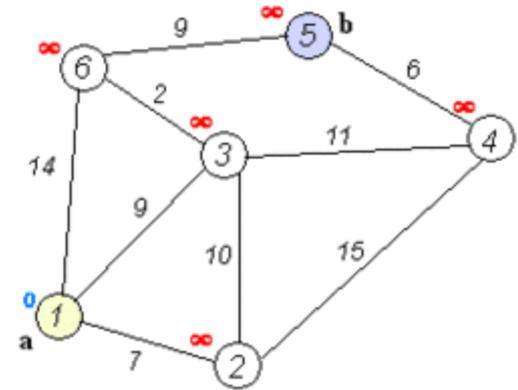




GrafoMin



Una herramienta para el aprendizaje y la construcción del camino más corto entre dos nodos de un grafo



¿Qué es **GrafoMin**?

- **GrafoMin** es una herramienta gráfica para el aprendizaje y la construcción del camino más corto entre dos vértices de un grafo.
- **GrafoMin** es innovadora porque es la primera aplicación existente programada en código HTML5, CSS y JavaScript.
- **GrafoMin** es independiente de la plataforma.
- **GrafoMin** puede estar alojada en un servidor o residente en una computadora.



Objetivos

- Uno de los problemas clásicos de la Teoría de Grafos es la determinación del camino más corto entre dos de sus vértices: el camino más corto entre dos ciudades, la ruta más corta entre dos routers, etc.
- Resuelto en 1959 Edsger Wybe Dijkstra.
- Dijkstra utilizó los principios de la búsqueda en anchura sobre un grafo para construir un árbol, cuyo vértice raíz es el nodo inicial y cuyas aristas definen el camino más corto, si existe, entre dicho vértice raíz, y el resto de los nodos del grafo.



Objetivos

- Implementación del algoritmo de Dijkstra con fines didácticos.
 - ❑ Aprendizaje del alumno:
 - ✓ puede plantear cualquier supuesto;
 - ✓ contrastará sus resultados: analíticos y de resolución.
 - ❑ Enseñanza del formador:
 - ✓ resolución gráfica para un planteamiento a un grupo de alumnos;
 - ✓ computadora y proyector.



Objetivos

- Elaboración de la herramienta en un entorno multiplataforma que permita el acceso tanto al alumno, como al profesor.
- El formador debe tener disponible la herramienta en diferentes aulas.
- El alumno también debe poder disponer de la herramienta desde su lugar habitual de estudio: su domicilio, la biblioteca u otros lugares de estudio.
- La herramienta debe ser de fácil aprendizaje y uso, para animar a los estudiantes a su utilización.



Estado del arte

➤ **PathFinder**

✓ Java (PFC U.P.M.).

➤ **Minimum Path**

✓ Applet Java (Universidad de las Américas).

➤ **Algraf Project**

✓ C# .Net (PFC Universidad de Informática de Sevilla).

➤ **Dijkstra Algorithm**

✓ Ejecutable (Chris Wenk).

➤ **Grafos**

✓ C# .NET (Universidad Politécnica de Valencia).



Elección de la tecnología

¿Por qué HTML5, CSS, JavaScript?

- Utilización de estándares: HTML, CSS
- Es necesario la versión 5 de HTML para la etiqueta `<CANVAS>` junto a un lenguaje de *scripting*
- JavaScript vs ECMAScript sobre SVG
 - Menor coste de procesamiento de imágenes
 - Imágenes vectoriales: posicionamiento inmediato
 - SVG es una recomendación, no un estándar.



Análisis, diseño e implementación

Ciclo de vida del software

Se ha elegido el modelo **en cascada** por ser el paradigma adecuado para este tipo de situaciones en las que **se deben superar** las pruebas para la **adaptación al nuevo estándar HTML5**, obligando a un rediseño si fuere necesario y una nueva codificación.



Requisitos del sistema

- Requisitos de **usuario**
 - **funcionales**
 - ✓ capacitivos
 - ✓ restrictivos
 - **no funcionales**
- Requisitos **tecnológicos**
- Requisitos de **desarrollo**
- Requisitos de la **interfaz**



Requisitos del sistema

- Requisitos de **usuario**
 - **funcionales**
 - ✓ capacitivos
 - ✓ restrictivos
 - no funcionales
- Requisitos **tecnológicos**
- Requisitos de **desarrollo**
- Requisitos de la **interfaz**



➤ Requisitos de **usuario**

➤ funcionales:

- grafos no dirigidos y dirigidos

- el **usuario indicará:**

- ✓ el **peso** de cada arista (valores positivos),

- ✓ vértices **inicial** y **final**,

- ✓ **rectificación de acciones** ejecutadas:

- borrado de un vértice (solo si no posee aristas)
- modificar el valor del peso de las aristas
- cambiar de grafo dirigido a no dirigido y viceversa
- en la construcción del camino más corto:
 - borrar una arista seleccionada
 - reordenar las aristas seleccionadas



Requisitos del sistema

➤ Requisitos de **usuario**

➤ funcionales:

- **resolución** del problema mediante el algoritmo de **Dijkstra**, si así lo solicita el usuario, sin construir el árbol.
- **indicativos analíticos**:
 - ✓ matriz de adyacencia
 - ✓ matriz analítica
- **ayuda en línea**:
 - ✓ vídeo-tutoriales
 - ✓ documentación complementaria mediante acceso a Internet



Requisitos del sistema

- Requisitos de **usuario**
 - funcionales
 - ✓ capacitivos
 - ✓ restrictivos
 - **no funcionales**
- Requisitos **tecnológicos**
- Requisitos de **desarrollo**
- Requisitos de la **interfaz**



Requisitos del sistema

➤ Requisitos de **usuario**

➤ no funcionales:

- **herramienta gráfica:** construcción de vértices, aristas y valoración de pesos,
- **sin límites** de número de **vértices** ni de **aristas** (no admisible multiarista ni lazos).
- arquitectura **cliente-servidor:**
 - ✓ servidor Web (mínimo Intel Pentium® o AMD K5 ®),
 - ✓ cliente con cualquier sistema operativo y navegador Mozilla o Chrome,
- herramienta autónoma e **independiente de otros sistemas** de información,
- bajo *petición al servidor:*
 - ✓ no se guardarán datos de usuario en el servidor (ley de protección 15/1999),
 - ✓ límite de usuarios: marcado por la capacidad del servidor y su ancho de banda,
- opción de trabajo *fuera de línea:*
 - ✓ no ejecutar operaciones en el servidor



Requisitos del sistema

➤ Requisitos de **usuario**

➤ no funcionales:

- **herramienta gráfica:** construcción de vértices, aristas y valoración de pesos,
- **sin límites** de número de **vértices** ni de **aristas** (no admisible multiarista ni lazos).
- arquitectura **cliente-servidor:**
 - ✓ servidor Web (mínimo Intel Pentium® o AMD K5 ®),
 - ✓ cliente con cualquier sistema operativo y navegador Mozilla o Chrome,
- herramienta autónoma e **independiente de otros sistemas** de información,
- bajo *petición al servidor:*
 - ✓ no se guardarán datos de usuario en el servidor (ley de protección 15/1999),
 - ✓ límite de usuarios: marcado por la capacidad del servidor y su ancho de banda,
- opción de trabajo *fuera de línea:*
 - ✓ no ejecutar operaciones en el servidor



Requisitos del sistema

- Requisitos de **usuario**
 - funcionales
 - ✓ capacitivos
 - ✓ restrictivos
 - no funcionales
- Requisitos **tecnológicos**
- Requisitos de **desarrollo**
- Requisitos de la **interfaz**



Análisis

➤ Requisitos de **usuario**

➤ funcionales:

▪ **capacitivos**

- ✓ El usuario podrá crear un vértice haciendo doble click
- ✓ El usuario indicará con el ratón el vértice donde se inicia la arista. Si el grafo es dirigido, éste corresponde al vértice inicial de la arista
- ✓ El usuario indicará con el ratón, el vértice (final si el grafo es dirigido) donde termina la arista.
- ✓ El usuario podrá cambiar el valor del peso de la arista en el valor indicado en ella
- ✓ Indica si el grafo es dirigido o no dirigido. Podrá seleccionarse en cualquier momento antes de la resolución del problema.
- ✓ El usuario indicará el identificador del nodo inicial antes de la resolución del problema.
- ✓ El usuario podrá borrar con el ratón un vértice, siempre y cuando no sea incidente con ninguna arista.
- ✓ El usuario podrá borrar, con el ratón, un vértice siempre y cuando dicho vértice no contenga aristas.
- ✓ El usuario seleccionará una arista, previamente no seleccionada ya, para incluirla en el camino más corto. Se utilizará el método “arrastrar y soltar”.
- ✓ El usuario podrá reordenar la posición de orden de una arista dentro de las elegidas para el camino más corto. “Arrastrar y soltar”.
- ✓ El usuario podrá eliminar una arista de las elegidas para el camino mínimo de dicho conjunto. “Arrastrar y soltar”.
- ✓ La herramienta generará y mostrará la matriz de adyacencia valorada con pesos.
- ✓ La herramienta generará la matriz analítica ejecutando el algoritmo de Dijkstra.
- ✓ La herramienta ejecutará el algoritmo de Dijkstra y dibujará sobre el propio grafo la solución. También mostrará de forma ordenada el conjunto de aristas seleccionadas para dicho camino, así como las matrices analítica y de adyacencia.
- ✓ La herramienta mostrará ayuda en línea si hay conexión a Internet. Lo hará en un IFRAME, sin modificar el tamaño del grafo.
- ✓ La herramienta mostrará en un IFRAME el acceso a cualquier página de Internet, si hubiere conexión, sin modificar el tamaño del grafo.



Requisitos del sistema

- Requisitos de **usuario**
 - **funcionales**
 - ✓ capacitivos
 - ✓ restrictivos
 - no funcionales
- Requisitos **tecnológicos**
- Requisitos de **desarrollo**
- Requisitos de la **interfaz**



Análisis

➤ Requisitos de **usuario**

➤ funcionales:

▪ **restrictivos**

- ✓ La aplicación se codificará en HTML5 y JavaScript con CSS3 para el diseño
- ✓ El español será el idioma por defecto
- ✓ La aplicación podrá ejecutarse sobre los navegadores Mozilla Firefox V 10 y Google Chrome V 18
- ✓ El navegador deberá tener habilitada la ejecución de JavaScript
- ✓ No se permitirán lazos para ningún vértice
- ✓ El usuario indicará con el ratón el vértice donde se inicia la arista. Si el grafo es dirigido, éste corresponde al vértice inicial de la arista
- ✓ No se permitirá al usuario crear dos aristas entre dos mismos vértices
- ✓ No se permitirá borrar vértices que contengan alguna arista
- ✓ No se calculará el camino más corto si en número de vértices o de aristas es menor que 2 o si no se han asignado los vértices inicial y final.
- ✓ Si el usuario desea acceder a información contenida en Internet para complementar el aprendizaje y que sea mostrado el área correspondiente IFRAME
- ✓ Para poder ver los video tutoriales de aprendizaje de la herramienta, el navegador deberá tener instalado el correspondiente complemento. Se mostrará en el área IFRAME
- ✓ Si el usuario no dispone de conexión a Internet, deberá descargar la versión de la herramienta que contiene los archivos tipo “*flash*” de aprendizaje, para disponer de ellos en la carpeta “help”.



Requisitos del sistema

- Requisitos de **usuario**
 - funcionales:
 - ✓ capacitivos
 - ✓ restrictivos
 - no funcionales
- Requisitos **tecnológicos**
- Requisitos de **desarrollo**
- Requisitos de la **interfaz**



Requisitos del sistema

➤ Requisitos de tecnológicos

➤ 256 RAM (recomendado 512)

➤ Tarjeta gráfica de 800 x 600 (recomendado 1280 x 1024)

➤ Navegadores

- Mozilla Firefox 3.5 ó,
- Google Chrome 1.0



Requisitos del sistema

- Requisitos de **usuario**
 - funcionales
 - ✓ capacitivos
 - ✓ restrictivos
 - no funcionales
- Requisitos **tecnológicos**
- Requisitos de **desarrollo**
- Requisitos de la **interfaz**



Requisitos del sistema

➤ Requisitos de **desarrollo**

- Firebug
- Tarjeta de red
- Conexión a Internet
- Microsoft® XP como sistema operativo
 - Microsoft Word®
 - Microsoft PowerPoint®
 - Microsoft Project®
 - Microsoft Visio®
 - Snagit®
 - Macromedia Fireworks®



Requisitos del sistema

- Requisitos de **usuario**
 - funcionales
 - ✓ capacitivos
 - ✓ restrictivos
 - no funcionales
- Requisitos **tecnológicos**
- Requisitos de **desarrollo**
- Requisitos de la **interfaz**



Requisitos del sistema

➤ Requisitos de la **interfaz**

➤ Pantalla

- 800 x 600 mínimo; 1280 x 1024 recomendado; mostrando el grafo y el IFRAME de ayuda

➤ Usabilidad: “click” de ratón, arrastrar y soltar, globos de ayuda, etc.

➤ Ayuda en línea con vídeo-tutoriales

➤ Idioma: español

➤ Cuadros de diálogo

- informativos de obligatoriedad para entrada de datos
- funciones no disponibles
- errores

➤ Uso de estándares de diseño: hoja de estilos en cascada

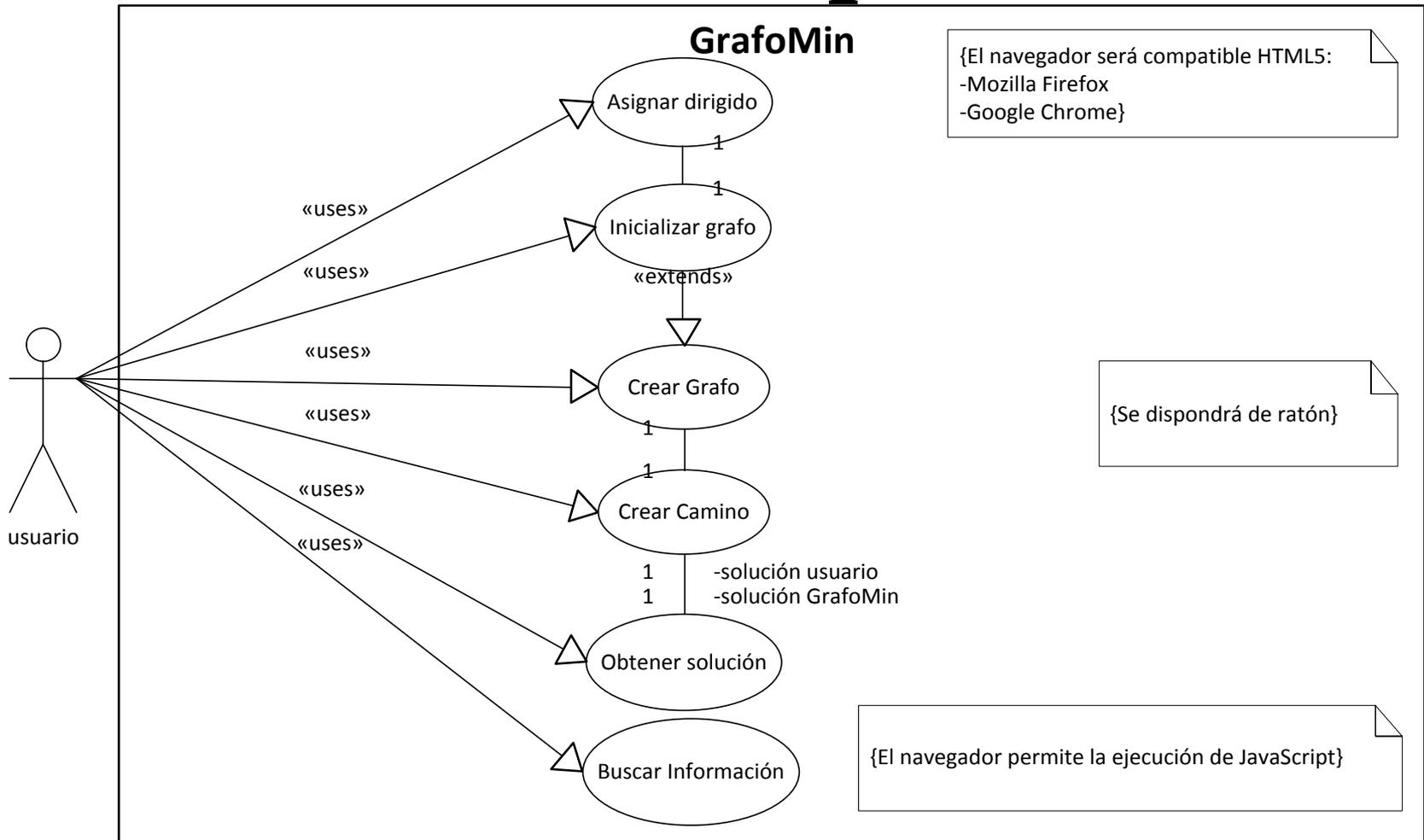


Conocidos los requisitos,
**¿cuál será la estructura de la
herramienta?**

- Diseño de la **arquitectura**
mediante *casos de uso* (*UML*)
- Diagrama de **clases**

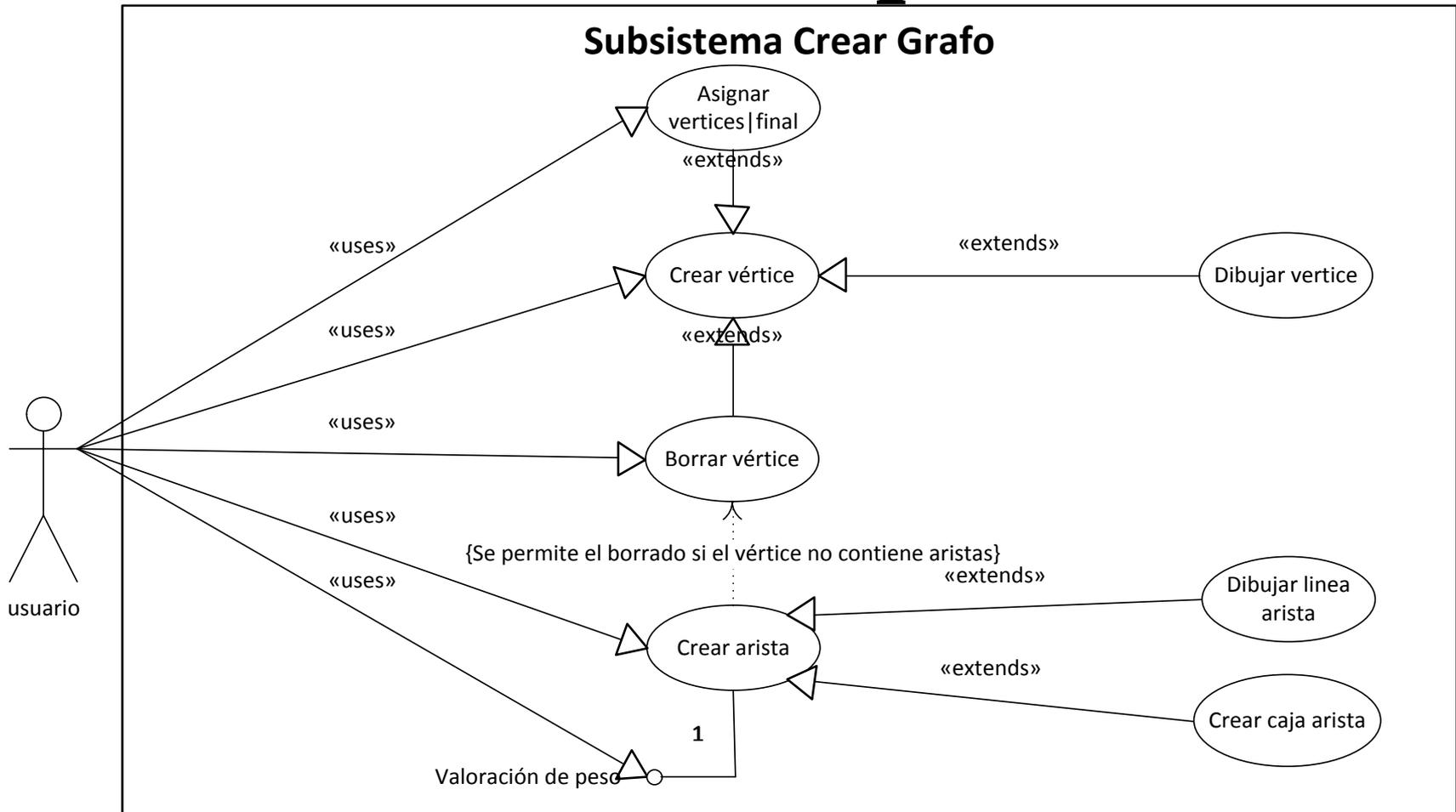


Diseño de la arquitectura





Diseño de la arquitectura





Diseño

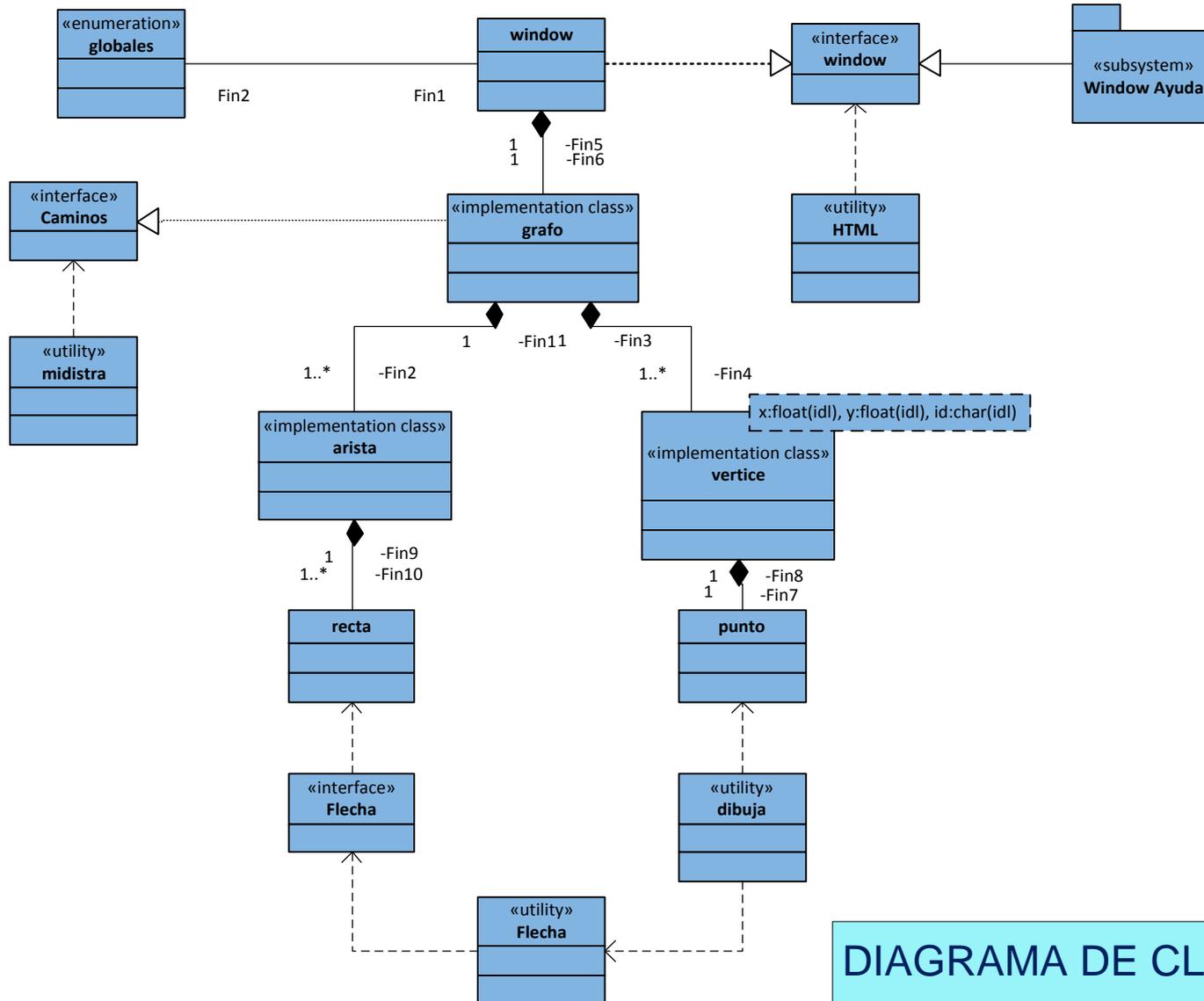


DIAGRAMA DE CLASES



Diseño de la interfaz

<= Índice =>

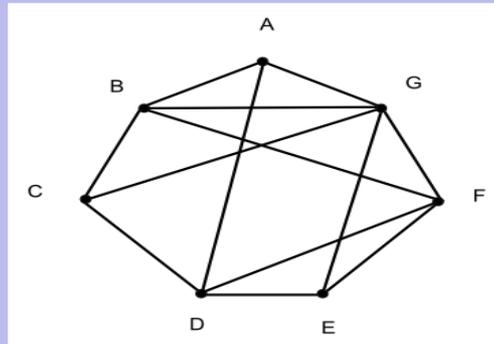
Teoría de grafos

En cada pregunta, marcar una sola respuesta.
El símbolo [Pxx] al final de cada pregunta señala el número xx de dicha pregunta en la base de datos.

Mostrar todas las preguntas

<= 3 / 10 =>

¿Cuál de las siguientes propiedades cumple el grafo de la figura? [P26]



- A. Es semi-euleriano.
- B. Tiene número cromático 4.
- C. Es bipartito.
- D. Es hamiltoniano.

<= Índice =>

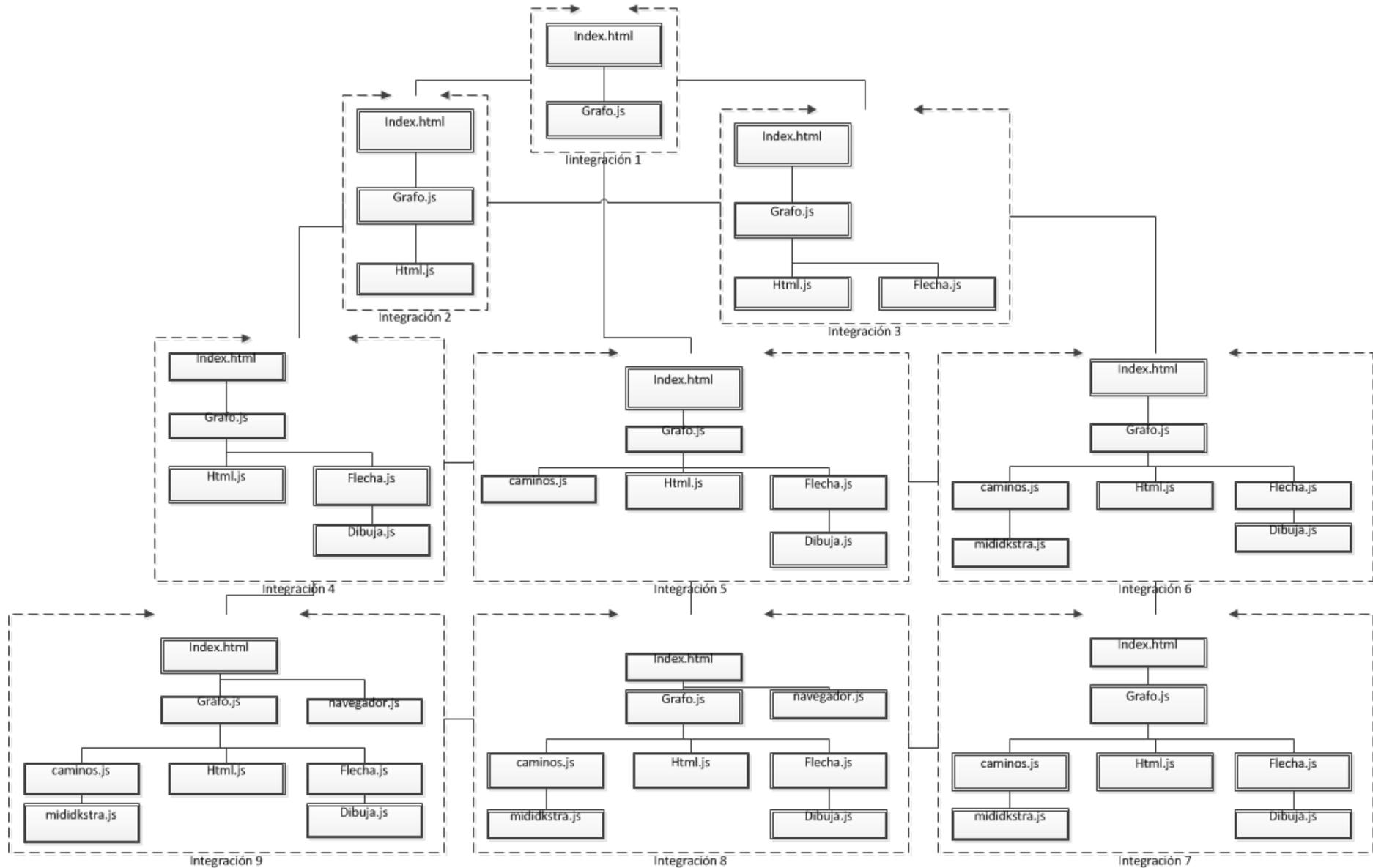


Implementación

Conocemos los requisitos del
sistema y
la arquitectura de la herramienta
¿Cómo se crea el código?



Implementación





GrafoMin



GrafoMin



Planificación y presupuesto

¿Cuáles son las **tareas** necesarias para la **ejecución del proyecto?**

(Microsoft Project®)

- Objetivos parciales
- Tiempos de demora o elasticidad
- Fases de la ejecución: hitos o tareas finalizadas
- Recursos
 - materiales
 - personal
- valoración del coste total



Planificación y presupuesto

➤ Estimación

➤ La **duración** del proyecto (**140 días** laborales trabajados)

✓ iniciado el lunes 9 de Enero de 2012

✓ finalizado el lunes 30 de Julio de 2012,

➤ Costes

✓ Horas: 2718 horas (duración de la jornada laboral de 8 horas)

✓ Económicos: 33.171 € (costes materiales no imputados)



Planificación y presupuesto

➤ Organización

➤ Materiales

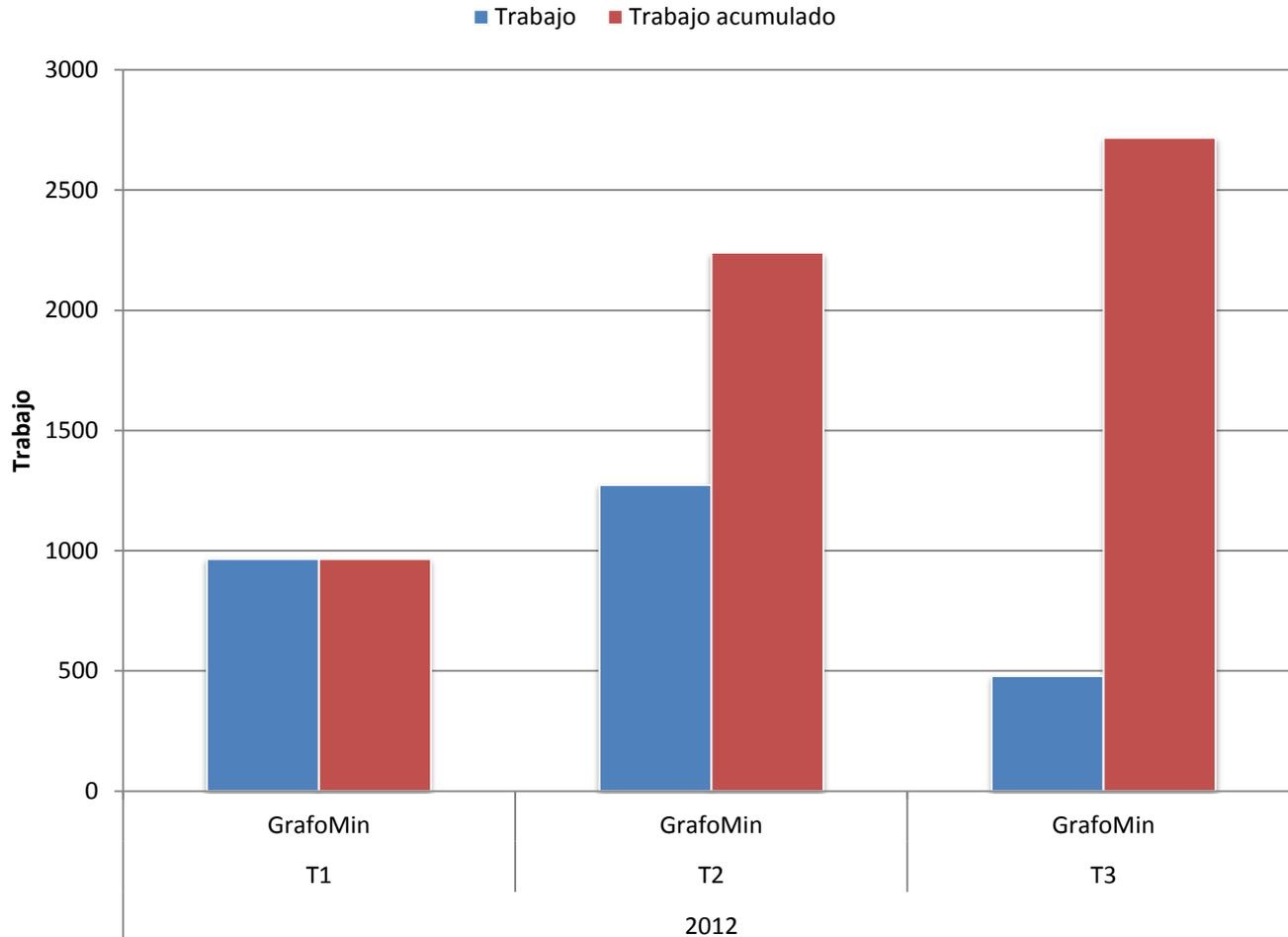
➤ **Humanos** [15 trabajadores]

- Jefe de proyecto [1 trabajador]; 80 €/h
- Analistas [2 trabajadores]; 60 €/h
 - ✓ Arquitecto del sistema (1)
 - ✓ Líder de desarrollo (1)
- Programadores [7 trabajadores]; 50 €/h
- Responsable de calidad y verificación de pruebas [1 trabajador]; 70 €/h
- Diseñadores [4 trabajadores];
 - ✓ Responsable de materiales y diseñador de arquitectura (1); 50 €/h
 - ✓ Diseñador de arquitectura (1); 50 €/h
 - ✓ Diseñador gráfico (2); 20 €/h



Planificación y presupuesto

Informe del trabajo presupuestado

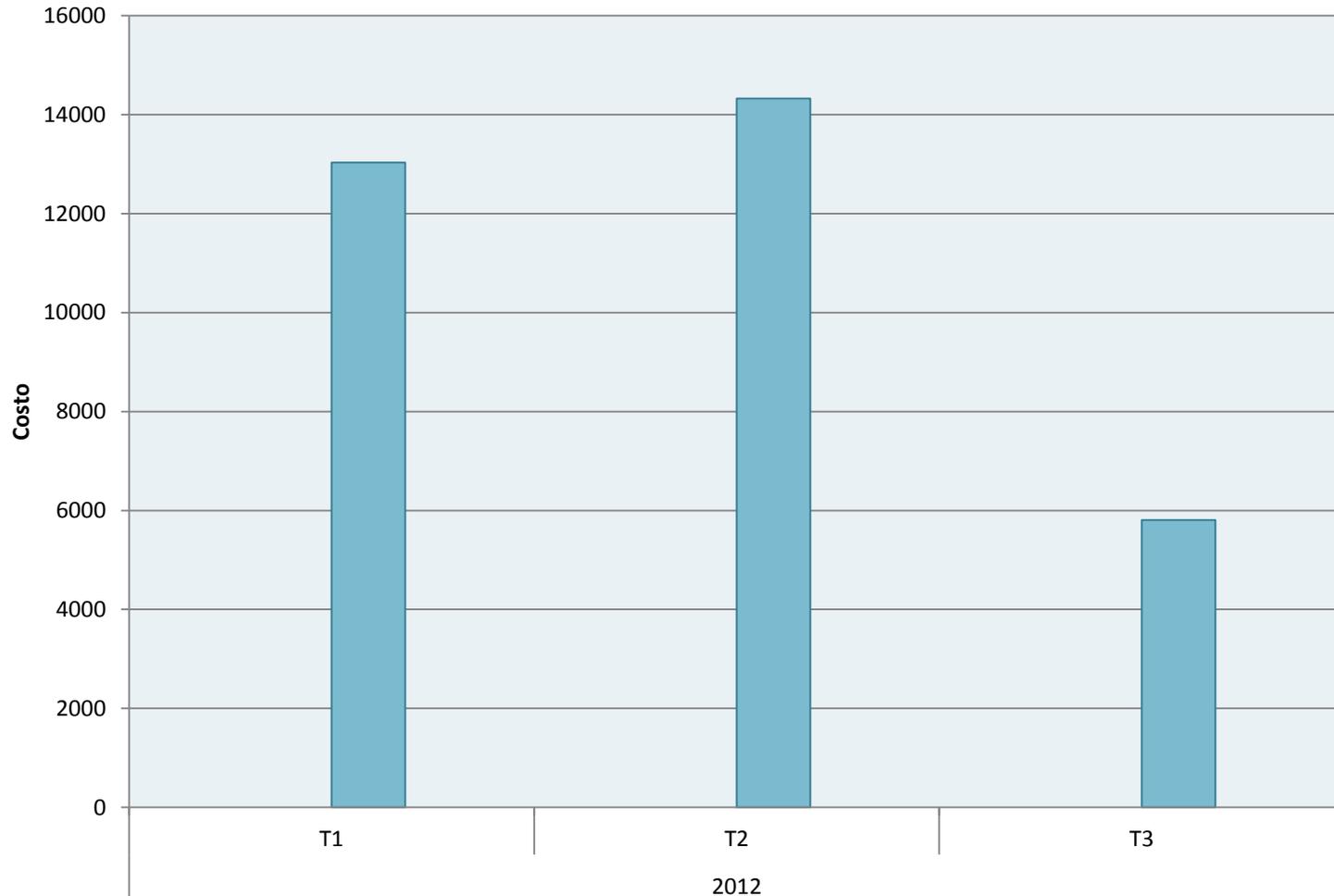




Planificación y presupuesto

Informe de costo presupuestado

■ Costo presupuestado ■ Costo previsto ■ Costo ■ Costo real





Planificación y presupuesto

➤ Planificación

➤ A.1 Planificación y especificación de requisitos

➤ A.2 Construcción

- A.2.1 Análisis
- A.2.1 Diseño
- A.2.3 Pruebas

➤ A.3 Gestión

- A.3.1 Estimación
- A.3.2 Organización
- A.3.3 Planificación
- A.3.4 Seguimiento
- A.3.5 Creación de líneas futuras



Planificación y presupuesto

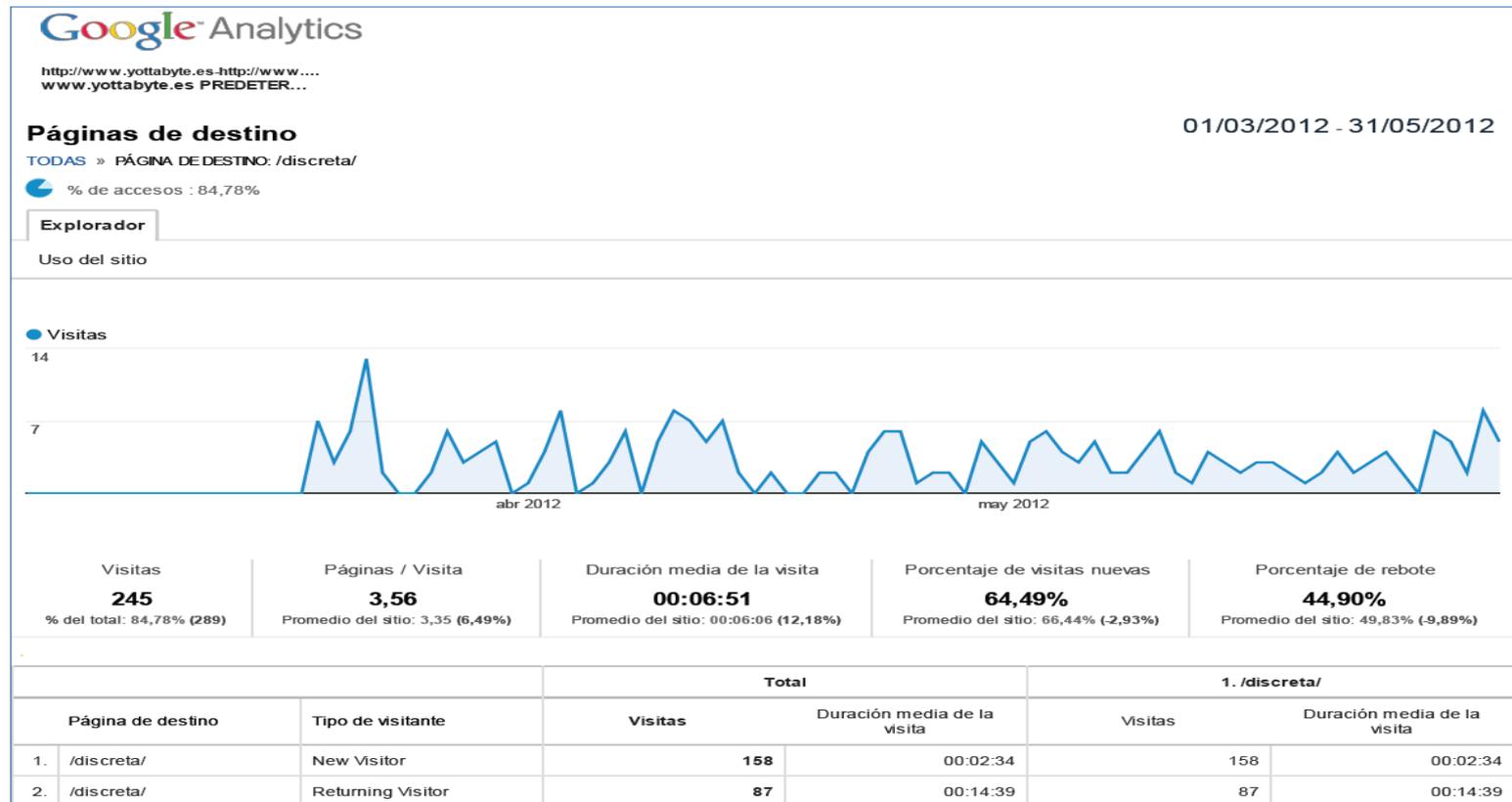
Id	Nombre de tarea	Duración	Comienzo	Fin	tri 1, 2012			tri 2, 2012			tri 3, 2012			tri 4, 2012			tri 1, 2013			tri 2, 2013			tri 3, 2013			tri 4, 2013						
					ene	feb	mar	abr	may	jun	jul	ago	sep	oct	nov	dic	ene	feb	mar	abr	may	jun	jul	ago	sep	oct						
1	INICIO PROYECTO	0 días	lun 09/01/12	lun 09/01/12																												
2	A.1 Planificación y especificación de requisitos	37 días	lun 09/01/12	mar 28/02/12	28/02																											
3	A.1.1 reunión con el cliente	1 día	lun 09/01/12	lun 09/01/12	Gustavo A. Iglesias Bello																											
4	A.1.1 definir plan borrador	2 días	mar 10/01/12	mié 11/01/12	Analista 2[50%];Analista 1[50%]																											
5	A.1.2 crear el informe de investigación preliminar	5 días	jue 12/01/12	mié 18/01/12	Analista 2;Analista 1[50%]																											
6	A.1.3 definir los requisitos	2 días	jue 19/01/12	vie 20/01/12	Analista 2[80%];Analista 1[20%]																											
7	A.1.4 registrar términos en el glosario	8 horas	lun 23/01/12	lun 23/01/12	Analista 1																											
8	A.1.5 implementar prototipo	1,5 días	lun 27/02/12	mar 28/02/12	Programador 1																											
9	A.1.6 definir casos de uso	6 días	lun 23/01/12	lun 30/01/12	Analista 2[50%]																											
10	A.1.7 definir el modelo conceptual - borrador	6 días	mar 31/01/12	mar 07/02/12	Analista 2[50%]																											
11	A.1.8 definir la arquitectura del sistema - borrador	6 días	mar 14/02/12	mar 21/02/12	Analista 1[50%]																											
12	A.1.9 firma de contrato con el cliente	1 día	mié 22/02/12	mié 22/02/12	Gustavo A. Iglesias Bello																											
13	A.1.10 refinar el plan	2 días	jue 23/02/12	vie 24/02/12	Analista 2[50%];Analista 1[50%]																											
14	A.2 Construcción	108,36 días	lun 20/02/12	vie 27/07/12																												
15	A.2.1 Análisis	6 días	lun 27/02/12	lun 05/03/12																												
16	A.2.1.1 definir casos de uso en formato expandido	1 día	lun 27/02/12	lun 27/02/12	Analista 2[50%];Analista 1[50%]																											
17	A.2.1.2 refinar los diagramas de casos de uso	2 días	mar 28/02/12	mié 29/02/12	Analista 2																											
18	A.2.1.3 refinar el modelo conceptual	1,5 días	jue 01/03/12	vie 02/03/12	Analista 1																											
19	A.2.1.5 definir los diagramas de secuencia del sistema	1,5 días	vie 02/03/12	lun 05/03/12	Analista 2																											
20	A.2.2 Diseño	78 días	lun 20/02/12	jue 14/06/12																												
21	A.2.2.1 definir los casos de uso reales	1,5 días	jue 01/03/12	vie 02/03/12	Diseñador 3;Diseñador 2																											
22	A.2.2.2 definir informes e interfaz de usuario	4,29 días	vie 02/03/12	jue 08/03/12	Diseñador 3[80%];Diseñador 4[80%];Diseñador 1[70%]																											
23	A.2.2.3 refinar la arquitectura del sistema	3,13 días	jue 08/03/12	mar 13/03/12	Diseñador 1[70%];Diseñador 3;Diseñador 4																											
24	A.2.2.5 definir el diagrama de clases de diseño	4,44 días	mar 13/03/12	mié 21/03/12	Diseñador 1[60%];Diseñador 3;Diseñador 2[20%]																											
25	A.2.3 Implementación	87 días	mié 21/03/12	vie 27/07/12	Programador 1;Programador 2[50%];Programador 3[50%];Programador 6[50%];Programador 5																											
26	A.2.4 Pruebas	79 días	mié 21/03/12	mar 17/07/12																												
27	A.2.4.1 Describir el plan de pruebas	6 días	mié 21/03/12	jue 29/03/12	Calidad 1;Programador 5[50%];Programador 3[50%]																											
28	A.2.4.2 Realizar pruebas de unitarias	14 días	jue 29/03/12	jue 05/07/12	Programador 3[50%];Programador 2[50%];Programador 6[50%]																											
29	A.2.4.3 Realizar pruebas de integración	8 días	jue 05/07/12	mar 17/07/12	Programador 3;Programador 2;Programador 1																											
30	A.3 Gestión	130 días	lun 23/01/12	lun 30/07/12																												
31	A.3.1 Estimación del Proyecto	6 días	lun 23/01/12	lun 30/01/12	Gustavo A. Iglesias Bello[50%]																											
32	A.3.2 Organización del proyecto	6 días	lun 23/01/12	lun 30/01/12	Gustavo A. Iglesias Bello[50%];Calidad 1[20%]																											
33	A.3.3 Planificación del Proyecto	6 días	mar 31/01/12	mar 07/02/12	Gustavo A. Iglesias Bello[80%];Analista 1[50%];Diseñador 2[20%];Calidad 1[50%];Programador 7[50%]																											
34	A.3.4 Seguimiento del Proyecto	129 días	lun 23/01/12	vie 27/07/12																												
35	A.3.4 Seguimiento del Proyecto 1	1 día	lun 23/01/12	lun 23/01/12																												
36	A.3.4 Seguimiento del Proyecto 2	1 día	vie 10/02/12	vie 10/02/12																												
37	A.3.4 Seguimiento del Proyecto 3	1 día	vie 02/03/12	vie 02/03/12																												
38	A.3.4 Seguimiento del Proyecto 4	1 día	vie 23/03/12	vie 23/03/12																												
39	A.3.4 Seguimiento del Proyecto 5	1 día	vie 13/04/12	vie 13/04/12																												
40	A.3.4 Seguimiento del Proyecto 6	1 día	vie 04/05/12	vie 04/05/12																												
41	A.3.4 Seguimiento del Proyecto 7	1 día	vie 25/05/12	vie 25/05/12																												
42	A.3.4 Seguimiento del Proyecto 8	1 día	vie 15/06/12	vie 15/06/12																												
43	A.3.4 Seguimiento del Proyecto 9	1 día	vie 06/07/12	vie 06/07/12																												
44	A.3.4 Seguimiento del Proyecto 10	1 día	vie 27/07/12	vie 27/07/12																												
45	A.3.5 Proceso de Creación de Líneas Futuras	10 días	mar 17/07/12	lun 30/07/12																												
46	A.3.5.1 Definición del Plan de futuro	5 días	mar 17/07/12	lun 23/07/12	Gustavo A. Iglesias Bello																											
47	A.3.6 Plan de Futuro	7,83 días	mar 17/07/12	vie 27/07/12	Gustavo A. Iglesias Bello[50%];Calidad 1[50%];Analista 1[16%];Programador 5[30%];Diseñador																											
48	A.4 Instalación	1 día	vie 27/07/12	lun 30/07/12																												
49	Fin PROYECTO	0 días	lun 30/07/12	lun 30/07/12	30/07																											



Conclusiones líneas de trabajo

➤ Conclusiones

- GrafoMin ha obtenido una respuesta aceptable por parte de la comunidad
 - Probada en internet con 245 accesos: (http://es.wikipedia.org/wiki/Algoritmo_de_Dijkstra) [1/3/2012–31/5/2012]
 - el 185 son nuevos visitantes
 - 87 son visitantes que repiten la experiencia





Conclusiones y líneas de trabajo

➤ Líneas futuras de trabajo

➤ Uso de la variable *globales.debug*

- Muestra los datos de variables durante la depuración.

➤ Grafos dirigidos con aristas opuestas entre dos vértices adyacentes

- la herramienta está lista para ello, solo falta la implementación gráfica.

➤ Compatibilidad para Microsoft Internet Explorer 8®

➤ Gestión de ejercicios

- Guardar un ejercicio
- Recuperar un ejercicio

➤ Seguimiento analítico

- Descripción paso a paso de cada iteración del algoritmo de Dijkstra
- Evaluar al usuario paso a paso de los valores de la matriz analítica en cada iteración del algoritmo de Dijkstra