



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA  
Escuela de Ingeniería Informática



# Gráfico interactivo de las relaciones derivativas entre palabras del español

Trabajo de fin de grado

28/11/2013

Grado en ingeniería informática

Enrique Segura Quintana

Tutor: Francisco Carreras Riudavets



*Trabajo fin de grado en Ingeniería Informática  
(especialización en Tecnología de la Información)  
de la universidad de Las Palmas de Gran Canaria,  
presentado por el alumno:*

*Enrique Segura Quintana*

***Título del trabajo:***

*Gráfico interactivo de las relaciones derivativas  
entre palabras del español*

***Tutor:***

*Francisco Carreras Riudavets*



“A todos los que me han apoyado y ayudado en esta etapa.

Gracias.”



# Agradecimientos

---

A todas las personas que me han apoyado, animado y aguantado. Quiero agradecerles la ayuda y consejos que me han dado mientras realizaba este proyecto.

La primera persona que debo mencionar es mi tutor Francisco Carreras Riudavets, quién siempre ha estado para resolverme dudas, aconsejarme y ayudarme en todo lo necesario. Gracias a él y a todos los profesores que me han formado, para conseguir esta meta.

A mis padres y resto de familia, que me han ayudado en todo lo que han podido y me han alentado a superar todos los retos encontrado.

Por último quisiera agradecer a mis compañeros y amigos el apoyo y los consejos prácticos.

Muchas gracias a todos.





# Resumen

---

En el primer apartado, como explicación del **estado actual** y los **objetivos** que se persiguen, se describen los temas que se han tenido en cuenta a la hora de desarrollar este proyecto, como son los antecedentes de dicha aplicación; las diferentes opciones de lenguajes de programación, que se manejaron, para llevar a cabo la aplicación; los servicios web, que surten a la aplicación de los datos necesarios; y finalmente el procesamiento gráfico de la aplicación que es la base y objetivo de este proyecto.

En el segundo apartado, como guía, se procede a definir específicamente el proyecto mediante las **competencias** que se han cubierto.

En el tercer apartado se describirán las **aportaciones** socio-económicas y técnicas al entorno general, además de comentar el estado actual de las aplicaciones dedicadas a las relaciones morfológicas.

En el siguiente apartado se repasa la normativa vigente, sobre aplicaciones de este tipo, que afectan o pueden afectar directamente al proyecto.

En el quinto apartado, se realizará la fase de **análisis**, donde se describen el formato de las tablas Hash de los que la aplicación obtiene los datos y la forma de acceder a las mismas mediante servicios web, además de la explicación y definición de un servicio web. También se definen los casos de uso del software resultante.

Una vez hecho el análisis se procede a la fase **diseño**, donde aparte de incluir los diagramas de clase y de actividades, se describe el diseño de la interfaz gráfica.

Antes de comenzar con la implementación se realiza el análisis de requisitos de software y usuario, no será necesario hacer un análisis de requisitos de hardware ya que se trata de una aplicación web.

A continuación, se describe la fase de **desarrollo**, indicando cómo se ha implementado la aplicación software de este proyecto señalando las librerías usadas y los módulos software generado.

En los siguientes apartados se podrán ver los **resultados** obtenidos tras la implementación, prueba y verificación del software y el presupuesto estimado para la realización de este proyecto.

Y finalmente se presentarán las conclusiones obtenidas, el posible trabajo futuro y la bibliografía utilizada.

# Abstract

---

In the first place, we explain the **actual state** and the looking for **objectives**, the themes that have been important in the design of this Project like the predecessor application, the different options of language programs that been studied to developer this application; the web services and finally the graphic design that will be basic and the main objective in this project.

In the second chapter it been defined a guide with the vocabulary and competences that have been covered with the execution of the project.

In the third chapter it will describe the social and economic contributions and the technical contributions to the general environment. Also similar applications are studied.

In the follow chapter the law about this type of applications is studied that they could affect to this project.

In the fifth chapter, the analysis phase it has been executed, where the Data Base kind was described and we defined the use cases for the resultant software.

When the analysis has finished, I have to include the class's diagrams and activities and it was described the graphic interface design.

Before to begin with the implementation I have to do the analysis of software requirements and user requirements. The hardware requirements are not needed because this is an application web.

The next step is to describe the developer phase where I describe the software implementation in this Project and the libraries used.

The last chapter agglutinates the results of testing and development and the estimated budget.

And finally the conclusions, the possible future Projects and bibliographic will be showed.



## Indice

1. Estado actual y objetivos .....	16
1.1 Estado actual.....	16
1.2 Objetivos.....	17
1.3 Estructura del documento .....	17
2. Competencias a cubrir.....	19
2.1.- CII01.....	19
2.2.- CII02.....	19
2.3.- CII04.....	20
2.4.- CII08.....	20
2.5.- TFG01.....	20
3. Aportaciones.....	21
4. Normativa y legislación .....	22
5. Metodología y plan de trabajo .....	23
5.1. – Metodología de desarrollo.....	23
5.1.1 - Análisis .....	24
5.1.2 - Diseño .....	24
5.1.3 - Implementación .....	24
5.1.4 - Prueba.....	24
5.1.5 - Mantenimiento .....	24
5.2. – Plan de trabajo .....	24
5.2.1 - Visión global del sistema.....	24
5.2.2 - Selección de herramientas.....	24
5.2.3 - Análisis y diseño del sistema.....	24
5.2.4 - Implementación del sistema .....	25
5.2.5 - Pruebas .....	25
5.2.6 -Documentación y defensa .....	25
6. Análisis.....	26
6.1. – Estudio de la aplicación antecesora .....	26
6.2. – Definición de requisitos .....	28
6.3. – Roles .....	28
6.4. – Diccionario de conceptos.....	29
6.5. – Análisis de requisitos .....	29
6.5.1 – Servicio Lematizador.....	30

6.5.2	- Servicio Familias.....	31
6.6	– Casos de palabras .....	31
6.6.1	– Parasintéticas .....	32
6.6.2	– Prefijales.....	32
6.6.3	– Sufijales .....	32
6.6.4	– Otras.....	32
6.7	– Información a almacenar .....	33
6.8	– Aplicación multilinguaje.....	34
7.	Requisitos de hardware y software .....	35
7.1	- Requisitos hardware.....	35
7.1.1	- Servidor Web.....	35
7.1.2	– Equipo de trabajo .....	35
7.2.	Requisitos software .....	36
7.2.1	- Editores .....	37
7.2.2	– Lenguajes de programación .....	38
7.2.3	- Librerías .....	42
7.2.4	– Entornos de desarrollo.....	43
7.2.5	– Control de versiones .....	45
7.2.6	– Recursos seleccionados.....	46
8.	Diseño.....	48
8.1	– Introducción .....	48
8.2	– Arquitectura Cliente-Servidor .....	48
8.3	– Servicios Web .....	49
8.3.1	– Servicio Lematizador.....	49
8.3.2	– Servicio Familias .....	50
8.3.3	– Diagrama de estados .....	51
8.4	– Aplicación multilinguaje.....	52
8.5	– Pantalla completa.....	53
9.	Desarrollo .....	54
9.1	– Librerías .....	54
9.1.1	– Arbor.js.....	54
9.1.2	– Json.NET .....	58
9.1.3	– Json .....	60
9.1.4	– NuGet .....	63

9.1.5 – AJAX Control Toolkit .....	63
9.2 – Conexión con los servicios Web .....	65
9.3 – Variables de sesión .....	66
9.4 – Post VS Get.....	68
9.5 – Interfaz.....	71
9.5 – Idioma.....	72
9.6 – Pantalla completa.....	74
10. Manual de usuario y software .....	76
Bibliografía .....	77

## Índice de figuras

Figura 1. Modelo de ciclo de vida en cascada .....	65
Figura 2. Portada Relaciones Morfológicas .....	65
Figura 3. Búsqueda Relaciones Morfológicas .....	65
Figura 4. Grafo Relaciones Morfológicas.....	65
Figura 5. Modelo Cliente/Servidor.....	37
Figura 6. Diagrama de estados .....	52
Figura 7. Pantalla normal .....	65
Figura 8. Pantalla completa.....	65
Figura 9. Aristas del grafo.....	57
Figura 10. Contenedor por pestañas .....	54
Figura 11. Conexión servicios web.....	65
Figura 12. Portada del sitio.....	71
Figura 13. Interfaz de consulta .....	71

# 1. Estado actual y objetivos

---

## 1.1 Estado actual

La aplicación actual realiza el procesamiento de formas relacionales morfológicas del lenguaje español. Dicha aplicación obtiene la familia de palabras relacionadas y muestra, al usuario, esta familia sea cual sea la categoría gramatical a la que pertenezca.

Dicha aplicación está desarrollada en lenguaje C++ y en el entorno desarrollado en Java, por tanto, aprovechando que se está actualizando el departamento de Informática y Sistemas de la ULPGC, también se están actualizando sus aplicaciones, implementación y alojamiento.

La aplicación actual se encuentra en el antiguo portal del departamento de Informática y Sistemas de la ULPGC siendo su dirección URL: <http://www.gedlc.ulpgc.es/investigacion/scogeme02/relmorfo.htm>

Esta aplicación está basada en la tesis doctoral de Francisco Carreras Riudavets, Profesor del Departamento de Informática y Sistemas de la Universidad de Las Palmas de Gran Canaria, la tesis se centra en desarrollar un sistema capaz de resolver y responder a cualquier aspecto morfológico de una palabra del español que abarca todo lo relacionado con la morfología flexiva, la derivativa y gran parte de la compositiva, de cualquier palabra del español, para el establecimiento de relaciones morfológicas. Permite el reconocimiento, la generación y la manipulación de las relaciones morfológicas a partir de cualquier palabra, así como el del campo morfosemántico al que pertenece, categoría gramatical de la base y de sus palabras relacionadas, incluye la recuperación de toda su información lexicogenética hasta llegar a una primitiva, la gestión y control de los afijos en el tratamiento de sus relaciones, así como la regularidad en la relación establecida y otros aspectos.

La información que contiene esta tesis se organiza, sobre un corpus suficientemente amplio, un estudio taxonómico, exhaustivo y sistemático de los afijos utilizados en las relaciones morfosemánticas que proporciona una visión global del comportamiento y productividad de las palabras del español en los principales procesos de formación —sufijación, prefijación, parasíntesis, supresión, regresión, modificación-cero, apócope, metátesis y otros no clasificables que generan grafías alternativas— y se establece una descripción pormenorizada de las relaciones entre palabra y afijo en el marco de la funcionalidad, formalidad y semántica.

Obteniéndose así un amplio almacén de datos fácilmente accesibles de forma ordenada que necesitan de una representación e interfaz gráfica más actualizada y adecuada a las necesidades de los usuarios de la misma.

Dado que la aplicación actual está desarrollada en lenguaje C++ y en el entorno desarrollado en Java, por tanto, aprovechando que se está actualizando el departamento de Informática y Sistemas de la ULPGC, también se están actualizando sus aplicaciones, implementación y alojamiento.



La aplicación actual se encuentra en el antiguo portal del departamento de Informática y Sistemas de la ULPGC siendo su dirección URL:

<http://www.gedlc.ulpgc.es/investigacion/scogeme02/relmorfo.htm>

La información que alimenta la aplicación está contenida en unas tablas hash, estructura de datos que asocian llaves o claves con valores. Dicha información está actualizada y tanto la aplicación como los datos están contenidos en un servidor con tecnología Windows Web Server R2 6.1.

El servidor Windows Web Server R2 6.1 funciona con el conjunto de servicios Internet Information Service 7.5 o IIS 7.5, el cual se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas.

Por tanto queda establecido que tanto la información de la base de datos como el servidor en el que se alojan la misma y la aplicación están convenientemente actualizadas, quedando como objetivo renovar la aplicación.

## 1.2 Objetivos

Los objetivos que se tratan de cubrir con este proyecto son, principalmente, crear una aplicación web amigable para el usuario, capaz de extraer la información morfológica de cada palabra que el usuario introduzca, tanto en forma de lista detallada como en una representación gráfica de dicha palabra en forma de árbol con toda la información ramificada a partir de la misma. De forma dinámica la información que se muestra podrá ser filtrada, por el usuario, en todo momento, mejorando así a la aplicación predecesora, la cual muestra la información en un modo más simple y no permite modificar la muestra de la misma en cualquier momento.

El objetivo de mejorar la interacción del usuario con la aplicación se ha concebido para que la información que aporta sea más concisa y facilitar la legibilidad por parte de los mismos, haciendo posible indagar en dicha información igualmente de forma interactiva y amigable, haciendo así más amena las consultas.

## 1.3 Estructura del documento

Con el objetivo de facilitar la lectura del presente documento se presenta en este apartado la estructura general del mismo.

Este documento se organiza en 10 capítulos:

- **Estado actual y objetivos:** repaso al estado inicial del aplicativo y exposición de las mejoras que se realizarán al mismo así como los hitos que se pretenden alcanzar en cada una de las etapas y apartados en los que está dividido el TFG.

- **Competencias:** se exponen las competencias cubiertas con este trabajo, indicando los apartados en las que éstas se ven satisfechas.
- **Aportaciones:** análisis de las contribuciones que el TFG aporta a nuestro entorno social, científico y técnico.
- **Normativa y legislación:** se citan las normativas y legislaciones que afectan al presente trabajo y al desarrollo web actual.
- **Metodología y plan de trabajo:** en este capítulo se expone la metodología seguida a la hora de desarrollar el TFG y la planificación llevada a cabo para cumplir objetivos.
- **Análisis:** se describen en detalle el análisis realizado para seleccionar las herramientas, en base a los recursos de los que se dispone, explicando su funcionamiento, ventajas que presentan y argumentando su elección.
- **Requisitos:** se especifican los requisitos tanto hardware como software asociados al proyecto.
- **Diseño:** todos los detalles relativos al diseño de la aplicación se encuentran en este capítulo.
- **Desarrollo:** todos los detalles relativos a los aspectos relacionados con el desarrollo de la aplicación se encuentran en este capítulo.
- **Anexos.**

En cada uno de los capítulos presentados en este documento se podrán encontrar diferentes apartados, organizados de manera que se facilite la lectura y comprensión de los mismos. A lo largo del documento también se citarán las referencias bibliográficas consultadas.

## 2. Competencias a cubrir

---

Este apartado se divide en cuatro sub-apartados independientes, CII01 donde se documentan: la capacidad para diseñar, desarrollar, evaluar y seleccionar sistemas informáticos y aplicaciones; evaluación de seguridad y calidad; comparando todo ello con la legislación y normativa vigente. El segundo apartado llamado CII02 documenta la capacidad de planificación, concesión, de despliegue y dirección de proyectos, servicios y sistemas informáticos en todos los ámbitos, valorando siempre su puesta en marcha y su impacto económico y social. El tercer apartado, CII04, plasma la capacidad de elaborar el pliego de condiciones técnicas de una instalación informática que cumpla la normativa vigente. En el último sub-apartado se detallan los conocimientos de la normativa y los conocimientos de la regulación de la informática en el ámbito nacional, europeo e intercontinental.

### 2.1.- CII01

La competencia CII01 abarca varios apartados de esta memoria y en ellos se especifican además las responsabilidades que se deben cubrir para la correcta ejecución de esta competencia.

En los apartados de “Estado Actual y Objetivos”, “Análisis”, “Requisitos tecnológicos”, “Diseño” y “Desarrollo”, se desarrolla esta competencia en profundidad.

Como estudiante, se deben cumplir una serie de responsabilidades:

#### *Análisis*

El estudiante debe inferir y analizar los requisitos para su posterior especificación y validación.

#### *Diseño*

El estudiante debe definir y diseñar la estructura de datos, la arquitectura y la interfaz de usuario.

#### *Implementación*

El estudiante debe aprender a usar las herramientas necesarias para el desarrollo del proyecto, investigar tecnologías externas que le puedan ayudar con el desarrollo y generar un código optimizado y de calidad.

### 2.2.- CII02

La competencia CII02 se entiende como la capacidad de concebir, diseñar y ejecutar un plan de trabajo para calcular y valorar los costes del proyecto así como su impacto económico y social.

La planificación es un elemento clave en la gestión de proyectos. Sin una adecuada planificación, el proyecto puede no satisfacer todos sus requisitos, lo que repercute en la calidad de la aplicación final y en los plazos de finalización.

Para el desarrollo de este proyecto, se ha realizado una planificación óptima gracias a la cual se ha finalizado el proyecto en los plazos establecidos, consiguiendo satisfacer todos los objetivos definidos inicialmente. La planificación cubre la competencia CII02, como se puede observar en el apartado “Metodología y plan de trabajo”. Asimismo, en el apartado “Aportaciones” se puede apreciar el impacto del trabajo en el entorno social y técnico.

Una especial mención al tutor por sus aportaciones para mejorar el proyecto durante su desarrollo.

### **2.3.- CII04**

La competencia CII04 engloba la *“capacidad para elaborar el pliego de condiciones técnicas de una instalación informática que cumpla los estándares y normativas vigentes”*.

Este proyecto no necesita cubrir esta competencia.

### **2.4.- CII08**

La competencia CII08 cubre el *“conocimiento de la normativa y la regulación de la informática en los ámbitos nacional, europeo e internacional”*.

Este proyecto no necesita cubrir esta competencia.

### **2.5.- TFG01**

La última competencia es la TFG01, que representa el ejercicio de realizar una presentación y defensa ante un tribunal compuesto por profesores de la comunidad universitaria de la Universidad de Las Palmas de Gran Canaria, que será designado por la Escuela de Ingeniería Informática de dicha universidad.

Esta última competencia es la suma de todos los apartados de este documento, llamado TFG, y se dará por satisfecha tras su presentación ante el tribunal correspondiente.

## 3. Aportaciones

---

La interfaz gráfica que proporciona los resultados visuales del sistema de resolución de aspectos morfológicos de las palabras del lenguaje español desarrollado en la tesis doctoral del profesor Francisco Carreras Riudavets, representa el punto de partida de este proyecto.

En apartados anteriores, se ha podido comprobar que la interfaz gráfica estaba obsoleta, antigua, muy simple y sin funcionalidades. Por lo que el desarrollo de este proyecto supone una optimización de la aplicación anterior de modo que proporcione al usuario una herramienta innovadora y completa que posibilite su interacción con la nueva interfaz.

Entre las nuevas funcionalidades destaca la visualización en forma de árbol de la familia de una palabra hasta dos subniveles. A partir del árbol formado, se pueden aplicar operaciones de filtro para visualizar diferentes partes de dicho árbol, y generar un nuevo árbol partiendo de alguna de las palabras de la familia. Todo esto se recoge en la aplicación web 'Familias TIP', objetivo de este proyecto.

A título personal, este TFG ha supuesto una oportunidad idónea para poner a prueba los conocimientos adquiridos durante estos años de estudio. La oportunidad de desarrollar un proyecto que va a estar de cara al público ha supuesto un reto ya que va a ser puesto a prueba en la realidad. La toma de decisiones a lo largo del desarrollo y la solución de los posibles imprevistos que pueden surgir en todo proyecto, me han proporcionado una gran cantidad de nuevos conocimientos y experiencia a la hora de utilizar tecnologías que a día de hoy son punteras en el mercado.

## 4. Normativa y legislación

---

En este apartado se estudia la normativa y legislación vigente sobre proyectos informáticos. A la aplicación de nuestro proyecto no le afecta ningún tipo de legislación, ya que no se requieren datos personales ni ningún tipo de información sensible que debamos considerar.

En cambio, para el desarrollo tenemos que tener en cuenta las licencias de las aplicaciones y librerías que utilizamos, tal y como se muestra a continuación:

- **Licencia MIT**

La licencia MIT es una de las muchas licencias de software que ha empleado el Instituto Tecnológico de Massachusetts (MIT). Es un licencia de software libre-permisiva que permite la reutilización y modificación del software propietario siempre que todas las copias del software incluyan una copia de los términos de la licencia MIT.

Las librerías **Arbor.js** y **Json.NET** hacen uso de este tipo de licencia.

- **Licencia MSDN**

MSDN (del inglés, *Microsoft Developer Network*) se refiere a los servicios web (basados en plataformas Microsoft) orientados a desarrolladores de software y al software incluido con sus compiladores (Visual Studio) y ciertos SDK. Tal y como Microsoft lo define, "contiene una gran cantidad de información técnica de programación, incluidos código de ejemplo, documentación, artículos técnicos y guías de referencia".

De forma histórica, MSDN pone a disposición del público un paquete de suscripción según el cual los desarrolladores tienen acceso y licencias de uso de casi todo el software de Microsoft.

**Microsoft Office, Windows Server, Visual Studio 2010 y ASP.NET-C#** hacen uso de este tipo de licencia

# 5. Metodología y plan de trabajo

## 5.1. - Metodología de desarrollo

En el desarrollo de cualquier proyecto de desarrollo de software es fundamental el uso de un método de desarrollo sobre el que trabajar para considerar todos los aspectos críticos de los que depende el éxito de dicho proyecto, ya que éste debe generar un producto eficiente y óptimo que cumpla con los requisitos definidos y esté acabado dentro de los plazos de entrega acordados.

La metodología de desarrollo permite definir, planificar y controlar el desarrollo de cualquier proyecto, según su envergadura y el tiempo de entrega, entre otros.

Para desarrollar el software de procesamiento estadístico de los telegramas de comunicación, hemos usado como metodología de desarrollo software el modelo de ciclo de vida en cascada, que es uno de los más utilizados en la actualidad por su eficacia y simplicidad.

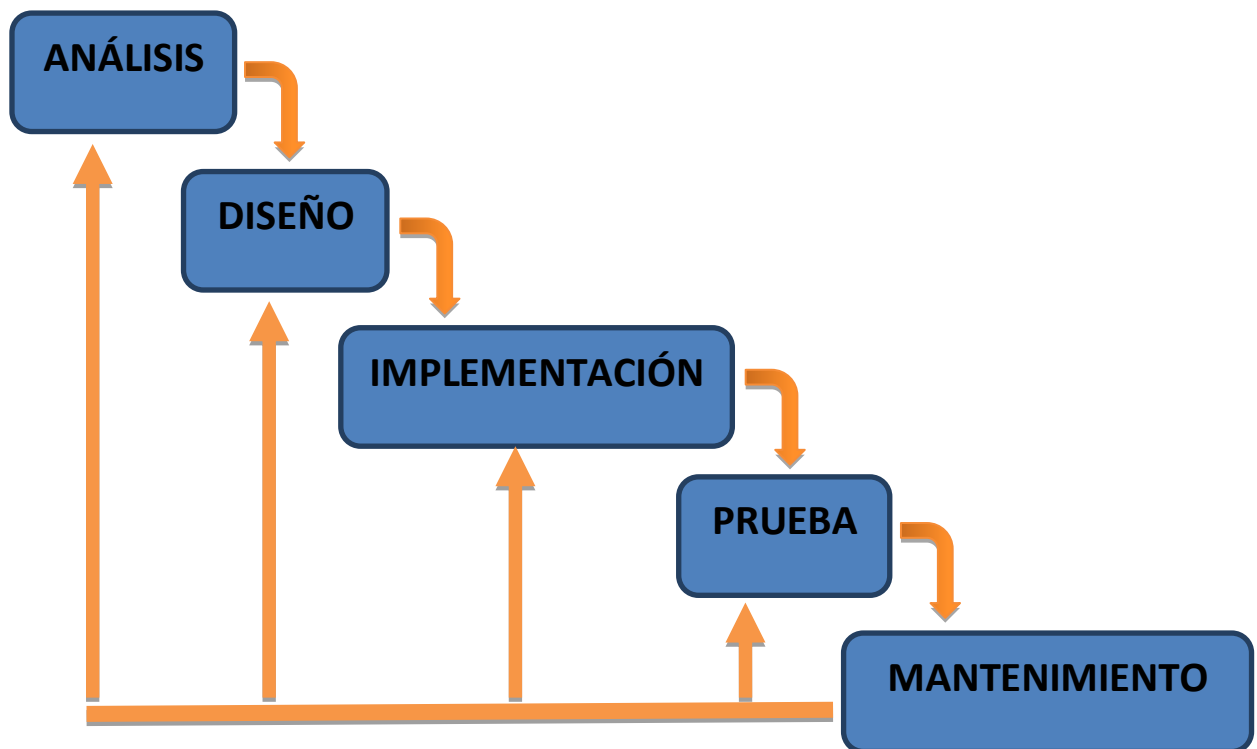


Figura 1. Modelo de ciclo de vida en cascada

Gracias a la retroalimentación en cada una de las etapas, podemos corregir errores detectados o realizar cambios para mejorar el proyecto en cualquiera de las fases.

El modelo de ciclo de vida se compone de las siguientes etapas:

#### **5.1.1 - Análisis**

En esta fase se analizan las necesidades de los usuarios finales del software para definir los requisitos que debe cumplir el software.

#### **5.1.2 - Diseño**

Durante el diseño se descompone y organiza el sistema en elementos separados para obtener una descripción de la estructura global del sistema y la especificación de requisitos del sistema.

#### **5.1.3 - Implementación**

En esta fase se implementa el código fuente generando prototipos.

#### **5.1.4 - Prueba**

En la fase de pruebas se comprueba que el sistema funcione correctamente cumpliendo con todos los requisitos de usuario.

#### **5.1.5 - Mantenimiento**

En la última fase se mantiene y mejora el software para corregir errores o añadir nuevos requisitos.

### **5.2. – Plan de trabajo**

Durante el desarrollo de este TFG, se han cubierto las siguientes etapas:

#### **5.2.1 - Visión global del sistema**

Inicialmente, estudiamos la documentación y la aplicación ya desarrollada por el profesor de la ULPGC Juan Carlos Rodríguez del Pino, con el objetivo de mejorar su interfaz y añadir más funcionalidades.

#### **5.2.2 - Selección de herramientas**

Instalaremos las herramientas software necesaria para desarrollar el TFG.

#### **5.2.3 - Análisis y diseño del sistema**

Se idearán los nuevos módulos que compondrán la aplicación. También se decidirá qué herramientas utilizar, se inspeccionarán las dependencias existentes entre módulos y los diferentes recursos de los que se compondrá el sistema.



#### **5.2.4 - Implementación del sistema**

En esta etapa implementamos el código según análisis y diseño planteados, utilizando el software instalado.

#### **5.2.5 - Pruebas**

Para comprobar el sistema implementado y comprobar que cumple con los requisitos especificados durante el análisis, se realizaron diversas pruebas para asegurar la calidad del sistema.

#### **5.2.6 - Documentación y defensa**

A partir de toda la información obtenida a lo largo del desarrollo del TFG se ha confeccionado el presente documento de TFG y se prepara la defensa del mismo.

## 6. Análisis

---

El objetivo del análisis es el marcar la definición de los requisitos de usuario de la estructura y diseño de software para el proyecto además de marcar todos los pasos entre ambos hitos.

El análisis de requisitos permite especificar características del software como su función, datos, rendimiento y además las restricciones que debe cumplir el mismo.

### 6.1. - Estudio de la aplicación antecesora

Teniendo en cuenta que este proyecto consiste en la renovación de un proyecto anterior, se hace necesario realizar un estudio de la misma para establecer bases.

Conocer el funcionamiento de la aplicación que se ha de ampliar para poder empezar la planificación del nuevo proyecto, se vuelve indispensable. Durante las primeras reuniones con el tutor encargado del proyecto proporcionó la información necesaria sobre la aplicación anterior y durante esos primeros días se realizaron una serie de pruebas para conocer todas las características de su funcionamiento y prestaciones que ofrece al usuario.

Como ya se ha comentado en puntos anteriores, la aplicación antecesora está disponible en la anterior página del Grupo de Estructuras de Datos y Lingüística Computacional de la Universidad de Las Palmas de Gran Canaria, enlace: <http://www.gedlc.ulpgc.es/investigacion/scogeme02/relmorfo.htm>

Como muestra la *figura X.X*, la portada de esta aplicación consta de un formulario donde se permite introducir una palabra del lenguaje natural a relacionar y varias condiciones sobre la derivación de esta palabra – Sustantivos, Adjetivos, Verbos, Adverbios u Otras-, Formas relacionadas de esta palabra – Forma primitiva, Formas cercanas, Formas derivadas (sufijales, prefijales, parasintéticas y otras) o Todas las formas relacionadas – Y la Regularidad – Regulares, Irregulares o Todas – de la misma palabra.



Figura 2. Portada Relaciones morfológicas

En esta aplicación una vez que el usuario introduce la palabra que quiera relacionar y ha seleccionado las condiciones de la búsqueda para que le sean devueltos los datos que necesita, la aplicación le devuelve los datos en forma de lista ordenada con la posibilidad de ver los datos de la misma en forma gráfica como muestra la *Figura X.X*.



Figura 3. Búsqueda relaciones morfológicas

Esta información, aunque útil y bien ordenada, resulta bastante tediosa para el usuario a la hora de establecer la relación que existe entre la familia de palabras solicitada. Este es uno de los puntos a tratar y mejorar en la nueva aplicación.

Otro de los puntos a mejorar es la representación gráfica de los datos, mostrando en ella las relaciones directas entre palabras y su clasificación, ya que en la aplicación anterior

esta representación esta echa en forma estática y no permite al usuario interacción alguna ni una legibilidad amena. La *Figura X.X* muestra la forma en la que la aplicación devuelve el grafo resultante de la forma derivada *Casar* a partir de la forma canónica *Casa*.

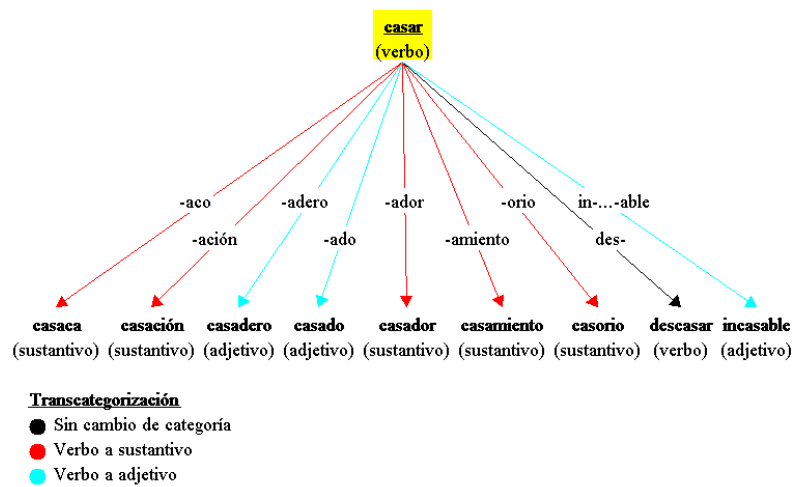


Figura 4. Grafo Relaciones morfológicas

## 6.2. – Definición de requisitos

Para definir los requisitos que este proyecto demanda se han llevado a cabo las fases de implementación necesarias en la documentación de los mismos y la forma más eficiente de cubrirlos.

- **Obtener los requisitos:** A partir de una reunión inicial con el tutor de este proyecto y responsable del anterior, se ha recopilado la información necesaria desde el punto de vista del desarrollo del proyecto.
- **Analizar requisitos:** Mediante el uso de la aplicación antecesora como usuario cualquiera se han establecido los casos de uso y se han delimitado además los privilegios de usuarios. Durante esta fase también se han reconocido nuevos puntos a mejorar en el futuro aplicativo.
- **Documentar requisitos:** Estos requisitos han sido documentados desde la primera reunión habiéndolos añadido a los ya facilitados por el tutor que fueron requisitos a cubrir en el primer proyecto y pueden volver a repetirse en este desarrollo.
- **Verificar y validar lo requisitos:** En las siguientes reuniones con el tutor se han verificado que los requisitos demandados han sido establecidos correctamente y que cumplen con las preferencias iniciales del proyecto cuando fue marcado.

## 6.3. – Roles

Un rol representa un elemento que se comunica con el sistema o aplicación siendo externo al mismo y teniendo, al menos, una meta al realizar dicha interacción.

Al tratarse de una aplicación de consulta a la que se puede acceder si estar autenticado de forma alguna ni solicita datos de ningún tipo, la definición de los roles resulta bastante sencilla. Se pueden definir, por tanto, al usuario. Cuyo rol y funciones quedan definidos como: Se trata de un usuario del sistema que accede al mismo sin estar identificado y puede realizar cualquier consulta sobre una palabra al sistema.

## 6.4. – Diccionario de conceptos

Además de definir los roles participante en esta aplicación deben quedar bien definidos una serie de conceptos que serán utilizados continuamente por la aplicación.

- **Forma canónica:** como todo vocablo con identidad propia susceptible de aplicársele o de habersele aplicado en su formación algún mecanismo de derivación. Ejemplo: *niña*, *niños*, *niñas*, *niñitos*, *niñitas*, etc. tienen como forma canónica a *niño*.
- **Categoría gramatical:** modo de distribución, en esta aplicación se dividen en sustantivos, adjetivos, verbos, adverbios y formas canónicas con otras categorías.
- **Tipo de familia:** en este proyecto se divide en derivaciones prefijales – prefijos, derivaciones sufijales – sufijos, derivaciones parasintéticas – combinación de prefijos y sufijos- y otras derivaciones – supresión, aféresis, alternativa, apócope, metátesis, terminación irregular, tiempo verbal y transcategorización.
- **Lematizador:** Servicio web que comprueba la existencia de una palabra del lenguaje castellano y devuelve su forma canónica.

## 6.5. – Análisis de requisitos

Con este proyecto se persigue mejorar la interfaz y legibilidad de la información, facilitar el acceso al servicio que presta la misma y añadir la interoperabilidad del usuario con el servicio de Familias TIP, nombre con que será denominado el proyecto y la aplicación. El significado de las siglas TIP – Text & Information Processing, viene dado por el nombre del grupo de investigación creado por los profesores del Departamento de Informática y Sistemas de la ULPGC.

Familias TIP, funcionará mediante una interfaz gráfica muy sencilla, en la que se verán una casilla para introducir una palabra y un botón llamado “*Relaciones*” donde el usuario introducirá una palabra y al hacer click en el botón la aplicación devolverá, en la misma ventana, bajo los componentes de la interfaz ya nombrados, una tabla con opciones en forma de Checkbox para filtrar las representaciones de familias de palabras en formas gráficas que se dividirán en las ventanas separadas por pestañas de la misma visual.

Hecha la introducción básica de lo que hará la aplicación ahora toca hacer un análisis más en profundidad del funcionamiento interno del proceso por el que para una palabra para mostrar su grafo de relaciones morfológicas.

### 6.5.1 – Servicio Lematizador

Para buscar la familia de palabras o su primitiva el usuario escribirá una palabra. Esta palabra deberá buscarse en el servicio de lematización para comprobar si es una palabra que existe y si está en su forma canónica. Los casos se tratan de la siguiente forma:

- Si es una palabra que no es reconocida por el lematizador se muestra el mensaje: "*palabra inexistente*"
- Si es una forma conjugada o flexionada (*niñito*) y no es una forma canónica (*niño*) se muestra la familia de palabras de la forma canónica *niño*.
- Si es una forma canónica se muestra la familia de palabras.

Para reconocer una palabra usando el servicio de lematización se debe invocar al siguiente método del Servicio.

El método la que se invoca es **Reconocer** mediante la llamada al servicio: <http://tip.dis.ulpgc.es/lematizador/Lematizador.aspx>. Este método se invoca con tres parámetros:

- Palabra
- Idioma
- multiPrefijos

Dando esta consulta como resultado un bloque con los siguientes parámetros:

- Prefijo de la palabra
- Clave de la forma canónica de la palabra
- Código de la flexión de la palabra
- Frecuencia de aparición
- Pronombre enclítico

Toda palabra tiene su forma canónica que no es más que su representante en la lengua, así, *perra*, *perros*, *perras*, *perritos*, *perritas*, etc. tienen como forma canónica a *perro*. Para el caso de los verbos, cualquier forma conjugada tiene su infinitivo como forma canónica.

El lematizador puede dar varias respuestas pues, por ejemplo, la forma *fui* puede ser una forma conjugada del verbo *ir* o del verbo *ser*.

El servicio lematizador también tiene otros métodos para averiguar la forma canónica, el texto de la flexión o el texto de la categoría gramatical se debe llamar al método **Consulta** del mismo **Servicio Lematizador**, escribiendo en el tercer parámetro "canónicas" o "flexiones" o "categorías" según sea el caso. En este caso únicamente usaremos el método **Consulta** ya que para la empresa de mostrar las familias de palabras este método nos es suficiente.

El servicio comentado será el mismo que en la llamada a la función **Reconocer**, <http://tip.dis.ulpgc.es/lematizador/Lematizador.aspx>, en este caso el método **Consulta** se invoca con tres parámetros:

- Clave o código
- Idioma

- nombre de la consulta

De la respuesta a la Consulta de canónicas se utilizan los siguientes campos:

- Forma canónica (*implementar*)
- Código de categoría gramatical ( Consulta(código,"es", "categorias"); )
- Código de la flexión de la forma canónica (1 para el infinitivo)
- Frecuencia de aparición
- Código de la categoría con género y número

### 6.5.2 - Servicio Familias

Una vez tenemos la forma canónica de la palabra usamos el Servicio de Familias que funciona igual que los anteriores. El método la que se invoca es **Consulta** mediante la llamada al servicio <http://tip.dis.ulpgc.es/ServicioFamilias/Familias.aspx>. Este método se invoca con tres parámetros:

- Clave o código
- Idioma
- nombre de la consulta

En función de la información que se necesite se deberá acceder a los campos de los parámetros que dan como resultado este método:

Respuesta a la Consulta ("forma canónica", "es", "**familias**"):

- Categoría de la forma canónica del primer parámetro.
- Código de la familia a la que pertenece.
- Forma canónica derivada de esta familia.
- Categoría de la forma canónica derivada de esta familia.
- Código del sufijo.
- Sufijo.
- 1 si la unión del sufijo ha sido regular y cero si ha sido irregular.
- Código del prefijo.
- Prefijo.
- 1 si la unión del prefijo ha sido regular y cero si ha sido irregular.

Este método sirve para saber las palabras (formas canónicas) que pertenecen a la familia.

## 6.6 – Casos de palabras

Como se ha explicado en sub-apartados anteriores la aplicación trabaja relacionando las palabras que han sufrido una derivación partir de su forma inicial y han mantenido su núcleo añadiendo a su cuerpo derivaciones prefijales, sufijales, ambas u otras ya clasificadas.

### 6.6.1 – Parasintéticas

Este tipo de derivación ocurre cuando la palabra ha añadido a su cuerpo tanto sufijo como prefijo como muestra el siguiente ejemplo:



**Automovilista:** (prefijo) auto-, (núcleo o palabra) -movil-, (Sufijo) -ista.

### 6.6.2 – Prefijales

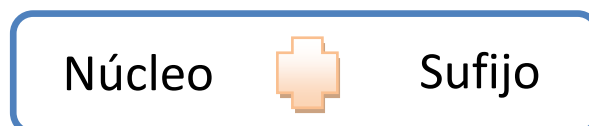
Derivaciones echas a partir de la adición de un prefijo al núcleo de la palabra como se muestra a continuación:



**Desconcertar:** (prefijo) des-, (núcleo o palabra) -concertar.

### 6.6.3 – Sufijales

Derivaciones nacidas de agregar al núcleo de la palabra un sufijo a continuación:



**Cochera:** (núcleo o palabra) coche-, (sufijo) -ra.

### 6.6.4 – Otras

Derivaciones nacidas, tanto, de agregar al núcleo de la palabra un sufijo o de agregar un prefijo al núcleo, pero que por sus características u orígenes están clasificadas como aféresis, metátesis, síncoas, apócopes y algunos cambios no clasificados.

En la siguiente tabla, se recogen todas las derivaciones que se tratan en el proyecto y la etiqueta que se les ha dado para mejorar la visualización de la misma en la representación del grafo:



SUFIJO
transcategorización/ trans.
aféresis/ afer.
alternativa / alt.
apócope / apóc.
alternativa/ alt.
terminación irregular/ irreg.
metátesis / met.
supresión / supr.
tiempo verbal / verb.

Un ejemplo de palabra que se corresponde con este caso es el siguiente:

**Incubar** posee una acepción, entre otras, equivalente al significado completo de **encobar** y las dos proceden etimológicamente del latín **incubare**.

## 6.7 – Información a almacenar

En vista de toda la información que devuelven todos los servicios utilizados y la necesitada para llevar a cabo el proyecto, hay que definir un tipo de estructura que almacene todos los datos ordenados, para que sean accesibles. Para realizar esta tarea hay que tener en cuenta la información necesaria de cada palabra que se necesitará.

La siguiente tabla muestra cómo se estructura dicha información y que tipo de dato contiene cada campo:

Campo	Contenido
<b>Característica del elemento</b>	Indican atributos como la forma del nodo que dependerá del nivel de profundidad que ocupe en el grafo, la tipografía de letras, el orden dentro del grafo y su color.
<b>Palabra</b>	El contenido del nodo, la palabra en sí que representa ese nodo.
<b>Sufijo</b>	Sufijo de la palabra, estará vacío no tiene sufijo.
<b>Prefijo</b>	Prefijo de la palabra, estará vacío no tiene prefijo.
<b>padre</b>	Indica el nodo padre, todos menos la raíz, tienen padre.
<b>Tipo de familia</b>	Indica el tipo de palabra que representa el nodo. Pueden ser sustantivo, Adjetivo, Adverbio o Verbo.
<b>Hijos</b>	Define los hijos que tiene un nodo en concreto.

## 6.8 – Aplicación multilinguaje

Un punto a tener en cuenta en el análisis de la aplicación será darle soporte multilinguaje, como mínimo en español – lenguaje por defecto e inglés para facilitar el acceso a la aplicación a usuarios de habla no castellana.

Para llevar a cabo este tipo de tarea los lenguajes Asp.Net y Visual C# disponen de almacenamiento llamados archivos de recursos, que permiten almacenar el texto multilinguaje de las aplicaciones en tablas – compuestas por parejas de datos clave o key y valor o value –las cuales pueden ser accedidas desde la interfaz por medio de un simple nomenclatura.

# 7. Requisitos de hardware y software

---

Durante el desarrollo de este trabajo de fin de grado se han utilizado una serie de recursos tecnológicos para cumplir con los objetivos del proyecto previamente definidos. Los recursos se clasifican en recursos hardware y software, respectivamente. La selección de cada recurso ha dependido de los siguientes ítems:

- Uso generalizado.
- Disposición de una amplia documentación.
- Soporte de lenguajes de programación web.
- Potencia y facilidad de uso de la herramienta.

En los siguientes apartados, analizamos algunas de las herramientas disponibles en el mercado, describiendo los puntos fuertes de cada una para finalmente seleccionar los recursos más adecuados para este proyecto, en base a los criterios previamente nombrados.

## 7.1 - Requisitos hardware

### 7.1.1 - Servidor Web

Esta aplicación web de este trabajo se alojará en el servidor de departamento de TIP. El servidor que utilizan es el **Windows Server 2008**, que es el nombre de un sistema operativo de Microsoft diseñado para servidores.

Windows Server 2008 se basa en el núcleo Windows NT 6.1., y entre las mejoras de esta versión destacan nuevas funcionalidades para el Active Directory, mejoras en el rendimiento de la virtualización y administración de sistemas, la inclusión de IIS 7.5, el soporte para más de 256 procesadores, un nuevo proceso de reparación de sistemas NTFS, la creación de sesiones de usuario en paralelo, la consola PowerShell y el núcleo del sistema renovado Server Core, entre otras muchas funcionalidad.

### 7.1.2 - Equipo de trabajo

Se ha usado un ordenador **portátil ACER Aspire 5542G** para el desarrollo de esta aplicación web, cuyas principales características son las siguientes:

Memoria	500GB HDD
Pantalla	15.6" HD LED LCD
Procesador y RAM	AMD Athlon™ II X2 Processor M320 (2.1GHz, 1 MB L2 CACHE) 6GB RAM DDR3
SO	Windows 7
Tarjeta gráfica	ATI Mobility Radeon™ HD4570

## 7.2. Requisitos software

Dada la amplia variedad de herramientas software que podemos encontrar en el mercado, nos hemos limitado a seleccionar y describir las herramientas más conocidas en cada campo para finalmente decidirnos por la más adecuada para nuestro trabajo valorando sus pros y sus contras.



Antes de seguir explicando las herramientas y las razones por las que se ha decantado la balanza hacia un lenguaje y un entorno de programación idóneos para este proyecto, es conveniente apuntar que la aplicación va a tener definida una estructura de programación cliente-servidor.

La **arquitectura cliente-servidor** es un modelo de aplicación distribuida en el que las tareas se reparten entre los servidores y los clientes. Diversas aplicaciones se ejecutan en un entorno cliente/servidor. Esto significa que los equipos clientes (equipos que forman parte de una red) contactan a un servidor, un equipo generalmente muy potente en materia de capacidad de entrada/salida, que proporciona servicios a los equipos clientes. Estos servicios son programas que proporcionan datos como la hora, archivos, una conexión, etc. El siguiente diagrama, figura X.X, muestra cómo funciona el sistema cliente/servidor mediante el protocolo TCP/IP.

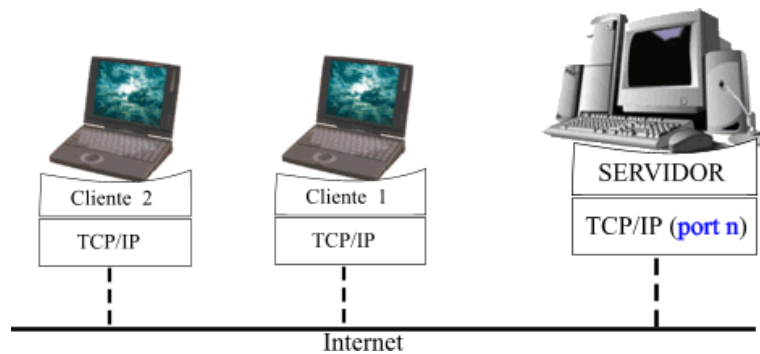


Figura 5 - Modelo Cliente/Servidor

El diagrama anterior muestra como el cliente envía una solicitud al servidor mediante su dirección IP y el puerto, que está reservado para un servicio en particular que se ejecuta en el servidor. Esta solicitud es recibida por el servidor y responde con la dirección IP del equipo cliente y su puerto.

A continuación, se detallan las herramientas más representativas de cada campo y se procede a seleccionar la más idónea:

## 7.2.1 - Editores

### 7.2.1.1- Microsoft Office Word 2010

Esta herramienta de edición de textos se ha utilizado a lo largo del desarrollo de la aplicación web para documentar las distintas fases del proyecto y elaborar un manual de ayuda de la aplicación destinado al usuario final.

Microsoft Office 2010 es una versión del paquete ofimático Microsoft Office de Microsoft y sucesora de Microsoft Office 2007. Ofrece compatibilidad extendida para diversos formatos de archivos, actualizaciones de la interfaz de usuario y una experiencia de usuario refinada.

Es compatible con los sistemas operativos Windows XP, Windows Vista y Windows 7. Además, el paquete está disponible en una compilación para arquitecturas de 64 bits, aunque sólo para los sistemas operativos de núcleo NT 6.x en adelante, como Windows Vista, Windows 7 y Windows Server 2008/2008.

### 7.2.1.2 - Notepad++

En diferentes puntos del desarrollo hemos utilizado el editor de texto Notepad++. Este programa es de código libre con soporte para una gran cantidad de lenguajes de programación y proporciona muchas facilidades para los programadores.

Es un editor de código totalmente gratuito que acepta todos los lenguajes contemplados para el desarrollo de esta aplicación, como por ejemplo HTML, XML, CSS y JavaScript.

Existen otros entornos como UltraEdit con similares características al Notepad++ en cuanto a prestaciones y lenguajes soportados, pero debido a que por costumbre ya

utilizábamos Notepad++ y cumple con nuestros requisitos, no hubo ninguna duda en su selección.

## 7.2.2 – Lenguajes de programación

### 7.2.2.1 - Java – J2EE

J2EE es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java. Permite utilizar arquitecturas de N capas distribuidas y se apoya ampliamente en componentes de software modulares ejecutándose sobre servidores de aplicación Java EE.

Entre las especificaciones API de este lenguaje encontramos:

- **JBDC:** API que permite la ejecución de operaciones sobre base de datos desde el lenguaje de aplicación Java. Es una colección de interfaces Java y métodos de gestión de manejadores de conexión hacia cada modelo específico de base de datos.
- **RMI:** mecanismo de Java que proporciona un mecanismo simple para la comunicación de servidores en aplicaciones distribuidas basadas exclusivamente en Java.
- **JavaMail:** es una expansión que facilita el envío y recepción de e-mail desde código Java.
- **Java Message Service** o **JMS:** es una API para crear, enviar, recibir y leer mensajes a los componentes de aplicaciones basados en la plataforma.
- **Servicios Web:** es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones, que pueden estar desarrolladas en distintos lenguajes de programación, y ejecutadas sobre cualquier plataforma.
- **XML:** es un lenguaje de marcado para almacenar datos en forma legible.

A parte de las características básicas de implementación y desarrollo de la API de Java, también se ha de tener en cuenta su potencial en cuanto a librerías gráficas para llevar a cabo parte del proyecto que nos ocupa. Para este tipo de servicios Java implementa la tecnología **Java Server Faces (JSF)**, que simplifica el desarrollo de interfaces de usuario en aplicaciones J2EE. Las librerías más conocidas y que más se ajustan a las características necesarias para este proyecto son:

- **RichFaces:** Agrega componentes visuales y soporte para AJAX, lenguaje en el que nos centraremos más adelante.
- **ICEfaces:** Contiene diversos componentes para interfaces de usuarios más enriquecidas, tales como editores de texto enriquecidos, reproductores de multimedia, entre otros.
- **jQuery4jsf:** Contiene diversos componentes sobre la base de uno de los más populares framework javascript jQuery.
- **PrimeFaces:** Es una librería muy liviana, todas las decisiones hechas son basadas en mantener a PrimeFaces lo más liviano posible, de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web. Está bajo la licencia de Apache License V2, además permite la integración con otros componentes como por ejemplo, el ya mencionado, RichFaces.

- **OpenFaces:** Librería open source que contiene diferentes componentes JSF, un Framework Ajax y un Framework de validación por parte del cliente.

#### 7.2.2.2 - PHP

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

Las características más reseñables de este lenguaje de programación que aportarían soluciones al desarrollo de esta aplicación son:

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito es invisible al navegador web y al cliente.
- Capacidad de conexión con la mayoría de los motores de base de datos.
- Capacidad de expandir su potencial utilizando módulos o extensiones.
- Posee una amplia documentación en su sitio web oficial.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones.
- Permite usar patrones de diseño como el Modelo Vista Controlador (MVC).
- Debido a su flexibilidad ha tenido una gran acogida como lenguaje base para las aplicaciones WEB de manejo de contenido, y es su uso principal.
- Puede ser ejecutado en la mayoría de sistemas operativos, entre ellos Microsoft Windows, requisito indispensable para este proyecto.

#### 7.2.2.3 - ASP.NET – Visual C#

ASP .NET es un framework para aplicaciones web, desarrollado y comercializado por Microsoft. Sus usos más frecuentes son sitios web dinámicos, aplicaciones web y servicios web XML. Está construido sobre el Common Language Runtime, el cual es un entorno de ejecución para códigos que se ejecutan sobre la plataforma Microsoft .NET y permite a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.

Entre sus características principales destacan: el modelo de programación *Code-behind*, el control del estado de la aplicación y de la sesión, y la construcción de las llamadas Páginas o

Web Forms, entre otros. Además, Asp.Net puede ser utilizado en aplicaciones cliente servidor, mediante el lenguaje C#.

A parte de las anteriores características, las extensiones más interesantes de este lenguaje que pueden ser útiles para desarrollar este proyecto donde se requiere una interfaz gráfica potente, son:

- **ASP.NET AJAX:** anteriormente llamado Atlas, es un conjunto de extensiones para ASP.NET desarrollado por Microsoft para implementar la funcionalidad de Ajax.
- **ASP.NET MVC Framework:** extensión de páginas ASP.NET que utiliza la arquitectura MVC.

Por otro lado, ASP.NET puede ser utilizado en aplicaciones cliente servidor, mediante el lenguaje C#, lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET.

Las características principales de este lenguaje son:

- Portabilidad del código fuente.
- Soporte para internacionalización.
- Adecuación para escribir aplicaciones de cualquier tamaño: desde las más grandes y sofisticadas como sistemas operativos hasta las más pequeñas funciones.
- Aplicaciones económicas en cuanto a memoria y procesado.
- Adecuación para escribir aplicaciones de cualquier tamaño.
- Inclusión de principios de ingeniería de software como revisión estricta de los tipos de datos y de límites de vectores, detección de usos de variables no inicializadas, y recolección de basura automática.
- Fácil migración del programador al nuevo lenguaje, especialmente para programadores familiarizados con C, C++ y Java.

#### **7.2.2.4 - Javascript**

JavaScript es un lenguaje de programación interpretado. Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web, ya que para interactuar con una página web, se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Las principales características de Javascript, que se ajustan al estándar ECMAScript, son:

- **Imperativo y estructurado:** JavaScript soporta gran parte de la estructura de programación de C (por ejemplo, sentencias if, bucles for, sentencias switch, etc.).
- **Dinámico:**



- **Tipado dinámico:** Como en la mayoría de lenguajes de scripting, el tipo está asociado al valor, no a la variable.
- **Objetual:** JavaScript está formado casi en su totalidad por objetos. Los objetos en JavaScript son arrays asociativos. Las propiedades y sus valores pueden ser creados, cambiados o eliminados en tiempo de ejecución.
- **Evaluación en tiempo de ejecución:** JavaScript incluye la función eval que permite evaluar expresiones como expresadas como cadenas en tiempo de ejecución.
- **Funcional:** Las funciones son objetos en sí mismos, y se puede anidar una función dentro de otra función definida. A diferencia de muchos lenguajes orientados a objetos, no hay distinción entre la definición de función y la definición de método.
- **Prototípico:**
  - **Prototipos:** JavaScript usa prototipos en vez de clases para el uso de herencia.
  - **Funciones como constructores de objetos:** Las funciones también se comportan como constructores.
- **Arrays y la definición literal de objetos:** Al igual que muchos lenguajes de script, arrays y objetos (arrays asociativos en otros idiomas) pueden ser creados con una sintaxis abreviada. De hecho, estos literales forman la base del formato de datos JSON.
- **Expresiones regulares:** JavaScript también soporta expresiones regulares de una manera similar a Perl, que proporcionan una sintaxis concisa y poderosa para la manipulación de texto.

#### 7.2.2.5 - HTML5

Lenguaje de marcado utilizado principalmente para la elaboración de páginas web. Permite escribir y describir el formato y contenido que dicha página tendrá al ser renderizada.

HTML5 es la quinta versión del lenguaje básico de marcado de toda página web, es decir, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML (XHTML).

HTML5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Algunos de ellos son técnicamente similares a las etiquetas <div> y <span>, pero tienen un significado semántico, como por ejemplo <nav> (bloque de navegación del sitio web) y <footer>. Otros elementos proporcionan nuevas funcionalidades a través de una interfaz estandarizada, como los elementos <audio> y <video>.

La mejora más significativa para nosotros es el elemento <canvas>, capaz de dibujar elementos 3D en los navegadores más importantes (Mozilla, Chrome, Opera, Safari e IE).

#### 7.2.2.6 - XML

XML es un lenguaje de marcas utilizado para almacenar datos en forma legible. Deriva del lenguaje SGML y permite definir la gramática de lenguajes específicos (como HTML) para

estructurar documentos grandes. Una característica que lo diferencia de otros lenguajes es que XML da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información.

#### 7.2.2.7 - JSON

JSON (acrónimo de JavaScript Object Notation) es un formato ligero para el intercambio de datos que permite escribir de manera más sencilla un analizador sintáctico de JSON. Gracias a su simplicidad, se ha generalizado su uso, especialmente como alternativa a XML en AJAX.

#### 7.2.2.8 - AJAX

AJAX, acrónimo de Asynchronous JavaScript And, es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (del inglés, Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta manera se pueden realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

Esta técnica es válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores ya que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

Ajax engloba las siguientes tecnologías:

- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño de las páginas.
- Document Object Model (DOM) accedido con JavaScript por parte del usuario para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos de forma asíncrona con el servidor web.
- Los formatos XML, texto plano o JSON, entre otros, son usados para la transferencia de datos solicitados al servidor.

### 7.2.3 - Librerías

#### 7.2.3.1 - JQuery

JQuery es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones (FLV) y agregar interacción con la técnica AJAX a páginas web. Ésta es la biblioteca más utilizada de Javascript. Es software libre y de código abierto, bajo licencia MIT y GNU v2 permitiendo su uso en proyectos libres y privados.

Sus características principales son:

- Selección de elementos DOM.
- Interactividad y modificaciones del árbol DOM, con soporte para CSS 1-3 y un plugin básico de XPath.
- Eventos.

- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones personalizadas.
- AJAX.
- Soporta extensiones.
- Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes, etc.
- Compatible con los navegadores más utilizados.

### **7.2.3.2 - Json.NET**

Json.NET es una librería de alto rendimiento para .NET, muy popular y que permite el tratamiento de archivos JSON. Entre sus ventajas destaca la conversión entre objetos .NET y JSON, y la fácil lectura y escritura de ficheros JSON. Además tiene soporte para .NET 4, que es la versión que se usa en este trabajo.

### **7.2.3.3 - Arbor.js**

Arbor.js es una librería de visualización gráfica construida con JQuery. En lugar de tratar de ser un framework global, Arbor.js proporciona un eficiente algoritmo y abstracciones para la organización gráfica en forma de grafo y la manipulación de los eventos de refresco de la pantalla. Deja a elección del programador el elemento gráfico a utilizar, lo que significa que se puede usar SVG, canvas o cualquier elemento posicionado de HTML. Se adapta a todo tipo de proyectos y sus necesidades de rendimiento.

### **7.2.3.4 - Ajax Control Toolkit**

El ASP.NET AJAX Control Toolkit es un proyecto de código abierto construido en la parte superior del framework de Microsoft ASP.NET AJAX. Proporciona una potente infraestructura de controles ASP.NET AJAX que son reutilizables y personalizables, así como un amplio abanico de controles que se pueden utilizar fuera de la caja para crear una experiencia web interactiva.

El kit de herramientas de control de AJAX contiene más de 30 controles que permiten crear fácilmente ricas páginas web interactivas.

## **7.2.4 - Entornos de desarrollo**

### **7.2.4.1 - Eclipse**

Eclipse es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido". Típicamente, esta plataforma ha sido usada para desarrollar entornos de desarrollo integrados (del inglés, IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).

Su arquitectura está basada en la plataforma de cliente enriquecido (del inglés, Rich Client Platform RCP). Los siguientes componentes constituyen la plataforma de cliente enriquecido:

- Plataforma principal - inicio de Eclipse, ejecución de plugins
- OSGi - una plataforma para bundling estándar.
- El Standard Widget Toolkit (SWT) - Un widget toolkit portable.
- JFace - manejo de archivos, manejo de texto, editores de texto
- El Workbench de Eclipse - vistas, editores, perspectivas, asistentes

El entorno de desarrollo integrado (IDE) de Eclipse emplea módulos (en inglés, *plugin*) para proporcionar toda su funcionalidad al frente de la plataforma de cliente enriquecido, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Asimismo, a través de los *plugins* libremente disponibles es posible añadir control de versiones con *Subversion* e integración con *Hibernate*.

El SDK de Eclipse incluye las herramientas de desarrollo de Java, ofreciendo un IDE con un compilador de Java interno y un modelo completo de los archivos fuente de Java. Esto permite técnicas avanzadas de refactorización y análisis de código. Mediante diversos *plugins* estas herramientas están también disponibles para otros lenguajes como C/C++ (Eclipse CDT) y en la medida de lo posible para lenguajes de script no tipados como PHP o Javascript. El IDE también hace uso de un espacio de trabajo, en este caso un grupo de metadato en un espacio para archivos plano, permitiendo modificaciones externas a los archivos en tanto se refresque el espacio de trabajo correspondiente.

Como características principales cabe destacar: un editor de texto con resaltado de sintaxis; compilación en tiempo real; disposición de pruebas unitarias con JUnit; control de versiones con CVS; integración con Ant; asistentes para creación de proyectos, clases, tests, etc., y refactorización.

#### 7.2.4.2 - NetBeans

NetBeans es un entorno de desarrollo integrado libre y gratuito sin restricciones de uso que permite desarrollar en una amplia variedad de lenguajes, aunque principalmente se utiliza para el lenguaje de programación Java. Posee además un número importante de módulos para extenderlo.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Todas las funciones del IDE son provistas por módulos. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, soporte para el sistema de control de versiones, un sistema de proyectos basado en Ant y refactoring.

Algunos de los posibles paquetes más interesantes que nos puede proporcionar NetBeans para el desarrollo del proyecto son:

- **NetBeans Enterprise pack:** Provee Soporte para la creación de aplicaciones orientadas a servicios (SOA), incluyendo herramientas de esquemas XML, un editor WSDL, y un editor BPEL para servicios web.
- **PHP:** NetBeans permite crear aplicaciones Web con PHP 5, un potente debugger integrado y además viene con soporte para Symfony. Además provee de soporte para AJAX.

Cabe destacar que NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permitiéndole al usuario comenzar a trabajar inmediatamente.

#### **7.2.4.3 - Microsoft Visual Studio 2010**

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, y Visual Basic .NET, al igual que entornos de desarrollo web como ASP.NET aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002).

La versión que se utilizará en este trabajo es Microsoft Visual Studio 2010, acompañada por .NET Framework 4.0

Además, cabe destacar que el compilador por defecto de C# está incluido en el entorno de desarrollo Visual Studio 2010 de Microsoft con su propio IDE, herramienta disponible, que será estudiada en este apartado.

### **7.2.5 – Control de versiones**

#### **7.2.5.1 - Team Foundation Service – Visual Studio Online**

Todo buen proyecto necesita llevar un control de versiones. Para este propósito, *Team Foundation Server* (TFS) proporciona gestión de código fuente (ya sea a través del control de versiones de Team Foundation o Git), elaboración de informes, gestión de requisitos, gestión de proyectos, gestión de laboratorio y pruebas, entre otras funcionalidades.

TFS cubre todo el proceso de desarrollo de software desde su inicio hasta su finalización, y puede ser integrado en numerosos entornos de desarrollo, pero donde proporcionan los máximos beneficios es en el Microsoft Visual Studio, ya que se trata de un producto de Microsoft.

### 7.2.6 – Recursos seleccionados

En apartados anteriores, se describieron los recursos software más conocidos, usados ampliamente por la comunidad y que podían ser de interés para el desarrollo de nuestro proyecto. Teniendo en cuenta las características de cada uno, se seleccionaron aquellas herramientas software que son las más adecuadas para nuestro proyecto.

En la siguiente tabla podemos ver el software que usamos para nuestro trabajo:

<b>SOFTWARE</b>	<b>RAZÓN DE SU SELECCIÓN</b>
<b>ASP.NET – Visual C#</b>	ASP.NET nos proporciona toda potencia y funcionalidad que buscamos en un framework junto con C#. Además, el tutor del proyecto nos puede asesorar perfectamente con este lenguaje, por lo que se selecciona como el lenguaje en el que vamos a codificar la aplicación web.
<b>Microsoft Visual Studio 2010</b>	Como el lenguaje que hemos seleccionado para codificar la aplicación es ASP.NET - Visual, Microsoft Visual Studio es el entorno idóneo, además de ser muy potente y gratuito.
<b>Team Foundation Service Visual Studio Online</b>	Para mantener un control de versiones de la aplicación, utilizamos esta herramienta que se integra perfectamente en el entorno de desarrollo Microsoft Visual Studio 2010 y para el cual el tutor del proyecto nos ha proporcionado una cuenta.
<b>Ajax Control Toolkit</b>	Al usar como lenguaje ASP.NET y el entorno de desarrollo Visual Studio 2010, seleccionamos esta librería para crear la interfaz gráfica.
<b>AJAX</b>	Al desarrollar una aplicación web y tener como requisito una interfaz gráfica potente, el uso de esta técnica de desarrollo web es indispensable.
<b>JavaScript</b>	Nuestra aplicación web demanda una capacidad gráfica bastante potente que permita la interacción con los usuarios, por lo que es básico que utilicemos este lenguaje del lado cliente.
<b>JQuery</b>	Al igual que con Javascript, por las posibilidades, potencialidad y pocas restricciones de JQuery, se hará uso de esta librería durante el desarrollo de este proyecto para agilizar y tratar los eventos de la interfaz gráfica de usuario.
<b>JSON</b>	Al tener que almacenar una gran cantidad de información, nos hemos decantado por la manera más sencilla de guardarla que es usando JSON.
<b>HTML5</b>	Como nuestro trabajo es desarrollar una aplicación, obviamente el lenguaje básico de la página web es HTML, y gracias a su evolución en HTML5 que proporciona grandes mejoras en la parte gráfica que uno de nuestros requisitos indispensables, trabajaremos con HTML5.

---

**Json.NET**

Al guardar la información en formato JSON, esta librería nos permite realizar conversiones de objetos JSON a ficheros y viceversa, por lo que es indispensable para el trabajo.

**Arbor.js**

Esta librería proporciona la construcción de grafos a partir de ficheros JSON, por lo que se convierte en una de las librerías más importantes de las que hacemos uso durante el desarrollo.

---

# 8. Diseño

---

## 8.1 – Introducción

El diseño del software se encuentra en el núcleo técnico de la ingeniería del software y se aplica independientemente del modelo de diseño del software que se utilice. Una vez que se analizan y especifican los requisitos del software, el diseño es la primera de las tres actividades técnicas – diseño, generación de código y pruebas – que se requieren para construir y verificar el software. Cada actividad transforma la información de manera que dé lugar, por último, a un software validado.

En general, la actividad del diseño se refiere al establecimiento de las estructuras de datos, la arquitectura general del software, las representaciones de la interfaz y los algoritmos. Por tanto, el diseño debe contemplar todos los requisitos explícitos obtenidos en la fase de análisis, debe ser una guía que puedan leer y entender los que construyen, prueban y mantienen el software, y debe proporcionar una idea completa, en este caso, de la aplicación.

## 8.2 – Arquitectura Cliente-Servidor

La arquitectura cliente-servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios y en el cliente permanece sólo lo particular de cada usuario.

En la aplicación los clientes realizan funciones como:

- Manejar la interfaz de usuario.
- Generar consultas e informes sobre la base de datos.

El servidor realiza, entre otras, las siguientes funciones:

- Controlar accesos concurrentes a las bases de datos compartidas.
- Enlaces de comunicaciones con otras redes de área local o extensa.



## 8.3 – Servicios Web

A continuación se describen los casos del diseño de requisitos más técnicamente para establecer los procedimientos y pliegos que deben cubrirse para utilizar los servicios web de los cuales el aplicativo se nutre de información.

### 8.3.1 – Servicio Lematizador

La llamada a las funciones del servicio lematizador, una vez realizada la conexión con él mismo se debe hacer mediante las siguientes nomenclaturas.

El método *Reconocer* se invoca mediante la llamada "*Lematizador.reconocer()*", pasándole como parámetros: la palabra a buscar, que será de tipo string o ristra; idioma, también de tipo string o ristra y finalmente el campo multiPrefijos que será de tipo boolean o buleano. Como ejemplos de esta llamada se detallan las llamadas con las palabras *hablar* y *cohablaría*.

- **Hablar:** Reconocer("hablar", "es", false);
- **Cohablaría:** Reconocer ("cohablaría", "es", false);

El primer campo será la palabra deseada, mientras que los dos siguientes campos siempre tendrán los mismos valores "es" para denotar el idioma que siempre será español y "false" para denotar que nunca se hará una llamada para una palabra multiprefijo.

Esta función siempre devolverá un bloque de tipo array o vector con la siguiente estructura:

```
<ArrayOfAnyType>
  <anyType xsi:type="ArrayOfAnyType">
    <anyType xsi:type="xsd:string">co</anyType>
    <anyType xsi:type="xsd:int">146265</anyType>
    <anyType xsi:type="xsd:int">21</anyType>
    <anyType xsi:type="xsd:int">0</anyType>
    <anyType xsi:type="xsd:string">lo</anyType>
  </anyType>
</ArrayOfAnyType>
```

Aunque en este bloque está especificado, como deben ser tratados los tipos de elementos que guarda, hay que tener en cuenta que el sistema devuelve un array en el que todos sus elementos son de tipo Object u Objeto. Dicho esto se explican que representa cada campo.

- **Campo 1:** un string que representa el prefijo de la palabra, si no tiene estará vacío.
- **Campo 2:** contiene un entero con la clave de la forma canónica de la palabra.
- **Campo 3:** entero que representa el código de flexión de la palabra consultada.
- **Campo 4:** irrelevante.
- **Campo 5:** string que representa el pronombre enclítico de la palabra consultada.

Cuando la palabra consultada no tiene pronombre enclítico este campo estará vacío. Para lo infinitivos el código de flexión es uno.

Una vez recibida la respuesta de la llamada al método “*Reconocer*” se tiene que hacer la llamada a otro método del servicio lematizador. El método utilizado es “*Consulta*”. Como la conexión al servicio está hecha desde el anterior método ahora solo será necesario hacer la llamada la función mediante la siguiente nomenclatura, “*Lematizador.consulta()*”.

Por tanto la llamada “*Consulta("146265" , "es", "canonicas");*” lleva como primero parámetro valor convertido a tipo string del campo 2 de la llamada al método “*Reconocer*”, el cual lleva el código de la de la forma canónica de la palabra introducida por el usuario. El segundo campo de esta llamada también es de tipo string y representa el código del idioma siempre será español y será “*es*”. Para terminar, el tercer parámetro de esta llamada siempre será el mismo, un string, con la palabra canónicas.

La respuesta a la Consulta de canónicas tiene los siguientes campos:

- 1º: **string** - Forma canónica.
- 2º: **int** - Código de categoría gramatical.
- 3º: **int** - Código de la flexión de la forma canónica.
- 4º: **int** - Frecuencia de aparición.
- 5º: **int** - Código de la categoría con género y número.

Al igual que en el método anterior este bloque viene devuelto conteniendo únicamente tipos array los cuales deben de ser tratador como strings o enteros dependiendo del dato que representen.

Con los datos obtenidos de la llamada a este último método se puede hacer la llamada al **servicio Familias** para obtener finalmente la familia de palabras relacionadas con la palabra introducida, teniendo como raíz la forma canónica de dicha palabra.

### 8.3.2 – Servicio Familias

La llamada a las funciones del servicio Familias, se realizan de forma similar al servicio anterior. En primer lugar se realiza la conexión con el servicio mediante la url: <http://tip.dis.ulpgc.es/ServicioFamilias/Familias.aspx>.

El método *Consulta* se invoca mediante la llamada “*Familias.consulta()*”, pasándole como parámetros: la palabra a buscar, que será de tipo string o ristra; idioma, también de tipo string o ristra y finalmente el campo nombre de la consulta:

- 1º: **string** - Clave o código
- 2º: **string** - Idioma
- 3º: **string** - nombre de la consulta

Por tanto, al realizar la llamada *Consulta("forma canónica", "es", "familias")*, se obtienen como resultado un array de tipo Object con los siguientes campos, uno por palabra:

- 1º: **int** - Categoría de la forma canónica del primer parámetro.
- 2º: **int** - Código de la familia a la que pertenece.
- 3º: **string** - Forma canónica derivada de esta familia.
- 4º: **int** - - Categoría de la forma canónica derivada de esta familia.
- 5º: **int** - Código del sufijo.
- 6º: **string** - Sufijo.
- 7º: **int** - 1 si la unión del sufijo ha sido regular y cero si ha sido irregular.
- 8º: **int** - Código del prefijo.
- 9º: **string** - Prefijo.
- 10º: **int** - 1 si la unión del prefijo ha sido regular y cero si ha sido irregular.

A partir de la información obtenida en del servicio Familias podemos pasar al tratamiento de la misma para almacenarla en un archivo de la forma que necesitarán las librerías utilizadas para la representación visual de la información. Dicho fichero tiene que almacenar la siguiente información:

	Campo	Contenido
<b>Para cada nodo</b>	Color	Característica del elemento.
	Forma	Característica del elemento.
	Posición del nodo	Característica del elemento.
	Palabra	La palabra en sí que representa ese nodo.
	Sufijo	Sufijo.
	Prefijo	Prefijo.
	padre	Nodo del que proviene.
	Forma canónica	Tipo de palabra.
	Hijos	Define los hijos que tiene un nodo.

### 8.3.3 – Diagrama de estados

Todo lo visto en los apartados anteriores se recoge en el siguiente esquema de flujo de estados.

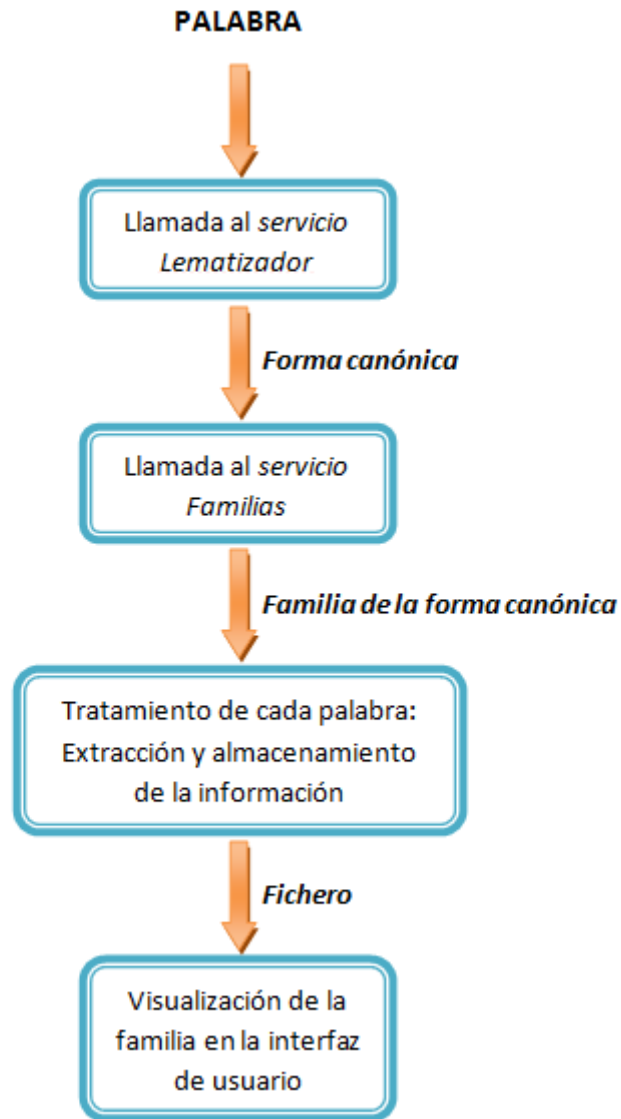


Figura 6. Diagrama de estados

## 8.4 – Aplicación multilinguaje

La aplicación multilinguaje se realiza creando archivos de lenguajes en la carpeta “App\_GlobalResources”, estos archivos son tablas con la siguiente estructura:

Nombre	Valor	Comentario
Clave que se corresponde con el ID del elemento de la estructura.	String con el texto en el idioma seleccionado.	Comentario explicativo opcional.

## 8.5 – Pantalla completa

La necesidad de cubrir este requisito surge, cuando el desarrollo estaba bastante avanzado y ocurre que algunas familias de palabras son tan grandes, incluyendo las relaciones raíz hijos e hijos nietos que se hace bastante incómodo para el usuario la consulta e interacción con el grafo y la información que este contiene. La *Figura 7. Pantalla normal* muestra la necesidad de desarrollar una pantalla completa por la incomodidad de utilizar la pantalla actual para algunas consultas.

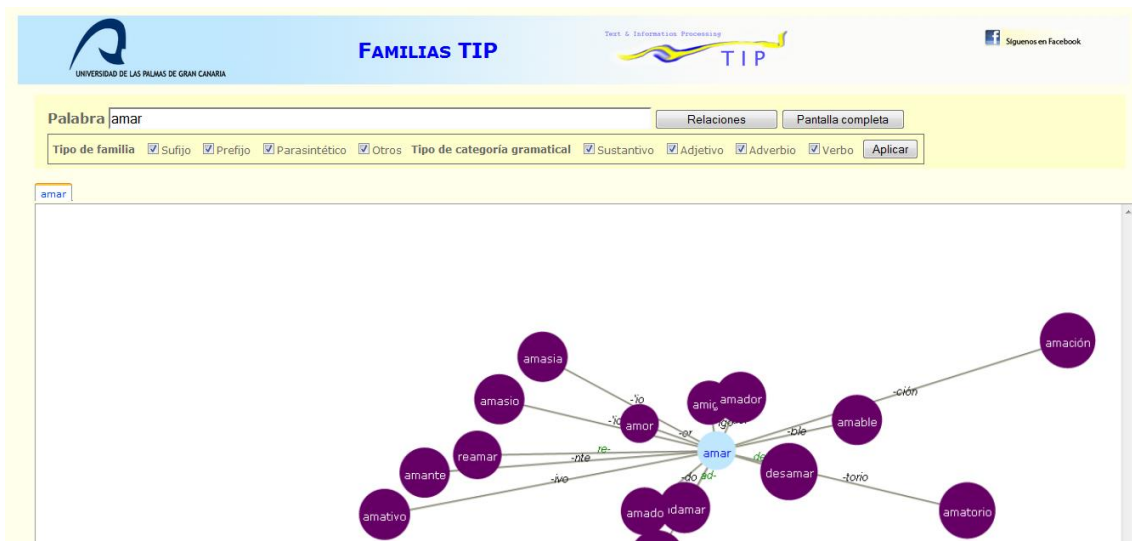


Figura 7. Pantalla normal

En principio este diseño se ha realizado sustituyendo el diseño de tabla de filtros anterior por uno representado en línea y añadiendo al diseño inicial un botón de **Pantalla completa**, para cambiar de vista cuando el usuario lo necesite. La siguiente figura muestra el diseño de pantalla completa donde desaparecen de la vista la cabecera de la página y el pie.

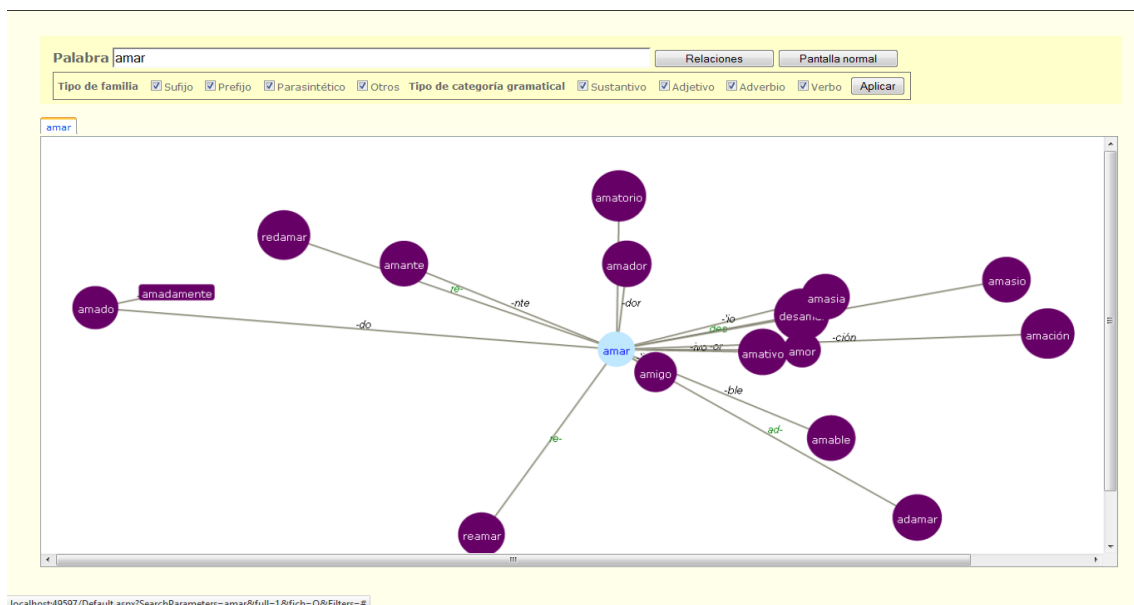


Figura 8. Pantalla completa

# 9. Desarrollo

---

En esta sección se procederá a explicar la fase de implementación realizada para el proyecto. Se detallarán aspectos muy importantes, por ejemplo, puntos relevantes de la interfaz, paquetes externos utilizados, partes más importantes de los módulos desarrollados, tecnologías utilizadas, etc.

El código fuente al completo se puede encontrar en el CD adjunto a la memoria. A continuación dedicaremos algunos apartados a los aspectos más relevantes de esta fase.

## 9.1 – Librerías

A continuación se detallarán las librerías, externas a ASP.Net, utilizadas y el trabajo realizado sobre cada una de ellas para obtener los resultados requeridos.

### 9.1.1 – Arbor.js

Como punto de partida y principal objetivo de este proyecto está el conseguir que la nueva aplicación proporcione una interfaz gráfica muy potente, visualmente amigable, fomentando la interacción del usuario con la información y animada.

Esta librería de AJAX proporciona algunos de estos atributos, con la posibilidad de desarrollarla aún más para llegar a cubrirlos todos e incluso obtener nuevas cualidades en la representación visual, para posibles futuros proyectos con esta aplicación. “Arbor.js” fue descubierta cuando se realizaban indagaciones sobre la representación gráfica de árboles o grafos y a partir de recibir el beneplácito del tutor se empezó a trabajar en la misma para cumplir objetivos.

El principal requisito a tener en cuenta es que los gráficos de esta librería deben ser dibujados en un elemento **Canvas** de *HTML5*, lo cual no resulta un problema porque la estructura de ASP.Net acepta elementos de *HTML5*. Además la instalación de esta librería es muy simple, únicamente hay que añadir los archivos siguientes:

- **Jquery.min.js:** Librería básica de Jquery para utilizar las funciones principales del lenguaje.
- **Arbor.js:** Contiene las funciones y procedimientos para formar un grafo.
- **Arbor-tween.js:** Aporta a la librería anterior colores, movimiento en tiempo real y animación.
- **Site.js:** librería que dibuja en el canvas el árbol, a partir de un archivo de almacenamiento Json, y aplica las funciones de eventos para la interacción con el árbol mediante el puntero.

Las modificaciones más importantes han sido sobre *“site.js”*, que como archivo más significativo de esta librería será el que aglutina las funciones que se deben modificar, añadir y borrar para obtener los resultados esperados.

Las primeras funciones añadidas han sido las que controlan la visualización del contenido de cada una de las pestañas del elemento AJAX *“TabContainer”* que separa en pestañas cada uno de los grafos que generan los servicios web a partir de la consulta del usuario.

Otro punto importante a destacar en este archivo es la modificación de la función que asigna la ruta del archivo que usa la librería para construir los grafos, siempre de tipo Json, para que utilice el archivo creado y modificado siempre que se cambien las condiciones o preferencias de muestreo de los mismos. El código del mismo se muestra a continuación:

```
var theUI = (function () {
    var url = "json/" + file;
    var theUI = null;
    $.ajax({
        'async': false,
        'global': false,
        'url': url,
        'dataType': "json",
        'success': function (data) {
            theUI = data;
        }
    });
    return theUI;
})
```

Una función que se ha modificado para conseguir un objetivo importante es la función *“clicked”* que originalmente hacía la función de permitir mover y arrastrar los nodos del grafo, además de enlazar los nodos que tenían enlaces con otras páginas. El código modificado ha permitido seguir moviendo y arrastrando los nodos además se ha añadido la función de poder realizar doble click sobre un nodo cualquiera que no tenga la posición raíz del grafo, y automáticamente la aplicación muestra el subárbol resultante de hacer el nodo seleccionado la raíz y poder ver así hasta sus nietos, si los hubiera. El código ejemplo se muestra a continuación:

```
//variable global para doble click
var clicks = 0;
clicked: function (e) {
    clicks++;
    if(clicks == 1){
        setTimeout(function(){
            var pos = $(canvas).offset();
            _mouseP = arbor.Point(e.pageX -
pos.left, e.pageY - pos.top)
            nearest = dragged =
sys.nearest(_mouseP);
            if(clicks == 1){
                if (dragged && dragged.node !==
null) dragged.node.fixed = false;
            }
            $(canvas).unbind('mousemove', handler.moved);
        });
    }
}
```

```

handler.dragged)                                $(canvas).bind('mousemove',
handler.dropped)                                $(window).bind('mouseup',
                                                }
                                                else{//Doble click
$(canvas).unbind('mousemove', handler.moved);
$(canvas).unbind('mousemove', handler.dragged)
                                                $(window).unbind('mouseup',
handler.dropped)
                                                var nombreNodo =
dragged.node.data.palabra;
                                                console.log(nombreNodo);
document.getElementById('MainContent_palabra').value = nombreNodo;
document.getElementById('MainContent_boton').click();
                                                }
                                                clicks = 0;
                                                }, 300)
                                                }
return false;
}

```

Otra función modificada, en este archivo “*site.js*” ha sido “*moved*” mostrar los nodos nietos que permanecen ocultos normalmente al pasar el puntero del ratón por encima de un nodo hoja. El código es el siguiente:

```

moved: function (e) {
    var pos = $(canvas).offset();
    _mouseP = arbor.Point(e.pageX - pos.left,
e.pageY - pos.top);
    nearest = sys.nearest(_mouseP);
    if (!nearest.node) return false;
    if (nearest.node.data.shape != 'dot') {
        selected = (nearest.distance < 50) ?
nearest : null;
    } else if ($.inArray(nearest.node.name,
allnodes) >= 0) { //creamos nuestro propio array
        if (nearest.node.name != _section) {
            _section = nearest.node.name;
            that.switchSection(_section);
        }
    }
    return false;
}

```

La última función que ha sido modificada de esta librería es la función “*redraw*”. Esta función redibuja en tiempo real el grafo para encuadrarlo de forma correcta dentro del elemento canvas y que los elementos del mismo no se superpongan entre ellos o sobre salgan



del marco del área de dibujo. Se ha modificado para poder añadir etiquetas a cada arista del grafo –Prefijos y/o Sufijos – como se muestra en la *Figura X. Aristas del grafo.*

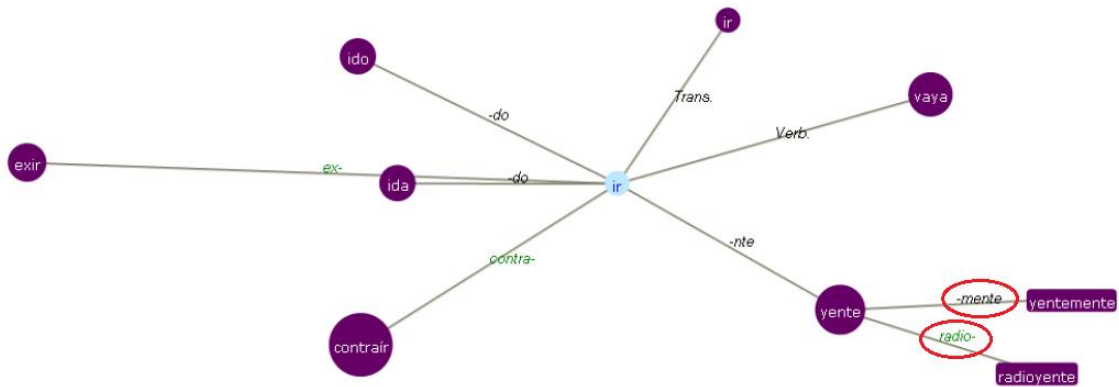


Figura 9. Aristas del grafo.

El código de esta función que posibilita añadir estas etiquetas en el siguiente:

```
redraw: function () {
    gfx.clear()
    sys.eachEdge(function (edge, p1, p2) {
        if (edge.source.data.alpha *
edge.target.data.alpha == 0) return;
        gfx.line(p1, p2, { stroke: "#b2b19d", width: 2,
alpha: edge.target.data.alpha });
        //Se escribe el label correspondiente a cada
edge encima de la línea
        ctx.strokeStyle = "rgba(0,0,0, .333)";
        ctx.lineWidth = 1;
        ctx.beginPath();
        ctx.moveTo(p1.x, p1.y);
        ctx.lineTo(p2.x, p2.y);
        ctx.stroke();
        ctx.font = 'italic 13px sans-serif';
        if (edge.data.suf !== "") {
            ctx.fillStyle = "black";
            ctx.fillText(edge.data.suf, (p1.x + p2.x) /
2, (p1.y + p2.y) / 2);
        }
        if (edge.data.pre !== "") {
            ctx.fillStyle = "green";
            var pos = 0;
            if (edge.data.suf !== "") { //Si no es nulo,
se coloca el prefijo arriba del sufijo
                pos = 2.22;
            }
            else { //Si no hay sufijo, se coloca en su
sitio correcto
                pos = 2;
            }
            ctx.fillText(edge.data.pre, (p1.x + p2.x) /
2, ((p1.y + p2.y)) / pos);
        }
    });
    sys.eachNode(function (node, pt) {
```

```

        var w = Math.max(15, 15 +
gfx.textWidth(node.data.palabra))
        if (node.data.alpha === 0) return;
        if (node.data.shape == 'dot') {
            gfx.oval(pt.x - w / 2, pt.y - w / 2, w, w,
{ fill: node.data.color, alpha: node.data.alpha });
            gfx.text(node.data.palabra, pt.x, pt.y + 7,
{ color: nodo.data.colorletra, align: "center", font: "Verdana",
size: 12 });
        } else {
            gfx.rect(pt.x - w / 2, pt.y - 8, w, 20, 4,
{ fill: node.data.color, alpha: node.data.alpha });
            gfx.text(node.data.palabra, pt.x, pt.y + 9,
{ color: nodo.data.colorletra, align: "center", font: " Verdana ",
size: 12 });
        }
    });
}

```

### 9.1.2 – Json.NET

Para crear los ficheros JSON, hacemos uso de la librería Json.NET que nos permite realizar esta operación a través de sus funciones de serialización y deserialización.

La función de serialización nos devuelve un String para guardar en un fichero, tal y como se ve a continuación:

```

//Serializamos la variable arbol, que es de tipo Dictionary
String json = Newtonsoft.Json.JsonConvert.SerializeObject(arbol,
Newtonsoft.Json.Formatting.Indented);
//A continuación, guarda la string en el fichero indicado por url
File.WriteAllText(url, json);

```

A dicha función le pasamos como parámetro un objeto de tipo Dictionary, ya que dicho tipo nos permite listar elementos distintos, teniendo en cuenta que los valores de Key no se repitan y permitiendo que el valor Value pueda repetirse.

Por otro lado, la función de deserialización nos devuelve un objeto a partir del contenido de un fichero JSON, como se muestra en el siguiente código:

```

//Se guarda el contenido del fichero JSON indicado por la url
String json = File.ReadAllText(url);
//Deserializamos el contenido en una variable de tipo Dictionary
Dictionary<String, Object> arbol =
Newtonsoft.Json.JsonConvert.DeserializeObject<Dictionary<String,
Object>>(json);

```

Tanto para serializar como deserializar, hemos podido observar que necesitamos utilizar una variable de tipo Dictionary. Esta variable se define de la siguiente manera:

```

Dictionary<String, Object> arbol = new Dictionary<String, Object>
{
    {"nodes", nodos},
    {"edges", edges}
};

```

Todo elemento del `Dictionary` se compone de un par *Key-Value*. En este caso se ha definido la *Key* como un `String`, y el *Value* de tipo `Object`. Cabe destacar que el *Value* realmente es otro `Dictionary`, siendo distinto según sea para la *Key* "nodes" o "edges".

- **Si se trata de "nodes":**

```
Dictionary<String, Object> nodos = new Dictionary<String, Object>();
```

De nuevo, cada elemento está formado por un par *Key-Value*, con los tipos `String` y `Object` respectivamente.

La variable *Value* que se inserta por cada nueva *Key*, en vez de ser de tipo `Object`, realmente es de la clase `Raiz`, definida por nosotros mismos:

```
public class Raiz
{
    public String color { get; set; }
    public String shape { get; set; }
    public String palabra { get; set; }
    public int alpha { get; set; }
    public String codSuf { get; set; }
    public String codPre { get; set; }
    public String padre { get; set; }
    public int codFormaCano { get; set; }
    public int colorletra { get; set; }
    public int codOtros { get; set; }

    public Raiz() { }
    public Raiz(String colorA, String shapeA, int alphaA,
        String palabraA, String codSufA, String codPreA, int
        codFormaCanoA, String padreA, String colorletraA, int
        codOtrosA)
    {
        color = colorA;
        shape = shapeA;
        alpha = alphaA;
        palabra = palabraA;
        codSuf = codSufA;
        codPre = codPreA;
        codFormaCano = codFormaCanoA;
        padre = padreA;
        colorletra = colorletraA;
        codOtros = codOtrosA;
    }
};
```

- **Si se trata de "edges":**

```
Dictionary<String, Object> edges = new Dictionary<String, Object>();
```

De nuevo, cada elemento está formado por un par *Key-Value*, con los tipos `String` y `Object` respectivamente.

La variable *Value* que se inserta por cada nueva *Key*, en vez de ser de tipo `Object`, realmente es de la clase `Data`, definida por nosotros mismos:

```
public class Data
{
```

```

        public String suf { get; set; }
        public String pre { get; set; }
        public int length { get; set; }

        public Data() { }
        public Data(String dataA, String preA)
        {
            suf = dataA;
            pre = preA;
            length = 1;
        }
    };

```

### 9.1.3 - Json

Gracias a la librería Json.NET se generó el fichero JSON, tal y como se vio en el apartado anterior. La estructura del fichero JSON es fija, es decir, cualquier palabra tiene un JSON similar, diferenciándose únicamente en los datos que lo rellenan.

La estructura del fichero es la siguiente:

```

{
  "nodes": {
    "0": {
      "color": "#BFE7FE",
      "shape": "dot",
      "palabra": "exir",
      "alpha": 1,
      "codSuf": "",
      "codPre": "",
      "padre": "",
      "codFormaCano": 0,
      "colorletra": "#0000FF",
      "codOtros": 0
    },
    "1": {
      "color": "#660066",
      "shape": "dot",
      "palabra": "exida",
      "alpha": 1,
      "codSuf": "-do",
      "codPre": "",
      "padre": "0",
      "codFormaCano": 1000,
      "colorletra": "#FFFFFF",
      "codOtros": 54
    },
    "2": {
      "color": "#660066",
      "shape": "dot",
      "palabra": "éxito",
      "alpha": 1,
      "codSuf": "-'ito",
      "codPre": "",
      "padre": "0",
      "codFormaCano": 1000,
      "colorletra": "#FFFFFF",
      "codOtros": 146
    },
    "100": {
      "color": "#660066",
      "shape": "",
      "palabra": "exitoso",
      "alpha": 0,
      "codSuf": "-oso",

```

```

        "codPre": "",
        "padre": "2",
        "codFormaCano": 1100,
        "colorletra": "#FFFFFF",
        "codOtros": 174
    },
    "101": {
        "color": "#660066",
        "shape": "",
        "palabra": "superéxito",
        "alpha": 0,
        "codSuf": "",
        "codPre": "super-",
        "padre": "2",
        "codFormaCano": 1000,
        "colorletra": "#FFFFFF",
        "codOtros": 0
    },
    "3": {
        "color": "#660066",
        "shape": "dot",
        "palabra": "exido",
        "alpha": 1,
        "codSuf": "-do",
        "codPre": "",
        "padre": "0",
        "codFormaCano": 1100,
        "colorletra": "#FFFFFF",
        "codOtros": 54
    }
},
"edges": {
    "2": {
        "100": {
            "suf": "-oso",
            "pre": "",
            "length": 1
        },
        "101": {
            "suf": "",
            "pre": "super-",
            "length": 1
        }
    },
    "0": {
        "1": {
            "suf": "-do",
            "pre": "",
            "length": 1
        },
        "2": {
            "suf": "-'ito",
            "pre": "",
            "length": 1
        },
        "3": {
            "suf": "-do",
            "pre": "",
            "length": 1
        }
    }
}
}
}

```

En el fichero anterior, se divide en dos partes claramente diferenciadas: “nodes” y “edges”. En la sección “nodes”, se listan todas las palabras que conforman la familia de una palabra en concreto, incluyendo dicha palabra que representa la raíz del árbol. En la sección

“edges” se relacionan los nodos, es decir, se puede ver cuáles son los hijos de cada uno de los nodos.

En “nodos”, la definición de cada nodo tiene la siguiente estructura:

```
"0": {
  "color": "#BFE7FE",
  "shape": "dot",
  "palabra": "exir",
  "alpha": 1,
  "codSuf": "",
  "codPre": "",
  "padre": "",
  "codFormaCano": 0,
  "colorletra": "#0000FF",
  "codOtros": 0
}
```

Dónde:

- "0", indica el identificador asignado a dicho nodo. Este identificador es único para cada nodo, por lo que no se repite.
- "color", es el color de fondo del círculo que se dibuja en la interfaz y que contiene a la palabra de este nodo, es decir, a *ser*.
- "shape", es el atributo que indica la forma del elemento visual que contiene a la palabra de este nodo. El valor “dot” representa un círculo y todos los hijos directos de la raíz tienen esta forma. Cuando el valor es vacío, indica que es hijo de un hijo.
- "palabra", contiene la palabra en sí, es decir, la palabra *exir*.
- "alpha", indica si el nodo tiene hijos (valor igual a 1) o es un nodo hoja (valor igual a 0).
- "codSuf" y "codPre", contienen los valores de sufijo y prefijo, respectivamente. Si alguno de estos atributos o ambos tienen valor vacío, significa que no tiene ese atributo o atributos en concreto.
- "padre", indica cuál es el padre del nodo en cuestión. Si el valor es vacío, significa que este nodo es la raíz del árbol.
- "codFormaCano", indica el código de la forma canónica de la palabra para poder diferenciar entre sustantivos, adjetivos, adverbios o verbos. Si el valor es 0 significa que este nodo es la raíz del árbol.
- "colorletra", indica el color de la letra de los nodos.
- "codOtros", indica el código del tipo de categoría **Otros**.

En la sección “edges”, se definen los hijos de un nodo de la siguiente manera:

```
"1": {
  "100": {
    "suf": "Trans.",
    "pre": "",
    "length": 1
  },
  "101": {
    "suf": "",
    "pre": "en-",
    "length": 1
  }
}
```

Dónde:

- "1", indica el identificador del nodo del que van a partir los nodos hijos.
- "100" y "101", representan los nodos hijos.
- "suf", representa el valor del sufijo. Si el valor es vacío, indica que no hay sufijo.
- "pre", representa el valor del prefijo. Si el valor es vacío, indica que no hay prefijo.
- "length", indica la longitud de la arista que une padre a hijo.

#### 9.1.4 – NuGet

Durante la ejecución de este proyecto ha surgido la necesidad de añadir a la interfaz un sistema de pantallas separadas por pestañas para visualizar las distintas familias de palabras que podrían generar los servicios web ante una palabra consultada por el usuario.

Para esta implementación ya existen diferentes tipos de soluciones, una muy utilizada por desarrolladores Asp.Net es la librería *AJAX Control Toolkit*, esta librería es instalable en el entorno *Visual Studio 2010* mediante el gestor de paquetes *Nuget*.

NuGet es una extensión de Visual Studio que hace que sea fácil agregar, actualizar y elimina bibliotecas (implementadas como paquetes) en un proyecto de Visual Studio. Un paquete de NuGet es un conjunto de archivos empaquetados en un solo archivo con extensión .nupkg que usa el formato Convenciones de empaquetado abierto.

La instalación se realizó a través del *Visual Studio 2010* y a partir de esta instalación se han podido realizar todas las inclusiones de paquetes necesarios, correctamente y a golpe de ratón.

#### 9.1.5 – AJAX Control Toolkit

Como ya se ha nombrado anteriormente esta librería ha sido necesaria para implementar la visualización por pestañas de este proyecto. Esta es una librería que mediante componentes del lado del cliente y del servidor, ASP.NET AJAX permite al desarrollador crear aplicaciones web en ASP.NET 2.0 que pueden actualizar datos en la página web sin un recarga completa de la misma. La tecnología clave que permite esta funcionalidad es el objeto XMLHttpRequest, junto con Javascript y DHTML.

El código introducido para tratar la estructura de estas pestañas ha sido incrustado directamente en el documento aspx, dado que son únicamente elementos visuales, en primer lugar se ha declarado un controlador del objeto para poder trabajar con él en la parte servidor.

Una vez implementado el controlador solo fue necesario añadir un objeto *TabContainer* y a este elemento introducirle tantas pestañas como fuera necesario para las posibles respuestas de los servicios web, como se muestra en el siguiente código:

```
<ajaxToolkit:ToolkitScriptManager ID="sm1" runat="server"
EnableHistory="true"
EnableSecureHistoryState="false"></ajaxToolkit:ToolkitScriptManager
>
```

```

    <ajaxToolkit:TabContainer ID="TabContainer1" runat="server"
ActiveTabIndex="0" AutoPostBack="false"
    Height="370px" Width="1163px" style="margin-right: 15px"
ScrollBars="Both" Visible="false">
    <ajaxToolkit:TabPanel runat="server" HeaderText="TabPanel1"
ID="TabPanel1" OnClientClick="cambio1">
        <ContentTemplate>
            <canvas height="522" width="1048" id="sitemap0"
class=""></canvas>
        </ContentTemplate>
    </ajaxToolkit:TabPanel>
    <ajaxToolkit:TabPanel ID="TabPanel2" runat="server"
HeaderText="TabPanel2" Visible="False" OnClientClick="cambio2">
        <ContentTemplate>
            <canvas height="522" width="1048" id="sitemap1"
class=""></canvas>
        </ContentTemplate>
    </ajaxToolkit:TabPanel>
    <ajaxToolkit:TabPanel ID="TabPanel3" runat="server"
HeaderText="TabPanel3" Visible="False" OnClientClick="cambio3">
        <ContentTemplate>
            <canvas height="522" width="1048" id="sitemap2"
class=""></canvas>
        </ContentTemplate>
    </ajaxToolkit:TabPanel>
    <ajaxToolkit:TabPanel ID="TabPanel4" runat="server"
HeaderText="TabPanel4" Visible="False" OnClientClick="cambio4">
        <ContentTemplate>
            <canvas height="522" width="1048" id="sitemap3"
class=""></canvas>
        </ContentTemplate>
    </ajaxToolkit:TabPanel>
    <ajaxToolkit:TabPanel ID="TabPanel5" runat="server"
HeaderText="TabPanel5" Visible="False" OnClientClick="cambio5">
        <ContentTemplate>
            <canvas height="522" width="1048" id="sitemap4"
class=""></canvas>
        </ContentTemplate>
    </ajaxToolkit:TabPanel>
</ajaxToolkit:TabContainer>

```

Tanto el panel como las pestañas están ocultos por defecto siendo solo mostradas a partir de que la consulta realizada por el usuario las necesite. Un ejemplo del modo de funcionamiento del panel es la palabra *fue* que devuelve dos pestañas con la forma canónica *Ir* y otra con la forma canónica *Ser*, como se puede ver en la figura *Figura16. Contenedor por pestaña*.



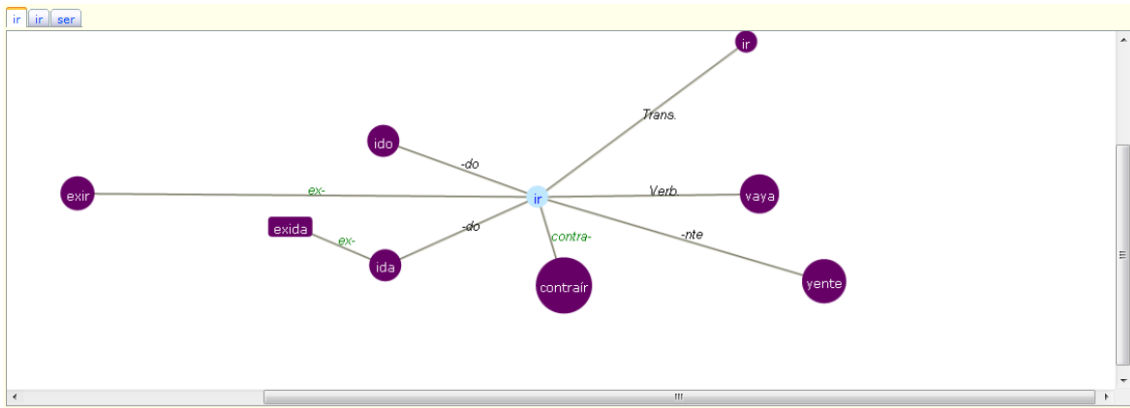


Figura 10. Contenedor por pestañas

## 9.2 – Conexión con los servicios Web

Para cubrir este aspecto se contó con información detallada adquirida tanto en las reuniones iniciales con el tutor del proyecto como con informes facilitados por el mismo.

En primer lugar hay que realizar la conexión en el entorno de desarrollo Visual Studio 2010, dando que los servicios Web están ya establecidos y funcionando con la aplicación anterior.

Para realizar la conexión con los servicios **Lematizador** y **Familias**, basta con agregar una referencia de servicio al proyecto web y en *opciones avanzadas* seleccionar *Agregar referencia de servicio*. En el cuadro de dialogo que se abre hay que rellenar los campos necesarios: dirección URL y nombre que tendrá el servicio en la aplicación. En el propio cuadro de dialogo se puede comprobar que la conexión se ha realizado correctamente, porque en la ventana web de la consola aparecen la información del servicio – métodos implementados, descripción general del servicio, forma de conectar con él, etc. – La figura X muestra que la conexión está correctamente realizada, al haber cargado la información del mismo en la ventana del navegador.

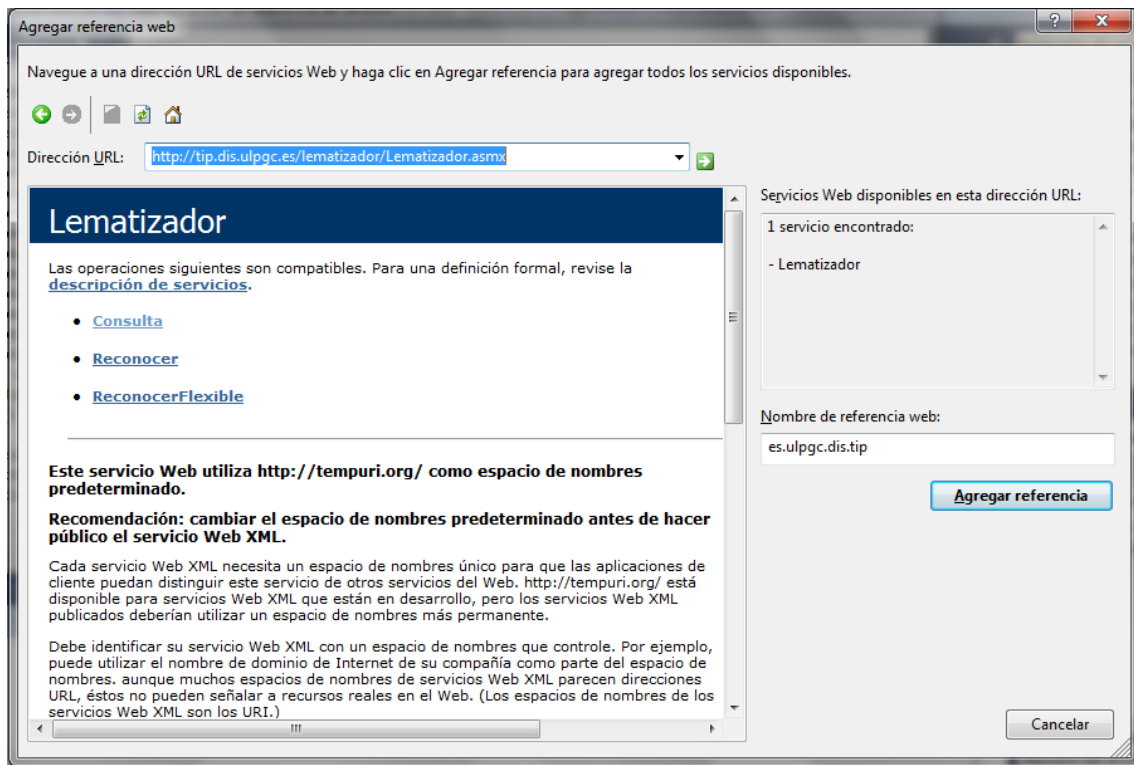


Figura 11. Conexión Servicios Web

Una vez agregados al proyecto ambas referencias al servicio web ya se puede hacer uso de los mismos en el código para obtener la información necesaria del corpus de palabras, así como las relaciones morfológicas de las mismas. Esta información debe recibir un tratamiento especial en algunos casos que serán explicados más adelante. Ahora solo cabe nombrar que la información debe ser almacenada linealmente para ello se ha utilizado la Clase proporcionada por Visual C#, Dictionary<key, value>. Esta clase provee al usuario de un almacén en línea donde se almacenan valores accesibles por clave. Para más información sobre esta clase véase el apartado 9.1.2 – *Json.NET*.

### 9.3 – Variables de sesión

En la aplicación web que estamos desarrollando, hacemos uso de variables de sesión para almacenar información sobre cada uno de los usuarios que acceden a la aplicación y leer dicha información en distintos momentos con los siguientes propósitos:

- Realizar un control de usuarios activos.
- Almacenar la información de las búsquedas en el espacio asignado a cada usuario.
- Eliminar la información generada durante el uso de la aplicación que deja un usuario concreto cuando cierra dicha aplicación.

El fichero *global.asax* contiene el código para implementar las funcionalidades descritas anteriormente. En primer lugar, se declara una lista para llevar el control de las

sesiones activas identificándolas por medio de su ID y que se va a mantener a lo largo de toda la vida de la aplicación:

```
private static List<String> _sessionInfo;
private static readonly object padlock = new object();

public static List<string> Sessions
{
    get
    {
        lock (padlock)
        {
            if (_sessionInfo == null)
            {
                _sessionInfo = new List<string>();
            }
            return _sessionInfo;
        }
    }
}
```

Al iniciarse la aplicación, automáticamente se lanza el procedimiento *Application\_Start*, donde se inicializa la ruta de repositorio que contiene toda la información de todos los usuarios y el número de usuarios que se conectan a la aplicación:

```
void Application_Start(object sender, EventArgs e)
{
    // Código que se ejecuta al iniciarse la aplicación
    Application["UsuariosConectados"] = 0;
    Application["RutaRepositorio"] = Server.MapPath("~/") + @"\json\";
}
```

Cada vez que un usuario accede a la aplicación web, se lanza el procedimiento *Session\_Start*, que incrementa el número de usuarios activos y crea una carpeta propia para sólo dicho usuario en el repositorio la cual contendrá sus búsquedas. Además, se añade el ID de la sesión de este usuario a la lista de sesiones activas para poder gestionar las sesiones adecuadamente.

```
void Session_Start(object sender, EventArgs e)
{
    // Código que se ejecuta cuando se inicia una nueva sesión
    Application["UsuariosConectados"] = (int)Application["UsuariosConectados"] + 1;
    Session["CodigoUsuario"] = Session.SessionID;
    String CodigoUsuario = (string)Session["CodigoUsuario"];
    String url = Server.MapPath("~/json/" + Session["CodigoUsuario"]);
    Directory.CreateDirectory(url);
    Sessions.Add(Session.SessionID);
}
```

Cuando el usuario permanece inactivo durante un periodo de tiempo o cierra la aplicación, entonces se lanza el procedimiento *Session\_End* que elimina su sesión de la lista de sesiones activas y elimina del repositorio la carpeta usada por dicho usuario y toda la información contenida en él:

```

void Session_End(object sender, EventArgs e)
{
    String url = Application["RutaRepositorio"].ToString();
    List<String> ss = Sessions;
    //Elimina el directorio usado por el usuario en su sesión.
    String[] carpetas = Directory.GetDirectories(url);
    foreach (String carpeta in carpetas)
    {
        String nombreS =
        carpeta.Substring(carpeta.LastIndexOf("\\") + 1);
        if (!Global.Sessions.Contains(nombreS))
        {
            Directory.Delete(carpeta, true);
        }
    }

    Application["UsuariosConectados"]=(int)Application["Usuario
sConectados"] - 1;
}

```

Además, en el fichero `Web.config` se comprueba que esté incluido la siguiente línea de código:

```
<sessionState mode="InProc" timeout="10"></sessionState>
```

Dónde:

- `mode="InProc"`, es el modo por defecto de estado de sesión. Especifica que los valores del estado de la sesión se almacenan en la memoria del servidor Web local. Éste es el único modo que soporta el evento `Session_End`.
- `timeout="10"`, especifica el número de minutos que una sesión puede estar inactiva antes de que se abandone. Por defecto, el valor de este atributo es de 20, indicando 20 minutos. Pero se ha reducido a 10 minutos.

## 9.4 – Post VS Get

Durante el desarrollo de la aplicación, el problema que se encontró fue que al presionar sobre el botón de ‘Atrás’ del navegador el documento devuelto había expirado.

Este problema no está relacionado con las cabeceras de control de caché, ya que el mismo error aparece para páginas que incluyen estas cabeceras. El problema reside en que al programar en ASP.NET – C#, por defecto, los eventos de cualquiera de los elementos se envían al servidor vía HTTP POST. Esto significa que el servidor ve esto como un envío de formulario, y los navegadores actuales tienen mecanismos de prevención contra el reenvío del mismo formulario múltiples veces, ya sea porque usuarios sin experiencia puedan realizar esta operación por accidente y sin tener conocimiento de ello.

La solución es enviar las consultas a través del método HTTP GET. Para ello, introducimos en la URL la palabra introducida para buscar su familia. De esta forma, cuando el usuario presiona sobre el botón *Relaciones* se lanza el procedimiento que controla dicho evento y se construye la URL con la palabra introducida como parámetro de búsqueda y se redirige a la misma página. El código es el siguiente:

```
protected void Button1_Click(object sender, EventArgs e)
```

```

{
    String antiguo = palabra.Text;
    String parametros = @"?SearchParameters=" + antiguo;
    String searchUrl = "Default.aspx" + parametros;
    Response.Redirect(searchUrl);
}

```

Al redirigir a la misma página, hay que tener en consideración que el servidor entiende esta operación como se está cargando por primera vez la página aunque la URL sea la misma, y no porque se haya refrescado. Por tanto, en el procedimiento `Page_Load`, lanzado por defecto al cargar cualquier página, primero se comprueba que la página se está cargando por primera vez mediante la comprobación de `IsPostBack`, a continuación se comprueba que en la URL el parámetro `SearchParameters` tenga algún valor y si es el caso, se introduce en el input y se llama al procedimiento `Relaciones` para buscar su familia. El código es el siguiente:

```

protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        if (!string.IsNullOrEmpty(Request.QueryString["SearchParameters"]))
        {
            if (!string.IsNullOrEmpty(Request.QueryString["full"]))
            {
                String full = Request.QueryString["full"];
                HtmlGenericControl header = (HtmlGenericControl)Master.FindControl("header");
                HtmlGenericControl footerGeneral = (HtmlGenericControl)Master.FindControl("footerGeneral");
                if (full.Contains("1")) {
                    header.Visible = false;
                    footerGeneral.Visible = false;
                }
                Session["PantallaCompleta"] = true;
                Button3.Visible = false;
                Button4.Visible = true;
            }
            else
            {
                header.Visible = true;
                footerGeneral.Visible = true;
            }
            Session["PantallaCompleta"] = false;
            Button3.Visible = true;
            Button4.Visible = false;
        }
        if (!string.IsNullOrEmpty(Request.QueryString["Filters"]))
        {
            String fil = Request.QueryString["Filters"];
            if (fil.Contains("1"))
                CheckBox1.Checked = false;
            if (fil.Contains("2"))
                CheckBox2.Checked = false;
        }
    }
}

```

```

        if (fil.Contains("3"))
        CheckBox3.Checked = false;
        if (fil.Contains("4"))
        CheckBox4.Checked = false;
        if (fil.Contains("5"))
        CheckBox5.Checked = false;
        if (fil.Contains("6"))
        CheckBox6.Checked = false;
        if (fil.Contains("7"))
        CheckBox7.Checked = false;
        if (fil.Contains("8"))
        CheckBox8.Checked = false;
    }
    palabra.Text =
Request.QueryString["SearchParameters"];
    if
(!string.IsNullOrEmpty(Request.QueryString["relacion
es"])) &&
Request.QueryString["relaciones"].Contains("1"))
    {
        Relaciones();
    }
    else if
(!string.IsNullOrEmpty(Request.QueryString["relacion
es"])) &&
Request.QueryString["relaciones"].Contains("2"))
    {
        Relaciones2();
    }
    else
    {
        TabContainer1.Visible =
true;
        Table1.Visible = true;
        String[] fich =
Session["ficherosActuales"].ToString().Split(',');
        int numFich = fich.Length;
        if (numFich == 0)
        {
            error.Visible = true;
            TabContainer1.Visible =
false;
            Table1.Visible = false;
        }
        if (numFich >= 1)
        {
            TabPanel1.HeaderText =
ObtenerNombre(fich[0]);
            TabPanel1.Visible =
true;
        }
        if (numFich >= 2)
        {
            TabPanel2.HeaderText =
ObtenerNombre(fich[1]);
            TabPanel2.Visible =
true;
        }
        if (numFich >= 3)
        {
            TabPanel3.HeaderText =
ObtenerNombre(fich[2]);
            TabPanel3.Visible =
true;
        }
        if (numFich >= 4)
    {

```

```

    ObtenerNombre(fich[3]);
    true;
}
if (numFich == 5)
{
    TabPanel4.HeaderText =
    TabPanel4.Visible =
    ObtenerNombre(fich[4]);
    TabPanel5.HeaderText =
    TabPanel5.Visible =
    true;
}

ClientScript.RegisterStartupScript(Page.GetType(),
"invocafuncion", "inicializa('" +
Session["ficherosActuales"] + "' + '"', '"
+Session["CodigoUsuario"] + "')";", true);
}
}
}
}

```

## 9.5 – Interfaz

El objetivo en cuanto a la interfaz ha sido el de realizar un trabajo de sencillez e intuición para que el usuario no tenga dudas en cuanto al funcionamiento del aplicativo. Por tanto, la portada del sitio solo muestra la información de cabecera a modo de banner, el contenido inicial –casilla para introducir la palabra y botón de búsqueda– y la información de pie de página además de la posibilidad de cambio de idioma. La *Figura 16. Portada del sitio*, será la siguiente:



Figura 12. Portada del sitio

Una vez el usuario ha introducido una palabra y realizado una búsqueda la interfaz mostrará además del contenido ya mencionado, una tabla de casillas Checkbox con las diferentes opciones de filtrado que el usuario puede hacer según tipo de familia y según tipo

de categoría gramatical y un elemento contenedor donde se dibujarán los diferentes grafos separados por pestañas, quedando la interfaz como muestra la *Figura 17. Interfaz de consulta*.

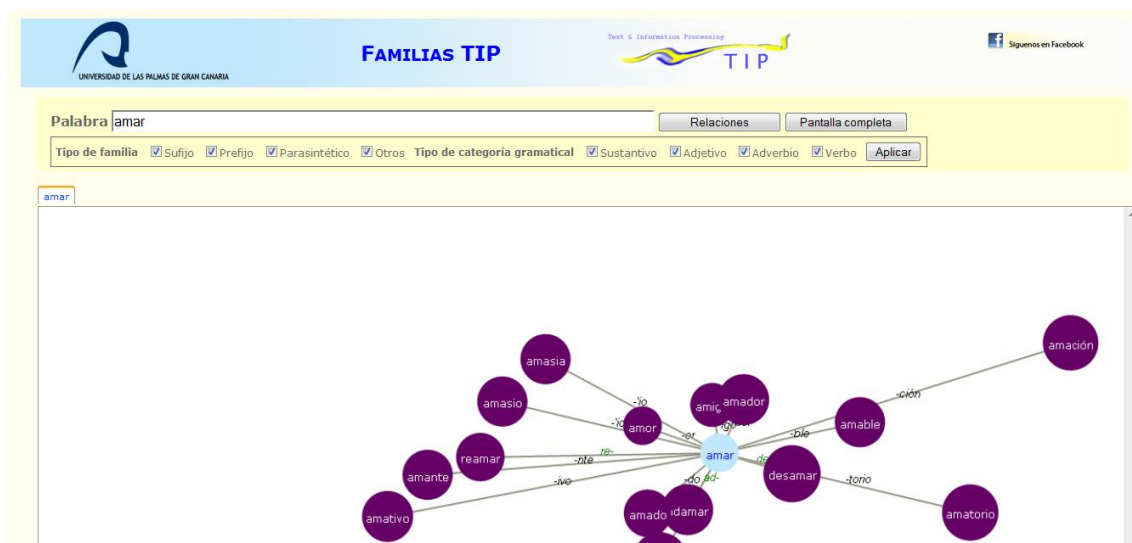


Figura 13. Interfaz de consulta

## 9.5 – Idioma

Otro requisito que se ha tenido en consideración en el desarrollo de este proyecto ha sido la posibilidad de desarrollar un sitio multilingüaje. Este trabajo se ha realizado de la siguiente forma.

En la interfaz donde se mostrarán los textos clasificados por idiomas están referenciados mediante un sistema de etiquetas que sustituye al texto donde normalmente estaría el campo del mismo en cada elemento de la estructura gráfica Asp.Net.

Ejemplo:

```
<asp:Label ID="Label2" runat="server" Text="<%=Resources.LocalizedText, Label2
%>" ></asp:Label>
```

En el campo *Text* se encuentra una etiqueta con la clave *Resources*, que se refiere a los archivos de recursos, nombrando el archivo requerido después de los dos puntos y añade después de la coma la clave del texto requerido.

El control de idiomas se hará desde la selección por defecto que tenga el navegador del usuario y además se añadirán dos imágenes botón con las banderas española y británica para que cada usuario, pueda cambiar el idioma de la aplicación a golpe de ratón.

Como la globalización de archivos de recursos, se ha declarado en la clase **MasterPage** se ha buscado una solución para implementarla, ya que no se puede llamar al método **InitializedCulture()**. El formato escogido para dicha implementación ha sido establecer la selección de idioma en el archivo **Global.asax**, donde se añadió el siguiente procedimiento:

```
//Código de prueba para aplicación multi-lenguaje
```



```

        protected void Application_BeginRequest(object sender,
        EventArgs e)
        {
            HttpCookie cookie = Request.Cookies["CultureInfo"];
            if (cookie != null && cookie.Value != null)
            {
                Thread.CurrentThread.CurrentUICulture = new
                System.Globalization.CultureInfo(cookie.Value);
                Thread.CurrentThread.CurrentCulture = new
                System.Globalization.CultureInfo(cookie.Value);
            }else{
                Thread.CurrentThread.CurrentUICulture = new
                System.Globalization.CultureInfo("es-ES");
                Thread.CurrentThread.CurrentCulture = new
                System.Globalization.CultureInfo("es-ES");
            }
        }
    }
}

```

A continuación en el fichero Site.Master se incluyeron, a pie de página, botones de enlace para los idiomas disponibles – inglés y español – con forma de banderas.

El código de estructura de los botones es el siguiente:

```

<asp:ImageButton ID="ImageButton1" runat="server" ImageAlign="Bottom"
ImageUr1=~\Graphics\bandera_ES.png OnClick="ImageButton1_Clicked" />
&nbsp;
<asp:ImageButton ID="ImageButton2" runat="server"
ImageUr1=~\Graphics\bandera_EN.png OnClick="ImageButton2_Clicked" />

```

Finalmente en el **Site.Master.cs** se incluyó el código para los manejadores de eventos de dichos botones:

```

public void ImageButton1_Clicked(object sender, EventArgs e)
{
    HttpCookie cookie = new HttpCookie("CultureInfo");
    cookie.Value = "es-ES";
    Response.Cookies.Add(cookie);
    Thread.CurrentThread.CurrentCulture = new CultureInfo("es-ES");
    Thread.CurrentThread.CurrentUICulture = new CultureInfo("es-ES");
    //Server.Transfer(Request.Path);
    if (Session["Parametros"] != null)
    {
        String p = Session["Parametros"].ToString();
        Session["Parametros"] = null;
        Response.Redirect(Request.Path + p);
    }
    else
    {
        Response.Redirect(Request.Path);
    }
}

public void ImageButton2_Clicked(object sender, EventArgs e)
{
    HttpCookie cookie = new HttpCookie("CultureInfo");
    cookie.Value = "EN";
    Response.Cookies.Add(cookie);
}

```

```

Thread.CurrentThread.CurrentCulture = new CultureInfo("EN");
Thread.CurrentThread.CurrentUICulture = new CultureInfo("EN");
//Server.Transfer(Request.Path);
if (Session["Parametros"] != null)
{
    String p = Session["Parametros"].ToString();
    Session["Parametros"] = null;
    Response.Redirect(Request.Path + p);
}
else
{
    Response.Redirect(Request.Path);
}
}

```

## 9.6 – Pantalla completa

Como se ha comentado en el apartado de diseño, añadirle la funcionalidad de mostrar una pantalla completa con los resultados del grafo que pudiera producir una consulta cualquiera ha sido prácticamente obligatoria. Como esta necesidad ha sido descubierta desarrollando el apartado de pruebas del proyecto en sí, su ejecución ha sido a posteriori con lo cual han sido cambiadas algunas funcionalidades para poder ajustarse a las nuevas necesidades del proyecto y se han añadido otras nuevas funciones.

En primer lugar se ha modificado la interfaz de usuario, una vez realizada la búsqueda, para que el elemento **Tabcontainer** ocupara todo el ancho de la web y estuviera lo más visible posible en la pantalla normal, para ello se ha realizado el cambio de la tabla de filtros por otra nueva, como ya se comentó en el apartado de diseño. El objetivo de esta función es que cuando el usuario haga click en el botón **Pantalla completa** la vista de la web sea únicamente el formulario de búsqueda con los botones **Relaciones** y **Pantalla normal**, siendo este segundo botón para volver a la vista habitual de la aplicación, la casilla de texto para introducir la palabra, la tabla de filtros para las familias de palabras y el elemento canvas con el grafo o grafos resultantes como muestra la *Figura 8. Pantalla completa* del capítulo de Diseño.

El código que ha posibilitado la opción de hacer invisible la cabecera y el pie de página ha sido el siguiente.

En el archivo que genera el evento del botón **pantalla completa**, es necesario referenciar la cabecera y pie de página que se encuentran en la página **Site.Master**, mediante la sentencia:

```
<%@ MasterType virtualpath="~/Site.Master" %>
```

Una vez establecida la relación con el archivo se hace la instancia a los objetos a esconder con este evento:

```

HtmlGenericControl header =
(HtmlGenericControl)Master.FindControl("header");
header.Visible = false;

```

Con esta referencia al hacer click al evento el **header** desaparecerá, repitiendo el mismo protocolo para el pie.

# 10. Manual de usuario y software

---

Esta aplicación está accesible en internet en el portal del grupo de investigación *TIP- Text & Information Processing*, accesible en <http://tip.dis.ulpgc.es/>. La aplicación está dentro de los recursos del portal, viéndose en el menú de la izquierda como *FAMILIAS TIP*.

El uso de la aplicación es de una gran sencillez para el usuario, por tratarse de consultas a un sistema web.

La portada del inicio muestra un formulario sencillo de una sola casilla para introducir una palabra, identificada mediante la etiqueta **Palabra**, y un botón al lado llamado **Relaciones**. El usuario introduce la palabra a relacionar y hace click en el botón.

Así aparece en la misma ventada una tabla de Checkbox con los diferentes filtros para las familias de palabras además de un botón para aplicar esos filtros, llamado **Aplicar** y un contenedor con los diferentes grafos de familias que pueda generar dicha palabra. Si la palabra no tiene familia o no existe el aplicativo mostrará un mensaje advirtiéndolo en ambos casos.

Una vez este desplegado el contenedor con el grafo, el usuario puede interactuar con dicho grafo, filtrando su contenido mediante el desactivado de los Checkbox y dándole al botón **Aplicar** que se encuentra en la misma tabla.

Además el usuario puede interactuar con el grafo, en sí, porque el grafo inicialmente muestra la raíz del mismo con la forma canónica de la palabra y sus descendientes directas a modos de nodos hoja. Si un nodo hoja tiene hijos al usuario pasar el puntero del ratón sobre dicho nodo estos se desplegarán mostrando el grafo el subárbol hasta el nivel siguiente.

Si un usuario quiere indagar más en un subárbol de familias de palabras puede hacer doble click sobre un nodo que no se el nodo raíz y se regenerará una nueva búsqueda apareciendo el nodos seleccionado como el nuevo nodo raíz y el subárbol resultante del mismo hasta tres niveles de profundidad, si existieran.

# Bibliografía

---

- **Página del grupo de investigación TIP:** <http://tip.dis.ulpgc.es/>.
- **Consultas de información general Wikipedia:** <https://es.wikipedia.org>.
- **Creación de archivo Json:** <http://stackoverflow.com/questions/14828520/how-to-create-my-json-string-by-using-c>.
- **Tratamiento de información a partir de archivos Json:** <http://james.newtonking.com/json/help/index.html>,  
<http://stackoverflow.com/questions/9110724/serializing-a-list-to-json>,  
<http://msdn.microsoft.com/es-es/library/bb410770%28v=vs.100%29.aspx>
- **Uso de la clase List C#:** <http://msdn.microsoft.com/es-es/library/bwabdf9z.aspx>
- **Uso de contenedores AJAX:** <http://geeks.ms/blogs/mrubino/archive/2007/11/28/asp-net-ajax-tabcontrol-tabs-optimizados.aspx>, <http://forums.asp.net/t/1069907.aspx>,
- **Administrar bibliotecas con Nuget:** <http://msdn.microsoft.com/es-es/magazine/hh547106.aspx>
- **Uso del paquete AJAX Control Toolkit:** <http://puntocode.com.mx/2010/09/12/error-de-referencia-cultural-en-asp-net-con-ajaxcontroltoolkit/>,  
<http://stephenwalther.com/archive/2011/05/23/install-the-ajax-control-toolkit-from-nuget>, <http://www.quizzpot.com/courses/aprendiendo-ext-js-3/articles/integracion-del-treepanel-y-tabpanel>,  
<http://www.quizzpot.com/courses/aprendiendo-ext-js-3/articles/las-pestanas-o-tabs>
- **Librerías Asp.Net:** <http://msdn.microsoft.com/es-es/library/bb350750.aspx>
- **Eliminación de archivos obsoletos Json:** <http://msdn.microsoft.com/es-es/library/vstudio/bb943484%28v=vs.100%29.aspx>
- **Tratamiento de información con la clase Dictionary:** <http://stackoverflow.com/questions/11598148/using-a-list-lookup-or-dictionary-for-a-large-list-of-data>
- **Variables de sesión Asp.Net:** <http://msdn.microsoft.com/es-es/library/87069683%28v=vs.80%29.aspx>, <http://msdn.microsoft.com/es-es/library/ms178581%28v=vs.100%29.aspx>, <http://msdn.microsoft.com/en-us/library/ms178583.aspx>
- **Aplicaciones multilinguaje Asp.Net:** <http://social.msdn.microsoft.com/Forums/es-ES/f62566c0-774d-4074-9459-778d8c6705d6/articulo-c-aplicaciones-multilinguaje-globalizacin-y-localizacin->
- **Manejo de eventos JQuery:** <http://api.jquery.com/category/events/event-handler-attachment/>, <http://www.mkyong.com/jquery/jquery-click-and-dblclick-example/>,  
<http://stackoverflow.com/questions/6330431/jquery-bind-double-click-and-single-click-separately>
- **Librería AJAX para grafos:** <http://www.ajaxshake.com/demo/ES/836/daf94a42/crear-grafos-animados-con-jquery-arborjs.html>

# Anexos

---

## Site.Master

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site.master.cs"
Inherits="FamiliasMorfolexicas.SiteMaster" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title><h1>Familias TIP</h1></title>
<link type="image/x-icon" rel="shortcut icon" href="/Graphics/favicon.ico">
<link href="~/Styles/Site.css" rel="stylesheet" type="text/css" />
<asp:ContentPlaceHolder ID="HeadContent" runat="server">

</asp:ContentPlaceHolder>
</head>
<body>
<form runat="server">
<div class="header" id="header" runat="server">
<div class="title" style="background-image:
url('./Graphics/fondo.png');">
<table style="width: 100em ; margin: auto;">
<tr>
<td>
<a target="_blank"
href="http://www.facebook.com/pages/Las-Palmas-De-Gran-Canaria-Spain/TIP-Text-
Information-Processing/111430348868237">
<asp:Image ID="Image1" runat="server"
Height="71px" ImageUrl="~/Graphics/ULPGC.png" Width="194px" />
</a>
</td>
<td>
<h1 style="color: #0000FF">
Familias TIP
</h1>
</td>
<td>
<a href="http://tip.dis.ulpgc.es">
<asp:Image ID="Image2" runat="server"
Height="46px" ImageUrl="~/Graphics/pajaro4.png" Width="229px" />
</a>
<a target="_blank"
href="http://www.facebook.com/pages/Las-Palmas-De-Gran-Canaria-Spain/TIP-Text-
Information-Processing/111430348868237">
<asp:Image ID="Image3" runat="server"
ImageAlign="Right" ImageUrl="~/Graphics/facebook.png"
BorderStyle="None" Height="20px"
Width="114px" />
</a>
</td>
</tr>
</table>
</div>
</div>
<div class="main">
<asp:ContentPlaceHolder ID="MainContent" runat="server"/>
</div>
```

```

        <div class="clear">
            </div>

        <div class="footer" id="footerGeneral" runat="server">
            <asp:ImageButton ID="ImageButton1" runat="server" ImageAlign="Bottom"
            ImageUrl="~/Graphics/bandera_ES.png" OnClick="ImageButton1_Clicked" />
            &nbsp;
            <asp:ImageButton ID="ImageButton2" runat="server"
            ImageUrl="~/Graphics/bandera_EN.png" OnClick="ImageButton2_Clicked" />
            <br />
            <asp:Label ID="footer" runat="server" Text="<%"$Resources:LocalizedText,
            footer %> "></asp:Label>
            <br />
        </div>
    </form>

</body>

</html>

```

## site.master.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Threading;
using System.Globalization;

namespace FamiliasMorfolexicas
{
    public partial class SiteMaster : System.Web.UI.MasterPage
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        public void ImageButton1_Clicked(object sender, EventArgs e)
        {
            //Session["idioma"] = "es";
            //Thread.CurrentThread.CurrentUICulture = new
            System.Globalization.CultureInfo("es-ES");
            HttpCookie cookie = new HttpCookie("CultureInfo");
            cookie.Value = "es-ES";
            Response.Cookies.Add(cookie);

            //Set the culture and reload for immediate effect.
            //Future effects are handled by Global.asax
            Thread.CurrentThread.CurrentCulture = new CultureInfo("es-ES");
            Thread.CurrentThread.CurrentUICulture = new CultureInfo("es-ES");
            //Server.Transfer(Request.Path);
            if (Session["Parametros"] != null)
            {
                String p = Session["Parametros"].ToString();
                Session["Parametros"] = null;
                Response.Redirect(Request.Path + p);
            }
        }
    }
}

```

```

        }
        else
        {
            Response.Redirect(Request.Path);
        }
    }

    public void ImageButton2_Clicked(object sender, EventArgs e)
    {
        //Session["idioma"] = "en";
        //Thread.CurrentThread.CurrentUICulture = new
        System.Globalization.CultureInfo("EN");

        HttpCookie cookie = new HttpCookie("CultureInfo");
        cookie.Value = "EN";
        Response.Cookies.Add(cookie);

        //Set the culture and reload for immediate effect.
        //Future effects are handled by Global.asax
        Thread.CurrentThread.CurrentCulture = new CultureInfo("EN");
        Thread.CurrentThread.CurrentUICulture = new CultureInfo("EN");
        //Server.Transfer(Request.Path);
        if (Session["Parametros"] != null)
        {
            String p = Session["Parametros"].ToString();
            Session["Parametros"] = null;
            Response.Redirect(Request.Path + p);
        }
        else
        {
            Response.Redirect(Request.Path);
        }
    }
}
}
}

```

## Default.aspx

```

<%@ Page Title="Familias TIP" Language="C#" MasterPageFile="~/Site.master"
AutoEventWireup="true"
    CodeBehind="Default.aspx.cs" Inherits="FamiliasMorfolexicas._Default" %>

<%@ MasterType virtualpath="~/Site.Master" %>
<%@ Import Namespace="System.Threading" %>
<%@ Import Namespace="System.Globalization" %>

<script runat="server">
    protected override void InitializeCulture()
    {
        String idioma = "";
        if (Session["idioma"] != null) idioma = Session["idioma"].ToString();
        if (Request.Url.ToString().IndexOf("lang=EN") > 0 ||
idioma.Contains("en"))
        {
            UICulture = "en";
            Culture = "en";
            Thread.CurrentThread.CurrentCulture =
CultureInfo.CreateSpecificCulture("en");
            Thread.CurrentThread.CurrentUICulture = new CultureInfo("en");
        }
    }
}

```



```

        if (Request.Url.ToString().IndexOf("lang=ES") > 0 ||
idioma.Contains("es"))
        {
            UICulture = "es";
            Culture = "es";
            Thread.CurrentThread.CurrentCulture =
CultureInfo.CreateSpecificCulture("es");
            Thread.CurrentThread.CurrentUICulture = new CultureInfo("es");
        }
        base.InitializeCulture();
    }
</script>

<asp:Content ID="HeaderContent" runat="server"
ContentPlaceHolderID="HeadContent">
    <link rel="stylesheet" href="Styles/style.css" type="text/css"/>
    <!--Librería añadida para utilización de la función doble click-->
    <script type="text/javascript" src="http://code.jquery.com/jquery-
1.9.1.js"></script>
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">

<div style="width: 100%; height: auto; padding-top: 15px; padding-left: 15px;
background-color: #ffc;">
    <asp:Label ID="Label2" runat="server" Text="<%=Resources.LocalizedText,
Label2 %>" Font-Size="Medium" Style="text-align: left" Font-Bold="True"
ForeColor="#666666"></asp:Label>
    <asp:TextBox ID="palabra" runat="server" MaxLength="500" Width="50%" Font-
Size="Medium" AutoCompleteType="Disabled" BorderStyle="Inset"
BorderWidth="2px"></asp:TextBox>
    <asp:Button ID="boton" runat="server" OnClick="Button1_Click"
Text="<%=Resources.LocalizedText, boton %>" Width="145px" />
    <asp:Button ID="Button3" runat="server" OnClick="Button3_Click"
Text="<%=Resources.LocalizedText, Button3 %>" Width="145px"
Visible="False" />
    <asp:Button ID="Button4" runat="server" OnClick="Button4_Click"
Text="<%=Resources.LocalizedText, Button4 %>" Width="145px Visible="False" />
</div>

    <div style="width: 100%; height: auto; padding-bottom: 5px; padding-left:
15px; background-color: #ffc; padding-top: 5px;" >
        <asp:Label ID="error" runat="server" Visible="false"
Text="<%=Resources.LocalizedText, error %>"></asp:Label>
        <asp:Label ID="aviso" runat="server" Visible="false"
Text="<%=Resources.LocalizedText, aviso %>"></asp:Label>
        <asp:Label ID="aviso2" runat="server" Visible="false"
Text="<%=Resources.LocalizedText, aviso2 %>"></asp:Label>
        <asp:Table ID="Table1" runat="server" CellPadding="5"
GridLines="horizontal"
Width="86em" visible="false">
            <asp:TableRow>
                <asp:TableCell ID="tip0" Text="<%=Resources.LocalizedText, tip0
%>" Font-Bold="True"></asp:TableCell>
                <asp:TableCell><asp:CheckBox ID="CheckBox1" runat="server"
Text="<%=Resources.LocalizedText, CheckBox1 %>" Checked="True" /></asp:TableCell>
                <asp:TableCell><asp:CheckBox ID="CheckBox2" runat="server"
Text="<%=Resources.LocalizedText, CheckBox2 %>" Checked="True" /></asp:TableCell>
                <asp:TableCell><asp:CheckBox ID="CheckBox3" runat="server"
Text="<%=Resources.LocalizedText, CheckBox3 %>" Checked="True" /></asp:TableCell>
            </asp:TableRow>
        </asp:Table>
    </div>

```

```

                <asp:TableCell><asp:CheckBox ID="CheckBox4" runat="server"
Text="<%"$Resources:LocalizedText, CheckBox4 %>" Checked="True" /></asp:TableCell>
                <asp:TableCell ID="gramatical" Text="<%"$Resources:LocalizedText,
gramatical %>" Font-Bold="True"></asp:TableCell>
                <asp:TableCell><asp:CheckBox ID="CheckBox5" runat="server"
Text="<%"$Resources:LocalizedText, CheckBox5 %>" Checked="True" /></asp:TableCell>
                <asp:TableCell><asp:CheckBox ID="CheckBox6" runat="server"
Text="<%"$Resources:LocalizedText, CheckBox6 %>" Checked="True" /></asp:TableCell>
                <asp:TableCell><asp:CheckBox ID="CheckBox7" runat="server"
Text="<%"$Resources:LocalizedText, CheckBox7 %>" Checked="True" /></asp:TableCell>
                <asp:TableCell><asp:CheckBox ID="CheckBox8" runat="server"
Text="<%"$Resources:LocalizedText, CheckBox8 %>" Checked="True" /></asp:TableCell>
                <asp:TableCell BorderStyle="None"><asp:Button ID="aplicar"
runat="server" Text="<%"$Resources:LocalizedText, aplicar %>"
OnClick="Button2_Click"/></asp:TableCell></asp:TableRow>
        </asp:Table>
    </div>

    <br />

    <script type="text/javascript" src="Scripts/jquery-1.6.1.min.js"></script>
    <script type="text/javascript" src="Scripts/arbor.js"></script>
    <script type="text/javascript" src="Scripts/arbor-tween.js"></script>
    <script type="text/javascript" src="Scripts/arbor-graphics.js"></script>
    <script type="text/javascript" src="Scripts/site.js"></script>

    <ajaxToolkit:ToolkitScriptManager ID="sm1" runat="server"
EnableHistory="true"
EnableSecureHistoryState="false"></ajaxToolkit:ToolkitScriptManager>
    <ajaxToolkit:TabContainer ID="TabContainer1" runat="server"
ActiveTabIndex="0" AutoPostBack="false"
Height="40em" Width="108em" style="margin-right: 15px" ScrollBars="Both"
Visible="false">
        <ajaxToolkit:TabPanel runat="server" HeaderText="TabPanel1"
ID="TabPanel1" OnClientClick="cambio1">
            <ContentTemplate>
                <canvas height="522" width="1048" id="sitemap0"
class=""></canvas>
            </ContentTemplate>
        </ajaxToolkit:TabPanel>
        <ajaxToolkit:TabPanel ID="TabPanel2" runat="server"
HeaderText="TabPanel2" Visible="False" OnClientClick="cambio2">
            <ContentTemplate>
                <canvas height="522" width="1048" id="sitemap1"
class=""></canvas>
            </ContentTemplate>
        </ajaxToolkit:TabPanel>
        <ajaxToolkit:TabPanel ID="TabPanel3" runat="server"
HeaderText="TabPanel3" Visible="False" OnClientClick="cambio3">
            <ContentTemplate>
                <canvas height="522" width="1048" id="sitemap2"
class=""></canvas>
            </ContentTemplate>
        </ajaxToolkit:TabPanel>
        <ajaxToolkit:TabPanel ID="TabPanel4" runat="server"
HeaderText="TabPanel4" Visible="False" OnClientClick="cambio4">
            <ContentTemplate>
                <canvas height="522" width="1048" id="sitemap3"
class=""></canvas>
            </ContentTemplate>
        </ajaxToolkit:TabPanel>
    </ajaxToolkit:TabContainer>

```

```

        <ajaxToolkit:TabPanel ID="TabPanel5" runat="server"
HeaderText="TabPanel5" Visible="False" OnClientClick="cambio5">
        <ContentTemplate>
            <canvas height="522" width="1048" id="sitemap4"
class=""></canvas>
        </ContentTemplate>
    </ajaxToolkit:TabPanel>
</ajaxToolkit:TabContainer>

</asp:Content>

```

## Default.aspx.cs

```

using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Xml;
using Newtonsoft;
using System.IO;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using FamiliasMorfolexicas.es.ulpgc.dis.tip;
using System.Threading;
using System.Security.Permissions;
using System.Web.UI.HtmlControls;
using System.Globalization;

namespace FamiliasMorfolexicas
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                if (!string.IsNullOrEmpty(Request.QueryString["SearchParameters"]))
                {
                    if (!string.IsNullOrEmpty(Request.QueryString["full"]))
                    {
                        String full = Request.QueryString["full"];
                        HtmlGenericControl header =
(HtmlGenericControl)Master.FindControl("header");
                        HtmlGenericControl footerGeneral =
(HtmlGenericControl)Master.FindControl("footerGeneral");
                        if(full.Contains("1")){
                            header.Visible = false;
                            footerGeneral.Visible = false;
                            Session["PantallaCompleta"] = true;
                            Button3.Visible = false;
                            Button4.Visible = true;
                        }
                        else
                        {
                            header.Visible = true;
                            footerGeneral.Visible = true;
                            Session["PantallaCompleta"] = false;
                            Button3.Visible = true;
                        }
                    }
                }
            }
        }
    }
}

```

```

        Button4.Visible = false;
    }
}
if (!string.IsNullOrEmpty(Request.QueryString["Filters"]))
{
    String fil = Request.QueryString["Filters"];
    if (fil.Contains("1")) CheckBox1.Checked = false;
    if (fil.Contains("2")) CheckBox2.Checked = false;
    if (fil.Contains("3")) CheckBox3.Checked = false;
    if (fil.Contains("4")) CheckBox4.Checked = false;
    if (fil.Contains("5")) CheckBox5.Checked = false;
    if (fil.Contains("6")) CheckBox6.Checked = false;
    if (fil.Contains("7")) CheckBox7.Checked = false;
    if (fil.Contains("8")) CheckBox8.Checked = false;
}
palabra.Text = Request.QueryString["SearchParameters"];
if (!string.IsNullOrEmpty(Request.QueryString["relaciones"])
&& Request.QueryString["relaciones"].Contains("1"))
{
    Relaciones();
}
else if
(!string.IsNullOrEmpty(Request.QueryString["relaciones"]) &&
Request.QueryString["relaciones"].Contains("2"))
{
    Relaciones2();
}
else
{
    TabContainer1.Visible = true;
    Table1.Visible = true;
    if (!string.IsNullOrEmpty(Request.QueryString["fich"]) &&
Request.QueryString["fich"].Contains("O"))
    {
        Session["ficherosActuales"] =
Session["ficherosOriginales"];
    }
    if (!string.IsNullOrEmpty(Request.QueryString["fich"]) &&
Request.QueryString["fich"].Contains("A"))
    {
        Session["ficherosActuales"] = Session["ficherosAux"];
    }
    String[] fich =
Session["ficherosActuales"].ToString().Split(',');
    String l = Session["ficherosActuales"].ToString();
    int numFich = fich.Length;
    if (fich[0] == "")
    {
        aviso2.Visible = true;
        TabContainer1.Visible = false;
        Table1.Visible = false;
    }
    if (numFich >= 1 && fich[0] != "")
    {
        TabPanel1.HeaderText = ObtenerNombre(fich[0]);
        TabPanel1.Visible = true;
    }
    if (numFich >= 2)
    {
        TabPanel2.HeaderText = ObtenerNombre(fich[1]);
        TabPanel2.Visible = true;
    }
}
}

```

```

        if (numFich >= 3)
        {
            TabPanel3.HeaderText = ObtenerNombre(fich[2]);
            TabPanel3.Visible = true;
        }
        if (numFich >= 4)
        {
            TabPanel4.HeaderText = ObtenerNombre(fich[3]);
            TabPanel4.Visible = true;
        }
        if (numFich == 5)
        {
            TabPanel5.HeaderText = ObtenerNombre(fich[4]);
            TabPanel5.Visible = true;
        }
        ClientScript.RegisterStartupScript(Page.GetType(),
"invocafuncion", "inicializa('" + Session["ficherosActuales"] + "',' +
Session["CodigoUsuario"] + "');", true);
    }
}
}
}
}
//Obtener el nombre de la raíz para las pestañas en pantalla completa
protected String ObtenerNombre(String file)
{
    String url = Server.MapPath("~/json") + "\\" +
Session["CodigoUsuario"] + "\\" + file;
    String json = File.ReadAllText(url);
    Dictionary<String, Dictionary<String, Object>> arbol =
Newtonsoft.Json.JsonConvert.DeserializeObject<Dictionary<String,
Dictionary<String, Object>>>(json);
    foreach (KeyValuePair<String, Dictionary<String, Object>> dic in
arbol)
    {
        if (dic.Key == "nodes")
        {
            Dictionary<String, Object> datos = dic.Value;
            foreach (KeyValuePair<String, Object> dic2 in datos)
            {
                Newtonsoft.Json.Linq.JObject nodo =
(Newtonsoft.Json.Linq.JObject)dic2.Value;
                String padre = nodo.GetValue("padre").ToString();
                int forma = (int)nodo.GetValue("codFormaCano");
                if (padre == "")//Nodo raíz
                {
                    return nodo.GetValue("palabra").ToString();
                }
            }
        }
    }
    return "";
}

//historial de vistas para controlar "Atrás" en el navegador en el botón
relaciones.
protected void Button1_Click(object sender, EventArgs e)
{
    String antiguo = palabra.Text;
    String parametros = @"?SearchParameters=" + antiguo +
"&relaciones=1&fich=0";
    if ((bool)Session["PantallaCompleta"] == true)

```

```

        {
            parametros = parametros + "&full=1";
        }
        else
        {
            parametros = parametros + "&full=0";
        }

        String searchUrl = "Default.aspx" + parametros;
        Session["Parametros"] = parametros;
        Response.Redirect(searchUrl);
    }

    //historial de vistas para controlar "Atrás" en el botón aplicar.
    protected void Button2_Click(object sender, EventArgs e)
    {
        String antiguo = palabra.Text;
        String parametros = @"?SearchParameters=" + antiguo +
"&relaciones=2&fich=A";
        if ((bool)Session["PantallaCompleta"] == true)
        {
            parametros = parametros + "&full=1";
        }
        else
        {
            parametros = parametros + "&full=0";
        }
        Session["Parametros"] = parametros + Filtros();
        String searchUrl = "Default.aspx" + parametros + Filtros();
        Response.Redirect(searchUrl);
    }

    //pantalla completa
    protected void Button3_Click(object sender, EventArgs e)
    {
        Session["PantallaCompleta"] = true;
        String antiguo = palabra.Text;
        String parametros = @"?SearchParameters=" + antiguo + "&full=1";
        if (Session["ficherosAux"] != null && Session["ficherosAux"].ToString()
== Session["ficherosActuales"].ToString())
        {
            parametros = parametros + "&fich=A";
        }
        if (Session["ficherosAux"] != null &&
Session["ficherosAux"].ToString() == "")
        {
            parametros = parametros + "&fich=A";
        }
        if (Session["ficherosOriginales"].ToString() ==
Session["ficherosActuales"].ToString())
        {
            parametros = parametros + "&fich=0";
        }
        String searchUrl = "Default.aspx" + parametros + Filtros();
        Session["Parametros"] = parametros + Filtros();
        Response.Redirect(searchUrl);
    }

    //pantalla normal
    protected void Button4_Click(object sender, EventArgs e)
    {
        Session["PantallaCompleta"] = false;
    }

```

```

        String antiguo = palabra.Text;
        String parametros = @"?SearchParameters=" + antiguo + "&full=0";
        if (Session["ficherosAux"] != null &&
Session["ficherosAux"].ToString() == Session["ficherosActuales"].ToString())
        {
            parametros = parametros + "&fich=A";
        }
        if (Session["ficherosAux"] != null &&
Session["ficherosAux"].ToString() == "")
        {
            parametros = parametros + "&fich=A";
        }
        if (Session["ficherosOriginales"].ToString() ==
Session["ficherosActuales"].ToString())
        {
            parametros = parametros + "&fich=0";
        }
        String searchUrl = "Default.aspx" + parametros + Filtros();
        Session["Parametros"] = parametros + Filtros();
        Response.Redirect(searchUrl);
    }

//filtros de url
protected String Filtros()
{
    String filtros = "&Filters=";
    if (!CheckBox1.Checked)
    {
        filtros = filtros + "1";
    }
    if (!CheckBox2.Checked)
    {
        filtros = filtros + "2";
    }
    if (!CheckBox3.Checked)
    {
        filtros = filtros + "3";
    }
    if (!CheckBox4.Checked)
    {
        filtros = filtros + "4";
    }
    if (!CheckBox5.Checked)
    {
        filtros = filtros + "5";
    }
    if (!CheckBox6.Checked)
    {
        filtros = filtros + "6";
    }
    if (!CheckBox7.Checked)
    {
        filtros = filtros + "7";
    }
    if (!CheckBox8.Checked)
    {
        filtros = filtros + "8";
    }
    return filtros;
}
}

```

```

protected void Relaciones()
{
    String CodigoUsuario = (string)(Session["CodigoUsuario"]);
    //Button3.Visible = true;
    TabContainer1.Visible = true;
    Table1.Visible = true;
    TabPanel2.Visible = false;
    TabPanel3.Visible = false;
    TabPanel4.Visible = false;
    TabPanel5.Visible = false;
    error.Visible = false;
    aviso.Visible = false;
    aviso2.Visible = false;
    CheckBox1.Checked = true;
    CheckBox2.Checked = true;
    CheckBox3.Checked = true;
    CheckBox4.Checked = true;
    CheckBox5.Checked = true;
    CheckBox6.Checked = true;
    CheckBox7.Checked = true;
    CheckBox8.Checked = true;
    FamiliasMorfolexicas.es.ulpgc.dis.tip.Lematizador x = new
Lematizador();
    Object[] p = x.Reconocer(palabra.Text, "es", false);
    int len = p.Length;
    if (len == 0)
    {
        error.Visible = true;
        TabContainer1.Visible = false;
        Table1.Visible = false;
        return;
    }
    int i = 1;
    if (p.Length == 0) return;
    else
    {
        //Este listado es para comprobar que no se repite el codigo de
formacanonica, porque Reconocer puede devolver
//varias iguales, que solo se diferencian en el código de
flexión.
        List<int> formascanonicas = new List<int>();
        String ficheros = "";
        foreach (Object formas in p)
        {
            Object[] vec = (Object[])formas;
            if (formascanonicas.Count != 0 &&
formascanonicas.Contains((int)vec[1])) { }
            else
            {
                String nombre = "" + vec[1] + ".json";
                //ficheros = ficheros + nombre + ",";
                formascanonicas.Add((int)vec[1]);

                String url = Server.MapPath("~/json") + "\\\" +
CodigoUsuario + "\\\" + nombre;
                if (!File.Exists("~/json/" + nombre))
                {
                    if (formarGrafo(vec, nombre, i))
                    {
                        ficheros = ficheros + nombre + ",";
                        if (i == 2)
                        {

```



```

        TabPanel2.Visible = true;
    }
    if (i == 3)
    {
        TabPanel3.Visible = true;
    }
    if (i == 4)
    {
        TabPanel4.Visible = true;
    }
    if (i == 5)
    {
        TabPanel5.Visible = true;
    }
    i++;
}
}
}
}
}
//Elimina el caracter final que es una coma.
ficheros = ficheros.TrimEnd(',');
Session["ficherosOriginales"] = ficheros;
Session["ficherosActuales"] = ficheros;
ClientScript.RegisterStartupScript(Page.GetType(),
"invocafuncion", "inicializa('" + ficheros + "',' + CodigoUsuario + "',');",
true);
}
}

protected void Relaciones2()
{
    TabContainer1.Visible = true;
    //Button3.Visible = true;
    Table1.Visible = true;
    TabPanel2.Visible = false;
    TabPanel3.Visible = false;
    TabPanel4.Visible = false;
    TabPanel5.Visible = false;
    error.Visible = false;
    aviso.Visible = false;
    aviso2.Visible = false;
    TabContainer1.ActiveTabIndex = 0;
    String sesion = Session["CodigoUsuario"].ToString();
    String ficheros = Session["ficherosOriginales"].ToString();
    String ficherosAux = "";
    String[] nombreFichero = ficheros.Split(',');
    int i = 1;
    foreach (String nombre in nombreFichero)
    {
        if (i == 1)
        {
            TabPanel1.Visible = true;
        }
        if (i == 2)
        {
            TabPanel2.Visible = true;
        }
        if (i == 3)
        {
            TabPanel3.Visible = true;
        }
        if (i == 4)

```

```

        {
            TabPanel4.Visible = true;
        }
        if (i == 5)
        {
            TabPanel5.Visible = true;
        }
        String nombreAux = nombre.Substring(0, nombre.IndexOf(".")) + "-
aux.json";
        ficherosAux = ficherosAux + nombreAux + ",";
        String url = Server.MapPath("~/json") + "\\\" + sesion + "\\\" +
nombre;
        String json = File.ReadAllText(url);
        Dictionary<String, Dictionary<String, Object>> arbol =
Newtonsoft.Json.JsonConvert.DeserializeObject<Dictionary<String,
Dictionary<String, Object>>>(json);
        Dictionary<String, Object> nodos = new Dictionary<String,
Object>();
        foreach (KeyValuePair<String, Dictionary<String, Object>> dic in
arbol)
        {
            if (dic.Key == "nodes")
            {
                Dictionary<String, Object> datos = dic.Value;
                foreach (KeyValuePair<String, Object> dic2 in datos)
                {
                    Newtonsoft.Json.Linq.JObject nodo =
(Newtonsoft.Json.Linq.JObject)dic2.Value;
                    String padre = nodo.GetValue("padre").ToString();
                    int forma = (int)nodo.GetValue("codFormaCano");
                    if (padre == "")//Nodo raíz
                    {
                        nodos.Add(dic2.Key, dic2.Value);
                        if (i == 1)
                        {
                            TabPanel1.HeaderText =
nodo.GetValue("palabra").ToString();
                        }
                        if (i == 2)
                        {
                            TabPanel2.HeaderText =
nodo.GetValue("palabra").ToString();
                        }
                        if (i == 3)
                        {
                            TabPanel3.HeaderText =
nodo.GetValue("palabra").ToString();
                        }
                        if (i == 4)
                        {
                            TabPanel4.HeaderText =
nodo.GetValue("palabra").ToString();
                        }
                        if (i == 5)
                        {
                            TabPanel5.HeaderText =
nodo.GetValue("palabra").ToString();
                        }
                    }
                    //Bloque para comprobar si estan checkeadas las
familias de palabras(Sus, adj, adv, ver).
                    //Sustantivos.
                }
            }
        }
    }
}

```

```

        Boolean metido = false;
        if (CheckBox5.Checked && forma >= 1000 && forma <=
1099 && (nodos.ContainsKey(nodo.GetValue("padre").ToString()))
        {
            //Bloque para comprobar si estan checkeados
sufijos, prefijos, parasintéticas u otros.
            if (CheckBox4.Checked &&
!nodo.GetValue("codSuf").ToString().StartsWith("-") &&
nodo.GetValue("codSuf").ToString() != "" && metido == false)
            {
                nodos.Add(dic2.Key, dic2.Value);
                metido = true;
            }
            if (CheckBox3.Checked &&
nodo.GetValue("codSuf").ToString() != "" && nodo.GetValue("codPre").ToString() !=
"" && metido == false)
            {
                nodos.Add(dic2.Key, dic2.Value);
                metido = true;
            }
            if (CheckBox1.Checked &&
nodo.GetValue("codSuf").ToString() != "" && nodo.GetValue("codPre").ToString() ==
"" && nodo.GetValue("codSuf").ToString().StartsWith("-") && metido == false)
            {
                nodos.Add(dic2.Key, dic2.Value);
                metido = true;
            }
            if (CheckBox2.Checked &&
nodo.GetValue("codPre").ToString() != "" && nodo.GetValue("codSuf").ToString() ==
"" && metido == false)
            {
                nodos.Add(dic2.Key, dic2.Value);
                metido = true;
            }
        }
        //Adjetivos
        metido = false;
        if (CheckBox6.Checked && forma >= 1100 && forma <=
1199 && (nodos.ContainsKey(nodo.GetValue("padre").ToString()))
        {
            //Bloque para comprobar si estan checkeados
sufijos, prefijos, parasintéticas u otros.
            if (CheckBox4.Checked &&
!nodo.GetValue("codSuf").ToString().StartsWith("-") &&
nodo.GetValue("codSuf").ToString() != "" && metido == false)
            {
                nodos.Add(dic2.Key, dic2.Value);
                metido = true;
            }
            if (CheckBox3.Checked &&
nodo.GetValue("codSuf").ToString() != "" && nodo.GetValue("codPre").ToString() !=
"" && metido == false)
            {
                nodos.Add(dic2.Key, dic2.Value);
                metido = true;
            }
            if (CheckBox1.Checked &&
nodo.GetValue("codSuf").ToString() != "" && nodo.GetValue("codPre").ToString() ==
"" && nodo.GetValue("codSuf").ToString().StartsWith("-") && metido == false)
            {
                nodos.Add(dic2.Key, dic2.Value);
                metido = true;
            }
        }

```

```

    }
    if (CheckBox2.Checked &&
nodo.GetValue("codPre").ToString() != "" && nodo.GetValue("codSuf").ToString() ==
"" && metido == false)
    {
        nodos.Add(dic2.Key, dic2.Value);
        metido = true;
    }
}
//Adverbios
metido = false;
if (CheckBox7.Checked && forma >= 1200 && forma <=
1299 && (nodos.ContainsKey(nodo.GetValue("padre").ToString())))
{
    //Bloque para comprobar si estan checkeados
sufijos, prefijos, parasintéticas u otros.
    if (CheckBox4.Checked &&
!nodo.GetValue("codSuf").ToString().StartsWith("-") &&
nodo.GetValue("codSuf").ToString() != "" && metido == false)
    {
        nodos.Add(dic2.Key, dic2.Value);
        metido = true;
    }
    if (CheckBox3.Checked &&
nodo.GetValue("codSuf").ToString() != "" && nodo.GetValue("codPre").ToString() !=
"" && metido == false)
    {
        nodos.Add(dic2.Key, dic2.Value);
        metido = true;
    }
    if (CheckBox1.Checked &&
nodo.GetValue("codSuf").ToString() != "" && nodo.GetValue("codPre").ToString() ==
"" && nodo.GetValue("codSuf").ToString().StartsWith("-") && metido == false)
    {
        nodos.Add(dic2.Key, dic2.Value);
        metido = true;
    }
    if (CheckBox2.Checked &&
nodo.GetValue("codPre").ToString() != "" && nodo.GetValue("codSuf").ToString() ==
"" && metido == false)
    {
        nodos.Add(dic2.Key, dic2.Value);
        metido = true;
    }
}
//Verbos
metido = false;
if (CheckBox8.Checked && forma >= 3000 && forma <=
3199 && (nodos.ContainsKey(nodo.GetValue("padre").ToString())))
{
    //Bloque para comprobar si estan checkeados
sufijos, prefijos, parasintéticas u otros.
    if (CheckBox4.Checked &&
!nodo.GetValue("codSuf").ToString().StartsWith("-") &&
nodo.GetValue("codSuf").ToString() != "" && metido == false)
    {
        nodos.Add(dic2.Key, dic2.Value);
        metido = true;
    }
    if (CheckBox3.Checked &&
nodo.GetValue("codSuf").ToString() != "" && nodo.GetValue("codPre").ToString() !=
"" && metido == false)

```

```

        {
            nodos.Add(dic2.Key, dic2.Value);
            metido = true;
        }
        if (CheckBox1.Checked &&
nodo.GetValue("codSuf").ToString() != "" && nodo.GetValue("codPre").ToString() ==
"" && nodo.GetValue("codSuf").ToString().StartsWith("-") && metido == false)
        {
            nodos.Add(dic2.Key, dic2.Value);
            metido = true;
        }
        if (CheckBox2.Checked &&
nodo.GetValue("codPre").ToString() != "" && nodo.GetValue("codSuf").ToString() ==
"" && metido == false)
        {
            nodos.Add(dic2.Key, dic2.Value);
            metido = true;
        }
    }
}
}
}
}
}
i++;
//Comprueba qsi al aplicar los filtros la palabra tiene familia o
no.
if (nodos.Count == 1)
{
    aviso2.Visible = true;
    TabContainer1.Visible = false;
    Table1.Visible = false;
    Session["ficherosAux"] = "";
    return;
}
Dictionary<String, Object> edges = new Dictionary<String,
Object>();
foreach (KeyValuePair<String, Dictionary<String, Object>> dic in
arbol)
{
    if (dic.Key == "nodes")
    {
        Dictionary<String, Object> datos = dic.Value;
        foreach (KeyValuePair<String, Object> dic2 in datos)
        {
            if (!edges.ContainsKey(dic2.Key))
            {
                Dictionary<String, Object> edgesData = new
Dictionary<String, Object>();
                foreach (KeyValuePair<String, Object> dic3 in
datos)
                {
                    Newtonsoft.Json.Linq.JObject nodo =
(Newtonsoft.Json.Linq.JObject)dic3.Value;
                    String padre =
nodo.GetValue("padre").ToString();
                    int forma =
(int)nodo.GetValue("codFormaCano");
                    //Bloque para comprobar si estan marcados los
checkbox (Sustantivo, Adjetivo, Adverbio, Verbo).
                    if (padre == dic2.Key)
                    {
                        String sufijo = "";
                        switch ((int)nodo.GetValue("codOtros"))

```

```

{
    case 1:
        sufijo = "Trans.";
        break;
    case 7:
        sufijo = "Afer.";
        break;
    case 15:
        sufijo = "Alt.";
        break;
    case 24:
        sufijo = "Apoc.";
        break;
    case 46:
        sufijo = "Alt.";
        break;
    case 134:
        sufijo = "Irreg.";
        break;
    case 158:
        sufijo = "Met.";
        break;
    case 185:
        sufijo = "Supr.";
        break;
    case 191:
        sufijo = "Verb.";
        break;
}
Boolean metido = false;
//Sustantivo
if (CheckBox5.Checked && forma >= 1000 &&
forma <= 1099 && (nodos.ContainsKey(nodo.GetValue("padre").ToString()))
{
    //Bloque para comprobar si estan
checkeados sufijos, prefijos, parasintéticas u otros.
    if (CheckBox4.Checked &&
!nodo.GetValue("codSuf").ToString().StartsWith("-") &&
nodo.GetValue("codSuf").ToString() != "" && metido == false)
    {
        edgesData.Add(dic3.Key, new
Data(sufijo, nodo.GetValue("codPre").ToString()));
        metido = true;
    }
    if (CheckBox3.Checked &&
nodo.GetValue("codSuf").ToString() != "" && nodo.GetValue("codPre").ToString() !=
"" && metido == false)
    {
        edgesData.Add(dic3.Key, new
Data(nodo.GetValue("codSuf").ToString(), nodo.GetValue("codPre").ToString()));
        metido = true;
    }
    if (CheckBox1.Checked &&
nodo.GetValue("codSuf").ToString() != "" && nodo.GetValue("codPre").ToString() ==
"" && nodo.GetValue("codSuf").ToString().StartsWith("-") && metido == false)
    {
        edgesData.Add(dic3.Key, new
Data(nodo.GetValue("codSuf").ToString(), nodo.GetValue("codPre").ToString()));
        metido = true;
    }
}

```

```

        if (CheckBox2.Checked &&
nodo.GetValue("codPre").ToString() != "" && nodo.GetValue("codSuf").ToString() ==
"" && metido == false)
        {
            edgesData.Add(dic3.Key, new
Data(nodo.GetValue("codSuf").ToString(), nodo.GetValue("codPre").ToString()));
            metido = true;
        }
    }
    metido = false;
    //Adjetivo
    if (CheckBox6.Checked && forma >= 1100 &&
forma <= 1199 && (nodos.ContainsKey(nodo.GetValue("padre").ToString())))
    {
        //Bloque para comprobar si estan
checkeados sufijos, prefijos, parasintéticas u otros.
        if (CheckBox4.Checked &&
!nodo.GetValue("codSuf").ToString().StartsWith("-") &&
nodo.GetValue("codSuf").ToString() != "" && metido == false)
        {
            edgesData.Add(dic3.Key, new
Data(sufijo, nodo.GetValue("codPre").ToString()));
            metido = true;
        }
        if (CheckBox3.Checked &&
nodo.GetValue("codSuf").ToString() != "" && nodo.GetValue("codPre").ToString() !=
"" && metido == false)
        {
            edgesData.Add(dic3.Key, new
Data(nodo.GetValue("codSuf").ToString(), nodo.GetValue("codPre").ToString()));
            metido = true;
        }
        if (CheckBox1.Checked &&
nodo.GetValue("codSuf").ToString() != "" && nodo.GetValue("codPre").ToString() ==
"" && nodo.GetValue("codSuf").ToString().StartsWith("-") && metido == false)
        {
            edgesData.Add(dic3.Key, new
Data(nodo.GetValue("codSuf").ToString(), nodo.GetValue("codPre").ToString()));
            metido = true;
        }
        if (CheckBox2.Checked &&
nodo.GetValue("codPre").ToString() != "" && nodo.GetValue("codSuf").ToString() ==
"" && metido == false)
        {
            edgesData.Add(dic3.Key, new
Data(nodo.GetValue("codSuf").ToString(), nodo.GetValue("codPre").ToString()));
            metido = true;
        }
    }
    metido = false;
    //Adverbio
    if (CheckBox7.Checked && forma >= 1200 &&
forma <= 1299 && (nodos.ContainsKey(nodo.GetValue("padre").ToString())))
    {
        //Bloque para comprobar si estan
checkeados sufijos, prefijos, parasintéticas u otros.
        if (CheckBox4.Checked &&
!nodo.GetValue("codSuf").ToString().StartsWith("-") &&
nodo.GetValue("codSuf").ToString() != "" && metido == false)
        {
            edgesData.Add(dic3.Key, new
Data(sufijo, nodo.GetValue("codPre").ToString()));

```

```

        metido = true;
    }
    if (CheckBox3.Checked &&
nodo.GetValue("codSuf").ToString() != "" && nodo.GetValue("codPre").ToString() !=
"" && metido == false)
    {
        edgesData.Add(dic3.Key, new
Data(nodo.GetValue("codSuf").ToString(), nodo.GetValue("codPre").ToString()));
        metido = true;
    }
    if (CheckBox1.Checked &&
nodo.GetValue("codSuf").ToString() != "" && nodo.GetValue("codPre").ToString() ==
"" && nodo.GetValue("codSuf").ToString().StartsWith("-") && metido == false)
    {
        edgesData.Add(dic3.Key, new
Data(nodo.GetValue("codSuf").ToString(), nodo.GetValue("codPre").ToString()));
        metido = true;
    }
    if (CheckBox2.Checked &&
nodo.GetValue("codPre").ToString() != "" && nodo.GetValue("codSuf").ToString() ==
"" && metido == false)
    {
        edgesData.Add(dic3.Key, new
Data(nodo.GetValue("codSuf").ToString(), nodo.GetValue("codPre").ToString()));
        metido = true;
    }
    }
    metido = false;
    //Verbo
    if (CheckBox8.Checked && forma >= 3000 &&
forma <= 3199 && (nodos.ContainsKey(nodo.GetValue("padre").ToString()))
    {
        //Bloque para comprobar si estan
checkeados sufijos, prefijos, parasintéticas u otros.
        if (CheckBox4.Checked &&
!nodo.GetValue("codSuf").ToString().StartsWith("-") &&
nodo.GetValue("codSuf").ToString() != "" && metido == false)
        {
            edgesData.Add(dic3.Key, new
Data(sufijo, nodo.GetValue("codPre").ToString()));
            metido = true;
        }
        if (CheckBox3.Checked &&
nodo.GetValue("codSuf").ToString() != "" && nodo.GetValue("codPre").ToString() !=
"" && metido == false)
        {
            edgesData.Add(dic3.Key, new
Data(nodo.GetValue("codSuf").ToString(), nodo.GetValue("codPre").ToString()));
            metido = true;
        }
        if (CheckBox1.Checked &&
nodo.GetValue("codSuf").ToString() != "" && nodo.GetValue("codPre").ToString() ==
"" && nodo.GetValue("codSuf").ToString().StartsWith("-") && metido == false)
        {
            edgesData.Add(dic3.Key, new
Data(nodo.GetValue("codSuf").ToString(), nodo.GetValue("codPre").ToString()));
            metido = true;
        }
        if (CheckBox2.Checked &&
nodo.GetValue("codPre").ToString() != "" && nodo.GetValue("codSuf").ToString() ==
"" && metido == false)
        {

```



```

            edgesData.Add(dic3.Key, new
Data(nodo.GetValue("codSuf").ToString(), nodo.GetValue("codPre").ToString()));
                metodo = true;
            }
        }
    }
    if (edgesData.Count != 0)
    {
        edges.Add(dic2.Key, edgesData);
    }
}
}
}
Dictionary<String, Object> arbol2 = new Dictionary<String,
Object>
{
    {"nodes", nodos},
    {"edges", edges}
};

String json2 =
Newtonsoft.Json.JsonConvert.SerializeObject(arbol2,
Newtonsoft.Json.Formatting.Indented); //Librería Newtonsoft.Json.
url = url.Substring(0, url.Length - 5);
url = url + "-aux.json";
File.WriteAllText(url, json2);
}
//Elimina el caracter final que es una coma.
ficherosAux = ficherosAux.TrimEnd(',');
if (ficheros.Length != 0)
{
    Session["ficherosAux"] = ficherosAux;
    Session["ficherosActuales"] = ficherosAux;
}
String CodigoUsuario = (String)Session["CodigoUsuario"];
ClientScript.RegisterStartupScript(Page.GetType(), "invocafuncion",
"inicializa('" + ficherosAux + "',' + CodigoUsuario + '");", true);
}

//*****Procedimiento formaGrafo, que crea el
arbol.*****//
private Boolean formarGrafo(Object[] vec, String nombreFichero, int
indice)
{
    FamiliasMorfolexicas.es.ulpgc.dis.tip.Lematizador x = new
Lematizador();
    String canonica = vec[1].ToString();
    //String flexion = vec[2].ToString();

    Object[] p2 = x.Consulta(canonica, "es", "canonicas");
    Object[] vec2 = (Object[])p2[0];
    String formacanonica = vec2[0].ToString();

    if (indice == 1)
    {
        TabPanel1.HeaderText = formacanonica;
    }
    if (indice == 2)
    {
        TabPanel2.HeaderText = formacanonica;
    }
}

```

```

    }
    if (indice == 3)
    {
        TabPanel13.HeaderText = formacanonica;
    }
    if (indice == 4)
    {
        TabPanel14.HeaderText = formacanonica;
    }
    if (indice == 5)
    {
        TabPanel15.HeaderText = formacanonica;
    }
    FamiliasMorfolexicas.es.ulpgc.dis.tip1.Familias y = new
es.ulpgc.dis.tip1.Familias();
    Object[] familia = y.Consulta(formacanonica, "es", "familias");
    if (familia.Length == 0)
    {
        aviso.Visible = true;
        TabContainer1.Visible = false;
        Table1.Visible = false;
        return false;
    }

    //La raiz del árbol (que se corresponde con la formacanonica) tiene
el id=0 SIEMPRE
    Raiz pr = new Raiz("#BFE7FE", "dot", 1, formacanonica, "", "", 0, "",
"#0000FF", 0);
    //Lista todos los nodos a partir de la raíz, hijos y nietos.
    Dictionary<String, Object> nodos = new Dictionary<String, Object>
    {
        {"0", pr}
    };
    //Diccionario para listar los enlaces a los hijos
    Dictionary<String, Object> edges = new Dictionary<String, Object>();
    int i = 1, j = 100;
    foreach (Object contador in familia)
    {
        Object[] elementos = (Object[])contador;
        nodos.Add("" + i, new Raiz("#660066", "dot", 1,
elementos[2].ToString(), elementos[5].ToString(), elementos[8].ToString(),
(int)elementos[3], "0", "#FFFFFF", (int)elementos[4]));
        //busca los hijos de los hijos
        Object[] nietos = y.Consulta(elementos[2].ToString(), "es",
"familias");
        int len = nietos.Length;
        if (len != 0)
        {
            //Diccionario para listar los enlaces a los nodos nietos.
            Dictionary<String, Object> edgesDatah = new
Dictionary<String, Object>();
            foreach (Object contador2 in nietos)
            {
                Object[] elementos2 = (Object[])contador2;
                nodos.Add("" + j, new Raiz("#660066", "", 0,
elementos2[2].ToString(), elementos2[5].ToString(), elementos2[8].ToString(),
(int)elementos2[3], i.ToString(), "#FFFFFF", (int)elementos2[4]));
                //Comprueba que estén marcados los prefijos y sufijos
para añadirlos a los enlaces(edges).
                String sufijo = "";
                String prefijo = "";
                if (CheckBox2.Checked)

```

```

        {
            prefijo = elementos2[8].ToString();
        }
        if (CheckBox1.Checked &&
elementos2[5].ToString().StartsWith("-"))
        {
            sufijo = elementos2[5].ToString();
        }
        if (CheckBox3.Checked)
        {
            prefijo = elementos2[8].ToString();
            sufijo = elementos2[5].ToString();
        }
        if (CheckBox4.Checked)
        {
            switch ((int)elementos2[4])
            {
                case 1:
                    sufijo = "Trans.";
                    break;
                case 7:
                    sufijo = "Afer.";
                    break;
                case 15:
                    sufijo = "Alt.";
                    break;
                case 24:
                    sufijo = "Apoc.";
                    break;
                case 46:
                    sufijo = "Alt.";
                    break;
                case 134:
                    sufijo = "Irreg.";
                    break;
                case 158:
                    sufijo = "Met.";
                    break;
                case 185:
                    sufijo = "Supr.";
                    break;
                case 191:
                    sufijo = "Verb.";
                    break;
            }
            Data data = new Data(sufijo, prefijo);
            edgesDataah.Add("" + j, data);
            j++;
        }
        edges.Add("" + i, edgesDataah);
    }
    i++;
}
//Diccionario para listar las propiedades que tiene cada nodo.
Dictionary<String, Object> edgesData = new Dictionary<String,
Object>();
i = 1;
foreach (Object contador in familia)
{
    Object[] elementos = (Object[])contador;

```

```

        //Comprueba que esten marcados los prefijos y sufijos para
añadirlos a los enlaces(edges).
String sufijo = "";
String prefijo = "";
if (CheckBox2.Checked)
{
    prefijo = elementos[8].ToString();
}
if (CheckBox1.Checked && elementos[5].ToString().StartsWith("-"))
{
    sufijo = elementos[5].ToString();
}
if (CheckBox3.Checked)
{
    prefijo = elementos[8].ToString();
    sufijo = elementos[5].ToString();
}
if (CheckBox4.Checked)
{
    //Case para cambiar las etiquetas de los enlaces por
palabras más cortas que se visualicen mejor.
    switch ((int)elementos[4])
    {
        case 1:
            sufijo = "Trans.";
            break;
        case 7:
            sufijo = "Afer.";
            break;
        case 15:
            sufijo = "Alt.";
            break;
        case 24:
            sufijo = "Apoc.";
            break;
        case 46:
            sufijo = "Alt.";
            break;
        case 134:
            sufijo = "Irreg.";
            break;
        case 158:
            sufijo = "Met.";
            break;
        case 185:
            sufijo = "Supr.";
            break;
        case 191:
            sufijo = "Verb.";
            break;
    }
}
Data data = new Data(sufijo, prefijo);
edgesData.Add("" + i, data);
i++;
}
//Elemento raíz, que se introduce en último lugar.
edges.Add("0", edgesData);
//Dictionary que lista el arbol con todas sus propiedades (Raiz,
hijos y nietos) y las propiedades de cada uno.
Dictionary<String, Object> arbol = new Dictionary<String, Object>
{

```

```

        {"nodes", nodos},
        {"edges", edges}
    };
    //Tratamiento de la clase Dictionary para convertir en archivo json.
    String json = Newtonsoft.Json.JsonConvert.SerializeObject(arbol,
Newtonsoft.Json.Formatting.Indented); //Librería Newtonsoft.Json.
    String url = Server.MapPath("~/json/" + Session["CodigoUsuario"]) +
"\\" + nombreFichero;
    File.WriteAllText(url, json);
    return true;
}

//*****Clases que usa el procedimiento
formaGrafo.*****//
public class Raiz
{
    public String color { get; set; }
    public String shape { get; set; }
    public String palabra { get; set; }
    public int alpha { get; set; }
    public String codSuf { get; set; }
    public String codPre { get; set; }
    public String padre { get; set; }
    public int codFormaCano { get; set; }
    public String colorletra { get; set; }
    public int codOtros { get; set; }
    public Raiz() { }
    public Raiz(String colorA, String shapeA, int alphaA, String
palabraA, String codSufA, String codPreA, int codFormaCanoA, String padreA,
String colorletraA, int codOtrosA)
    {
        color = colorA;
        shape = shapeA;
        alpha = alphaA;
        palabra = palabraA;
        codSuf = codSufA;
        codPre = codPreA;
        codFormaCano = codFormaCanoA;
        padre = padreA;
        colorletra = colorletraA;
        codOtros = codOtrosA;
    }
};

public class Data
{
    public String suf { get; set; }
    public String pre { get; set; }
    public int length { get; set; }

    public Data() { }
    public Data(String dataA, String preA)
    {
        suf = dataA;
        pre = preA;
        length = 1;
    }
};
}
}
}

```

## Global.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.SessionState;
using System.IO;
using System.Threading;

namespace FamiliasMorfolexicas
{
    public class Global : System.Web.HttpApplication
    {
        private static List<String> _sessionInfo;
        private static readonly object padlock = new object();

        public static List<string> Sessions
        {
            get
            {
                lock (padlock)
                {
                    if (_sessionInfo == null)
                    {
                        _sessionInfo = new List<string>();
                    }
                    return _sessionInfo;
                }
            }
        }

        //Código de prueba para aplicación multi-lenguaje
        protected void Application_BeginRequest(object sender, EventArgs e)
        {
            HttpCookie cookie = Request.Cookies["CultureInfo"];
            if (cookie != null && cookie.Value != null)
            {
                Thread.CurrentThread.CurrentUICulture = new
                System.Globalization.CultureInfo(cookie.Value);
                Thread.CurrentThread.CurrentCulture = new
                System.Globalization.CultureInfo(cookie.Value);
            }else{
                Thread.CurrentThread.CurrentUICulture = new
                System.Globalization.CultureInfo("es-ES");
                Thread.CurrentThread.CurrentCulture = new
                System.Globalization.CultureInfo("es-ES");
            }
        }

        void Application_Start(object sender, EventArgs e)
        {
            // Código que se ejecuta al iniciarse la aplicación
            Application["UsuariosConectados"] = 0;
            Application["RutaRepositorio"] = Server.MapPath("~/") + @"\json\";
        }

        void Application_End(object sender, EventArgs e)
        {
            // Código que se ejecuta cuando se cierra la aplicación
        }
    }
}
```

```

void Application_Error(object sender, EventArgs e)
{
    // Código que se ejecuta al producirse un error no controlado
}

void Session_Start(object sender, EventArgs e)
{
    // Código que se ejecuta cuando se inicia una nueva sesión
    Application["UsuariosConectados"] =
(int)Application["UsuariosConectados"] + 1;
    Session["CodigoUsuario"] = Session.SessionID;
    String CodigoUsuario = (string)(Session["CodigoUsuario"]);
    String url = Server.MapPath("~/json/" + Session["CodigoUsuario"]);
    Directory.CreateDirectory(url);
    Sessions.Add(Session.SessionID);
    Session["PantallaCompleta"] = false;
}

void Session_End(object sender, EventArgs e)
{
    Sessions.Remove(Session.SessionID);
    String url = Application["RutaRepositorio"].ToString();
    List<String> ss = Sessions;
    //Elimina el directorio utilizado por el usuario en su sesion.
    String[] carpetas = Directory.GetDirectories(url);
    foreach (String carpeta in carpetas)
    {
        String nombresS = carpeta.Substring(carpeta.LastIndexOf("\\") +
1);
        if (!Global.Sessions.Contains(nombresS))
        {
            Directory.Delete(carpeta, true);
        }
    }
    Application["UsuariosConectados"] =
(int)Application["UsuariosConectados"] - 1;
}
}
}

```

## Site.js

```

//
// site.js
//
// the arbor.js website
//
function cambio1() {
    redibuja(0);
}

function cambio2() {
    redibuja(1);
}

function cambio3() {
    redibuja(2);
}

```

```

function cambio4() {
    redibuja(3);
}

function cambio5() {
    redibuja(4);
}

//Variables sitemap y global para usarlar en la funcion redibuja y crear los
grafos en cada pestaña.
var url;
var ids;
//variable para doble click
var clicks = 0;

//Se llama desde Relaciones para obtener las variables que dicen que fiche y
dirección hay que mirar.
function inicializa(ficheros, session) {
    url = session;
    ids = ficheros.split(',');
    redibuja(0);
}

//Añade la pestaña que se van visualizar a la dirección del archivos y llama a
redibuja.
function redibuja(x) {
    var global = url + "/" + ids[x];
    var site = "#sitemap" + x
    redibujaCanvas(site, global);
}

//Recibe el archivo y la pestaña donde dibujarlo y utiliza las funciones de la
librería para crear el grafo.
function redibujaCanvas(sitemap, file){

    var Renderer = function (elt) {
        console.log('elt=' + sitemap);
        var dom = $(sitemap); //elt
        var canvas = dom.get(0);
        var ctx = canvas.getContext("2d");
        var gfx = arbor.Graphics(canvas);
        var sys = null;
        var _vignette = null;
        var selected = null,
            nearest = null,
            _mouseP = null;

        var that = {
            init: function (pSystem) {
                sys = pSystem;
                sys.screen({ size: { width: dom.width(), height: dom.height() },
padding: [36, 60, 36, 60] });

                $(window).resize(that.resize);
                that.resize();
                that._initMouseHandling();

            },
            resize:function(){
                canvas.width = $(window).width();
                canvas.height = .75* $(window).height();
            }
        };
    };
}

```



```

sys.screen({size:{width:canvas.width, height:canvas.height}});
_vignette = null;
that.redraw();
},
redraw: function () {
    gfx.clear()
    sys.eachEdge(function (edge, p1, p2) {
        if (edge.source.data.alpha * edge.target.data.alpha == 0)
return;
        gfx.line(p1, p2, { stroke: "#b2b19d", width: 2, alpha:
edge.target.data.alpha });
        //Se escribe el label correspondiente a cada edge encima de
la línea

        ctx.strokeStyle = "rgba(0,0,0, .333)";
        ctx.lineWidth = 1;
        ctx.beginPath();
        ctx.moveTo(p1.x, p1.y);
        ctx.lineTo(p2.x, p2.y);
        ctx.stroke();
        ctx.font = 'italic 13px sans-serif';
        if (edge.data.suf !== "") {
            ctx.fillStyle = "black";
            ctx.fillText(edge.data.suf, (p1.x + p2.x) / 2, (p1.y +
p2.y) / 2);
        }
        if (edge.data.pre !== "") {
            ctx.fillStyle = "green";
            var pos = 0;
            if (edge.data.suf !== "") { //Si no es nulo, se coloca el
prefijo arriba del sufijo
                pos = 2.22;
            }
            else { //Si no hay sufijo, se coloca en su sitio correcto
                pos = 2;
            }
            ctx.fillText(edge.data.pre, (p1.x + p2.x) / 2, ((p1.y +
p2.y)) / pos);
        }
    });
    sys.eachNode(function (node, pt) {
        var w = Math.max(15, 15 + gfx.textWidth(node.data.palabra))
        if (node.data.alpha === 0) return;
        if (node.data.shape == 'dot') {
            gfx.oval(pt.x - w / 2, pt.y - w / 2, w, w, { fill:
node.data.color, alpha: node.data.alpha });
            gfx.text(node.data.palabra, pt.x, pt.y + 7, { color:
node.data.colorletra, align: "center", font: "Verdana", size: 12 });
        } else {
            gfx.rect(pt.x - w / 2, pt.y - 8, w, 20, 4, { fill:
node.data.color, alpha: node.data.alpha });
            gfx.text(node.data.palabra, pt.x, pt.y + 9, { color:
node.data.colorletra, align: "center", font: "Verdana", size: 12 });
        }
    });
},
switchMode: function (e) {
    if (e.mode == 'hidden') {
        dom.stop(true).fadeTo(e.dt, 0, function () {
            if (sys) sys.stop();
            $(this).hide();
        })
    } else if (e.mode == 'visible') {

```

```

        dom.stop(true).css('opacity', 0).show().fadeTo(e.dt, 1,
function () {
        that.resize();
    })
    if (sys) sys.start();
    },
    },
    switchSection: function (newSection) {
        var parent = sys.getEdgesFrom(newSection)[0].source;
        var children = $.map(sys.getEdgesFrom(newSection), function
(edge) {
            return edge.target;
        })

        sys.eachNode(function (node) {
            if (node.data.shape == 'dot') return; // skip all but
leafnodes

            var nowVisible = ($.isArray(node, children) >= 0);
            var newAlpha = (nowVisible) ? 1 : 0;
            var dt = (nowVisible) ? .5 : .5;
            sys.tweenNode(node, dt, { alpha: newAlpha });

            if (newAlpha == 1) {
                node.p.x = parent.p.x + .05 * Math.random() - .025;
                node.p.y = parent.p.y + .05 * Math.random() - .025;
                node.tempMass = .001;
            }
        });
    },
    _initMouseHandling: function () {
        // no-nonsense drag and drop (thanks springy.js)
        selected = null;
        nearest = null;
        var dragged = null;
        var oldmass = 1

        var _section = null

        var handler = {
            moved: function (e) {
                var pos = $(canvas).offset();
                _mouseP = arbor.Point(e.pageX - pos.left, e.pageY -
pos.top);

                nearest = sys.nearest(_mouseP);

                if (!nearest.node) return false;

                if (nearest.node.data.shape != 'dot') {
                    selected = (nearest.distance < 50) ? nearest : null;
                } else if ($.isArray(nearest.node.name, allnodes) >= 0)
{//creamos nuestro propio array
                    if (nearest.node.name != _section) {
                        _section = nearest.node.name;
                        that.switchSection(_section);
                    }
                }

                return false;
            },
        },

```

```

        clicked: function (e) {
            clicks++;
            if(clicks == 1){
                setTimeout(function(){
                    var pos = $(canvas).offset();
                    _mouseP = arbor.Point(e.pageX - pos.left, e.pageY
- pos.top)

                    nearest = dragged = sys.nearest(_mouseP);
                    if(clicks == 1){
                        if (dragged && dragged.node !== null)
dragged.node.fixed = false;
                        handler.moved);
                        handler.dragged)
                        handler.dropped)

                            $(canvas).unbind('mousemove',
                            $(canvas).bind('mousemove',
                            $(window).bind('mouseup',

                                }
                                else{//Doble click
                                    $(canvas).unbind('mousemove',
                                    $(canvas).unbind('mousemove',
                                    $(window).unbind('mouseup',

                                        var nombreNodo = dragged.node.data.palabra;
                                        console.log(nombreNodo);

document.getElementById('MainContent_palabra').value = nombreNodo;
document.getElementById('MainContent_boton').click();
                                }
                                clicks = 0;
                                }, 300)

                            }
                            return false;
                        },
                    dragged: function (e) {
                        var old_nearest = nearest && nearest.node._id
                        var pos = $(canvas).offset();
                        var s = arbor.Point(e.pageX - pos.left, e.pageY -
pos.top)

                        if (!nearest) return
                        if (dragged !== null && dragged.node !== null) {
                            var p = sys.fromScreen(s)
                            dragged.node.p = p
                        }

                        return false
                    },
                    dropped: function (e) {
                        if (dragged === null || dragged.node === undefined)

return
                        if (dragged.node !== null) dragged.node.fixed = false
                        dragged.node.tempMass = 1000
                        dragged = null;
                        // selected = null
                        $(canvas).unbind('mousemove', handler.dragged)

```

```

        $(window).unbind('mouseup', handler.dropped)
        $(canvas).bind('mousemove', handler.moved);
        _mouseP = null
        return false
    }
}
$(canvas).mousedown(handler.clicked);
$(canvas).mousemove(handler.moved);
}
}

return that
}
var allnodes = null;

$(document).ready(function () {
    //Variable a la que se le asigna la funcion que devuelve la ruta del
    archivo json que almacena los datos del grafo.
    var theUI = (function () {
        var url = "json/" + file;
        var theUI = null;
        $.ajax({
            'async': false,
            'global': false,
            'url': url,
            'dataType': "json",
            'success': function (data) {
                theUI = data;
            }
        });
        return theUI;
    })();

    allnodes = (function () {
        var allnodes = new Array();
        var i = 0;
        $.each(theUI.nodes, function (name, info) {
            if (info.alpha === 1) {
                allnodes[i] = name;
                i++;
            }
        });
        console.log('-' + allnodes);
        return allnodes;
    })();
    //Variables que denominan las características del grafo(longitud de
    aristas, tamaño nodos y movimiento).
    var sys = arbor.ParticleSystem()
    sys.parameters({ stiffness: 900, repulsion: 2000, gravity: false, dt:
    0.0015 })
    sys.renderer = Renderer(sitemap)
    sys.graft(theUI)
});
};

```

## Arbor.js

```

//
// arbor.js - version 0.91
// a graph vizualization toolkit
//

```

```

// Copyright (c) 2011 Samizdat Drafting Co.
// Physics code derived from springy.js, copyright (c) 2010 Dennis Hotson
//
// Permission is hereby granted, free of charge, to any person
// obtaining a copy of this software and associated documentation
// files (the "Software"), to deal in the Software without
// restriction, including without limitation the rights to use,
// copy, modify, merge, publish, distribute, sublicense, and/or sell
// copies of the Software, and to permit persons to whom the
// Software is furnished to do so, subject to the following
// conditions:
//
// The above copyright notice and this permission notice shall be
// included in all copies or substantial portions of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
// EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
// OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
// NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
// HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
// WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
// FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
// OTHER DEALINGS IN THE SOFTWARE.
//

(function($){

    /*      etc.js */ var
    trace=function(msg){if(typeof(window)=="undefined"||!window.console){return}var
    len=arguments.length;var args=[];for(var
    i=0;i<len;i++){args.push("arguments["+i+"]")}eval("console.log("+args.join(",")+
    ")");var dirname=function(a){var
    b=a.replace(/^\/?(.*?)\/?$/, "$1").split("/");b.pop();return"/"+b.join("/");var
    basename=function(b){var c=b.replace(/^\/?(.*?)\/?$/, "$1").split("/");var
    a=c.pop();if(a==""){return null}else{return a}};var
    _ordinalize_re=/(\d)(?=(\d\d\d)+(?!\d))/g;var ordinalize=function(a){var
    b=""+"a;if(a<11000){b=""+a).replace(_ordinalize_re, "$1,")}else{if(a<10000000){b=Math.
    floor(a/1000)+"k"}else{if(a<10000000000){b=""+Math.floor(a/1000)).replace(_ord
    inalize_re, "$1,")+m}}}}return b};var nano=function(a,b){return
    a.replace(/\{([\w\-\.\}]*\})/g, function(f,c){var
    d=c.split("."),e=b[d.shift()];$.each(d, function(){if(e.hasOwnProperty(this)){e=e[
    this]}else{e=f}});return e});var objcopy=function(a){if(a===undefined){return
    undefined}if(a===null){return null}if(a.parentNode){return a}switch(typeof
    a){case"string":return a.substring(0);break;case"number":return
    a+0;break;case"boolean":return a===true;break}var
    b=($.isArray(a))?[]:{$};$.each(a, function(d,c){b[d]=objcopy(c)});return b};var
    objmerge=function(d,b){d=d||{};b=b||{};var c=objcopy(d);for(var a in
    b){c[a]=b[a]}return c};var objcmp=function(e,c,d){if(!e||!c){return
    e===c}if(typeof e!=typeof c){return false}if(typeof e!="object"){return
    e===c}else{if($.isArray(e)){if(!($.isArray(c))){return
    false}if(e.length!=c.length){return false}}else{var h=[];for(var f in
    e){if(e.hasOwnProperty(f)){h.push(f)}}var g=[];for(var f in
    c){if(c.hasOwnProperty(f)){g.push(f)}}if(!d){h.sort();g.sort()}if(h.join(",")!=g
    .join(",")){return false}}var i=true;$.each(e, function(a){var
    b=objcmp(e[a],c[a]);i=i&& b;if(!i){return false}});return i}};var
    objkeys=function(b){var
    a=[];$.each(b, function(d,c){if(b.hasOwnProperty(d)){a.push(d)}});return a};var
    objcontains=function(c){if(!c||typeof c!="object"){return false}for(var
    b=1,a=arguments.length;b<a;b++){if(c.hasOwnProperty(arguments[b])){return
    true}}return false};var uniq=function(b){var a=b.length;var d={};for(var
    c=0;c<a;c++){d[b[c]]=true}return objkeys(d)};var arbor_path=function(){var
    a=$( "script" ).map(function(b){var

```

```

c=$(this).attr("src");if(!c){return}if(c.match(/arbor[^\./\.\.]*.js|dev.js/)){return
c.match(/.*\//)||"/"};if(a.length>0){return a[0]}else{return null}};
/* kernel.js */ var Kernel=function(b){var
k=window.location.protocol=="file:"&&navigator.userAgent.toLowerCase().indexOf("c
hrome")>-1;var a=(window.Worker!==undefined&&!k);var i=null;var c=null;var
f=[];f.last=new Date();var l=null;var e=null;var d=null;var h=null;var
g=false;var
j={system:b,tween:null,nodes:{},init:function(){if(typeof(Tween)!="undefined"){c=
Tween()}else{if(typeof(arbor.Tween)!="undefined"){c=arbor.Tween()}else{c={busy:fu
nction(){return false},tick:function(){return true},to:function(){trace("Please
include arbor-tween.js to enable
tweens");c.to=function(){}};return}}}}j.tween=c;var
m=b.parameters();if(a){trace("using web
workers");l=setInterval(j.screenUpdate,m.timeout);i=new
Worker(arbor_path()+"arbor.js");i.onmessage=j.workerMsg;i.onerror=function(n){tra
ce("physics:",n)};i.postMessage({type:"physics",physics:objmerge(m,{timeout:Math.
ceil(m.timeout)}})}else{trace("couldn't use web workers, be
careful...");i=Physics(m.dt,m.stiffness,m.repulsion,m.friction,j.system._updateGe
ometry);j.start()}return
j},graphChanged:function(m){if(a){i.postMessage({type:"changes",changes:m})}else{
i._update(m)}j.start()},particleModified:function(n,m){if(a){i.postMessage({type:
"modify",id:n,mods:m})}else{i.modifyNode(n,m)}j.start()},physicsModified:func
tion(m){if(!isNaN(m.timeout)){if(a){clearInterval(l);l=setInterval(j.screenUpdate,m.t
imeout)}else{clearInterval(d);d=null}}if(a){i.postMessage({type:"sys",param:m})}e
lse{i.modifyPhysics(m)}j.start()},workerMsg:function(n){var
m=n.data.type;if(m=="geometry"){j.workerUpdate(n.data)}else{trace("physics:",n.da
ta)}},_lastPositions:null,workerUpdate:function(m){j._lastPositions=m;j._lastBoun
ds=m.bounds},_lastFrametime:new
Date().valueOf(),_lastBounds:null,_currentRenderer:null,screenUpdate:function(){v
ar n=new Date().valueOf();var
m=false;if(j._lastPositions!==null){j.system._updateGeometry(j._lastPositions);j.
_lastPositions=null;m=true}if(c&&c.busy()){m=true}if(j.system._updateBounds(j._la
stBounds)){m=true}if(m){var
o=j.system.renderer;if(o!==undefined){if(o!==e){o.init(j.system);e=o}if(c){c.tick
()}o.redraw();var p=f.last;f.last=new Date();f.push(f.last-
p);if(f.length>50){f.shift()}}},physicsUpdate:function(){if(c){c.tick()}i.tick()
;var n=j.system._updateBounds();if(c&&c.busy()){n=true}var
o=j.system.renderer;var m=new Date();var
o=j.system.renderer;if(o!==undefined){if(o!==e){o.init(j.system);e=o}o.redraw({ti
mestamp:m})}var q=f.last;f.last=m;f.push(f.last-q);if(f.length>50){f.shift()}var
p=i.systemEnergy();if((p.mean+p.max)/2<0.05){if(h===null){h=new
Date().valueOf()}if(new Date().valueOf()-
h>1000){clearInterval(d);d=null}else{}else{h=null}},fps:function(n){if(n!==undef
ined){var q=1000/Math.max(1,targetFps);j.physicsModified({timeout:q})}var
r=0;for(var p=0,o=f.length;p<o;p++){r+=f[p]}var
m=r/Math.max(1,f.length);if(!isNaN(m)){return Math.round(1000/m)}else{return
0}},start:function(m){if(d!==null){return}if(g&&!m){return}g=false;if(a){i.postMe
ssage({type:"start"})}else{h=null;d=setInterval(j.physicsUpdate,j.system.paramete
rs().timeout)}},stop:function(){g=true;if(a){i.postMessage({type:"stop"})}else{if
(d!==null){clearInterval(d);d=null}}};return j.init()};
/* atoms.js */ var
Node=function(a){this._id=_nextNodeId++;this.data=a||{};this._mass=(a.mass!==unde
fined)?a.mass:1;this._fixed=(a.fixed===true)?true:false;this._p=new
Point((typeof(a.x)=="number"?a.x:null,(typeof(a.y)=="number"?a.y:null);delete
this.data.x;delete this.data.y;delete this.data.mass;delete this.data.fixed};var
_nextNodeId=1;var Edge=function(b,c,a){this._id=_nextEdgeId--
;this.source=b;this.target=c;this.length=(a.length!==undefined)?a.length:1;this.d
ata=(a!==undefined)?a:{};delete this.data.length};var _nextEdgeId=-1;var
Particle=function(a,b){this.p=a;this.m=b;this.v=new Point(0,0);this.f=new
Point(0,0)};Particle.prototype.applyForce=function(a){this.f=this.f.add(a.divide(
this.m))};var
Spring=function(c,b,d,a){this.point1=c;this.point2=b;this.length=d;this.k=a};Spri

```

```

ng.prototype.distanceToParticle=function(a){var
c=that.point2.p.subtract(that.point1.p).normalize().normal();var
b=a.p.subtract(that.point1.p);return Math.abs(b.x*c.x+b.y*c.y);var
Point=function(a,b){if(a&&a.hasOwnProperty("y")){b=a.y;a=a.x}this.x=a;this.y=b};P
oint.random=function(a){a=(a!==undefined)?a:5;return new
Point(2*a*(Math.random()-0.5),2*a*(Math.random()-
0.5))};Point.prototype={exploded:function(){return(isNaN(this.x)||isNaN(this.y))}
,add:function(a){return new
Point(this.x+a.x,this.y+a.y)},subtract:function(a){return new Point(this.x-
a.x,this.y-a.y)},multiply:function(a){return new
Point(this.x*a,this.y*a)},divide:function(a){return new
Point(this.x/a,this.y/a)},magnitude:function(){return
Math.sqrt(this.x*this.x+this.y*this.y)},normal:function(){return new Point(-
this.y,this.x)},normalize:function(){return this.divide(this.magnitude())};
/* system.js */ var ParticleSystem=function(d,p,e,f,t,l,q){var j=[];var
h=null;var k=0;var u=null;var m=0.04;var i=[20,20,20,20];var n=null;var
o=null;if(typeof p=="object"){var
s=p;e=s.friction;d=s.repulsion;t=s.fps;l=s.dt;p=s.stiffness;f=s.gravity;q=s.preci
sion}e=isNaN(e)?0.5:e;d=isNaN(d)?1000:d;t=isNaN(t)?55:t;p=isNaN(p)?600:p;l=isNaN(
l)?0.02:l;q=isNaN(q)?0.6:q;f=(f===true);var r=(t!==undefined)?1000/t:1000/50;var
b={repulsion:d,stiffness:p,friction:e,dt:l,gravity:f,precision:q,timeout:r};var
a;var
c={renderer:null,tween:null,nodes:{},edges:{},adjacency:{},names:{},kernel:null};
var
g={parameters:function(v){if(v!==undefined){if(!isNaN(v.precision)){v.precision=M
ath.max(0,Math.min(1,v.precision))}$.each(b,function(x,w){if(v[x]!==undefined){b[
x]=v[x]}});c.kernel.physicsModified(v)}return
b},fps:function(v){if(v===undefined){return
c.kernel.fps()}else{g.parameters({timeout:1000/(v||50)})},start:function(){c.ker
nel.start()},stop:function(){c.kernel.stop()},addNode:function(w,B){B=B||{};var
C=c.names[w];if(C){C.data=B;return C}else{if(w!==undefined){var
v=(B.x!==undefined)?B.x:null;var D=(B.y!==undefined)?B.y:null;var
A=(B.fixed)?1:0;var z=new
Node(B);z.name=w;c.names[z._id]=z;j.push({t:"addNode",id:z._id,m:z.m
ass,x:v,y:D,f:A});g._notify();return z}},pruneNode:function(w){var
v=g.getNode(w);if(typeof(c.nodes[v._id])!="undefined"){delete
c.nodes[v._id];delete
c.names[v.name]}$.each(c.edges,function(y,x){if(x.source._id===v._id||x.target._i
d===v._id){g.pruneEdge(x)}});j.push({t:"dropNode",id:v._id});g._notify();getNode
:function(v){if(v._id!==undefined){return v}else{if(typeof v=="string"||typeof
v=="number"){return
c.names[v]}}},eachNode:function(v){$.each(c.nodes,function(y,x){if(x._p.x===null||
x._p.y===null){return}var
w=(u!==null)?g.toScreen(x._p):x._p;v.call(g,x,w)}},addEdge:function(z,A,y){z=g.g
etNode(z)||g.addNode(z);A=g.getNode(A)||g.addNode(A);y=y||{};var x=new
Edge(z,A,y);var B=z._id;var
C=A._id;c.adjacency[B]=c.adjacency[B]||{};c.adjacency[B][C]=c.adjacency[B][C]||[
];var
w=(c.adjacency[B][C].length>0);if(w){$.extend(c.adjacency[B][C].data,x.data);retu
rn}else{c.edges[x._id]=x;c.adjacency[B][C].push(x);var
v=(x.length!==undefined)?x.length:1;j.push({t:"addSpring",id:x._id,fm:B,to:C,l:v}
);g._notify();return
x},pruneEdge:function(A){j.push({t:"dropSpring",id:A._id});delete
c.edges[A._id];for(var v in c.adjacency){for(var B in c.adjacency[v]){var
w=c.adjacency[v][B];for(var z=w.length-1;z>=0;z--
){if(c.adjacency[v][B][z]._id===A._id){c.adjacency[v][B].splice(z,1)}}}g._notifi
e()},getEdges:function(w,v){w=g.getNode(w);v=g.getNode(v);if(!w||!v){return[]}if(t
ypeof(c.adjacency[w._id])!="undefined"&&typeof(c.adjacency[w._id][v._id])!="und
efined"){return
c.adjacency[w._id][v._id]}return[]},getEdgesFrom:function(v){v=g.getNode(v);if(!v
){return[]}if(typeof(c.adjacency[v._id])!="undefined"){var
w=[];$.each(c.adjacency[v._id],function(y,x){w=w.concat(x)});return

```

```

w}return[]},getEdgesTo:function(v){v=g.getNode(v);if(!v){return[]}var
w=[];$.each(c.edges,function(y,x){if(x.target==v){w.push(x)}});return
w},eachEdge:function(v){$.each(c.edges,function(z,x){var
y=c.nodes[x.source._id]._p;var
w=c.nodes[x.target._id]._p;if(y.x==null||w.x==null){return}y=(u!=null)?g.toScreen(y):y;w=(u!=null)?g.toScreen(w):w;if(y&&w){v.call(g,x,y,w)}}),prune:function(w){var
v={dropped:{nodes:[],edges:[]};if(w===undefined){$.each(c.nodes,function(y,x){v.dropped.nodes.push(x);g.pruneNode(x)}})else{g.eachNode(function(y){var
x=w.call(g,y,{from:g.getEdgesFrom(y),to:g.getEdgesTo(y)});if(x){v.dropped.nodes.push(y);g.pruneNode(y)}})}return v},graft:function(w){var
v={added:{nodes:[],edges:[]};if(w.nodes){$.each(w.nodes,function(y,x){var
z=g.getNode(y);if(z){z.data=x}else{v.added.nodes.push(g.addNode(y,x))}c.kernel.start()})}if(w.edges){$.each(w.edges,function(z,x){var
y=g.getNode(z);if(!y){v.added.nodes.push(g.addNode(z,{})}$$.each(x,function(D,A){var
C=g.getNode(D);if(!C){v.added.nodes.push(g.addNode(D,{})})}var
B=g.getEdges(z,D);if(B.length>0){B[0].data=A}else{v.added.edges.push(g.addEdge(z,D,A))}})}return v},merge:function(w){var
v={added:{nodes:[],edges:[]},dropped:{nodes:[],edges:[]};$.each(c.edges,function(A,z){if((w.edges[z.source.name]===undefined||w.edges[z.source.name][z.target.name]===undefined)){g.pruneEdge(z);v.dropped.edges.push(z)}});var
y=g.prune(function(A,z){if(w.nodes[A.name]===undefined){v.dropped.nodes.push(A);return true}});var
x=g.graft(w);v.added.nodes=v.added.nodes.concat(x.added.nodes);v.added.edges=v.added.edges.concat(x.added.edges);v.dropped.nodes=v.dropped.nodes.concat(y.dropped.nodes);v.dropped.edges=v.dropped.edges.concat(y.dropped.edges);return
v},tweenNode:function(y,v,x){var
w=g.getNode(y);if(w){c.tween.to(w,v,x)},tweenEdge:function(w,v,z,y){if(y===undefined){g._tweenEdge(w,v,z)}else{var
x=g.getEdges(w,v);$.each(x,function(A,B){g._tweenEdge(B,z,y)}}),_tweenEdge:function(w,v,x){if(w&&w._id!==undefined){c.tween.to(w,v,x)},_updateGeometry:function(y){if(y!==undefined){var
v=(y.epoch<k);a=y.energy;var
z=y.geometry;if(z!==undefined){for(var
x=0,w=z.length/3;x<w;x++){var
A=z[3*x];if(v&&c.nodes[A]===undefined){continue}c.nodes[A]._p.x=z[3*x+1];c.nodes[A]._p.y=z[3*x+2]}}},screen:function(v){if(v===undefined){return}size:(u)?objcopy(u):undefined,padding:i.concat(),step:m}if(v.size!==undefined){g.screenSize(v.size.width,v.size.height)}if(!isNaN(v.step)){g.screenStep(v.step)}if(v.padding!==undefined){g.screenPadding(v.padding)},screenSize:function(v,w){u={width:v,height:w};g._updateBounds(),screenPadding:function(y,z,v,w){if($.isArray(y)){trbl=y}else{trbl=[y,z,v,w]}var
A=trbl[0];var
x=trbl[1];var
B=trbl[2];if(x===undefined){trbl=[A,A,A]}else{if(B===undefined){trbl=[A,x,A,x]}i=trbl},screenStep:function(v){m=v},toScreen:function(x){if(!n||!u){return}var
w=i||[0,0,0,0];var
v=n.bottomright.subtract(n.topleft);var
z=w[3]+x.subtract(n.topleft).divide(v.x)*(u.width-(w[1]+w[3]));var
y=w[0]+x.subtract(n.topleft).divide(v.y)*(u.height-(w[0]+w[2]));return
arbor.Point(z,y)},fromScreen:function(z){if(!n||!u){return}var
y=i||[0,0,0,0];var
x=n.bottomright.subtract(n.topleft);var
w=(z.x-y[3])/(u.width-(y[1]+y[3]))*x.x+n.topleft.x;var
v=(z.y-y[0])/(u.height-(y[0]+y[2]))*x.y+n.topleft.y;return
arbor.Point(w,v)},_updateBounds:function(w){if(u===null){return}if(w){o=w}else{o=g.bounds()}var
z=new Point(o.bottomright.x,o.bottomright.y);var
y=new Point(o.topleft.x,o.topleft.y);var
B=z.subtract(y);var
v=y.add(B.divide(2));var
x=4;var
D=new Point(Math.max(B.x,x),Math.max(B.y,x));o.topleft=v.subtract(D.divide(2));o.bottomright=v.add(D.divide(2));if(!n){if($.isEmptyObject(c.nodes)){return false}n=o;return true}var
C=m;_newBounds={bottomright:n.bottomright.add(o.bottomright.subtract(n.bottomright).multiply(C)),topleft:n.topleft.add(o.topleft.subtract(n.topleft).multiply(C))};var
A=new Point(n.topleft.subtract(_newBounds.topleft).magnitude(),n.bottomright.subtract(_newBounds.bottomright).magnitude());if(A.x*u.width>1||A.y*u.height>1){n=_newBounds;return true}else{return false}},energy:function(){return

```



```

a}, bounds: function() { var w = null; var
v = null; $.each(c.nodes, function(z, y) { if (!w) { w = new Point(y._p); v = new
Point(y._p); return } var
x = y._p; if (x.x === null || x.y === null) { return } if (x.x > w.x) { w.x = x.x } if (x.y > w.y) { w.y = x.y }
if (x.x < v.x) { v.x = x.x } if (x.y < v.y) { v.y = x.y } }); if (w && v) { return { bottomright: w, topleft:
v } } else { return { topleft: new Point(-1, -1), bottomright: new
Point(1, 1) } } }, nearest: function(x) { if (u !== null) { x = g.fromScreen(x) } var
w = { node: null, point: null, distance: null }; var v = g; $.each(c.nodes, function(B, y) { var
z = y._p; if (z.x === null || z.y === null) { return } var
A = z.subtract(x).magnitude(); if (w.distance === null || A < w.distance) { w = { node: y, point: z
, distance: A }; if (u !== null) { w.screenPoint = g.toScreen(z) } } }); if (w.node) { if (u !== null)
{ w.distance = g.toScreen(w.node.p).subtract(g.toScreen(x)).magnitude() } return
w } else { return
null } }, _notify: function() { if (h === null) { k++ } else { clearTimeout(h) } h = setTimeout(g._s
ynchronize, 20) }, _synchronize: function() { if (j.length > 0) { c.kernel.graphChanged(j); j
= []; h = null } }, }; c.kernel = Kernel(g); c.tween = c.kernel.tween || null; Node.prototype.__d
efineGetter__("p", function() { var w = this; var
v = {}; v.__defineGetter__("x", function() { return
w._p.x }); v.__defineSetter__("x", function(x) { c.kernel.particleModified(w._id, { x: x
}); }); v.__defineGetter__("y", function() { return
w._p.y }); v.__defineSetter__("y", function(x) { c.kernel.particleModified(w._id, { y: x
}); }); v.__proto__ = Point.prototype; return
v }); Node.prototype.__defineSetter__("p", function(v) { this._p.x = v.x; this._p.y = v.y; c
.kernel.particleModified(this._id, { x: v.x, y: v.y }); }); Node.prototype.__defineGetter__
("mass", function() { return
this._mass }); Node.prototype.__defineSetter__("mass", function(v) { this._mass = v; c.ke
rnel.particleModified(this._id, { m: v }); }); Node.prototype.__defineSetter__("tempMass
", function(v) { c.kernel.particleModified(this._id, { m: v }); }); Node.prototype.__defin
eGetter__("fixed", function() { return
this._fixed }); Node.prototype.__defineSetter__("fixed", function(v) { this._fixed = v; c
.kernel.particleModified(this._id, { f: v ? 1 : 0 }); }); return g;
/* barnes-hut.js */ var BarnesHutTree = function() { var b = []; var a = 0; var
e = null; var d = 0.5; var
c = { init: function(g, h, f) { d = f; a = 0; e = c._newBranch(); e.origin = g; e.size = h.subtract(g)
}, insert: function(j) { var f = e; var g = [j]; while (g.length) { var h = g.shift(); var
m = h._m || h.m; var
p = c._whichQuad(h, f); if (f[p] === undefined) { f[p] = h; f.mass += m; if (f.p) { f.p = f.p.add(h.p
.multiply(m)) } else { f.p = h.p.multiply(m) } } else { if ("origin" in
f[p]) { f.mass += (m); if (f.p) { f.p = f.p.add(h.p.multiply(m)) } else { f.p = h.p.multiply(m) } } f
= f[p]; g.unshift(h) } else { var l = f.size.divide(2); var n = new
Point(f.origin); if (p[0] == "s") { n.y += 1.y } if (p[1] == "e") { n.x += 1.x } var
o = f[p]; f[p] = c._newBranch(); f[p].origin = n; f[p].size = l; f.mass = m; f.p = h.p.multiply(m)
; f = f[p]; if (o.p.x == h.p.x && o.p.y == h.p.y) { var k = l.x * 0.08; var
i = l.y * 0.08; o.p.x = Math.min(n.x + 1.x, Math.max(n.x, o.p.x -
k / 2 + Math.random() * k)); o.p.y = Math.min(n.y + 1.y, Math.max(n.y, o.p.y -
i / 2 + Math.random() * i)); g.push(o); g.unshift(h) } } }, applyForces: function(m, g) { var
f = [e]; while (f.length) { node = f.shift(); if (node === undefined) { continue } if (m === node) { c
ontinue } if ("f" in node) { var k = m.p.subtract(node.p); var
l = Math.max(1, k.magnitude()); var
i = ((k.magnitude() > 0) ? k : Point.random(1)).normalize(); m.applyForce(i.multiply(g * (no
de._m || node.m)).divide(l * 1)) } else { var
j = m.p.subtract(node.p.divide(node.mass)).magnitude(); var
h = Math.sqrt(node.size.x * node.size.y); if (h / j > d) { f.push(node.ne); f.push(node.nw); f.
push(node.se); f.push(node.sw) } else { var
k = m.p.subtract(node.p.divide(node.mass)); var l = Math.max(1, k.magnitude()); var
i = ((k.magnitude() > 0) ? k : Point.random(1)).normalize(); m.applyForce(i.multiply(g * (no
de.mass)).divide(l * 1)) } } }, _whichQuad: function(i, f) { if (i.p.exploded()) { return
null } var h = i.p.subtract(f.origin); var
g = f.size.divide(2); if (h.y < g.y) { if (h.x < g.x) { return "nw" } else { return "ne" } } else { if (h.
x < g.x) { return "sw" } else { return "se" } } }, _newBranch: function() { if (b[a]) { var
f = b[a]; f.ne = f.nw = f.se = f.sw = undefined; f.mass = 0; delete

```

```

f.p})else{f={origin:null,size:null,nw:undefined,ne:undefined,sw:undefined,se:undef
ined,mass:0};b[a]=f}a++;return f}};return c};
/* physics.js */ var Physics=function(a,m,n,e,h){var f=BarnesHutTree();var
c={particles:{},springs:{}};var l={particles:{}};var o=[];var k=[];var d=0;var
b={sum:0,max:0,mean:0};var g={topleft:new Point(-1,-1),bottomright:new
Point(1,1)};var j=1000;var
i={stiffness:(m!==undefined)?m:1000,repulsion:(n!==undefined)?n:600,friction:(e!=
=undefined)?e:0.3,gravity:false,dt:(a!==undefined)?a:0.02,theta:0.4,init:function
(){return
i},modifyPhysics:function(p){$.each(["stiffness","repulsion","friction","gravity"
,"dt","precision"],function(r,s){if(p[s]!==undefined){if(s=="precision"){i.theta=
1-p[s];return}i[s]=p[s];if(s=="stiffness"){var
q=p[s];$.each(c.springs,function(u,t){t.k=q}})},addNode:function(u){var
t=u.id;var q=u.m;var p=g.bottomright.x-g.topleft.x;var s=g.bottomright.y-
g.topleft.y;var r=new
Point((u.x!==null)?u.x:g.topleft.x+p*Math.random(),(u.y!==null)?u.y:g.topleft.y+s*M
ath.random());c.particles[t]=new
Particle(r,q);c.particles[t].connections=0;c.particles[t].fixed=(u.f===1);l.parti
cles[t]=c.particles[t];o.push(c.particles[t]);},dropNode:function(s){var
r=s.id;var q=c.particles[r];var p=$.inArray(q,o);if(p>-1){o.splice(p,1)}delete
c.particles[r];delete l.particles[r];modifyNode:function(r,p){if(r in
c.particles){var q=c.particles[r];if("x" in p){q.p.x=p.x}if("y" in
p){q.p.y=p.y}if("m" in p){q.m=p.m}if("f" in p){q.fixed=(p.f===1)}if("_m" in
p){if(q._m===undefined){q._m=q.m}q.m=p._m}},addSpring:function(t){var s=t.id;var
p=t.l;var r=c.particles[t.fm];var
q=c.particles[t.to];if(r!==undefined&&q!==undefined){c.springs[s]=new
Spring(r,q,p,i.stiffness);k.push(c.springs[s]);r.connections++;q.connections++;de
lete l.particles[t.fm];delete l.particles[t.to]}},dropSpring:function(s){var
r=s.id;var q=c.springs[r];q.point1.connections--;q.point2.connections--;var
p=$.inArray(q,k);if(p>-1){k.splice(p,1)}delete
c.springs[r]},_update:function(p){d++;$.each(p,function(q,r){if(r.t in
i){i[r.t](r)}});return
d},tick:function(){i.tendParticles();i.eulerIntegrator(i.dt);i.tock()},tock:func
tion(){var
p=[];$.each(c.particles,function(r,q){p.push(r);p.push(q.p.x);p.push(q.p.y)});if(
h){h({geometry:p,epoch:d,energy:b,bounds:g})},tendParticles:function(){$.each(c.
particles,function(q,p){if(p._m!==undefined){if(Math.abs(p.m-
p._m)<1){p.m=p._m;delete
p._m}else{p.m*=0.98}}p.v.x=p.v.y=0}},eulerIntegrator:function(p){if(i.repulsion>
0){if(i.theta>0){i.applyBarnesHutRepulsion()}else{i.applyBruteForceRepulsion()}}i
f(i.stiffness>0){i.applySprings()}i.applyCenterDrift();if(i.gravity){i.applyCente
rGravity()}i.updateVelocity(p);i.updatePosition(p)},applyBruteForceRepulsion:func
tion(){$.each(c.particles,function(q,p){$.each(c.particles,function(s,r){if(p!==r
){var u=p.p.subtract(r.p);var v=Math.max(1,u.magnitude());var
t=((u.magnitude())>0)?u:Point.random(1).normalize();p.applyForce(t.multiply(i.rep
ulsion*(r._m|p.m)*0.5).divide(v*v*0.5));r.applyForce(t.multiply(i.repulsion*(p._
m|p.m)*0.5).divide(v*v*-
0.5))}})}},applyBarnesHutRepulsion:function(){if(!g.topleft||!g.bottomright){ret
urn}var q=new Point(g.bottomright);var p=new
Point(g.topleft);f.init(p,q,i.theta);$.each(c.particles,function(s,r){f.insert(r)
});$.each(c.particles,function(s,r){f.applyForces(r,i.repulsion)}),applySprings:
function(){$.each(c.springs,function(t,p){var
s=p.point2.p.subtract(p.point1.p);var q=p.length-s.magnitude();var
r=((s.magnitude())>0)?s:Point.random(1).normalize();p.point1.applyForce(r.multipl
y(p.k*q*-
0.5));p.point2.applyForce(r.multiply(p.k*q*0.5))},applyCenterDrift:function(){v
ar q=0;var r=new
Point(0,0);$.each(c.particles,function(t,s){r.add(s.p);q++});if(q==0){return}var
p=r.divide(-
q);$.each(c.particles,function(t,s){s.applyForce(p)}),applyCenterGravity:functio
n(){$.each(c.particles,function(r,p){var q=p.p.multiply(-
1);p.applyForce(q.multiply(i.repulsion/100))}},updateVelocity:function(p){$.each

```

```

(c.particles,function(t,q){if(q.fixed){q.v=new Point(0,0);q.f=new
Point(0,0);return}var s=q.v.magnitude();q.v=q.v.add(q.f.multiply(p)).multiply(1-
i.friction);q.f.x=q.f.y=0;var
r=q.v.magnitude();if(r>j){q.v=q.v.divide(r*r)}}},updatePosition:function(q){var
r=0,p=0,u=0;var t=null;var
s=null;$.each(c.particles,function(w,v){v.p=v.p.add(v.v.multiply(q));var
x=v.v.magnitude();var z=x*x;r+=z;p=Math.max(z,p);u++;if(!t){t=new
Point(v.p.x,v.p.y);s=new Point(v.p.x,v.p.y);return}var
y=v.p;if(y.x===null||y.y===null){return}if(y.x>t.x){t.x=y.x}if(y.y>t.y){t.y=y.y}i
f(y.x<s.x){s.x=y.x}if(y.y<s.y){s.y=y.y}});b={sum:r,max:p,mean:r/u,n:u};g={topleft
:s||new Point(-1,-1),bottomright:t||new
Point(1,1)},systemEnergy:function(p){return b}};return i.init());var
_nearParticle=function(b,c){var c=c||0;var a=b.x;var f=b.y;var e=c*2;return new
Point(a-c+Math.random()*e,f-c+Math.random()*e)};

```

```

// if called as a worker thread, set up a run loop for the Physics object and
bail out
if (typeof(window)=='undefined') return (function(){
/* hermetic.js */ $={each:function(d,e){if($.isArray(d)){for(var
c=0,b=d.length;c<b;c++){e(c,d[c])}else{for(var a in
d){e(a,d[a])}}}},map:function(a,c){var b=[];$.each(a,function(f,e){var
d=c(e);if(d!==undefined){b.push(d)}});return b},extend:function(c,b){if(typeof
b!="object"){return c}for(var a in b){if(b.hasOwnProperty(a)){c[a]=b[a]}return
c},isArray:function(a){if(!a){return
false}return(a.constructor.toString().indexOf("Array")!=-
1)},inArray:function(c,a){for(var d=0,b=a.length;d<b;d++){if(a[d]===c){return
d}}return -1},isEmptyObject:function(a){if(typeof a!="object"){return false}var
b=true;$.each(a,function(c,d){b=false});return b},});
/* worker.js */ var PhysicsWorker=function(){var b=20;var a=null;var
d=null;var c=null;var g=[];var f=new Date().valueOf();var
e={init:function(h){e.timeout(h.timeout);a=Physics(h.dt,h.stiffness,h.repulsion,h
.friction,e.tock);return
e},timeout:function(h){if(h!=b){b=h;if(d!==null){e.stop();e.go()}}},go:function()
{if(d!==null){return}c=null;d=setInterval(e.tick,b)},stop:function(){if(d===null)
{return}clearInterval(d);d=null},tick:function(){a.tick();var
h=a.systemEnergy();if((h.mean+h.max)/2<0.05){if(c===null){c=new
Date().valueOf()}if(new Date().valueOf()-
c>1000){e.stop()}else{}}else{c=null}},tock:function(h){h.type="geometry";postMess
age(h)},modifyNode:function(i,h){a.modifyNode(i,h);e.go()},modifyPhysics:function
(h){a.modifyPhysics(h)},update:function(h){var i=a._update(h)};return e};var
physics=PhysicsWorker();onmessage=function(a){if(!a.data.type){postMessage(";kérn
ël?");return}if(a.data.type=="physics"){var
b=a.data.physics;physics.init(a.data.physics);return}switch(a.data.type){case"mod
ify":physics.modifyNode(a.data.id,a.data.mods);break;case"changes":physics.update
(a.data.changes);physics.go();break;case"start":physics.go();break;case"stop":phy
sics.stop();break;case"sys":var
b=a.data.param||{};if(!isNaN(b.timeout)){physics.timeout(b.timeout)}physics.modif
yPhysics(b);physics.go();break}};
})();

```

```

arbor = (typeof(arbor)!='undefined') ? arbor : {}
$.extend(arbor, {
// object constructors (don't use 'new', just call them)
ParticleSystem:ParticleSystem,
Point:function(x, y){ return new Point(x, y) },

// immutable object with useful methods
etc:{
trace:trace, // f(msg) -> safe console logging
dirname:dirname, // f(path) -> leading part of path
basename:basename, // f(path) -> trailing part of path

```

```

        ordinalize:ordinalize,    // f(num) -> abbrev integers (and add commas)
        objcopy:objcopy,         // f(old) -> clone an object
        objcmp:objcmp,           // f(a, b, strict_ordering) -> t/f comparison
        objkeys:objkeys,         // f(obj) -> array of all keys in obj
        objmerge:objmerge,       // f(dst, src) -> like $.extend but non-
destructive
        uniq:uniq,                // f(arr) -> array of unique items in arr
        arbor_path:arbor_path,    // f() -> guess the directory of the lib code
    }
})
})(this.jQuery)

```

## Arbor-tween.js

```

//
// arbor-tween.js
// smooth transitions with a realtime clock
//
// Copyright (c) 2011 Samizdat Drafting Co.
//
// Permission is hereby granted, free of charge, to any person
// obtaining a copy of this software and associated documentation
// files (the "Software"), to deal in the Software without
// restriction, including without limitation the rights to use,
// copy, modify, merge, publish, distribute, sublicense, and/or sell
// copies of the Software, and to permit persons to whom the
// Software is furnished to do so, subject to the following
// conditions:
//
// The above copyright notice and this permission notice shall be
// included in all copies or substantial portions of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
// EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
// OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
// NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
// HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
// WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
// FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
// OTHER DEALINGS IN THE SOFTWARE.
//
// Easing Equations in easing.js:
// Copyright © 2001 Robert Penner. All rights reserved.
//
// Open source under the BSD License. Redistribution and use in source
// and binary forms, with or without modification, are permitted
// provided that the following conditions are met:
//
// Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// Redistributions in binary form must reproduce the above copyright
// notice, this list of conditions and the following disclaimer in the
// documentation and/or other materials provided with the distribution.
//
// Neither the name of the author nor the names of contributors may be
// used to endorse or promote products derived from this software
// without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT

```

```
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

```
(function($){
    /* etc.js */ var
    trace=function(msg){if(typeof(window)=="undefined"||!window.console){return}var
    len=arguments.length;var args=[];for(var
    i=0;i<len;i++){args.push("arguments["+i+"]")}eval("console.log("+args.join(",")+
    ")");var dirname=function(a){var
    b=a.replace(/^\//?(\.*?)\//?$/,"$1").split("/");b.pop();return"/"+b.join("/");var
    basename=function(b){var c=b.replace(/^\//?(\.*?)\//?$/,"$1").split("/");var
    a=c.pop();if(a==""){return null}else{return a}};var
    _ordinalize_re=/(\d)(?=(\d\d\d)+(?!\d))/g;var ordinalize=function(a){var
    b=""+a;if(a<11000){b=""+a}.replace(_ordinalize_re,"$1,")}else{if(a<1000000){b=Math.
    floor(a/1000)+"k"}else{if(a<1000000000){b=""+Math.floor(a/1000)}.replace(_ord
    inalize_re,"$1,")+m}}return b};var nano=function(a,b){return
    a.replace(/\{([\w\-\.\.]*\})/g,function(f,c){var
    d=c.split("."),e=b[d.shift()];$.each(d,function(){if(e.hasOwnProperty(this)){e=e[
    this]}else{e=f}});return e});var objcopy=function(a){if(a===undefined){return
    undefined}if(a===null){return null}if(a.parentNode){return a}switch(typeof
    a){case"string":return a.substring(0);break;case"number":return
    a+0;break;case"boolean":return a===true;break}var
    b=($.isArray(a))?[]:{$};$.each(a,function(d,c){b[d]=objcopy(c)});return b};var
    objmerge=function(d,b){d=d||{};b=b||{};var c=objcopy(d);for(var a in
    b){c[a]=b[a]}return c};var objcmp=function(e,c,d){if(!e||!c){return
    e===c}if(typeof e!=typeof c){return false}if(typeof e!="object"){return
    e===c}else{if($.isArray(e)){if(!($.isArray(c))){return
    false}if(e.length!=c.length){return false}}else{var h=[];for(var f in
    e){if(e.hasOwnProperty(f)){h.push(f)}}var g=[];for(var f in
    c){if(c.hasOwnProperty(f)){g.push(f)}}if(!d){h.sort();g.sort()}if(h.join(",")!=g.
    join(",")){return false}}var i=true;$.each(e,function(a){var
    b=objcmp(e[a],c[a]);i=i&& b;if(!i){return false}});return i}};var
    objkeys=function(b){var
    a=[];$.each(b,function(d,c){if(b.hasOwnProperty(d)){a.push(d)}});return a};var
    objcontains=function(c){if(!c||typeof c!="object"){return false}for(var
    b=1,a=arguments.length;b<a;b++){if(c.hasOwnProperty(arguments[b])){return
    true}}return false};var uniq=function(b){var a=b.length;var d={};for(var
    c=0;c<a;c++){d[b[c]]=true}return objkeys(d)};var arbor_path=function(){var
    a=$("#script").map(function(b){var
    c=$(this).attr("src");if(!c){return}if(c.match(/arbor[^\.\.]*.js|dev.js/)){return
    c.match(/.*\//)||"/}});if(a.length>0){return a[0]}else{return null}};
    /* colors.js */ var Colors=(function(){var f=#[0-9a-f]{6}/i;var
    b=#[(..)(..)(..)/;var c=function(h){var
    g=h.toString(16);return(g.length==2)?g:"0"+g};var a=function(g){return
    parseInt(g,16)};var d=function(g){if(!g||typeof g!="object"){return false}var
    h=objkeys(g).sort().join("");if(h=="abgn"){return true}};var
    e={CSS:{aliceblue:"#f0f8fff",antiquewhite:"#faebd7",aqua:"#00ffff",aquamarine:"#7f
    ffd4",azure:"#00ffff",beige:"#f5f5dc",bisque:"#ffe4c4",black:"#000000",blanchedal
    mond:"#ffebcd",blue:"#0000ff",blueviolet:"#8a2be2",brown:"#a52a2a",burlywood:"#de
    b887",cadetblue:"#5f9ea0",chartreuse:"#7fff00",chocolate:"#d2691e",coral:"#ff7f50
    ",cornflowerblue:"#6495ed",cornsilk:"#fff8dc",crimson:"#dc143c",cyan:"#00ffff",da
    rkblue:"#00008b",darkcyan:"#008b8b",darkgoldenrod:"#b8860b",darkgray:"#a9a9a9",da
    rkgrey:"#a9a9a9",darkgreen:"#006400",darkkhaki:"#bdb76b",darkmagenta:"#8b008b",da
```

```

rkolivegreen: "#556b2f", darkorange: "#ff8c00", darkorchid: "#9932cc", darkred: "#8b0000",
darksalmon: "#e9967a", darkseagreen: "#8fbcbf", darkslateblue: "#483d8b", darkslategray: "#2f4f4f",
darkslategrey: "#2f4f4f", darkturquoise: "#00ced1", darkviolet: "#9400d3", deeppink: "#ff1493",
deepskyblue: "#00bfff", dimgray: "#696969", dimgrey: "#696969", dodgerblue: "#1e90ff",
firebrick: "#b22222", floralwhite: "#fffaf0", forestgreen: "#228b22", fuchsia: "#ff00ff",
gainsboro: "#dcdcdc", ghostwhite: "#f8f8ff", gold: "#ffd700", goldenrod: "#daa520",
gray: "#808080", grey: "#808080", green: "#008000", greenyellow: "#adff2f", honeydew: "#f0ffff",
hotpink: "#ff69b4", indianred: "#cd5c5c", indigo: "#4b0082", ivory: "#fffff0", khaki: "#f0e68c",
lavender: "#e6e6fa", lavenderblush: "#fff0f5", lawngreen: "#7cfc00", lemonchiffon: "#ffffac",
lightblue: "#add8e6", lightcoral: "#f08080", lightcyan: "#e0ffff", lightgoldenrodyellow: "#fafad2",
lightgray: "#d3d3d3", lightgrey: "#d3d3d3", lightgreen: "#90ee90", lightpink: "#ffb6c1",
lightsalmon: "#ffa07a", lightseagreen: "#20b2aa", lightskyblue: "#87cefa",
lightslategray: "#778899", lightslategrey: "#778899", lightsteelblue: "#b0c4de",
lightyellow: "#ffffe0", lime: "#00ff00", limegreen: "#32cd32", linen: "#faf0e6",
magenta: "#ff00ff", maroon: "#800000", mediumaquamarine: "#66cdaa", mediumblue: "#0000cd",
mediumorchid: "#ba55d3", mediumpurple: "#9370d8", mediumseagreen: "#3cb371",
mediumslateblue: "#7b68ee", mediumspringgreen: "#00fa9a", mediumturquoise: "#48d1cc",
mediumvioletred: "#c71585", midnightblue: "#191970", mintcream: "#f5fffa", mistyrose: "#ffe4e1",
moccasin: "#ffe4b5", navajowhite: "#fffdead", navy: "#000080", oldlace: "#fdf5e6",
olive: "#808000", olivedrab: "#6b8e23", orange: "#ffa500", orangered: "#ff4500", orchid: "#da70d6",
palegoldenrod: "#eee8aa", palegreen: "#98fb98", paleturquoise: "#afeeee", palevioletred: "#d87093",
papayawhip: "#ffefd5", peachpuff: "#ffdab9", peru: "#cd853f", pink: "#ffc0cb", plum: "#dda0dd",
powderblue: "#b0e0e6", purple: "#800080", red: "#ff0000", rosybrown: "#bc8f8f",
royalblue: "#4169e1", saddlebrown: "#8b4513", salmon: "#fa8072", sandybrown: "#f4a460",
seagreen: "#2e8b57", seashell: "#fff5ee", sienna: "#a0522d", silver: "#c0c0c0",
skyblue: "#87ceeb", slateblue: "#6a5acd", slategray: "#708090", slategrey: "#708090",
snow: "#fffafa", springgreen: "#00ff7f", steelblue: "#4682b4", tan: "#d2b48c", teal: "#008080",
thistle: "#d8bfd8", tomato: "#ff6347", turquoise: "#40e0d0", violet: "#ee82ee", wheat: "#f5deb3",
white: "#ffffff", whitesmoke: "#f5f5f5", yellow: "#ffff00", yellowgreen: "#9acd32"};
decode: function(h){var g=arguments.length;for(var l=g-1;l>=0;l--){if(arguments[l]==undefined){g--}var k=arguments;if(!h){return null}if(g==1&&d(h)){return h}var j=null;if(typeof h=="string"){var o=1;if(g==2){o=k[1]}var n=e.CSS[h.toLowerCase()];if(n!==undefined){h=n}var m=h.match(f);if(m){vals=h.match(b);if(!vals||!vals.length||vals.length!=4){return null}j={r:a(vals[1]),g:a(vals[2]),b:a(vals[3]),a:o}}else{if(typeof h=="number"){if(g>=3){j={r:k[0],g:k[1],b:k[2],a:1};if(g>=4){j.a*=k[3]}}else{if(g>=1){j={r:k[0],g:k[0],b:k[0],a:1};if(g==2){j.a*=k[1]}}}}return j},validate: function(g){if(!g||typeof g!="string"){return false}if(e.CSS[g.toLowerCase()]!==undefined){return true}if(g.match(f)){return true}return false},mix: function(h,g,k){var j=e.decode(h);var i=e.decode(g)},blend: function(g,j){j=(j!==undefined)?Math.max(0,Math.min(1,j)):1;var h=e.decode(g);if(!h){return null}if(j==1){return g}var h=g;if(typeof g=="string"){h=e.decode(g)}var i=objcopy(h);i.a*=j;return nano("rgba({r},{g},{b},{a})",i)},encode: function(g){if(!d(g)){g=e.decode(g);if(!d(g)){return null}}if(g.a==1){return nano("#{r}{g}{b}",{r:c(g.r),g:c(g.g),b:c(g.b)})}else{return nano("rgba({r},{g},{b},{a})",g)}}};
/* easing.js */
var Easing=(function(){var a={linear: function(f,e,h,g){return h*(f/g)+e},quadin: function(f,e,h,g){return h*(f/=g)*f+e},quadout: function(f,e,h,g){return -h*(f/=g)*(f-2)+e},quadinout: function(f,e,h,g){if((f/=g/2)<1){return h/2*f*f+e}return -h/2*((-f)*(f-2)-1)+e},cubicin: function(f,e,h,g){return h*(f/=g)*f*f+e},cubicout: function(f,e,h,g){return h*((f=f/g-1)*f*f+1)+e},cubicinout: function(f,e,h,g){if((f/=g/2)<1){return h/2*f*f*f+e}return h/2*((f=-2)*f*f+2)+e},quartin: function(f,e,h,g){return h*(f/=g)*f*f*f+e},quartout: function(f,e,h,g){return -h*((f=f/g-1)*f*f*f-1)+e},quartinout: function(f,e,h,g){if((f/=g/2)<1){return h/2*f*f*f*f+e}return -h/2*((f=-2)*f*f*f*f-2)+e},quintin: function(f,e,h,g){return h*(f/=g)*f*f*f*f+e},quintout: function(f,e,h,g){return h*((f=f/g-1)*f*f*f*f+1)+e},quintinout: function(f,e,h,g){if((f/=g/2)<1){return h/2*f*f*f*f*f+e}return h/2*((f=-2)*f*f*f*f*f+2)+e},sinein: function(f,e,h,g){return -h*Math.cos(f/g*(Math.PI/2))+h+e},sineout: function(f,e,h,g){return

```

```

h*Math.sin(f/g*(Math.PI/2))+e},sineinout:function(f,e,h,g){return -
h/2*(Math.cos(Math.PI*f/g)-
1)+e},expoin:function(f,e,h,g){return(f==0)?e:h*Math.pow(2,10*(f/g-
1))+e},expoout:function(f,e,h,g){return(f==g)?e+h:h*(-Math.pow(2,-
10*f/g)+1)+e},expoinout:function(f,e,h,g){if(f==0){return e}if(f==g){return
e+h}if((f/=g/2)<1){return h/2*Math.pow(2,10*(f-1))+e}return h/2*(-Math.pow(2,-
10*-f)+2)+e},circin:function(f,e,h,g){return -h*(Math.sqrt(1-(f/=g)*f)-
1)+e},circout:function(f,e,h,g){return h*Math.sqrt(1-(f=f/g-
1)*f)+e},circinout:function(f,e,h,g){if((f/=g/2)<1){return -h/2*(Math.sqrt(1-
f*f)-1)+e}return h/2*(Math.sqrt(1-(f-=2)*f)+1)+e},elasticin:function(g,e,k,j){var
h=1.70158;var i=0;var f=k;if(g==0){return e}if((g/=j)==1){return
e+k}if(!i){i=j*0.3}if(f<Math.abs(k)){f=k;var h=i/4}else{var
h=i/(2*Math.PI)*Math.asin(k/f)}return -(f*Math.pow(2,10*(g-=1))*Math.sin((g*j-
h)*(2*Math.PI/i))+e},elasticout:function(g,e,k,j){var h=1.70158;var i=0;var
f=k;if(g==0){return e}if((g/=j)==1){return
e+k}if(!i){i=j*0.3}if(f<Math.abs(k)){f=k;var h=i/4}else{var
h=i/(2*Math.PI)*Math.asin(k/f)}return f*Math.pow(2,-10*g)*Math.sin((g*j-
h)*(2*Math.PI/i)+k)+e},elasticinout:function(g,e,k,j){var h=1.70158;var i=0;var
f=k;if(g==0){return e}if((g/=j/2)==2){return
e+k}if(!i){i=j*(0.3*1.5)}if(f<Math.abs(k)){f=k;var h=i/4}else{var
h=i/(2*Math.PI)*Math.asin(k/f)}if(g<1){return -0.5*(f*Math.pow(2,10*(g-
=1))*Math.sin((g*j-h)*(2*Math.PI/i))+e}return f*Math.pow(2,-10*(g-
=1))*Math.sin((g*j-
h)*(2*Math.PI/i)*0.5+k)+e},backin:function(f,e,i,h,g){if(g==undefined){g=1.70158}
return i*(f/=h)*f*((g+1)*f-
g)+e},backout:function(f,e,i,h,g){if(g==undefined){g=1.70158}return i*((f=f/h-
1)*f*((g+1)*f+g)+1)+e},backinout:function(f,e,i,h,g){if(g==undefined){g=1.70158}i
f((f/=h/2)<1){return i/2*(f*f*((g*(1.525))+1)*f-g)+e}return i/2*((f-
=2)*f*((g*(1.525))+1)*f+g)+2)+e},bouncein:function(f,e,h,g){return h-
a.bounceOut(g-f,0,h,g)+e},bounceout:function(f,e,h,g){if((f/=g)<(1/2.75)){return
h*(7.5625*f*f)+e}else{if(f<(2/2.75)){return h*(7.5625*(f-
(1.5/2.75))*f+0.75)+e}else{if(f<(2.5/2.75)){return h*(7.5625*(f-
(2.25/2.75))*f+0.9375)+e}else{return h*(7.5625*(f-
(2.625/2.75))*f+0.984375)+e}}}},bounceinout:function(f,e,h,g){if(f<g/2){return
a.bounceIn(f*2,0,h,g)*0.5+e}return a.bounceOut(f*2-g,0,h,g)*0.5+h*0.5+e}};return
a})();
/* tween.js */ var Tween=function(){var a={};var c=true;var
b={init:function(){return b},busy:function(){var e=false;for(var d in
a){e=true;break}return e},to:function(g,e,p){var f=new Date().valueOf();var
d={};var q={from:{},to:{},colors:{},node:g,t0:f,t1:f+e*1000,dur:e*1000};var
o="linear";for(var j in p){if(j=="easing"){var h=p[j].toLowerCase();if(h in
Easing){o=h}continue}else{if(j=="delay"){var
m=(p[j]||0)*1000;q.t0+=m;q.t1+=m;continue}}if(Colors.validate(p[j])){q.colors[j]=
[Colors.decode(g.data[j]),Colors.decode(p[j]),p[j]];d[j]=true}else{q.from[j]=(g.d
ata[j]!=undefined)?g.data[j]:p[j];q.to[j]=p[j];d[j]=true}q.ease=Easing[o];if(a[g
._id]==undefined){a[g._id]=[]a[g._id].push(q);if(a.length>1){for(var
l=a.length-2;l>=0;l++){var n=a[l];for(var j in n.to){if(j in d){delete
n.to[j]}else{d[j]=true}}for(var j in n.colors){if(j in d){delete
n.colors[j]}else{d[j]=true}}if($.isEmptyObject(n.colors)&&$.isEmptyObject(n.to)){
a.splice(1,1)}}}c=false},interpolate:function(e,h,i,g){g=(g||"").toLowerCase();va
r d=Easing.linear;if(g in Easing){d=Easing[g]}var
f=d(e,0,1,1);if(Colors.validate(h)&&Colors.validate(i)){return
lerpRGB(f,h,i)}else{if(!isNaN(h)){return lerpNumber(f,h,i)}else{if(typeof
h=="string"){return(f<0.5)?h:i}}}},tick:function(){var f=true;for(var d in
a){f=false;break}if(f){return}var e=new
Date().valueOf();$.each(a,function(i,h){var g=false;$.each(h,function(p,t){var
o=t.ease((e-t.t0),0,1,t.dur);o=Math.min(1,o);var r=t.from;var s=t.to;var
j=t.colors;var l=t.node.data;var m=(o==1);for(var n in s){switch(typeof
s[n]){case"number":l[n]=lerpNumber(o,r[n],s[n]);if(n=="alpha"){l[n]=Math.max(0,Math
.min(1,l[n]))}break;case"string":if(m){l[n]=s[n]}break}}for(var n in
j){if(m){l[n]=j[n][2]}else{var
q=lerpRGB(o,j[n][0],j[n][1]);l[n]=Colors.encode(q)}}if(m){t.completed=true;g=true

```

```

    });if(g){a[i]=$.map(h,function(j){if(!j.completed){return
j}});if(a[i].length==0){delete a[i]}}};c=$.isEmptyObject(a);return c}};return
b.init());var lerpNumber=function(a,c,b){return c+a*(b-c)};var
lerpRGB=function(b,d,c){b=Math.max(Math.min(b,1),0);var
a={};$.each("rgba".split(""),function(e,f){a[f]=Math.round(d[f]+b*(c[f]-
d[f]))});return a};

    arbor = (typeof(arbor)!='undefined') ? arbor : {}
    $.extend(arbor, {
        // not really user-serviceable; use the ParticleSystem's .tween* methods
    instead
        Tween:Tween,

        // immutable object with useful methods
    colors:{
        CSS:Colors.CSS,          // dictionary: {colorname:#fef2e2,...}
        validate:Colors.validate, // f(str) -> t/f
        decode:Colors.decode,     // f(hexString_or_cssColor) -> {r,g,b,a}
        encode:Colors.encode,     // f({r,g,b,a}) -> hexOrRgbaString
        blend:Colors.blend       // f(color, opacity) -> rgbaString
    }
})
})(this.jQuery)

```

## Arbor-graphics.js

```

//
// arbor-graphics.js
// canvas fructose
//
// Copyright (c) 2011 Samizdat Drafting Co.
//
// Permission is hereby granted, free of charge, to any person
// obtaining a copy of this software and associated documentation
// files (the "Software"), to deal in the Software without
// restriction, including without limitation the rights to use,
// copy, modify, merge, publish, distribute, sublicense, and/or sell
// copies of the Software, and to permit persons to whom the
// Software is furnished to do so, subject to the following
// conditions:
//
// The above copyright notice and this permission notice shall be
// included in all copies or substantial portions of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
// EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
// OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
// NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
// HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
// WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
// FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
// OTHER DEALINGS IN THE SOFTWARE.
//

(function($){

    /*      etc.js */ var
    trace=function(msg){if(typeof(window)=='undefined' || !window.console){return}var
    len=arguments.length;var args=[];for(var

```



```

i=0;i<len;i++){args.push("arguments["+i+"]")}eval("console.log("+args.join(",")+
)");var dirname=function(a){var
b=a.replace(/^\//?(\.?)\//?$/,"$1").split("/");b.pop();return"/"+b.join("/");var
basename=function(b){var c=b.replace(/^\//?(\.?)\//?$/,"$1").split("/");var
a=c.pop();if(a==""){return null}else{return a}};var
_ordinalize_re=/(\d)(?=(\d\d\d)+(?!\d))/g;var ordinalize=function(a){var
b=""+a;if(a<11000){b=""+a}.replace(_ordinalize_re,"$1,")}else{if(a<1000000){b=Ma
th.floor(a/1000)+"k"}else{if(a<1000000000){b=""+Math.floor(a/1000)}.replace(_ord
inalize_re,"$1,")+m}}return b};var nano=function(a,b){return
a.replace(/\{([\w\-\.\.]*\})/g,function(f,c){var
d=c.split("."),e=b[d.shift()];$.each(d,function(){if(e.hasOwnProperty(this)){e=e[
this]}else{e=f}});return e});var objcopy=function(a){if(a===undefined){return
undefined}if(a===null){return null}if(a.parentNode){return a}switch(typeof
a){case"string":return a.substring(0);break;case"number":return
a+0;break;case"boolean":return a===true;break}var
b=($.isArray(a))?[[]:{};$.each(a,function(d,c){b[d]=objcopy(c)});return b};var
objmerge=function(d,b){d=d||{};b=b||{};var c=objcopy(d);for(var a in
b){c[a]=b[a]}return c};var objcmp=function(e,c,d){if(!e||!c){return
e===c}if(typeof e!==typeof c){return false}if(typeof e!="object"){return
e===c}else{if($.isArray(e)){if(!($.isArray(c))){return
false}if(e.length===c.length){return false}else{var h=[];for(var f in
e){if(e.hasOwnProperty(f)){h.push(f)}}var g=[];for(var f in
c){if(c.hasOwnProperty(f)){g.push(f)}}if(!d){h.sort();g.sort()}if(h.join(",")!==(g
.join(","))){return false}}var i=true;$.each(e,function(a){var
b=objcmp(e[a],c[a]);i=i&&b;if(!i){return false}});return i}};var
objkeys=function(b){var
a=[];$.each(b,function(d,c){if(b.hasOwnProperty(d)){a.push(d)}});return a};var
objcontains=function(c){if(!c||typeof c!="object"){return false}for(var
b=1,a=arguments.length;b<a;b++){if(c.hasOwnProperty(arguments[b])){return
true}}return false};var uniq=function(b){var a=b.length;var d={};for(var
c=0;c<a;c++){d[b[c]]=true}return objkeys(d)};var arbor_path=function(){var
a=$( "script" ).map(function(b){var
c=$(this).attr("src");if(!c){return}if(c.match(/arbor[^\./\.\.]*.js|dev.js/)){return
c.match(/.*\//)||"/"}));if(a.length>0){return a[0]}else{return null}};
/* colors.js */ var Colors=(function(){var f=/#[0-9a-f]{6}/i;var
b=/#(..)(..)(..)/;var c=function(h){var
g=h.toString(16);return(g.length==2)?g:"0"+g};var a=function(g){return
parseInt(g,16)};var d=function(g){if(!g||typeof g!="object"){return false}var
h=objkeys(g).sort().join("");if(h=="abgr"){return true}};var
e={CSS:{aliceblue:"#f0f8ff",antiquewhite:"#faebd7",aqua:"#00ffff",aquamarine:"#7f
ffd4",azure:"#00ffff",beige:"#f5f5dc",bisque:"#ffe4c4",black:"#000000",blanchedal
mond:"#ffebcd",blue:"#0000ff",blueviolet:"#8a2be2",brown:"#a52a2a",burlywood:"#de
b887",cadetblue:"#5f9ea0",chartreuse:"#7fff00",chocolate:"#d2691e",coral:"#ff7f50
",cornflowerblue:"#6495ed",cornsilk:"#fff8dc",crimson:"#dc143c",cyan:"#00ffff",da
rkblue:"#00008b",darkcyan:"#008b8b",darkgoldenrod:"#b8860b",darkgray:"#a9a9a9",da
rkgrey:"#a9a9a9",darkgreen:"#006400",darkkhaki:"#bdb76b",darkmagenta:"#8b008b",da
rklivegreen:"#556b2f",darkorange:"#ff8c00",darkorchid:"#9932cc",darkred:"#8b0000
",darksalmon:"#e9967a",darkseagreen:"#8fbc8f",darkslateblue:"#483d8b",darkslategr
ay:"#2f4f4f",darkslategrey:"#2f4f4f",darkturquoise:"#00ced1",darkviolet:"#9400d3"
,deeppink:"#ff1493",deepskyblue:"#00bfff",dimgray:"#696969",dimgrey:"#696969",dod
gerblue:"#1e90ff",firebrick:"#b22222",floralwhite:"#fffaf0",forestgreen:"#228b22"
,fuchsia:"#ff00ff",gainsboro:"#dcdcdc",ghostwhite:"#f8f8ff",gold:"#ffd700",golden
rod:"#daa520",gray:"#808080",grey:"#808080",green:"#008000",greenyellow:"#adff2f"
,honeydew:"#f0fff0",hotpink:"#ff69b4",indianred:"#cd5c5c",indigo:"#4b0082",ivory:
"#fffff0",khaki:"#f0e68c",lavender:"#e6e6fa",lavenderblush:"#fff0f5",lawngreen:"#
7cfc00",lemonchiffon:"#ffffac",lightblue:"#add8e6",lightcoral:"#f08080",lightcyan
:"#e0ffff",lightgoldenrodyellow:"#fafad2",lightgray:"#d3d3d3",lightgrey:"#d3d3d3"
,lightgreen:"#90ee90",lightpink:"#ffb6c1",lightsalmon:"#ffa07a",lightseagreen:"#2
0b2aa",lightskyblue:"#87cefa",lightslategray:"#778899",lightslategrey:"#778899",l
ightsteelblue:"#b0c4de",lightyellow:"#ffffe0",lime:"#00ff00",limegreen:"#32cd32",
linen:"#fafae6",magenta:"#ff00ff",maroon:"#800000",mediumaquamarine:"#66cdaa",med
iumblue:"#0000cd",mediumorchid:"#ba55d3",mediumpurple:"#9370d8",mediumseagreen:"#

```

```

3cb371",mediumslateblue:"#7b68ee",mediumspringgreen:"#00fa9a",mediumturquoise:"#4
8d1cc",mediumvioletred:"#c71585",midnightblue:"#191970",mintcream:"#f5fffa",misty
rose:"#ffe4e1",moccasin:"#ffe4b5",navajowhite:"#ffdead",navy:"#000080",oldlace:"#
fdf5e6",olive:"#808000",olivedrab:"#6b8e23",orange:"#ffa500",orangered:"#ff4500",
orchid:"#da70d6",palegoldenrod:"#eee8aa",palegreen:"#98fb98",paleturquoise:"#afee
ee",palevioletred:"#d87093",papayawhip:"#ffefd5",peachpuff:"#ffdab9",peru:"#cd853
f",pink:"#ffc0cb",plum:"#dda0dd",powderblue:"#b0e0e6",purple:"#800080",red:"#ff00
00",rosybrown:"#bc8f8f",royalblue:"#4169e1",saddlebrown:"#8b4513",salmon:"#fa8072
",sandybrown:"#f4a460",seagreen:"#2e8b57",seashell:"#fff5ee",sienna:"#a0522d",sil
ver:"#c0c0c0",skyblue:"#87ceeb",slateblue:"#6a5acd",slategray:"#708090",slategrey
:"#708090",snow:"#ffffff",springgreen:"#00ff7f",steelblue:"#4682b4",tan:"#d2b48c"
,teal:"#008080",thistle:"#d8bfd8",tomato:"#ff6347",turquoise:"#40e0d0",violet:"#e
e82ee",wheat:"#f5deb3",white:"#ffffff",whitesmoke:"#f5f5f5",yellow:"#ffff00",yell
owgreen:"#9acd32"},decode:function(h){var g=arguments.length;for(var l=g-
1;l>=0;l--){if(arguments[l]==undefined){g--}}var k=arguments;if(!h){return
null}if(g==1&&d(h)){return h}var j=null;if(typeof h=="string"){var
o=1;if(g==2){o=k[1]}var n=e.CSS[h.toLowerCase()];if(n!==undefined){h=n}var
m=h.match(f);if(m){vals=h.match(b);if(!vals||!vals.length||vals.length!=4){return
null}j={r:a(vals[1]),g:a(vals[2]),b:a(vals[3]),a:o}}else{if(typeof
h=="number"){if(g>=3){j={r:k[0],g:k[1],b:k[2],a:1};if(g>=4){j.a*=k[3]}}else{if(g>
=1){j={r:k[0],g:k[0],b:k[0],a:1};if(g==2){j.a*=k[1]}}}}return
j},validate:function(g){if(!g||typeof g!="string"){return
false}if(e.CSS[g.toLowerCase()]!==undefined){return true}if(g.match(f)){return
true}return false},mix:function(h,g,k){var j=e.decode(h);var
i=e.decode(g)},blend:function(g,j){j=(j!==undefined)?Math.max(0,Math.min(1,j)):1;
var h=e.decode(g);if(!h){return null}if(j==1){return g}var h=g;if(typeof
g=="string"){h=e.decode(g)}var i=objcopy(h);i.a*=j;return
nano("rgba({r},{g},{b},{a})",i)},encode:function(g){if(!d(g)){g=e.decode(g);if(!d
(g)){return null}}if(g.a==1){return
nano("#{r}{g}{b}",{r:c(g.r),g:c(g.g),b:c(g.b)})}else{return
nano("rgba({r},{g},{b},{a})",g)}};return e}});
/* primitives.js */ var Primitives=function(c,f,g){var
b=function(i,m,j,l,k){this.x=i;this.y=m;this.w=j;this.h=l;this.style=(k!==undefin
ed)?k:{};b.prototype={draw:function(h){this._draw(h)},_draw:function(i,n,j,l,k){
if(objcontains(i,"stroke","fill","width")){k=i}if(this.x!==undefined){i=this.x,n=
this.y,j=this.w,l=this.h;k=objmerge(this.style,k)}k=objmerge(f,k);if(!k.stroke&&!
k.fill){return}var
m=0.5522848;ox=(j/2)*m,oy=(l/2)*m,xe=i+j,ye=n+l,xm=i+j/2,ym=n+l/2;c.save();c.begi
nPath();c.moveTo(i,ym);c.bezierCurveTo(i,ym-oy,xm-
ox,n,xm,n);c.bezierCurveTo(xm+ox,n,xe,ym-
oy,xe,ym);c.bezierCurveTo(xe,ym+oy,xm+ox,ye,xm,ye);c.bezierCurveTo(xm-
ox,ye,i,ym+oy,i,ym);c.closePath();if(k.fill!==null){if(k.alpha!==undefined){c.fil
lStyle=Colors.blend(k.fill,k.alpha)}else{c.fillStyle=Colors.encode(k.fill)}c.fill
()}if(k.stroke!==null){c.strokeStyle=Colors.encode(k.stroke);if(!isNaN(k.width)){
c.lineWidth=k.width}c.stroke()}c.restore()};var
a=function(i,n,j,l,m,k){if(objcontains(m,"stroke","fill","width")){k=m;m=0}this.x
=i;this.y=n;this.w=j;this.h=l;this.r=(m!==undefined)?m:0;this.style=(k!==undefin
ed)?k:{};a.prototype={draw:function(h){this._draw(h)},_draw:function(j,o,k,m,n,l)
{if(objcontains(n,"stroke","fill","width","alpha")){l=n;n=0}else{if(objcontains(j
,"stroke","fill","width","alpha")){l=j}}if(this.x!==undefined){j=this.x,o=this.y,
k=this.w,m=this.h;l=objmerge(this.style,l)}l=objmerge(f,l);if(!l.stroke&&!l.fill)
{return}var i=(n>0);c.save();c.beginPath();c.moveTo(j+n,o);c.lineTo(j+k-
n,o);if(i){c.quadraticCurveTo(j+k,o,j+k,o+n)}c.lineTo(j+k,o+m-
n);if(i){c.quadraticCurveTo(j+k,o+m,j+k-
n,o+m)}c.lineTo(j+n,o+m);if(i){c.quadraticCurveTo(j,o,j+n,o)}if(l.fill!==null){if(l.alp
ha!==undefined){c.fillStyle=Colors.blend(l.fill,l.alpha)}else{c.fillStyle=Colors.
encode(l.fill)}c.fill()}if(l.stroke!==null){c.strokeStyle=Colors.encode(l.stroke)
;if(!isNaN(l.width)){c.lineWidth=l.width}c.stroke()}c.restore()};var
e=function(i,l,h,j,k){if(k!==undefined||typeof
j=="number"){this.points=[{x:i,y:l},{x:h,y:j}];this.style=k||{}}else{if($.isArray
(i)){this.points=i;this.style=l||{}}else{this.points=[i,l];this.style=h||{}}};e.

```

```

prototype={draw:function(h){if(this.points.length<2){return}var
j=[];if(!$.isArray(this.points[0])){j.push(this.points)}else{j=this.points}c.save
();c.beginPath();$.each(j,function(n,m){c.moveTo(m[0].x+0.5,m[0].y+0.5);$.each(m,
function(o,p){if(o==0){return}c.lineTo(p.x+0.5,p.y+0.5)}});var
i=$.extend(objmerge(f,this.style),h);if(i.closed){c.closePath()}if(i.fill!==undef
ined){var
l=Colors.decode(i.fill,(i.alpha!==undefined)?i.alpha:1);if(l){c.fillStyle=Colors.
encode(l)}c.fill()}if(i.stroke!==undefined){var
k=Colors.decode(i.stroke,(i.alpha!==undefined)?i.alpha:1);if(k){c.strokeStyle=Col
ors.encode(k)}if(!isNaN(i.width)){c.lineWidth=i.width}c.stroke()}c.restore()};va
r d=function(i,h,l,k){var
j=Colors.decode(i,h,l,k);if(j){this.r=j.r;this.g=j.g;this.b=j.b;this.a=j.a}};d.pr
ototype={toString:function(){return
Colors.encode(this)},blend:function(){trace("blend",this.r,this.g,this.b,this.a)}
};return{_Oval:b,_Rect:a,_Color:d,_Path:e}};
/* graphics.js */ var Graphics=function(c){var h=$(c);var
q=$(h).get(0).getContext("2d");var i=null;var l="rgb";var e="origin";var m={};var
p={background:null,fill:null,stroke:null,width:0};var b={};var g={font:"sans-
serif",size:12,align:"left",color:Colors.decode("black"),alpha:1,baseline:"ideogr
aphic"};var k=[];var o=Primitives(q,p,g);var f=o._Oval;var n=o._Rect;var
d=o._Color;var a=o._Path;var j={init:function(){if(!q){return null}return
j},size:function(s,r){if(!isNaN(s)&&!isNaN(r)){h.attr({width:s,height:r})}return{
width:h.attr("width"),height:h.attr("height")}},clear:function(r,u,s,t){if(argume
nts.length<4){r=0;u=0;s=h.attr("width");t=h.attr("height")}q.clearRect(r,u,s,t);i
f(p.background!==null){q.save();q.fillStyle=Colors.encode(p.background);q.fillRec
t(r,u,s,t);q.restore()}}},background:function(s,r,v,t){if(s==null){p.background=nu
ll;return null}var
u=Colors.decode(s,r,v,t);if(u){p.background=u;j.clear()}}},noFill:function(){p.fil
l=null},fill:function(s,r,v,t){if(arguments.length==0){return
p.fill}else{if(arguments.length>0){var
u=Colors.decode(s,r,v,t);p.fill=u;q.fillStyle=Colors.encode(u)}}},noStroke:functi
on(){p.stroke=null;q.strokeStyle=null},stroke:function(s,r,v,u){if(arguments.leng
th==0&&p.stroke!==null){return p.stroke}else{if(arguments.length>0){var
t=Colors.decode(s,r,v,u);p.stroke=t;q.strokeStyle=Colors.encode(t)}}},strokeWidth
:function(r){if(r===undefined){return
q.lineWidth}q.lineWidth=p.width=r,Color:function(r){return new
d(r)},drawStyle:function(s){if(arguments.length==0){return
objcopy(p)}if(arguments.length==2){var r=arguments[0];var
v=arguments[1];if(typeof r=="string"&&typeof v=="object"){var
u={};if(v.color!==undefined){var
t=Colors.decode(v.color);if(t){u.color=t}}$.each("background fill stroke
width".split("
"),function(w,x){if(v[x]!==undefined){u[x]=v[x]}});if(!$.isEmptyObject(u)){m[r]=u
}}return}if(arguments.length==1&&m[arguments[0]]!==undefined){s=m[arguments[0]]}i
f(s.width!==undefined){p.width=s.width}q.lineWidth=p.width;$.each("background
fill
stroke",function(y,x){if(s[x]!==undefined){if(s[x]===null){p[x]=null}else{var
w=Colors.decode(s[x]);if(w){p[x]=w}}}});q.fillStyle=p.fill;q.strokeStyle=p.stroke
},textStyle:function(s){if(arguments.length==0){return
objcopy(g)}if(arguments.length==2){var r=arguments[0];var
v=arguments[1];if(typeof r=="string"&&typeof v=="object"){var
u={};if(v.color!==undefined){var
t=Colors.decode(v.color);if(t){u.color=t}}$.each("font size align baseline
alpha".split("
"),function(w,x){if(v[x]!==undefined){u[x]=v[x]}});if(!$.isEmptyObject(u)){b[r]=u
}}return}if(arguments.length==1&&b[arguments[0]]!==undefined){s=b[arguments[0]]}i
f(s.font!==undefined){g.font=s.font}if(s.size!==undefined){g.size=s.size}q.font=n
ano("{size}px
{font}",g);if(s.align!==undefined){q.textAlign=g.align=s.align}if(s.baseline!==un
defined){q.textBaseline=g.baseline=s.baseline}if(s.alpha!==undefined){g.alpha=s.a
lpha}if(s.color!==undefined){var
t=Colors.decode(s.color);if(t){g.color=t}}if(g.color){var

```

```

t=Colors.blend(g.color,g.alpha);if(t){q.fillStyle=t}},text:function(s,r,z,v){if(
arguments.length>=3&&!isNaN(r)){v=v||{};v.x=r;v.y=z}else{if(arguments.length==2&&
typeof(r)=="object"){v=r}else{v=v||{}}}}var
u=objmerge(g,v);q.save();if(u.align!==undefined){q.textAlign=u.align}if(u.baselin
e!==undefined){q.textBaseline=u.baseline}if(u.font!==undefined&&!isNaN(u.size)){q
.font=nano("{size}px {font}",u)}var w=(u.alpha!==undefined)?u.alpha:g.alpha;var
t=(u.color!==undefined)?u.color:g.color;q.fillStyle=Colors.blend(t,w);if(w>0){q.f
illText(s,Math.round(u.x),u.y)}q.restore(),textWidth:function(r,t){t=objmerge(g,
t||{});q.save();q.font=nano("{size}px {font}",t);var
s=q.measureText(r).width;q.restore();return s},Rect:function(s,A,t,v,z,u){return
new
n(s,A,t,v,z,u)},rect:function(s,A,t,v,z,u){n.prototype._draw(s,A,t,v,z,u)},Oval:f
unction(r,v,s,u,t){return new
f(r,v,s,u,t)},oval:function(r,v,s,u,t){t=t||{};f.prototype._draw(r,v,s,u,t)},line
:function(s,v,r,t,u){var w=new
a(s,v,r,t);w.draw(u)},lines:function(s,u,r,t){if(typeof
t=="number"){k.push([x:s,y:u],[x:r,y:t])}else{k.push([s,u])}},drawLines:functio
n(r){var s=new a(k);s.draw(r);k=[]};return j.init());

arbor = (typeof(arbor)!='undefined') ? arbor : {}
$.extend(arbor, {
  // object constructor (don't use 'new', just call it)
  Graphics:function(ctx){ return Graphics(ctx) },

  // useful methods for dealing with the r/g/b
  colors:{
    CSS:Colors.CSS,          // dict:{colorname:"#fef2e2", ...}
    validate:Colors.validate, // f(str) -> t/f
    decode:Colors.decode,    // f(hexString_or_cssColor) -> {r,g,b,a}
    encode:Colors.encode,    // f({r,g,b,a}) -> hexOrRgbaString
    blend:Colors.blend      // f(color, opacity) -> rgbaString
  }
})
})(this.jQuery)

```