/lib/utils';
export interface NuevoComponenteProps
  extends React.HTMLAttributes<HTMLDivElement> {
  variant?: 'default' | 'special';
}
const NuevoComponente = React.forwardRef<</pre>
 HTMLDivElement,
 NuevoComponenteProps
>(({ className, variant = 'default', ...props }, ref) => {
    <div
      ref={ref}
      className={cn(
        "base-classes",
        variant === 'default' && "default-classes",
        variant === 'special' && "special-classes",
        className
      ) }
      {...props}
    />
 );
});
NuevoComponente.displayName = "NuevoComponente";
export { NuevoComponente };
```

### DEBUGGING

### Ver Logs del Tutor IA

```
# En desarrollo, los logs aparecen en la consola donde corriste yarn dev
# Buscar por: "Error generating tutor response:"
```

#### Ver Base de Datos

```
# Abrir Prisma Studio
yarn prisma studio
# O conectar directamente con psql
psql "postgresql://role f7dfd0c44:bmSuoaocSJKxYSqWigBD5ZpRJfieebf7@db-f7df-
d0c44.db002.hosteddb.reai.io:5432/f7dfd0c44"
```

### Resetear Base de Datos (CUIDADO)

```
# Esto BORRARÁ todos los datos
yarn prisma migrate reset
# Ejecutar seeds nuevamente
yarn prisma db seed
```

### Verificar Sesión de Usuario

```
// En cualquier componente de cliente
'use client';
import { useSession } from 'next-auth/react';

export function MiComponente() {
  const { data: session, status } = useSession() || {};

  console.log('Session:', session);
  console.log('Status:', status); // "loading" | "authenticated" | "unauthenticated"
  // ...
}
```

### Ver Variables de Entorno

```
# En terminal
cat .env

# En código (solo server-side)
console.log('DB URL:', process.env.DATABASE_URL);
console.log('API Key:', process.env.ABACUSAI_API_KEY);
```

# **ESTILOS Y TEMAS**

## **Colores Principales**

### **Gradientes Comunes**

```
// Fondo de sección
className="bg-gradient-to-br from-blue-50 to-indigo-100"

// Fondo de tarjeta especial
className="bg-gradient-to-br from-indigo-500 to-purple-600"

// Texto gradiente
className="bg-gradient-to-r from-blue-600 to-purple-600 bg-clip-text text-transparent"
```

### **Animaciones Comunes**

```
import { motion } from 'framer-motion';
// Fade in
<motion.div
 initial={{ opacity: 0 }}
 animate={{ opacity: 1 }}
 transition={{ duration: 0.3 }}
// Slide up
<motion.div
 initial={{ opacity: 0, y: 20 }}
 animate={{ opacity: 1, y: 0 }}
 transition={{ duration: 0.5 }}
// Stagger children
<motion.div
  initial="hidden"
  animate="visible"
  variants={{
   hidden: { opacity: 0 },
    visible: {
      opacity: 1,
      transition: { staggerChildren: 0.1 }
    }
  }}
  {items.map(item => (
    <motion.div
      key={item.id}
      variants={{
        hidden: { opacity: 0, y: 20 },
        visible: { opacity: 1, y: 0 }
      }}
      {item.content}
    </motion.div>
  ))}
</motion.div>
```

# **TESTING MANUAL**

# **Checklist Completo**

Dashboard: [ ] Login funciona [ ] Muestra semana actual correctamente [ ] Muestra progreso correcto [ ] Muestra rachas correctas [ ] Actividades se pueden marcar/desmarcar [ ] Se otorgan puntos al completar actividad [ ] Logo redirige a dashboard
Tutor IA:  [ ] Puede iniciar conversación [ ] Puede cambiar contexto [ ] Recibe respuestas del tutor [ ] Muestra traducción (nivel A1-A2) [ ] Muestra feedback gramatical (nivel A1-B1) [ ] Historial funciona [ ] Puede crear nueva conversación [ ] Se otorgan puntos por mensaje [ ] Logo redirige a dashboard
Vocabulario: [] Muestra categorías [] Puede filtrar por categoría [] Puede filtrar por dificultad [] Muestra términos correctamente [] Puede marcar como aprendido [] Logo redirige a dashboard
Recursos: [ ] Muestra categorías [ ] Puede filtrar por categoría [ ] Enlaces funcionan [ ] Logo redirige a dashboard
Guía:  [ ] Tutorial interactivo funciona  [ ] Checklist funciona  [ ] Quiz funciona  [ ] Planificador funciona  [ ] Gráficos se muestran  [ ] Logo redirige a dashboard
Gamificación:  [ ] Muestra nivel correcto [ ] Muestra puntos correctos [ ] Muestra racha correcta [ ] Logros se desbloquean [ ] Modal de level up aparece [ ] Modal de achievement aparece [ ] Animaciones funcionan

# **MÉTRICAS Y ANALÍTICAS**

### Ver Estadísticas de Usuario

```
// En Prisma Studio o psql
SELECT
 u.email,
 u.points,
 u.level,
 u.currentStreak.
 u.bestStreak.
 COUNT(DISTINCT up.id) as completed_activities,
 COUNT(DISTINCT cc.id) as conversations,
 COUNT(DISTINCT ua.id) as achievements
FROM "User" u
LEFT JOIN "UserProgress" up ON u.id = up."userId"
LEFT JOIN "ChatConversation" cc ON u.id = cc."userId"
LEFT JOIN "UserAchievement" ua ON u.id = ua."userId"
WHERE u.email = 'user@example.com'
GROUP BY u.id, u.email, u.points, u.level, u.currentStreak, u.bestStreak;
```

#### **Ver Conversaciones Recientes**

```
SELECT
    cc.title,
    cc.context,
    cc."startedAt",
    cc."lastMessageAt",
    COUNT(cm.id) as message_count
FROM "ChatConversation" cc
LEFT JOIN "ChatMessage" cm ON cc.id = cm."conversationId"
WHERE cc."userId" = 'user_id_here'
GROUP BY cc.id, cc.title, cc.context, cc."startedAt", cc."lastMessageAt"
ORDER BY cc."lastMessageAt" DESC
LIMIT 10;
```

### Ver Progreso de Vocabulario

```
SELECT
   vc.name as category,
   COUNT(vt.id) as total_terms,
   COUNT(uvp.id) as learned_terms,
   COUNT(CASE WHEN uvp.mastered = true THEN 1 END) as mastered_terms
FROM "VocabularyCategory" vc
LEFT JOIN "VocabularyTerm" vt ON vc.id = vt."categoryId"
LEFT JOIN "UserVocabularyProgress" uvp ON vt.id = uvp."wordId" AND uvp."userId" = 'use
r_id_here'
GROUP BY vc.id, vc.name;
```



# PROBLEMAS COMUNES Y SOLUCIONES

### Error: "Module not found"

```
# Limpiar cache y reinstalar
rm -rf node_modules .next
yarn install
```

### Error: "Prisma Client not initialized"

```
# Regenerar cliente Prisma
yarn prisma generate
```

### Error: "Session is null"

```
// Verificar que el componente esté envuelto en SessionProvider
// Ver app/layout.tsx o app/providers.tsx
```

### **Error: "Hydration failed"**

```
// Verificar que no estés usando Date.now(), Math.random() en render
// Ver sección de prevención de errores de hidratación en ARQUITECTURA_TECNICA.md
```

### Error: "Select value cannot be empty"

```
// Nunca usar value="" en SelectItem
// Usar un valor por defecto válido como "all" o "default"
```

### Lentitud en desarrollo

```
# Aumentar límite de memoria de Node
NODE_OPTIONS='--max-old-space-size=4096' yarn dev
```

### Puerto 3000 ya en uso

```
# Matar proceso
lsof -ti:3000 | xargs kill -9
# 0 usar otro puerto
PORT=3001 yarn dev
```

# **SNIPPETS ÚTILES**

### Hook de Gamificación

```
import { useState, useEffect } from 'react';
export function useGamification(userId: string) {
 const [stats, setStats] = useState(null);
 const [loading, setLoading] = useState(true);
 useEffect(() => {
    async function fetchStats() {
     try {
        const res = await fetch('/api/tutor/gamification');
        const data = await res.json();
       setStats(data);
      } catch (error) {
        console.error('Error:', error);
      } finally {
        setLoading(false);
    }
   fetchStats();
 }, [userId]);
  return { stats, loading };
```

### Toast de Éxito

```
import { toast } from 'sonner';

toast.success(';Excelente!', {
  description: 'Has completado la actividad',
  duration: 3000,
});
```

### Modal de Confirmación

```
import {
 AlertDialog,
 AlertDialogAction,
 AlertDialogCancel,
 AlertDialogContent,
 AlertDialogDescription,
 AlertDialogFooter,
 AlertDialogHeader,
 AlertDialogTitle,
} from '@/components/ui/alert-dialog';
<AlertDialog open={is0pen} on0penChange={setIs0pen}>
  <AlertDialogContent>
    <AlertDialogHeader>
     <AlertDialogTitle>¿Estás seguro?
     <AlertDialogDescription>
       Esta acción no se puede deshacer.
     </AlertDialogDescription>
    </AlertDialogHeader>
    <AlertDialogFooter>
     <AlertDialogCancel>Cancelar/AlertDialogCancel>
     <AlertDialogAction onClick={handleConfirm}>
       Continuar
     </AlertDialogAction>
    </AlertDialogFooter>
  </AlertDialogContent>
</AlertDialog>
```

# 

#### Documentación

Next.js: https://nextjs.org/docs

• Prisma: https://www.prisma.io/docs

• NextAuth: https://next-auth.js.org

• Tailwind CSS: https://tailwindcss.com/docs

• Shadcn/ui: https://ui.shadcn.com

• Radix UI: https://www.radix-ui.com

• Framer Motion: https://www.framer.com/motion

#### **APIs**

· Abacus AI: https://api.abacus.ai/docs

#### Herramientas

• Prisma Studio: yarn prisma studio

• Next.js DevTools: Incluido en desarrollo

React DevTools: Extensión de Chrome

Última actualización: 09 de Octubre de 2025