Cronograma de Práctica Profesional - EnglishApp Web

Resumen Ejecutivo

Período: 12 de Mayo - 19 de Julio 2024 (9 semanas) **Horario**: Lunes a Viernes, 8 horas diarias **Total**: 360 horas (9 semanas × 5 días × 8 horas) **Proyecto**: Desarrollo de aplicación JavaFX + Migración a React **Empresa**: Colegio Arauco **Rol**: Desarrollador Full Stack

Evolución del Proyecto

- Fase JavaFX (Mayo Primera semana de Junio): Aplicación desktop básica
- Punto de Migración: Primera/Segunda semana de Junio
- Fase React (Junio Julio): Aplicación web completa y moderna

Fecha actual: Julio 2024 (Semana 9 de la práctica) **Progreso**: Aplicación React completamente funcional **Estado**: Proyecto finalizado con todas las funcionalidades implementadas

Tronograma Detallado

FASE 1: DESARROLLO DE APLICACIÓN JAVAFX (Semanas 1-4)

Horas estimadas: 160 horas

Semana 1 (12-16 Mayo): Planificación y Configuración JavaFX

- Análisis de Requerimientos
 - Definición de funcionalidades educativas
 - Planificación de juegos a implementar
 - Diseño de la interfaz de usuario

Configuración del Entorno JavaFX

- Setup del proyecto JavaFX
- o Configuración de Scene Builder
- Estructuración de paquetes y clases

Semana 2 (19-23 Mayo): Navegación y Estructura Base

• Sistema de Navegación

- o Implementación de menús principales
- Navegación entre diferentes secciones
- o Interfaz de usuario básica

Estructura de Datos

- Definición de clases para juegos
- o Sistema de puntuación básico
- Gestión de datos locales

Semana 3 (26-30 Mayo): Implementación de Juegos Básicos

Math Game

- o Lógica de problemas matemáticos
- Una dificultad implementada
- o Sistema de respuestas múltiples

Memory Game

- Lógica de emparejamiento de cartas
- o Animaciones básicas de flashcard
- o Temática única: Figuras geométricas

Semana 4 (2-6 Junio): Juegos Adicionales y Finalización JavaFX

• Typing Game

- o Sistema de escritura con validación
- Temática: Frutas
- o Interfaz de usuario básica

Sorting Game

- o Implementación de días de la semana
- Lógica de ordenamiento básica
- o Interfaz simple

Finalización de Aplicación JavaFX

- Testing de funcionalidades
- Corrección de bugs
- Documentación básica

FASE 2: MIGRACIÓN A REACT (Semana 5)

Horas estimadas: 40 horas

Semana 5 (9-13 Junio): Punto de Migración y Configuración React

• Análisis de Migración

- Evaluación de funcionalidades JavaFX existentes
- o Planificación de migración a React
- o Definición de stack tecnológico web

• Configuración del Proyecto React

- o Inicialización del proyecto React con Vite
- o Configuración de ESLint y herramientas de desarrollo
- o Estructuración de carpetas y archivos
- o Instalación de dependencias (React Router, Framer Motion, etc.)

Migración de Funcionalidades Básicas

- o Recreación de navegación en React Router
- o Migración de lógica de juegos existentes
- o Adaptación de interfaz a componentes React

(Semana 6)

Horas estimadas: 40 horas

Semana 6 (16-20 Junio): Componentes Core y Utilidades

• Desarrollo de Componentes Base

- AssetPreloader.jsx Sistema de precarga de recursos
- BackgroundAudio.jsx Gestión de música de fondo
- ThemeToggle.jsx Sistema de temas claro/oscuro
- MusicToggle.jsx Control de música
- VolumeControls.jsx Control de volumen

• Sistema de Audio y Utilidades

- o Implementación de soundManager.js
- Integración con Howler.js
- Sistema de efectos de sonido
- o progressManager. js Gestión de progreso del usuario
- themeContext.jsx Context API para temas
- o musicState.js Estado global de audio
- o speechSynthesis.js Síntesis de voz

Sistema de Persistencia

- Integración con localStorage
- Configuración de Supabase
- o Sistema de backup de datos

© FASE 4: EXPANSIÓN Y MEJORAS DE JUEGOS (Semana 7)

Horas estimadas: 40 horas

Semana 7 (23-27 Junio): Mejoras y Nuevas Funcionalidades

• Mejoras de Juegos Existentes

- o Memory Game: Múltiples temáticas (Frutas, Animales, Colores, Familia)
- o Typing Game: Múltiples dificultades (Fácil, Medio, Difícil)
- o Math Game: Múltiples temáticas (Números, Partes del Cuerpo)
- o Sorting Game: Nuevas temáticas (Meses, Números)

Nuevos Juegos

- o Sound Matching Game: Emparejamiento de sonidos con imágenes
- o Identification Game: Identificación de objetos y palabras

Desarrollo de Selectores

- o MemoryThemeSelector.jsx
- TypingThemeSelector.jsx y TypingDifficultySelector.jsx
- MathThemeSelector.jsx
- SortingThemeSelector.jsx
- SoundMatchingThemeSelector.jsx
- IdentificationThemeSelector.jsx

III FASE 5: SISTEMA DE ESTADÍSTICAS Y PROGRESO (Semana 8)

Horas estimadas: 40 horas

Semana 8 (30 Junio-4 Julio): Estadísticas Completas

• Página de Estadísticas

- Statistics.jsx Componente principal
- Visualización de progreso por juego
- Métricas: puntuaciones, tiempos, rachas
- Gráficos con Recharts

Sistema de Progreso

- Seguimiento de mejores puntuaciones
- o Cálculo de promedios y tendencias
- o Sistema de rachas diarias
- Historial de partidas

• Filtros y Exportación

- o Filtros por juego, tema y fecha
- Exportación a PDF con jsPDF y html2canvas
- Exportación a Excel con xlsx
- Gráficos exportables

• Panel del Profesor

- TeacherPanel.jsx Acceso protegido
- Visualización de estadísticas globales

- o Exportación de datos avanzada
- o Sistema de autenticación básico

FASE 6: PÁGINAS PRINCIPALES, PWA Y FINALIZACIÓN (Semana 9)

Horas estimadas: 40 horas

Semana 9 (7-11 Julio): Páginas Principales, PWA y Testing

• Página Principal y Navegación

- Home.jsx Dashboard principal
- o Grid de juegos con animaciones
- o Controles globales de audio y tema
- o Diseño responsive
- Instructions.jsx Guía completa de uso

• Configuración PWA

- o manifest.json Configuración de app
- sw.js Service Worker
- Funcionalidad offline
- Instalación como app nativa

Optimizaciones y Testing

- o Lazy loading de componentes
- o Optimización de imágenes
- Tests unitarios con Vitest
- Tests de componentes con Testing Library
- Debugging de funcionalidades

FASE 7: DEPLOYMENT Y DOCUMENTACIÓN FINAL (Semana 9 - Continuación)

Horas estimadas: 20 horas

Semana 9 (7-11 Julio): Deployment y Documentación

• Configuración de Build y Deployment

- Optimización para producción
- o Configuración de variables de entorno
- Minificación y bundling
- Configuración de Netlify
- o Deployment en producción

• Testing de Producción

- o Pruebas en diferentes navegadores
- o Testing de funcionalidad offline
- Verificación de PWA
- o Testing de exportación de datos

• Documentación Final

- README.md principal
- README_TECHNICAL.md
- GUIA_USUARIO.md
- Documentación de API
- Manual de usuario
- Código fuente comentado

■ Distribución de Horas por Área

Área de Trabajo	Horas	Porcentaje
Desarrollo JavaFX	160	44.4%
Migración a React	40	11.1%
Componentes Base	40	11.1%
Expansión de Juegos	40	11.1%
Sistema de Estadísticas	40	11.1%
PWA y Finalización	40	11.1%
TOTAL	360	100%

COMPARACIÓN: APLICACIÓN JAVAFX vs APLICACIÓN REACT

Aplicación JavaFX (Mayo - Primera semana de Junio)

Característica	Estado JavaFX	
Navegación	Menús básicos entre secciones	
Math Game	iame Una dificultad, problemas básicos	
Memory Game	Animaciones básicas, solo temática "Figuras"	
Typing Game	Sistema básico, solo temática "Frutas"	
Sorting Game	Solo "Días de la semana"	
Audio	Sin sistema de audio	
Estadísticas	Sin sistema de seguimiento	
Temas	Sin sistema de temas	
	-	

	Característica	Estado JavaFX
-	Exportación	Sin funcionalidad de exportación
	PWA	No aplicable (aplicación desktop)

Aplicación React (Junio - Julio)

Característica	Estado React	
Navegación	React Router, navegación SPA completa	
Math Game	Múltiples dificultades, temáticas: Números y Partes del Cuerpo	
Memory Game	Animaciones avanzadas con Framer Motion, 4 temáticas	
Typing Game	3 dificultades, múltiples temáticas, métricas WPM	
Sorting Game	3 temáticas: Días, Meses, Números	
Sound Matching	Nuevo juego con audio y emparejamiento	
Identification Game	Nuevo juego de identificación	
Audio	Sistema completo con Howler.js, música y efectos	
Estadísticas	Sistema completo con gráficos y exportación	
Temas	Modo claro/oscuro con persistencia	
Exportación	PDF y Excel con gráficos	
PWA	Funcionalidad offline completa	
Panel Profesor	Acceso protegido con estadísticas globales	

© Entregables por Fase

Fase 1: Planificación

- Documento de análisis de requerimientos
- Arquitectura del sistema
- Stack tecnológico definido

Fase 2: Componentes Base

- Sistema de audio completo
- Gestión de temas
- Componentes de UI reutilizables

Fase 3: Juegos

- **☑** 6 juegos educativos completos
- Sistema de puntuación
- Múltiples temáticas por juego

Fase 4: Estadísticas

- ✓ Sistema de progreso completo
- Panel del profesor
- Exportación de datos

Fase 5: Navegación

- Página principal
- Sistema de navegación
- Página de instrucciones

Fase 6: PWA

- Funcionalidad offline
- Service Worker
- Manifest de PWA

Fase 7: Deployment

- Aplicación desplegada en Netlify
- Documentación completa
- Código fuente entregado

% Tecnologías Utilizadas

Frontend

- React 18.2.0 Framework principal
- Vite 4.4.5 Build tool
- React Router 6.8.1 Navegación
- Framer Motion 10.16.4 Animaciones
- CSS3 Estilos y efectos

Audio y Multimedia

- Howler.js 2.2.4 Gestión de audio
- Web Audio API Audio nativo
- Speech Synthesis Síntesis de voz

Backend y Datos

- Supabase 2.50.0 Backend-as-a-Service
- Recharts 3.0.0 Gráficos
- localStorage Persistencia local

Exportación

• jsPDF 3.0.1 - Generación de PDFs

- html2canvas 1.4.1 Captura de pantalla
- xlsx 0.18.5 Exportación a Excel

Desarrollo

- ESLint Linting
- Vitest Testing
- **Testing Library** Tests de componentes

Código

• Líneas de código: ~15,000+

• Componentes React: 25+

• Utilidades JavaScript: 8

• Tests unitarios: 20+

Funcionalidades

• Juegos educativos: 6

• Temáticas por juego: 4-6

• Sistema de puntuación: Completo

• Exportación de datos: PDF y Excel

Performance

• Tiempo de carga inicial: segundos

• Tamaño del bundle: <2MB

• Funcionalidad offline: Completa

• **PWA Score**: 95+

Aprendizajes y Competencias Desarrolladas

Técnicas

- Desarrollo full-stack con React
- Gestión de estado con Context API
- Integración de APIs y servicios externos
- Desarrollo de PWA
- Testing de aplicaciones web
- Optimización de performance

Blandas

- Gestión de proyectos
- Documentación técnica
- Trabajo independiente

- Resolución de problemas
- Comunicación con stakeholders

Específicas del Dominio

- Desarrollo de aplicaciones educativas
- UX/UI para niños
- Gamificación en educación
- Accesibilidad web
- Responsive design



Notas para el Informe

Puntos Destacados

- 1. Migración exitosa de JavaFX a React
- 2. 6 juegos educativos completamente funcionales
- 3. Sistema de progreso avanzado con exportación
- 4. PWA completa con funcionalidad offline
- 5. Panel del profesor con estadísticas detalladas

Impacto Educativo

- Aplicación utilizada por estudiantes del Colegio Arauco
- Mejora en el aprendizaje del inglés
- Sistema de seguimiento para profesores
- Interfaz adaptada a diferentes edades

Innovación Técnica

- Uso de tecnologías modernas (React 18, Vite)
- Implementación de PWA para acceso offline
- Sistema de audio avanzado
- Exportación de datos educativos

Documento generado para el informe de práctica profesional - Colegio Arauco