



I E S   A L B A R R E G A S

TÉCNICO SUPERIOR EN DESARROLLO DE  
APLICACIONES WEB

Departamento de Informática

PROYECTO HISPANIA EV

**Manual Técnico**

Autor/es: Daniel Indias Corral  
Curso Académico: 2º DAW

## Índice

- 1. Introducción**
- 2. Arquitectura de la aplicación**
  - 2.1. Frontend**
    - 2.1.1. Tecnologías usadas**
    - 2.1.2. Entorno de desarrollo**
  - 2.2. Backend**
    - 2.2.1. Tecnologías usadas**
    - 2.2.2. Entorno de desarrollo**
- 3. Documentación técnica**
  - 3.1. Análisis**
    - 3.1.1. Requisitos funcionales**
    - 3.1.2. Requisitos no funcionales**
  - 3.2. Desarrollo**
    - 3.2.1. Diseño de la BBDD**
      - **Modelo Entidad / Relación de la BBDD**
      - **Modelo relacional de la BBDD**
    - 3.2.2. Diagrama de casos de uso**
  - 3.3. Pruebas realizadas**
- 4. Proceso de despliegue**
- 5. Propuesta de mejoras**
- 6. Bibliografía**

## 1. Introducción

Hispania EV es una aplicación web de alquiler de vehículos híbridos y eléctricos.

El objetivo es proporcionar una alternativa de movimiento sencilla a las actuales plataformas de alquiler de vehículos que no esté ligada al coste incremental de los combustibles fósiles, a la vez que manteniendo costes accesibles.

## 2. Arquitectura de la aplicación

### 2.1. Frontend

En el Frontend se ha usado Bootstrap 5 junto a CSS y SCSS con el objetivo de lograr una página responsive y ordenada y un entorno precargado mas potente que CSS.

En la modificación del HTML de forma dinámica se han integrado llamadas asíncronas con Javascript dado que Bootstrap 5 no hace uso de jquery. Además, se han usado emoticonos de FontAwesome un agradable aspecto visual y distintas funcionalidades como serían las notificaciones.

#### 2.1.1. Tecnologías usadas

**HTML5 y CSS3.**

**SCSS:** Preprocesador CSS para facilitar la escritura y el mantenimiento del código.

**Bootstrap 5.3.2.**

**JavaScript.**

**FontAwesome**

#### 2.1.2. Entorno de desarrollo

**XAMPP/LAMP.**

**Git:** Control de versiones.

**PHP Storm 2023.3:**

### 2.2. Backend

Dentro del Backend se ha hecho uso de la última versión de PHP disponible junto a MySQL y Javascript.

Se aprovechan al máximo las llamadas asíncronas con Javascript y la funcionalidad de comprobar los datos en servidor de PHP para realizar una capa intermedia entre cliente y datos. Manejando así todos los posibles errores y mitigando posibles problemas de seguridad.

## 2.2.1. Tecnologías usadas

**PHP 8.2**

**MySQL.**

**Javascript:** Usado para llamadas asíncronas entre cliente-servidor-sql.

## 2.2.2. Entorno de desarrollo

**XAMPP/LAMP:** Para crear un servidor local con Apache, MySQL y PHP.

**phpMyAdmin:** para gestionar la base de datos durante el desarrollo.

**PHPStorm 2023.3:** IDE específicamente hecho para PHP.

## 3. Documentación técnica

### 3.1. Análisis

Para el desarrollo de la aplicación se ha seguido el siguiente diagrama. A partir del cual se ha desarrollado la base de datos

Diagrama de clases:

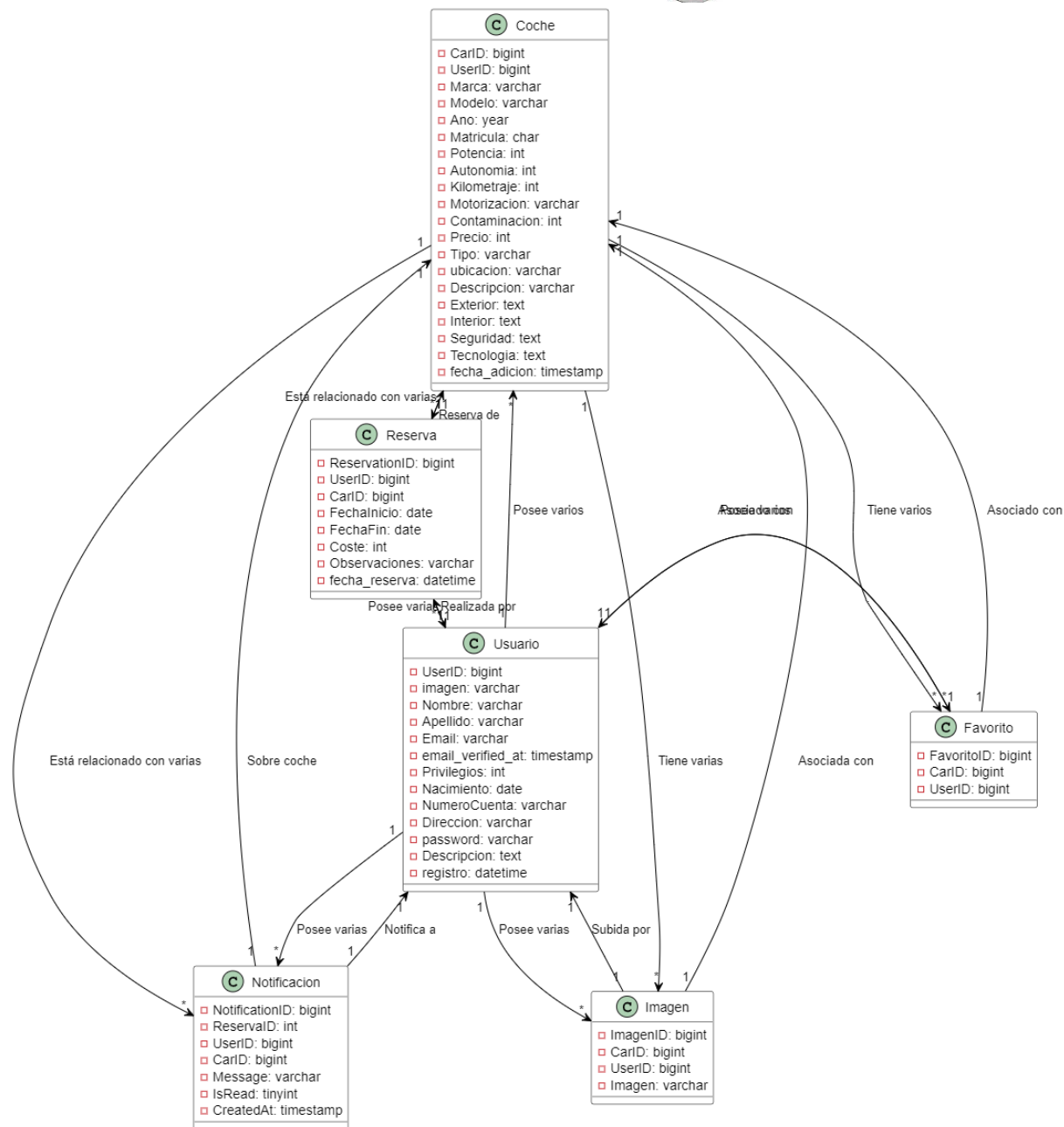
# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo



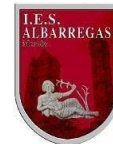
A continuación el código de plantuml para entender mejor el diagrama:

```
@startuml
' Estilo de las clases
skinparam class {
    BackgroundColor White
    BorderColor Black
    ArrowColor Black
}

' Clases
class Usuario {
    - UserID: bigint
```

# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

```
- imagen: varchar
- Nombre: varchar
- Apellido: varchar
- Email: varchar
- email_verified_at: timestamp
- Privilegios: int
- Nacimiento: date
- NumeroCuenta: varchar
- Direccion: varchar
- password: varchar
- Descripcion: text
- registro: datetime
}
```

```
class Coche {
- CarID: bigint
- UserID: bigint
- Marca: varchar
- Modelo: varchar
- Ano: year
- Matricula: char
- Potencia: int
- Autonomia: int
- Kilometraje: int
- Motorizacion: varchar
- Contaminacion: int
- Precio: int
- Tipo: varchar
- ubicacion: varchar
- Descripcion: varchar
- Exterior: text
- Interior: text
- Seguridad: text
- Tecnologia: text
- fecha_adicion: timestamp
}
```

```
class Favorito {
- FavoritoID: bigint
- CarID: bigint
- UserID: bigint
}
```

```
class Imagen {
- ImagenID: bigint
- CarID: bigint
- UserID: bigint
- Imagen: varchar
}
```

```
}

class Notificacion {
- NotificationID: bigint
- ReservaID: int
- UserID: bigint
- CarID: bigint
- Message: varchar
- IsRead: tinyint
- CreatedAt: timestamp
}

class Reserva {
- ReservationID: bigint
- UserID: bigint
- CarID: bigint
- FechaInicio: date
- FechaFin: date
- Coste: int
- Observaciones: varchar
- fecha_reserva: datetime
}

' Relaciones
Usuario "1" -down-> "*" Coche : "Posee varios"
Usuario "1" -down-> "*" Favorito : "Posee varios"
Usuario "1" -down-> "*" Imagen : "Posee varias"
Usuario "1" -down-> "*" Notificacion : "Posee varias"
Usuario "1" -down-> "*" Reserva : "Posee varias"

Coche "1" -left-> "*" Favorito : "Tiene varios"
Coche "1" -left-> "*" Imagen : "Tiene varias"
Coche "1" -up-> "*" Notificacion : "Está relacionado con varias"
Coche "1" -right-> "*" Reserva : "Está relacionado con varias"

Favorito "1" -left-> "1" Usuario : "Asociado con"
Favorito "1" -right-> "1" Coche : "Asociado con"

Imagen "1" -up-> "1" Usuario : "Subida por"
Imagen "1" -down-> "1" Coche : "Asociada con"

Notificacion "1" -down-> "1" Usuario : "Notifica a"
Notificacion "1" -down-> "1" Coche : "Sobre coche"

Reserva "1" -right-> "1" Usuario : "Realizada por"
Reserva "1" -left-> "1" Coche : "Reserva de"
@enduml
```

## 3.2. Desarrollo

Diagrama Entidad-Relacion:

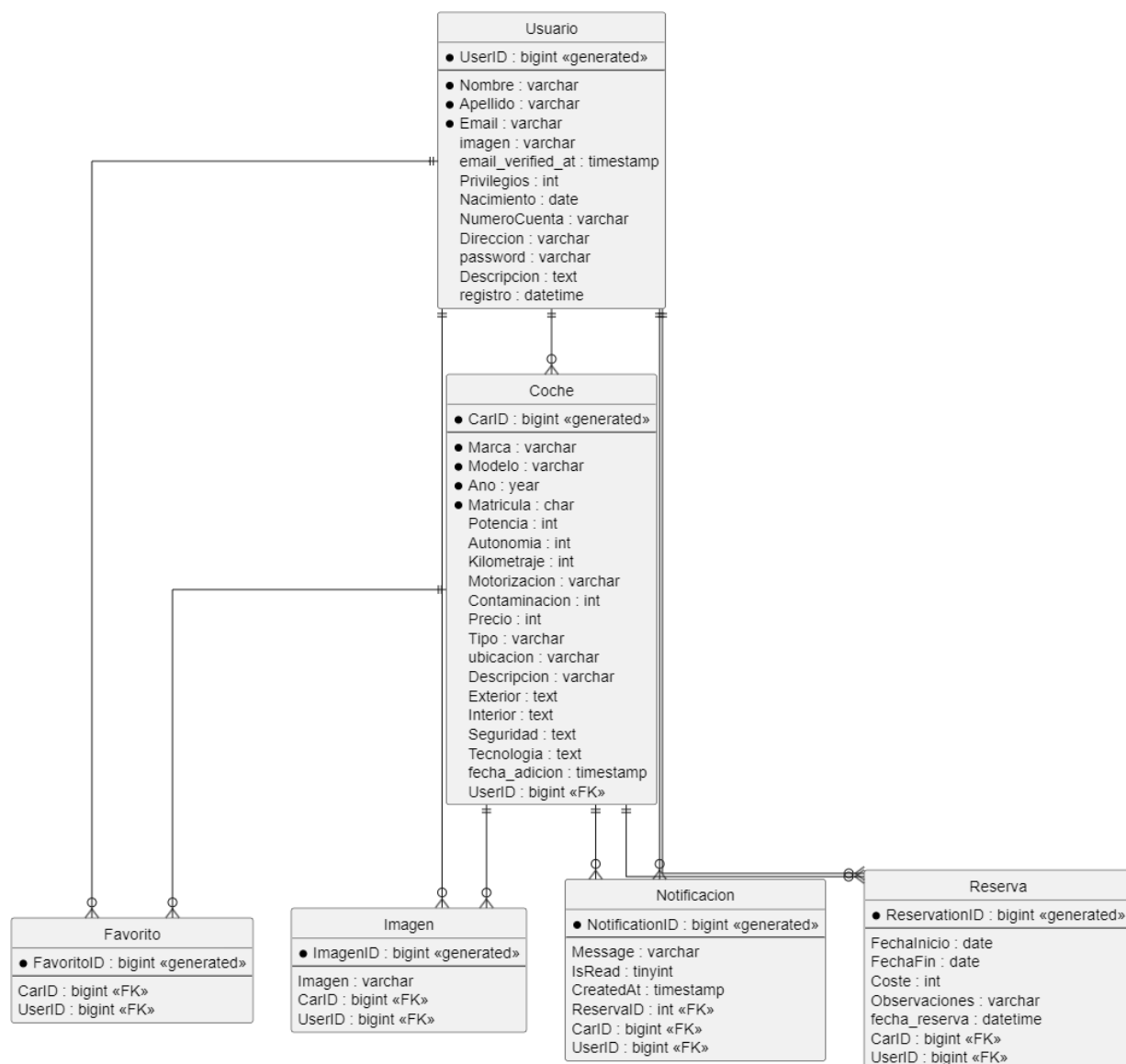
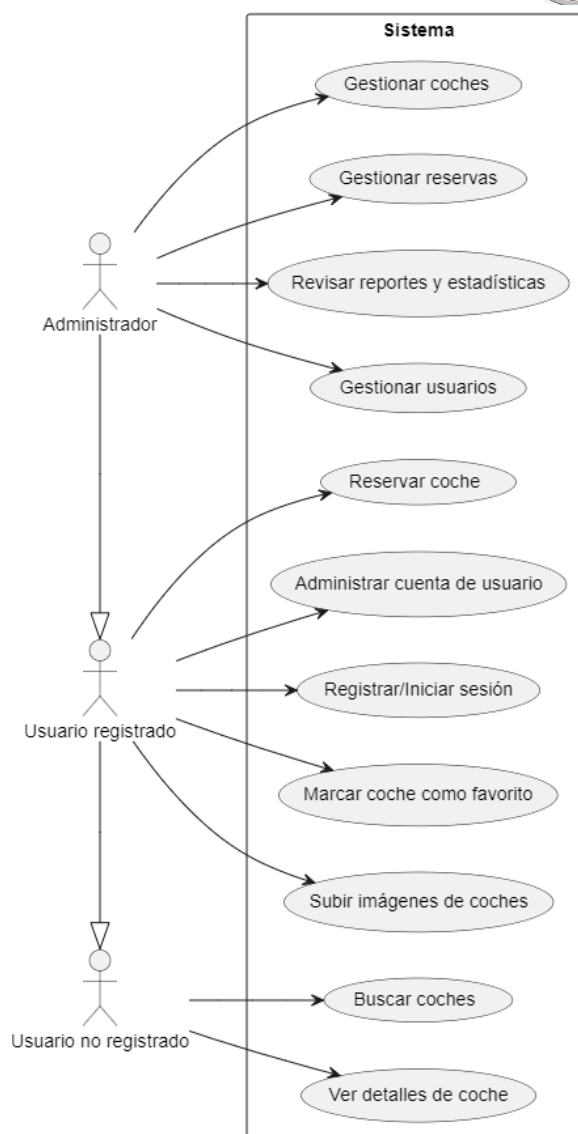


Diagrama casos de uso:





## 3.3. Pruebas realizadas

Se ha comprobado la funcionalidad de la aplicación tanto en local como en remoto haciendo uso de mensajes iterativos en el terminal, mostrando que datos se pasaban, hacia donde, y debido a qué.

## 4. Proceso de despliegue

Para desplegar la aplicación es necesario tener lo siguiente:

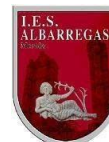
Una cuenta de Oracle Cloud y una instancia de máquina virtual. Véase esta [documentación](#).

Un dominio

Con esto listo, el primer paso será crear una instancia Para ello nos dirigimos a instancias en Oracle Cloud:

# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

Home

Compute

Storage

Home

Pinned

Instances Compute

Virtual cloud networks Networking

Crear instancia:

Compute

Overview

Instances

## Instances *in* dindias (root) *compartment*

An [instance](#) is a compute host. Choose between virtual machines (VMs) and bare

Create instance

Actions

Introducimos un nombre para dicha instancia, una potencia para dicha máquina virtual y descargamos la clave privada y pública para conectar mas adelante por SSH.

### Add SSH keys

Generate an [SSH key pair](#) to connect to the instance using a Secure Shell (SSH) connection, or upload a public key that you already have.

☒ Generate a key pair for me

☐ Upload public key files (.pub)

☐ Paste public keys

☐ No SSH keys



Download the private key so that you can connect to the instance using SSH. It will not be shown again.

↓ Save private key

↓ [Save public key](#)

Tras esto creamos la máquina virtual:

Create

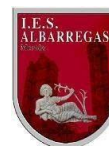
Save as stack

[Cancel](#)

Aparecerá lo siguiente en el apartado instancias ahora:

# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

<div>Create instance</div> <div>Actions ▾</div>		
<input type="checkbox"/>	<b>Name</b>	<b>State</b>
<input type="checkbox"/>	<a href="#">dindias VM</a>	<span>●</span> Running
0 selected		

Tras esperar un rato que la máquina se cree. Hacemos click encima del nombre de esta y copiamos la “Public IP address” en la ventana “Instance Information”.

## Instance access

You [connect to a running Linux instance](#) us

Public IP address: ██████████ [Copy](#)

El siguiente paso es abrir un terminal e introducir el siguiente comando:

```
ssh -i .\ssh-key-AquiTuArchivoDeClaves.key opc@TuIPpublica
```

Una vez realizado este paso ya estarás dentro de tu máquina virtual de Oracle. El siguiente paso es añadir el dominio web para poder acceder mas adelante.

En el menú, networking hacemos click en “Overview” dentro de DNS management:

<b>Networking</b>	
<b>Overview</b>	<b>DNS management</b>
<b>Virtual cloud networks</b>	Overview
	Zones

Ahora click en “Zones”:

## Manage DNS services

### Zones

DNS zones hold the trusted DNS records that will reside on Oracle Cloud Infrastructure's nameservers. You can create public zones with publicly available domain names or private zones to define your own domain name for private address resolution.

Y creamos una zona:

DNS management

Overview

Zones

Traffic management steering policies

### Zones in dindias (root) *Compartment*

A DNS zone holds the trusted DNS records that will reside on Oracle Cloud Infrastructure's nameservers.

Public zones


Private zones

Create zone

Delete

Dentro de esta zona añadiremos el nombre del dominio, en mi caso “hispania-ev.es”:


## Create public zone




You can only view or manage a zone when working in the region where it was

Method

☒ Manual ☐ Import

Zone type 


Primary

Zone name 

hispania-ev.es

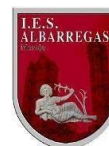
Create in compartment

dindias (root)

 [Show advanced options](#)

# Proyecto “Desarrollo de Aplicaciones Web”

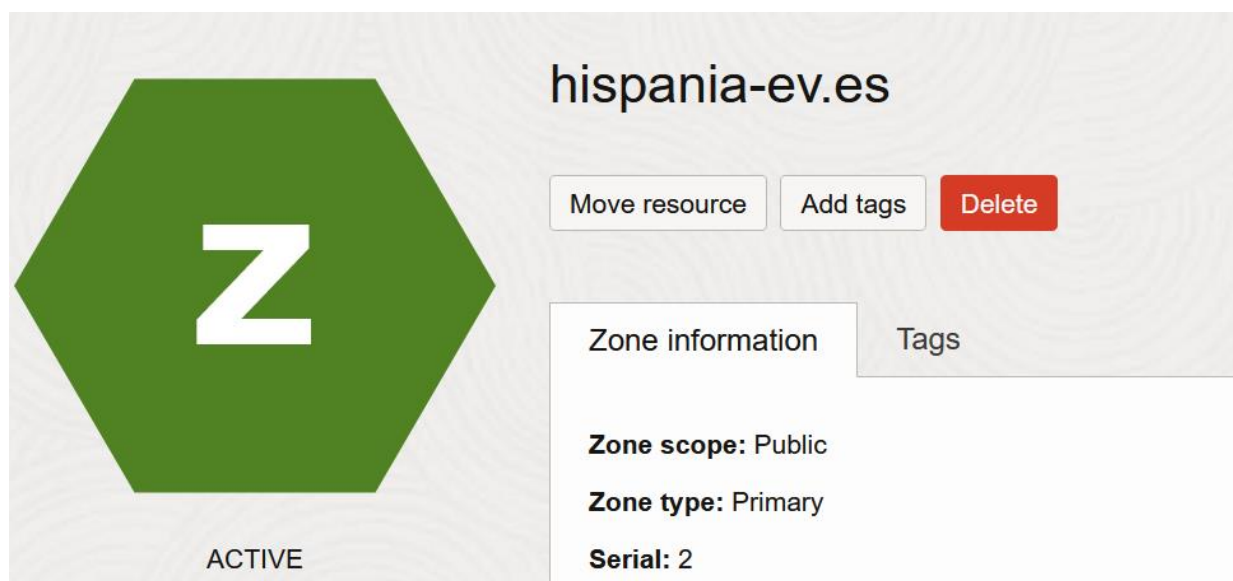
Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

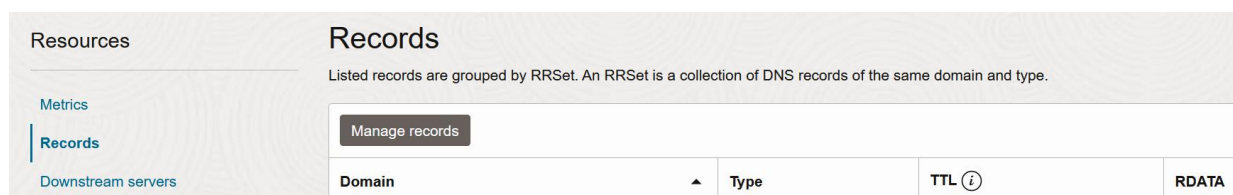
Entonces se creará, el proceso es parecido al de la máquina virtual:



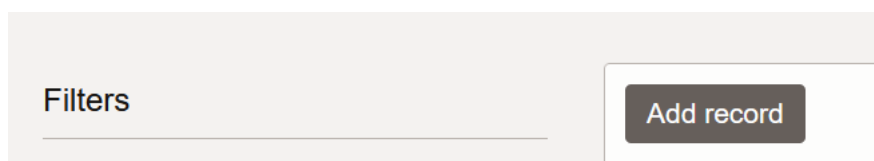
El siguiente paso es coger las 4 url en el apartado “Nameservers” e irte a la página donde hayas comprado el dominio para introducir la información de los nameservers, esto hará que la url de tu dominio apunte a las dns que proporcionan el acceso.

Ahora en el apartado records debemos recoger la conexión enviada por tu dominio para enrutarla a la IP pública de tu máquina virtual:

Click en “Manage records”:



Add record:



El primer record debe ser A – IPv4 address, este recoge los datos directamente desde tu dominio:

## Record information

Name *Optional*

.hispania-ev.es

Type

A - IPv4 address



Host record, used to point a hostname to an IPv4 address.

TTL in seconds

14400

Aquí añadimos nuestra IP pública de nuestra instancia de máquina virtual:

## RDATA/Answers

RDATA mode



Basic

Address



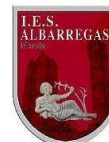
+ Another record



Guardamos los cambios y volvemos a añadir otro record pero esta vez de tipo CNAME:

# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

## Record information

Name *Optional*

.hispania-ev.es

Type

CNAME - CNAME

A canonical name (CNAME) is a special type of DNS record used to create an alias from one domain to another. It's used to identify the CNAME of the domain. For example, a CNAME record at www.example.com with data "example.com" means that someone accessing www.example.com will be pointed to the same IP address that example.com points to. In the event your IP address changes, you only have to update example.com's entry, and www.example.com automatically points to the right place.

TTL in seconds

3600

Ahora en RDATA/Answers en vez de apuntar a nuestra IP pública, apuntaremos a nuestro dominio

## RDATA/Answers ⓘ

RDATA mode

☒ Basic

Target

+ Another record

Añadimos este record también, publicamos y confirmamos los cambios:

Publish changes

Discard changes

Ahora podremos acceder a nuestro servidor mediante el dominio. Se tarda aproximadamente 24 horas en cachear las dns por los servidores por lo que no podemos

# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

acceder instantáneamente.

El siguiente paso es instalar LAMP en nuestra máquina virtual. Tras haberlo hecho nos dirigiremos a la carpeta /var/www y clonaremos la aplicación:

```
[opc@dindias-vm ~]$ cd /var/www
[opc@dindias-vm www]$ sudo su
[root@dindias-vm www]# git clone https://github.com/dindias/proyecto_ev.git
```

Movemos la base de datos a mysql:

```
[root@dindias-vm www]# mysql -u root -p proyecto-ev < proyecto_ev.sql
```

Y configuramos el virtualhost existente para que el DirectoryRoot sea la carpeta de nuestra aplicación web:

```
DocumentRoot "/var/www/proyecto_ev"
```

Reiniciamos HTTPD y MySQL y cargamos el dominio, en este caso hispania-ev.es, o a falta de esto su IP pública: 143.47.57.201

## 5. Propuestas de mejoras

Originalmente la aplicación estaría adaptada para personas con visibilidad reducida o diferentes afectaciones a la vista, entre otras cosas. El abarque de la aplicación resulta ser mas grande de lo esperado por lo que las mejoras a realizar son las siguientes:

- Modo día/noche
- Modo daltónicos
- Selección de colores de contraste personalizados para menús
- Sistema de mensajería, aprovechando el sistema de notificaciones existente.
- Perfil de usuario, similar al de una red social
- Sistema de comentarios y puntuación en vehículos y perfiles

## 6. Bibliografía

[Stackoverflow](#): Resolución de problemas.

[Oracle Cloud](#): Documentacion para poner en marcha la IP pública.

[Bootstrap docs](#): Documentacion para el uso y manejo de Bootstrap.

[PHP Documentation](#): Funciones y uso.

[SCSS](#)



# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



Consejería de Educación y Empleo

[Charts.js](#): Herramienta para hacer gráficos con datos.

[Litepicker](#): Seleccionador de fechas