

**Trabajo de Programación Avanzada**

**Manuel Benítez Sánchez**

**Francisco López Baena**

**Edwin Mauricio Quishpe Maldonado**

**Equipo 2**

**ÍNDICE**

**1)Descripción del problema pág. 2**

**2)Requisitos pág. 3**

**3)Esquema funcional pág. 4**

**4)Diagrama entidad-relación pág. 5**

**5)Información Base de datos pág. 6**

**6)Implementación pág. 7**

**7)Pruebas pág. 9**

**8)Manual de usos pág21**

**8)Asociaciones pág22**

**9)Diagramas de secuencia de casos de uso pág23**

**10)Contratos de las operaciones pág28**

**11)Diagramas de secuencia detallados pág40**

**12)Diagrama de clases pág47**

**Notas pág. 50**

1. Descripción del problema

Actualmente los diferentes campus universitarios son muy extensos y tienen una gran cantidad de sitios de interés para estudiantes y profesores, estos pueden ser aprendidos con el paso de los años, pero hay que tener en cuenta que, no paran de recibir personal tanto estudiantes como profesores de distintas nacionalidades o de fuera de la ciudad como estudiantes séneca, con motivo a este problema pensamos que sería una buena idea disponer de una herramienta que facilitara encontrar sitios de interés tanto para personal ajeno a la universidad como para los propios estudiantes y profesores.

Por ello queríamos desarrollar una aplicación con la que podamos encontrar cualquier punto de todo el campus que sea de interés para alguien, desde paradas de autobús a un aula en concreto.

Gestionaremos asignaturas, profesores, aulas, edificios, despachos, grados y lugares importantes como pueden ser cafeterías, kioskos, paradas de bus, metro, zonas de deportes…

Toda esta gestión no será meramente un control de ellas, sino que cada lugar estará acompañado de sus correspondiente mapa de ubicación, así solucionamos el perderse en el campus o dar “vueltas” buscando un lugar.

La herramienta tiene una interfaz con la que directamente accederás a los distintos tipos de lugares que podrás visualizar en los mapas, como edificios, cafeterías, aparcamientos…

Tiene la opción de ingresar en el sistema, una vez logueado podrás gestionar asignaturas, profesores, aulas, edificios, despachos, grados y lugares, pudiendo crear, eliminar, consultar o modificar cualquiera de ellos.

La herramienta está orientada especialmente para ámbito académico, pues no trata de lugares de servicios como puede ser el de limpieza, mantenimiento… o despachos de autoridades encargadas de la gestión de la universidad en un ámbito no educativo.

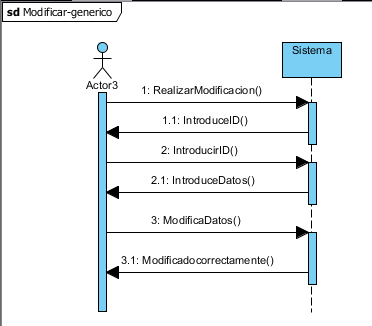
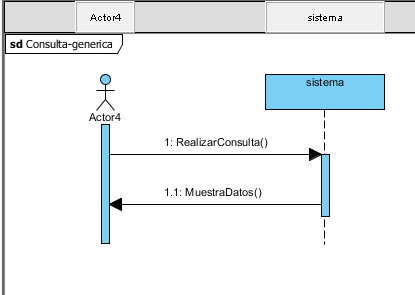
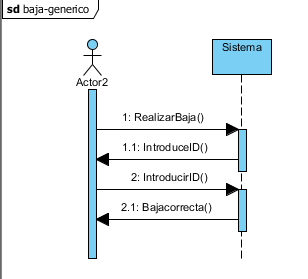
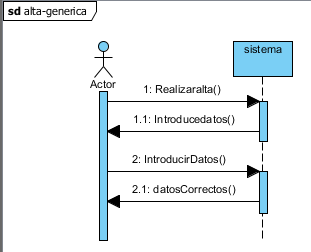
.

1. Objetivos

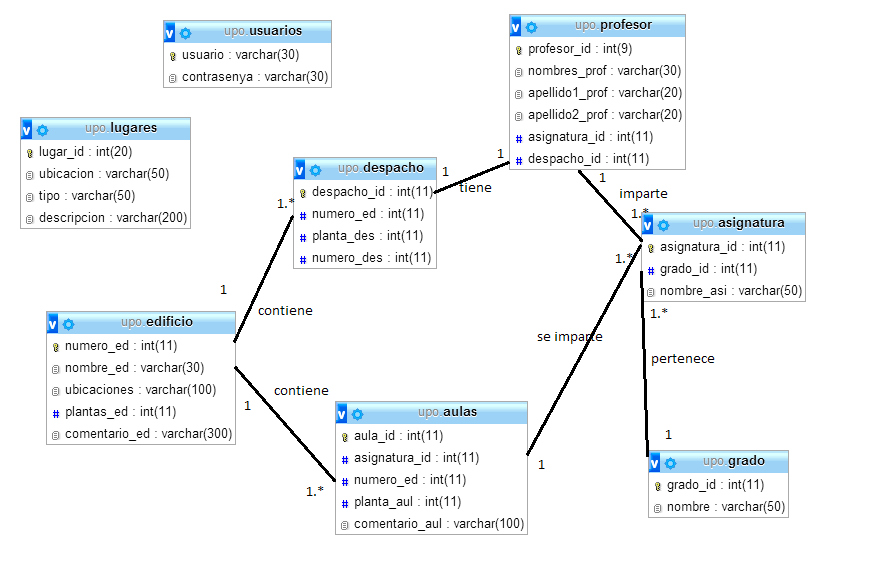
* Debe permitir dar de alta asignaturas
* Debe permitir dar de baja asignaturas
* Debe permitir consultar datos de asignaturas
* Debe permitir modificar los datos de las asignaturas
* Debe permitir dar de alta profesores
* Debe permitir dar de baja profesores
* Debe permitir consultar datos de profesores
* Debe permitir modificar los datos de los profesores
* Debe permitir dar de alta grados
* Debe permitir dar de baja grados
* Debe permitir consultar datos de grados
* Debe permitir modificar los datos de los grados
* Debe permitir dar de alta aulas
* Debe permitir dar de baja aulas
* Debe permitir consultar datos de aulas
* Debe permitir modificar los datos de los aulas
* Debe permitir dar de alta edificios
* Debe permitir dar de baja edificios
* Debe permitir consultar datos de edificios
* Debe permitir modificar los datos de los edificios
* Debe permitir dar de alta lugares de interés
* Debe permitir dar de baja lugares de interés
* Debe permitir consultar datos de lugares de interés
* Debe permitir modificar los datos de los lugares de interés
* Debe permitir dar de alta despachos
* Debe permitir dar de baja despachos
* Debe permitir consultar datos de despachos
* Debe permitir modificar los datos de despachos
* Debe permitir visualizar en el mapa un edificio
* Debe permitir visualizar en el mapa un despacho
* Debe permitir visualizar en el mapa cafeterías, restaurantes.
* Debe permitir visualizar en el mapa aparcamientos
* Debe permitir visualizar en el mapa instalaciones deportivas
* Debe permitir visualizar en el mapa lugares de gestión
* Debe permitir restringir la gestión del sistemas a solo usuarios con permisos
* Debe permitir el envío de sugerencias por email.

1. Requisitos
2. El sistema debe permitir el alta de nuevas asignaturas introduciendo nombre, y el grado al que pertenece
3. El sistema debe permitir la baja a una asignatura existente.
4. El sistema debe permitir la consulta de los datos de asignaturas
5. El sistema debe permitir la modificación de los datos de una asignatura existente.
6. El sistema debe permitir el alta de nuevos profesores introduciendo nombre, apellidos, id de la asignatura que impartirá y la id de su despacho.
7. El sistema debe permitir la baja a un profesor existente.
8. El sistema debe permitir la consulta de los datos de profesores
9. El sistema debe permitir la modificación de los datos de un profesor existente.
10. El sistema debe permitir el alta de nuevos grados introduciendo nombre.
11. El sistema debe permitir la baja a un grado existente.
12. El sistema debe permitir la consulta de los grados
13. El sistema debe permitir el alta de nuevas aulas introduciendo el número del edificio, la planta donde se encuentra, el id de la asignatura que se impartirá en ella y un comentario.
14. El sistema debe permitir la baja a un aula existente.
15. El sistema debe permitir la consulta de los datos de aulas
16. El sistema debe permitir la modificación de los datos de un aula existente.
17. El sistema debe permitir el alta de nuevos edificios introduciendo nombre, la ubicación, el número de plantas y un comentario del mismo.
18. El sistema debe permitir la baja a un edificio existente.
19. El sistema debe permitir la consulta de los datos de edificios
20. El sistema debe permitir la modificación de los datos de un edificio existente.
21. El sistema debe permitir el alta de nuevos lugares introduciendo su ubicación, el tipo y una descripción del mismo.
22. El sistema debe permitir la baja a un lugar existente.
23. El sistema debe permitir la consulta de los lugares existentes.
24. El sistema debe permitir la modificación de los datos de un lugar existente.
25. El sistema debe permitir el alta de nuevos despachos introduciendo el número del edificio, la planta y el número del despacho.
26. El sistema debe permitir la baja a un despacho existente.
27. El sistema debe permitir la consulta de los despachos existentes.
28. El sistema debe permitir la modificación de los datos de un despacho existente.
29. El sistema debe permitir visualizar los edificios existentes en un mapa seleccionando ver edificios
30. El sistema debe permitir visualizar en el mapa cafeterías, restaurantes... Seleccionando ver comidas.
31. El sistema debe permitir visualizar en el mapa aparcamientos. Seleccionando ver aparcamientos.
32. El sistema debe permitir visualizar en el mapa medios de transportes (bus, metro...). Seleccionando ver transporte.
33. El sistema debe permitir visualizar en el mapa instalaciones deportivas (pista de futbol, baloncesto…). Seleccionando ver instalaciones deportivas.
34. El sistema debe permitir visualizar en el mapa de puntos de gestión de la universidad (registro general, casa…). Seleccionando ver para estudiantes.
35. El sistema debe permitir restringir el área de gestión a los usuarios con permisos.
36. El sistema debe permitir enviar email con sugerencias sobre la aplicación.
37. Diagrama de secuencias

Los diagramas de secuencias son genéricos para todas las entidades.

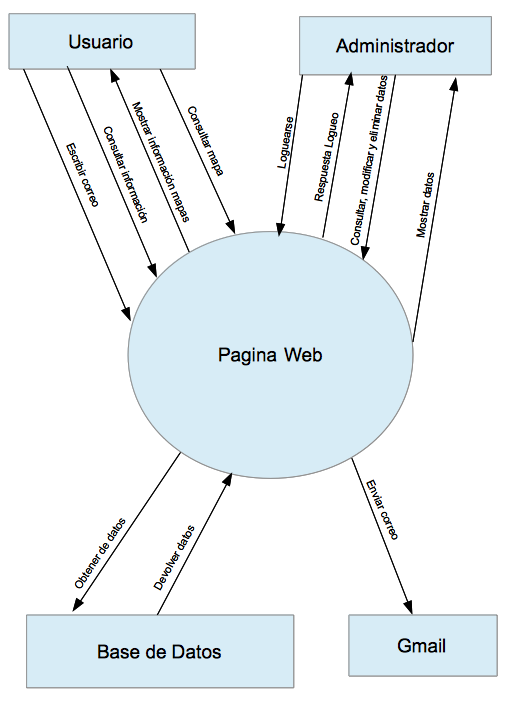


1. Diagrama de clases



1. Esquema Funcional

En esta parte vamos a explicar el funcionamiento, a través de un diagrama de funcional, de nuestra página web y como se relaciona con la base de datos y los correos. Para ello hemos dividido la web en dos. Una parte desde la interfaz de usuario y la otra parte desde la interfaz de administrador.



1. **Interfaz de usuario.**

Es la parte que corresponde al usuario final, al que va dedicado la web. Se trata de que cualquier usuario se pueda conectar a través de internet a nuestra página web y pueda consultar mediante unas funciones internas con nuestra base de datos y los mapas correspondiente para poder ubicar los edificio o lugares que desea buscar del entorno de la universidad Pablo de Olavide, de Sevilla “UPO”. El usuario no necesita tener que registrarse en nuestra web para poder hacer uso de ella. La únicas funcionalidades que tendrá el usuario será el de consultar el mapa a través de un menú lateral en el cual se mostraran los distintos edificios o lugares que pertenecen al recinto de la UPO. También podrá enviar correos sugerencias a los desarrolladores de la web para informar por si ve algún error o tal.

* index.html es la página principal, muestra una aplicación web, donde el usuario realizará la gran mayoría de las acciones. No hace falta estar registrado para acceder a ella, cualquier usuario puede acceder. Aquí es donde el usuario consultará el mapa de la UPO.

La forma en la que esta implementada la aplicación principal consta de el framework enyo (explicada en el documento). Los datos del mapa serán tratados por la web desde la carpeta Source donde tenemos varios archivos JavaScript. Para obtener los datos de la BBDD usamos ajax que funciona por medio de una api que la implementamos en php y se llama api-datos, Esta api nos devuelve un objeto JSON con el cual mostramos los datos de la petición en las listas.

* informacion.html en esta página encontraremos información sobre el proyecto web e información de los desarrolladores web que hemos realizado este proyecto. Tampoco necesitamos estar registrados para acceder a esta parte de la aplicación web. Es un simple archivo HTML.
* sugerencia.php aquí el usuario podrá enviar correos a los desarrolladores sobre sugerencias o errores pasados por alto. Se trata de una carpeta Sugerencias donde tenemos un archivo php (sugerencias.php), con su respectiva función en JavaScript para validar el formulario, que manda los datos a otro archivo php (enviar.php) el que se encarga de enviar el correo.

1. **Interfaz de administrador.**

Es la parte correspondiente a las personas que administraran la página web. Estas tarea de administración trata de mantener la base de datos a través de unos controles input y mantener la página web en el correcto funcionamiento. Estos controles estarán controlados para que no se pueda meter cosas incoherentes, con sus respectivas funciones en JavaScript.

En la carpeta Gestión tenemos todos los archivos correspondientes a esta tarea. Por cada entidad tenemos tres archivos (gestion-NombreEntidad-api.php ,gestion-NombreEntidad-script.php, gestion-NombreEntidad.php) tenemos el archivo entrar.php, gestion.php, autentificacion.php y saneo.php.

* entrar.php es la encargada de que el usuario administrador meta su usuario y contraseña para poder gestionar la aplicación web. Ésta se encargara de llamar al archivo gestion.php una vez que se haya verificado con una consulta a la base de datos de que ese usuario existe y la contraseña es válida y creara la variable sesión.

Para poder entrar en las siguientes páginas debemos de tener creado la variable sesión de lo contrario nos redirigirá a entrar.php.

* gestion.php es la página principal para el administrador. En ella tenemos un menú con las distintas entidades.
* gestion-NombreEntidad.php digamos que es la encargada de cargar la página principal de cada entidad.
* gestion-NombreEntidad-script.php En esta página tenemos los correspondientes botones e inputs de modificación, la tabla donde mostraremos los datos, validaciones de datos, etc. Para poder modificar o añadir datos tenemos que pasar unas ciertas validaciones las cuales hemos implementado con unas funciones JavaScript en esta misma página. Hemos usado ajax para el tema de las modificaciones para no tener que volver a cargar la página.

Las consultas para las modificaciones en la base de datos las tenemos implementadas otro archivo distinto llamado gestion-NombreEntidad-api.php.

* gestion-NombreEntidad-api.php es un archivo encargado de ejecutar las consultas sql que llegan desde gestion-NombreEntidad.php.
* saneo.php es la encargada de que no nos puedan hacer inyección sql.
* sesion.php controlamos que los usuarios esten autentificados.

* salir.php para destruir la variable sesión.

Para poder acceder a esta parte debemos de estar registrado en la base de datos, ya que esta parte es restringida por un fichero en php.

Todas las páginas las tenemos relacionadas con hojas de estilos.

Anexo: hemos usado muchas funciones las cuales no hemos descrito, pero que la aplicación están comentadas, ya que creemos que están mejor explicado por las personas que la han desarrollado.

Framewoks adicionales.

Para la presentación principal hemos usado el framewok ENYO y para la gestión usamos JQuery.

¿Por qué enyo?

Hemos optado por este framework porque la encapsulacíon es fácil de implementar y se mantiene de forma ordenada. Por lo tanto el motivo principal es que al trabajar en grupo es más fácil dividirnos el trabajo sin depender de los avances de otros compañeros. Ahora pasamos a explicar el funcionamiento básico del framework.

Enyo es un framework javascript, con el cual tenemos la posibilidad del despliegue de la aplicación, tanto para plataformas móviles y plataformas de escritorio(próximamente a TV). El framework nos asegura una amplia compatibilidad con los navegadores más populares del mercado(con algunas excepciones).

**Funcionamiento.**

En Enyo todo es tratado como objetos. Para crear uno basta con hacer una llamada a “enyo.kind”

Un ejemplo de cómo quedaría un objeto creado

*enyo.kind({*

*name: "holaAPP",*

*components: [*

*{tag: "p",name: "hola", content: "hola mundo"},*

*{kind: "Button", content: "Presiona", ontap: "holaTap"}*

*],*

*holaTap: function() {*

*this.$.hola.applyStyle("color", "red");*

*}*

*});*

Hemos creado un objeto llamado “holaAPP” el cual tiene elementos hijos que están definidos dentro de “components”. Estos hijos a su vez son objetos los cuales pueden contener métodos y atributos propios.

Para acceder a un objeto solo debemos llamarlo como this.$.<nombre del objeto> y seguimos una sintaxis como la de java o c pata los métodos.

Cada objeto cuenta con métodos predeterminados, por ejemplo el de cambiar el estilo. Como podemos ver en el ejemplo anterior el objeto “hola” tiene el método “applyStyle” con el que podemos cambiar el estilo del objeto. En el mismo objeto debemos hacer referencia a los eventos que obedece el mismo objeto, este es el caso de “ontap” en el cual se especifica la función a la cual obedece el evento mencionado.

**Llamadas a los objetos**

Como en Enyo todo es tratado como objetos, vamos a llamar al objeto que antes creamos, este es un ejemplo de cómo se haría.

*enyo.kind({*

*name: "MyApp",*

*components: [*

*{kind: "holaAPP"},*

*{kind: "holaAPP"}*

*]*

*});*

Hemos creado otro objeto llamado “MyApp” el cual contiene dos objetos “holaAPP” que son objetos del tipo que creamos antes.

Cada uno de estos objetos cuenta con sus métodos y atributos de forma independiente y también obedecen a métodos de su padre, en este caso “MyAPP”.

Esta es la facilidad que nos brinda Enyo, cada objeto lo podemos almacenar en un archivo distinto y hacer llamadas a sus componentes dentro de otros componentes. Esta es la principal razón por la que seleccionamos el framework para poder trabajar en equipo.

Por ultimo llamamos a nuestro objeto

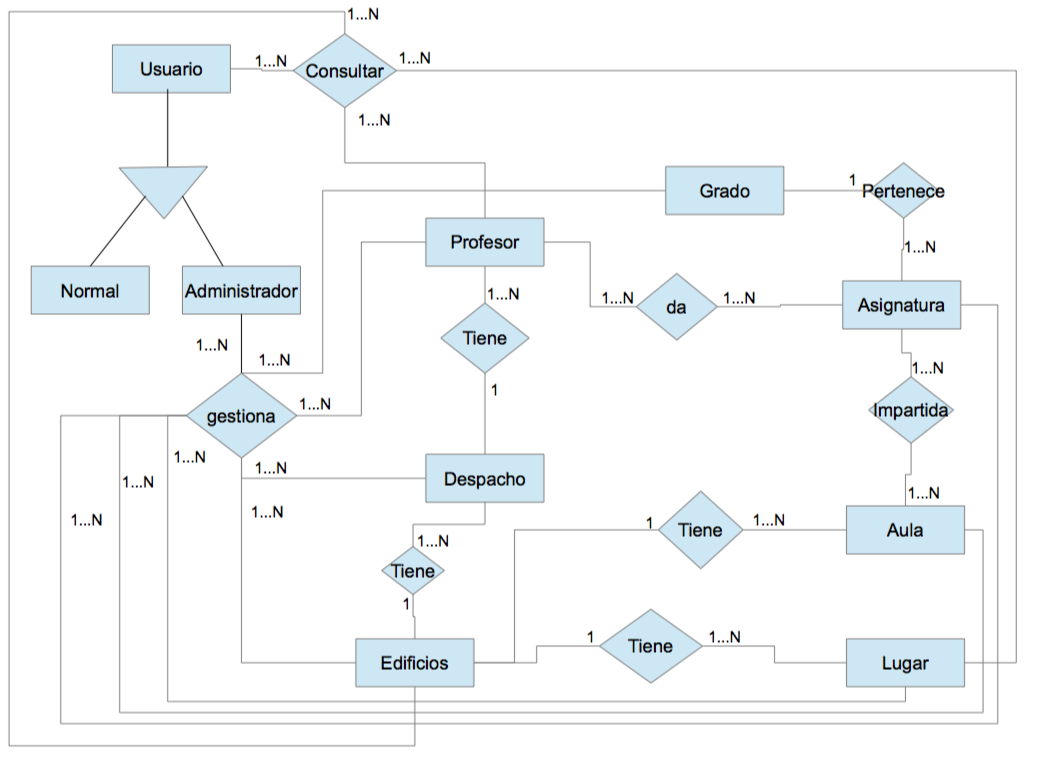
new MyApp().renderInto(document.body);

Por su puesto que si queremos introducirnos a fondo en el funcionamiento podemos visitar la página.

<http://enyojs.com>

Con estos principios básicos sobre el framework hemos finalizado con la documentación sobre los framework usados. No nos referimos a JQuery porque es el que vimos en clase.

1. Diseño entidad-relación



Nota: La relación gestiona debería ser un gestiona para cada entidad y la relación consulta debe tener una relación por cada entidad que se relaciona.

Hemos escogidos estas entidades porque son las necesarias para poder realizar una solución al problema propuesto que era la disposición de todos los puntos, por ello necesitábamos cada una de las entidades, las asociaciones son las siguientes.

* edificios contiene despachos
* edificios contiene aulas
* profesores imparten asignaturas
* asignaturas pertenecen algún grado
* profesor tiene despacho

Por otro lado tenemos a lugares y a usuarios, que, los lugares son gestionados por los usuarios.