Claro, aquí tienes el cronograma detallado **por día**, ajustado a las 2 horas de trabajo diarias que tendrás, desde el **14 de marzo** hasta el **6 de abril**. Este cronograma incluye las tareas específicas para cada día, divididas en bloques manejables para que avances paso a paso.

**Cronograma Detallado por Día**

**Semana 1: Planificación y Estructuración Inicial**

**Día 1 (14 de marzo):**

* **Tarea:** Definir la arquitectura del sistema (Backend, Frontend, Base de Datos).
* **Acciones:**
  + Definir las tecnologías a usar: React.js (Frontend), Node.js con Express.js (Backend), PostgreSQL (Base de Datos).
  + Crear un esquema de la base de datos (tablas para estudiantes, exámenes, resultados, etc.).
  + Desarrollar el diagrama de flujo de la aplicación (inicio de sesión, registro, simulacros, resultados).

**Día 2 (15 de marzo):**

* **Tarea:** Configurar el entorno de desarrollo y el repositorio.
* **Acciones:**
  + Instalar Visual Studio Code y las extensiones necesarias (Prettier, ESLint).
  + Crear el repositorio en GitHub.
  + Configurar un servidor local con Express.js y conectar a la base de datos PostgreSQL.

**Día 3 (16 de marzo):**

* **Tarea:** Finalizar la estructura de la base de datos y las relaciones entre las tablas.
* **Acciones:**
  + Crear las tablas de estudiantes, simulacros, resultados y exámenes.
  + Establecer relaciones entre las tablas (1 a muchos, muchos a muchos, etc.).

**Día 4 (17 de marzo):**

* **Tarea:** Comenzar el desarrollo del Backend.
* **Acciones:**
  + Crear las rutas iniciales del backend (login, registro, creación de simulacros).
  + Configurar JWT para autenticación.

**Día 5 (18 de marzo):**

* **Tarea:** Continuar el desarrollo de la API en Backend.
* **Acciones:**
  + Implementar la lógica de negocio para manejar el registro de usuarios y la autenticación.
  + Añadir validación de datos con express-validator.

**Día 6 (19 de marzo):**

* **Tarea:** Desarrollar las funcionalidades básicas de los simulacros.
* **Acciones:**
  + Implementar el almacenamiento de los simulacros en la base de datos.
  + Crear una función para generar un puntaje después de resolver un simulacro.

**Semana 2: Desarrollo del Backend y API**

**Día 7 (20 de marzo):**

* **Tarea:** Finalizar la API del backend para la gestión de los resultados.
* **Acciones:**
  + Crear las rutas para obtener los resultados de los simulacros.
  + Integrar los datos de los simulacros con los puntajes y evaluaciones de los estudiantes.

**Día 8 (21 de marzo):**

* **Tarea:** Comenzar con la implementación del Frontend.
* **Acciones:**
  + Crear el formulario de inicio de sesión en React.
  + Configurar la conexión con la API para el login.

**Día 9 (22 de marzo):**

* **Tarea:** Continuar con el desarrollo de la interfaz del frontend.
* **Acciones:**
  + Crear la vista de bienvenida después de iniciar sesión.
  + Crear el layout básico de la plataforma (barra lateral, menú).

**Día 10 (23 de marzo):**

* **Tarea:** Integrar el backend con el frontend.
* **Acciones:**
  + Conectar la interfaz de inicio de sesión con la API de autenticación.
  + Implementar la visualización de cursos y simulacros disponibles para los estudiantes.

**Día 11 (24 de marzo):**

* **Tarea:** Comenzar con la visualización de los simulacros.
* **Acciones:**
  + Crear una vista que permita a los estudiantes seleccionar y realizar un simulacro.
  + Integrar los datos de los simulacros con la base de datos.

**Día 12 (25 de marzo):**

* **Tarea:** Desarrollar la lógica para la resolución de simulacros.
* **Acciones:**
  + Permitir que los estudiantes respondan a las preguntas del simulacro.
  + Crear la lógica para calcular los puntajes obtenidos por los estudiantes.

**Día 13 (26 de marzo):**

* **Tarea:** Finalizar la interfaz de resultados.
* **Acciones:**
  + Mostrar los resultados del simulacro con la calificación obtenida.
  + Añadir detalles del simulacro (tiempo, porcentaje, etc.).

**Semana 3: Desarrollo del Frontend y Optimización**

**Día 14 (27 de marzo):**

* **Tarea:** Crear componentes de diseño más complejos (dashboard, perfil del usuario).
* **Acciones:**
  + Crear la página de perfil del estudiante con su historial de simulacros.
  + Mejorar la interfaz de usuario con Material-UI o Tailwind CSS.

**Día 15 (28 de marzo):**

* **Tarea:** Optimizar la visualización de datos y resultados.
* **Acciones:**
  + Añadir gráficos para representar los resultados obtenidos en los simulacros.
  + Mejorar la experiencia de usuario utilizando gráficos dinámicos (Chart.js o Recharts).

**Día 16 (29 de marzo):**

* **Tarea:** Desarrollar la funcionalidad de exportación de resultados.
* **Acciones:**
  + Crear una opción para que los estudiantes puedan descargar los resultados en PDF o Excel.

**Día 17 (30 de marzo):**

* **Tarea:** Test de usuario y pruebas de integración.
* **Acciones:**
  + Realizar pruebas funcionales de la plataforma para garantizar que todas las funcionalidades estén operativas.
  + Solucionar cualquier error o bug identificado.

**Semana 4: Finalización, Optimización y Testing**

**Día 18 (31 de marzo):**

* **Tarea:** Optimizar el backend y la base de datos.
* **Acciones:**
  + Mejorar la velocidad de las consultas a la base de datos.
  + Realizar pruebas de carga y rendimiento.

**Día 19 (1 de abril):**

* **Tarea:** Desplegar la aplicación en un servidor de producción.
* **Acciones:**
  + Desplegar el backend en Heroku o AWS.
  + Desplegar el frontend en Netlify o Vercel.
  + Configurar la base de datos en producción (PostgreSQL en Heroku o DigitalOcean).

**Día 20 (2 de abril):**

* **Tarea:** Realizar pruebas de usuario finales.
* **Acciones:**
  + Validar la experiencia de usuario con un pequeño grupo de estudiantes.
  + Ajustar cualquier detalle según los comentarios recibidos.

**Día 21-22 (3-4 de abril):**

* **Tarea:** Revisión final con el alcalde.
* **Acciones:**
  + Presentar el proyecto al alcalde, explicar el funcionamiento de la plataforma.
  + Recoger retroalimentación final y hacer ajustes si es necesario.

**Día 23 (5 de abril):**

* **Tarea:** Entrega del proyecto final.
* **Acciones:**
  + Entregar la documentación del código y de la API.
  + Proporcionar acceso a los estudiantes y docentes para que comiencen a usar la plataforma.

**Tecnologías Clave:**

* **Frontend:** React.js, Material-UI o Tailwind CSS, Axios
* **Backend:** Node.js, Express.js, JWT
* **Base de Datos:** PostgreSQL
* **Autenticación:** JWT
* **Despliegue:** Heroku, AWS, Netlify, Vercel
* **Pruebas:** Jest, Mocha

Este cronograma te da tareas detalladas para cada día, adaptadas a las 2 horas que trabajarás diariamente. ¡Estoy seguro de que avanzarás de manera eficiente y el proyecto será un éxito! Si necesitas más detalles en cualquier área o alguna aclaración, no dudes en pedírmelo.

**14-marzo-dia 1**