ADSO
(Análisis y Desarrollo de Software)

15/02/2025

Servicio Nacional de Aprendizaje

TABLA DE **CONTENIDO**

Servicio Nacional de Aprendizaje

PLAN CONCERTADO

Plan de trabajo del trimestre, contenido de formación, entregables y fechas.

Competencia	Resultado de aprendizaje	Temática	Evidencia de aprendizaje
CONSTRUCCIÓN DEL SOFTWARE. - DESARROLLAR LA SOLUCIÓN DE SOFTWARE DE ACUERDO CON EL DISEÑO Y METODOLOGÍAS DE DESARROLLO	Planear actividades de construcción del software de acuerdo con el diseño establecido.	Determinar la plataforma tecnológica a utilizar para el desarrollo del software	Documento técnico de las tecnologías usadas en el proyecto.
		Evaluar el alcance real de la plataforma	Documento actualizado del proyecto - Diagrama de Clases, Diagrama de Datos.
		Diferencia conceptual de API, Aplicación Web, Aplicación Móvil	- Investigación del concepto - Diagrama básico de arquitectura
		Elaborar plan de desarrollo de software	-Documento del proyecto - Cronograma de desarrollo
	Codificar el software de acuerdo con el diseño establecido. PARTE 1	Determinar las funcionalidades propias de cada aplicación. Realizar	- Cuadro de funcionalidades Web y móvil
		Desarrollo de API	- Codificación de API Rest
		Desarrollar back-end de la aplicación	Codificación estructural de la plataforma sobre un lenguaje de programación. -Creación de validaciones
		Desarrollar front-end de la aplicación	Codificación estructural de la plataforma sobre un framework. -Creación de validaciones
		Desarrollar front-end y Back-end del proyecto en Clase	- Codificación de API Rest -Documentar API Rest -Integrar API Rest con Frond-end

Lista de entregables (Sustentaciones)

Trimestre 6	En el proyecto se evidencia la implementación de la API REST agregando seguridad.
Trimestre 6	En el proyecto se evidencia que el front end web consuma la API REST (Angular, Vue.js, react u otros), con un avance del 80% de la codificación
Trimestre 6	En el proyecto se evidencia el uso de sistemas de control de versiones.

PROYECTO EN CLASE

Introducción

En este taller intensivo, nos sumergiremos en el mundo del desarrollo de APIs RESTful creando la columna vertebral de un emocionante juego de cartas de personajes de guerreros.

Aprenderemos a diseñar una API robusta y escalable que permita a los jugadores interactuar, seleccionar sus guerreros estratégicamente y participar en épicas batallas virtuales.

PROYECTO EN CLASE

Objetivo

El objetivo principal de este taller es proporcionar a los participantes una experiencia práctica en el diseño e implementación de APIs RESTful utilizando un escenario de juego atractivo y desafiante. Al finalizar el taller, los participantes serán capaces de:

- Comprender los principios fundamentales de la arquitectura REST.
- Diseñar y desarrollar APIs RESTful utilizando herramientas y tecnologías relevantes.
- Implementar la lógica de juego, incluyendo la gestión de jugadores, personajes y partidas.
- Asegurar la API contra vulnerabilidades comunes y garantizar su correcto funcionamiento.

PROYECTO EN CLASE

- ***Alcance***

El alcance del proyecto incluye el diseño, desarrollo e implementación del juego web, incluyendo las siguientes funcionalidades:

- **Introducción a las APIs RESTful:**
  - Conceptos básicos y principios de diseño.
  - Protocolo HTTP y métodos (GET, POST, PUT, DELETE).
  - Formatos de datos JSON y XML.
- **Diseño de la API del Juego:**
  - Definición de los recursos y endpoints de la API.
  - Modelado de datos para jugadores, personajes y partidas.
  - Documentación de la API utilizando herramientas como Swagger.
- **Implementación de la API:**
  - Selección de un lenguaje de programación y framework (ej. Python con Flask/Django, Node.js con Express).
  - Implementación de la lógica de juego y las reglas de combate.
  - Gestión de la persistencia de datos (ej. bases de datos relacionales o NoSQL).

PROYECTO EN CLASE

- ***Alcance***

El alcance del proyecto incluye el diseño, desarrollo e implementación del juego web, incluyendo las siguientes funcionalidades:

- **Seguridad de la API:**
  - Autenticación y autorización de usuarios.
  - Protección contra ataques comunes (ej. inyección SQL, XSS).
  - Manejo de errores y validación de datos.
- **Pruebas y Documentación:**
  - Escritura de pruebas unitarias e integración.
  - Generación de documentación interactiva de la API.

PROYECTO EN CLASE

- ***Requerimientos Funcionales***

El alcance del proyecto incluye el diseño, desarrollo e implementación del juego web, incluyendo las siguientes funcionalidades:

- **Gestión de Jugadores:**
  - Registro de Jugadores: La API debe permitir el registro de nuevos jugadores, incluyendo información como nombre de usuario y contraseña.
  - Autenticación de Jugadores: La API debe permitir a los jugadores iniciar sesión de forma segura para acceder a sus personajes y participar en partidas.

- **Gestión de Personajes:**
-
  - Creación de Personajes: La API debe permitir la creación de nuevos personajes con las siguientes características:
  - Raza
  - Tipo de Guerrero
  - Poder
  - Vida
  - Magia
  - Nombre
  - Selección de Personajes: La API debe permitir a los jugadores seleccionar 5 personajes de su propiedad para participar en una partida.
  - Visualización de Personajes: La API debe permitir a los jugadores ver las características de sus personajes y los personajes de otros jugadores.

PROYECTO EN CLASE

- ***Requerimientos Funcionales***

  - **Gestión de Partidas:**

    - Creación de Partidas: La API debe permitir la creación de nuevas partidas, especificando el modo de juego (por poder, magia o suma de ambos).
    - Unirse a Partidas: La API debe permitir a los jugadores unirse a partidas existentes.
    - Desarrollo de Partidas: La API debe gestionar el desarrollo de las partidas, incluyendo la lógica de combate entre personajes y la determinación del ganador.
    - Finalización de Partidas: La API debe finalizar las partidas y registrar los resultados.

  - **Modos de Juego:**
    - Por Poder: El ganador se determina comparando el poder de los personajes en combate.
    - Por Magia: El ganador se determina comparando la magia de los personajes en combate.
    - Suma de Poder y Magia: El ganador se determina sumando el poder y la magia de los personajes en combate.

PROYECTO EN CLASE

- ***Requerimientos Funcionales***

  - **Ranking y Estadísticas:**

    - Ranking de Jugadores: La API debe mantener un ranking de jugadores basado en sus victorias y otros criterios relevantes.
    - Estadísticas de Jugadores: La API debe registrar estadísticas de cada jugador, como número de partidas jugadas, victorias, derrotas, etc.

PROYECTO EN CLASE

- ***Requerimientos No Funcionales***

  - **Rendimiento:**
    - La API debe responder a las solicitudes de los jugadores de forma rápida y eficiente.
    - La API debe ser capaz de manejar un gran número de jugadores y partidas simultáneamente.
  - **Seguridad:**
    - La API debe proteger la información de los jugadores y las partidas de accesos no autorizados.
    - La API debe implementar medidas de seguridad para prevenir ataques maliciosos.
  - **Mantenibilidad:**
    - La API debe ser fácil de mantener y actualizar.
    - El código de la API debe ser limpio y bien estructurado.

PROYECTO EN CLASE

- ***Modelo de Datos***

- **WARRIOR**
  - **warrior_id (PK): Unique identifier for each warrior.**
  - **name: Warrior's name.**
  - **total_power: Sum of all power attributes.**
  - **total_magic: Sum of all magic attributes.**
  - **health: Warrior's health points.**
  - **speed: Warrior's speed.**
  - **intelligence: Warrior's intelligence level.**
  - **status: Warrior's current state (e.g., active, injured, defeated).**

- **WARRIOR_TYPE**
  - **type_id (PK): Unique identifier for each warrior type.**
  - **name: Type name (e.g., Knight, Mage, Archer).**
  - **description: Description of the warrior type.**

Servicio Nacional de Aprendizaje

PROYECTO EN CLASE

- ***Modelo de Datos***

- **RACE**
    - **race_id (PK): Unique identifier for each race.**
    - **name: Race name (e.g., Human, Elf, Orc).**
    - **description: Description of the race.**
- **POWERS**
    - **power_id (PK): Unique identifier for each power.**
    - **name: Power name (e.g., Fireball, Healing Touch).**
    - **description: Description of the power.**
    - **percentage: Power's effectiveness percentage.**
- **SPELLS**
    - **spell_id (PK): Unique identifier for each spell.**
    - **name: Spell name (e.g., Lightning Bolt, Ice Shield).**
    - **description: Description of the spell.**
    - **percentage: Spell's effectiveness percentage.**

PROYECTO EN CLASE

- ***Modelo de Datos - Relaciones***

- **WARRIOR - WARRIOR_TYPE: A warrior must have one warrior type. A warrior type can have multiple warriors. (One-to-many relationship)**
- **WARRIOR - RACE: A warrior must have one race. A race can have multiple warriors. (One-to-many relationship)**
- **WARRIOR - POWERS: A warrior can have many powers. A power can belong to multiple warriors. (Many-to-many relationship)**
- **WARRIOR - SPELLS: A warrior can have many spells. A spell can belong to multiple warriors. (Many-to-many relationship)**

PROYECTO EN CLASE

- ***Condiciones de Entrega***

- **Aplicación desplegada (local o en línea)**
- **Máximo dos integrantes**
- **Presentación de la aplicación**
- **Documentación del API**
- **Mejores prácticas de programación, codificación y pruebas unitarias**

Servicio Nacional de Aprendizaje

PREGUNTAS

- ??

G R A C I A S

Línea de atención al ciudadano: 01 8000 910270
Línea de atención al empresario: 01 8000 910682

@SENAComunica

www.sena.edu.co