



UNIVERSIDAD
POLITECNICA
DE VALENCIA

Escuela Técnica Superior de Ingeniería Informática



Proyecto Final de Carrera

Portal Web de Gestión de un Ayuntamiento

Autor: David Corbín Cortés

Director: Félix Buendía García



UNIVERSIDAD
POLITECNICA
DE VALENCIA

Escuela Técnica Superior de Ingeniería Informática



INDICE DEL CONTENIDO

1. Introducción.....	6
1.1. OBJETIVOS	6
1.2. MOTIVACIÓN	6
1.3. ESTRUCTURA.....	7
2. Especificación de Requisitos	9
2.1. INTRODUCCIÓN	9
2.1.1. <i>Propósito</i>	9
2.1.2. <i>Ámbito</i>	9
2.1.3. <i>Definiciones, siglas y abreviaturas</i>	9
2.1.4. <i>Referencias</i>	11
2.1.5. <i>Visión Global</i>	11
2.2. DESCRIPCIÓN GENERAL	11
2.2.1. <i>Perspectiva de producto</i>	11
2.2.2. <i>Funciones del producto</i>	12
2.2.3. <i>Características del usuario</i>	13
2.2.4. <i>Restricciones</i>	13
2.2.5. <i>Supuestos y dependencias</i>	13
2.3. REQUISITOS ESPECÍFICOS.....	13
2.3.1. <i>Requisitos de interfaces externas</i>	13
2.3.2. <i>Requisitos funcionales</i>	14
2.3.3. <i>Rendimiento y atributos</i>	16
3. Análisis.....	17
3.1. CASOS DE USO.	17
3.2. DIAGRAMA DE CLASES.	18
3.3. DIAGRAMAS DE SECUENCIA.....	20
4. Diseño.....	26
4.1. INTERFAZ DE USUARIO.....	27
4.1.1. <i>Diagramas de navegabilidad</i>	29
4.1.1.1. <i>Usuario invitado</i>	29
4.1.1.2. <i>Usuario Registrado</i>	29
4.1.1.3. <i>Usuario Administrador</i>	30
4.2. NIVEL LÓGICO O DE NEGOCIO.	31
4.3. NIVEL DE PERSISTENCIA DE DATOS.	32
4.3.1. <i>Diseño entidad relación</i>	33
5. Implementación.	34
5.1. TECNOLOGÍAS.....	34
5.2. HERRAMIENTAS.....	35
5.3. DETALLES DE LA IMPLEMENTACIÓN.....	36
5.3.1. <i>Introducción</i>	36
5.3.2. <i>Implementación de las funcionalidades</i>	36
5.3.3. <i>Uso de código externo</i>	45
5.3.4. <i>Implementación de la Interfaz gráfica</i>	46
5.3.5. <i>Implementación de la gestión de datos</i>	47
6. Evaluación y pruebas.....	49
6.1. PRUEBAS	49
6.2. ANÁLISIS DE LOS RESULTADOS.	56

7. Conclusiones.	57
7.1. VALORACIÓN PERSONAL	57
7.2. FUTURAS AMPLIACIONES O MEJORAS.	58
8. Bibliografía.....	59
8.1. LIBROS CONSULTADOS.	59
8.2. PAGINAS WEB VISITADAS.....	59
9. Anexos.....	60
9.1. ANEXO A. DISEÑO LÓGICO DE LA BASE DE DATOS	60
9.2. TOMA DE DATOS	61

INDICE FIGURAS

Figura 1.1 Portales de ejemplo-----	6
Figura 3.1: Casos de uso-----	16
Figura 3.2: Casos Uml -----	18
Figura 3.3: Diagrama de secuencia Registro usuario-----	19
Figura 3.4: Diagrama de secuencia contáctanos -----	20
Figura 3.5: Diagrama de secuencia descarga de Tramites -----	20
Figura 3.6: Diagrama de secuencia reservar instalación -----	21
Figura 3.7: Diagrama de secuencia modificar datos -----	21
Figura 3.8: Diagrama de secuencia insertar datos -----	22
Figura 3.9: Diagrama de secuencia Añadir instalación -----	22
Figura 3.10: Diagrama de secuencia administrar instalación -----	23
Figura 3.11: Diagrama de secuencia sugerencia -----	23
Figura 4.1: Arquitectura de diseño-----	24
Figura 4.2: Interfaz -----	26
Figura 4.3: Diagrama navegabilidad invitado -----	27
Figura 4.4: Diagrama navegabilidad registrado -----	28
Figura 4.5: Diagrama navegabilidad administrador -----	28
Figura 4.6: Nivel lógico -----	29
Figura 4.7: Diseño entidad relación -----	31
Figura 5.1: Código verificar -----	35
Figura 5.2: Conexión -----	36
Figura 5.3: Consulta insertar -----	36
Figura 5.4: Consulta select -----	36
Figura 5.5: Crear sesión -----	37
Figura 5.6: Finalizar sesión -----	37
Figura 5.7: Iniciar sesión -----	37
Figura 5.8: Consultar datos -----	38
Figura 5.9: Insertar datos -----	39
Figura 5.10: Actualizar datos -----	39
Figura 5.11: Eliminar datos -----	40
Figura 5.12: Código visitas -----	40
Figura 5.13: Realizar reservar-1 -----	41
Figura 5.14: Realizar reservar-2 -----	42
Figura 5.15: Reservar instalación -----	43
Figura 5.16: Álbum -----	44
Figura 5.17: Tiempo -----	44
Figura 5.18: CSS -----	45
Figura 5.19: Crear BD -----	46
Figura 5.20: Crear BD consultas -----	46
Figura 5.20: Base de datos creada -----	46
Figura 6.0: depurador Dreamweaver-----	47
Figura 6.1: Estadísticas navegadores -----	47
Figura 6.2: Visión en el opera -----	48
Figura 6.3: Visión en el Firefox -----	48
Figura 6.3: Visión en el Internet Explorer -----	49
Figura 6.5: Validar links -----	49
Figura 6.6: Resolución 800x600 -----	50
Figura 6.7: Resolución 1024x768 -----	51
Figura 6.8: Resolución 1440x900 -----	51
Figura 6.9: Validar html y CSS -----	52
Figura 6.10: Código validación html -----	52
Figura 6.11: Código validación CSS -----	52

1. Introducción

La siguiente memoria describe el trabajo realizado en el desarrollo de la creación de un portal Web para el ayuntamiento de Silla, cabe decir que este portal no tendrá nada que ver con la página oficial del ayuntamiento de Silla.

1.1. *Objetivos*

El objetivo principal es crear un portal Web donde los usuarios que accedan, aparte de poder ver contenidos estáticos de información como pueden ser la historia del pueblo, sus fiestas, localización, equipo del gobierno, también puedan acceder a contenidos dinámicos como puedan ser noticias, sugerencias, votaciones, reserva de instalaciones, etc.

Los usuarios que accedan a la Web podrán registrarse de forma gratuita para así tener acceso a servicios adicionales, como ponerse en contacto con personas del ayuntamiento, añadir sugerencias y votar a la mejor, reservar instalaciones, etc.

Los empleados del ayuntamiento también tendrán acceso a una parte privada del portal donde podrán añadir nuevas noticias, organizar actividades, ver las sugerencias por orden de votación que añadan los usuarios, etc.

1.2. *Motivación*

La motivación que me lleva a la realización de este proyecto es la de aprender a desarrollar por completo una aplicación Web que interactúa con los usuarios, y no se limita solamente a ofrecer información.

Este proyecto me ha motivado a aprender un lenguaje nuevo como PHP que hasta ahora aun no sabia utilizarlo, y ahora tengo algo de soltura, y el manejo de bases de datos MySQL, con los que ya había realizado una pequeña página aplicación en Visual Basic relacionada con un proyecto anterior de la carrera. De esta forma he aprendido a unir estas dos tecnologías para realizar este proyecto

Para coger ideas he visitado algunas páginas Web de ayuntamientos de grandes ciudades como Valencia, Barcelona, como también la de pueblos como Chirivella, alcacer, Játiva, Gandia. Y así poder ver qué funcionalidades son más comunes y necesarias que pueda necesitar un pueblo interactuando con sus habitantes y así implementarlo.

Algunos ejemplos de estos portales:



Figura 1.1 Portales de ejemplo.

1.3. Estructura

La memoria de este proyecto está estructurada en el orden cronológico en el que se han realizado cada una de las fases para desarrollarlo.

A continuación vemos un resumen del contenido de cada uno de los capítulos:

Introducción: se hace una pequeña presentación del proyecto donde se destacan los objetivos a cumplir, y las motivaciones.

Especificación de requisitos: en este capítulo se detallan las funcionalidades que debe tener la aplicación Web.

Análisis: aquí se indica de que forma debe comportarse la aplicación Web ante los distintos tipos de usuario.

Diseño: en este capítulo se define de que forma está estructurada la aplicación Web.

Implementación: aquí se detalla el proceso de implementación de la Web, indicando las tecnologías y herramientas usadas.

Evaluación y pruebas: en este capítulo se describen las pruebas realizadas sobre la aplicación Web para comprobar que su funcionamiento es el correcto. Se muestran los resultados y se hace un análisis de ellos.

Conclusiones: aquí se dan las conclusiones que se han obtenido sobre la realización del proyecto, y una valoración personal sobre el trabajo realizado.

Bibliografía: aquí indicamos el material que se ha consultado (libros, artículos, páginas Web) para obtener ayuda durante la realización del proyecto.

Anexos: en este último apartado colocamos la información que complementa a algunos de los puntos expuestos en otros capítulos.

2. Especificación de Requisitos

2.1. *Introducción*

2.1.1. Propósito

La finalidad de aquí es conseguir una aplicación Web de un ayuntamiento y explicar de una forma general y sin entrar en detalles del software y objetivos del portal, ya que aun no se ha implementado.

2.1.2. Ámbito

Se desarrollara una aplicación Web para el ayuntamiento de silla, pero podría ser adaptada para cualquier pueblo. Esta aplicación constara de varias secciones que el usuario podrá elegir entre sus intereses. Cabe destacar que es una pagina no oficial de este pueblo.

Será un portal para cualquier persona, pudiendo ser visualizada por cualquier con acceso a Internet y tener acceso a cualquier servicio como ver la historia y fiestas populares, localización en el mapa, noticias, agricultura local, sin embargo para hacer otros servicios como pagos de impuestos como el agua, la basura, añadir sugerencias etc., tendrá que registrarse para poder realizar dichos servicios.

Por otra parte los trabajadores del ayuntamiento tendrán su propio espacio y por tanto tendrán acceso para poder modificar ciertos apartados, añadir noticias, y cualquier otra cosa.

2.1.3. Definiciones, siglas y abreviaturas

Portal Web: es una aplicación basada en html que trata algún tema en particular pudiendo hacer ser interactiva con el usuario

Foro: lugar de discusión para cualquier usuario y dar su opinión sobre algún tema

Registro: darse de alta en una base de datos, con datos personales para poder saber que es la persona que dice ser.

Usuario registrado: cualquier usuario que ha pasado por el registro

Usuario no registrado: persona que solo le interesa el portal Web para visualizarlo y ver sus contenidos sin necesidad de registrarse.

Usuario con privilegios: persona que será capaz de actualizar el portal Web y manejar la base de datos, siempre que no infrinja la ley orgánica de protección de datos (LOPD).

Login: a la hora de registrarse es el nombre que la persona decide para poder entrar en el portal.

Logout: Desconectarse como usuario registrado, pasando a ser usuario invitado.

Pass Word: será una cadena alfanumérica, que solo el usuario conoce para poder acceder a zonas exclusivas del Portal.

Bd: base de datos

Php: es un lenguaje de programación diseñado para crear páginas Web. Usado para interpretar scripts del lado del servidor. Puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz grafica.

MySQL: es una base de datos desarrolladas bajo código abierto, es un lenguaje de consulta de datos estructurada.

Apache tomcat: servidor por el cual podemos ejecutar programas desde el servidor y alojados en el cliente, además de poder alojar la pagina Web creada.

Servidor: ordenador o conjunto de ordenadores que proporcionan servicios, como paginas Web, servicios de correo, etc. De forma centralizada.

WampServer: servidor de prueba utilizado para testear el portal como local que simula a pion, aquí también se instalan phpadmin y un Server apache entre otras cosas.

Eclipse: compilador con el cual se pueden realizar paginas Web, programas ejecutados por el servidor para que pueda visualizarlos el cliente

CSS: Es una hoja de estilos para hacer la pagina Web mas atractiva al usuario, se usa en programación html con Dreamweaver.

JavaScript: programa que se ejecuta al iniciar una pagina Web como algún tipo de movimiento o al presionar sobre algún enlace.

Dreamweaver: programa de Adobe para la creación de documentos Html (paginas Web) con la cual se podrán usar CSS y java script

Html: Lenguaje de Marcas de Hipertexto, es el lenguaje predominante para la construcción de páginas Web, es usado para describir la estructura y el contenido en forma de texto. Pueden incluir scripts y las etiquetas están rodeadas por <>

Ftp: es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basada en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

Http: Protocolo de transferencia de hipertexto es el protocolo usado en cada transacción de la Web (WWW). HTTP fue desarrollado por el consorcio W3C.

Https: es el mismo protocolo anterior pero con transacciones de forma segura usando SSL.

Banner: anuncios publicitarios.

Pop - us: anuncios emergentes.

2.1.4. Referencias

- Guía para la realización y supervisión de proyectos Web (pfc) REF 2007.297
- Guía para la realización y supervisión de proyectos Web (pfc) REF 2008
- Estándar IEEE 830-1998
- Enciclopedia libre Wikipedia.

2.1.5. Visión Global

Los dos siguientes apartados del documento ERS, hacen una descripción de las necesidades de este proyecto Web.

En primer lugar se hace una descripción general de los apartados que tendrá la Web, y los usuarios que intervienen en ella.

En el siguiente apartado, se describe de una forma más específica el funcionamiento de los distintos módulos que componen las distintas secciones del portal Web.

2.2. *Descripción general*

2.2.1. Perspectiva de producto

Este producto cualquier otro producto software, y por tanto no forma parte de ningún proyecto más amplio que lo englobe.

El portal Web se desarrollará en HTML, con Dreamweaver, junto con el lenguaje de programación PHP para generar el contenido dinámico. Accederá a una base de datos MySQL para obtener los datos necesarios para generar dichas páginas. Es necesario por lo tanto trabajar con un sistema que nos ofrezca un servidor Web con soporte para PHP junto con una base de datos MySQL.

Para llevar a cabo el desarrollo de la aplicación de una forma más cómoda, y antes de que sea publicada en Internet, he optado por utilizar el paquete WampServer para Windows.

Que nos ofrece de forma conjunta el servidor Web, la base de datos, y el soporte de PHP

Las funcionalidades de este servidor son las siguientes:

- Gestionar los servicios de Apache y MySQL,
- Cambiar de línea / fuera de línea (dar acceso a todos o sólo local)
- Instalar Apache, MySQL y PHP
- Gestión de la configuración de los servidores
- Acceder a los registros
- Acceder a los archivos de configuración
- Crear alias

2.2.2. Funciones del producto

Proporcionar un portal Web compuesto por varias secciones donde los usuarios que accedan a él puedan realizar las siguientes tareas:

- Obtener información sobre el municipio.
 - Leer la historia del pueblo.
 - Ver su localización.
 - Informarse sobre sus fiestas.
 - Leer noticias de actualidad del pueblo.
 - Ver las fechas de contribución
 - Ver la formación del gobierno
 - El saludo del alcalde
 - Escuelas deportivas y como contactar
- Poder registrarse como usuario para poder acceder a zonas restringidas.
 - Añadir sugerencias y poder votarlas.
 - Modificar sus datos
 - Reservar instalaciones deportivas
- Leer noticias publicadas por empleados del ayuntamiento se distinguen entre:
 - Deportivas
 - Fiestas
 - Oposiciones al ayuntamiento
 - Culturales
 - Otros
- Realizar trámites administrativos.
 - Descargarse los archivos correspondientes
 - Enviar un E-mail para alguna consulta interna o problema
- Enlaces de interés.
 - DGT
 - Ayuntamiento de valencia
 - Agencia tributaria
- Administrador
 - Añadir noticias
 - Administras las reservas instalaciones deportivas y las sugerencias en la base de datos (esto es de forma interna en el php admin)

2.2.3. Características del usuario

Este producto va dirigido para todo el público, aunque principalmente para los ciudadanos del municipio, y para los trabajadores del ayuntamiento.

Hay tres tipos de usuario:

- **Ciudadano (registrado o no):** este usuario podrá ver los contenidos del sitio, y si está registrado podrá realizar otras tareas adicionales como tramitar recibos, reservar instalaciones deportivas, etc. Pero en ningún caso podrá realizar tareas de administración sobre el sitio.
- **Trabajador del ayuntamiento:** es necesario que esté registrado en el portal, y según los privilegios que tenga en el sitio, podrá realizar determinadas tareas, como por ejemplo actualizar contenidos o recibir mensajes de ciudadanos.
- **Administrador:** es el usuario que tiene todos los privilegios sobre el sitio, y será capaz de dar privilegios a los usuarios, gestionar cualquier tipo de contenido, como la base de datos etc.

2.2.4. Restricciones

Para poder acceder al portal, se deberá usar un navegador Web.

Ciertas secciones de la Web deben ser de acceso restringido, se exigirá al usuario su registro si aún no lo ha hecho, y su posterior autenticación. De esta forma se consigue que un usuario según sus privilegios pueda acceder a determinados contenidos.

2.2.5. Supuestos y dependencias

En este caso, el sistema operativo utilizado para acceder al portal es indiferente. Por tanto podemos decir que este producto es multiplataforma.

Asimismo, el navegador Web utilizado también puede ser cualquiera de los disponibles en el mercado. Aunque este producto se ha optimizado para el navegador Mozilla Firefox versión 3 o superior.

2.3. *Requisitos específicos*

2.3.1. Requisitos de interfaces externas

El portal Web se divide en varias secciones, claramente diferenciadas:

- Área de información y servicios. Donde los ciudadanos podrán consultar cualquier tipo de información, y realizar trámites administrativos con el ayuntamiento.

- Área de registro de usuarios. Donde cualquier usuario puede registrarse gratuitamente para poder acceder a las zonas cuyos contenidos son restringidos.
- Área de administración. Donde los empleados del ayuntamiento o el administrador del sitio Web pueden acceder mediante login para poder manejar la parte interna del sitio Web. El usuario administrador tendrá permisos para poder realizar cualquier operación. Mientras que los empleados del ayuntamiento únicamente tendrá permisos para poder manejar las secciones internas de las que sean encargados.

Boceto de interfaz

Título: pantalla inicial con el logotipo del ayuntamiento y el escudo que lo caracteriza.

Propósito: A un simple vistazo al usuario le impactara la imagen inicial y se quedara con su imagen principal.

Poción: centro que ira cambiando según los enlaces presionados sin cambiar el inicio del interfase.

Formato: será una imagen del conjunto escudo y letra.

Texto: no tiene.

Interacción: no tiene, en cualquier caso se podrá hacer algún tipo de animación en el logotipo.

Tamaño: fijo

Medidas: según convenga.

2.3.2. Requisitos funcionales

Usuarios sin registrar

- Registro de usuarios.

Cualquier visitante de la Web puede registrarse para poder acceder a los contenidos o servicios restringidos. Asimismo, los empleados del ayuntamiento también deberán registrarse para que posteriormente el administrador les asigne los permisos necesarios para acceder a la intranet donde podrán actualizar contenidos de la Web, publicar noticias, leer mensajes, etc.

Los campos que serán requeridos para el registro de cualquier usuario son: nombre, apellidos, dirección, ciudad, correo electrónico, suscripción a noticias, nombre de login, y contraseña. Se comprobará que no se deje ningún campo obligatorio sin rellenar, y que el nombre de login o el correo electrónico no existan ya en la base de datos.

- Gestor de mensajes.

Cualquier usuario podrá enviar un email al ayuntamiento, podrán recibir mensajes que los ciudadanos les envíen a través de la Web. También tendrán la opción de contestar a esos mensajes mediante un formulario a la dirección de correo electrónico que haya dejado el ciudadano. Estos mensajes son enviados a un correo especial del ayuntamiento que será leído periódicamente.

- Consultar información.

Los visitantes podrán consultar los apartados de información de la Web, como puede ser la historia del pueblo, las fiestas, noticias divididas por secciones, ver sugerencias, etc.

- Consultar callejero

Los visitantes podrán ver un mapa con las calles pueblo, y la forma de cómo acceder a la localidad.

- Visualizar sugerencias

El usuario podrá ver las sugerencias que hay en marcha en la Web pero para dar su voto o introducir una nueva tendrá que registrarse.

- Visualizar instalaciones

El usuario podrá ver las instalaciones que dispone el ayuntamiento para sus ciudadanos. Aunque para poder hacer una reserva a través de la web, el usuario deberá registrarse.

Usuarios registrados

Podrán realizar todas las acciones de un usuario sin registrar, incluyendo las siguientes:

- Editar su perfil

Los usuarios podrán modificar algunos de sus datos personales que introdujeron cuando se registraron en el portal, como su nombre completo, su email, o la posibilidad de cambiar la contraseña.

- Sugerencias

Los usuarios registrados pueden insertar una sugerencia para ayudar a mejorar el pueblo y además podrán votar la mejor sugerencia para que se lleve a cabo por parte del ayuntamiento.

- Reservar instalaciones

Los usuarios podrán reservar las instalaciones de las que dispone el ayuntamiento. Se comprobará si la instalación está libre para la fecha que se quiere reservar.

Trabajador ayuntamiento

- Noticias

El trabajador del ayuntamiento mediante un usuario y pass Word especial podrán añadir o borrar noticias en cualquier sitio con tan solo su identificación, así las noticias siempre estarán actualizadas a la última. Los campos de estas noticias contendrán: una categoría, un título, una descripción, y una imagen (opcional) y podrán ser divididas por secciones.

- Sugerencias

Gestionara las sugerencias para que no exista ninguna ofensiva, además las podrá ver la sugerencia mas votada

- Reserva instalaciones

Ira actualizando las lugares o plazas que quedan libres ya sea como instalaciones deportivas o reservas para algún evento

2.3.3. Rendimiento y atributos

La eficiencia del portal Web, dependerá de la capacidad de proceso del servidor donde se aloje, y del ancho de banda de la red disponible para el servidor y para el cliente. El sistema operativo elegido por el servidor o por el cliente no es un factor importante para la eficiencia, ya que el portal Web puede trabajar correctamente sobre cualquiera de ellos.

Para realizar las tareas de mantenimiento o actualización del sitio Web, el administrador o cualquier usuario con los permisos necesarios podrá acceder desde cualquier lugar con conexión a Internet. Esto es posible gracias a que la información se encuentra guardada en la base de datos del servidor.

3. Análisis.

3.1. Casos de uso.

Los casos de uso es una técnica para capturar los requisitos de un nuevo sistema y las acciones que pueden realizar los usuarios con los escenarios que proporciona el sistema. Se intenta evitar aquí las jergas técnicas para que así un usuario no iniciado en la materia pueda entender el papel de cada usuario en el portal web y de que forma interactuar.

En este sistema intervienen tres tipos de usuarios: el usuario invitado o sin registrar, el usuario registrado y el usuario registrado administrador., que tendrá permisos para modificar las bases de datos.

A continuación en la figura 3.1 se muestran los casos de uso del sistema.

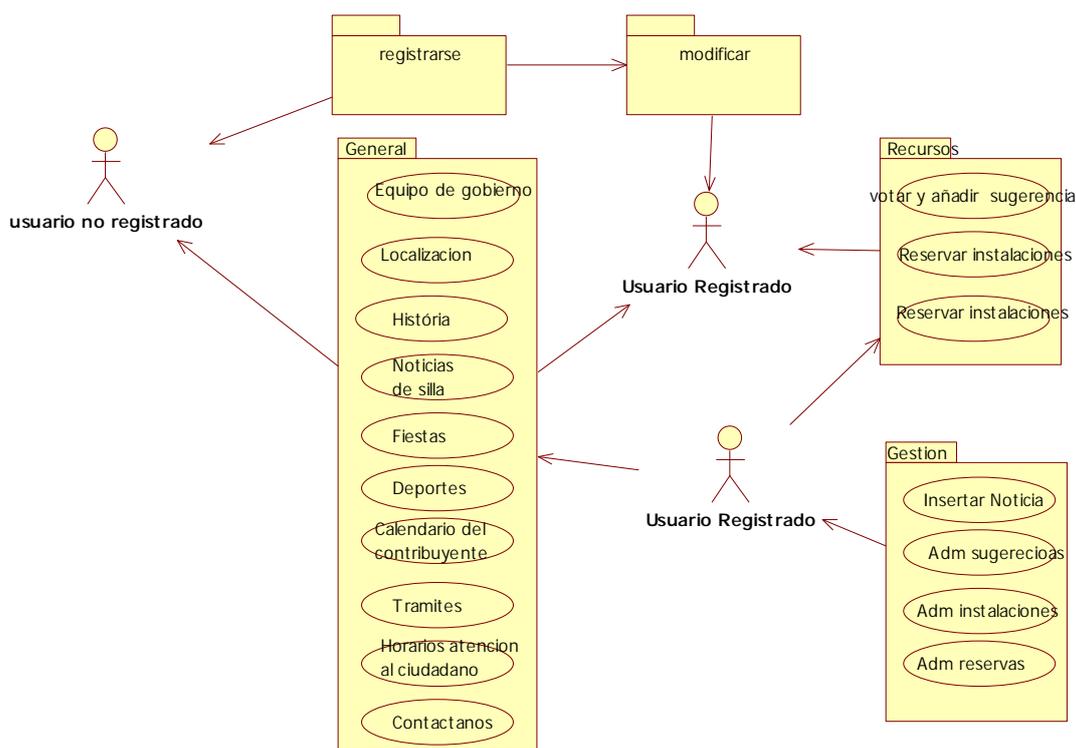


Figura 3.1: Casos de uso

3.2. *Diagrama de clases.*

Con el diagrama de clases UML podemos ver de una forma esquemática la organización de las clases en nuestro sistema.

La clase principal es el 'Ayuntamiento', que a su vez se compone de las clases 'Gestión', 'Servicio', 'Ciudad' y 'Usuario'.

- La clase 'Ciudad' está compuesta de otras clases que son de tipo informativo como: la historia, la localización, calendario contribuyente, información deportes y las fiestas.
-
- La clase 'Servicios' esta formada por clases que componen servicios que ofrece el ayuntamiento para usuarios registrados
 - Sugerencias. Aquí los usuarios registrados podran añadir y/o votar la sugerencia que mas les convenga.
 - Instalaciones. Aquí, estos usuarios podrán reservar las instalaciones que deseen dependiendo si el horario esta disponible y quedan pistas libres.
 - Noticias. Aquí solo los usuarios registrador con privilegios podrán añadir diferentes tipos de noticias para que cualquier usuario pueda estar informado
- La clase 'Usuario' está compuesta por los tipos de usuario que se contemplan en el uso del portal web. Por un lado están los usuarios invitados o sin registrar, y por otro los registrados, que se dividen a su vez en dos tipos de usuario: los registrados y los administradores.
- La clase 'Gestión' esta compuesta por a su ves las clases sugerencias, noticias e instalaciones, aquí solo los usuarios registrados con remisos de administración podrán gestionar estos submenús.

A continuación se muestra el diagrama UML donde se puede ver de forma grafica lo antes explicado:

3.3. Diagramas de secuencia.

Son usados para modelar la interacción entre objetos en un sistema modelado mediante UML. Se muestran a continuación un conjunto de objetos de aplicación a través del tiempo y se modela para cada método de la clase. Estos diagramas contienen detalles de implementación del escenario ayuntamiento.

Registro usuario:

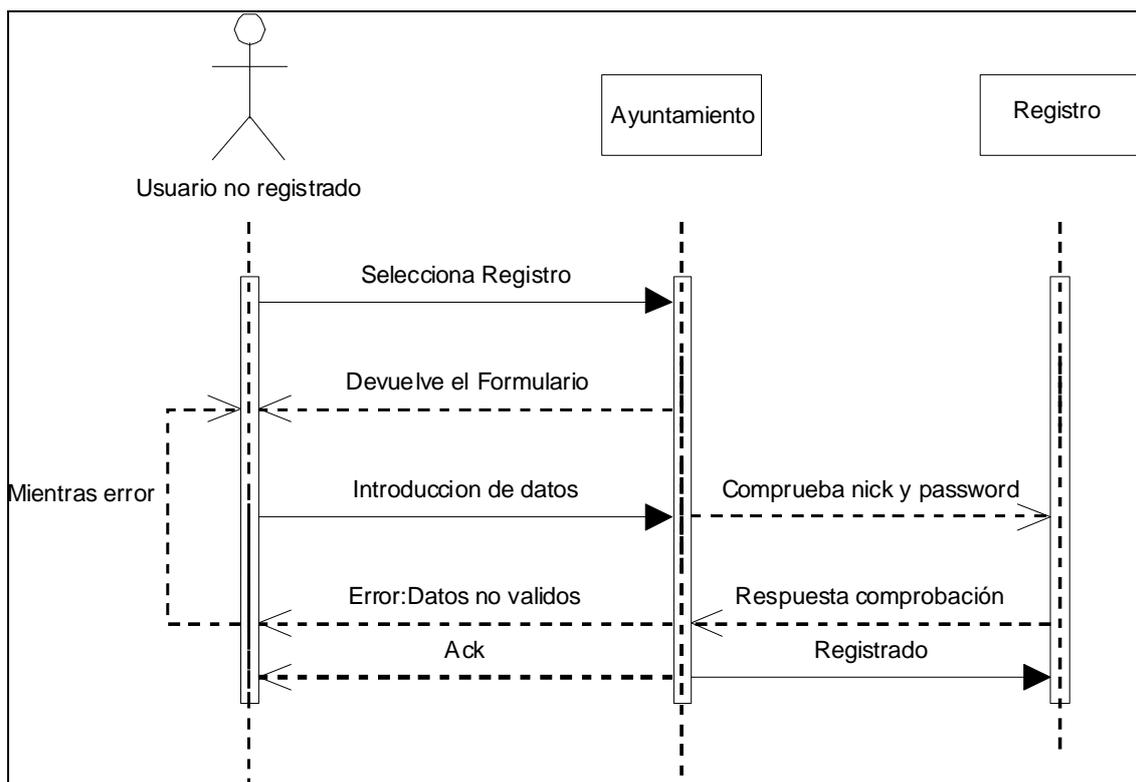


Figura 3.3: Diagrama de secuencia Registro usuario

Cuando se selecciona el registro se devuelve un formulario que mientras no sean correctos se les volverá a pedir a los usuarios que introduzcan los datos hasta que sean correctos

Contáctanos:

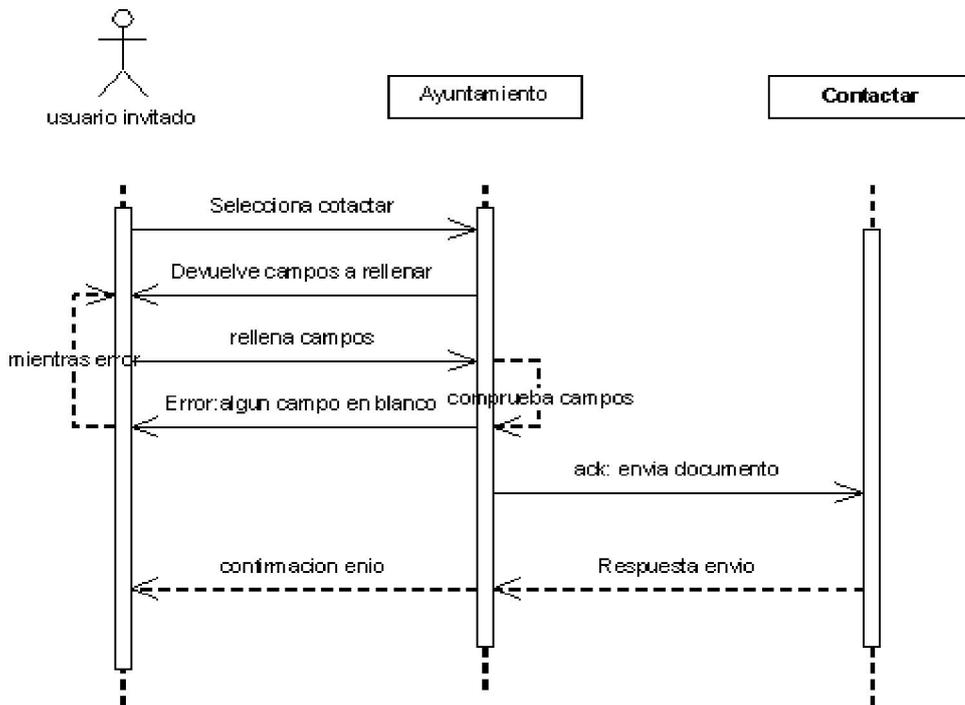


Figura 3.4: Diagrama de secuencia contáctanos

Al seleccionar la opción contactar el usuario debe rellenar unos campos requeridos sin fallos para que se pueda enviar de forma correcta

Tramites:

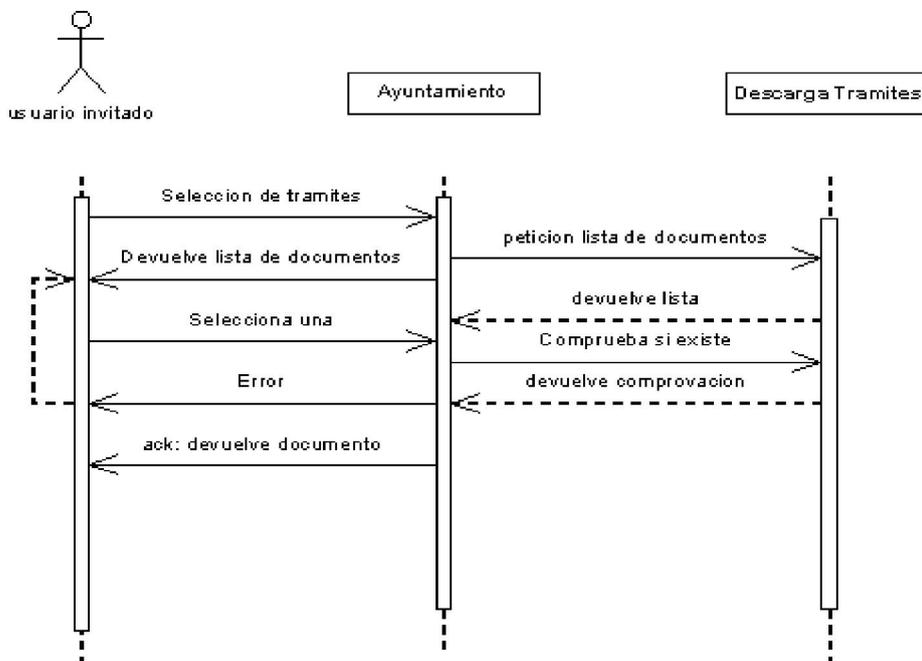


Figura 3.5: Diagrama de secuencia descarga de Trámites

Al seleccionar trámites se devuelve una lista de los documentos disponibles en el portal en pdf para rellenar por el ciudadano para realizar algún trámite personal.

Reservar instalación:

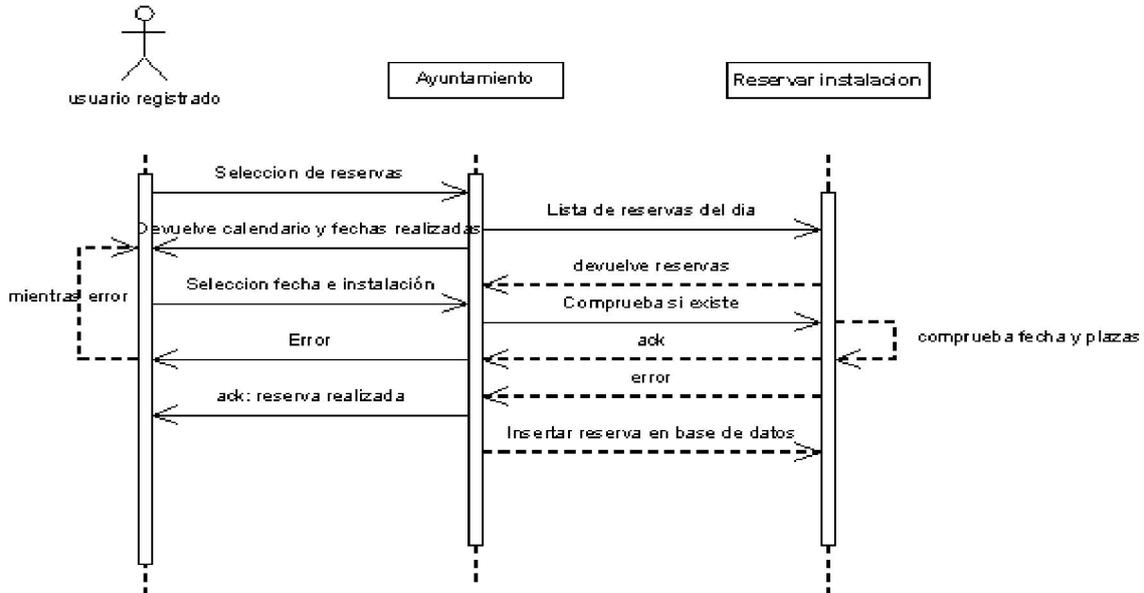


Figura 3.6: Diagrama de secuencia reservar instalación

Al seleccionar las reservas te devuelve una lista de instalaciones disponibles, reservas y si esta disponible se almacena y muestra la lista de reservas para ese día, en caso contrario da error y te muestra la misma lista para que veas que horas están ocupadas

Modificar datos:

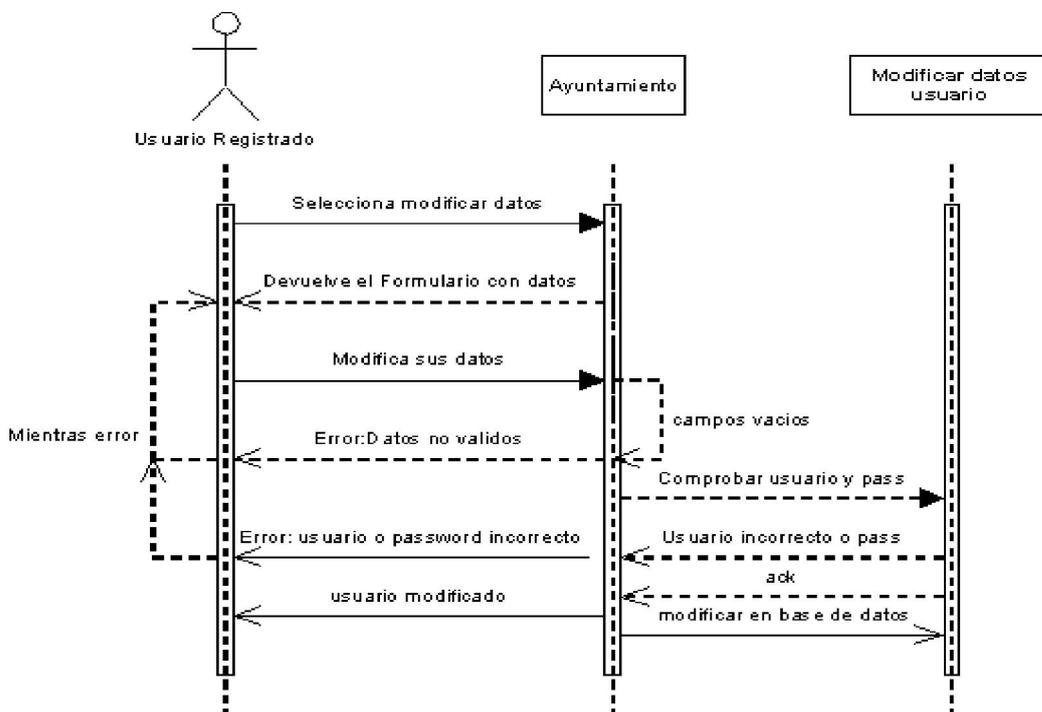


Figura 3.7: Diagrama de secuencia modificar datos

Los usuarios registrados pueden modificar sus datos, al seleccionarlo les devuelve un formulario para que lo modifiquen a su gusto mientras los datos sean correctos.

Insertar noticia:

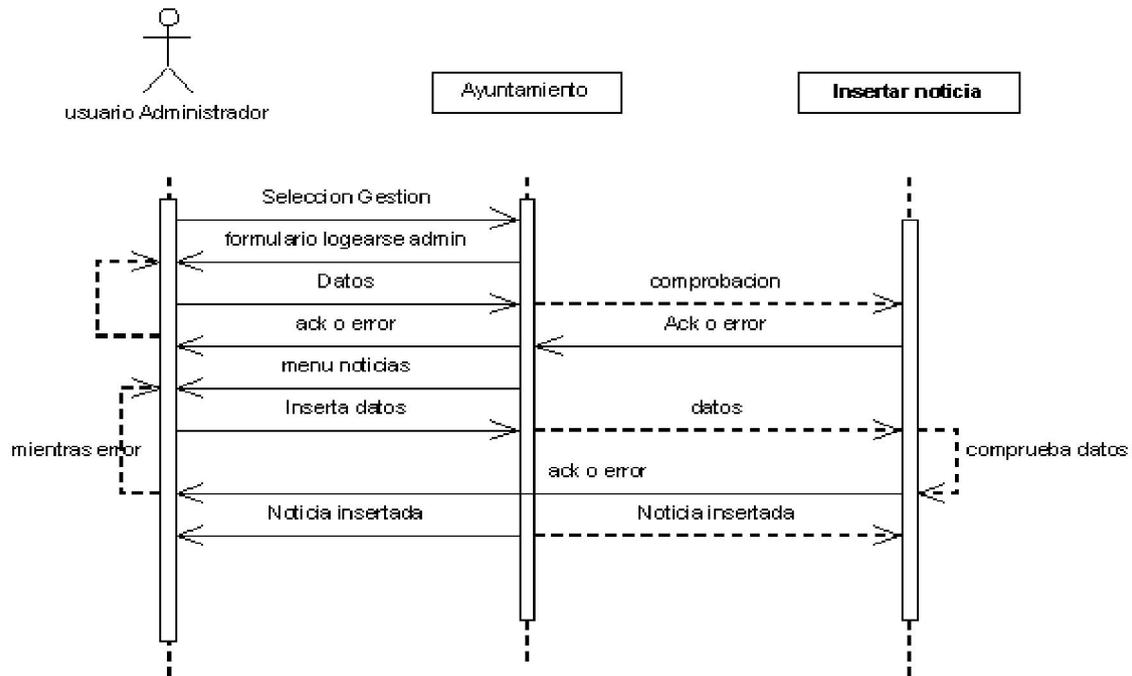


Figura 3.8: Diagrama de secuencia insertar datos

Los usuarios administradores insertan noticias siempre que se validen como tal, después de la validación se le devuelve un formulario con los campos a rellenar mientras sean correctos

Añadir instalación

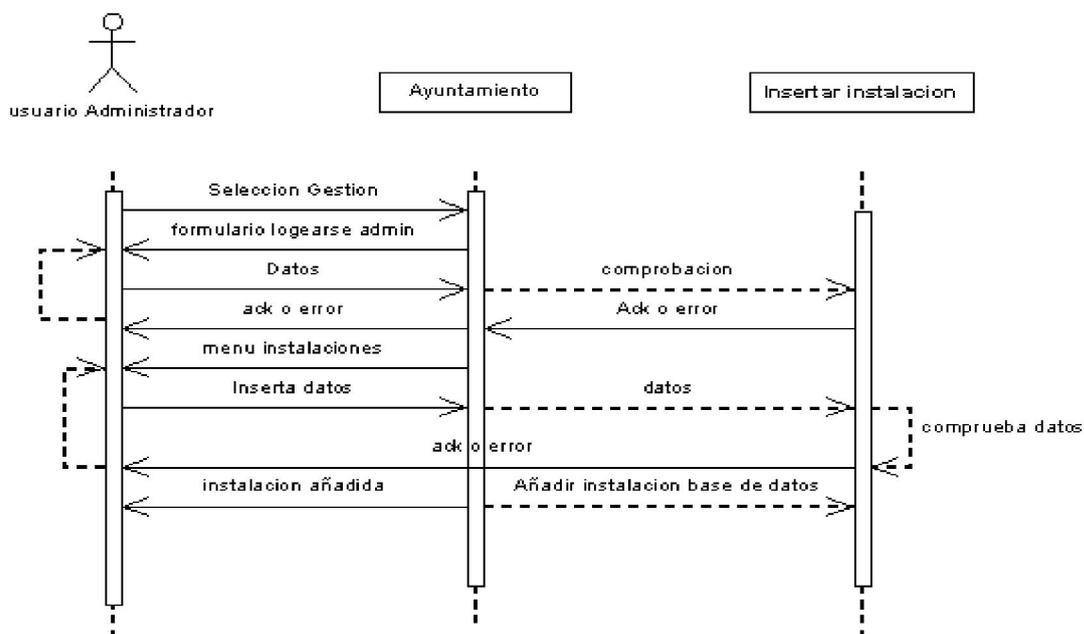


Figura 3.9: Diagrama de secuencia Añadir instalación

El administrador después de identificarse, se le devuelve un formulario con campos a rellenar para insertar una instalación mientras los datos sean correctos.

Administrar instalaciones:

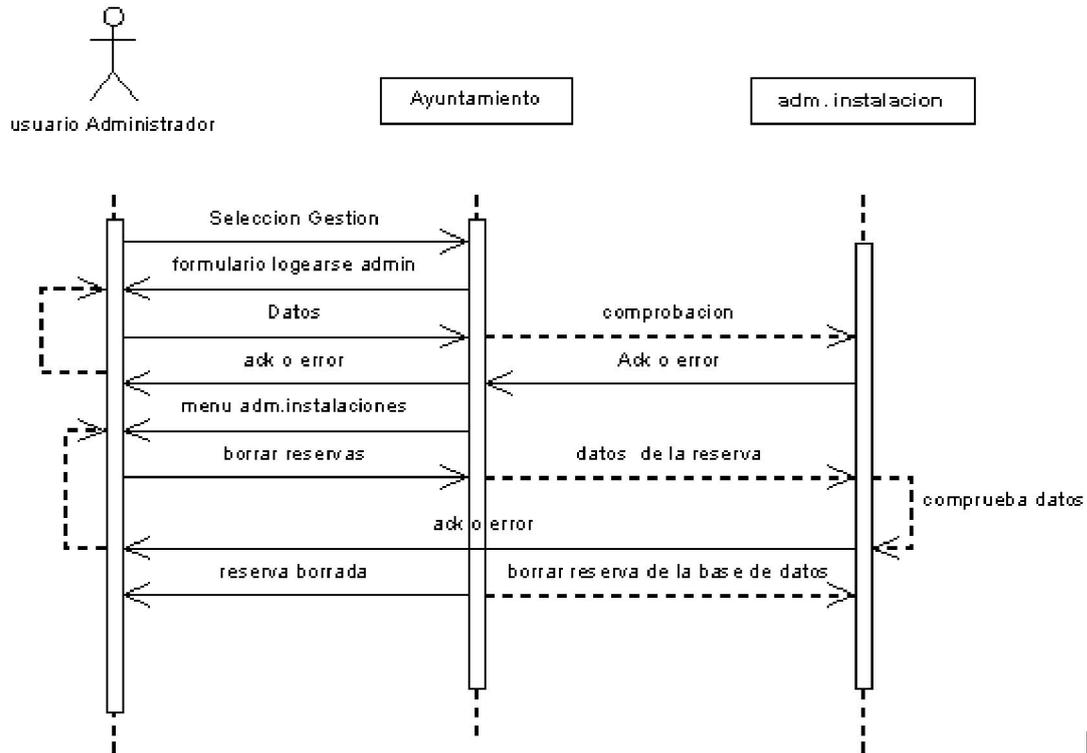


Figura 3.10: Diagrama de secuencia administrar instalación

El administrador después de logearse y seleccionar administrar instalación, se le devuelve todas las reservas realizadas hasta el momento pudiendo borrarlas todas menos el día actual o seleccionando la reserva a eliminar

Sugerencia:

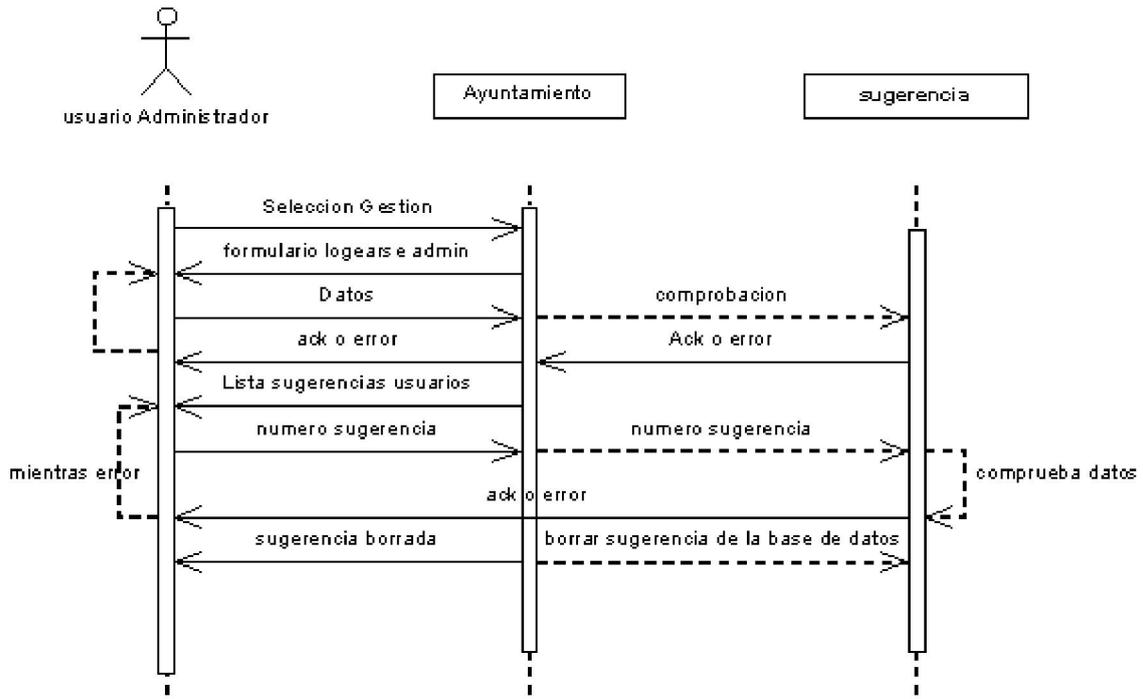


Figura 3.11: Diagrama de secuencia sugerencia

El usuario administrador después de logearse, y seleccionando el apartado sugerencias, podrá eliminar el numero de sugerencia siempre que sea un valor numérico y este entre los valores de las sugerencias registradas.

4. Diseño.

La implementación que he realizado se basa en la programación por capas, las principales ventajas que se obtiene usando este tipo de programación es que si hace falta modificar algo, se ataca directamente a la capa afectada sin necesidad que las otras se vean alteradas. En mi caso he elaborado las tres capas, pero en un entorno ha gran escala las diferentes capas se pueden dividir en diferentes grupos de trabajo incluso puede ser escalable pudiendo superar los tres niveles básicos.

Los tres niveles utilizados son los siguientes:

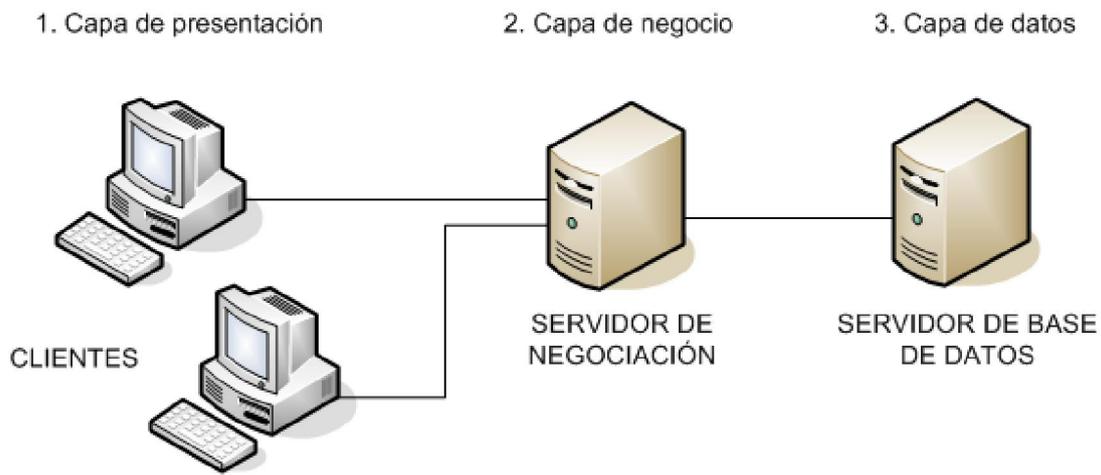


Figura 4.1: Arquitectura de diseño.

1. Capa de presentación, es la que ve el usuario o cliente, esta capa también puede denominarse capa de usuario, esta capa es la que comunica la información y recoge los datos necesarios del usuario, comprobando a su vez los datos errores de formularios etc. Esta capa debería ser entendible y fácil de usar para el usuario, debe tener una interfaz grafica amigable. Esta capa solo se comunica con la de negocio
2. Capa de negocio, aquí es donde se ejecutan las operaciones, los datos a operar son los enviados por el usuario en la anterior capa, y tras el computo es la que envía a la capa superior el resultado del proceso, también se denomina a esta capa lógica de negocio, porque es aquí donde están las reglas que deben cumplirse, para así comunicarse con la capa de datos para solicitar al gestor de base de datos que debe almacenarla o recuperar datos de el, e incluso eliminarlos.
3. Capa de datos, es básicamente la que se encarga de almacenar los datos y acceso a los mismos, a su vez puede estar gestionada por uno o mas gestores, que realicen el almacenamiento o acceso, esta capa solo se comunica con la capa de negocio.

Todas estas capas pueden residir en un único ordenador, si bien lo más usual es que haya una multitud de ordenadores en donde reside la capa de presentación (son los clientes de la arquitectura cliente/servidor). Las capas de negocio y de datos pueden residir en el mismo ordenador, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más ordenadores. Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varios ordenadores los cuales recibirán las peticiones del ordenador en que resida la capa de negocio.

4.1. Interfaz de Usuario.

Se ha intentado realizar una página intuitiva y a la vez sencilla para que así los usuarios no se pierdan en menús y submenús y al final se pierdan como en muchas paginas. De esta forma al pinchar en alguna sección solo se modifica la zona central para el contenido pudiendo ir a cualquier parte desde cualquier posición.

La pagina se ve bien sin que aparezcan barras de desplazamiento desde una resolución de 800 x 600 la pagina dependiendo del tamaño de ventana se va modificando pero hasta cierto punto. Además esta optimizado para navegadores mozilla y otros que usen su misma arquitectura base como opera y chrome no siendo tan bien en Internet Explorer 8.

En la siguiente imagen se puede ver un esquema de la interface principal del portal.

Menús de navegabilidad

Acceso a usuarios



Figura 4.2: Interfaz

Zona para el contenido

Pie de página

Hay dos partes para interactuar con el usuario, columna de la izquierda y derecha, la parte superior es sin duda la cabecera donde esta el nombre del ayuntamiento y bajo del todo la cabecera, estas zonas son inamovibles durante toda la navegabilidad pero pudiendo añadir opciones según se mejore la pagina.

Hay una parte donde los usuarios pueden registrarse y en la misma donde pueden identificarse para acceder a ciertas ventajas que ello da.

Se puede ver que el apartado de contenido en el inicio, se tiene como una visita mediante fotos por el pueblo así como sus monumentos y fiestas, y un resumen del tiempo actual y de los próximos días para así hacer más ameno la visita a la pagina inicial.

Además en la columna superior derecha se dispone del menú administrador donde un usuario miembro de diseñadores, administradores, moderadores con sus respectivos permisos pueden modificar cierto contenido de la página.

4.1.1. Diagramas de navegabilidad.

A continuación se muestra para los diferentes usuarios, que pueden manejar de la página.

4.1.1.1. Usuario invitado.

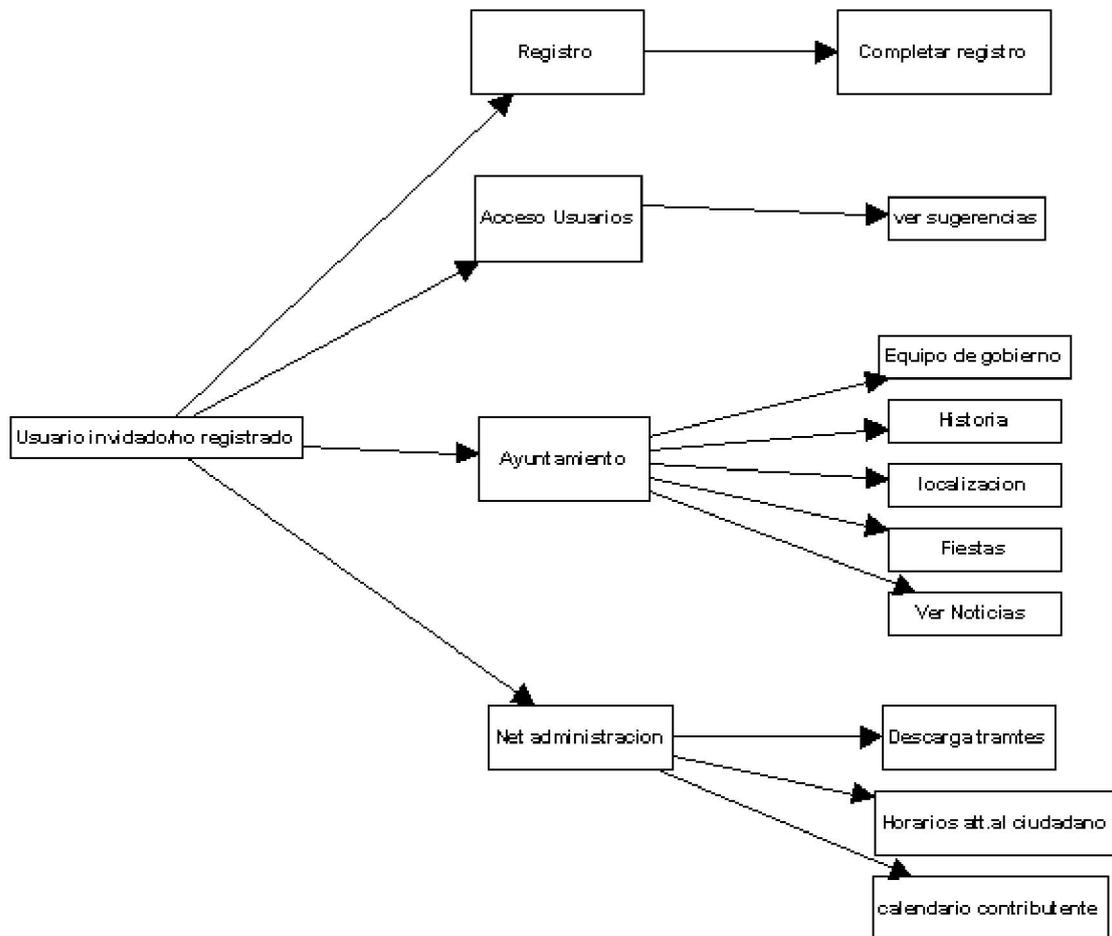


Figura 4.3: Diagrama navegabilidad invitado

4.1.1.2. Usuario Registrado.

Además de todas las acciones que un usuario invitado puede hacer, el usuario registrado puede hacer las siguientes:

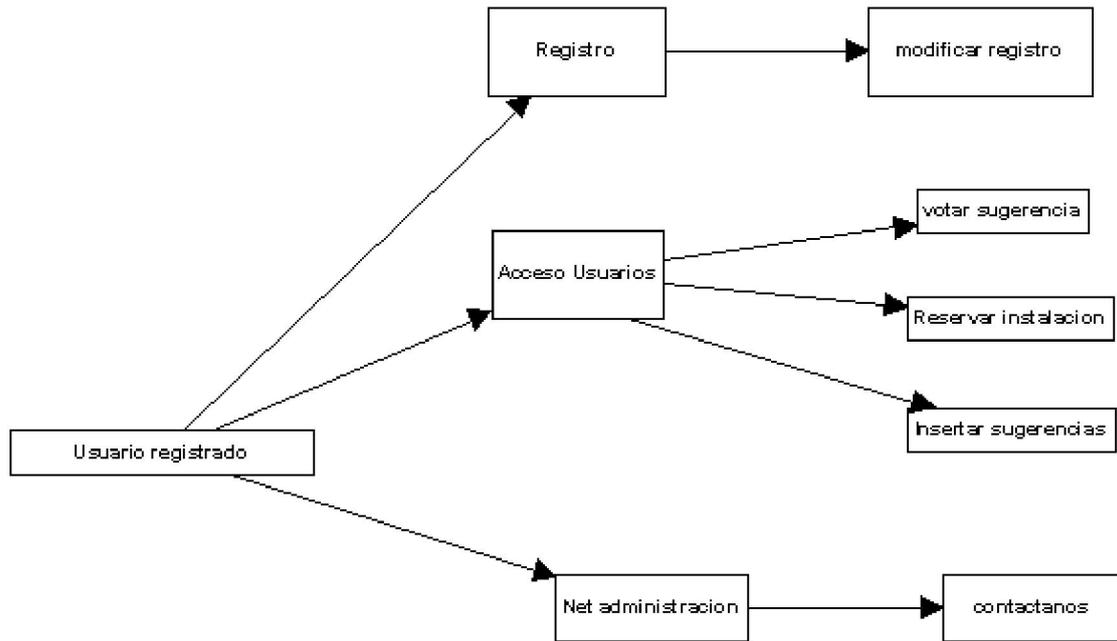


Figura 4.4: Diagrama navegabilidad registrado.

4.1.1.3. Usuario Administrador.

El usuario Administrador es el único que puede acceder a la base de datos para poder modificar apartados importantes del portal.

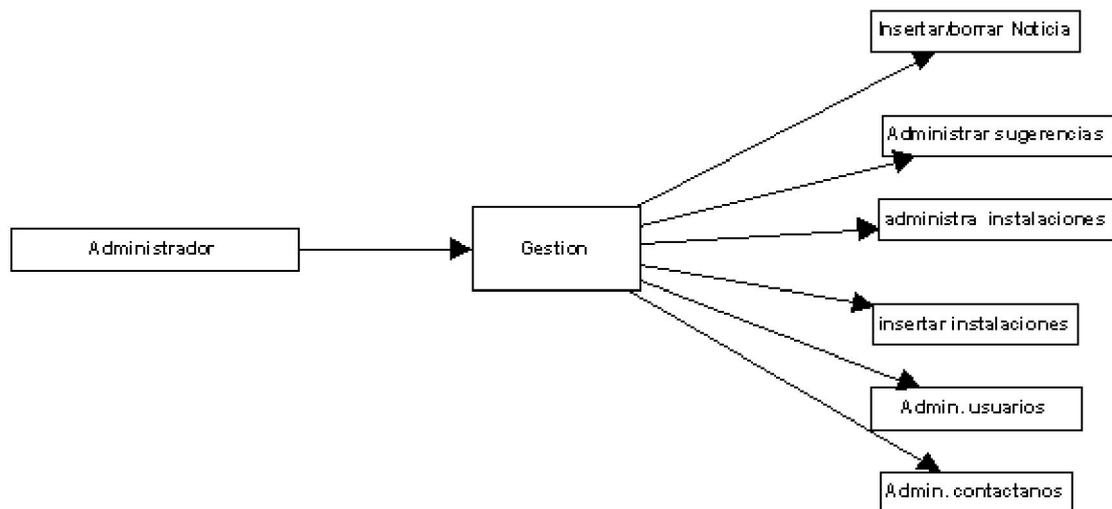


Figura 4.5: Diagrama navegabilidad administrador

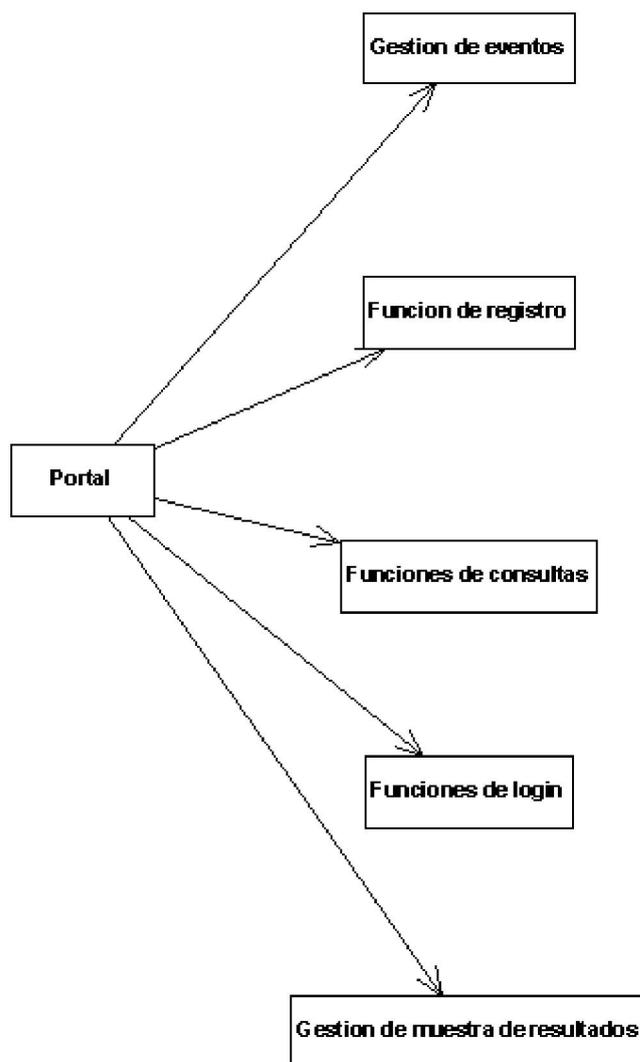
4.2. Nivel lógico o de negocio.

EL Nivel lógico permite relacionar la capa de interface de usuario con la capa de datos. Se encuentra el código que permite llevar a cabo esta relación.

En esta capa se hacen comprobaciones para validar datos, de esta forma se evita mostrar un error inesperado al usuario que debe de pasar inalterado ante estas acciones.

En caso de modificación o cambio en el sistema gestor de base de datos, bastaría con modificar el nivel lógico para adaptarlo a la nueva gestión de datos. De esta forma el nivel de interfaz de usuario no se vería afectado

Este diagrama muestra la estructura del nivel lógico de la aplicación:



En esta parte se encontraría todas las funciones de administración, tales como la gestión de instalaciones sugerencias, etc.

Aquí se encontrara una parte importante que es cuando un usuario se da de alta en el portal, además tendrán funciones de modificación de su propio perfil.

Las consultas serán todos los datos de interés como los tramites historia, noticias, etc.

Será cuando un usuario una vez logeado, inicia su sesión, donde le permite acceder a mas cosas que un usuario normal

Es la parte grafica de todos los resultados que se muestren, normalmente en centro de la pagina, para que así el usuario no pueda desorientarse.

Figura 4.6: nivel lógico

4.3. Nivel de persistencia de datos.

A este nivel se encuentra la base de datos junto con el sistema que la gestiona (SGBD).

Para diseñar la base de datos del ayuntamiento, empezamos con la base del diagrama de clases UML expuesto anteriormente. Y a partir del UML definiremos las entidades fuertes del modelo entidad relación.

Las tablas creadas para este portal sacadas desde el diagrama UML son: ‘Usuario’, ‘Noticias’, ‘instalación’, ‘sugerencias’. Después obtenemos la tabla correspondiente a la relación entre tablas, ‘Reservar_instalaciones’, que sirve para almacenar las reservas que hacen los usuarios a las instalaciones del ayuntamiento.

En todas las tablas encontraremos un identificador único autonumérico para identificar cada registro de la tabla.

Más tarde en el anexo pondremos ver en detalle la base de datos donde se especifican con más detalle los tipos de campos claves primarias, claves ajenas, auto numéricos, valores nulos o no nulos, etc.

4.3.1. Diseño entidad relación.

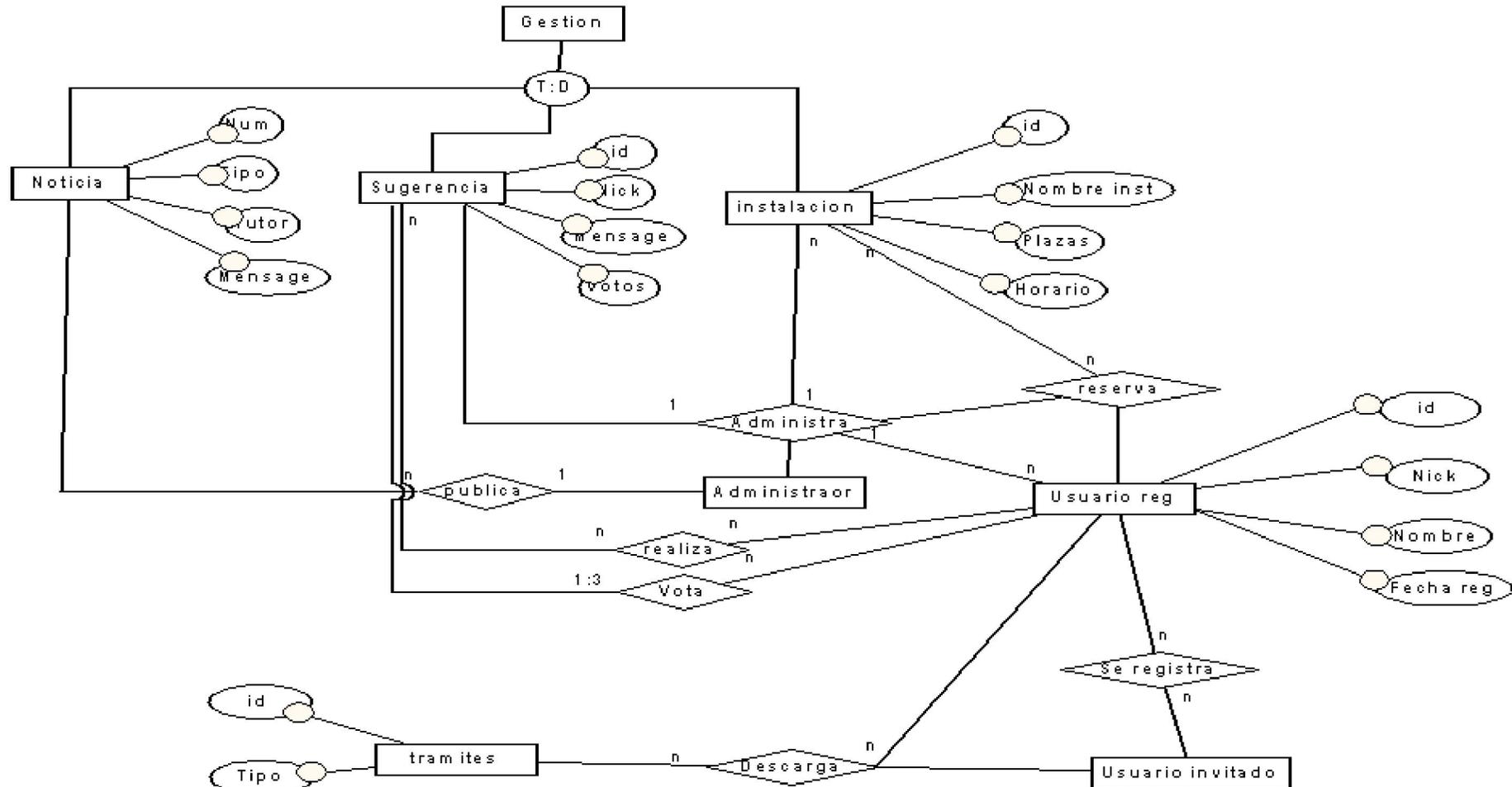


Figura 4.7: Diseño entidad relación

5. Implementación.

Para llevar a cabo la implementación del proyecto, se han utilizado distintas tecnologías tanto para la programación de la web, como para almacenar los datos que nos interesen, y para poner en marcha el servidor web.

5.1. *Tecnologías.*

Las tecnologías que se han utilizado en este proyecto son los lenguajes HTML, CSS y PHP para la programación de la web. Para la base de datos se ha utilizado MySQL. Y para el servidor web, un servidor Apache. Se ha utilizado un servidor local WAMPserver para hacer pruebas antes de subirlo al servidor pion de la upv.

WAMP es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

Windows, como sistema operativo.

Apache, como servidor Web.

MySQL, como gestor de bases de datos;

PHP (generalmente), Perl, o Python, como lenguajes de programación.

El uso de un WAMP permite servir páginas html a Internet, además de poder gestionar datos en ellas, al mismo tiempo un WAMP, proporciona lenguajes de programación para desarrollar aplicaciones web.

LAMP es el sistema análogo que corre bajo ambiente Linux

MAMP es el sistema análogo que corre bajo ambiente Mac.

Para este proyecto se ha usado la versión de Windows

HTML siglas de HyperText Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo Java script), el cual puede afectar el comportamiento de navegadores Web y otros procesadores de HTML.

Normalmente este lenguaje siempre va unido a hojas de estilo en cascada los CSS.

CSS. Las hojas de estilo en cascada (en inglés Cascading Style Sheets), CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores. Estos estilos pueden formar parte de un fichero y así poderlas usar en varios documentos html, o ir dentro del mismo fichero html.

Java script. JavaScript es un lenguaje de scripting basado en objetos sin tipo, utilizado para acceder a objetos en aplicaciones. Principalmente, se utiliza integrado en un navegador web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. JavaScript es un dialecto de ECMAScript y se caracteriza por ser un lenguaje basado en prototipos, con entrada dinámica y con funciones de primera clase. JavaScript ha tenido influencia de múltiples lenguajes y se diseñó con una sintaxis similar al lenguaje de programación Java, aunque más fácil de utilizar para personas que no programan.

Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

El uso que le he dado a este lenguaje a sido para retocar ciertas cosas de los formularios, validar datos, además de usarlo para el calendario de reservas de instalaciones donde la mayoría del código es externo.

PHP. Es un lenguaje de programación interpretado en la parte del servidor. Su sintaxis es similar a la de C. Hoy en día es uno de los más usado en la programación web. Este lenguaje se encuentra bajo la licencia 'PHP License' que es de carácter libre.

Su uso nos permite conectarnos a la base de datos para realizar funciones de consulta, inserción, actualización, o borrado. También nos permite hacer una tarea muy útil, que es la de generar código HTML de forma dinámica.

MySQL. Es el sistema gestor de bases de datos que hemos utilizado en este proyecto. Se distribuye bajo la licencia del software libre, y además es multiplataforma. Esas son algunas de las razones por las que es uno de los más usados. La utilidad de la base de datos en este proyecto ha sido la poder almacenar información y datos, como pueden ser los de los usuarios que se registran en el portal, las noticias que se publican, las reservas de las instalaciones, etc.

5.2. Herramientas.

PhpMyadmin. Es una herramienta escrita en PHP para ser usada en un navegador web, y poder administrar una base de datos MySQL. En este proyecto ha sido útil para crear las tablas de la base de datos. Para el usuario administrador al ser único es el que se me dio al principio de proyecto que es el único que puede añadir y modificar en las tablas.

StarUml. Es un editor de diagramas UML, aunque también permite trabajar con otros tipos de diagramas. En este proyecto se ha utilizado para el realizar el diagrama de clases.

Pacestar UML Diagrammer. Este programa se ha usado para el diagrama entidad relación además de todos los diagramas de secuencia.

Dreamweaver. Programa con el que se ha hecho gran parte del proyecto en el se ha programado en html, javascript, css y php.

Jalbum. Programa editor de álbumes en php, con este programa se ha diseñado la portada del portal con fotos del pueblo

PhotoShop. Es un programa de retoque y edición de imágenes de gran potencia, ha sido usado para modificar ciertos apartados del portal referente a imágenes.

5.3. Detalles de la implementación.

5.3.1. Introducción.

En este apartado se explicara el contenido de las partes más importantes de la implementación, como es el registro de usuario la gestión del menú del administrador (sugerencias, noticias, usuarios, instalaciones) y realizar reservas.

La implementación se realiza con el programa Dreamweaver, con los lenguajes de programación php, html, y en algunas partes Java script.

La parte principal se llama index.php donde se encuentran todos los enlaces y la parte principal del portal.

Hay código que se repite y es usado por varios apartados modificando solo ciertas partes, se explicara aquí ejemplos de cada uno.

5.3.2. Implementación de las funcionalidades.

Registro de usuarios

Para el registro de usuarios se emplea un código en java script para validar los campos según el estándar del portal, para luego añadirlo dentro de la base de datos.

La función verificar que aparece a continuación se usa en todos los apartados donde existe un formulario, la función puede ser mas larga o mas corta dependiendo del tamaño del formulario, una vez validados los datos pasara a una función interna que conectara la base de datos y añadirá los datos.

```
,  
function verificar() {  
  var form = document.alta;  
  
  if (form.usuario.value == ""){  
    alert("Por favor, escriba un nombre de usuario.");  
    form.usuario.focus();  
    return;  
  }  
  else if(form.pass.value == ""){  
    alert("Por favor, escriba una contrase\u00f1a.");  
    form.pass.focus();  
    return; |  
  }  
  else if(form.repass.value == ""){  
    alert("Por favor, reescriba la contrase\u00f1a.");  
    form.repass.focus();  
    return;  
  }  
  else if(form.pass.value != form.repass.value){  
    alert("Las contrase\u00f1as no coinciden, vuelva a escribirlas.");  
    form.pass.focus();  
    form.pass.value = "";  
    form.repass.value = "";  
    return;  
  }  
  else if(form.nombre.value == ""){  
    alert("Por favor, escriba su nombre y apellidos.");  
    form.nombre.focus();  
    return;  
  }  
  else if(form.email.value == ""){  
    alert("Por favor, escriba su email.");  
    form.email.focus();  
    return;  
  }  
  
  else {  
    form.submit();  
    return;  
  }  
}
```

Figura 5.1: C\u00f3digo verificar

Cuando son correctos todos los datos se hace un submit y se redirecciona a una función superior que se encuentra en otro archivo.

Conexión a la base de datos.

La primera parte de esta función es la conexión a la base de datos.

```
$config['host'] = 'localhost';
$config['user'] = 'ayto';
$config['pass'] = 'dacorcor';
$config['db'] = 'ayuntamiento_silla';

mysql_connect($config['host'], $config['user'], $config['pass']) or die(' Error en tu
usuario o password');
mysql_select_db($config['db']) or die('Fallo en la conexion a la Base de datos ');
```

Figura 5.2: conexión

Una vez la conexión sea correcta se recogen los datos con un GET o POST y se almacena en variables locales para después insertarlas como muestra la siguiente figura.

```
mysql_query('INSERT INTO usuarios (usuario, pass, nombre, email, date)
VALUES (" . $usuario . "', " . $pass . "', " . $nombre . "', " .
$email . "', " . $date . "')') or $error = true;
```

Figura 5.3: consulta insertar

Este tipo de código no solo se utiliza en el registro de usuarios, es empleado en otras funciones como sugerencias, noticias, instalaciones etc.

Autenticar usuario.

Esta acción esta implementada en el archivo login.php. Además en todas aquellas operaciones donde se requiera que un usuario este logeado se añadirá líneas de código para verificar que de verdad es así.

Se conecta a la base de datos como antes hemos explicado. Luego se busca al usuario que se ha registrado en la base de datos.

```
$log = $_POST['user'];
$pass = $_POST['pass'];
$sql = mysql_query("SELECT * FROM usuarios WHERE usuario = '$log' and pass='$pass'");
```

Figura 5.4: consulta select

Después de autenticar al usuario se crean las variables de sesión que serán utilizadas a lo largo de su visita por el portal Web según sean necesarias.

```
$num=mysql_num_rows($sql);
if($num==0){echo $err; ?> <META HTTP-EQUIV="REFRESH" CONTENT="2; http://pion.disca.upv.es/ayto/Ayuntamiento_Silla/"> <? }
else {

    $autenticado="si";
    $row = mysql_fetch_array($sql);
    $_SESSION["autenticado"]=$autenticado;
    $_SESSION["id"] = $row["num"];
    $_SESSION["usuario"] = $row["usuario"];
    $_SESSION["nombre"] = $row["nombre"];
    $_SESSION["email"] = $row["email"];
    $_SESSION["pass"] = $row["pass"];
    $_SESSION["asvotado"] = 0;
    $_SESSION["reserva_num"] = 0;
    $_SESSION["reserva_nom"] = "";

    ?> <META HTTP-EQUIV="REFRESH" CONTENT="0; http://pion.disca.upv.es/ayto/Ayuntamiento_Silla/"> <?
}
```

Figura 5.5: crear sesión

En caso de que el usuario o la contraseña no sea correcta, se destruyen las variables de sesión.

```
}
else{
{
    session_start();
    session_destroy();
    header("location:index.php");
}
```

?>

Figura 5.6: finalizar sesión

Después en todas las páginas donde se requiere que un usuario este registrado para ver algún contenido se añadirá en todas las páginas una función específica como esta:

```
!<?php
session_start();

if (isset ($_SESSION["autenticado"]) and $_SESSION["autenticado"]=="si" )
{
```

Figura 5.7: iniciar sesión

Dentro del “If” ira todo lo que pertenece al usuario con acceso a determinadas opciones.

Consultar datos.

Las consultas han sido algo muy frecuente en todo el portal ya sea para mostrar las noticias las sugerencias o cualquier otro servicio, un ejemplo de uso es el que se muestra a continuación.

```
{tpl = '<font color ="#000000" >
<!--/%f=fecha-->
<!--/%t=Titulo-->
<!--/%n=noticia-->
<!--/%a=nombre del periodista-->
<!--/%i=imagen-->
<!--/%s=seccion-->

<div style="width:550">
<h2 align="center" style="background:#FFFBF0"><h><t</h></h2>

<br>
<p style="text-align:justify; padding-right:3px; padding-left:96px; font-size: 16px;">
%n
<br>
</p>
<p style="padding-right:3px; font-size: 12px; text-align: right; font-weight: bold; font-style: italic;">
Noticia de %a de la seccion de %s, %f</p>

</div>
<hr size="1" align="right" width="100%" color="70809A">
<br>
</font>';

mysql_connect($config['host'], $config['user'], $config['pass']) or die('Connection failed');
mysql_select_db($config['db']) or die('DB selection failed');
$max = mysql_fetch_array(mysql_query("SELECT MAX(id) FROM noticias"));
$max = $max['MAX(id)']+1;

while($result = mysql_fetch_array(mysql_query('SELECT * FROM noticias WHERE id< ' . $max . ' ORDER BY id DESC;')))
{
    echo str_replace(Array('%t', '%a', '%n', '%s', '%i', '%f'), Array($result['titulo'], $result['nombre'], $result['mensaje'], $result['seccion'], $result[
'imagen'], $result['fecha']), $tpl);
    $max = $result['id'];
}
}
```

Figura 5.8: consultar datos

Se seleccionan todas las filas que cuadren con determinados requisitos según el sitio de la pagina donde nos encontremos, después mediante una variable definida como \$tpl donde se le a escrito en html donde va cada variable sacada de la base de datos.

Inserción de datos

En varios sitios se usan consultas de inserción, una de ellas es la usada en la de insertar noticias, otra es la de insertar sugerencias otra en hacer reservas de instalaciones y en añadir instalaciones. La que aquí comentare es la de añadir una noticia.

```
// name, message, date
$tit = $_POST['titulo'];
$message = $_POST['news'];
$author = $_POST['author'];
$date = strftime("%A %d de %B del %Y");
$seccion = $_POST['seccion'];
$image = $_POST['imagen'];
mysql_query('INSERT INTO noticias (nombre, titulo, mensaje, fecha, seccion, imagen)
VALUES (" . $author . "', " . $tit . "', " . $message . "', " . $date .
| " . $seccion . "', " . $image . "') or $error = true;
```

Figura 5.9: insertar datos

Como se puede ver en el código se recogen las variables enviadas por parte de un formulario que ha pasado las respectivas comprobaciones de errores, después con la función SQL insert se añadirá la fila en la base datos.

Actualizar datos.

En esta parte, la función mas representativa es la de modificar los datos personales del usuario, de esta forma podrá modificar su perfil, otro sitio donde se usa es en la modificación de noticias.

En el siguiente ejemplo se mostrara la actualización de los usuarios ya registrados.

```
$id=$_SESSION["id"];
$usuario = $_POST['usuario'];
$pass = $_POST['pass'];
$nombre = $_POST['nombre'];
$email = $_POST['email'];

mysql_query("UPDATE usuarios SET usuario = '$usuario' , pass='$pass', nombre='$nombre', email='$email'
| WHERE num = '$id'") or $error=true;
```

Figura 5.10: actualizar datos

Una vez más se ha hecho un tratamiento de errores de datos en el formulario que envía los datos y con la función SQL update actualizaremos los campos de usuario que cuadren con el id asignado en su registro.

Eliminar datos.

La eliminación solo la puede realizar en la parte de gestión, donde el administrador podrá borrar a los usuarios que lo deseen, borrar noticias, reservas, instalaciones sugerencias, etc.

El ejemplo que se muestra a continuación es el referente a la eliminación de una reserva realizada.

```
$bor=$_POST['del'];  
  
mysql_query("delete from reservar_instalaciones where id = '$bor' ");  
|  
?> <META HTTP-EQUIV="REFRESH" CONTENT="0";  
http://pion.disca.upv.es/ayto/Ayuntamiento_Silla/instalaciones/gestion_Reservar_instalaciones.php"></meta>
```

Figura 5.11: eliminar datos

La función recibe el id de reserva que ayudara en la eliminación. Una vez realizada, el usuario es redirigido a la página en la que estaba.

Visitas al portal.

Se lleva un pequeño control de visitas a la pagina, no esta totalmente depurado ya que no controla las direcciones ip para no repetir la misma visita si el usuario entra varias veces al dia, el numero de visitaas se actualiza en un fichero donde después se visualiza.

```
<?php  
$fichero = "numero.dat";  
$logout= 900;  
$count = 0;  
$res = '';  
$fp = fopen($fichero,"r");  
$t = time();  
$content = fread($fp,filesize($fichero));  
fclose($fp);  
$lineas = split(" ",$content);  
for ($i=0; $i < count($lineas); $i ) {  
$datos = split(':', $lineas[$i]);  
$n = $t - $logout;  
if ($datos[0] != $IP && $datos[1] > ($n)) {  
$res .= "$datos[0]:$datos[1] ";  
$count ;  
}  
}  
$res .= "$IP:$t ";  
$count ;  
$fp = fopen($fichero,"w");  
fwrite ($fp,$res);  
fclose($fp);  
echo "Hay $count visitantes activos";  
?>
```

Figura 5.12: código visitas

Realizar reservas.

Esta es una parte larga, se encuentra en el fichero “instalaciones/gestion_reservar_instalacion”. Básicamente al principio se realiza un tratamiento de fechas tanto en formato binario como en formato d/m/a. Se suma dos horas a la fecha de reserva, y se trata que no se solape la reserva con otras realizadas en esa misma instalación.

```
$fecha_reserva = $_POST['fecha_reserva'];  
list($dial,$mesl,$año,$horal)=explode('-', $fecha_reserva);  
list($ho,$min)=explode(':', $horal);  
$fecha_reserva_unix=mktime($ho,$min,0,$mesl,$dial,$año);  
  
$fecha_res=$año . '-' . $mesl . '-' . $dial;  
  
$fecha_ini=$dial.'-'. $mesl.'-'. $año.' ' . $horal;  
$h=$ho + 2;  
$fecha_fin=$dial.'-'. $mesl.'-'. $año.' ' . $h.':'. $min;  
  
$fecha_i=mktime($horal,$min,0,$mesl,$dial,$año);  
//al pasar la fecha a formato unix nos ayudara para comparar dos fechas a la hora  
$fecha_f=mktime($h,$min,0,$mesl,$dial,$año);  
$fecha_res=mktime(0,0,0,$mesl,$dial,$año);  
  
$link=Conectarse();  
if ($fecha_hoy_unix > $fecha_reserva_unix){
```

```
> <br><h3>La fecha ya ha pasado y no se puede reservar, por favor elige otra</h3> <?>
```

Figura 5.13: realizar reservar-1

Una vez tratadas las fechas se realizan varias consultas y comprobaciones para que no se solapen las reservas, y una vez todo sea correcto se inserta en la tabla relacional “reservar_instalaciones” la solicitud.

```
$nombre = $_SESSION["log"];
$id_ins= $_SESSION["reserva_num"];

$nom_ins=mysql_query("select * from instalacion where id='$id_ins'",$link);
while($datos=mysql_fetch_array($nom_ins))
{
    $nombre_inst=$datos['nom_instalacion'];
    $_SESSION["reserva_nom"] = $nombre_inst;
}

$rest = mysql_query("SELECT * FROM reservar_instalaciones WHERE id_instalacion='$id_ins' and fecha_i <= '$fecha_i' and fecha_f > '$fecha_i'",$link);
$sql=mysql_num_rows($rest);

$res = mysql_query("SELECT * FROM reservar_instalaciones WHERE id_instalacion='$id_ins' and fecha_i < '$fecha_f' and fecha_f > '$fecha_f'",$link);
$sql2=mysql_num_rows($res);

if($sql2>=1 or $sql>=1) {print "Hay reservas en ese intervalo por favor elige otra "; ?><br><? }

else{
    $result=mysql_query("INSERT INTO reservar_instalaciones (id_instalacion,nom_instalacion,
usuario,fecha_res,fecha_ini,fecha_fin,fecha_i,fecha_f)VALUES
('{$id_ins}','{$nombre_inst}','{$nombre}','{$fecha_res}','{$fecha_ini}','{$fecha_fin}','{$fecha_i}','{$fecha_f}')", $link);
}
```

Figura 5.14: realizar reservar-2

Otros

Hay otras pequeñas funciones para saber por el ejemplo, votar la sugerencia que es una consulta normal y actualizar el valor. Sacar la sugerencia mas votada o insertar una imagen en la base de datos en la parte de noticias.

5.3.3. Uso de código externo.

Se ha utilizado código externo bajo licencia libre GPL para ayudar a la implementaron en ciertas partes del código como en la reserva de instalaciones.

Es una aplicación hecha en php y javascript que contiene varias opciones y clases pero, solo la utilizaremos como entorno grafico para sacar de forma mas clara la fecha y hora de las reservas y de esta forma que el usuario lo encuentre mas intuitivo.

Reservar instalación para 2;

La instalación se reserva por un tiempo de 2 horas a partir de la hora indicada se ruega que sean horas exactas y sin minutos.

Fecha y hora de reserva:

sem	Lun	Mar	Mié	Jue	Vie	Sáb	Dom
44	1	2	3	4	5	6	7
45	8	9	10	11	12	13	14
46	15	16	17	18	19	20	21
47	22	23	24	25	26	27	28
48	29	30					

Hora: :

Seleccionar fecha

Figura 5.15: reservar instalación

Otro de los sitios donde se a utilizado código externo es en el inicio de la pagina. Donde se ha usado la pagina Web <http://jalbum.net/> para recoger el código de creación de un álbum digital en php, modificando el código para crear el aspecto que e deseado y añadido las imágenes representativas del pueblo



Figura 5.16: álbum

Además he añadido otro programa en php que accede a una pagina web que recoge el tiempo meteorológico en tiempo real. La página utilizada para tal fin es <http://www.meteored.com>.



Figura 5.17: tiempo

5.3.4. Implementación de la Interfaz gráfica.

Se ha implementado hojas de estilo en cascada, (CSS) para dar una apariencia similar en todos los apartados, este tipo de hoja se a utilizado en todos los apartados de forma dinámica haciendo referencia al archivo en concreto “style.css”, excepto en algunos donde se han implementado estáticamente el diseño grafico ya que se han modificado pequeñas cosas que no merecía la pena crearlas en el archivo general.

A continuación veremos una parte del archivo style.css:

```
body {
    font: small/1.6em verdana, sans-serif;
    margin: 0px;
    font: normal 10px verdana, arial, sans-serif;
    color: #000;
    background:url(../imagenes/t6_00005.jpg)
}
a {
    color: #888;
    text-decoration: none;
}
a:hover {
    color: #000;
    text-decoration: none;
}
a:active {
    color: #000;
}
input, textarea, select {
    border-top: 1px solid #858585;
    border-right: 1px solid #BEBEBE;
    border-bottom: 1px solid #DCDCDC;
    border-left: 1px solid #BEBEBE;
    background: #fff url(input_image.gif) repeat-x;
    font: normal 10px verdana, arial, sans-serif;
    color: #000;
    margin: 1px;
}
ul li {
    padding: 2px 0 2px 20px;
    line-height: 1.3em;
    color: #333333;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 120%;
    color: #7C7C7C;
}
ul {
    list-style: none;
    margin: 0px;
    padding-left: 0px;
    list-style-type: none;
}
```

Figura 5.18: CSS

Cada apartado del portal que utilice estas funciones como son: la lista, los enlaces los botones, los campos de texto, etc. se les aplicara automáticamente estas reglas graficas.

5.3.5. Implementación de la gestión de datos.

Para almacenar los datos de la aplicación he utilizado la una base de datos en MySQL donde se quedaran guardados permanentemente.

La herramienta utilizada es phpMyAdmin, creando una base de datos con el usuario “ayto” y contraseña “dacorcor” que a sido proporcionada por el tutor del proyecto, siendo este usuario el único que puede acceder a esta base de datos para realizar cualquier consulta.

Un usuario con mas permisos puede crear mas bases de datos forma:

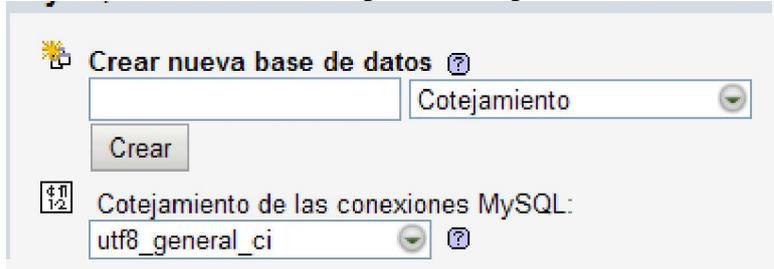


Figura 5.19: crear bd

en mi caso solo tenia una base de datos y se me permitía crear tantas tablas como hicieran falta.

Luego la forma de crear tablas se puede hacer de dos formas, añadiéndolas de forma grafica o por código SQL como el siguiente:

```
CREATE TABLE IF NOT EXISTS `instalacion` (
  `id` int(11) NOT NULL auto_increment,
  `nom_instalacion` text collate utf8_spanish_ci NOT NULL,
  `plazas` text collate utf8_spanish_ci NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci AUTO_INCREMENT=6 ;

CREATE TABLE IF NOT EXISTS `noticias` (
  `id` int(11) NOT NULL auto_increment,
  `nombre` text collate utf8_spanish_ci NOT NULL,
  `titulo` text collate utf8_spanish_ci NOT NULL,
  `mensaje` longtext collate utf8_spanish_ci NOT NULL,
  `fecha` text collate utf8_spanish_ci NOT NULL,
  `seccion` text collate utf8_spanish_ci NOT NULL,
  `imagen` text collate utf8_spanish_ci,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci AUTO_INCREMENT=10 ;
```

Figura 5.20: crear bd consultas

Así con todas las tablas, y una vez creadas todas la que necesitemos junto con las relaciones que estas tendrán quedaran de esta forma en phpMyAdmin:

<input type="checkbox"/>	instalacion							~5	InnoDB	utf8_spanish_ci	16.0 KB	-
<input type="checkbox"/>	noticias							8	MyISAM	utf8_spanish_ci	7.6 KB	-
<input type="checkbox"/>	reservar_instalaciones							~3	InnoDB	utf8_spanish_ci	32.0 KB	-
<input type="checkbox"/>	sugerencias							4	MyISAM	utf8_spanish_ci	2.7 KB	-
<input type="checkbox"/>	usuarios							3	MyISAM	utf8_spanish_ci	2.4 KB	60 Bytes
5 tabla(s)		Número de filas		~23	MyISAM	utf8_spanish_ci	60.8 KB	60 Bytes				

Figura 5.20: base de datos creada

6. Evaluación y pruebas.

Ya terminada la implementación de código y de la interfaz grafica, es el momento de evaluar y comprobar su funcionamiento, que sea el correcto y cumpla los estándares de la World Wide Web.

6.1. Pruebas

Se ha pasado por el depurador de Dreamweaver para encontrar posibles links rotos alguna etiqueta mal cerrada, etc, este depurador no detecta errores de programación ya que depende del propio programador.

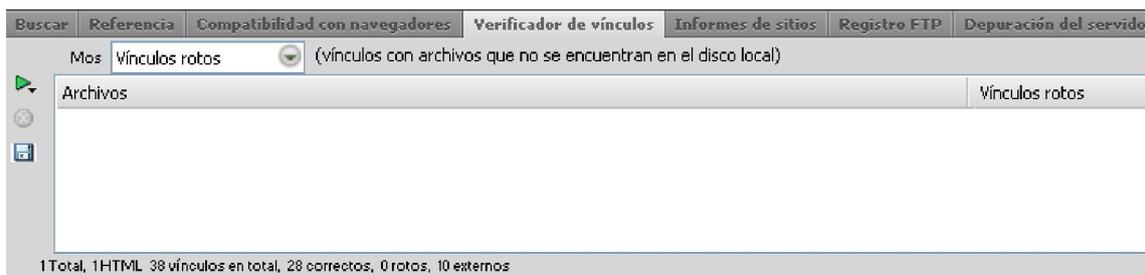


Figura 6.0: depurador Dreamweaver

En cuanto al modo de funcionalidad en diferentes navegadores Web, se ha centrado el portal en Firefox desde la 3.2 hasta su ultima versión 3.6, también se a probado en navegadores opera desde su versión 9, hasta la 10.10, chrome desde la versión 6 hasta la 7, Internet Explorer versiones 7 y 8.

Como apunte muestro una grafica del año 2010 en el que se puede observar el uso de los navegadores mas usados en Internet, notar que Firefox a pesar de su arranque mas lento consigue superar por primera vez el uso de Internet Explorer

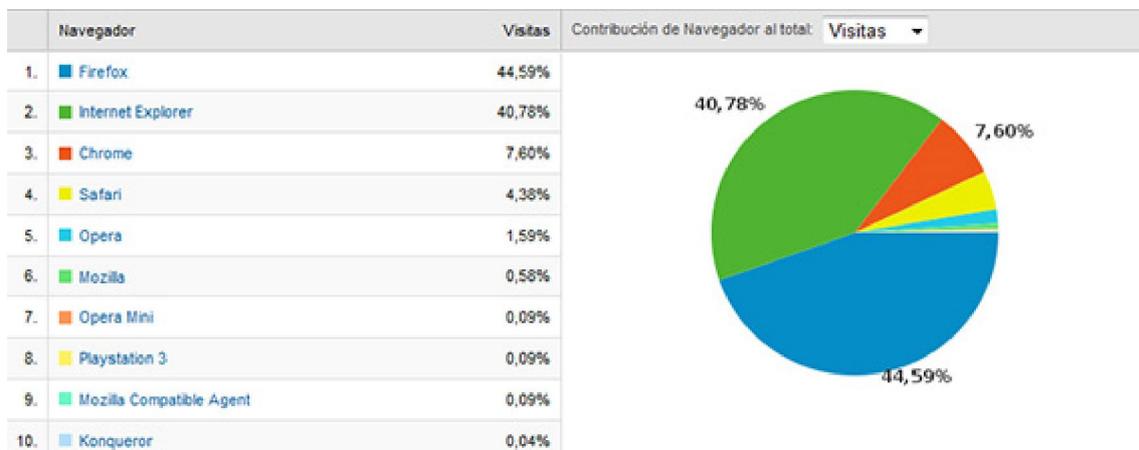


Figura 6.1: estadísticas navegadores

Otra forma de poder observar el tipo de visualización en los navegadores actuales es en la pagina Web www.browsershots.org, donde te devuelve en forma de capturas de imagen las diferentes formas en que los navegadores muestran en inicio de la pagina.

Funcionamiento en navegadores más comunes

A continuación se mostrara en los navegadores que he usado la forma de visualización, en la mayoría de ellos se ve de forma parecida y correcta, excepto en Internet Explorer 8 que he tenido un problema en una función java script referente a la fecha y hora en el inicio de la pagina que no se visualiza.

A continuación se muestran diferentes navegadores en los que la visualización es muy similar:

Opera V10



Figura 6.2: visión en el opera

Mozilla Firefox 3.6



Figura 6.3: visión en el Firefox

Internet Explorer 8



Figura 6.4: visión en el Internet Explorer

En este último explorador podemos observar la incompatibilidad de la función en java script de la fecha y hora, sin embargo en todos los demás exploradores funciona correctamente.

Comprobar enlaces.

Una vez finalizada la comprobación en los diferentes visualizadores es el momento de comprobar si todos los enlaces a los que accede la pagina son correctos, para ello nos

serviremos de la pagina Web <http://www.w3c.es/> que pertenece a la World Wide Web consortium.

En la siguiente imagen se muestran los resultados del escaneo de los enlaces.

```
Status: Done. Document processed in 30.72 seconds.

Checking link http://pion.disca.upv.es/ayto/Ayuntamiento_Silla/atcciudadano.html
HEAD http://pion.disca.upv.es/ayto/Ayuntamiento_Silla/atcciudadano.html  fetched in 0.14 seconds

Checking link http://pion.disca.upv.es/ayto/Ayuntamiento_Silla/Deportes.html
HEAD http://pion.disca.upv.es/ayto/Ayuntamiento_Silla/Deportes.html  fetched in 1.15 seconds

Checking link http://www.w3.org/TR/html4/strict.dtd
HEAD http://www.w3.org/TR/html4/strict.dtd  fetched in 0.01 seconds

Processed in 30.72 seconds.
```

Results

Links

Valid links!

Figura 6.5: validar links

Resoluciones.

En cuanto a las resoluciones, se ha usado una herramienta Web gratuita <http://viewlike.us/> donde se le puede indicar que tipo de resolución queremos probar. He probado varias resoluciones mas comunes de los usuarios siendo la mas optima a partir de 1024x768, en el supuesto de usar una resolución mayor no habría problemas pero en caso de usarla por de bajo aparecerían las flechas de desplazamiento

A continuación mostrare diferentes capturas con varias resoluciones.

Resolución 800x600



Figura 6.6: resolución 800x600

En esta resolución haría falta usar las barras de desplazamiento sería mas incomodo navegar por la pagina aunque esta resolución apenas se esta usando en la actualidad, ya que seria para monitores de 14 o 15 pulgadas.

Resolución 1024x768



Figura 6.7: resolución

Aquí ya se puede apreciar toda la pagina Web sin necesidad de barras de desplazamiento por tanto a partir de aquí se podría ver de forma optima.

Resolución 1440x900



Figura 6.8: resolución 1440x900

Esta es la resolución mas común en monitores actuales, a partir de un monitor de 17 pulgadas esta sería la resolución optima.

Validar HTML y CSS

Para este fin, he utilizado la pagina Web de la W3C (World Wide Web Consortium) para asegurarnos que la pagina pasa todos los estándares y no contiene ningún tipo de error sintáctico.

Para la validación HTML he usado la siguiente página.

<http://validator.w3.org/> obteniendo el siguiente resultado

This document was successfully checked as HTML 4.01 Transitional!	
Result:	Passed
Address:	<input type="text" value="http://pion.disca.upv.es/ayto/Ayuntamiento_Silla/"/>
Encoding:	utf-8 <input type="button" value="(detect automatically)"/>
Doctype:	HTML 4.01 Transitional <input type="button" value="(detect automatically)"/>
Root Element:	HTML

Figura 6.9: validar html y CSS

Al ser correcto nos permite añadir a nuestra pagina un código propuesto por ellos para demostrar que nuestro portal cumple todos los requisitos del HTML.

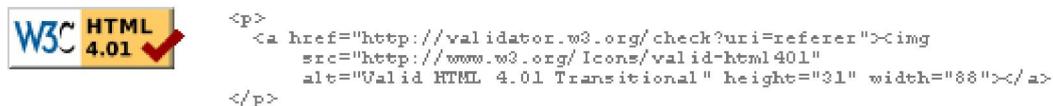


Figura 6.10: código validación html

De un modo parecido pasamos a validar el CSS con el siguiente enlace:

<http://jigsaw.w3.org/css-validator/> mostrando este resultado.

¡Este documento es [CSS versión 2.1](#) válido!

Puede mostrar este icono en cualquier página que valide para que los usuarios vean que se ha preo que puede usar para añadir el icono a su página Web:



Figura 6.11: código validación CSS

Este es el código que valida nuestra página, en donde hemos añadido en el inicio del portal.

6.2. Análisis de los resultados.

Después de todas las pruebas haremos un breve resumen de cada prueba obtenida.

Navegadores

Desde el principio se ha estado diseñando para el navegador Firefox, todas las pruebas y comprobaciones de toda la programación se a hecho desde este. Una vez finalizado comprobé que en todos los demás navegadores funcionaban de la misma forma exceptuando el Internet Explorer 8 teniendo problemas en ciertos Java scripts.

En un principio para creas las sesiones de los usuarios se utilizo las funciones cookie, (`$_COOKIE["mi galleta"]`), donde en los navegadores Firefox funcionaba de forma correcta, pero en el IE8 por algún motivo no iniciaba sesión de la misma forma.

Por eso se tomo la decisión de usar las funciones `$_SESSION["mí _ sesión"]` el cual funcionaba de mejor manera en todos los navegadores.

Otro de los aspectos era el tema del CSS, las visualizaciones no eran correctas en estos dos tipos de navegadores y se tuvo que redefinir varias clases del CSS añadiendo algún parámetro para que la visualización fuera correcta en los dos tipos de navegadores.

Comprobando enlaces.

Después de realizar varias pruebas en todo el código del portal se dejaron sueltos varios enlaces a paginas de prueba que había utilizado para tal fin, gracias e esta herramienta puede depurar y limpiar el código.

Resoluciones de pantalla.

En un principio no tuve en cuenta la resolución de pantalla usando la que tengo normalmente en mi pantalla 1440x900, el diseño estaba muy avanzado para volver a redefinirlo, pero tras las pruebas vi que desde una resolución de 1024x768 eran optimas y mostraban todo el contenido, siendo estas las mas usadas por los usuarios. Después de buscar encontré que para los atributos de los `<DIV>`, de las `<TABLE>` de las imágenes `` se podían usar en lugar de tamaños relativos, el uso de porcentajes de esta forma en cualquier navegador con cualquier resolución se vería de forma correcta. Sin ninguna duda en futuros proyectos será una de las cosas que tendré en cuenta.

Validación de HTML y CSS.

Aquí también se obtuvieron errores de validación, como el cierre de algunas etiquetas, la incompatibilidad de ciertas funciones que estaban deprecated (en fase de transición a la desaparición), en un principio se decidió por el html 4.01 strict pero a la vista de las incompatibilidades se decidió pasar al transaccional.

7. Conclusiones.

En cuanto a la conclusión del proyecto, a nivel personal son satisfactorias. Se ha invertido mucho tiempo e investigación para conseguir hacer ciertas cosas de la funcionalidad del portal, ha sido un esfuerzo el cual a merecido la pena, ya que he aprendido bastantes cosas de los lenguajes de programación que he utilizado aquí. Bien es cierto que actualmente podría realizar una página Web mas vistosa y con mas funcionalidades.

Sin embargo es un portal que perfectamente podría adaptarse a cualquier ayuntamiento cambiando pocas cosas o mejorando otras, e incluso sin cambiar nada de la programación y solo los datos.

7.1. Valoración personal

El proyecto consistía en realizar un portal Web desde cero, aplicando conocimientos que se han ido adquiriendo durante la carrera he innovar en algo que no se ha utilizado en esta como es el php o java script.

Como es lógico he encontrado ciertas dificultades en ciertos puntos más que en otros por ejemplo en reservar instalaciones que para mi a sido la mas complicada, sin embargo ahora podría mejorar el código. En cuanto al desarrollo grafico, el lenguaje html, ya contaba con ciertos conocimientos como en la asignatura de CDH el cual me ayudo.

En otras situaciones como la programación php o java script a sido de investigación propia a base de leer manuales por Internet, sin embargo estos lenguajes tienen cierta semejanza con java, c++ o c, siendo mas fácil su comprensión y puesta en funcionamiento.

En cuanto al los apartados teóricos como son los casos de uso, diagramas de secuencia, diagramas uml, o diseño de la base de datos, e tenido que hacer memoria y recuperar los apuntes de las no tan lejanas asignaturas de ingeniería del software (ISS) o Bases de datos (BDA).

También he aprendido a como estructurar un proyecto y dividirlo en partes y cumplir ciertas fechas, a pesar de las demoras de mi proyecto, soy consciente de la importancia que tiene el tiempo en cuanto a la entrega de un portal en una empresa.

7.2. Futuras ampliaciones o mejoras.

Finalizado el portal con todas sus opciones siempre es bueno notar que se puede mejorar, añadir o incluso cambiar para así conseguir que sea un portal dinámico y acorde con los tiempos que corren en Internet.

Estos son algunas opciones personales que se podría mejorar en este portal Web.

- Creación de un foro de discusión, donde los usuarios puedan crear canales y discutir sobre ellos, siempre que sean acorde con la política de la Web.
- Mejoras en el entorno gráfico de la página. Que sea más dinámica a cambios gráficos y a la inserción de publicidad por parte de agencias externas. Además de añadirle funciones en Flash para así sea más vistosa.
- Crear una búsqueda general, para así encontrar en el portal ciertas cosas que no veamos a simple vista, en caso de que este portal crezca mas, aquí no se ha considerado oportuno, ya que todos sus enlaces están a la vista y son fáciles para encontrarlos y acceder.
- Creación de un canal RSS de noticias el cual si se ha investigado en este portal pero no se ha podido acoplar debido al diseño de la pagina y a la difícil implementación en el diseño grafico.
- Otros. Portal abierto a ideas de otras personas.

8. Bibliografía

8.1. Libros consultados.

- Una guía para la realización y supervisión de proyectos web (PFC)
Buendía García, Félix
Editorial UPV
- Creación de un portal con PHP y MySQL
Pavón Puertas, Jacobo
Editorial Ra-ma
- CSS, hojas de estilo en cascada para el diseño Web
Christopher Schmitt
Editorial Anaya Multimedia

8.2. Páginas Web visitadas

PHP:

<http://www.manualdephp.com/>

<http://www.php-es.com/>

HTML:

<http://www.desarrolloweb.com/htm>

<http://www.webestilo.com/html/>

CSS:

<http://www.desarrolloweb.com/css>

Otros

<http://jalbum.net/>

<http://www.webtaller.com/tallerscripts/scripts/34/>

Asignatura CDS

http://personales.upv.es/jagil/CDH/Transpas/tema1_html_archivos/frame.htm

http://personales.upv.es/jagil/CDH/Transpas/tema2_css_archivos/frame.htm

http://personales.upv.es/jagil/CDH/Transpas/tema3_javascript_archivos/frame.htm

9. Anexos

9.1. Anexo A. diseño lógico de la base de datos

- **Estructura de tabla para la tabla `instalacion`**

```
CREATE TABLE IF NOT EXISTS `instalacion` (
  `id` int(11) NOT NULL auto_increment,
  `nom_instalacion` text collate utf8_spanish_ci NOT NULL,
  `plazas` text collate utf8_spanish_ci NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
COLLATE=utf8_spanish_ci AUTO_INCREMENT=6 ;
```

- **Estructura de tabla para la tabla `noticias`**

```
CREATE TABLE IF NOT EXISTS `noticias` (
  `id` int(11) NOT NULL auto_increment,
  `nombre` text collate utf8_spanish_ci NOT NULL,
  `titulo` text collate utf8_spanish_ci NOT NULL,
  `mensaje` longtext collate utf8_spanish_ci NOT NULL,
  `fecha` text collate utf8_spanish_ci NOT NULL,
  `seccion` text collate utf8_spanish_ci NOT NULL,
  `imagen` text collate utf8_spanish_ci,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8
COLLATE=utf8_spanish_ci AUTO_INCREMENT=10 ;
```

- **Estructura de tabla para la tabla `reservar_instalaciones`**

```
CREATE TABLE IF NOT EXISTS `reservar_instalaciones` (
  `id` int(100) NOT NULL auto_increment,
  `id_instalacion` int(100) NOT NULL,
  `nom_instalacion` text collate utf8_spanish_ci NOT NULL,
  `usuario` text collate utf8_spanish_ci NOT NULL,
  `fecha_res` text collate utf8_spanish_ci NOT NULL,
  `fecha_ini` text collate utf8_spanish_ci NOT NULL,
  `fecha_fin` text collate utf8_spanish_ci NOT NULL,
  `fecha_i` double NOT NULL,
  `fecha_f` double NOT NULL,
  PRIMARY KEY (`id`),
  KEY `id_instalacion` (`id_instalacion`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
COLLATE=utf8_spanish_ci AUTO_INCREMENT=20 ;
```

- **Estructura de tabla para la tabla `sugerencias`**

```
CREATE TABLE IF NOT EXISTS `sugerencias` (
  `id` int(11) NOT NULL auto_increment,
  `nick` text collate utf8_spanish_ci NOT NULL,
  `tel_mail` text collate utf8_spanish_ci NOT NULL,
  `direccion` text collate utf8_spanish_ci,
  `comentario` longtext collate utf8_spanish_ci NOT NULL,
  `date` text collate utf8_spanish_ci NOT NULL,
  `votos` int(11) NOT NULL default '0',
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8
COLLATE=utf8_spanish_ci AUTO_INCREMENT=6 ;
```

- **Estructura de tabla para la tabla `usuarios`**

```
CREATE TABLE IF NOT EXISTS `usuarios` (
  `num` int(11) NOT NULL auto_increment,
  `usuario` text collate utf8_spanish_ci NOT NULL,
  `pass` text collate utf8_spanish_ci NOT NULL,
  `nombre` text collate utf8_spanish_ci NOT NULL,
  `email` text collate utf8_spanish_ci,
  `date` text collate utf8_spanish_ci NOT NULL,
  PRIMARY KEY (`num`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8
COLLATE=utf8_spanish_ci AUTO_INCREMENT=11 ;
```

- **Relaciones para la tabla `reservar_instalaciones`**

```
ALTER TABLE `reservar_instalaciones`
ADD CONSTRAINT `reservar_instalaciones_ibfk_1` FOREIGN KEY
(`id_instalacion`) REFERENCES `instalacion` (`id`);
```

9.2. Toma de datos

Cuestionario sobre una entrevista hipotética a la persona que nos encargaría el proyecto del ayuntamiento de silla.

Datos del usuario: Nombre y apellidos, cargo en el ayuntamiento. Edad

Conocimientos: ¿tipo de explorador que usa Web?, ¿para que usa el ordenador?, ¿quien actualizara los apartados de noticias del pueblo?, ¿Qué tipo de conocimientos tiene el que modifica?, ¿a que le gustaría que estuviera orientada mas el portal? etc.

Intereses: obtener información histórica del pueblo para su introducción.

Organización del semicírculo de entidades políticas, y sus datos.

Fiestas patronales, culturas.

- ¿Le interesa algún foro público y/o privado?
- ¿Esta dispuesto a colaborar en el desarrollo del proyecto Web?
- ¿Quiere alguna aplicación específica que quiera que realice la página Web?

Planificación de un seminario sobre el proyecto a desarrollar.
Definición de objetivos del seminario.

Planificación del problema a resolver: ¿Cuál será la finalidad del portal de ayuntamiento?, ¿Qué servicios ofrecen? (gestión interna para trabajadores, y externa para ciudadanos). Tipo de ciudadano al que ira mas indicado, y trabajadores los cuales tendrán acceso.

Selección del grupo de trabajo: ira a cargo del jefe de proyectos el decidir quien se encargara de la aportación al portal.

Elaboración de una guía de debate: que personas se encargaran de realizar las tareas, y el tiempo que se dedicara para desarrollar cada apartado.
Desarrollo del debate: opiniones de los desarrolladores, preguntas que se han podido responder y sus respuestas.

Análisis de resultados: Aquí se verán los posibles problemas que puede ocasionar la aplicación, la valoración de cada desarrollador y las necesidades mínimas de la aplicación.

Boceto de escenario de la aplicación del ayuntamiento.

Acceso: como ciudadano registrado, ciudadano sin registrar o como trabajador.
Menú principal con opciones del portal (localización, historia ciudad, pago de recibos, miembros del ayuntamiento, etc.)
Introducción de datos o visualización de contenidos: en caso de que sea de pagar recibos u otros movimientos que requieran ser usuario registrado, y/o visualización de contenidos ofrecidos.
Envío de datos y cierre aplicación.