**Manual de Programador**

**1. Introducción**

* **Objetivo del Proyecto**: Descripción breve del juego, su temática, cómo interactúan las diferentes secciones (combate, inventario, personajes, etc.), y el propósito general.
* **Estructura del Código**: Resumen sobre cómo se organiza el código (carpetas, archivos, etc.).

**2. Estructura del Proyecto**

* **HTML**: Descripción de los archivos HTML y cómo están organizados.
  + combate.html: Contiene la vista para el combate.
  + CrearPersonaje.html: Página para crear un nuevo personaje.
  + Inventario.html: Interfaz para gestionar el inventario.
  + Lobby.html: Pantalla principal del juego donde los jugadores eligen opciones.
  + prueba.html: Archivo para pruebas (si es necesario).
  + Regiones.html: Mapa del mundo o áreas del juego.
  + tienda.html: Página de compras dentro del juego.
* **CSS**: Descripción de las hojas de estilo, explicando cómo se organiza el diseño visual.
  + combate.css: Estilos específicos para la vista de combate.
  + CrearPersonaje.css: Estilos para la creación de personajes.
  + Inventario.css: Estilos para la gestión de inventario.
  + Lobby.css: Estilos para la pantalla principal.
  + Regiones.css: Estilos para la visualización de regiones.
  + tienda.css: Estilos para la tienda.
* **JS**: Detalle de las clases y controladores.
  + **Clases**: Explica el propósito de cada clase en el código.
    - arma.js: Representa las armas en el juego, con propiedades como daño, durabilidad, etc.
    - combate.js: Lógica de combate entre el jugador y el enemigo.
    - enemigo.js: Define las características y comportamiento de los enemigos.
    - inventario.js: Gestiona los objetos que el jugador posee.
    - jugador.js: Contiene las estadísticas y habilidades del jugador.
    - musica.js: Maneja la reproducción de música o sonidos en el juego.
    - personaje.js: Define el personaje del jugador con sus atributos (fuerza, salud, magia, etc.).
    - region.js: Define las áreas del juego y su interacción con el jugador.
    - tienda.js: Lógica para el sistema de compras dentro del juego.
  + **Controladores**: Explica el rol de cada controlador en la interacción con las vistas y clases.
    - ControladorCombate.js: Controla las interacciones durante el combate.
    - ControladorCrearPersonaje.js: Maneja la creación del personaje.
    - ControladorInventario.js: Gestiona las acciones del jugador sobre su inventario.
    - ControladorLobby.js: Coordina las interacciones en la pantalla principal.
    - ControladorRegiones.js: Controla las acciones en las distintas regiones del juego.
    - ControladorTienda.js: Maneja las compras en la tienda del juego.

**3. Estructura de Archivos y Carpetas**

* **Carpeta de Recursos**:
  + imagenes/: Contiene todas las imágenes del juego (por ejemplo, sprites, fondos, etc.).
  + sonido/: Archivos de sonido y música (por ejemplo, efectos de sonido, música de fondo, etc.).
  + videos/: Archivos de video (si los hay, como introducciones o animaciones).
* **Carpeta de Documentación**:
  + manual\_usuario.md: Guía para los usuarios finales sobre cómo jugar.
  + manual\_programador.md: Este manual para describir la estructura del código y facilitar modificaciones.

**4. Funciones Principales**

* **Funciones en arma.js**: Explicación de las funciones para crear armas, aplicar daño, etc.
* **Funciones en combate.js**: Lógica para determinar los ataques, calcular daño, etc.
* **Funciones en inventario.js**: Métodos para agregar o quitar objetos del inventario, gestionar cantidades, etc.
* **Funciones en musica.js**: Métodos para reproducir, pausar o cambiar música dentro del juego.
* **Funciones en tienda.js**: Métodos para realizar compras, actualizar el inventario del jugador.

**5. Cómo Realizar Modificaciones o Ampliaciones**

* **Ampliación de Clases**: Guía sobre cómo agregar nuevas clases, cómo integrarlas en el flujo general del juego y ejemplos de cómo usarlas.
* **Modificaciones en el Combate**: Si se desea modificar la lógica de combate (por ejemplo, agregar nuevos tipos de armas o enemigos), cómo hacerlo.
* **Añadir Nuevas Regiones o Zonas**: Cómo crear nuevas áreas en el juego, qué cambios hacer en region.js y cómo integrarlas con el sistema de navegación.
* **Modificación del Inventario**: Explicación de cómo añadir más tipos de objetos o cambiar la forma en que se gestionan dentro del inventario.

**6. Conclusiones**

* Resumen de la estructura del código, mejores prácticas para futuras modificaciones y cómo extender el juego.
* Enlaces a documentación adicional si es necesario.

**7. Glosario**

* Definición de términos técnicos y abreviaturas utilizadas en el manual.