|  |
| --- |
| DUOC UC - ESCUELA DE INFORMÁTICA Y TELECOMUNICACIONES |
| Especificación de Requisitos de Software |
| *InterviewAI: Simulador de Entrevistas Laborales con Inteligencia Artificial* |
|  |
|  |
|  |

**Versión:** 1.0  
**Fecha:** 16 de Octubre de 2025  
**Estado:** Aprobado

Especificación de Requisitos según estándar de IEEE 830.

**Equipo de Desarrollo:**

* Cristian Álvarez - AI Specialist & QA Lead
* Matías Coloma Contreras - Project Leader & Backend Developer
* Ulises Muñoz - Frontend Developer & UX/UI Designer

**Sección:** 008V  
**Profesor Guía:** Félix Eduardo Cifuentes

**CONTROL DE VERSIONES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| *16/10/2025* | *1.0* | *Matías/Cristian/Ulises* | *complementación sección 1* |
|  |  |  |  |

|  |
| --- |
|  |

**Contenido**

[**Ficha del documento**](#_heading=h.1fob9te) 3

[**1.Introducción**](#_heading=h.3znysh7) **4**

[1.1.](#_heading=h.2et92p0) Propósito 4

[1.2.](#_heading=h.tyjcwt) Alcance del Proyecto 4

[1.3.](#_heading=h.3dy6vkm) Definiciones, Acrónimos y Abreviaturas 5

[1.4.](#_heading=h.1t3h5sf) Referencias 6

[1.5.](#_heading=h.4d34og8) Visión General del Documento 6

[**2.**](#_heading=h.2s8eyo1)**Descripción General 6**

[2.1.](#_heading=h.17dp8vu) Perspectiva del Producto 7

[2.2.](#_heading=h.3rdcrjn) Funciones del Producto 7

[2.3.](#_heading=h.26in1rg) Características de los Usuarios 8

[2.4.](#_heading=h.lnxbz9) Restricciones Generales 8

[2.5.](#_heading=h.35nkun2) Suposiciones y Dependencias 9

[**3.**](#_heading=h.44sinio)**Requisitos Específicos 9**

[3.1](#_heading=h.2jxsxqh) Requisitos Funcionales 10

[3.2](#_heading=h.2xcytpi) Requisitos no funcionales 12

[3.3](#_heading=h.1ci93xb) Requisitos de Interfaz 13

**4. Casos de Usos**

**5. Requisitos de Rendimiento**

**6. Restricciones de Diseño**

**7. Atributos del Sistema**

**8. Apéndices**

# Ficha del documento

Documento validado por las partes en fecha:

|  |  |  |
| --- | --- | --- |
| Por el cliente |  | Por la empresa suministradora |
| [Firma] |  | [Firma] |
| Sr./Sra. |  | Sr./Sra. |

# 

# 1. Introducción

## 1.1. Propósito

## Este documento de Especificación de Requisitos de Software (ERS) describe las especificaciones funcionales y no funcionales del sistema InterviewAI, una plataforma web inteligente diseñada para simular entrevistas laborales mediante inteligencia artificial.

El documento está dirigido a:

* **Equipo de desarrollo:** Como guía técnica para la implementación
* **Evaluadores académicos:** Para validación del alcance del proyecto de título
* **Stakeholders:** Para comprensión del sistema y sus capacidades
* **Usuarios finales:** Para entendimiento de funcionalidades disponibles

## 1.2. Alcance del Proyecto

**Nombre del Producto:**

**InterviewAI - Simulador de Entrevistas Laborales con Inteligencia Artificial**

**Objetivos Principales**

* Proporcionar una plataforma accesible para práctica de entrevistas laborales del sector tecnológico
* Reducir la ansiedad y mejorar la preparación de candidatos mediante simulaciones realistas
* Ofrecer retroalimentación inteligente y personalizada basada en análisis de IA
* Democratizar el acceso a recursos de preparación profesional

**Beneficios del Sistema**

* **Para usuarios:** Práctica ilimitada, feedback instantáneo, seguimiento de progreso
* **Para el mercado laboral:** Candidatos mejor preparados, procesos de selección más eficientes
* **Para la industria:** Reducción de brechas de habilidades blandas en profesionales técnicos

**Alcance del Desarrollo**

**El sistema incluirá:**

* **✅** Gestión completa de usuarios y perfiles
* ✅ Motor de generación de preguntas contextualizadas por IA
* ✅ Avatar conversacional con sincronización de voz
* ✅ Captura y análisis de respuestas verbales
* ✅ Sistema de retroalimentación inteligente
* ✅ Dashboard de métricas y progreso
* ✅ Historial de entrevistas realizadas
* ✅ Sistema de gamificación básico

**El sistema NO incluirá (fuera de alcance):**

* **❌** Entrevistas en vivo con reclutadores reales
* ❌ Conexión directa con plataformas de empleo (LinkedIn, Indeed)
* ❌ Sistema de pago o suscripciones
* ❌ Aplicación móvil nativa (solo web responsive)
* ❌ Soporte multi-idioma en fase inicial (solo español)

## 1.3. Definiciones, Acrónimos y Abreviaturas

| **Término** | **Definición** |
| --- | --- |
| **IA** | Inteligencia Artificial |
| **API** | Application Programming Interface (Interfaz de Programación de Aplicaciones) |
| **LLM** | Large Language Model (Modelo de Lenguaje de Gran Escala) |
| **NLP** | Natural Language Processing (Procesamiento de Lenguaje Natural) |
| **STT** | Speech-to-Text (Conversión de Voz a Texto) |
| **TTS** | Text-to-Speech (Conversión de Texto a Voz) |
| **UI/UX** | User Interface / User Experience (Interfaz y Experiencia de Usuario) |
| **WebRTC** | Web Real-Time Communication |
| **JWT** | JSON Web Token |
| **CRUD** | Create, Read, Update, Delete |
| **QA** | Quality Assurance (Aseguramiento de Calidad) |
| **RF** | Requisito Funcional |
| **RNF** | Requisito No Funcional |
| **Avatar IA** | Representación digital animada que simula un entrevistador |
| **Feedback** | Retroalimentación o evaluación de respuestas |
| **Dashboard** | Panel de control con visualización de métricas |
| **Método STAR** | Situation, Task, Action, Result (metodología para respuestas estructuradas) |
| **Prompt** | Instrucción o contexto dado a un modelo de IA |
| **Token** | Unidad de procesamiento de texto en modelos de lenguaje |

## 1.4. Referencias

* **IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications**
* **OpenAI GPT-4 Technical Report (2024)**
* **Google Cloud Speech-to-Text API Documentation (2024)**
* **D-ID Creative Reality Studio API Documentation (2024)**
* **React Documentation - Building User Interfaces (2024)**
* **Web Speech API Documentation - MDN Web Docs (2024)**
* **SWEBOK v3.0: Software Engineering Body of Knowledge**
* **Documento de Definición de Proyecto APT (Fase 1)**

## 1.5. Visión General del Documento

**Este documento está organizado en 8 secciones principales:**

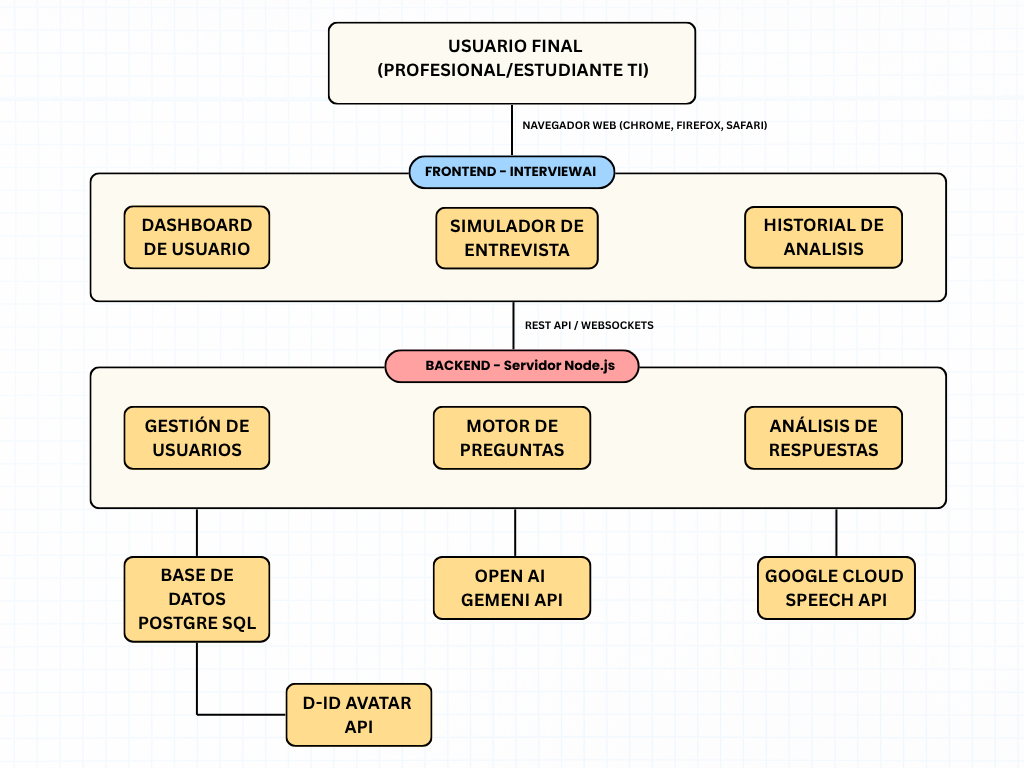
* **Sección 1-2**: Contexto general del proyecto, descripción del producto y características de usuarios
* **Sección 3**: Especificación detallada de requisitos funcionales y no funcionales
* **Sección 4**: Casos de uso del sistema con diagramas y descripciones
* **Sección 5-7**: Requisitos técnicos de rendimiento, restricciones de diseño y atributos de calidad
* **Sección 8**: Apéndices con información complementaria y trazabilidad

# 2. Descripción General

# 2.1. Perspectiva del Producto

InterviewAI es un sistema web independiente que no requiere integración con sistemas existentes en su primera versión. El producto se posiciona como una herramienta de auto preparación para candidatos del sector tecnológico.

**Contexto del Sistema:**



**Interfaces del Sistema**

**Interfaces de Usuario:**

* Interfaz web responsive compatible con desktop y tablets
* Dashboard visual con gráficos de progreso
* Interfaz de simulación de entrevista con video del avatar
* Panel de configuración de perfil y preferencias

**Interfaces de Hardware:**

* Micrófono para captura de voz
* Altavoces o audífonos para reproducción de audio
* Webcam (opcional, para futuras funcionalidades)
* Conexión a internet estable (mínimo 5 Mbps)

**Interfaces de Software:**

* API OpenAI GPT-4 / Google Gemini para generación de contenido
* Google Cloud Speech-to-Text para transcripción
* D-ID API para animación de avatar
* PostgreSQL para persistencia de datos
* Servicios de hosting cloud (Vercel/Railway)

**Interfaces de Comunicación:**

* Protocolo HTTPS para comunicación segura
* WebSockets para comunicación en tiempo real
* REST API para operaciones CRUD
* OAuth 2.0 para autenticación (futura implementación)

## 

## 2.2. Funciones del Producto

**Funcionalidades Principales**

**1. Gestión de Perfiles de Usuario**

* Registro y autenticación segura
* Personalización de perfil técnico (área, nivel de experiencia)
* Gestión de preferencias de entrevista
* Recuperación de contraseñas

**2. Simulación de Entrevistas Interactivas**

* Generación de preguntas contextualizadas por IA
* Avatar conversacional con sincronización de voz y gestos
* Captura y transcripción de respuestas verbales
* Adaptación dinámica de dificultad según desempeño

**3. Análisis y Retroalimentación Inteligente**

* Evaluación multidimensional de respuestas
* Generación de feedback constructivo personalizado
* Identificación de fortalezas y áreas de mejora
* Sugerencias prácticas de optimización

**4. Seguimiento de Progreso**

* Dashboard con métricas de rendimiento
* Gráficos de evolución temporal
* Comparación con benchmarks de industria
* Historial completo de entrevistas realizadas

**5. Sistema de Motivación**

* Gamificación con badges y logros
* Establecimiento de objetivos personales
* Rachas de práctica consecutiva
* Niveles de competencia progresivos

## 

## 2.3. Características de los Usuarios

**Perfil de Usuario Principal**

**Tipo de Usuario:** Profesionales y Estudiantes del Sector Tecnológico

| **Característica** | **Descripción** |
| --- | --- |
| **Edad** | 20-35 años (rango principal) |
| **Nivel Educativo** | Estudiantes universitarios, egresados, profesionales con formación técnica |
| **Experiencia Técnica** | Intermedia a avanzada en tecnologías de la información |
| **Experiencia con Sistemas** | Alta familiaridad con aplicaciones web y herramientas digitales |
| **Motivación Principal** | Mejorar habilidades de entrevista, prepararse para procesos de selección |
| **Contexto de Uso** | Práctica individual desde casa u oficina, sesiones de 15-45 minutos |
| **Idioma** | Español (fase inicial) |
| **Dispositivos** | Computadores desktops/laptop con navegador moderno |

**Perfiles Específicos de Usuario**

**1. Estudiante en Búsqueda de Primera Oportunidad**

* **Necesidades:** Conocer estructura de entrevistas técnicas, reducir ansiedad, practicar respuestas comportamentales
* **Nivel de Experiencia:** Junior (0-2 años)
* **Áreas de Interés:** Desarrollo web, QA, DevOps básico
* **Frecuencia de Uso:** 2-3 veces por semana antes de procesos reales

**2. Profesional en Transición de Carrera**

* **Necesidades:** Actualizar conocimientos, practicar preguntas de nivel senior, mejorar storytelling profesional
* **Nivel de Experiencia:** Semi-Senior a Senior (2+ años)
* **Áreas de Interés:** Arquitectura de software, liderazgo técnico, especialidades avanzadas
* **Frecuencia de Uso:** Sesiones intensivas previo a aplicaciones específicas

**3. Profesional en Búsqueda Activa**

* **Necesidades:** Simulación realista de entrevistas específicas de empresas, feedback rápido y accionable
* **Nivel de Experiencia:** Variable (Junior a Senior)
* **Áreas de Interés:** Preparación multi-área (técnica + comportamental)
* **Frecuencia de Uso:** Uso intensivo durante período de búsqueda (diario o inter-diario)

## 

**Restricciones Técnicas**

**RT-01: Plataforma de Desarrollo**

* El sistema debe desarrollarse como aplicación web (no nativa)
* Frontend: React.js o Vue.js
* Backend: Node.js con Express o NestJS
* Base de datos: PostgreSQL

**RT-02: Compatibilidad de Navegadores**

* Soporte obligatorio: Chrome 90+, Firefox 88+, Safari 14+, Edge 90+
* No se garantiza compatibilidad con Internet Explorer

**RT-03: Dependencia de APIs Externas**

* El sistema depende críticamente de APIs de terceros (OpenAI, Google Cloud, D-ID)
* Se deben implementar mecanismos de fallback en caso de caída de servicios

**RT-04: Requisitos de Conexión**

* Conexión a internet mínima: 5 Mbps para funcionalidad completa
* Latencia máxima recomendada: 200ms

**Restricciones Operacionales**

**RO-01: Disponibilidad de APIs**

* Uso limitado por cuotas de APIs gratuitas/estudiantiles
* Costos incrementales por uso intensivo deben ser monitoreados

**RO-02: Almacenamiento de Datos**

* Audio de respuestas no se almacena permanentemente (solo transcripciones)
* Límite de retención de historial: últimas 50 entrevistas por usuario

**RO-03: Privacidad y GDPR**

* Los datos de usuario deben ser anonimizables
* El usuario debe poder solicitar eliminación completa de datos

**Restricciones Regulatorias**

**RR-01: Protección de Datos Personales**

* Cumplimiento con Ley N° 19.628 sobre Protección de la Vida Privada (Chile)
* Implementación de políticas de privacidad y términos de uso

**RR-02: Uso Ético de IA**

* El sistema no debe perpetuar sesgos discriminatorios
* Transparencia sobre uso de IA en evaluaciones
* No se reemplaza criterio humano en decisiones de contratación reales

**Restricciones de Proyecto**

**RP-01: Tiempo de Desarrollo**

* Duración del proyecto: 18 semanas (un semestre académico)
* Fecha límite de entrega: Final del semestre 2025

**RP-02: Recursos Humanos**

* Equipo fijo de 3 desarrolladores
* Distribución de roles pre-establecida

**RP-03: Presupuesto**

* Presupuesto limitado a créditos estudiantiles gratuitos
* Sin financiamiento para servicios premium

**2.5 Suposiciones y Dependencias**

**Suposiciones**

**SA-01: Disponibilidad Tecnológica**

* Se asume que las APIs de IA (OpenAI, Gemini) mantendrán disponibilidad >95%
* Se asume que las cuotas gratuitas serán suficientes para desarrollo y pruebas

**SA-02: Comportamiento de Usuarios**

* Los usuarios tienen conocimiento básico de uso de aplicaciones web
* Los usuarios tienen acceso a dispositivos con micrófono funcional
* Los usuarios practican en entornos con ruido controlado

**SA-03: Infraestructura de Usuario**

* Conexión a internet estable durante sesiones de entrevista
* Navegador actualizado con soporte para Web Speech API

**Dependencias**

**D-01: APIs de Terceros**

* **OpenAI GPT-4 / Google Gemini**: Generación de preguntas y análisis de respuestas
* **Google Cloud Speech-to-Text**: Transcripción de audio
* **D-ID API**: Animación de avatar conversacional
* **Impacto**: Falla en estas APIs afecta funcionalidad core del sistema

**D-02: Servicios de Hosting**

* **Vercel/Netlify**: Hosting del frontend
* **Railway/Heroku**: Hosting del backend
* **Supabase/Neon**: Hosting de base de datos PostgreSQL
* **Impacto**: Rendimiento y disponibilidad del sistema

**D-03: Frameworks y Librerías**

* **React.js / Vue.js**: Framework frontend
* **Node.js**: Runtime backend
* **Express/NestJS**: Framework backend
* **Chart.js / Recharts**: Visualización de datos
* **Impacto**: Cambios en versiones pueden requerir refactorización

**D-04: Aprobaciones Académicas**

* Validación del alcance por profesor guía
* Cumplimiento de rúbricas de evaluación del proyecto de título
* **Impacto**: Modificaciones de alcance según feedback académico

# 3. Requisitos Específicos

## 3.1 Requisitos Funcionales

**Módulo 1: Gestión de Usuarios**

**RF-001: Registro de Usuario**

**Prioridad:** Alta **| Complejidad:** Media

**Descripción:**  
El sistema debe permitir el registro de nuevos usuarios proporcionando información básica de perfil.

**Entradas:**

* Nombre completo
* Correo electrónico
* Contraseña (mínimo 8 caracteres, debe incluir mayúsculas, minúsculas y números)
* Área técnica de interés (Frontend, Backend, Full-Stack, DevOps, QA, Data Science, etc.)
* Nivel de experiencia (Junior 0-2 años, Semi-Senior 2-5 años, Senior 5+ años)

**Proceso:**

* Usuario completa formulario de registro
* Sistema valida formato de correo electrónico
* Sistema verifica que correo no esté registrado previamente
* Sistema valida fortaleza de contraseña
* Sistema encripta contraseña con bcrypt
* Sistema crea perfil de usuario en base de datos
* Sistema envía email de confirmación (opcional en v1.0)

**Salidas:**

* Confirmación de registro exitoso
* Redirección a página de inicio de sesión o dashboard

**Criterios de Aceptación:**

* **CA-001.1:** El sistema debe rechazar correos con formato inválido
* **CA-001.2:** El sistema debe rechazar contraseñas que no cumplan requisitos mínimos
* **CA-001.3:** El sistema debe impedir registro con correos duplicados
* **CA-001.4:** El perfil de usuario debe crearse con valores por defecto para campos opcionales
* **CA-001.5:** La contraseña debe almacenarse encriptada (nunca en texto plano)

**Mensajes de Error:**

* "El correo electrónico ya está registrado"
* "La contraseña debe tener mínimo 8 caracteres e incluir mayúsculas, minúsculas y números"

**RF-002: Autenticación de Usuario**

**Prioridad:** Alta | **Complejidad:** Media

**Descripción:**  
El sistema debe permitir a usuarios registrados iniciar sesión de forma segura.

**Entradas:**

* Correo electrónico
* Contraseña

**Proceso:**

* Usuario ingresa credenciales
* Sistema busca usuario por correo electrónico
* Sistema verifica contraseña usando bcrypt
* Sistema genera token JWT con expiración de 24 horas
* Sistema registra timestamp de último inicio de sesión
* Sistema retorna token y datos básicos de perfil

**Salidas:**

* Token de autenticación JWT
* Datos de perfil de usuario
* Redirección a dashboard principal

**Criterios de Aceptación:**

* CA-002.1: El sistema debe bloquear la cuenta tras 5 intentos fallidos consecutivos
* CA-002.2: El token JWT debe incluir ID de usuario, rol y timestamp de emisión
* CA-002.3: El sistema debe mantener la sesión activa por 24 horas
* CA-002.4: El sistema debe validar token en cada petición protegida

**Mensajes de Error:**

* "Credenciales incorrectas"
* "Cuenta bloqueada por múltiples intentos fallidos. Intente en 15 minutos"
* "Sesión expirada. Por favor, inicie sesión nuevamente"

**RF-003: Recuperación de Contraseña**

**Prioridad:** Media | **Complejidad:** Media

**Descripción:**  
El sistema debe permitir a usuarios recuperar acceso a su cuenta mediante reset de contraseña.

**Entradas:**

* Correo electrónico del usuario

**Proceso:**

* Usuario solicita recuperación ingresando correo
* Sistema verifica existencia de cuenta
* Sistema genera token único de recuperación (válido 1 hora)
* Sistema envía email con enlace de recuperación
* Usuario accede al enlace y proporciona nueva contraseña
* Sistema valida token y actualiza contraseña

**Salidas:**

* Email con enlace de recuperación
* Confirmación de cambio de contraseña exitoso

**Criterios de Aceptación:**

* CA-003.1: El token de recuperación debe expirar en 1 hora
* CA-003.2: El token debe ser de uso único (invalidarse tras ser usado)
* CA-003.3: La nueva contraseña debe cumplir requisitos mínimos de seguridad
* CA-003.4: El sistema no debe revelar si un correo existe o no (prevención de enumeración)

**RF-004: Gestión de Perfil de Usuario**

**Prioridad:** Media | **Complejidad:** Baja

**Descripción:**  
El sistema debe permitir a usuarios visualizar y editar su información de perfil.

**Entradas:**

* Datos editables: Nombre, área técnica, nivel de experiencia, foto de perfil (opcional)
* Datos no editables: Correo electrónico, fecha de registro

**Proceso:**

* Usuario accede a sección "Mi Perfil"
* Sistema muestra datos actuales
* Usuario modifica campos deseados
* Sistema valida cambios
* Sistema actualiza información en base de datos
* Sistema muestra confirmación de actualización

**Salidas:**

* Vista de perfil actualizada
* Mensaje de confirmación

**Criterios de Aceptación:**

* CA-004.1: Los cambios deben reflejarse inmediatamente en la interfaz
* CA-004.2: El sistema debe mantener historial de cambios (auditoría)
* CA-004.3: La foto de perfil debe tener límite de 2MB y formatos JPG/PNG
* CA-004.4: El usuario debe poder eliminar su cuenta (solicitud de eliminación)

**Módulo 2: Configuración de Entrevistas**

**RF-005: Creación de Sesión de Entrevista**

**Prioridad:** Alta | **Complejidad:** Media

**Descripción:**  
El sistema debe permitir al usuario configurar parámetros para una nueva sesión de entrevista.

**Entradas:**

* Área técnica (Frontend, Backend, Full-Stack, DevOps, QA, Data Science, Mobile, Cloud, Cybersecurity)
* Nivel de experiencia (Junior, Semi-Senior, Senior)
* Tipo de entrevista (Técnica, Comportamental, Mixta)
* Duración estimada (15, 30, 45 minutos)
* Empresa objetivo (opcional - para personalización)

**Proceso:**

* Usuario accede a "Nueva Entrevista"
* Sistema presenta formulario de configuración
* Usuario selecciona parámetros deseados
* Sistema valida selecciones
* Sistema genera sesión con ID único
* Sistema pre-carga banco de preguntas relevantes

**Salidas:**

* ID de sesión generado
* Confirmación de configuración
* Redirección a sala de entrevista

**Criterios de Aceptación:**

* CA-005.1: Todos los campos son obligatorios excepto "Empresa objetivo"
* CA-005.2: El sistema debe sugerir parámetros basados en historial del usuario
* CA-005.3: La sesión debe guardarse, aunque el usuario no la complete
* CA-005.4: El sistema debe permitir modificar configuración antes de iniciar

**Módulo 3: Simulación de Entrevista**

**RF-006: Presentación de Avatar Conversacional**

**Prioridad:** Alta | **Complejidad:** Alta

**Descripción:**  
El sistema debe presentar un avatar digital que simule visualmente a un entrevistador profesional.

**Entradas:**

* Configuración de avatar (género, apariencia - en versión futura)
* Script de pregunta generado por IA

**Proceso:**

* Sistema carga avatar mediante D-ID API
* Sistema convierte texto de pregunta a audio (TTS)
* D-ID sincroniza movimientos labiales con audio
* Sistema reproduce animación en interfaz del usuario

**Salidas:**

* Video del avatar pronunciando la pregunta
* Audio sincronizado
* Transcripción de la pregunta (para accesibilidad)

**Criterios de Aceptación:**

* CA-006.1: El avatar debe tener apariencia profesional y neutral
* CA-006.2: Los movimientos labiales deben estar sincronizados con el audio
* CA-006.3: El sistema debe incluir expresiones faciales naturales (asentimientos, parpadeo)
* CA-006.4: El avatar debe cargarse en menos de 3 segundos
* CA-006.5: Debe existir fallback (audio sin video) en caso de error de API

**RF-007: Generación Inteligente de Preguntas**

**Prioridad:** Alta | **Complejidad:** Alta

**Descripción:**  
El sistema debe generar preguntas contextualizadas y relevantes según el perfil y configuración del usuario.

**Entradas:**

* Perfil del usuario (área técnica, nivel de experiencia)
* Configuración de sesión (tipo de entrevista, duración)
* Historial de respuestas en sesión actual (para adaptación dinámica)

**Proceso:**

* Sistema consulta LLM (GPT-4/Gemini) con prompt estructurado
* LLM genera pregunta apropiada al contexto
* Sistema valida relevancia y dificultad de pregunta
* Sistema verifica que pregunta no se repita en sesión actual
* Sistema registra pregunta en historial

**Salidas:**

* Pregunta formulada en lenguaje natural
* Metadatos: tipo (técnica/comportamental), nivel de dificultad, área específica
* Criterios de evaluación predefinidos

**Criterios de Aceptación:**

* CA-007.1: Las preguntas deben ser apropiadas al nivel de experiencia seleccionado
* CA-007.2: El sistema debe alternar tipos de preguntas (técnica/comportamental) en entrevistas mixtas
* CA-007.3: La dificultad debe incrementar si el usuario responde bien (adaptación dinámica)
* CA-007.4: El banco debe incluir mínimo 50 preguntas por área técnica
* CA-007.5: Las preguntas no deben repetirse en una misma sesión

**Ejemplos de Preguntas por Nivel:**

*Junior (Frontend):*

* "Explica qué es el Virtual DOM en React y por qué es importante"
* "¿Cómo manejarías la validación de un formulario de registro?"

*Semi-Senior (Backend):*

* "Diseña la arquitectura de una API REST para un sistema de reservas"
* "¿Cómo implementarías rate limiting en una API pública?"

*Senior (Full-Stack):*

* "Describe cómo escalarías horizontalmente un monolito existente"
* "Diseña un sistema de notificaciones en tiempo real con 1M de usuarios concurrentes"

**RF-008: Captura y Transcripción de Respuestas**

**Prioridad:** Alta | **Complejidad:** Media

**Descripción:**  
El sistema debe capturar el audio de las respuestas del usuario y convertirlo a texto en tiempo real.

**Entradas:**

* Audio del micrófono del usuario

**Proceso:**

* Sistema solicita permisos de micrófono al navegador
* Usuario activa grabación mediante botón "Responder"
* Sistema captura audio en formato compatible (WebM/WAV)
* Sistema envía chunks de audio a Google Cloud Speech-to-Text API
* API retorna transcripción en tiempo real
* Sistema muestra transcripción progresiva al usuario
* Usuario finaliza respuesta (botón "Terminar Respuesta" o detección de silencio prolongado)

**Salidas:**

* Transcripción completa de la respuesta
* Timestamp de inicio y fin
* Duración de la respuesta
* Indicador de confianza de transcripción

**Criterios de Aceptación:**

* CA-008.1: El sistema debe solicitar permisos de micrófono al inicio de sesión
* CA-008.2: La transcripción debe mostrarse en tiempo real con latencia < 2 segundos
* CA-008.3: El usuario debe poder editar manualmente la transcripción si es incorrecta
* CA-008.4: El sistema debe detectar silencios de >3 segundos como fin de respuesta
* CA-008.5: Debe existir botón manual "Terminar Respuesta" para control del usuario

**Manejo de Errores:**

* Permisos de micrófono denegados: mostrar instrucciones de habilitación
* API de transcripción caída: permitir respuesta escrita manual
* Ruido excesivo: notificar al usuario y sugerir reducir ruido ambiental

**RF-009: Análisis Inteligente de Respuestas**

**Prioridad:** Alta | **Complejidad:** Alta

**Descripción:**  
El sistema debe evaluar las respuestas del usuario en múltiples dimensiones utilizando IA.

**Entradas:**

* Transcripción de respuesta del usuario
* Pregunta realizada
* Criterios de evaluación predefinidos
* Contexto de perfil del usuario

**Proceso:**

* Sistema construye prompt de análisis para LLM incluyendo:
  + La pregunta realizada
  + La respuesta transcrita
  + Criterios de evaluación específicos
  + Nivel de experiencia esperado
* Sistema envía request a LLM (GPT-4/Gemini)
* LLM analiza respuesta en dimensiones clave
* Sistema parsea respuesta estructurada de LLM
* Sistema calcula puntuaciones por categoría

**Salidas:**

* Puntuación global (0-100)
* Puntuaciones por categoría:
  + **Claridad y estructura** (0-100)
  + **Precisión técnica** (0-100)
  + **Uso de metodologías** (0-100) - ej: método STAR
  + **Profesionalismo** (0-100)
  + **Completitud** (0-100)
* Palabras clave identificadas
* Fortalezas detectadas
* Áreas de mejora identificadas

**Criterios de Aceptación:**

* CA-009.1: El análisis debe completarse en menos de 5 segundos
* CA-009.2: Cada dimensión debe tener justificación específica
* CA-009.3: El sistema debe identificar al menos 3 palabras clave relevantes
* CA-009.4: Para preguntas comportamentales, debe validar uso del método STAR
* CA-009.5: Las puntuaciones deben ser consistentes entre evaluaciones similares

**Ejemplo de Análisis:**

Pregunta: *"Describe una situación donde tuviste que resolver un conflicto en tu equipo"*

Respuesta: *"En mi último proyecto, hubo desacuerdo sobre la arquitectura. Organicé una reunión donde cada parte explicó su propuesta. Evaluamos pros y contras juntos. Al final, elegimos una solución híbrida que satisfizo a todos."*

Análisis:

* Claridad: 75/100 - Respuesta estructurada pero podría ser más detallada
* Uso de STAR: 60/100 - Situación y Acción claras, falta Task y Result explícitos
* Profesionalismo: 90/100 - Tono apropiado y enfoque colaborativo
* Completitud: 65/100 - Falta mencionar resultado medible del conflicto resuelto

**RF-010: Generación de Retroalimentación Personalizada**

**Prioridad:** Alta | **Complejidad:** Alta

**Descripción:**  
El sistema debe generar feedback constructivo y accionable basado en el análisis de respuestas.

**Entradas:**

* Análisis previo de la respuesta
* Pregunta original
* Respuesta del usuario
* Nivel de experiencia del usuario

**Proceso:**

* Sistema toma resultados de análisis (RF-009)
* Sistema construye prompt para generación de feedback
* LLM genera retroalimentación estructurada
* Sistema formatea feedback en secciones legibles

**Salidas:**

* **Resumen ejecutivo**: Evaluación general en 2-3 líneas
* **Fortalezas específicas**: Lista de 2-4 puntos positivos con ejemplos
* **Áreas de mejora**: Lista de 2-4 puntos accionables
* **Sugerencias concretas**: Recomendaciones específicas de cómo mejorar
* **Ejemplo de respuesta mejorada**: Versión optimizada de la respuesta del usuario

**Criterios de Aceptación:**

* CA-010.1: El feedback debe ser constructivo y motivacional (no punitivo)
* CA-010.2: Cada sugerencia debe ser específica y accionable
* CA-010.3: El ejemplo mejorado debe mantener la esencia de la respuesta original
* CA-010.4: El tono debe adaptarse al nivel del usuario (más detallado para juniors)
* CA-010.5: El feedback debe generarse en menos de 7 segundos

**Ejemplo de Feedback:**

**RESUMEN GENERAL**

Tu respuesta demuestra habilidades de resolución de conflictos, pero podría estructurarse

mejor utilizando el método STAR para mayor impacto.

**FORTALEZAS IDENTIFICADAS**

- Enfoque colaborativo: Reconociste la importancia de escuchar todas las perspectivas

- Proactividad: Tomaste iniciativa organizando la reunión de resolución

- Solución pragmática: Buscaste un compromiso que satisficiera a ambas partes

**ÁREAS DE MEJORA**

- Falta claridad en el resultado: No mencionaste el impacto medible del conflicto resuelto

- Método STAR incompleto: La sección "Task" (tu responsabilidad específica) no está clara

- Contexto limitado: Podrías especificar qué tipo de arquitectura se discutía

**SUGERENCIAS CONCRETAS**

1. Inicia con contexto claro: "En mi rol de Tech Lead del proyecto X..."

2. Define la tarea: "Mi responsabilidad era asegurar decisión técnica en 1 semana..."

3. Cuantifica el resultado: "Esto redujo el tiempo de desarrollo en 20% y evitó 2 semanas

de refactorización futura"

**EJEMPLO DE RESPUESTA MEJORADA**

"En mi último proyecto como desarrollador senior, surgió un desacuerdo crítico sobre si usar

arquitectura microservicios o monolito modular. Mi responsabilidad era facilitar una decisión

técnica en 1 semana para no retrasar el sprint. Organicé un workshop técnico donde cada parte

presentó su propuesta con pros/contras documentados. Evaluamos criterios objetivos: escalabilidad,

complejidad operativa y time-to-market. Decidimos un monolito modular con clara separación de

bounded contexts, lo que permitió comenzar desarrollo inmediatamente y tener opción de migrar

a microservicios en el futuro. El resultado fue inicio del proyecto sin retrasos y reducción

del 25% en complejidad operativa inicial comparado con microservicios."

**Módulo 4: Dashboard y Seguimiento**

**RF-011: Dashboard de Métricas de Usuario**

**Prioridad:** Media | **Complejidad:** Media

**Descripción:**  
El sistema debe mostrar un panel visual con métricas clave del rendimiento del usuario.

**Entradas:**

* Historial de entrevistas del usuario
* Puntuaciones de cada sesión
* Datos de progreso temporal

**Proceso:**

1. Sistema consulta base de datos por entrevistas del usuario
2. Sistema calcula estadísticas agregadas
3. Sistema genera gráficos de visualización
4. Sistema identifica tendencias de mejora o estancamiento

**Salidas:**

* **Panel de Estadísticas Generales:**
  + Total de entrevistas realizadas
  + Promedio de puntuación global
  + Tiempo total practicado
  + Racha actual de práctica consecutiva
* **Gráfico de Evolución Temporal:**
  + Línea de tiempo con puntuaciones por fecha
  + Tendencia de mejora (pendiente positiva/negativa)
* **Radar Chart de Habilidades:**
  + Visualización de fortalezas por categoría (Claridad, Técnica, STAR, etc.)
* **Historial Detallado:**
  + Lista de últimas 10 entrevistas con fecha, área, puntuación
* **Recomendaciones Personalizadas:**
  + Áreas sugeridas para practicar
  + Próximos objetivos de mejora

**Criterios de Aceptación:**

* CA-011.1: El dashboard debe cargarse en menos de 2 segundos
* CA-011.2: Los gráficos deben ser interactivos (tooltips, zoom)
* CA-011.3: El usuario debe poder filtrar por rango de fechas
* CA-011.4: Las recomendaciones deben actualizarse tras cada entrevista
* CA-011.5: El dashboard debe ser responsive (mobile-friendly)

**RF-012: Historial de Entrevistas**

**Prioridad:** Media | **Complejidad:** Baja

**Descripción:**  
El sistema debe permitir al usuario revisar entrevistas pasadas en detalle.

**Entradas:**

* ID de sesión de entrevista

**Proceso:**

* Usuario selecciona entrevista del historial
* Sistema carga datos completos de la sesión
* Sistema muestra preguntas, respuestas transcritas y feedback

**Salidas:**

* Vista detallada de entrevista con:
  + Fecha y duración
  + Configuración usada (área, tipo, nivel)
  + Lista de preguntas realizadas
  + Transcripciones de respuestas
  + Puntuaciones por pregunta
  + Feedback recibido
  + Opción de exportar a PDF

**Criterios de Aceptación:**

* CA-012.1: El usuario debe poder buscar entrevistas por área técnica o fecha
* CA-012.2: El sistema debe permitir comparar dos entrevistas lado a lado
* CA-012.3: El usuario debe poder eliminar entrevistas individuales
* CA-012.4: El sistema debe mantener últimas 50 entrevistas (límite de almacenamiento)
* CA-012.5: La exportación a PDF debe incluir gráficos y feedback completo

**Módulo 5: Gamificación**

**RF-013: Sistema de Logros y Badges**

**Prioridad:** Baja | **Complejidad:** Baja

**Descripción:**  
El sistema debe otorgar insignias virtuales por cumplimiento de objetivos específicos.

**Entradas:**

* Actividad del usuario (número de entrevistas, mejoras, rachas, etc.)

**Proceso:**

* Sistema monitorea eventos disparadores de logros
* Al cumplir criterio, sistema desbloquea badge
* Sistema notifica al usuario del logro obtenido
* Sistema registra badge en perfil del usuario

**Salidas:**

* Notificación visual de badge desbloqueado
* Lista de badges en perfil de usuario
* Progreso hacia próximos badges

**Ejemplos de Badges:**

* **Primera Entrevista**: Completar primera simulación
* **Racha de Fuego**: 7 días consecutivos practicando
* **En Ascenso**: Mejorar puntuación 20+ puntos entre sesiones
* **Experto en [Área]**: Completar 10 entrevistas en un área técnica
* **Perfeccionista**: Obtener puntuación >90 en 5 entrevistas
* **Dedicación Total**: Completar 50 entrevistas en total

**Criterios de Aceptación:**

* CA-013.1: Debe haber mínimo 15 badges diferentes desbloqueables
* CA-013.2: Los badges deben tener diseño visual atractivo
* CA-013.3: El usuario debe poder ver badges bloqueados y sus requisitos
* CA-013.4: El sistema debe mostrar progreso hacia próximo badge

**RF-014: Sistema de Objetivos Semanales**

**Prioridad:** Baja | **Complejidad:** Baja

**Descripción:**  
El sistema debe permitir establecer y monitorear objetivos de práctica semanales.

**Entradas:**

* Objetivo definido por usuario o sugerido por sistema
* Actividad semanal del usuario

**Proceso:**

* Usuario establece objetivo (ej: "3 entrevistas esta semana")
* Sistema monitorea progreso automáticamente
* Sistema envía notificaciones de recordatorio (opcional)
* Al cumplir objetivo, sistema registra logro y sugiere nuevo objetivo

**Salidas:**

* Barra de progreso visual del objetivo
* Notificaciones de progreso (50%, 100%)
* Historial de objetivos cumplidos

**Criterios de Aceptación:**

* CA-014.1: El usuario debe poder modificar objetivos en cualquier momento
* CA-014.2: El sistema debe sugerir objetivos realistas basados en historial
* CA-014.3: Los objetivos no cumplidos no deben penalizar al usuario
* CA-014.4: El sistema debe reiniciar objetivos semanales cada lunes

## 3.2 Requisitos No Funcionales

**Rendimiento (Performance)**

**RNF-001: Tiempo de Respuesta de Interfaz**

**Descripción:** La interfaz de usuario debe ser altamente responsiva.

**Criterios:**

* Carga inicial de página: < 3 segundos
* Transiciones entre vistas: < 500ms
* Respuesta a interacciones del usuario: < 200ms
* Actualización de dashboard con gráficos: < 2 segundos

**Justificación:** Tiempos de respuesta lentos degradan la experiencia de usuario y pueden causar abandono de la plataforma.

**RNF-002: Latencia de APIs de IA**

**Descripción:** Las respuestas de servicios de IA deben ser razonablemente rápidas.

**Criterios:**

* Generación de pregunta: < 3 segundos
* Transcripción de voz (streaming): < 2 segundos de latencia
* Análisis de respuesta: < 5 segundos
* Generación de feedback: < 7 segundos
* Animación de avatar: inicio en < 3 segundos

**Justificación:** Latencias prolongadas rompen la fluidez de la simulación de entrevista y generan frustración.

**RNF-003: Capacidad Concurrente**

**Descripción:** El sistema debe soportar múltiples usuarios simultáneos.

**Criterios:**

* Soporte mínimo: 20 usuarios concurrentes en sesiones activas
* Degradación gradual: mantener funcionalidad básica hasta 50 usuarios
* Colas de procesamiento para picos de demanda

**Justificación:** Aunque es un proyecto académico, debe simular condiciones realistas de uso.

**Disponibilidad (Availability)**

**RNF-004: Uptime del Sistema**

**Descripción:** El sistema debe estar disponible la mayor parte del tiempo.

**Criterios:**

* Disponibilidad objetivo: 95% (excluye mantenimientos programados)
* Ventana de mantenimiento: domingos 2:00-4:00 AM CLT
* Tiempo máximo de recuperación ante fallas: 4 horas

**Justificación:** Los usuarios deben poder acceder al sistema cuando lo necesiten para preparar entrevistas reales.

**RNF-005: Tolerancia a Fallos de APIs Externas**

**Descripción:** El sistema debe manejar gracefully las caídas de APIs de terceros.

**Criterios:**

* Implementar circuit breaker para APIs externas
* Fallback: si D-ID falla, usar avatar estático con solo audio
* Fallback: si Speech-to-Text falla, permitir entrada de texto manual
* Fallback: si LLM falla, usar preguntas pre-generadas del banco
* Mensajes informativos al usuario sobre funcionalidad reducida

**Justificación:** Las APIs externas pueden tener interrupciones que no deben bloquear completamente el sistema.

**Seguridad (Security)**

**RNF-006: Autenticación y Autorización**

**Descripción:** El sistema debe proteger el acceso a datos de usuario.

**Criterios:**

* Todas las contraseñas encriptadas con bcrypt (salt rounds = 12)
* Tokens JWT con expiración de 24 horas
* Validación de tokens en cada request a endpoints protegidos
* Implementar rate limiting (max 100 requests/minuto por IP)
* Protección contra SQL injection mediante ORM (Prisma/TypeORM)

**Justificación:** Protección de datos sensibles de usuarios y prevención de accesos no autorizados.

**RNF-007: Protección de Datos Personales**

**Descripción:** Los datos de usuario deben manejarse de forma segura y ética.

**Criterios:**

* Comunicaciones siempre por HTTPS (SSL/TLS)
* Datos en reposo encriptados en base de datos
* Auditoría de accesos a datos sensibles (logs)
* Implementar políticas de retención de datos (máximo 1 año inactivo)
* Permitir eliminación completa de cuenta y datos asociados

**Justificación:** Cumplimiento con Ley 19.628 de Protección de Vida Privada en Chile y buenas prácticas de privacidad.

**RNF-008: Prevención de Abusos**

**Descripción:** El sistema debe prevenir uso indebido de recursos.

**Criterios:**

* Límite de 10 entrevistas por día por usuario
* Cooldown de 5 minutos entre entrevistas consecutivas
* Implementar CAPTCHA en registro y login tras intentos fallidos
* Monitoreo de uso anómalo de APIs (alertas de costos)

**Justificación:** Protección contra abuso de recursos computacionales y control de costos de APIs.

**Usabilidad (Usability)**

**RNF-009: Facilidad de Uso**

**Descripción:** La interfaz debe ser intuitiva y fácil de navegar.

**Criterios:**

* Flujo de usuario no debe requerir más de 3 clics para acción principal
* Tooltips y ayuda contextual en elementos clave
* Onboarding tutorial para nuevos usuarios (primera sesión)
* Diseño consistente con guías de Material Design o similar
* Mensajes de error claros y accionables

**Justificación:** Los usuarios deben poder usar el sistema sin curva de aprendizaje significativa.

**RNF-010: Accesibilidad**

**Descripción:** El sistema debe ser usable por personas con diferentes capacidades.

**Criterios:**

* Cumplimiento nivel AA de WCAG 2.1 (Web Content Accessibility Guidelines)
* Navegación completa por teclado (sin mouse)
* Contraste de colores > 4.5:1 para textos
* Alt text en todas las imágenes
* Transcripciones visibles de preguntas de voz
* Compatible con lectores de pantalla (ARIA labels)

**Justificación:** Inclusión de usuarios con discapacidades visuales, motoras o auditivas.

**RNF-011: Responsive Design**

**Descripción:** La interfaz debe adaptarse a diferentes tamaños de pantalla.

**Criterios:**

* Soporte para resoluciones:
  + Desktop: 1920x1080, 1366x768
  + Tablet: 768x1024 (iPad)
  + Mobile: 375x667 (iPhone SE) en adelante
* Diseño mobile-first
* Elementos táctiles mínimo 44x44px (touch-friendly)

**Justificación:** Aunque el foco es desktop, muchos usuarios acceden desde tablets.

**Mantenibilidad (Maintainability)**

**RNF-012: Calidad de Código**

**Descripción:** El código debe ser legible, estructurado y mantenible.

**Criterios:**

* Cobertura de pruebas unitarias > 70%
* Documentación inline para funciones complejas
* Arquitectura modular con separación de responsabilidades
* Uso de linter (ESLint) y formateador (Prettier)
* Convención de nombres consistente (camelCase para funciones, PascalCase para componentes)

**Justificación:** Facilita mantenimiento futuro y posible escalamiento del proyecto.

**RNF-013: Documentación Técnica**

**Descripción:** El sistema debe estar bien documentado.

**Criterios:**

* README completo con instrucciones de setup
* Documentación de API endpoints (Swagger/OpenAPI)
* Diagramas de arquitectura actualizados
* Guía de deployment y configuración de entorno

**Justificación:** Permite que nuevos desarrolladores entiendan el sistema rápidamente.

**Portabilidad (Portability)**

**RNF-014: Compatibilidad de Navegadores**

**Descripción:** El sistema debe funcionar en navegadores modernos principales.

**Criterios:**

* Soporte obligatorio:
  + Google Chrome 90+
  + Mozilla Firefox 88+
  + Safari 14+
  + Microsoft Edge 90+
* NO soporte: Internet Explorer (descontinuado)
* Testing cross-browser obligatorio antes de releases

**Justificación:** Los usuarios usan variedad de navegadores y el sistema debe funcionar consistentemente.

**RNF-015: Independencia de Plataforma**

**Descripción:** El sistema backend debe ser independiente del sistema operativo.

**Criterios:**

* Backend deployable en Linux, Windows, macOS
* Uso de Docker para containerización
* Variables de entorno para configuraciones específicas de plataforma

**Justificación:** Facilita deployment en diferentes proveedores cloud y ambientes de desarrollo.

**Escalabilidad (Scalability)**

**RNF-016: Escalabilidad Horizontal**

**Descripción:** El sistema debe poder escalar agregando más instancias.

**Criterios:**

* Backend stateless (sin sesiones en memoria del servidor)
* Uso de JWT para autenticación (no sesiones server-side)
* Base de datos preparada para replicación (PostgreSQL con read replicas)
* Cache distribuido (Redis) para datos frecuentes

**Justificación:** Permite crecimiento futuro del número de usuarios.

**RNF-017: Optimización de Costos de IA**

**Descripción:** El uso de APIs de IA debe ser eficiente para controlar costos.

**Criterios:**

* Implementar caché de preguntas frecuentes (evitar regeneración)
* Limitar longitud máxima de prompts (tokens)
* Usar modelos más económicos cuando sea suficiente (ej: GPT-3.5 vs GPT-4)
* Monitoreo de costos en tiempo real con alertas

**Justificación:** Las APIs de IA tienen costos por uso que pueden escalar rápidamente.

**Fiabilidad (Reliability)**

**RNF-018: Integridad de Datos**

**Descripción:** Los datos deben mantenerse consistentes y correctos.

**Criterios:**

* Transacciones ACID en base de datos
* Validación de datos en backend (no solo frontend)
* Backups automáticos diarios de base de datos
* Logs de auditoría para cambios críticos

**Justificación:** Prevención de pérdida o corrupción de datos de usuario.

**RNF-019: Recuperación ante Errores**

**Descripción:** El sistema debe manejar errores de forma elegante.

**Criterios:**

* Try-catch en todas las operaciones críticas
* Logging estructurado de errores (con contexto)
* Mensajes de error informativos al usuario (no stack traces)
* Reintentos automáticos en llamadas a APIs (con exponential backoff)

**Justificación:** Los errores son inevitables, pero no deben romper completamente el sistema.

## 3.3 Requisitos de Interfaces

**Interfaz de Usuario**

**RIU-001: Diseño Visual**

**Descripción:** La interfaz debe seguir principios de diseño moderno y profesional.

**Especificaciones:**

* **Paleta de colores:**
  + Primario: #3B82F6 (azul corporativo)
  + Secundario: #10B981 (verde éxito)
  + Acento: #F59E0B (naranja llamadas a acción)
  + Neutrales: #1F2937, #6B7280, #F3F4F6 (texto y fondos)
* **Tipografía:**
  + Títulos: Inter Bold, 24-32px
  + Cuerpo: Inter Regular, 14-16px
  + Código: Fira Code, 14px (para snippets técnicos)
* **Espaciado:**
  + Grid de 8px para consistencia
  + Márgenes estándar: 16px, 24px, 32px
* **Componentes:**
  + Botones con estados (hover, active, disabled)
  + Cards con sombras sutiles (elevation)
  + Formularios con validación visual

**RIU-002: Navegación**

**Descripción:** El sistema debe tener navegación clara y consistente.

**Estructura:**

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Elementos de navegación:**

* Navbar superior fijo con logo y menú principal
* Sidebar colapsable con sub-secciones (en dashboard)
* Breadcrumbs para navegación profunda
* Botón "Volver" en todas las sub-páginas

**RIU-003: Feedback Visual**

**Descripción:** El sistema debe proporcionar retroalimentación visual clara de las acciones del usuario.

**Especificaciones:**

* **Loaders:** Spinners animados durante carga de datos
* **Skeleton screens:** Placeholders durante carga inicial de componentes
* **Toast notifications:** Mensajes temporales de éxito/error (5 segundos)
* **Confirmaciones:** Modals para acciones destructivas (eliminar cuenta, borrar entrevista)
* **Estados de carga:** Botones deshabilitados con spinner durante peticiones async

**Interfaz de Hardware**

**RIH-001: Micrófono**

**Descripción:** Requisitos de captura de audio.

**Especificaciones:**

* Frecuencia de muestreo: 16 kHz (mínimo para STT)
* Formato: mono (un canal)
* Calidad: 16-bit PCM
* Soporte para micrófonos integrados y externos
* Permiso explícito del navegador requerido

**RIH-002: Audio de Salida**

**Descripción:** Requisitos de reproducción de audio.

**Especificaciones:**

* Formatos soportados: MP3, WAV, OGG
* Volumen ajustable desde interfaz
* Soporte para auriculares y altavoces
* Sincronización con animación de avatar (<50ms de desfase)

**Interfaz de Software**

**RIS-001: API Backend**

**Descripción:** Especificación de endpoints RESTful.

**Estructura base:**

Base URL: https://api.interviewai.cl/v1

Authentication: Bearer JWT token en header

Content-Type: application/json

**Endpoints principales:**

**Autenticación:**

POST /auth/register - Registro de usuario

POST /auth/login - Inicio de sesión

POST /auth/logout - Cierre de sesión

POST /auth/refresh-token - Renovar token JWT

POST /auth/forgot-password - Solicitar reset de contraseña

POST /auth/reset-password - Confirmar nueva contraseña

**Usuarios:**

GET /users/me - Obtener perfil propio

PATCH /users/me - Actualizar perfil

DELETE /users/me - Eliminar cuenta

GET /users/me/stats - Obtener estadísticas de usuario

**Entrevistas:**

POST /interviews - Crear nueva sesión de entrevista

GET /interviews - Listar entrevistas del usuario

GET /interviews/:id - Obtener detalles de entrevista específica

DELETE /interviews/:id - Eliminar entrevista

POST /interviews/:id/start - Iniciar entrevista

POST /interviews/:id/finish - Finalizar entrevista

**Preguntas:**

POST /interviews/:id/questions - Generar nueva pregunta

GET /interviews/:id/questions - Listar preguntas de sesión

GET /interviews/:id/questions/:qid - Obtener pregunta específica

**Respuestas:**

POST /interviews/:id/questions/:qid/answers - Enviar respuesta

GET /interviews/:id/questions/:qid/answers - Obtener respuesta

POST /interviews/:id/questions/:qid/analyze - Analizar respuesta

GET /interviews/:id/questions/:qid/feedback - Obtener feedback

**Transcripción:**

POST /speech/transcribe - Transcribir audio a texto (streaming)

**Avatar:**

POST /avatar/generate - Generar video de avatar con texto

**Ejemplo de Request/Response:**

json

*// POST /interviews*

{

"area": "Frontend",

"level": "Senior",

"type": "Mixta",

"duration": 30

}

*// Response 201 Created*

{

"id": "uuid-1234-5678",

"userId": "user-uuid",

"area": "Frontend",

"level": "Senior",

"type": "Mixta",

"duration": 30,

"status": "created",

"createdAt": "2025-10-16T14:30:00Z"

}

**RIS-002: WebSocket para Tiempo Real**

**Descripción:** Comunicación bidireccional para transcripción en vivo.

**Especificación:**

WebSocket URL: wss://api.interviewai.cl/ws

Event: audio-chunk

Payload: { sessionId, audioData: base64 }

Event: transcription-update

Payload: { text, confidence, isFinal }

**RIS-003: Integración con APIs Externas**

**Descripción:** Interfaces con servicios de terceros.

**OpenAI GPT-4 / Gemini:**

javascript

POST https://api.openai.com/v1/chat/completions

Headers:

Authorization: Bearer sk-...

Content-Type: application/json

Body:

{

"model": "gpt-4",

"messages": [

{"role": "system", "content": "Eres un entrevistador técnico..."},

{"role": "user", "content": "Genera una pregunta de React nivel Senior"}

],

"temperature": 0.7,

"max\_tokens": 500

}

**Google Cloud Speech-to-Text:**

javascript

POST https://speech.googleapis.com/v1/speech:recognize

Headers:

Authorization: Bearer ya29...

Body:

{

"config": {

"encoding": "LINEAR16",

"sampleRateHertz": 16000,

"languageCode": "es-CL"

},

"audio": {

"content": "base64-encoded-audio"

}

}

**D-ID Avatar API:**

javascript

POST https://api.d-id.com/talks

Headers:

Authorization: Basic ...

Body:

{

"source\_url": "https://avatar-image.jpg",

"script": {

"type": "text",

"input": "Hola, soy tu entrevistador virtual..."

},

"config": {

"fluent": true,

"pad\_audio": 0

}

}

# 4. Casos de Uso

**Diagrama de Casos de Uso General**

**Diagrama

El contenido generado por IA puede ser incorrecto.**

**CU-001: Registrar Nueva Cuenta**

**Actor Principal:** Usuario nuevo (candidato)

**Precondiciones:**

* El usuario no tiene una cuenta existente
* El usuario tiene acceso a un navegador web

**Flujo Principal:**

1. Usuario accede a la página de registro
2. Sistema muestra formulario de registro
3. Usuario completa campos obligatorios:
   * Nombre completo
   * Correo electrónico
   * Contraseña
   * Área técnica de interés
   * Nivel de experiencia
4. Usuario hace clic en "Crear Cuenta"
5. Sistema valida datos ingresados
6. Sistema encripta contraseña
7. Sistema crea perfil de usuario en base de datos
8. Sistema redirige a página de inicio de sesión
9. Sistema muestra mensaje de confirmación

**Flujos Alternativos:**

* **3a. Correo electrónico ya registrado:**
  + Sistema muestra mensaje "El correo ya está registrado"
  + Sistema ofrece opción "¿Olvidaste tu contraseña?"
  + Caso de uso termina
* **3b. Contraseña no cumple requisitos:**
  + Sistema muestra mensaje específico de error de contraseña
  + Usuario modifica contraseña
  + Retorna al paso 4
* **3c. Formato de correo inválido:**
  + Sistema muestra mensaje "Ingrese un correo válido"
  + Usuario corrige correo
  + Retorna al paso 4

**Postcondiciones:**

* Nueva cuenta de usuario creada en el sistema
* Usuario puede iniciar sesión con credenciales creadas

**Requisitos Especiales:**

* La contraseña debe almacenarse encriptada (bcrypt)
* El formulario debe validar datos en tiempo real (UX)

**CU-002: Realizar Entrevista Completa**

**Actor Principal:** Usuario registrado (candidato)

**Precondiciones:**

* Usuario autenticado en el sistema
* Usuario tiene permisos de micrófono habilitados

**Flujo Principal:**

1. Usuario hace clic en "Nueva Entrevista" desde dashboard
2. Sistema muestra formulario de configuración (incluye CU-003)
3. Usuario selecciona:
   * Área técnica
   * Nivel de experiencia
   * Tipo de entrevista
   * Duración
4. Usuario hace clic en "Iniciar Entrevista"
5. Sistema crea sesión de entrevista
6. Sistema carga interfaz de simulación con avatar
7. **LOOP** [Hasta completar tiempo o número de preguntas]:
   * 7a. Sistema genera pregunta contextualizada (CU-004)
   * 7b. Avatar presenta pregunta verbalmente
   * 7c. Usuario responde verbalmente (CU-005)
   * 7d. Sistema transcribe respuesta en tiempo real
   * 7e. Usuario finaliza respuesta (botón o detección de silencio)
   * 7f. Sistema analiza respuesta (CU-006)
   * 7g. Sistema genera feedback (CU-007)
   * 7h. Sistema muestra feedback al usuario
   * 7i. Usuario hace clic en "Siguiente Pregunta"
8. Sistema finaliza sesión de entrevista
9. Sistema muestra resumen de rendimiento
10. Sistema guarda entrevista en historial
11. Sistema redirige a dashboard actualizado

**Flujos Alternativos:**

* **2a. Usuario cancela configuración:**
  + Sistema descarta sesión no guardada
  + Retorna a dashboard
  + Caso de uso termina
* **7b. Error de API de avatar:**
  + Sistema muestra pregunta en formato texto
  + Sistema reproduce audio sin video
  + Continúa en paso 7c
* **7c. Permisos de micrófono denegados:**
  + Sistema muestra instrucciones para habilitar micrófono
  + Sistema ofrece opción de respuesta escrita manual
  + Continúa en paso 7e
* **7d. Error de transcripción:**
  + Sistema permite edición manual de transcripción
  + Usuario corrige texto
  + Continúa en paso 7e
* **8a. Usuario abandona entrevista antes de finalizar:**
  + Sistema muestra modal de confirmación "¿Seguro que deseas salir?"
  + Si confirma: Sistema guarda progreso parcial
  + Si cancela: Retorna a paso 7

**Postcondiciones:**

* Entrevista completa guardada en historial del usuario
* Métricas de rendimiento actualizadas en dashboard
* Feedback disponible para revisión posterior

**Requisitos Especiales:**

* Latencia total por pregunta (generación + transcripción + análisis) < 15 segundos
* El usuario debe poder pausar la entrevista en cualquier momento

**CU-003: Configurar Nueva Entrevista**

**Actor Principal:** Usuario registrado

**Precondiciones:**

* Usuario autenticado
* Usuario accede desde dashboard o botón "Nueva Entrevista"

**Flujo Principal:**

1. Sistema muestra formulario de configuración
2. Sistema pre-rellena campos con valores por defecto basados en historial
3. Usuario selecciona o confirma:
   * Área técnica (dropdown con opciones)
   * Nivel de experiencia (radio buttons)
   * Tipo de entrevista (checkboxes)
   * Duración (slider 15-45 minutos)
   * [Opcional] Empresa objetivo (text input)
4. Sistema valida selecciones en tiempo real
5. Usuario hace clic en "Confirmar Configuración"
6. Sistema genera ID único de sesión
7. Sistema pre-carga banco de preguntas relevantes
8. Sistema retorna control a caso de uso padre (CU-002)

**Flujos Alternativos:**

* **3a. Usuario no modifica valores sugeridos:**
  + Sistema usa configuración por defecto
  + Continúa en paso 5
* **4a. Combinación de parámetros inválida:**
  + Sistema muestra mensaje "Esta configuración no está disponible"
  + Usuario ajusta selecciones
  + Retorna al paso 3

**Postcondiciones:**

* Sesión de entrevista creada con configuración definida
* Banco de preguntas relevantes cargado en memoria

**Requisitos Especiales:**

* El formulario debe recordar última configuración usada
* Debe existir botón "Configuración Rápida" con preset común

**CU-004: Generar Pregunta Contextualizada**

**Actor Principal:** Sistema (automatizado)

**Precondiciones:**

* Sesión de entrevista activa
* Configuración de entrevista definida

**Flujo Principal:**

1. Sistema consulta contexto de sesión:
   * Área técnica seleccionada
   * Nivel de experiencia
   * Tipo de pregunta siguiente (técnica/comportamental)
   * Preguntas ya realizadas en sesión actual
   * Rendimiento en preguntas previas (para adaptación)
2. Sistema construye prompt estructurado para LLM:

Rol: Entrevistador técnico senior

Contexto: Entrevista de [área] para nivel [experiencia]

Tarea: Generar pregunta [tipo] de dificultad [nivel]

Restricciones: No repetir temas [lista]

1. Sistema envía request a API de LLM (GPT-4/Gemini)
2. Sistema recibe pregunta generada
3. Sistema valida:
   * Longitud apropiada (50-200 caracteres)
   * No repetición con preguntas previas
   * Relevancia al área técnica
4. Sistema registra pregunta en base de datos con metadata:
   * Timestamp
   * Tipo
   * Dificultad estimada
   * Categoría técnica
5. Sistema retorna pregunta a módulo de presentación

**Flujos Alternativos:**

* **3a. API de LLM no disponible:**
  + Sistema consulta banco de preguntas pre-generadas
  + Sistema selecciona pregunta aleatoria filtrada por contexto
  + Continúa en paso 6
* **5a. Pregunta generada es inválida:**
  + Sistema descarta pregunta
  + Sistema realiza segundo intento con prompt ajustado
  + Si falla nuevamente, ejecuta flujo 3a
* **5b. Pregunta se repite:**
  + Sistema descarta pregunta
  + Sistema regenera con restricción explícita de no repetición
  + Continúa en paso 4

**Postcondiciones:**

* Pregunta válida y contextualizada disponible
* Pregunta registrada en historial de sesión

**Requisitos Especiales:**

* Tiempo máximo de generación: 3 segundos
* Implementar caché de preguntas frecuentes

**CU-005: Capturar y Transcribir Respuesta**

**Actor Principal:** Usuario registrado

**Precondiciones:**

* Pregunta ha sido presentada
* Permisos de micrófono habilitados

**Flujo Principal:**

1. Sistema muestra pregunta en pantalla
2. Avatar presenta pregunta verbalmente
3. Sistema muestra botón "Iniciar Respuesta" habilitado
4. Usuario hace clic en "Iniciar Respuesta"
5. Sistema activa captura de audio del micrófono
6. Sistema muestra indicador visual de grabación activa
7. **LOOP** [Mientras usuario responde]:
   * 7a. Sistema captura chunks de audio
   * 7b. Sistema envía audio a Google Speech-to-Text API (streaming)
   * 7c. Sistema recibe transcripción parcial
   * 7d. Sistema muestra transcripción en tiempo real en pantalla
8. Usuario hace clic en "Terminar Respuesta" O sistema detecta silencio >3 segundos
9. Sistema detiene captura de audio
10. Sistema espera transcripción final de API
11. Sistema muestra transcripción completa
12. Sistema permite edición manual de transcripción
13. Usuario confirma transcripción o edita errores
14. Usuario hace clic en "Confirmar Respuesta"
15. Sistema registra respuesta en base de datos:
    * Transcripción final
    * Timestamp inicio/fin
    * Duración
    * ID de pregunta asociada
16. Sistema retorna control a CU-002

**Flujos Alternativos:**

* **5a. Error al acceder al micrófono:**
  + Sistema muestra mensaje "No se puede acceder al micrófono"
  + Sistema ofrece opción "Responder por escrito"
  + Usuario ingresa respuesta manualmente
  + Continúa en paso 14
* **7b. API de transcripción no disponible:**
  + Sistema muestra mensaje "Transcripción no disponible"
  + Sistema almacena audio temporalmente
  + Sistema ofrece opción "Escribir respuesta"
  + Usuario ingresa respuesta manualmente
  + Continúa en paso 14
* **8a. Usuario hace pausa prolongada pero continúa:**
  + Sistema muestra notificación "¿Has terminado?"
  + Usuario puede continuar hablando o finalizar
* **12a. Transcripción tiene errores significativos:**
  + Usuario edita manualmente transcripción
  + Sistema resalta texto editado (para auditoría)
  + Continúa en paso 14

**Postcondiciones:**

* Respuesta transcrita y confirmada guardada
* Duración de respuesta registrada
* Sistema listo para análisis de respuesta (CU-006)

**Requisitos Especiales:**

* Latencia de transcripción < 2 segundos
* Feedback visual claro de estado de grabación

**CU-006: Analizar Respuesta del Usuario**

**Actor Principal:** Sistema (automatizado)

**Precondiciones:**

* Respuesta del usuario transcrita y confirmada
* Pregunta original disponible en contexto

**Flujo Principal:**

1. Sistema recupera datos de contexto:
   * Pregunta original
   * Transcripción de respuesta
   * Nivel de experiencia del usuario
   * Tipo de pregunta (técnica/comportamental)
2. Sistema construye prompt de análisis para LLM:

Rol: Evaluador experto en entrevistas técnicas

Contexto: Pregunta [tipo] de [área] nivel [experiencia]

Pregunta: [texto pregunta]

Respuesta candidato: [transcripción]

Tarea: Analizar en 5 dimensiones con puntuación 0-100

- Claridad y estructura

- Precisión técnica

- Uso de metodologías (STAR si aplica)

- Profesionalismo

- Completitud

Formato: JSON estructurado

1. Sistema envía request a API de LLM
2. Sistema recibe análisis estructurado en JSON:

json

{

"scores": {

"clarity": 75,

"technical\_accuracy": 80,

"methodology": 60,

"professionalism": 90,

"completeness": 70

},

"overall\_score": 75,

"keywords\_identified": ["react", "hooks", "useState"],

"strengths": ["Explicación clara del concepto"],

"areas\_for\_improvement": ["Falta ejemplo práctico"]

}

1. Sistema valida respuesta del LLM:
   * Todas las puntuaciones están entre 0-100
   * Al menos 2 fortalezas y 2 áreas de mejora identificadas
2. Sistema calcula métricas agregadas:
   * Promedio ponderado de scores
   * Comparación con benchmark de nivel
3. Sistema registra análisis en base de datos
4. Sistema retorna análisis a CU-002 para generación de feedback

**Flujos Alternativos:**

* **3a. API de LLM no disponible:**
  + Sistema aplica análisis basado en reglas simples:
    - Longitud de respuesta
    - Presencia de palabras clave técnicas
    - Estructura básica (inicio, desarrollo, cierre)
  + Sistema genera scores aproximados
  + Continúa en paso 7
* **5a. Respuesta del LLM es inválida:**
  + Sistema registra error en logs
  + Sistema realiza segundo intento con prompt simplificado
  + Si falla nuevamente, ejecuta flujo 3a

**Postcondiciones:**

* Análisis completo de respuesta guardado
* Puntuaciones por dimensión calculadas
* Datos listos para generación de feedback (CU-007)

**Requisitos Especiales:**

* Tiempo máximo de análisis: 5 segundos
* Análisis debe ser determinístico (misma respuesta = mismo análisis)

**CU-007: Generar Feedback Personalizado**

**Actor Principal:** Sistema (automatizado)

**Precondiciones:**

* Análisis de respuesta completado (CU-006)
* Datos de contexto disponibles

**Flujo Principal:**

1. Sistema recupera análisis previo de CU-006
2. Sistema construye prompt para generación de feedback:

Rol: Coach de entrevistas profesional

Contexto: Respuesta de nivel [experiencia] en pregunta [tipo]

Análisis: [scores y observaciones de CU-006]

Tarea: Generar feedback constructivo con:

- Resumen ejecutivo (2-3 líneas)

- 2-4 fortalezas específicas con ejemplos

- 2-4 áreas de mejora accionables

- Sugerencias concretas de optimización

- Ejemplo de respuesta mejorada manteniendo esencia original

Tono: Motivacional pero honesto

1. Sistema envía request a API de LLM
2. Sistema recibe feedback estructurado
3. Sistema formatea feedback para presentación visual:
   * Secciones claramente delimitadas
   * Íconos y códigos de color por sección
   * Highlight de puntos clave
4. Sistema registra feedback en base de datos
5. Sistema retorna feedback formateado a CU-002

**Flujos Alternativos:**

* **3a. API de LLM no disponible:**
  + Sistema genera feedback genérico basado en scores:
    - Identificar score más alto → mencionar como fortaleza
    - Identificar score más bajo → mencionar como área de mejora
    - Usar plantillas pre-definidas de sugerencias
  + Continúa en paso 6
* **4a. Feedback generado es demasiado genérico:**
  + Sistema detecta frases genéricas ("podrías mejorar", "está bien")
  + Sistema regenera con prompt más específico exigiendo ejemplos concretos
  + Continúa en paso 5

**Postcondiciones:**

* Feedback completo y personalizado generado
* Feedback guardado y asociado a respuesta
* Feedback listo para presentación al usuario

**Requisitos Especiales:**

* Tiempo máximo de generación: 7 segundos
* Feedback debe ser específico (no genérico)
* Tono siempre constructivo y motivacional

**CU-008: Revisar Historial de Entrevistas**

**Actor Principal:** Usuario registrado

**Precondiciones:**

* Usuario autenticado
* Usuario ha realizado al menos 1 entrevista

**Flujo Principal:**

1. Usuario accede a sección "Historial" desde menú principal
2. Sistema consulta base de datos por entrevistas del usuario
3. Sistema muestra lista de entrevistas ordenadas por fecha (más reciente primero)
4. Para cada entrevista, sistema muestra:
   * Fecha y hora
   * Área técnica
   * Tipo de entrevista
   * Duración
   * Puntuación global
   * Estado (completada/incompleta)
5. Usuario selecciona una entrevista específica
6. Sistema carga vista detallada de entrevista con:
   * Configuración utilizada
   * Lista de preguntas realizadas
   * Transcripciones de respuestas
   * Análisis y puntuaciones por pregunta
   * Feedback completo por pregunta
   * Gráfico de puntuaciones
7. Usuario puede:
   * Exportar entrevista a PDF
   * Comparar con otra entrevista
   * Eliminar entrevista
   * Volver a lista de historial

**Flujos Alternativos:**

* **2a. Usuario no tiene entrevistas:**
  + Sistema muestra mensaje "Aún no has realizado entrevistas"
  + Sistema muestra botón "Realizar Primera Entrevista"
  + Caso de uso termina
* **5a. Usuario aplica filtros:**
  + Usuario selecciona filtros (área técnica, rango de fechas, tipo)
  + Sistema actualiza lista con entrevistas filtradas
  + Continúa en paso 5
* **7a. Usuario solicita exportar a PDF:**
  + Sistema genera documento PDF con todos los datos de entrevista
  + Sistema incluye gráficos y feedback formateado
  + Sistema descarga PDF en navegador
  + Retorna a paso 6
* **7b. Usuario solicita eliminar entrevista:**
  + Sistema muestra modal de confirmación
  + Usuario confirma eliminación
  + Sistema marca entrevista como eliminada (soft delete)
  + Sistema actualiza vista de lista
  + Retorna a paso 3
* **7c. Usuario solicita comparar entrevistas:**
  + Sistema muestra selector de segunda entrevista
  + Usuario selecciona entrevista a comparar
  + Sistema muestra vista comparativa lado a lado
  + Retorna a paso 6

**Postcondiciones:**

* Usuario ha revisado información histórica
* Posiblemente ha exportado entrevista
* Historial actualizado si eliminó entrevistas

**Requisitos Especiales:**

* Lista debe cargarse en <2 segundos
* Exportación a PDF debe mantener formato visual atractivo

**CU-009: Visualizar Dashboard de Progreso**

**Actor Principal:** Usuario registrado

**Precondiciones:**

* Usuario autenticado
* Usuario ha realizado al menos 3 entrevistas (para métricas significativas)

**Flujo Principal:**

1. Usuario accede a Dashboard desde menú principal
2. Sistema consulta datos históricos del usuario
3. Sistema calcula métricas agregadas:
   * Total de entrevistas realizadas
   * Promedio de puntuación global
   * Tiempo total practicado
   * Racha actual de práctica
   * Tendencia de mejora (pendiente de regresión lineal)
4. Sistema genera visualizaciones:
   * **Gráfico de líneas**: Evolución de puntuación en el tiempo
   * **Radar chart**: Fortalezas por categoría (Claridad, Técnica, STAR, etc.)
   * **Barra de progreso**: Hacia objetivo semanal
   * **Mapa de calor**: Días de actividad en el mes
5. Sistema identifica insights automáticos:
   * Área técnica con mejor rendimiento
   * Categoría que más ha mejorado
   * Horario del día con mejores resultados
6. Sistema muestra recomendaciones personalizadas:
   * "Practica más [área técnica] donde tienes margen de mejora"
   * "Tu mejor horario es [hora], considera practicar entonces"
7. Sistema muestra sección de badges y logros
8. Usuario interactúa con gráficos (hover, zoom, filtros de fecha)

**Flujos Alternativos:**

* **2a. Usuario tiene menos de 3 entrevistas:**
  + Sistema muestra mensaje "Realiza más entrevistas para ver tendencias"
  + Sistema muestra solo métricas básicas disponibles
  + Caso de uso termina parcialmente
* **4a. Usuario aplica filtros de fecha:**
  + Usuario selecciona rango personalizado (última semana, mes, 3 meses, año)
  + Sistema recalcula métricas para período seleccionado
  + Sistema actualiza todos los gráficos
  + Continúa en paso 5

**Postcondiciones:**

* Usuario ha visualizado su progreso
* Usuario tiene insights de sus fortalezas y áreas de mejora
* Motivación incrementada por visualización de progreso

**Requisitos Especiales:**

* Dashboard debe cargarse en <2 segundos
* Gráficos deben ser interactivos y responsive

# Requisitos de Rendimiento

**Rendimiento de Procesamiento**

**RP-001: Tiempo de Generación de Pregunta**

* Generación de pregunta mediante LLM: máximo 3 segundos
* Fallback a banco pre-generado: máximo 500ms
* Justificación: Mantener fluidez de simulación

**RP-002: Latencia de Transcripción**

* Transcripción en streaming: latencia <2 segundos
* Actualización de texto en UI: cada 500ms
* Justificación: Feedback en tiempo real al usuario

**RP-003: Análisis de Respuesta**

* Análisis completo mediante LLM: máximo 5 segundos
* Análisis por reglas simples (fallback): máximo 1 segundo
* Justificación: Balance entre profundidad de análisis y experiencia de usuario

**RP-004: Generación de Feedback**

* Feedback personalizado: máximo 7 segundos
* Feedback basado en plantillas: máximo 2 segundos
* Justificación: Tiempo total de ciclo pregunta-feedback <20 segundos

**RP-005: Carga de Dashboard**

* Tiempo de carga inicial: <2 segundos
* Renderizado de gráficos: <1 segundo
* Actualización de métricas: <500ms
* Justificación: Experiencia fluida al navegar

**Rendimiento de Base de Datos**

**RP-006: Consultas Simples**

* SELECT básico (perfil de usuario): <100ms
* INSERT de respuesta: <200ms
* UPDATE de perfil: <150ms
* Justificación: Operaciones frecuentes deben ser instantáneas

**RP-007: Consultas Complejas**

* Historial de entrevistas con JOIN: <500ms
* Cálculo de métricas agregadas: <1 segundo
* Generación de reporte completo: <3 segundos
* Justificación: Consultas complejas son menos frecuentes, toleran mayor latencia

**Capacidad y Escalabilidad**

**RP-008: Usuarios Concurrentes**

* Capacidad mínima: 20 usuarios simultáneos en sesiones activas
* Degradación gradual: hasta 50 usuarios (funcionalidad básica mantenida)
* Justificación: Proyecto académico con crecimiento proyectado moderado

**RP-009: Almacenamiento**

* Promedio por usuario: 50 MB (transcripciones + metadata)
* Capacidad inicial: 100 usuarios = 5 GB
* Plan de crecimiento: escalamiento a 10 GB en 6 meses
* Justificación: Audio no se almacena, solo transcripciones reducen tamaño

**RP-010: Throughput de APIs Externas**

* Llamadas a LLM: máximo 100 requests/hora (según cuota gratuita)
* Transcripciones STT: máximo 60 minutos/día (según cuota gratuita)
* Avatar API: máximo 50 videos/día (según cuota gratuita)
* Justificación: Restricciones de planes gratuitos/estudiantiles

**Optimización de Recursos**

**RP-011: Uso de Memoria**

* Backend: máximo 512 MB RAM por instancia
* Frontend: máximo 100 MB de assets iniciales
* Justificación: Compatibilidad con hosting gratuito (Vercel, Railway)

**RP-012: Ancho de Banda**

* Streaming de audio: ~128 kbps (16 kHz mono)
* Video de avatar: máximo 2 MB por pregunta (compresión H.264)
* Assets estáticos: CDN con cache de 7 días
* Justificación: Minimizar costos de transferencia

**RP-013: Optimización de Tokens de LLM**

* Prompt máximo: 1000 tokens
* Respuesta esperada: 500 tokens
* Costo por interacción: <$0.02 USD
* Justificación: Control de costos operacionales

### 6. Restricciones de Diseño

**Restricciones de Arquitectura**

**RD-001: Arquitectura Cliente-Servidor** **Descripción:** El sistema debe implementarse como aplicación web con separación frontend-backend.

**Especificación:**

* **Frontend:** Single Page Application (SPA) en React.js o Vue.js
* **Backend:** API RESTful en Node.js con Express/NestJS
* **Base de Datos:** PostgreSQL (relacional)
* **Comunicación:** HTTP/HTTPS + WebSockets para streaming

**Justificación:**

* Separación de responsabilidades
* Facilita mantenimiento y testing
* Permite escalamiento independiente de capas

**RD-002: Patrón MVC/Arquitectura en Capas** **Descripción:** El backend debe seguir arquitectura en capas claramente definidas.

**Estructura obligatoria:**

backend/

├── src/

│ ├── controllers/ # Manejo de requests HTTP

│ ├── services/ # Lógica de negocio

│ ├── models/ # Modelos de datos (ORM)

│ ├── middlewares/ # Autenticación, validación

│ ├── utils/ # Funciones auxiliares

│ └── config/ # Configuraciones

**Justificación:**

* Código organizado y mantenible
* Facilita testing unitario por capa
* Estándar de la industria

**RD-003: Comunicación Stateless** **Descripción:** El backend debe ser stateless (sin estado en servidor).

**Implementación:**

* Sin sesiones en memoria del servidor
* Autenticación mediante JWT en cada request
* Estado del cliente manejado en frontend

**Justificación:**

* Facilita escalamiento horizontal
* Permite múltiples instancias sin sincronización compleja
* Reduce acoplamiento

**Restricciones Tecnológicas**

**RD-004: Stack Tecnológico Obligatorio** **Descripción:** Tecnologías definidas por requisitos del proyecto y experiencia del equipo.

**Tecnologías Obligatorias:**

**Frontend:**

* Framework: React.js 18+ o Vue.js 3+
* Estado: Redux/Zustand (React) o Pinia (Vue)
* Estilos: Tailwind CSS o Material-UI
* Gráficos: Chart.js o Recharts
* HTTP Client: Axios

**Backend:**

* Runtime: Node.js 18+ LTS
* Framework: Express.js 4+ o NestJS 10+
* ORM: Prisma o TypeORM
* Autenticación: JWT (jsonwebtoken)
* Validación: Joi o class-validator

**Base de Datos:**

* DBMS: PostgreSQL 14+
* Migraciones: Prisma Migrate o TypeORM migrations

**APIs Externas:**

* LLM: OpenAI GPT-4 o Google Gemini
* STT: Google Cloud Speech-to-Text
* Avatar: D-ID API
* (Alternativas permitidas si servicios principales fallan)

**Justificación:**

* Consistencia en el desarrollo
* Experiencia previa del equipo
* Amplia documentación y comunidad

**RD-005: Versionamiento con Git** **Descripción:** Control de versiones obligatorio con flujo de trabajo definido.

**Especificación:**

* Repositorio: GitHub
* Branching strategy: Git Flow
  + main: producción
  + develop: integración
  + feature/\*: nuevas funcionalidades
  + hotfix/\*: correcciones urgentes
* Commits: Conventional Commits (feat, fix, docs, etc.)
* Pull Requests obligatorios con revisión de código

**Justificación:**

* Trazabilidad de cambios
* Trabajo colaborativo sin conflictos
* Recuperación ante errores

**RD-006: Containerización con Docker** **Descripción:** El sistema debe ser containerizable para deployment consistente.

**Especificación:**

* Dockerfile para backend
* Dockerfile para frontend (build estático)
* docker-compose.yml para ambiente de desarrollo
* Imagen base: node:18-alpine (ligera)

**Justificación:**

* Portabilidad entre ambientes (dev, staging, prod)
* Facilita CI/CD
* Aislamiento de dependencias

**Restricciones de Interfaz de Usuario**

**RD-007: Diseño Mobile-First** **Descripción:** Aunque el foco es desktop, la interfaz debe diseñarse primero para mobile y escalar hacia arriba.

**Breakpoints obligatorios:**

* Mobile: 320px - 767px
* Tablet: 768px - 1023px
* Desktop: 1024px+

**Justificación:**

* Garantiza usabilidad en todos los dispositivos
* Mejora accesibilidad

**RD-008: Librería de Componentes** **Descripción:** Uso de sistema de diseño establecido para consistencia visual.

**Opciones permitidas:**

* Material-UI (MUI) para React
* Ant Design para React
* Vuetify para Vue.js
* Tailwind CSS + Headless UI

**Justificación:**

* Aceleración del desarrollo
* Consistencia visual automática
* Accesibilidad built-in

**RD-009: Internacionalización (i18n) Preparada** **Descripción:** Aunque fase 1 es solo español, el código debe estar preparado para futura internacionalización.

**Implementación:**

* Textos en archivos de traducción (no hardcoded)
* Uso de react-i18next o vue-i18n
* Claves descriptivas: dashboard.welcome\_message en lugar de textos directos

**Justificación:**

* Facilita expansión futura a inglés/portugués
* Buena práctica de desarrollo

**Restricciones de Seguridad**

**RD-010: HTTPS Obligatorio** **Descripción:** Toda comunicación debe ser encriptada.

**Especificación:**

* Certificados SSL/TLS (Let's Encrypt gratuito)
* Redirección automática HTTP → HTTPS
* HSTS (HTTP Strict Transport Security) habilitado

**Justificación:**

* Protección de credenciales en tránsito
* Prevención de ataques man-in-the-middle

**RD-011: Sanitización de Inputs** **Descripción:** Todas las entradas de usuario deben ser sanitizadas.

**Implementación:**

* XSS: escape de HTML en outputs
* SQL Injection: uso obligatorio de ORM con queries parametrizadas
* Validación de tipos en backend (TypeScript + class-validator)

**Justificación:**

* Prevención de inyecciones maliciosas
* Protección de integridad de datos

**RD-012: Rate Limiting** **Descripción:** Protección contra abuso de endpoints.

**Especificación:**

* Login: 5 intentos / 15 minutos por IP
* API pública: 100 requests / minuto por IP
* API autenticada: 200 requests / minuto por usuario
* Implementación: express-rate-limit o similar

**Justificación:**

* Prevención de brute force
* Protección contra DDoS básico

**Restricciones de Datos**

**RD-013: Modelo de Datos Normalizado** **Descripción:** Base de datos debe seguir tercera forma normal (3NF) mínimo.

**Entidades principales:**

* Users (id, email, password\_hash, name, created\_at)
* Profiles (user\_id, area, level, preferences)
* Interviews (id, user\_id, config, status, created\_at)
* Questions (id, interview\_id, text, metadata)
* Answers (id, question\_id, transcription, analysis)
* Feedback (id, answer\_id, content, scores)

**Justificación:**

* Elimina redundancia
* Facilita integridad referencial
* Mejora eficiencia de consultas

**RD-014: Retención de Datos** **Descripción:** Políticas de almacenamiento y eliminación de datos.

**Especificación:**

* Audio: NO se almacena (solo transcripciones)
* Transcripciones: retención permanente (o hasta eliminación de cuenta)
* Sesiones JWT: expiración 24 horas
* Logs de errores: retención 30 días
* Backups de BD: retención 7 días

**Justificación:**

* Cumplimiento con privacidad
* Optimización de almacenamiento

**Restricciones de Despliegue**

**RD-015: Plataformas de Hosting** **Descripción:** Uso de plataformas con capa gratuita para minimizar costos.

**Opciones aprobadas:**

* **Frontend:** Vercel, Netlify, GitHub Pages
* **Backend:** Railway, Render, Fly.io
* **Base de Datos:** Supabase, Neon, Railway PostgreSQL
* **CDN:** CloudFlare (gratuito)

**Justificación:**

* Presupuesto cero del proyecto académico
* Plataformas confiables con buena documentación

**RD-016: Variables de Entorno** **Descripción:** Configuraciones sensibles deben estar en variables de entorno, no en código.

**Variables obligatorias:**

DATABASE\_URL

JWT\_SECRET

OPENAI\_API\_KEY

GOOGLE\_CLOUD\_API\_KEY

DID\_API\_KEY

NODE\_ENV (development/production)

**Gestión:**

* Archivo .env localmente (no commitear)
* Variables configuradas en plataformas de hosting
* Uso de librerías: dotenv (Node.js)

**Justificación:**

* Seguridad (no exponer secrets en repositorio)
* Flexibilidad entre ambientes

### 7. Atributos del Sistema

### 7.1 Confiabilidad

**Disponibilidad**

**Objetivo:** 95% uptime mensual

**Métricas:**

* Tiempo medio entre fallos (MTBF): > 168 horas (7 días)
* Tiempo medio de recuperación (MTTR): < 4 horas
* Ventana de mantenimiento: Domingos 02:00-04:00 CLT

**Estrategias:**

* Monitoreo con UptimeRobot (gratuito)
* Alertas automáticas por email/Slack
* Health checks en endpoints críticos (/health)

**Tolerancia a Fallos**

**Mecanismos implementados:**

1. **Circuit Breaker para APIs externas:**
   * Tras 5 fallos consecutivos, activar fallback
   * Reintentar cada 60 segundos
   * Notificar al usuario de funcionalidad reducida
2. **Fallbacks por servicio:**
   * LLM caído → banco de preguntas pre-generadas
   * STT caído → input de texto manual
   * Avatar caído → solo audio + texto
3. **Degradación gradual:**
   * Mantener funcionalidades core aunque features avanzadas fallen
   * Prioridad: registro, login, simulación básica

**Recuperación de Datos**

**Estrategias:**

* **Backups automáticos:**
  + Frecuencia: diarios a las 03:00 UTC
  + Retención: últimos 7 días
  + Almacenamiento: servicio de backup de hosting
* **Replicación de BD:**
  + Considerar read replica si carga aumenta
  + Snapshots antes de migraciones

### 7.2 Mantenibilidad

**Modularidad**

**Principios aplicados:**

* Separación de responsabilidades (SoC)
* Principio de responsabilidad única (SRP)
* Inversión de dependencias (DI)

**Estructura modular:**

* Cada servicio backend tiene responsabilidad única
* Componentes frontend reutilizables
* Funciones puras sin efectos secundarios cuando sea posible

**Testabilidad**

**Cobertura de pruebas objetivo:** 70%

**Tipos de pruebas:**

1. **Unitarias:**
   * Framework: Jest (JavaScript/TypeScript)
   * Cobertura: funciones de lógica de negocio
   * Ejecución: automática en pre-commit hook
2. **Integración:**
   * Framework: Supertest (testing de API)
   * Cobertura: endpoints críticos
   * Ejecución: en CI/CD pipeline
3. **End-to-End:**
   * Framework: Cypress o Playwright
   * Cobertura: flujos principales (registro, login, entrevista completa)
   * Ejecución: pre-release
4. **Usabilidad:**
   * Testing con usuarios reales (mínimo 5 usuarios)
   * Métricas: tiempo de completar tareas, tasa de error
   * Feedback cualitativo

**Documentación**

**Componentes de documentación:**

1. **Código:**
   * JSDoc para funciones complejas
   * Comentarios explicativos en lógica no obvia
   * README por módulo
2. **API:**
   * Documentación Swagger/OpenAPI
   * Ejemplos de requests/responses
   * Códigos de error documentados
3. **Usuario:**
   * FAQ integrado en plataforma
   * Tutorial interactivo para nuevos usuarios
   * Video demostrativo (2-3 minutos)
4. **Técnica:**
   * Diagrama de arquitectura
   * Diagrama de base de datos (ERD)
   * Guía de setup para desarrolladores
   * Guía de deployment

### 7.3 Usabilidad

**Curva de Aprendizaje**

**Objetivo:** Usuario puede completar primera entrevista en < 10 minutos desde registro.

**Estrategias:**

* Onboarding interactivo en primer login
* Tooltips contextuales en elementos clave
* Valores por defecto inteligentes (configuración sugerida)

**Consistencia**

**Principios de diseño:**

* Misma paleta de colores en toda la aplicación
* Botones primarios siempre en misma posición
* Feedback visual consistente (success = verde, error = rojo)
* Terminología consistente (no alternar "entrevista" y "sesión")

**Mensajes de Error**

**Características:**

* Claros y específicos: "El correo ya está registrado" en lugar de "Error 409"
* Accionables: "Habilita el micrófono en Configuración" en lugar de "Permiso denegado"
* No técnicos para usuario final
* Con enlace a ayuda si aplica

**Accesibilidad (WCAG 2.1 Nivel AA)**

**Criterios implementados:**

1. **Perceptible:**
   * Alt text en imágenes
   * Contraste de colores > 4.5:1
   * Subtítulos en transcripciones de voz
2. **Operable:**
   * Navegación completa por teclado
   * Tiempo suficiente para responder (configurable)
   * No uso de elementos que causen convulsiones (sin flashes)
3. **Comprensible:**
   * Lenguaje claro y simple
   * Comportamiento predecible
   * Ayuda contextual disponible
4. **Robusto:**
   * Marcado semántico HTML5
   * Compatibilidad con lectores de pantalla
   * ARIA labels donde sea necesario

### 7.4 Eficiencia

**Optimización de Rendimiento**

**Técnicas implementadas:**

1. **Frontend:**
   * Lazy loading de rutas
   * Memoization de componentes (React.memo)
   * Virtualización de listas largas (react-window)
   * Compresión de assets (gzip/brotli)
2. **Backend:**
   * Índices en BD para consultas frecuentes
   * Cache de preguntas frecuentes (Redis)
   * Conexión pooling a BD
   * Compresión de responses
3. **APIs:**
   * Streaming de transcripciones (no esperar audio completo)
   * Debouncing de validaciones en formularios
   * Pagination de historial (10 items por página)

**Uso de Recursos**

**Monitoreo:**

* CPU del servidor: alerta si >80% por >5 minutos
* Memoria: alerta si >90%
* Disco: alerta si <20% libre
* APIs: alerta si se alcanza 80% de cuota

### 7.5 Portabilidad

**Independencia de Plataforma**

**Backend:**

* Código Node.js funciona en Linux, Windows, macOS
* Containerización con Docker garantiza consistencia
* Variables de entorno para configs específicas de plataforma

**Frontend:**

* SPA funciona en cualquier servidor web estático
* Compatible con principales navegadores modernos

**Adaptabilidad**

**Configuración externalizada:**

* Endpoints de APIs en variables de entorno
* Límites de rate en configuración (no hardcoded)
* Textos en archivos de i18n (preparado para traducción)

### 7.6 Seguridad

**Autenticación y Autorización**

**Mecanismos:**

* Passwords hasheadas con bcrypt (12 salt rounds)
* JWT con firma HMAC SHA-256
* Refresh tokens para renovación (implementación futura)
* Validación de token en cada request protegido

**Protección de Datos**

**En tránsito:**

* HTTPS obligatorio con TLS 1.2+
* Headers de seguridad (HSTS, X-Frame-Options, CSP)

**En reposo:**

* Passwords nunca en texto plano
* Variables sensibles en secrets encriptados
* Datos de usuario con restricción de acceso por roles

**Auditoría**

**Loging:**

* Registro de intentos de login (exitosos y fallidos)
* Registro de cambios en datos críticos (perfil, configuración)
* Logs de errores con stack trace y contexto
* Retención de logs: 30 días

**Cumplimiento Normativo**

**Ley 19.628 (Chile):**

* Política de privacidad visible y aceptada en registro
* Derecho a acceso, modificación y eliminación de datos
* Consentimiento explícito para uso de datos
* Notificación de brechas de seguridad (si ocurren)

### 8. Apéndices

**Apéndice A: Glosario Técnico Extendido**

**Algoritmo de Machine Learning:** Conjunto de instrucciones matemáticas que permite a un sistema aprender patrones de datos sin programación explícita.

**Análisis de Sentimiento:** Técnica de NLP que determina tono emocional de un texto (positivo, negativo, neutral).

**Bcrypt:** Función de hash criptográfica diseñada específicamente para hashear contraseñas de forma segura.

**Bounded Context:** Concepto de Domain-Driven Design que define límites claros de responsabilidad de un módulo.

**CI/CD:** Continuous Integration / Continuous Deployment - automatización de testing y deployment.

**Circuit Breaker:** Patrón de diseño que previene cascada de fallos al detectar servicios caídos y activar fallbacks.

**Content Delivery Network (CDN):** Red de servidores distribuidos geográficamente que entregan contenido web de forma optimizada.

**CORS:** Cross-Origin Resource Sharing - mecanismo que permite solicitudes HTTP desde dominios diferentes.

**Debouncing:** Técnica que retrasa ejecución de función hasta que cese actividad por período definido.

**Domain-Driven Design (DDD):** Enfoque de desarrollo que centra el diseño en el modelo del dominio de negocio.

**Exponential Backoff:** Estrategia de reintentos con tiempos de espera incrementalmente mayores.

**Feature Flag:** Técnica que permite activar/desactivar funcionalidades sin deployment.

**Hashing:** Proceso de convertir datos de longitud variable en string de longitud fija (irreversible).

**Idempotencia:** Propiedad de operación que produce mismo resultado sin importar cuántas veces se ejecute.

**JSON Web Token (JWT):** Estándar abierto para transmitir información segura como objeto JSON firmado.

**Load Balancer:** Sistema que distribuye tráfico de red entre múltiples servidores.

**Método STAR:** Situation-Task-Action-Result, metodología para estructurar respuestas comportamentales.

**Migration:** Script que modifica esquema de base de datos de forma versionada y controlada.

**N+1 Query Problem:** Anti-patrón de BD donde se ejecutan consultas adicionales innecesarias en loops.

**ORM:** Object-Relational Mapping - capa que mapea objetos de código a tablas de BD.

**Polling:** Técnica de verificar periódicamente por actualizaciones (alternativa a WebSockets).

**Prepared Statement:** Consulta SQL pre-compilada que previene SQL injection.

**Rate Limiting:** Restricción de número de requests que un cliente puede hacer en período de tiempo.

**Salt:** Valor aleatorio agregado a password antes de hashear para prevenir ataques de rainbow table.

**Semantic Versioning:** Sistema de versionado (MAJOR.MINOR.PATCH) que comunica compatibilidad de cambios.

**Skeleton Screen:** Placeholder visual que imita estructura de contenido mientras carga.

**Soft Delete:** Marcado lógico de registro como eliminado sin borrarlo físicamente de BD.

**Throttling:** Limitación de ejecución de función a máximo una vez por período de tiempo.

**Time to Interactive (TTI):** Métrica de rendimiento que mide cuándo página es completamente interactiva.

**WebSocket:** Protocolo de comunicación bidireccional en tiempo real sobre TCP.

**Apéndice B: Matriz de Trazabilidad**

Relación entre Requisitos Funcionales y Competencias del Perfil de Egreso:

| **Requisito** | **Competencia PE** | **Módulo** | **Prioridad** |
| --- | --- | --- | --- |
| RF-001 a RF-004 | Desarrollar solución de software | Gestión Usuarios | Alta |
| RF-005 a RF-010 | Implementar sistemas inteligentes | Simulación Entrevista | Alta |
| RF-011, RF-012 | Construir modelos de datos | Dashboard | Media |
| RF-013, RF-014 | Desarrollar solución de software | Gamificación | Baja |
| RNF-001 a RNF-005 | Realizar pruebas de certificación | Testing | Alta |
| RNF-006 a RNF-008 | Ofrecer alternativas para toma de decisiones | Seguridad | Alta |

**Apéndice C: Estimaciones de Esfuerzo**

**Desglose por Módulo (en horas-persona)**

**Fase 1: Investigación y Requisitos (Semanas 1-3)**

* Investigación de tecnologías de IA: 20h
* Definición de requisitos detallados: 15h
* Diseño de arquitectura: 20h
* Diseño de base de datos: 10h
* Prototipado de UI/UX: 25h
* **Total Fase 1: 90h**

**Fase 2: Desarrollo Core (Semanas 4-11)**

*Backend (120h):*

* Setup proyecto + infraestructura: 10h
* Módulo autenticación: 20h
* Módulo gestión de entrevistas: 25h
* Integración OpenAI/Gemini: 20h
* Integración Speech-to-Text: 15h
* Integración D-ID Avatar: 20h
* Desarrollo de API REST: 10h

*Frontend (100h):*

* Setup proyecto + estructura: 10h
* Sistema de autenticación: 15h
* Dashboard principal: 20h
* Interfaz de simulación: 30h
* Historial de entrevistas: 15h
* Componentes reutilizables: 10h

*Base de Datos (20h):*

* Diseño de esquema: 5h
* Migraciones iniciales: 5h
* Seeds para testing: 5h
* Optimización de queries: 5h

**Total Fase 2: 240h**

**Fase 3: Funcionalidades Avanzadas (Semanas 12-15)**

* Dashboard de métricas: 25h
* Sistema de feedback inteligente: 30h
* Gamificación: 15h
* Exportación a PDF: 10h
* Optimizaciones de rendimiento: 15h
* **Total Fase 3: 95h**

**Fase 4: Testing y Deployment (Semanas 16-18)**

* Pruebas unitarias: 30h
* Pruebas de integración: 20h
* Pruebas E2E: 15h
* Testing con usuarios reales: 20h
* Corrección de bugs: 25h
* Setup CI/CD: 10h
* Deployment a producción: 15h
* Documentación final: 20h
* **Total Fase 4: 155h**

**Total General: 580 horas-persona**

**Distribución por rol:**

* Backend Developer (Matías): 200h
* Frontend Developer (Ulises): 200h
* AI Specialist & QA (Cristian): 180h

**Apéndice D: Plan de Contingencia**

**Escenarios de Riesgo y Mitigaciones**

**Riesgo 1: APIs de IA inaccesibles o agotamiento de cuotas**

* **Probabilidad:** Media
* **Impacto:** Alto
* **Mitigación:**
  + Implementar banco de preguntas pre-generadas (500+ preguntas)
  + Sistema de análisis basado en reglas simples (keywords, estructura)
  + Considerar API alternativas (Anthropic Claude, Cohere)
  + Optimizar uso de tokens con prompts más eficientes

**Riesgo 2: Dificultad técnica en sincronización de avatar**

* **Probabilidad:** Media
* **Impacto:** Medio
* **Mitigación:**
  + Fallback a avatar estático con solo audio
  + Usar videos pre-renderizados en lugar de generación en tiempo real
  + Priorizar funcionalidad sobre estética del avatar

**Riesgo 3: Retrasos en cronograma**

* **Probabilidad:** Alta
* **Impacto:** Alto
* **Mitigación:**
  + Implementación modular con priorización clara (MVP primero)
  + Buffer de 2 semanas en planificación
  + Reducir alcance de gamificación si es necesario
  + Aumentar horas semanales de trabajo en sprints finales

**Riesgo 4: Problemas de rendimiento con usuarios concurrentes**

* **Probabilidad:** Baja
* **Impacto:** Medio
* **Mitigación:**
  + Implementar cola de procesamiento (Bull/BullMQ)
  + Optimizar queries de BD con índices
  + Implementar cache agresivo (Redis)
  + Considerar serverless functions para picos de carga

**Riesgo 5: Costos inesperados de APIs**

* **Probabilidad:** Media
* **Impacto:** Alto
* **Mitigación:**
  + Monitoreo diario de uso de APIs
  + Alertas al alcanzar 70% de cuota gratuita
  + Limitar número de entrevistas por usuario/día
  + Implementar cache de respuestas comunes

**Apéndice E: Mockups y Prototipos**

*(Referencia a mockups incluidos en documento de proyecto original)*

**Vistas principales incluidas:**

1. Landing page
2. Dashboard de usuario
3. Configuración de entrevista
4. Interfaz de simulación con avatar
5. Vista de feedback detallado
6. Historial de entrevistas

**Herramientas de prototipado:**

* Figma para diseño de UI/UX
* Miro para diagramas de flujo
* Draw.io para diagramas técnicos

**Apéndice F: Referencias Bibliográficas**

1. IEEE Computer Society. (1998). *IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998)*.
2. OpenAI. (2024). *GPT-4 Technical Report*. OpenAI Research.
3. Google Cloud. (2024). *Speech-to-Text API Documentation*. Google Cloud Platform Documentation.
4. D-ID Technologies. (2024). *Creative Reality Studio API Documentation*.
5. Sommerville, I. (2015). *Software Engineering* (10th ed.). Pearson.
6. Pressman, R. S., & Maxim, B. R. (2014). *Software Engineering: A Practitioner's Approach* (8th ed.). McGraw-Hill Education.
7. Martin, R. C. (2017). *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall.
8. Fowler, M. (2018). *Refactoring: Improving the Design of Existing Code* (2nd ed.). Addison-Wesley.
9. Society for Human Resource Management (SHRM). (2024). *2024 State of the Workplace Report*. SHRM Foundation.
10. McDowell, G. L. (2023). *Cracking the Coding Interview: 189 Programming Questions and Solutions* (6th ed.). CareerCup.
11. World Wide Web Consortium (W3C). (2018). *Web Content Accessibility Guidelines (WCAG) 2.1*.
12. OWASP Foundation. (2021). *OWASP Top Ten Web Application Security Risks*.

**Apéndice G: Cronograma Detallado (Carta Gantt)**

*(Referencia al diagrama de Gantt incluido en documento original)*

**Hitos principales:**

* Semana 3: Finalización de diseño y arquitectura
* Semana 7: MVP funcional (login + entrevista básica)
* Semana 11: Funcionalidades core completadas
* Semana 15: Todas las funcionalidades implementadas
* Semana 18: Testing completo y deployment a producción

**Apéndice H: Contactos del Proyecto**

**Equipo de Desarrollo:**

* **Matías Coloma Contreras**
  + Rol: Project Leader & Backend Developer
  + Responsabilidades: Arquitectura, gestión de proyecto, desarrollo backend
* **Cristian Álvarez**
  + Rol: AI Specialist & QA Lead
  + Responsabilidades: Integración de IA, testing, calidad
* **Ulises Muñoz**
  + Rol: Frontend Developer & UX/UI Designer
  + Responsabilidades: Interfaz de usuario, experiencia de usuario

**Supervisión Académica:**

* **Profesor Guía:** Felix Eduardo Cifuentes
* **Institución:** Duoc UC
* **Sección:** 008V

**Apéndice I: Licencias y Atribuciones**

**Licencia del Proyecto:**

* Código fuente: MIT License (open source)
* Documentación: Creative Commons BY-NC-SA 4.0

**APIs de Terceros:**

* OpenAI GPT-4: Términos de servicio de OpenAI
* Google Cloud Speech-to-Text: Términos de servicio de Google Cloud
* D-ID Avatar API: Términos de servicio de D-ID

**Librerías de Código Abierto:**

* React.js: MIT License
* Node.js: MIT License
* PostgreSQL: PostgreSQL License
* (Lista completa en package.json del proyecto)

**APROBACIONES**

| **Rol** | **Nombre** | **Firma** | **Fecha** |
| --- | --- | --- | --- |
| Project Leader | Matías Coloma Contreras | \_\_\_\_\_\_\_\_\_\_ | 16/10/2025 |
| AI Specialist | Cristian Álvarez | \_\_\_\_\_\_\_\_\_\_ | 16/10/2025 |
| Frontend Developer | Ulises Muñoz | \_\_\_\_\_\_\_\_\_\_ | 16/10/2025 |
| Profesor Guía | Félix Eduardo Cifuentes | \_\_\_\_\_\_\_\_\_\_ | **/**/2025 |

**FIN DEL DOCUMENTO**

*Este documento de Especificación de Requisitos de Software (ERS) es un documento vivo que será actualizado a medida que el proyecto evolucione. Cualquier cambio significativo en requisitos debe ser aprobado por el equipo y el profesor guía, y documentado mediante control de versiones.*