Anexo 3. Diagrama de Tareas

HU08

Graficar rutas y red

HU03

Soluciones Completas

HU04

Ranking de Pareto

HU05

La mejor solución

HU09

Interfaz

gráfica

HU01

Verifica modelo red de transporte

HU02

Nodos correctamente secuenciados

HU07

Costo, tiempo y tramos de una solución

HU06

Versión del programa en C++

HU10

Análisis de resultados y eficiencia

HU11

Versión final prototipo en Java

Mejorar eficiencia algoritmos

Análisis eficiencia del prototipo

Medición eficiencia algoritmos

Graficar red de transporte

Graficar soluciones

Seleccionar nodo siguiente

Construir solución

Verificar nodos pertenecen al modelo

Implementar tramos a pie

Construir frente pareto

Elaborar ranking de pareto

Depurar ranking de pareto

Cálculo de distancia

Cálculo de tiempo

Dividir en tramos

Cálculo de costo

Ranking de pareto

Inicializar listas

Cargar nodos

Cargar grafo

Cargar rutas

Normaizar datos

Exploración

Construir solución

Verificar cálculo tiempos y distancias

Diseñar interfaz gráfica según requisitos

Búsqueda

Verificar ranking de pareto

Mejorar eficiencia de código

Análisis de resultados

Optimización de código

Optimizar cálculo de tramos

Crear tutoriales y ayuda

Probar interfaz gráfica

Diseñar funcionalidades principales

Lista de nodos alcanzables

Optimizar cálculo de costos

Análisis de resultados

Optimización y arreglos finales

Evaluación de parámetros

Integración del código

Anexo 4. Historias de Usuario

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HU01  **VERIFICA MODELO RED DE TRANSPORTE**  Desarrollador comprueba que el modelo de la red de transporte sea igual en el programa que el modelado en papel de modo que el modelo esté bien construido.  **Cómo probarlo:** | |  | Importancia |  |
|  | 100 |  |
|  |  |  |
|  | 5 pares de datos aleatorios de la red coinciden tanto en el programa como en el papel |  | Estimación |  |
|  | **Notas:** |  | 03 |  |
|  | Los archivos base de entrada deben estar correctamente construidos.  C++, modo consola  Categoría: MODELO |  | Días |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HU02  **NODOS CORRECTAMENTE SECUENCIADOS**  Usuario verifica que la secuencia de nodos añadidos a una solución sea correcta de modo que el modelo y la búsqueda de nodos trabajen integradamente  **Cómo probarlo:** | |  | Importancia |  |
|  | 80 |  |
|  |  |  |
|  | El nodo seleccionado está entre los alcanzables del nodo actual |  | Estimación |  |
|  | **Notas:** |  | 05 |  |
|  | C++, modo consola  Categoría: PROGRAMACIÓN |  | Días |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HU03  **SOLUCIONES COMPLETAS**  Usuario verifica que las soluciones sean completas de modo que tengan correlación con el modelo  **Cómo probarlo:** | |  | Importancia |  |
|  | 75 |  |
|  |  |  |
|  | Solución va desde el orígen hasta el destino pasando por nodos alcanzables |  | Estimación |  |
|  | **Notas:** |  | 07 |  |
|  | C++, modo consola  Categoría: PROGRAMACIÓN |  | Días |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HU04  **RANKING DE PARETO**  Usuario verifica que las soluciones sean completas de modo que tengan correlación con el modelo  **Cómo probarlo:** | |  | Importancia |  |
|  | 65 |  |
|  |  |  |
|  | 2 ó mínimo 1 solución en el frente de pareto son no dominadas |  | Estimación |  |
|  | **Notas:** |  | 06 |  |
|  | C++, modo consola  Categoría: OPTIMIZACIÓN |  | Días |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HU05  **PARÁMETROS DE LAS SOLUCIONES**  Usuario verifica los datos de la mejor solución de modo que el cálculo de los parámetros de la solución es correcto  **Cómo probarlo:** | |  | Importancia |  |
|  | 60 |  |
|  |  |  |
|  | Se verifica que en costo, tiempo y distancia la solución es la mejor de todas |  | Estimación |  |
|  | **Notas:** |  | 08 |  |
|  | C++, modo consola  Categoría: OPTIMIZACIÓN |  | Días |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HU06  **PROTOTIPO EN C++**  Usuario verifica que el prototipo muestra resultados válidos de modo que los algoritmos funcionan  **Cómo probarlo:** | |  | Importancia |  |
|  | 58 |  |
|  |  |  |
|  | Se ingresa Orígen y Destino y se obtienen resultados en modo texto |  | Estimación |  |
|  | **Notas:** |  | 09 |  |
|  | C++, modo consola  Categoría: PROTOTIPO |  | Días |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HU07  **COSTO, TIEMPO Y TRAMOS DE CADA SOLUCIÓN**  Desarrollador comprueba y optimiza el cálculo correcto de los parámetros de cada solución de modo que los algoritmos calculan correctamente los parámetros  **Cómo probarlo:** | |  | Importancia |  |
|  | 48 |  |
|  |  |  |
|  | Suma de costos, tiempos y tramos de cada solución es correcta |  | Estimación |  |
|  | **Notas:** |  | 06 |  |
|  | C++, modo consola  Categoría: OPTIMIZACIÓN |  | Días |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HU08  **GRAFICAR RUTAS Y REDES**  Usuario visualiza las rutas y las soluciones de modo que Algoritmos grafican correctamente  **Cómo probarlo:** | |  | Importancia |  |
|  | 50 |  |
|  |  |  |
|  | Se dibujan la red y las soluciones correctamente |  | Estimación |  |
|  | **Notas:** |  | 02 |  |
|  | Java, modo visual  Categoría: MODELO |  | Días |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HU09  **INTERFAZ GRÁFICA**  Usuario interactúa con todo el prototipo de modo que la interfaz del prototipo está bien diseñada  **Cómo probarlo:** | |  | Importancia |  |
|  | 35 |  |
|  |  |  |
|  | Ubicar orígen, paraderos cercanos, mostrar tablas de las soluciones y las soluciones funcionan correctamente |  | Estimación |  |
|  | **Notas:** |  | 03 |  |
|  | Java, modo visual  Categoría: PROTOTIPO |  | Días |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HU10  **ANÁLISIS DE RESULTADOS Y EFICIENCIA**  Desarrollador comprueba eficiencia de los resultados de modo que los algoritmos son eficientes  **Cómo probarlo:** | |  | Importancia |  |
|  | 45 |  |
|  |  |  |
|  | Tiempo corto, pocas iteraciones, resultados variados y exitosos |  | Estimación |  |
|  | **Notas:** |  | 07 |  |
|  | Java, modo visual  Categoría: OPTIMIZACIÓN |  | Días |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HU11  **INTEGRACIÓN Y PRUEBA DEL PROTOTIPO EN JAVA**  Usuario comprueba que el prototipo es funcional de modo que se integraron bien todos los módulos  **Cómo probarlo:** | |  | Importancia |  |
|  | 25 |  |
|  |  |  |
|  | Cada funcionalidad es exitosa |  | Estimación |  |
|  | **Notas:** |  | 09 |  |
|  | Java, modo visual  Categoría: PROTOTIPO |  | Días |  |
|  |  |  |  |  |

Historias épicas:

HE01 Modelar la red de transporte público 1 día

HE02 Diagramar la red 2 días

HE03 Crear archivos base de entrada 1 día

HE04 Migrar código 2 días

HE05 Optimizar código 3 días

Anexo 5. Pila del Producto

|  |
| --- |
| HU01 Verifica modelo red de transporte  Importancia: 100 |
|  |
| HU02  Nodos correctamente secuenciados  Importancia: 80 |
|  |
| HU03  Soluciones completas  Importancia: 75 |
|  |
| HU04  Ranking de pareto  Importancia: 65 |
|  |
| HU05  Parámetros de las soluciones  Importancia: 60 |
|  |
| HU06  Prototipo en C++  Importancia: 58 |
|  |
| HU08  Graficar rutas y redes  Importancia: 50 |
|  |
| HU07  Costo tiempo y tramos de cada solución  Importancia: 48 |
|  |
| HU10  Análisis de resultados y eficiencia  Importancia: 45 |
|  |
| HU09  Interfaz gráfica  Importancia: 35 |
|  |
| HU11  Integración y prueba prototipo en Java  Importancia: 25 |

Anexo 6. Pila de Sprints

|  |  |
| --- | --- |
| **SPRINT 1:** “Modelado de red y búsqueda de nodos” | |
| Duración | 9 días |
| **Meta** | **Que el modelo y la búsqueda de nodos funcionen correctamente** |
| Fecha de la Demo | 13 junio 2016 |
| Lugar y momento Scrum | A diario a las 9:00 p.m. |
| Pila del Sprint  HU01 Verifica modelo de la red de transporte (3 días)  HU02 Nodos correctamente secuenciados (5 días)  HE01 Modelar la red de transporte (1 día) | |

|  |  |
| --- | --- |
| **SPRINT 2:** “Obtener los primeros resultados” | |
| Duración | 14 días |
| **Meta** | **Obtener resultados clasificados según su calidad** |
| Fecha de la Demo | 30 junio 2016 |
| Lugar y momento Scrum | A diario a las 9:00 p.m. |
| Pila del Sprint  HU03 Soluciones completas (7 días)  HU04 Ranking de Pareto (6 días)  HE03 Crear archivos de la base de datos (1 día) | |

|  |  |
| --- | --- |
| **SPRINT 3:** “Prototipo en C++” | |
| Duración | 17 días |
| **Meta** | **Mostrar los parámetros de cada solución en el prototipo de C++** |
| Fecha de la Demo | 23 julio 2016 |
| Lugar y momento Scrum | A diario a las 9:00 p.m. |
| Pila del Sprint  HU05 Parámetros de las soluciones (8 días)  HU06 Prototipo en C++ (9 días) | |

|  |  |
| --- | --- |
| **SPRINT 4:** “Optimizar resultados, mejorar eficiencia” | |
| Duración | 19 Días |
| **Meta** | **Optimizar resultados y eficiencia, mostrar soluciones gráficamente** |
| Fecha de la Demo | 17 agosto 2016 |
| Lugar y momento Scrum | A diario a las 9:00 p.m. |
| Pila del Sprint  HU08 Graficar rutas y redes (2 días)  HU07 Costo, tiempo y tramos de cada solución (6 días)  HU10 Análisis de resultados y eficiencia (7 días)  HE02 Diagramar la red (2 días)  HE04 Migrar código (2 días) | |

|  |  |
| --- | --- |
| **SPRINT 5:** “Prototipo en Java” | |
| Duración | 15 días |
| **Meta** | **Entregar prototipo en Java** |
| Fecha de la Demo | 03 setiembre 2016 |
| Lugar y momento Scrum | A diario a las 9:00 p.m. |
| Pila del Sprint  HU09 Interfaz gráfica (3 días)  HU11 Integración y prueba del prototipo en Java (9 días)  HE05 Optimizar código (3 días) | |

Anexo 7. Fichas Resumen de los Sprints

**SPRINT 1**

Modelado de Red y Búsqueda de Nodos

**Fecha de entrega:** 13 junio 2016

|  |  |  |
| --- | --- | --- |
| En Curso | Terminado! | Avances |
|  | Modelar la Red  Cargar nodos  Búsqueda  Exploración  Lista de nodos alcanzables  Cargar rutas  Cargar grafo  Inicializar listas |  |
| No Terminado | No Programado | Siguientes |
| Normaizar datos | Crear tipos de datos |  |

**SPRINT 2**

Obtener los primeros resultados

**Fecha de entrega:** 30 junio 2016

|  |  |  |
| --- | --- | --- |
| En Curso | Terminado! | Avances |
|  | Crear archivos base de datos  Verificar ranking de pareto  Depurar ranking de pareto  Elaborar ranking de pareto  Construir frente pareto  Seleccionar nodo siguiente  Verificar nodos pertenecen al modelo  Construir solución |  |
| No Terminado | No Programado | Siguientes |
| Mejorar eficiencia de código  Implementar tramos a pie |  |  |

**SPRINT 3**

Prototipo en C++

**Fecha de entrega:** 23 julio 2016

|  |  |  |
| --- | --- | --- |
| En Curso | Terminado! | Avances |
|  | Optimización de código  Análisis de resultados  Ranking de pareto  Construir solución  Dividir en tramos  Cálculo de tiempo  Cálculo de distancia |  |
| No Terminado | No Programado | Siguientes |
| Cálculo de costo |  |  |

**SPRINT 4**

Optimizar resultados, mejorar eficiencia

**Fecha de entrega:** 17 agosto 2016

|  |  |  |
| --- | --- | --- |
| En Curso | Terminado! | Avances |
|  | Migrar código  Diagramar la red  Mejorar eficiencia de algoritmos  Análisis de eficiencia del programa  Medición de eficiencia de algoritmos  Graficar soluciones  Graficar red de transporte  Optimizar cálculo de costos  Optimizar cálculo de tramos  Verificar cálculo tiempos y distancias |  |
| No Terminado | No Programado | Siguientes |
|  | Mejorar rapidez de gráficos |  |

**SPRINT 5**

Prototipo en Java

**Fecha de entrega:** 03 setiembre 2016

|  |  |  |
| --- | --- | --- |
| En Curso | Terminado! | Avances |
|  | Optimizar código  Análisis de resultados  Optimización y arreglos finales  Evaluación de parámetros  Integración del código  Probar interfaz gráfica  Diseñar funcionalidades principales  Diseñar interfaz gráfica según requisitos |  |
| No Terminado | No Programado | Siguientes |
| Crear tutoriales y ayuda |  |  |

Anexo 8. Interfaz Gráfica del Prototipo

