

Proyecto Camino Mínimo Metodologías Avanzadas
De Desarrollo De Software.

Leidy T Camacho

Liset P Mahecha

Carlos A Lopez

Noviembre 2022.

Fundación Universitaria Compensar.

Bogotá D.C.

Metodologías Avanzadas De Desarrollo De Software

Proyecto Final Metodologías Avanzadas De
Desarrollo De Software.

Leidy Tatiana Camacho Parra

Liset Paola Mahecha Gama

Carlos Alberto Lopez Cardozo

Facultad Ingeniería de Sistemas.

Metodologías Avanzadas De Desarrollo De Software

Norbey Danilo Muñoz Cañón.

Fundación Universitaria Compensar.

Metodologías Avanzadas De Desarrollo De Software

Bogotá D.C.

Noviembre de 2022

Copyright © 2022 por Leidy Tatiana Camacho Parra, Liset Paola Mahecha Gama, Carlos

Alberto Lopez. Todos los derechos reservados.

Contenido

Proyecto Final Camino Mínimo.	5
Reglas por considerar.....	5
Proyecto.....	5
Rúbrica de calificación:	6
Recursos.....	7
Palabras Clave.....	8
Camino Mínimo.....	9
Introducción	9
Funcionamiento.....	10
Tecnologías	12
Historias de Usuario.....	13
Conclusiones	15
Referencias	16

Proyecto Final Camino Mínimo.

Reglas por considerar

El proyecto final tiene un valor del 67% del último momento evaluativo. El 33% restante corresponde al quiz.

Nota: si algún estudiante desea mejorar la nota del último corte, es decir, intentar tener lo más alta posible la nota del proyecto o quiz, puede realizar el ejercicio de Mapas de historia de usuario que dejará en la sesión del 27 de octubre de 2022 (clase en la que se ve el tema). Ojo, esto es OPCIONAL para aquellos que quieran sumar puntos para este último momento evaluativo.

Proyecto:

Se debe entregar un proyecto que utilice BDD (Behavior Driven Development); es decir un desarrollo orientado por el comportamiento, bajo los siguientes criterios:

- El lenguaje de desarrollo y el framework empleado para BDD son de libre elección.
- El tema de aplicación es de libre elección (la dificultad debe ser mayor al ejemplo clásico de una calculadora o similares).
- Definir mínimo 5 historias de usuario.
- Definir mínimo 2 features.
- Cada feature debe contar con sus escenarios.
- Deben contar con tabla de ejemplos para las pruebas.
- Las pruebas que se implementen, independiente de su naturaleza, deben pasar automáticamente.
- Debe generarse el reporte de las pruebas que genera la herramienta utilizada.
- Se debe tener un repositorio con el proyecto donde debe incluirse en la documentación del

mismo:

- Nombres de los integrantes
- Herramientas utilizadas y sus versiones
- Qué hace la aplicación
- Instrucciones para usar el código del repo (clonar, build, run, etc)
- Historias de usuario
- Features con lenguaje Gherkin
- Conclusiones de aprendizaje con el ejercicio

El máximo de integrantes es tres (3) por equipo.

Rúbrica de calificación:

Para la calificación se tendrán en cuenta los siguientes aspectos, los cuales serán ponderados para la nota final. Cada uno de los items será evaluado en una escala de 0 a 1, donde 0 es NO CUMPLE y 1 es CUMPLE; puede, por ejemplo, en alguno de los items llegar a tener un valor de 0.5; 0.3; 0.7; etc., dependiendo del grado de cumplimiento. Basicamente es una lista de chequeo.

- Grupal:
 - a) Se subió el repositorio del proyecto.
 - b) Explicación del desarrollo (es decir, qué hace, qué es).
 - c) Originalidad y/o creatividad.
 - d) Historias de Usuario.

- e) Features.
- f) Pruebas.
- g) Uso de lenguaje Gherkin.
- h) Paso correcto de las pruebas.
- i) Reporte de pruebas.
- j) Muestra del código y de la ejecución.
- k) Sustentación clara.
- Individual:
 - a) Participación en la sustentación.
 - b) Pregunta realizada por el profesor sobre algún aspecto del desarrollo y que debe responder el estudiante.

El total de puntos es 13 (11 grupal + 2 individual). Se suman los puntos y se realiza la operación:
 $SUMA_DE_PUNTOS * 5 / 13$; el resultado dará la nota del 67% del último corte.

Recursos:

Se adjunta documentación, guías, tutoriales de utilidad:

- Breve explicación de Gherkin: <https://openwebinars.net/blog/que-es-gherkin/>
- Repositorio de ejemplo visto en clase: <https://github.com/Ataches/MCIC-Java21>
- Behave para hacer BDD en Python: <https://jenisys.github.io/behave.example/>
- Specflow para hacer BDD en .NET: <https://specflow.org>
- Lettuce para hacer BDD en Python: <https://pypi.org/project/lettuce/>
- JBehave para hacer BDD en JAVA: <https://jbehave.org>

- Cucumber para hacer BDD en Ruby: <https://cucumber.io>
- Java + Cucumber: <https://www.youtube.com/watch?v=DtSYzJtjXQ>;
<https://www.youtube.com/watch?v=JPLk4Z0U0yQ>

La sustentación se realizará el día 17 de noviembre de 2022 en el horario y salón de clase habituales.

Quién no sustente ese día en el horario establecido tendrá 0. NO HABRÁ HORARIO DE SUSTENTACIÓN ADICIONAL.

Palabras Clave

<i>Vértice</i>	Punto en el que coinciden dos o más aristas de un poliedro o polígono
<i>Arista</i>	Línea formada por la intersección de dos planos.
<i>Algoritmo</i>	Conjunto de instrucciones definidas, ordenadas, no ambiguas y finitas que permiten solucionar un problema, procesar datos y realizar tareas.
<i>Algoritmo Dijkstra</i>	Algoritmo diseñado por Edsger Dijkstra, que permite determinar el camino más corto.
<i>Historia de Usuario</i>	Explicación informal de la función de un software, definida por el usuario final o el cliente.
<i>Software</i>	Conjunto de programas que permiten a la computadora ejecutar diferentes tareas.
<i>Sistema</i>	Se refiere al conjunto de elementos físicos y lógicos que permiten llegar la manipulación de información.
<i>Aplicación</i>	Software diseñado para ejecutar operaciones o funciones específicas.
<i>Rol</i>	Función o papel que una persona desempeña en un lugar o situación.

Camino Mínimo

Introducción

Así como existen muchas formas de expresar nuestros sentimientos (regalos, palabras, acciones, etc.) a nuestros seres queridos, hay muchos caminos o rutas para llegar de un punto A a un punto B. El objetivo de nuestro proyecto Camino Mínimo es encontrar la mejor ruta que requiera menos esfuerzo ya sea en distancias, recurso o tiempo para llegar de un punto A a un punto Z; veamos un ejemplo sencillo para mayor entendimiento:

- Imagina que tienes un mapa del tesoro y deseas encontrarlo en el menor tiempo posible, suponiendo que el tesoro se encuentra oculto en la parte norte de una gran isla y nosotros nos encontramos en la parte sur de la misma, tenemos dos caminos para llegar a él: atravesando la isla, lo cual nos tardaría 4 horas o, rodeando la isla por agua, lo cual constaría de 5 horas. Para lograr llegar al tesoro en el menor tiempo, necesitamos hallar la mejor ruta para llegar al tesoro, minimizando el tiempo que invertiremos en recorrer el camino.



Para lograr el objetivo principal del software, nos basamos en el algoritmo de Dijkstra, el cual consiste en explorar los caminos más cortos que parten de un vértice origen y que, conllevan a todos los demás vértices por medio de un grafo; en el momento en que se obtiene la ruta más corta desde el vértice origen hasta el resto de los vértices del grafo, el algoritmo termina. Con lo esto, podemos indicar que, para nuestro ejemplo anterior, el camino mínimo

resultante sería atravesar la isla, ya que nos constaría una hora menos que si rodeáramos la isla por agua.

Funcionamiento

Nuestro software, muestra un mapa gráfico de una pequeña ciudad hipotética llamada ThunderHollow, donde el usuario final puede definir el esfuerzo en distancia, recursos o tiempo que le toma trasladarse de un punto a otro; para posteriormente, indicar el punto de partida y de llegada deseado, con el cual, el producto mostrará la ruta más óptima para recorrer. Para ello vamos a notar un breve ejemplo del funcionamiento de la aplicación:

I. La pantalla se muestra información diligenciada:



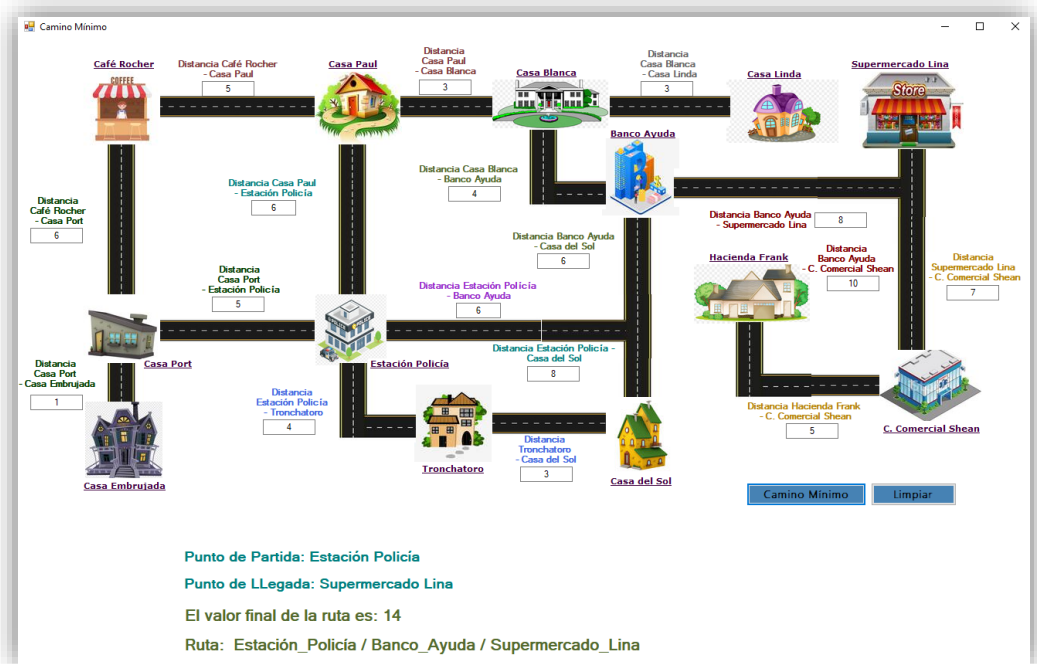
II. Se llena la información del esfuerzo que se toma en recorrer cada arista:



III. Se elige el punto de partida y de llegada seleccionando la imagen de la casa o edificio:



IV. Se selecciona el botón Camino Mínimo para hallar el resultado del calculo realizado por el sistema:



Tecnologías

Las tecnologías implementadas en el desarrollo del sistema son:

Tecnología	Versión
Visual Studio Community	2022
SpecFlow	2022.1.75.53377
.Net Framework	4.8.1
.Core	6

Historias de Usuario

Características de la Historia.	
I	Independiente. No requiere de otra historia.
N	Negociable. Se puede remplazar por otra de diferente prioridad.
V	Valuable / de Valor. Necesaria y de valor para el producto y los consumidores.
E	Estimable. El equipo se siente seguro al estimar el esfuerzo requerido.
S	Sucinta / Pequeña. Se puede construir en una iteración junto a otras historias.
T	Comprobable. Se puede probar y verificar.

A continuación, se refieren las historias de usuario que definen la función y comportamiento que debe tener el Software desde la perspectiva del usuario final:

Historia de usuario: HU01		Título de la historia de usuario: Ver información			
Como	Jugador			I	N
Quiero	Ver la información	X		V	E
		X			
Para	Saber el punto de origen, destino y la distancia que debo recorrer	X		S	T
		X			
Validación / Descripción: <ul style="list-style-type: none">- El jugador puede ver el punto de origen- El jugador puede ver el punto destino- El jugador puede ver la distancia y el camino mínimo que debe recorrer		Valor:		10	
		Prioridad:		Alta	
		Estimación:		1 semana	

Historia de usuario: HU02	Título de la historia de usuario: Definir esfuerzo			
Como	Jugador	X	I	
			N	
Quiero	Definir el esfuerzo que me toma trasladarme de un punto a otro	X	V	
		X	E	
Para	Saber la ruta óptima a recorrer	X	S	
		X	T	
Validación / Descripción: - El jugador debe diligenciar el campo del esfuerzo (campo obligatorio)		Valor:	8	
		Prioridad:	Alta	

<ul style="list-style-type: none"> - El jugador puede definir si el esfuerzo que le tomara de un punto a otro será en tiempo, distancia o recursos monetarios - El jugador puede trasladarse de un punto a otro 	Estimación:	1 semana
---	--------------------	----------

Historia de usuario: HU03	Título de la historia de usuario: Elegir origen y destino			
Como	Jugador	X	I	
			N	
Quiero	Elegir el punto de partida y de llegada	X	V	
		X	E	
Para	Saber el camino mínimo	X	S	
		X	T	
Validación / Descripción: - El jugador debe elegir el punto de partida, seleccionando la imagen de la casa o edificio (campo obligatorio) - El jugador debe elegir el punto de llegada, seleccionando la imagen de la casa o edificio (campo obligatorio)		Valor:	9	
		Prioridad:	Alta	
		Estimación:	1 semana	

Historia de usuario: HU04	Título de la historia de usuario: Seleccionar botón			
Como	Jugador	X	I	
			N	
Quiero	Seleccionar el botón de camino mínimo	X	V	
		X	E	
Para	Saber la ruta óptima a recorrer	X	S	
		X	T	
Validación / Descripción: - El jugador puede seleccionar el botón Camino Mínimo - El jugador puede ver el resultado del cálculo realizado por el sistema		Valor:	9	
		Prioridad:	Alta	
		Estimación:	1 semana	

Historia de usuario: HU05	Título de la historia de usuario: Explorar caminos cortos			
Como	Sistema	X	I	
			N	
Quiero	Investigar cual es el camino corto	X	V	
		X	E	
Para	Mostrarle al jugador	X	S	
		X	T	

Validación / Descripción: <ul style="list-style-type: none"> - El sistema debe tener definido unos vértices (casas o edificios) - El sistema debe estar unido a otros vértices por medio de grafos (camino) - El sistema debe permitir el ingreso de números enteros en los grafos (campo del esfuerzo) - El sistema debe identificar cuáles de los vértices vecinos al origen tiene el grafo de menor valor - El sistema debe mostrarle al jugador el camino más corto que debe recorrer entre el vértice origen y el vértice destino seleccionados 	Valor:	10
	Prioridad:	Alta
	Estimación:	2 semana

Conclusiones

- BDD es una forma de desarrollar, permitiendo que el desarrollo sea orientado a las pruebas y al comportamiento del sistema.
- Para usar BDD, el desarrollo del proyecto debe estar orientado a objetos.
- Gherkin es un lenguaje DSL que facilita la codificación, interpretación y comprensión de la solución de un problema para cualquier tipo de usuario involucrado en un proyecto de desarrollo BDD.
- Las Feauteres y Steps se complementan para verificar y validar el correcto funcionamiento de métodos codificados en el desarrollo.
- Las Features manejan un patrón para dar solución a un problema plasmándolo en un ejemplo:
 - ❖ Given (Dado que), se refiere a las acciones y precondiciones que debe cumplir el escenario.
 - ❖ When (Cuando algo), hace referencia a las condiciones de las acciones que se realizan en el Given.
 - ❖ Then (Entonces), es el resultado que se espera obtener con el escenario.

Referencias

Asana. (n.d.). *Historias de usuario: 3 ejemplos para generar valor para el usuario*. Asana.

Retrieved November 11, 2022, from <https://asana.com/es/resources/user-stories>

Cultura tecnológica: ¿Qué son las aplicaciones o programas? (n.d.). Gcfglobal.org. Retrieved November 11, 2022, from <https://edu.gcfglobal.org/es/cultura-tecnologica/que-son-las-aplicaciones-o-programas/1/>

Oxford Languages and Google - Spanish. (2020, May 20). Oup.com.

<https://languages.oup.com/google-dictionary-es/>

Rehkopf, D. M. (n.d.). *Historias de usuario*. Atlassian. Retrieved November 11, 2022, from

<https://www.atlassian.com/es/agile/project-management/user-stories>

Sistema en Informática. (n.d.). Sistemas.com. Retrieved November 11, 2022, from

<https://sistemas.com/sistema-informatica.php>

Wikipedia contributors. (n.d.-a). *Algoritmo*. Wikipedia, The Free Encyclopedia.

<https://es.wikipedia.org/w/index.php?title=Algoritmo&oldid=146592655>

Wikipedia contributors. (n.d.-b). *Algoritmo de Dijkstra*. Wikipedia, The Free Encyclopedia.

https://es.wikipedia.org/w/index.php?title=Algoritmo_de_Dijkstra&oldid=146185477

Wikipedia contributors. (n.d.-c). *Sistema*. Wikipedia, The Free Encyclopedia.

<https://es.wikipedia.org/w/index.php?title=Sistema&oldid=147087198>

Wikipedia contributors. (n.d.-d). *Vértice (teoría de grafos)*. Wikipedia, The Free Encyclopedia.

[https://es.wikipedia.org/w/index.php?title=V%C3%A9rtice_\(teor%C3%ADa_de_grafos\)&oldid=131817620](https://es.wikipedia.org/w/index.php?title=V%C3%A9rtice_(teor%C3%ADa_de_grafos)&oldid=131817620)

(N.d.-a). Dreamstime.com. Retrieved November 10, 2022, from

<https://thumbs.dreamstime.com/b/mapa-del-juego-tesoro-de-islas-tropicales-dibujos-animados-que-muestran-la-direcci%C3%B3n-camino-al-oro-pirata-treasure-game-map-197448830.jpg>

(N.d.-b). Retrieved November 10, 2022, from [http://chrome-](http://chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/http://www.oia.unsam.edu.ar/wp-content/uploads/2017/11/dijkstra-prim.pdf)

[extension://efaidnbmnnnibpcajpcglclefindmkaj/http://www.oia.unsam.edu.ar/wp-content/uploads/2017/11/dijkstra-prim.pdf](http://chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/http://www.oia.unsam.edu.ar/wp-content/uploads/2017/11/dijkstra-prim.pdf)