



# Linkia FP

Formación Profesional Oficial a Distancia

---

## Desarrollo de aplicaciones web

---

*M12 – Proyecto*

*Memoria del proyecto*

**Jon Ander Oribe Sánchez**  
**Daniel Fernández García**

# Índice

---

<b>Índice</b>	<b>2</b>
<b>1. Introducción</b>	<b>3</b>
Objetivo	3
Motivación	3
Estructura	5
<b>2. Desarrollo</b>	<b>6</b>
Tecnologías	6
Herramientas	7
Tipos de usuario	8
Funcionalidades	9
Interfaz	11
<b>3. Implementación</b>	<b>15</b>
Base de Datos	15
Funcionalidades	16
Interfaz	25
Servidor	32
<b>4. Conclusiones</b>	<b>34</b>
Valoración del proyecto	34
Posibles ampliaciones	35
<b>5. Bibliografía</b>	<b>36</b>
<b>6. Anexos</b>	<b>37</b>
Base de Datos	37

# 1. Introducción

---

## Objetivo

En el presente proyecto, nos hemos inclinado por desarrollar una aplicación de coordinación de voluntariado para PRIDE Barcelona, organización que cuenta con un nutrido grupo de voluntarios. Si bien el grueso de colaboradores lo hace puntualmente durante la celebración estival del orgullo LGTBI+, existe un grupo más reducido que colabora en acciones durante todo el año. De este modo, el equipo de voluntarios de PRIDE colabora con otras entidades de Barcelona como apoyo en la realización de sus actividades: desde la organización de conferencias, mesas redondas o cinefòrums, hasta la celebración de fiestas de barrio, la participación en actos en pro de colectivos invisibilizados (especialmente el trans\* y el de afectados por el VIH).

La aplicación permitirá que, cada vez que la organización participe en un evento, los coordinadores puedan publicar la actividad, al tiempo que los voluntarios puedan visualizar cuáles son los eventos venideros y apuntarse a los mismos.

## Motivación

Los dos integrantes del equipo hemos ejercido como voluntarios en el llamado Tercer Sector. Si bien el tipo de organizaciones con las que hemos colaborado son diferentes, todas ellas comparten una problemática común: la dificultad a la hora de coordinar a un equipo de voluntarios. Dicho equipo es a la organización sin ánimo de lucro lo que una plantilla de trabajadores a una empresa, con la particularidad de no recibir una contraprestación económica a cambio de su trabajo.

Por este motivo, resulta primordial simplificar todas aquellas tareas organizativas inherentes a todo tipo de trabajo en equipo susceptible de necesitar cierto nivel de planificación y organización. Dicho de otro modo: si un voluntario o coordinador dedica demasiado tiempo a cuestiones de tipo administrativo (elaboración de horarios, planificación de equipos, comunicaciones, etc.), existe el peligro de que, con el tiempo, se replantee su rol en la organización y acabe desistiendo de colaborar.

Este problema se da, particularmente, entre los coordinadores de voluntarios, muchos de los cuales acaban por sufrir una suerte de *Síndrome del Coordinador Quemado*. A menudo, lograr un equipo de voluntarios para un evento concreto requiere de tantas llamadas, emails y mensajes que, para cuando llega la celebración del mismo, el coordinador se encuentra crispado. Simplificar la manera en que se publican los eventos y se forman los equipos, evitando dirigirse a cada voluntario individualmente, aminoraría el estrés de los coordinadores.

Del mismo modo, a muchos voluntarios les cuesta enterarse de los eventos. El hecho de que en algunas ocasiones se cree un grupo de Whatsapp para informarles, en otras se opte por un evento en Facebook, o en otras se recurre al envío de emails, contribuye a aumentar la confusión.

En resumen, la organización de voluntarios para formar equipos para participar en actividades/acciones requiere de una herramienta sencilla, clara e intuitiva a través de la cual los voluntarios puedan ver los eventos publicados y apuntarse a ellos. Al mismo tiempo, los coordinadores han de ser capaces de saber quiénes participaran en cada una de estas acciones con antelación suficiente para saber con qué recursos cuentan y sin necesidad de acabar llamando voluntario a voluntario porque la mayor parte de éstos no hayan respondido a la convocatoria enviada/publicada.

Existen algunas aplicaciones enfocadas a este fin, con el inconveniente de que suelen ser de pago, con el coste de oportunidad que conlleva una subscripción: gastar dinero en ello supone dejar de gastarlo en las acciones que definen la actividad de la entidad. Asimismo, estas aplicaciones disponen de un sinfín de funcionalidades que acaban por generar ruido visual, desincentivando al usuario a utilizar el software.

## Estructura

Esta memoria está estructurada según el orden de desarrollo que ha seguido el proyecto. Inicialmente, se han planteado 4 grandes secciones, abarcando cada una de ellas una de las etapas principales del proyecto. Cada sección contiene, a su vez, diversos subepígrafes que segmentan y estructuran la información alrededor de los principales puntos sobre los que se ha apoyado la implementación de cada sección. Asimismo, se han añadido los epígrafes *Bibliografía/Webgrafía y Anexos*.

Las secciones son:

**Introducción** -> en ella se expone cuál es el objetivo que el proyecto pretende alcanzar, así como los motivos que han llevado al equipo a plantear el desarrollo de la aplicación.

**Desarrollo** -> se detallan las tecnologías que se utilizarán para el desarrollo del proyecto, así como las herramientas de la que se servirá el equipo. En esta etapa, se plantea cómo será la aplicación en base a las necesidades del proyecto y atendiendo a la tipología de usuarios de la futura aplicación, así como a las funcionalidades que estos requieren ejecutar.

**Implementación** -> contiene la información relativa a cómo se ha plasmado el trabajo de la etapa de desarrollo. Se describe el código empleado para implementar las funcionalidades requeridas, así como para dar a la interfaz la forma deseada.

**Conclusiones** -> una vez finalizado el proyecto, esta sección recoge las principales conclusiones del desarrollo del mismo, principalmente a modo de evaluación. Además, muestra cuáles serían las líneas a seguir en caso de que el cliente de la aplicación requiriera de nuevas funcionalidades.

**Bibliografía** -> relación de la documentación consultada durante la elaboración del proyecto.

**Anexos** -> información relacionada.

## 2. Desarrollo

---

### Tecnologías

Dada la orientación de este proyecto al sector del desarrollo web se ha previsto el uso de diferentes herramientas y lenguajes de programación ampliamente probados en otras aplicaciones de similares características.

Empezaremos dividiendo las tecnologías a utilizar en la parte cliente y la que será desarrollada en el servidor. Por su parte, el Front-End de la aplicación hará uso de:

- HTML y CSS, para la generación del aspecto visual de la web.
- Javascript y JQuery, de cara al desarrollo de la lógica de negocio.
- Leaflet como librería de apoyo para la integración de mapas que reflejen la localización de los diferentes eventos administrados desde la web.

En cuanto a la configuración de los servicios Back-End, se trabajara con:

- MySQL como base de datos relacional orientada a la gestión y persistencia de toda la información que circule por la web.
- PHP como lenguaje de comunicación entre la parte cliente y la servidor.

Por su parte, el desarrollo del proyecto hará uso de la herramienta de gestión de versiones conocida como GitHub, a través de la cual se propone el trabajo en dos ramas, una Master y una Develop, a fin de permitir un trabajo en paralelo entre ambos miembros del equipo con la seguridad de no estar replicando código o sobreescribiendo, por accidente, versiones funcionales por otras con errores.

Se plantea también la integración a futuro, en caso de que el tiempo lo permita, de un gestor de eventos con interfaz gráfica, el cual sería desarrollado mediante el uso de la librería Vis.js y la monitorización de la calidad del código mediante otros servicios web, como los ofrecidos por plataformas similares a Codecov, con vistas a generar una aplicación lo más óptima y escalable posible.

## Herramientas

**Github** -> Esta plataforma diseñada para el desarrollo colaborativo ha sido fundamental para que cada miembro del equipo pudiera trabajar en el desarrollo del proyecto manteniendo un correcto control de versiones y permitiéndoles trabajar en nuevas funcionalidades sin necesidad de modificar la versión funcional del proyecto.

**Github Desk** -> Aplicación que permita una sencilla conexión con Github, facilitando la subida de cambios al repositorio general, así como la descarga de nuevas actualizaciones.

**Visual Studio Code** -> Editor de texto/IDE utilizado para elaborar el código (tanto el html/css, como JS y PHP). Además de ser de código abierto, resultaba especialmente útil para el proyecto por su compatibilidad con prácticamente cualquier lenguaje.

**XAMPP** -> Antes de subir el proyecto a un servidor, el uso de XAMPP ha permitido al equipo poder ejecutar la aplicación web en local.

**phpMyAdmin** -> esta herramienta ha permitido el manejo y la administración de la base de datos.

**MysqlWorkbench** -> también hemos utilizado esta herramienta (conectada con nuestro servidor local) para la creación de la base de datos.

**MockFlow.com** -> Aplicación para la creación de WireFrames.

**Draw.io (diagramas)** -> Aplicación para la creación de diagramas. Utilizada para el diseño de la base de datos.

## Tipos de usuario

La aplicación dispondrá de diferentes funcionalidades en función de si el usuario es coordinador o bien voluntario.

Se nos pide que un usuario de tipo coordinador pueda:

- Acceder a un listado de eventos
- Acceder a un listado de voluntarios
- Acceder a un listado de incidencias
- Gestionar la base de datos de voluntarios (incluyendo altas, bajas, modificaciones de datos).
- Publicar y editar actividades/eventos.
- Obtener informes de eventos.
- Obtener informes de voluntarios.
- Obtener informes de incidencias.

A su vez, un usuario tipo voluntario ha de ser capaz de:

- Acceder a un listado de eventos disponibles.
- Acceder a un listado de eventos en los que está apuntado.
- Apuntarse a un evento.
- Enviar una incidencia
- Obtener su informe de voluntario.



## Funcionalidades

### Pantalla login

- Login - acceso mediante nombre de usuario y contraseña.
- Recordar login - recordar usuario y password para próximas sesiones.
- Olvidé mi contraseña - enviar email para restablecer contraseña.
- Nombre de usuario o Password erróneo - advertir de error en los datos introducidos

## COORDINADOR

### Home Coordinador

- Ver listado de eventos
- Buscar evento por nombre
- Ver listado de voluntarios
- Buscar voluntario por nombre
- Ver listado de incidencias
- Acceso a detalles de evento
- Acceso a ficha de voluntario
- Publicación nuevo evento (formulario datos evento)
- Alta Voluntario (formulario datos voluntario)
- Obtener informe de evento
- Obtener informe de voluntario
- Obtener informe de Incidencias
- Mostrar/ocultar menú
- Log out

### Ficha Datos del Voluntario

- Editar datos del Voluntario (acceso a todos los datos).
- Dar de Baja Voluntario (cambiar status del mismo a inactivo).
- Obtener informe Voluntario

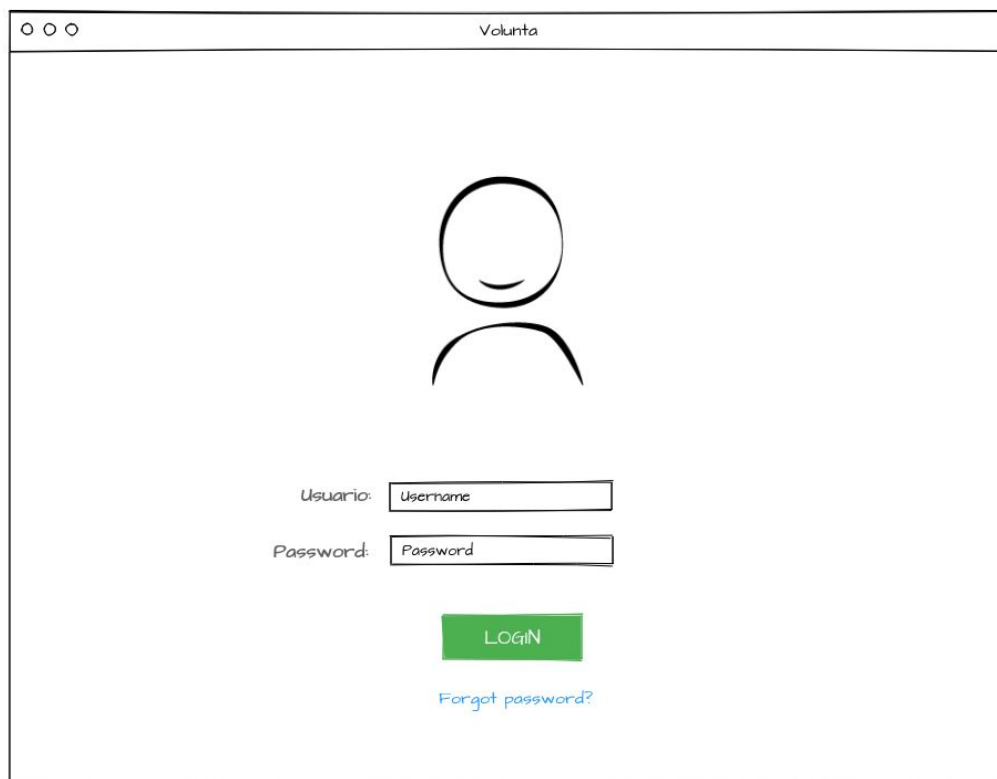
**Home Voluntario**

- Ver ficha perfil personal
- Obtener informe voluntario
- Ver listado de eventos disponibles
- Ver listado Mis Eventos (aquellos a los que el voluntario está apuntado).
- Buscar evento por nombre
- Acceso a detalles de evento (pop up)
- Apuntarse al Evento
- Enviar incidencia
- Mostrar/ocultar menú
- Log out

## Interfaz

En la fase de desarrollo se realizó una primera aproximación a cómo debería visualizarse la interfaz para que permitiera una interacción efectiva por parte del usuario. En los Wireframes que pueden observarse a continuación, aparecen los principales elementos de dicho diseño. Sin embargo, cabe destacar que, durante la fase de implementación, el equipo ha ido realizando pequeñas modificaciones sobre el diseño original, siempre con el fin de favorecer la navegabilidad a través de la aplicación. Estos cambios pueden apreciarse en la tercera sección de esta memoria, dedicada a la implementación, que contiene una subsección dedicada a la interfaz.

### Login Page



The wireframe shows a login page within a browser window. The browser's title bar contains three window control buttons (minimize, maximize, close) and the text 'Volunta'. The main content area of the browser displays a login form. At the top center of the form is a simple line-art icon of a person's head and shoulders. Below this icon are two input fields: the first is labeled 'Usuario:' and contains the placeholder text 'Username'; the second is labeled 'Password:' and contains the placeholder text 'Password'. Below the password field is a green rectangular button with the text 'LOGIN' in white. At the bottom of the form is a blue hyperlink that reads 'Forgot password?'.

## Voluntario View

**Home Voluntario**

Logo

Mostrar/Ocultar Menú

NUM VOLUNTARIO: 55555555 [Edit](#)

NOM: [Redacted]  
 COGNOM: [Redacted]  
 DNI: 44445555L  
 TELF: [Redacted]  
 EMAIL: [Redacted]  
 STATUS: Active  
 Horas de Voluntariado: 15

[Cambiar](#)

Menú

- Mi perfil
- Eventos
- Mis Eventos
- Informe
- Incidencias

[Logout](#)

Q. Búsqueda evento

Título Evento	Organizador	Tipo de Evento	Participantes	Disponibilidad	Más Información
B					<a href="#">Info</a>
B					<a href="#">Info</a>
B					<a href="#">Info</a>

Evento - Información

NOM: [Redacted]  
 GUAN: [Redacted]  
 LOCALIZACIÓ: [Redacted]  
 COORDINADOR: [Redacted]  
 STATUS: Active  
 DESCRIPCIÓ: [Redacted]

[Apuntarse/Desapuntarse](#) [Cerrar](#)

**Voluntario - Listado Eventos**

Logo

Mostrar/Ocultar Menú

Q. Búsqueda evento

Título Evento	Organizador	Tipo de Evento	Participantes	Disponibilidad	Más Información
B					<a href="#">Info</a>
B					<a href="#">Info</a>
B					<a href="#">Info</a>

**Voluntario - Listado de mis Eventos**

Logo

Mostrar/Ocultar Menú

Q. Búsqueda evento

Título Evento	Organizador	Tipo de Evento	Participantes	Disponibilidad	Más Información
B					<a href="#">Info</a>
B					<a href="#">Info</a>
B					<a href="#">Info</a>

Descarga Informe del Voluntario en el que contendrá sus datos personales, el historial de eventos en los que ha participado y el número de horas dedicadas.

Panel

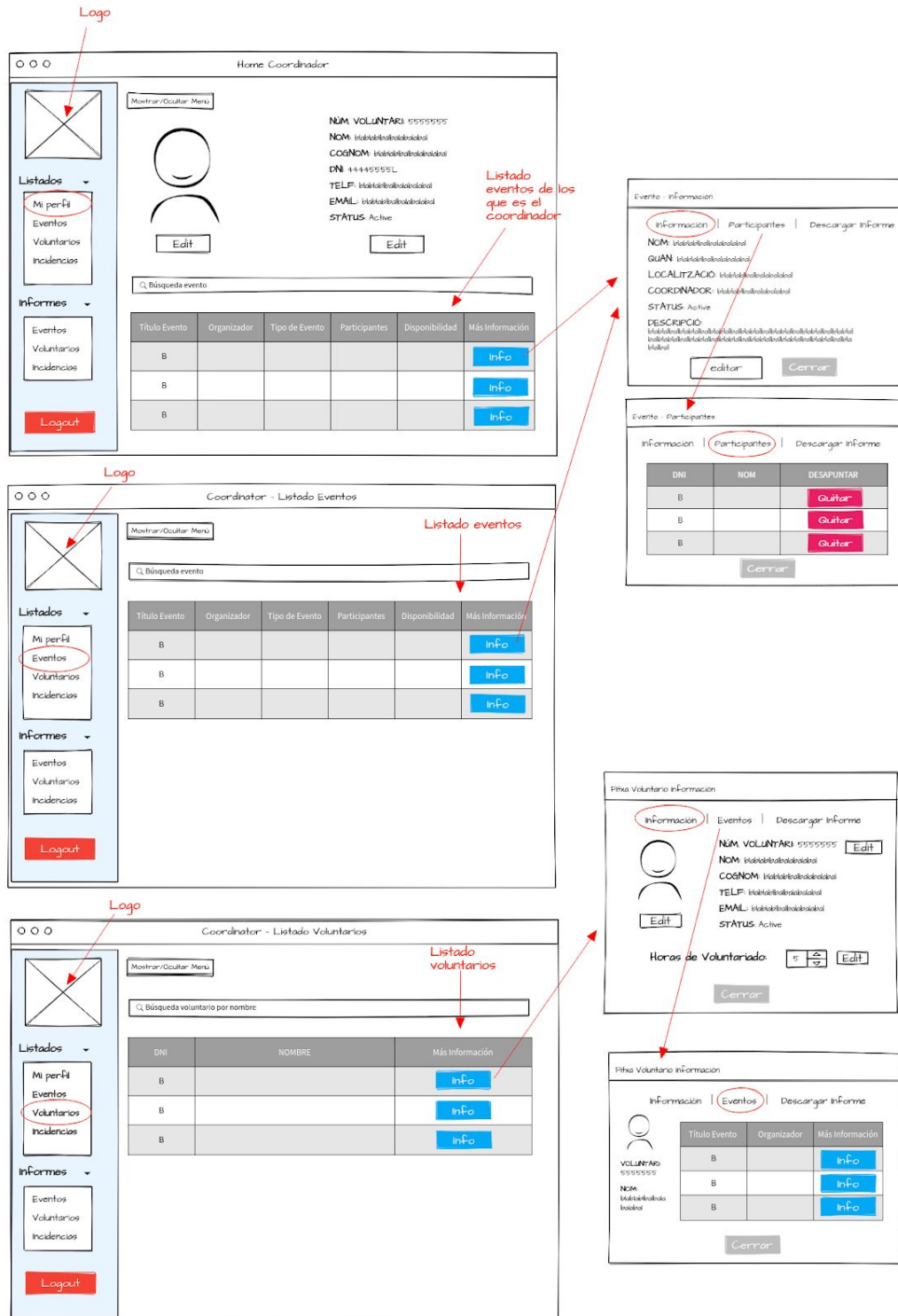
Seleccionar el evento

Describe aquí la incidencia.

[Enviar](#) [Cerrar](#)

Para generar una incidencia (se envía al coordinador y se publica en el listado de incidencias, al que solo tienen acceso los coordinadores).

## Coordinador View



## Coordinador View

Coordinator - Listado Incidencias

Logo

Mostrar/Ocultar Menu

Búsqueda voluntario por evento

Listado voluntarios

Listados

- Mi perfil
- Eventos
- Voluntarios
- Incidencias**

Informes

- Eventos
- Voluntarios
- Incidencias

Logout

FECHA	EVENTO	Informador	Descripción
B			Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien du mattis. Duis non pulvinar lorem. Felle nec erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien du mattis. Duis non pulvinar lorem. Felle nec erat.
B			Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien du mattis. Duis non pulvinar lorem. Felle nec erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien du mattis. Duis non pulvinar lorem. Felle nec erat.
B			Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien du mattis. Duis non pulvinar lorem. Felle nec erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien du mattis. Duis non pulvinar lorem. Felle nec erat.

Coordinator - Report Eventos

Seleccione los campos del informe

Tipo de Evento  2 May 2016

Generar Informe

Coordinator - Report Voluntario

Seleccione los campos del informe

Seleccionar Voluntario  2 May 2016

Generar Informe

Coordinator - Report Incidencias

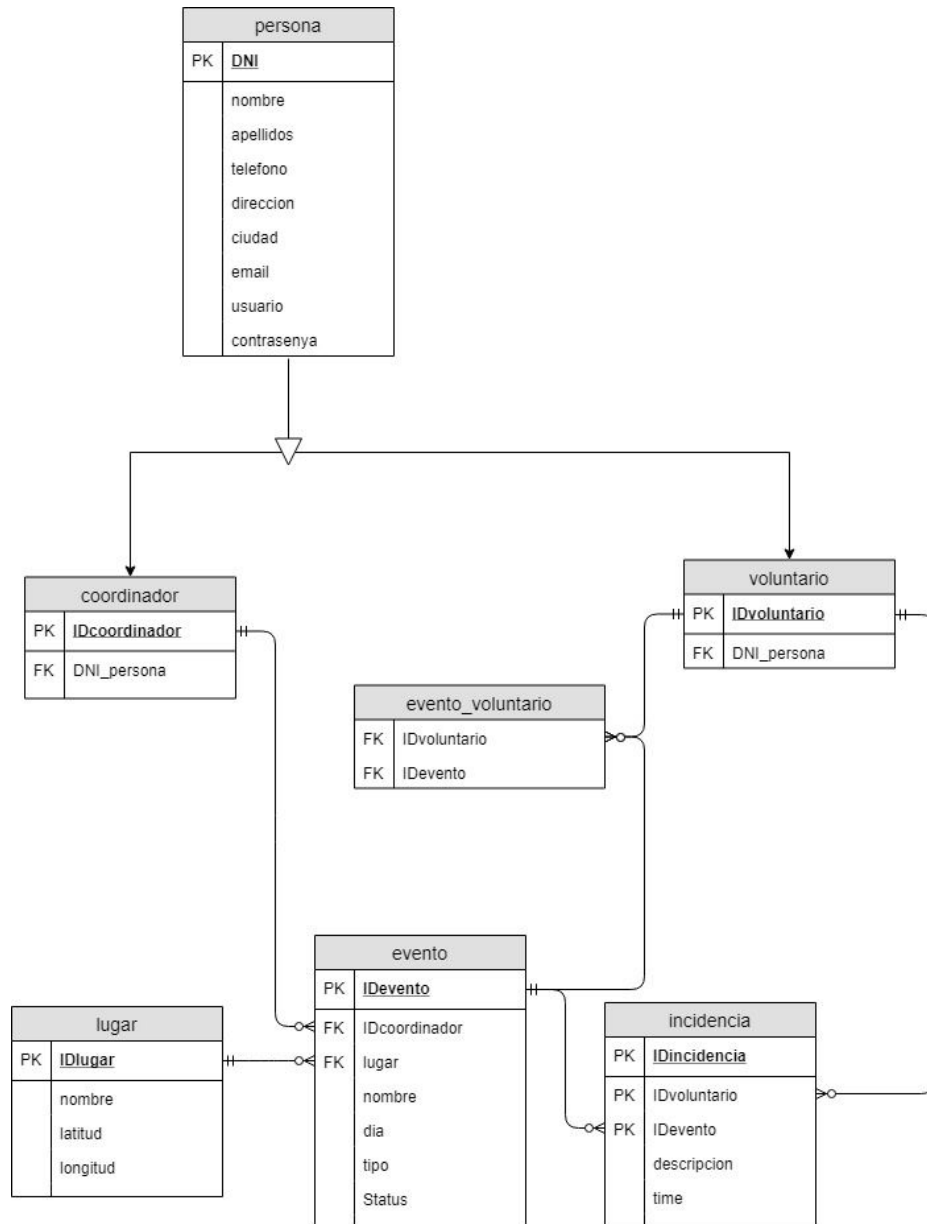
Seleccione los campos del informe

Seleccionar Evento  2 May 2016

Generar Informe

## 3. Implementación

### Base de Datos



NOTA: al diseñar la base de datos, nos hemos encontrado con que la relación entre evento y voluntario era de muchos a muchos. Para facilitar la relación entre éstas, hemos creado la tabla evento\_voluntario.

## Funcionalidades

### Login

Cuando el usuario desea hacer uso de la aplicación se dirige a index.php para poder insertar sus credenciales. Al introducir sus datos en el formulario, éstos se enviarán al archivo login.php. Este archivo comprueba, en primer lugar, que el usuario haya introducido los datos solicitados, en caso de no ser así, devuelve al usuario a la misma página y le indica que debe rellenar los campos solicitados:

```
if(empty($_POST['uname']) || empty($_POST['psw'])){  
    $_SESSION['error_login'] = "Debes introducir nombre de usuario y contraseña";  
    header("Location: index.php");  
}
```

Si el usuario ha introducido los datos, comprobaremos que estos coincidan con los de alguno de los usuarios incluidos en la base de datos. Para poder comprobar si existe, se llama a una función creada en database.php. Dicha función buscará al usuario (y su contraseña) en la base de datos y devolverá el resultado:

```
function login($con, $usuario, $contrasena){  
    $result = mysqli_query($con, "select * from persona where usuario='".$usuario.'" and contrasena='".$contrasena.'";  
    if(mysqli_num_rows($result)==0){  
        return 0; //Si no existe el usuario devuelvo 0  
    }  
    else{  
        $usuario = mysqli_fetch_array($result);  
        return $usuario; //Si existe el usuario devuelvo un array con sus datos  
    }  
}
```

En caso de que devuelva cero (ergo, de que no existan usuarios con las credenciales introducidas), devolverá al usuario a la página principal para que vuelva a intentar el login.

```
$resultado = login($con, $usuario, $contrasena);  
if($resultado == 0){  
    $_SESSION['error_login'] = "Datos incorrectos";  
    header("Location: index.php");  
}
```



Si, por el contrario, arroja un resultado (si usuario y password existen), entonces creará unas variables de sesión (que resultarán muy útiles de cara a otras funcionalidades).

```
else{  
    $_SESSION['usuario'] = $resultado['usuario'];  
    $_SESSION['contrasena'] = $resultado['contrasena'];  
    $_SESSION['dni'] = $resultado['dni'];  
}
```

Finalmente, se recurrirá a una nueva función para averiguar en qué tabla se encuentra el usuario (coordinador o voluntario). Dependiendo del resultado de esta consulta, se enviará al usuario a una vista u otra.

```
$result2 = mysqli_query($con, "select * from coordinador where persona='".$_SESSION['dni']."'");  
if(mysqli_num_rows($result2)==0){  
    header("Location: vistaPrincipalVoluntario.php");  
}  
else{  
    header("Location: listadoEventos.php");  
}
```

También se han creado para el control de sesión, de manera que antes de abrirse una página concreta, se comprueba que el usuario que va a acceder a ella es coordinador o voluntario (según proceda).

Por ejemplo, para comprobar que quien accede a una página es un administrador, se ha creado en database.php la función controlAdmin. Esta función comprueba que el usuario *loggeado* (recordemos que las credenciales se han almacenado en variables de sesión) esté en la lista de coordinadores:

```
// control admin //  
function controlAdmin($con, $dni){  
    $result = mysqli_query($con, "select * from coordinador where persona='".$dni."'");  
    while($fila = mysqli_fetch_array($result)){  
        $coordinadores[] = $fila;  
    }  
    return $coordinadores;  
}
```

En caso de no contar con la variable de sesión o de que el usuario no se encuentre en dicha tabla, se devolverá al usuario a la página de login.

```
?php
session_start();
require_once("database.php");
$dni=$_SESSION['dni'];

$coordinadores = controlAdmin($con, $dni);
if(!isset($_SESSION['usuario'])){
    header("Location: index.php");
}

function controlSesionAdmin($coordinadores){

    if(count($coordinadores) == 0){
        header("Location: index.php");
    }
}
```

## Vista Voluntario

La vista de voluntario incluye básicamente la posibilidad de ver los detalles de usuario, el listado de eventos, así como el listado de eventos (más adelante veremos cómo se han desarrollado las funcionalidades para generar incidencias y/o apuntarse a un evento que también pueden ejecutarse desde esta vista.

Asimismo, para que el usuario pueda ver los eventos disponibles, es necesario que el recorrer la tabla de eventos, e imprimir los resultados de dicha lectura por pantalla, creando la lista de eventos. Para ello, existe una función en database.php que permite consultar los eventos que contiene la tabla:

```
//FUNCIÓN PARA LISTAR EVENTOS
function listarEventos($con){

    $result = mysqli_query($con, "select e.idevento as id, e.nombre as nombre,
    l.nombre as lugar, p.nombre as coordinador from lugar l, persona p, coordinador c,
    evento e where c.persona = p.dni and e.coordinador = c.idcoordinador and e.lugar = l.idlugar;");

    $coordinadores = array();
    while($fila = mysqli_fetch_array($result)){
        $eventos[] = $fila;
    }
    return $eventos;//Devuelvo un array con los datos de todos los coordinadores
}
```

En la misma vista del usuario incluiremos el código PHP para que, una vez llamada a la función que permite listar los eventos, se “impriman” las filas de la tabla que ocupa el espacio principal de la vista.

```
<?php
require_once("../php/database.php");
echo "<h3>LISTADO EVENTOS</h3>";

$eventos = listarEventos($con);
if(count($eventos) == 0){
    echo "<br/>No hay eventos<br/>";
}
else{
    foreach($eventos as $evento){
        $id_evento=$evento['id'];
        $num_evento= 1;
        echo "<tr>
            <td id= evento".$id_evento.">".$id_evento."</td>
            <td>".$evento['nombre']."</td>
            <td>".$evento['lugar']."</td>
            <td>".$evento['coordinador']."</td>
            <td><button type='button' class='btn btn-info btn-block' onclick='addRowHandlers()'>Info</button></td>
            <td><button type='button' class='btn btn-info btn-block'><a href='php/apuntarse.php?evento=".$evento['id']."'>Apuntarse</a></button></td>
        </tr>";
        $num_evento++;
    }
}
```

NOTA: cabe destacar que al crear la fila de cada evento, le hemos añadido un id que se incrementa dinámicamente. Además, la última celda de la fila contiene un botón para que el usuario pueda apuntarse al evento.

En el caso de querer ver solamente los eventos en los que está inscrito el usuario, el procedimiento es el mismo, si bien cambia la consulta mediante la que obtenemos los resultados de la consulta sobre la tabla eventos. Para ello, cruzamos dicha tabla con la tabla evento\_voluntario para así seleccionar sólo aquellos eventos que cuentan con éste inscrito.

```
function listarMisEventos($con, $dni){
    $result = mysqli_query($con, "select e.idevento as id, e.nombre as nombre, l.nombre as lugar, p.nombre as coordinador
    from evento e, persona p, coordinador c, lugar l, evento_voluntario ev, voluntario v where c.persona = p.dni and
    e.coordinador = c.idcoordinador and e.lugar = l.idlugar and ev.evento=e.idevento and ev.voluntario=v.idvoluntario and v.persona='".$dni."'");
    $coordinadores = array();
    while($fila = mysqli_fetch_array($result)){
        $eventos[] = $fila;
    }
    return $eventos;//Devuelvo un array con los datos de todos los coordinadores
}
```

## Apuntarse a un evento

El mencionado botón permite que al hacer click el usuario pueda apuntarse automáticamente al evento. Para ello, se utiliza la variable de sesión que contiene la variable de usuario. Asimismo, al hacer click en este botón, el usuario ha enviado también el id del evento. Con estos dos datos, podemos insertar el nuevo registro en la tabla evento\_voluntario, que relaciona a ambos.

```
1 <?php
2 require_once("database.php");
3 require_once("control_sesion_voluntario.php");
4 $idevento = $_GET['evento'];
5 $persona=obtenerIdVoluntario($con, $dni);
6 $idvoluntario=$persona['idvoluntario'];
7 voluntarioEvento($con, $idvoluntario, $idevento);
8 echo "¡Te has inscrito correctamente!";
9 </?>
```

En database.php:

```
function obtenerIdVoluntario($con, $dni){
    $resultado = mysqli_query($con, "select v.idvoluntario from voluntario v, persona p where p.dni=v.persona and p.dni='$dni'");
    if(mysqli_num_rows($resultado)==0){
        return 0; //Si no existe el usuario devuelvo 0
    }
    else{
        $persona = mysqli_fetch_array($resultado);
        return $persona;
    }
}
```

```
//APUNTARSE A UN EVENTO

function voluntarioEvento($con, $idvoluntario, $idevento){
    mysqli_query($con, "insert into evento_voluntario(voluntario,evento) values('$idvoluntario', '$idevento')");
}
```

## Generar Incidencia

Al hacer click sobre Incidencias, se abre una ventana que permite al voluntario dar cuenta de su incidencia. En esta ventana podrá seleccionar el tipo de incidencia (de entre un listado de posibilidades, el evento al que hace referencia la incidencia y una descripción de la misma.

El formulario, aunque sencillo, incluye un desplegable que, para contener el listado de eventos, necesita de una funcionalidad que recorra la tabla homónima y los liste como opciones de dicho desplegable.

Para ello, en el archivo database.php, la función desplegableLocalizaciones se encarga de generar tantas filas como resultados de la búsqueda de eventos hay:

```
function desplegableEventos($con){
    $consulta3 = "select nombre, idevento
    FROM evento";
    $resultado3 = mysqli_query($con, $consulta3);
    while($fila = mysqli_fetch_array($resultado3)){
        extract($fila);
        echo "<option value='$idevento'>$nombre</option>";
    }
}
```

Una vez seleccionado y cumplimentado el formulario, se envían los datos a insertarIncidenciaUsuario.php, que procederá a la creación de la incidencia, mediante el uso de la función insertarIncidencia() que contiene database.php:

*insertarIncidenciaUsuario.php*

```
1 <?php
2 require_once("database.php");
3 require_once("control_sesion_voluntario.php");
4 $dni=$_SESSION['dni'];
5 $evento = $_POST["tituloEventoSeleccionado"];
6 $tipoIncidencia = $_POST["tipoIncidenciaSeleccionada"];
7 $detalleIncidencia = $_POST["textAreaModal"];
8 $persona=obtenerIdVoluntario($con, $dni);
9 $voluntario=$persona['idvoluntario'];
10 if(empty($tipoIncidencia) || empty($detalleIncidencia) || empty($evento)){
11     echo "Debes rellenar todos los campos";
12 }
13 else{
14     insertarIncidencia($con, $voluntario, $evento, $tipoIncidencia, $detalleIncidencia);
15     echo '<BUTTON onclick="window.close();">Close me.</BUTTON>';
16 }
17 cerrarConexion($con);
```

database.php

```
//CREAR INCIDENCIAS
function insertarIncidencia($con, $voluntario, $evento, $tipoIncidencia, $detalleIncidencia){
    mysqli_query($con, "insert into incidencia(voluntario, evento, tipoIncidencia, detalleIncidencia)
    values('$voluntario', '$evento', '$tipoIncidencia', '$detalleIncidencia')");
}
```

## Vista Administrador

Esta vista, además de contener algunos botones y funcionalidades en el menú que no existen para los voluntarios, ofrece al coordinador la posibilidad de ver un listado de los voluntarios.

Para ello, se utiliza un mecanismo similar al que se utiliza para formar una tabla con los eventos, pero, en este caso, mediante una consulta sobre las tablas de voluntarios y personas.

```
<table id="myTable">
    <tr class="header">
        <th style="width:60%;">DNI</th>
        <th style="width:40%;">Nombre</th>
    </tr>
</table>
<?php
require_once("../php/database.php");
echo "<h3>LISTADO PERSONAS</h3>";

$personas = listarPersonas($con);
if(count($personas) == 0){
    echo "<br/>No hay personas<br/>";
}
else{
    foreach($personas as $persona){
        echo "<tr>
            <td>".$persona['dni']. "</td>
            <td>".$persona['nombre']. "</td>
        </tr>";
    }
}
cerrarConexion($con);
</table>
```

Asimismo, en database.php encontramos, nuevamente, la consulta:

```
//FUNCIÓN LISTAR PERSONAS
function listarPersonas($con){
    $result = mysqli_query($con, "select * from persona");
    $personas = array();
    while($fila = mysqli_fetch_array($result)){
        $personas[] = $fila;
    }
    return $personas; //Devuelvo un array con los datos de todos los usuarios
}
```



## Alta Voluntario

Para facilitar algunas funciones, hemos creado algunos botones de acceso rápido que se pueden ver en la parte superior de la vista de Administrador. Uno de ellos es el que lleva al coordinador a un formulario para poder crear un voluntario nuevo en la base de datos. Los datos del nuevo voluntario se envían a crearVoluntario.php:

```
require_once("database.php");
$dni = $_POST["dni"];
$nombre = $_POST["nombre"];
$apellidos = $_POST["apellidos"];
$telefono = $_POST["telefono"];
$direccion = $_POST["direccion"];
$ciudad = $_POST["ciudad"];
$email = $_POST["email"];
$susuario = $_POST["usuario"];
$contrasena = $_POST["contrasena"];
$horas = 0;
if(empty($dni) || empty($nombre) || empty($apellidos) || empty($telefono) || empty($direccion) || empty($ciudad) || empty($email))
    echo "Debes rellenar todos los campos";
else{
    insertarPersona($con, $dni, $nombre, $apellidos, $telefono, $direccion, $ciudad, $email, $susuario, $contrasena);
    //header("Location: admin.php");
    insertarVoluntario($con, $dni, $horas);
    echo '<BUTTON onclick="window.close();">Close me.</BUTTON>';
}
cerrarConexion($con);
```

Como vemos, mediante las funciones insertarPersona() y insertarVoluntario, se procede a insertar el nuevo registro en las dos tablas (persona y voluntario). Ambas funciones se hayan definidas en database.php:

```
function insertarPersona($con, $dni, $nombre, $apellidos, $telefono, $direccion, $ciudad, $email, $susuario, $contrasena){
    mysqli_query($con, "insert into persona values('$dni', '$nombre', '$apellidos', '$telefono', '$direccion', '$ciudad', '$email', '$susuario', '$contrasena')");
}

function insertarVoluntario($con, $dni, $horas){
    mysqli_query($con, "INSERT INTO `voluntario`(`persona`, `horas`) VALUES ('$dni', $horas)");
}
```

## Crear localización

Nuevamente, se utiliza un formulario que envía datos a crearLocalización.php para crear al localización (insertar un nuevo registro en la tabla lugar). Este archivo se sirve de la función definida en database.php que se muestra a continuación:

```
//CREAR LOCALIZACIÓN
function insertarLocalizacion($con, $nombreLugar, $longitud, $latitud){
    mysqli_query($con, "insert into lugar(nombre, longitud, latitud) values('$nombreLugar', '$longitud', '$latitud')");
}
```

## Crear Evento (desplegables)

Siguiendo el patrón establecido, mediante un formulario enviamos los datos al archivo `insertarEvento.php` para la creación del mismo. Ya hemos explicado cómo se realiza el proceso en los párrafos anteriores: se envían los datos del formulario a un archivo php que a su vez llama a las funciones definidas en `database.php` para ejecutar la query que permite insertar el registro en la tabla adecuada.

La particularidad en este caso se encuentra en el formulario, ya que contiene dos desplegados, uno con las localizaciones y otro con el nombre de los coordinadores (que permite elegir quién es el coordinador responsable de una actividad).

Para el desplegable con localizaciones utilizamos la función con una consulta simple sobre la tabla que almacena los lugares, mientras que para el desplegable de coordinadores, la consulta será algo más compleja, al tener que cotejar las tablas `persona` y `coordinador` al mismo tiempo:

```
//////// DESPLEGABLES //////////

//localizaciones//
function desplegableLocalizaciones($con){
    $consulta2 = "select idlugar, nombre FROM lugar;";
    $resultado2 = mysqli_query($con, $consulta2);

    while($fila = mysqli_fetch_array($resultado2)){
        extract($fila);
        echo "<option value='$idlugar'>$nombre</option>";
    }
}

//coordinadores//
function desplegableCoordinadores($con){
    $consulta1 = "select coordinador.idcoordinador, persona.nombre, persona.apellidos
    FROM coordinador INNER JOIN persona ON coordinador.persona = persona.DNI;";
    $resultado1 = mysqli_query($con, $consulta1);

    while($fila = mysqli_fetch_array($resultado1)){
        extract($fila);
        echo "<option value='$idcoordinador'>$nombre - $apellidos</option>";
    }
}
```



## Interfaz

### Funcionalidades Front-End

Durante la fase de planteamiento inicial del proyecto se propusieron diferentes formas de enfocar la visualización de datos de cara al usuario, tanto en la vista de administrador como en la de voluntario.

Sobre la mesa se planteaba el reto de desarrollar una interfaz dinámica y amigable, a la que vez potente en funciones. De cara a lograr este objetivo se buscó una solución del tipo presentación de los datos mediante tablas, desde las cuales se pudiese navegar a ventanas modales donde se ofrecía una vista ampliada de la información del evento seleccionado. Como extra dentro de esta funcionalidad modal, se puede ver una segunda subsección llamada “Mapa” que nos carga un mapa de la zona donde se desarrollara el evento y un marcador azul indicando la posición geolocalizada del mismo. Este “canvas” es un mapa como el que podemos encontrar en herramientas tan conocidas como Google Maps, y nos permite realizar zooms in & out, desplazarnos por él y presentar información extra sobre que representa el marcador en un pop-up desplegable sobre este mismo.

En lo referente al aspecto general de la aplicación, en el margen superior izquierdo podemos apreciar un botón que nos permite “Mostrar/Ocultar” el área de navegación a fin de que esta no obstaculice la interacción con la web. Este mismo botón también gestiona la redimensión de las tablas, buscando ofrecer siempre un balance entre información a mostrar y comodidad a la hora de leer la misma.

Tanto el menú de navegación lateral como las ventanas de tipo modal son dos elementos con los cuales nos cruzamos repetidas veces a lo largo de la web, dado que se buscaba dar una cohesión a las diferentes vistas que forman la página a fin de que la navegación fuese lo más intuitiva posible.

### Descripción Interfaz

Una vez bajado el proyecto e instaladas las dependencias o tras haber accedido a la página web vía URL (depende del modo elegido), nos encontramos con una ventana de login donde se nos solicita usuario y contraseña.

Durante el proceso de desarrollo del proyecto se crearon en torno a media docena de perfiles, los cuales pueden verse en los anexo relativos a creación de la base de datos e inserción de datos en la misma. Dentro de estos, como ejemplo, podemos escoger las combinaciones:

**Administrador:**

Usuario: admin2

Password: admin2

**Voluntarios:**

Usuario: tamara

Password: tamara


Usuario: mikel

Password: mikel

En el caso de acceder con las credenciales de administrador iremos a la vista administrador y de usar las de voluntario, terminaremos en la vista preparada para estos. A continuación procedemos a presentar ambas secciones del proyecto.

*Login:*

**Formulario Login**



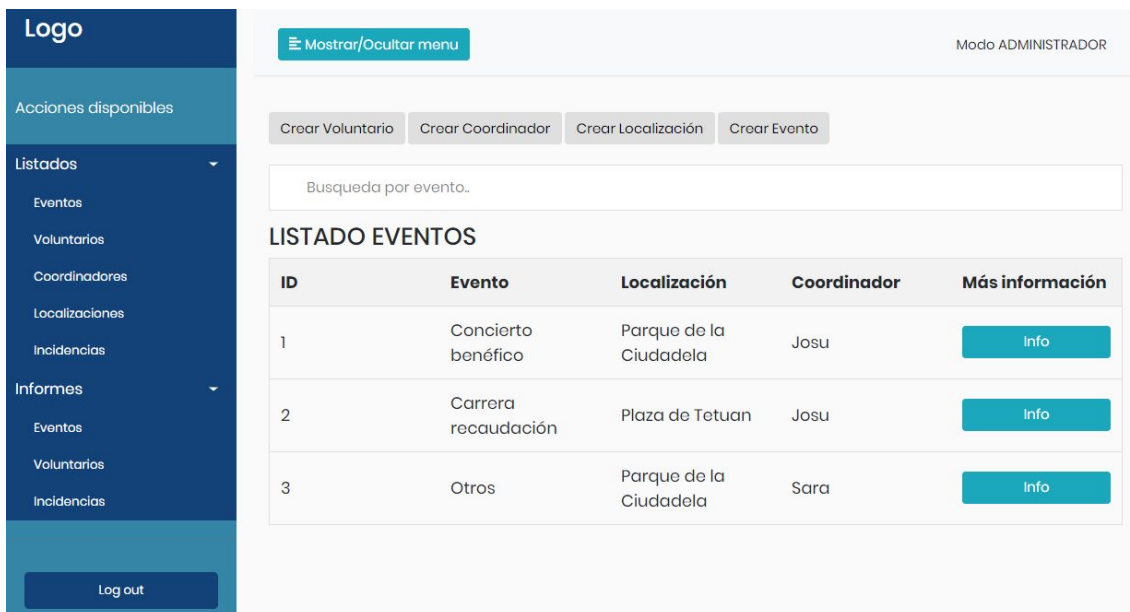
**Username**

**Password**

Login

Según accedemos a la web, se nos presenta el formulario de Login mostrado en la imagen precedente. En el mismo, tenemos la opción de acceder como voluntario, o bien como administrador, dependiendo de qué credenciales le suministremos. Dichas credenciales ya han sido presentadas en el punto anterior; así que queda en mano de quien vaya a consultar este manual, el elegir que vista ver primero. En nuestro caso, primero iremos a la versión accesible tras un registro positivo como administrador.

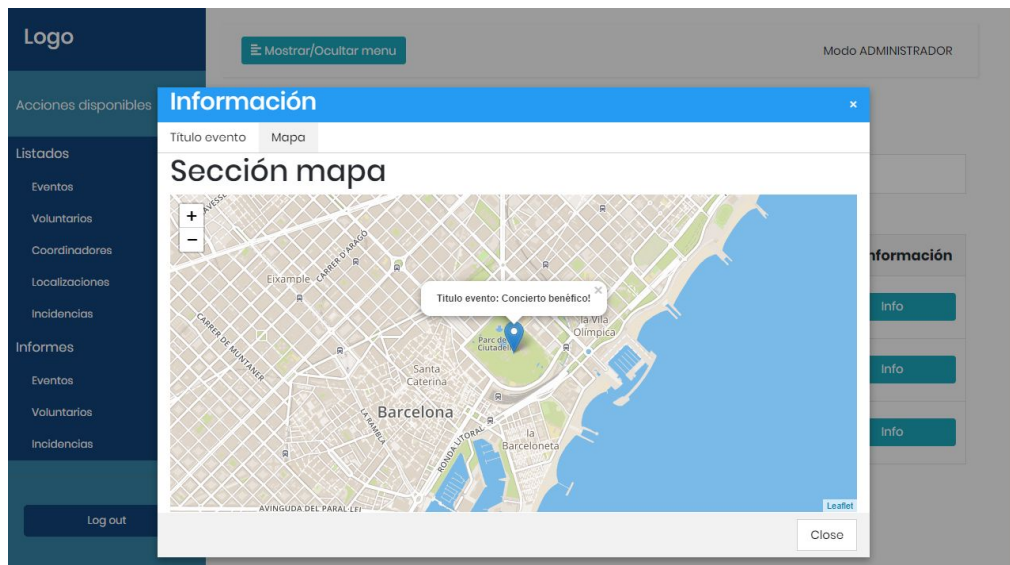
#### Vista administrador:



ID	Evento	Localización	Coordinador	Más información
1	Concierto benéfico	Parque de la Ciudadela	Josu	<a href="#">Info</a>
2	Carrera recaudación	Plaza de Tetuan	Josu	<a href="#">Info</a>
3	Otros	Parque de la Ciudadela	Sara	<a href="#">Info</a>

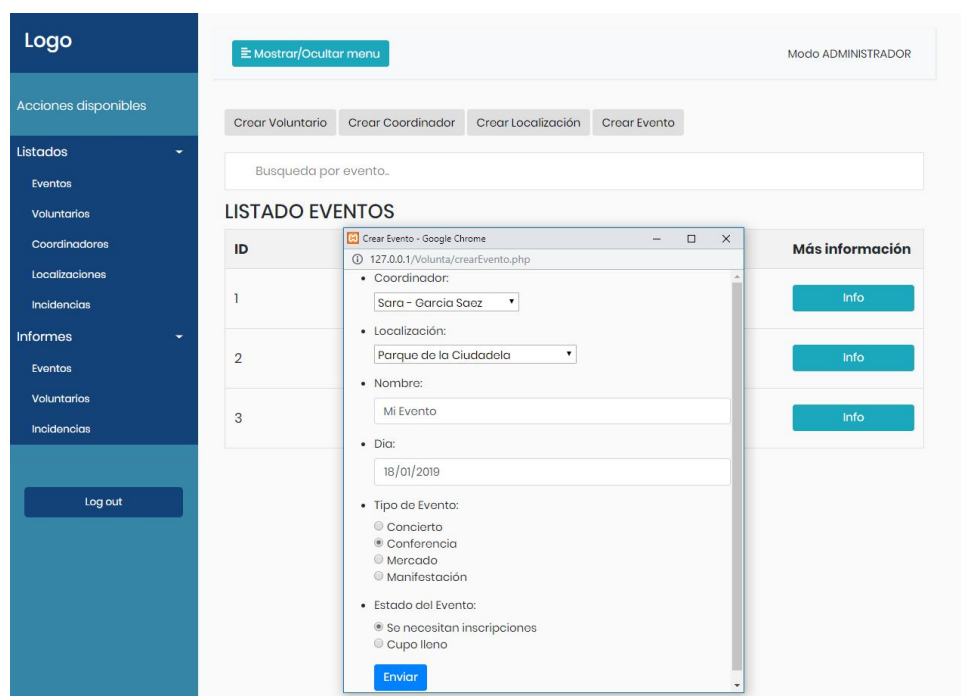
La vista administrador muestra el patrón de diseño que sigue toda la aplicación, tanto en su modo administrador, como orientado a voluntarios. En la parte izquierda, tenemos un menú con una serie de desplegados, que nos dan acceso a las diferentes opciones de cada modo. Este menú, se puede mostrar u ocultar, con el botón “Mostrar/Ocultar menú”. Esta funcionalidad fue implementada pensando en aprovechar el máximo espacio de pantalla para mostrar información y solo presentar el menú lateral si el usuario así lo desea.

Dentro de esta vista podemos encontrar diferentes opciones. Por ejemplo, si hacemos click en “Info” se nos muestra una ventana modal que nos presenta una parte de la información relativa al evento seleccionado. En este modal, encontramos la pestaña “Mapa”, la cual nos lleva a una vista de una canvas Leaflet, el cual nos indica mediante un marcador, el lugar donde el evento tendrá lugar.



Por otra parte, se puede apreciar en la parte superior de esta vista una barra de búsqueda, que nos permitirá filtrar los resultados de la tabla en base al texto introducido. Sobre esta barra, encontramos una serie de botones, los cuales nos permiten crear nuevos voluntarios, coordinadores, lugares o eventos.

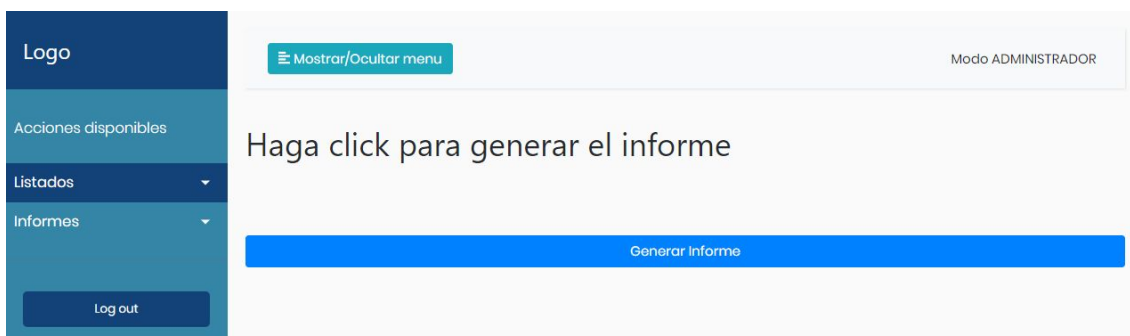
En la imagen siguiente podemos apreciar lo que sucede al hacer clic en uno de esos botones.



El modal que se nos abre nos muestra una serie de opciones, en este caso relativas a la creación de un nuevo evento. Una vez rellenados todos los campos y pulsado “Enviar”, encontraremos la nueva información en base de datos y en la tabla de la web que corresponda (en este último caso, una vez refresquemos el navegador).

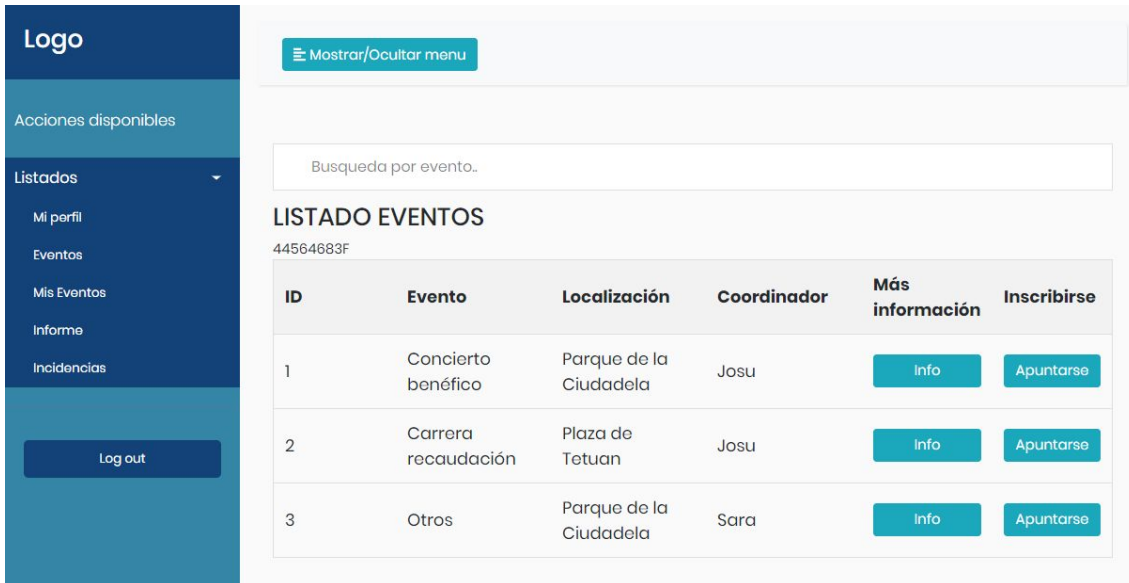
Dado que el resto de las vistas son similares a esta, salvo por pequeñas variaciones, saltaremos directamente a explicar la sección relativa a informes.

Como podemos apreciar en la imagen de abajo, la plantilla de la web es la misma, con la diferencia de que una vez elegido el tipo de informe que queremos, procedemos a “onclick” generarlo. Esto activa la descarga de un PDF en el cual podemos ver la información en base de datos relativa a la sección escogida.



### Vista voluntario:

De haber entrado a través de la pantalla de login con unas credenciales de voluntario, nos encontraremos con una diseño como el que se ve a continuación.



ID	Evento	Localización	Coordinador	Más información	Inscribirse
1	Concierto benéfico	Parque de la Ciudadela	Josu	<a href="#">Info</a>	<a href="#">Apuntarse</a>
2	Carrera recaudación	Plaza de Tetuan	Josu	<a href="#">Info</a>	<a href="#">Apuntarse</a>
3	Otros	Parque de la Ciudadela	Sara	<a href="#">Info</a>	<a href="#">Apuntarse</a>

La estructura es prácticamente idéntica a la vista administrador, salvo que este caso toda la información mostrada hace referencia únicamente al usuario con el que nos hayamos registrado.

Por una parte, en “Mi perfil” encontramos un resumen de todos los datos relativos al voluntario que tenemos en la base de datos. Después, la vista eventos muestra todos los eventos disponibles y da la opción de apuntarse a ellos con el consecuente botón.

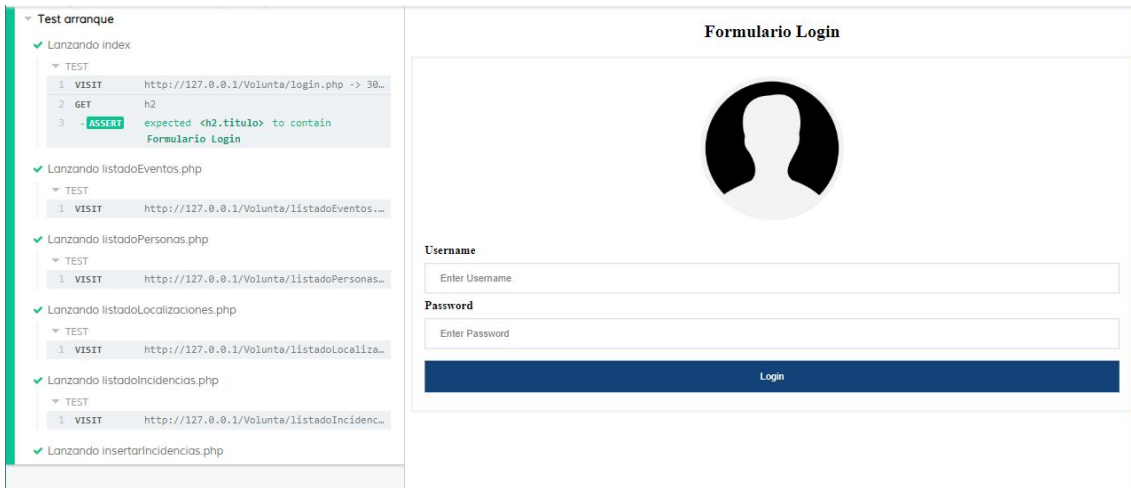
Por su parte, en “Mis Eventos”, podemos ver los eventos a los que el voluntario ya esta apuntado y en “Informe” nos descargamos un PDF con los datos del voluntario y su resumen de horas como voluntario.

Para finalizar, haciendo clic en “Incidencias” tenemos la posibilidad de colocar una incidencia relativa a un evento e insertarlo en base de datos para que después el administrador pueda verlo desde su perfil.

## Pruebas unitarias

De cara a comprobar que la aplicación funciona correctamente, se desarrollaron unas pequeñas pruebas unitarias en los inicios del proyecto. Ahora mismo no cubren más que una pequeña parte del proyecto, pero sirven para comprobar que al menos algunas de las vistas funcionan correctamente. Para acceder a las mismas debemos seguir las instrucciones que aparecen en el archivo README.md.

Si todo ha ido bien, deberemos ver una vista como la siguiente:



The image shows a web browser window displaying a login form titled "Formulario Login". The form includes a circular profile picture placeholder, a "Username" field with the placeholder text "Enter Username", a "Password" field with the placeholder text "Enter Password", and a blue "Login" button. To the left of the browser window, a test runner interface is visible, showing a list of tests under the heading "Test arranque". The tests are as follows:

- ✓ Lanzando index
  - TEST
    - 1 VISIT http://127.0.0.1/Volunte/login.php -> 30...
    - 2 GET h2
    - 3 ASSERT expected <h2.titulo> to contain Formulario Login
- ✓ Lanzando listadoEventos.php
  - TEST
    - 1 VISIT http://127.0.0.1/Volunte/listadoEventos...
- ✓ Lanzando listadoPersonas.php
  - TEST
    - 1 VISIT http://127.0.0.1/Volunte/listadoPersonas...
- ✓ Lanzando listadoLocalizaciones.php
  - TEST
    - 1 VISIT http://127.0.0.1/Volunte/listadoLocaliza...
- ✓ Lanzando listadoIncidencias.php
  - TEST
    - 1 VISIT http://127.0.0.1/Volunte/listadoIncidenc...
- ✓ Lanzando insertarIncidencias.php

## Servidor

El despliegue en web se ha realizado sobre un servicio de host ofrecido por “000webhost”, en el cual hemos alojado todo el código del repositorio <https://github.com/JonanOribes/Volunta> con las URLs modificadas para trabajar bajo este servicio. Por su parte la base de datos es MySQL gestionada desde PHPAdmin. El modelo de esta, es idéntico al que se ve si se despliega la versión del repositorio sobre localhost.

En caso de querer ir a la versión web acceder directamente a:  
<http://voluntatest.000webhostapp.com/index.php>

Para entrar como administrador usar **user:** admin2 y **password:** admin2

o

Para entrar como voluntario usar **user:** tamara y **password:** tamara

o

Para entrar como voluntario usar **user:** mikel y **password:** mikel

o

Cualquiera de los que vienen ya creados en la base de datos o se creen después desde dentro de la web.

Para desplegar el proyecto desde localhost necesitamos contar con XAMPP. Dentro de este, en la carpeta “htdocs”, descomprimos el proyecto. Tras esto abrimos el mismo con un IDE, por ejemplo Visual Code, y lanzamos desde dentro del directorio el comando “npm install” el cual proceder a instalar todas las dependencias que son necesarias para que Volunta funcione.



Una vez realizados los pasos anteriores, procedemos a iniciar el XAMPP y activar los módulos Apache y MySQL. Con todo esto corriendo, ya deberíamos poder acceder a la web abriendo un navegador en la URL "<http://127.0.0.1/Volunta/index.php>". Además de esto, debemos crear la base de datos en el PHPAdmin (<http://localhost/phpmyadmin/>), creando primero la base de datos y las tablas, copiando el contenido de "**Volunta/BDD/CREACION/Tablas.sql**" dentro del ejecutor de SQL y después insertando datos en las mismas, copiando y ejecutando el contenido de "**Volunta/BDD/CREACION/DatosEjemplo.sql**". Con estos dos últimos pasos ya contamos con una serie de administradores, voluntarios, eventos, etc con los que empezar a explorar el proyecto.

Si por algún motivo, el archivo comprimido con el proyecto no se abriese bien, contamos con una copia en el repositorio <https://github.com/JonanOribes/Volunta>, donde además se puede ir viendo cómo este proyecto se ha ido desarrollando a lo largo de los meses.

## 4. Conclusiones

---

### Valoración del proyecto

Consideramos que durante los últimos meses, el proyecto ha ido tomando un buen rumbo, desde la idea inicial que se planteó hasta el resultado que hemos obtenido. Si bien, en un primer momento, sobre el papel, todo parecía que iba a funcionar sin problemas, según se fueron desarrollando las diversas funcionalidades de las que dependería el éxito o fracaso del producto final, fuimos encontrándonos con alguna que otra dificultad.

Principalmente, estas complicaciones surgieron a la hora de conectar la parte desarrollada en javascript con el PHP, dado que aun, conociendo ambos lenguajes de programación, jamás habíamos tenido que trabajar con ellos dos juntos a un nivel de complejidad tan alto. Aun con todo eso, conseguimos sacar adelante la propuesta planteada en un primer momento y hacer modificaciones, que consideramos pequeñas mejoras, sobre la marcha.

Si bien, algunas cosas se tendrán que quedar para futuras versiones de la web - por ejemplo, las líneas temporales dinámicas que pretendíamos implementar mediante la librería `Vis.js` (aquí dejamos un ejemplo <http://visjs.org/examples/timeline/groups/groups.html>) - otras muchas, como el sistema de mapas con Leaflet o el uso de JQuery para trabajar contra el DOM, han sido integradas con éxito.

En resumidas cuentas, estamos satisfechos con el resultado tras haber conseguido dos objetivos que considerábamos fundamentales: que el resultado fuera una aplicación útil y apreciada por el cliente potencial y superar el reto que supone utilizar diferentes lenguajes entrelazados entre sí (en contraposición a las asignaturas del módulo, en que cada lenguaje se estudiaba de manera algo más aislada).

## Posibles ampliaciones

Una vez terminada de desarrollar la presente versión de Volunta, nos empezamos a preguntar cuáles serían los posibles caminos hacia donde podríamos enfocar las siguientes etapas.

Dado que nos interesa lograr una versión 2.0 que resulte fácil de escalar, lo primero de cara a la parte del servidor, sería estudiar la implementación de un framework PHP, como Symfony. Además al tratarse de un framework muy extendido en el mundo profesional, existe una gran documentación y comunidad tras él, lo que facilita no solo su adopción como parte de nuestro proyecto, si no también el poder tener la seguridad de que se trata de un desarrollo que a medio-largo plazo seguirá recibiendo mejoras continuas.

Por otra parte, nos planteamos la implementación de una plataforma de evaluación de código. Dentro de este área, la idea, sería utilizar SonarQube, gracias a la cual podríamos ir generando un código más limpio, eficiente y sencillo de mantener, pues a medida que el proyecto vaya creciendo, el código duplicado, los pequeños bugs y malas prácticas podrían terminar por hacerlo algo imposible de continuar.

Una vez estuviesen las dos herramientas arriba mencionadas funcionando dentro de nuestra web, nos gustaría hacer uso de Cordova para desarrollar el interfaz de voluntario dentro de una aplicación móvil. Este sistema nos permitirá seguir desarrollando toda la aplicación en Javascript, pero a su vez disponer de una versión tanto para Android como para IOS, lo cual consideramos que es mucho más realista como forma de acercarse a nuestro público objetivo. Por su parte, la parte administrador seguirá siendo solo web.

Asimismo, y tras haber hablado con los potenciales usuarios de la aplicación, no descartamos ampliar las funcionalidades de la misma. Por ejemplo, y en el caso de PRIDE BCN, la entidad baraja la creación de un carnet para voluntarios, que permita acumular puntos y vaya ligado a las horas dedicadas. La entidad aún no tiene claro cómo será este programa de fidelización del voluntariado, pero constituirá, sin duda alguna, una interesante línea a seguir para ampliar las funcionalidades de la aplicación una vez la organización tenga definido dicho programa.

Para finalizar, nos gustaría generar un fichero de internacionalización desde el cual presentar la aplicación, al menos, tanto en castellano como inglés.

## 5. Bibliografía

---

*Javascript y JQuery. 3ª Edición (David Sawyer MCFarland, Editorial Anaya)*

*Desarrollo Web con PHP y MySQL. 5ª Edición (Luke Welling, Laura Thomson, Editorial Anaya).*

<https://www.w3schools.com/>

<https://developer.mozilla.org/es/>

<https://leafletjs.com/>

<https://support.mockflow.com/category/92-video-tutorials>

<https://help.github.com/desktop/guides/getting-started-with-github-desktop/>

<https://help.github.com/desktop/guides/contributing-to-projects/>

<https://code.visualstudio.com/docs>

<https://blog.udemy.com/tutorial-de-xampp-como-usar-xampp-para-ejecutar-su-propio-servidor-web/>

<https://docs.phpmyadmin.net/en/latest/>

<https://dev.mysql.com/doc/workbench/en/>

<https://support.mockflow.com/category/92-video-tutorials>

<https://about.draw.io/category/tutorial/>

<https://symfony.com/what-is-symfony>

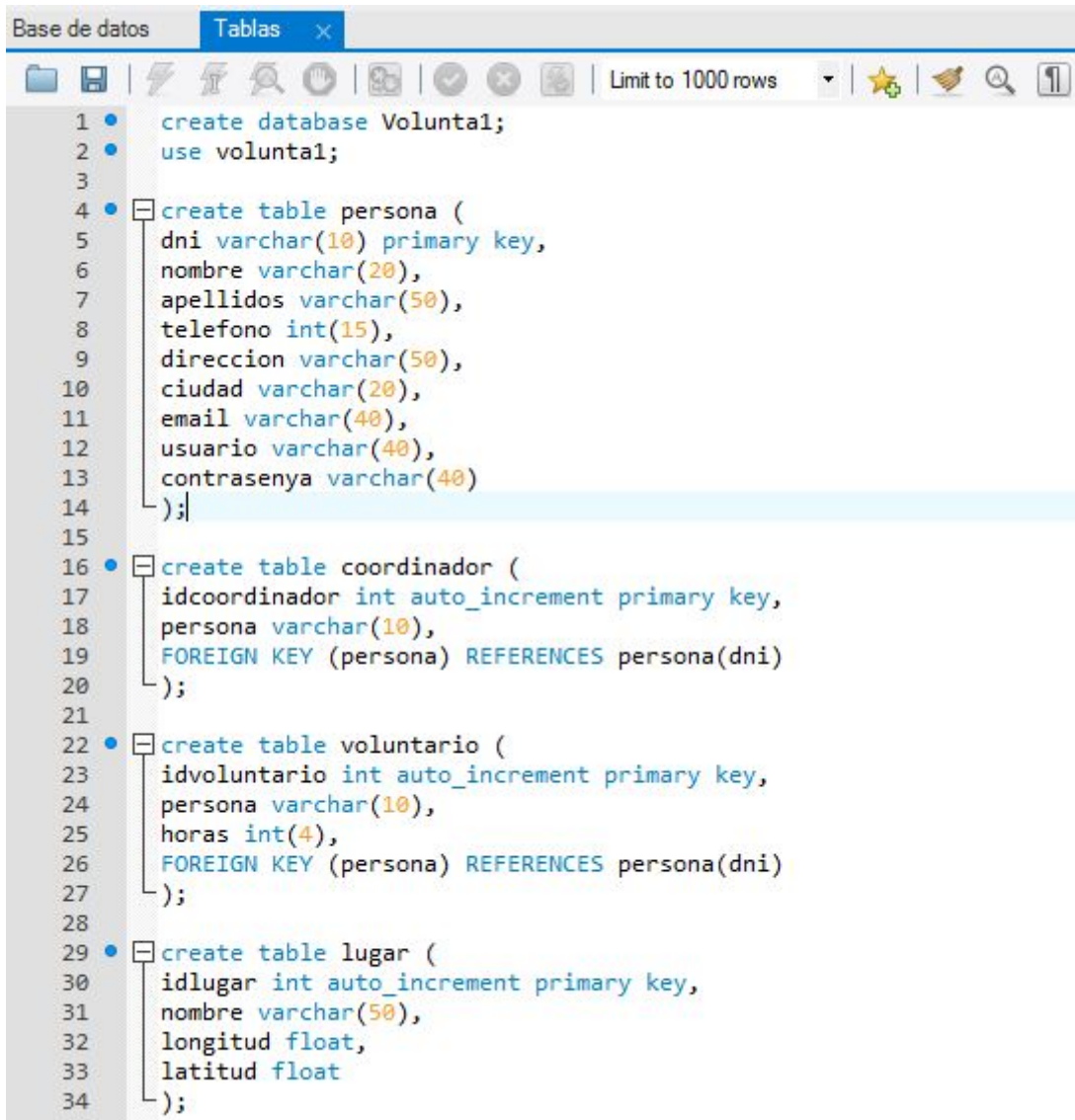
<https://en.wikipedia.org/wiki/SonarQube>

<https://en.wikipedia.org/wiki/SonarQube>

**NOTA:** asimismo, se ha consultado el material facilitado por Linkia durante el módulo, particularmente presentaciones de profesores.

## 6. Anexos

### Base de Datos



```
Base de datos  Tablas x
Limit to 1000 rows

1 • create database Voluntar1;
2 • use voluntar1;
3
4 • create table persona (
5   dni varchar(10) primary key,
6   nombre varchar(20),
7   apellidos varchar(50),
8   telefono int(15),
9   direccion varchar(50),
10  ciudad varchar(20),
11  email varchar(40),
12  usuario varchar(40),
13  contrasena varchar(40)
14 );
15
16 • create table coordinador (
17   idcoordinador int auto_increment primary key,
18   persona varchar(10),
19   FOREIGN KEY (persona) REFERENCES persona(dni)
20 );
21
22 • create table voluntario (
23   idvoluntario int auto_increment primary key,
24   persona varchar(10),
25   horas int(4),
26   FOREIGN KEY (persona) REFERENCES persona(dni)
27 );
28
29 • create table lugar (
30   idlugar int auto_increment primary key,
31   nombre varchar(50),
32   longitud float,
33   latitud float
34 );
35
```

```
36 • create table evento (  
37     idevento int auto_increment primary key,  
38     coordinador int,  
39     lugar int,  
40     nombre varchar(50),  
41     diaEvento date,  
42     tipo varchar(50),  
43     estado boolean,  
44     FOREIGN KEY (coordinador) REFERENCES coordinador(idcoordinador),  
45     FOREIGN KEY (lugar) REFERENCES lugar(idlugar)  
46 );  
47  
48 • create table evento_voluntario (  
49     voluntario int,  
50     evento int,  
51     FOREIGN KEY (voluntario) REFERENCES voluntario(idvoluntario),  
52     FOREIGN KEY (evento) REFERENCES evento(idevento)  
53 );  
54  
55  
56 • create table incidencia (  
57     idincidencia int auto_increment primary key,  
58     voluntario int,  
59     evento int,  
60     tipoIncidencia VARCHAR(260),  
61     detalleIncidencia VARCHAR(260),  
62     FOREIGN KEY (voluntario) REFERENCES voluntario(idvoluntario),  
63     FOREIGN KEY (evento) REFERENCES evento(idevento)  
64 );  
65
```