**Juan Esteban Ruiz Ome – Tomas Quintero**

**Requerimientos funcionales**

**RF1: Representación de Galaxia en Forma de Grafo**

* La galaxia ficticia debe representarse mediante un grafo dividido en 5 mapas.
* En los primeros 3 mapas, se utilizará un grafo no dirigido; en los últimos 2, un grafo dirigido.
* Los planetas serán los vértices, y las conexiones entre planetas serán las aristas del grafo.
* Se debe mostrar el peso de cada conexión en la interfaz gráfica.

**RF2: Distribución del número total de vértices y aristas**

* El juego debe contar con un mínimo de 50 vértices y 50 aristas distribuidos en los 5 mapas.
* Cada mapa debe tener al menos 10 vértices y 10 aristas.
* El número de aristas por mapa aumentará progresivamente para incrementar la complejidad del grafo a medida que se avanza de nivel.

**RF3: Implementación de Algoritmos de Grafos**

* Se deben implementar dos algoritmos de grafos, como Dijkstra, BFS, DFS, u otros.
* Estos algoritmos se utilizarán para encontrar el camino óptimo desde el vértice inicial (donde comienza Goku) hasta el vértice final (que representa un enemigo final de cada mapa).
* En la interfaz gráfica, se incluirán dos botones que permitirán a los jugadores ejecutar estos algoritmos.

**RF4: Personajes y Jugabilidad**

* El personaje principal del juego es Goku.
* Cada mapa consta de 10 vértices: 9 representan planetas y el décimo representa un enemigo final de Dragon Ball Z (por ejemplo: Cell, Bills, Freezer, etc.).
* Al ejecutar un algoritmo para encontrar el camino óptimo, Goku se desplazará recorriendo el camino optimo encontrado por las operaciones hacia la posición del enemigo final, rcambiando al siguiente mapa al llegar.

**R5: Pantalla final**

* Cuando goku llegue al mapa 5 (final) y le gane la batalla a su enemigo, se debera mostrar una pantalla que indique que el juego ha finalizado.