Planning Individual - Desarrollador A (IA y Lógica de Negocio)

👤 Perfil y Responsabilidades Principales

Rol: AI & Business Logic Engineer
Especialización: Inteligencia Artificial, LLMs, análisis educativo y experiencia de usuario
Áreas de Dominio:

- Integración y optimización de modelos de lenguaje
- Diseño de sistemas de generación de contenido educativo
- Algoritmos de personalización y adaptación
- Análisis de datos educativos
- Experiencia de usuario con IA
- Evaluación y feedback automatizado

🎯 Objetivos Clave del Rol

1. **Crear el mejor sistema de generación de exámenes** adaptativo del mercado
2. **Optimizar costos de IA** en 50% mediante routing inteligente
3. **Lograr 90% de precisión** en evaluaciones automáticas
4. **Personalización efectiva** para cada estudiante
5. **Tiempo de respuesta <2s** para interacciones con IA

🗓️ Planning Detallado por Fases

FASE 0: Análisis y Diseño de IA (Semana 0)

Día	Tareas	Entregables	Criterios de Éxito
Lunes	• Análisis de LLMs disponibles (GPT-4, Claude, Gemini, Llama) • Comparativa de capacidades y costos • Definir casos de uso por modelo	• Matriz comparativa LLMs • Documento de selección	• ROI claro por modelo • Casos de uso mapeados
Martes	• Diseñar arquitectura de prompts • Definir templates base • Estrategia de few-shot learning	• Prompt engineering guide • Template library v0.1	• Prompts estructurados • Versionado definido
Miércoles	• Investigar vector databases • Diseñar sistema de embeddings • RAG architecture planning	• Comparativa vector DBs • Arquitectura RAG	• Solución escalable • <100ms búsqueda
Jueves	• Definir lógica de evaluación • Diseñar sistema de scoring • Métricas de aprendizaje	• Algoritmos de evaluación • Rúbricas automatizadas	• Evaluación objetiva • Feedback actionable
Viernes	• Workshop con Dev B sobre APIs • Definir flujos de datos • Establecer SLAs de IA	• API contracts • Flujos documentados	• Interfaces claras • SLAs realistas

FASE 1: Integración Base de IA (Semanas 1-2)

Semana 1: APIs y Modelos Base

Tarea	Descripción Detallada	Dependencias	Riesgos
1.1 Cliente Multi-LLM	• Abstracción para múltiples providers • Gestión de API keys • Retry logic y fallbacks • Rate limiting por provider	API Gateway setup	Límites de rate imprevistos
1.2 Servicio de Autenticación	• JWT implementation • Roles y permisos • Refresh tokens • Session management	Redis disponible	Seguridad de tokens
1.3 Vector Database Setup	• Pinecone/Weaviate/Qdrant • Índices por tipo de contenido • Estrategia de chunking • Metadata structure	Storage disponible	Costos de escalado
1.4 Prompt Management	• Versionado de prompts • A/B testing framework • Variables y templates • Prompt optimization	Base de datos	Gestión de versiones
1.5 Primeros Endpoints	• POST /analyze-document • POST /generate-exam • GET /study-plan • WebSocket para chat	API Gateway	Complejidad de WS

Semana 2: Lógica de Negocio Core

Tarea	Descripción Detallada	Dependencias	Riesgos
2.1 Analizador de Estudiantes	• Perfil learning style • Análisis de fortalezas/debilidades • Histórico de rendimiento • Predicciones ML	DB esquemas	Privacidad de datos
2.2 Motor de Personalización	• Adaptación de contenido • Dificultad dinámica • Recomendaciones personalizadas • Learning paths	Perfil estudiante	Over-personalización
2.3 Sistema de Embeddings	• Generación para documentos • Chunking strategies • Semantic search • Similarity scoring	Vector DB	Calidad embeddings
2.4 Cache de Prompts	• Respuestas frecuentes • Semantic caching • TTL inteligente • Invalidación selectiva	Redis cluster	Cache coherence
2.5 Analytics Base	• Event tracking • Métricas de uso • Performance metrics • User journey	Analytics DB	Volume de datos

FASE 2: Generación de Contenido (Semanas 3-4)

Semana 3: Sistema de Exámenes

Tarea	Descripción Detallada	Skills Necesarios	Métricas de Éxito
3.1 Generador Tipo Test	• Multiple choice questions • Distractores inteligentes • Balanceo de dificultad • Validación automática	Prompt engineering	95% preguntas válidas
3.2 Generador Desarrollo	• Preguntas abiertas • Rúbricas automáticas • Ejemplos de respuesta • Criterios de evaluación	NLP, LLMs	Rúbricas precisas
3.3 Generador Problemas	• Problemas matemáticos • Casos prácticos • Soluciones paso a paso • Variaciones automáticas	Domain expertise	Soluciones correctas
3.4 Mixer de Exámenes	• Combinación de tipos • Balance por temas • Tiempo estimado • Difficulty curve	Algoritmos	Exámenes balanceados
3.5 Adaptador de Estilo	• Análisis exámenes previos • Extracción de patrones • Aplicación de estilo • Validación formato	ML, Pattern matching	90% similitud estilo

Semana 4: Evaluación y Feedback

Tarea	Descripción Detallada	Skills Necesarios	Métricas de Éxito
4.1 Corrector Automático	• NLP para respuestas texto • Partial scoring • Detección de conceptos • Explicación de errores	NLP, LLMs	85% accuracy vs humano
4.2 Generador de Feedback	• Feedback personalizado • Sugerencias de mejora • Enlaces a recursos • Motivación adaptativa	Psychology, LLMs	Feedback actionable
4.3 Sistema de Métricas	• Learning analytics • Progress tracking • Predicción de éxito • Early warning system	ML, Statistics	Predicciones 80% accurate
4.4 Recomendador de Estudio	• Gaps identification • Material prioritization • Schedule optimization • Adaptive planning	ML algorithms	Mejora 20% resultados

FASE 3: Optimización de IA (Semanas 5-6)

Semana 5: Routing y Optimización

Tarea	Descripción Detallada	Herramientas	Objetivo
5.1 Router Inteligente	• Clasificación de queries • Modelo selection logic • Cost/quality balance • Fallback strategies	RouteLLM, ML	50% reducción costo
5.2 Fine-tuning Models	• Dataset preparation • Training pipelines • Evaluation metrics • Model versioning	OpenAI API, Llama	+15% accuracy
5.3 Prompt Optimization	• Automatic prompt tuning • A/B testing results • Template refinement • Context optimization	DSPy, LangChain	-30% tokens used
5.4 Batch Processing	• Async generation • Priority queues • Bulk operations • Progress tracking	Celery, Redis	10x throughput
5.5 Edge Cases Handling	• Error recovery • Content filtering • Safety checks • Quality assurance	LLM guards	<1% bad outputs

Semana 6: Caché y Performance

Tarea	Descripción Detallada	Herramientas	Objetivo
6.1 Semantic Cache	• Embedding-based cache • Similarity threshold • Cache warming • Hit rate optimization	Pinecone, Redis	70% cache hit
6.2 Response Streaming	• SSE implementation • Chunked responses • Progressive rendering • Perceived performance	FastAPI, SSE	<500ms first byte
6.3 Precomputed Responses	• Common questions DB • Template responses • Quick replies • Instant feedback	PostgreSQL	<50ms common queries
6.4 Model Quantization	• Local model optimization • Quantization testing • Performance benchmarks • Accuracy trade-offs	GGUF, llama.cpp	2x speed, -5% acc

FASE 4: Features Avanzadas (Semanas 7-8)

Semana 7: Agentes y Automación

Tarea	Descripción Detallada	Complejidad	Prioridad
7.1 Agente de Estudio	• Multi-step reasoning • Tool usage (calculator, search) • Context management • Goal tracking	Alta	P0
7.2 Tutor Conversacional	• Context-aware chat • Socratic method • Adaptive responses • Emotion detection	Alta	P0
7.3 Auto-curriculum	• Learning path generation • Prerequisite mapping • Pacing optimization • Milestone setting	Media	P1
7.4 Collaborative Learning	• Peer matching • Group study sessions • Shared resources • Competition elements	Media	P2

Semana 8: Análisis Avanzado

Tarea	Descripción Detallada	Complejidad	Prioridad
8.1 Predictive Analytics	• Success prediction • Risk identification • Intervention triggers • Outcome modeling	Alta	P0
8.2 Knowledge Graphs	• Concept mapping • Relationship extraction • Prerequisite chains • Gap analysis	Alta	P1
8.3 Adaptive Testing	• CAT implementation • IRT models • Real-time adaptation • Precision stopping	Alta	P1
8.4 Learning Insights	• Deep analytics • Pattern recognition • Recommendations engine • Parent dashboards	Media	P2

FASE 5: Innovación y Escala (Semanas 9-10)

Semana 9: Características Experimentales

Tarea	Descripción Detallada	Stack Técnico	Innovación
9.1 Voice Interaction	• Speech-to-text • Oral examinations • Pronunciation feedback • Audio lessons	Whisper API	Accessibility
9.2 AR/VR Integration	• 3D visualizations • Immersive learning • Spatial memory • Gamification	Three.js, WebXR	Engagement
9.3 Multi-modal Learning	• Image understanding • Diagram generation • Video analysis • Mixed media	GPT-4V, DALL-E	Comprehension
9.4 Peer Learning AI	• Student matching • Collaborative AI • Group dynamics • Social learning	Graph algorithms	Retention

Semana 10: Preparación para Escala

Tarea	Descripción Detallada	Stack Técnico	Impacto
10.1 A/B Testing Platform	• Feature flags • Experiment design • Statistical significance • Rollout strategies	LaunchDarkly	Data-driven
10.2 ML Pipeline	• Feature engineering • Model training • Deployment automation • Monitoring	MLflow, Kubeflow	Continuous improvement
10.3 API Documentation	• OpenAPI specs • Interactive docs • Code examples • SDKs generation	Swagger, Postman	Developer adoption
10.4 Knowledge Transfer	• Technical documentation • Architecture decisions • Best practices guide • Training sessions	Confluence	Team scalability

📊 KPIs y Métricas de Éxito

Métricas de IA

- **Accuracy de evaluación**: >85% vs evaluación humana
- **Costo por usuario**: <$0.50/mes en LLM
- **Latencia generación**: <2s para respuestas
- **Cache hit rate**: >70% consultas similares

Métricas de Producto

- **Satisfacción usuario**: NPS >50
- **Engagement**: 5+ sesiones/semana por usuario
- **Mejora académica**: +20% en calificaciones

- **Retención**: >80% usuarios activos mensual

Métricas de Calidad

- **Calidad de exámenes**: 95% válidos sin revisión
- **Personalización efectiva**: 90% usuarios reportan mejora
- **Tiempo ahorro profesor**: 10h/semana
- **Cobertura curricular**: 100% temas

🛠️ Stack Tecnológico Principal

AI/ML Stack

- **LLMs**: OpenAI, Anthropic, Google, Open models
- **Embeddings**: OpenAI Ada, Sentence Transformers
- **Vector DB**: Pinecone/Weaviate/Qdrant
- **ML Framework**: PyTorch, Transformers

Backend Framework

- **API**: FastAPI + Pydantic
- **Async**: asyncio, aiohttp
- **Task Queue**: Celery + Redis
- **WebSockets**: Socket.io

AI Tools

- **Orchestration**: LangChain/LlamaIndex
- **Optimization**: RouteLLM, vLLM
- **Evaluation**: Ragas, LangSmith
- **Fine-tuning**: OpenAI API, Axolotl

Data & Analytics

- **Analytics**: Mixpanel/Amplitude
- **Experiments**: Statsig/LaunchDarkly
- **Monitoring**: Weights & Biases
- **Data Pipeline**: Apache Beam

🚀 Estrategias de Optimización

Reducción de Costos LLM

1. **Clasificación de Queries**
   - Simple → Modelo pequeño/caché
   - Complejo → GPT-4/Claude
   - Creativo → Modelo especializado
2. **Caché Inteligente**
   - Semantic similarity matching
   - Template-based responses
   - Precomputed common queries
3. **Batch Processing**
   - Agrupar requests similares
   - Off-peak processing
   - Bulk generation

Mejora de Calidad

1. **Fine-tuning Continuo**
   - Feedback loop de usuarios
   - Dataset curation
   - Model evaluation
2. **Prompt Engineering**
   - A/B testing prompts
   - Few-shot examples
   - Chain-of-thought
3. **Ensemble Methods**
   - Multiple model voting
   - Confidence scoring
   - Quality filtering

🧠 Algoritmos Clave a Implementar

Sistema de Evaluación

python

```python
# Pseudocódigo conceptual
class ExamEvaluator:
    def evaluate_answer(self, question, student_answer, rubric):
        # 1. Extraer conceptos clave
        # 2. Comparar con rúbrica
        # 3. Partial credit calculation
        # 4. Generar feedback específico
        pass
```

Personalización Adaptativa

```python
# Pseudocódigo conceptual
class AdaptiveLearning:
    def next_content(self, student_profile, performance_history):
        # 1. Calcular nivel actual
        # 2. Identificar gaps
        # 3. Seleccionar siguiente contenido
        # 4. Ajustar dificultad
        pass
```

Router de Modelos

```python
# Pseudocódigo conceptual
class ModelRouter:
    def route_query(self, query, context, constraints):
        # 1. Clasificar tipo de query
        # 2. Estimar complejidad
        # 3. Check caché
        # 4. Seleccionar modelo óptimo
        pass
```

🚨 Riesgos y Mitigaciones Específicos

Riesgo	Probabilidad	Impacto	Mitigación
Costos LLM explotan	Alta	Alto	Límites duros, alertas, caché agresivo
Hallucinations en exámenes	Media	Alto	Validation layers, human review option
Sesgo en evaluaciones	Media	Alto	Bias testing, diverse training data
Latencia inaceptable	Baja	Alto	Streaming, perceived performance tricks
Cambios en APIs LLM	Media	Medio	Abstraction layer, multiple providers

📚 Plan de Formación Continua

Cursos y Certificaciones

1. **Fast.ai Practical Deep Learning**
2. **Stanford CS224N (NLP)**
3. **Anthropic's Constitutional AI**
4. **Google's Machine Learning Crash Course**

Investigación y Papers

- Attention mechanisms y transformers
- Few-shot y zero-shot learning
- Retrieval Augmented Generation
- Educational technology research

Comunidad y Networking

- Participar en Hugging Face community
- Contribuir a open source LLM projects
- Asistir a conferencias AI/EdTech
- Blog técnico sobre learnings

🎯 Hitos Clave por Fase

Fase 1: Foundation

- ✓ Primera generación de examen exitosa
- ✓ API respondiendo <2s
- ✓ 3 tipos de LLM integrados

Fase 2: Core Features

- ✓ 95% accuracy en tipo test
- ✓ Feedback personalizado funcionando
- ✓ 1000 exámenes generados

Fase 3: Optimization

- ✓ 50% reducción en costos
- ✓ 70% cache hit rate
- ✓ <500ms perceived latency

Fase 4: Advanced

- ✓ Agente tutor conversacional
- ✓ Predicciones 80% accurate
- ✓ Multi-modal support

Fase 5: Scale

- ✓ A/B testing platform live
- ✓ ML pipeline automatizado
- ✓ 100% documentación

✅ Definition of Done para IA

Para cada feature de IA:

- [ ] Prompt optimizado y versionado
- [ ] Evaluación contra ground truth
- [ ] Métricas de calidad definidas
- [ ] Fallbacks implementados
- [ ] Costos calculados y optimizados
- [ ] Tests de edge cases
- [ ] Documentación de limitaciones
- [ ] Monitoring en producción

🤝 Puntos de Sincronización con Dev B

Semanal

- API contracts review
- Performance bottlenecks
- Infrastructure needs
- Cost optimization

Quincenal

- Architecture decisions
- Integration testing
- Deployment planning
- Retrospectiva técnica

Mensual

- Roadmap alignment

- Technical debt review
- Innovation planning
- Knowledge sharing