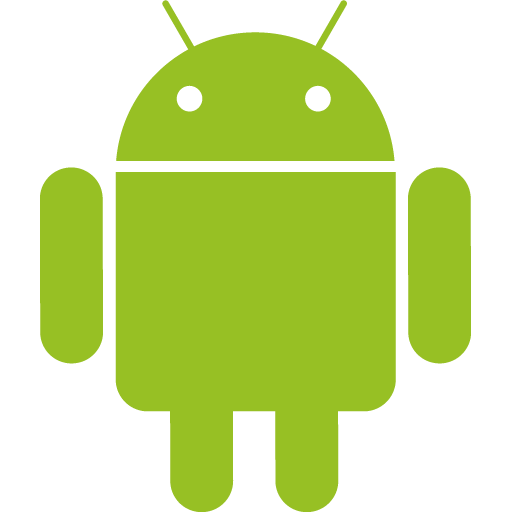
18-11-2016

Luis Enrique Sosa Hernández

Tecnológico nacional de méxico, INSTITUTO tecnológico de veracruz

APLICACIÓN ANDROID Pokemon

Registrar, Actualizar, Mostrar, Eliminar registros de una base de datos mysql



Fundamentos de bases de datos 7-8 am

ISC. Ezequiel Piña Ortiz

Contenido

[Introducción 2](#_Toc467253937)

[Diseño 3](#_Toc467253938)

[**Screen1** 3](#_Toc467253939)

[**Componentes** 3](#_Toc467253940)

[**Registro** 5](#_Toc467253941)

[**Componentes** 5](#_Toc467253942)

[**Actualizar** 8](#_Toc467253943)

[**Componentes** 8](#_Toc467253944)

[**Mostrar** 11](#_Toc467253945)

[**Componentes** 11](#_Toc467253946)

[**Eliminar** 13](#_Toc467253947)

[**Componentes** 13](#_Toc467253948)

[**Base de datos** 15](#_Toc467253949)

# Introducción

En el presente manual, se enlistan y se muestran imágenes de las pantallas y componentes todos los componentes de la aplicación Pokemon.

El manual pretende dar a conocer el funcionamiento de la aplicación, y los elementos claves, así como la forma en la que se creó, con ayuda de imágenes y referencias a las imágenes de los bloques de código.

La aplicación fue creada con la aplicación web Mit App inventor y probada en un dispositivo android con versión de android 5.1 lollipop, también fue probada con el emulador de android que ofrece la plataforma, la cual requiere preinstalar una aplicación en el ordenador llamada aiStarter.

La base de datos está montada en un servidor web, por lo tanto la aplicación funciona de manera remota, conectándose a servidor web, y accediendo a archivos .php que realizan las acciones de insertar, actualizar, eliminar y mostrar registros de la base de datos.

La aplicación permite crear una pokedex., ingresando, actualizando, eliminando y mostrando registros.

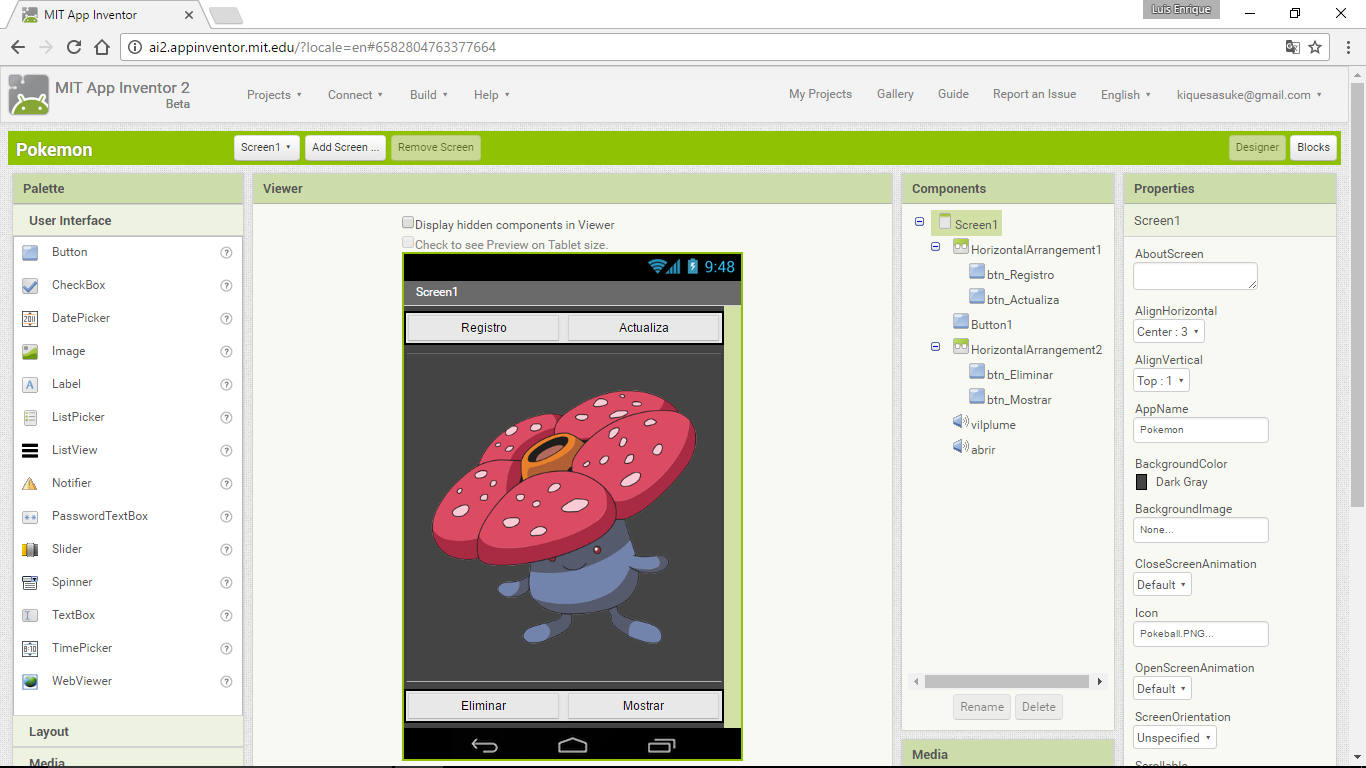


Imagen 1(Interfaz mit App inventor)

# Diseño

## **Screen1**

### **Componentes**

**Screen1**: Pantalla principal de la aplicación, menú de la aplicación donde se proporcionan enlaces a las demás opciones por medio de botones.

Con el evento Initialize la aplicación oculta al iniciar, la barra de estado y el título de la ventana.

**HorizontalArrangement1**: Layout que permite dar formato a los componentes que estén dentro.

**btn\_Registro**: Botón que redirecciona a una nueva pantalla llamada Registro .Contiene el texto (“Registro”).

Con el evento click, hace un llamado al audio abrir, que reproduce un sonido, y abre una nueva ventana llamada Registro.

**btn\_Actualiza**: Botón que redirecciona a una nueva pantalla llamada Actualizar. Contiene el texto (“Actualiza”).

Con el evento click, hace un llamado al audio abrir, que reproduce un sonido, y abre una nueva ventana llamada Actualizar.

**Button1**: Botón que permite acceder a un componente de audio. El botón cuenta con una imagen de fondo.

Con el evento click, hace un llamado al audio vilplume, que reproduce un sonido.

**HorizontalArrangement2**: Layout que permite dar formato a los componentes que estén dentro.

**btn\_Eliminar**: Botón que redirecciona a una nueva pantalla llamada Eliminar. Contiene el texto (“Eliminar”).

Con el evento click, hace un llamado al audio abrir, que reproduce un sonido, y abre una nueva ventana llamada Eliminar.

**btn\_Mostrar**: Botón que redirecciona a una nueva pantalla llamada Mostrar. Contiene el texto (“Mostrar”).

Con el evento click, hace un llamado al audio abrir, que reproduce un sonido, y abre una nueva ventana llamada Mostrar.

**vilplume**: Componente de audio que permite reproducir un sonido.

**abrir**: Componente de audio que permite reproducir un sonido.



Imagen 2(Interfaz Screen1)

**Nota**: Los Eventos están relacionados en la sección blocks para su visualización se anexaron imágenes de los bloques en la carpeta de Imágenes(Blocks)/ Blocks Screen1.png.

## **Registro**

### **Componentes**

**Registro**: La pantalla Registro, permite añadir registros a una base de datos, por medio de componente web. Con el evento Initialize la aplicación oculta al iniciar, la barra de estado y el título de la ventana.

**TextBox\_Nombre**: Campo de texto que permite hacer uso del teclado y escribir texto dentro. Dentro del tiene un Hint con el texto (“Nombre”).

**TextBox\_Tipo**: Campo de texto que permite hacer uso del teclado y escribir texto dentro. Dentro contiene un Hint con el texto (“Tipo”).

**TextBox\_Region**: Campo de texto que permite hacer uso del teclado y escribir texto dentro. Dentro contiene un Hint con el texto (“Región”).

**TextBox\_Generacion**: Campo de texto que permite hacer uso del teclado y escribir texto dentro. Dentro contiene un Hint con el texto (“Generación”), Este campo debe ser numérico.

**HorizontalArrangement1**: Layout que permite dar formato a los componentes que estén dentro.

**btn\_Registrar**: Contiene el texto (“Registrar”).

Cuando se acciona el evento click, se pasa a las instructivas de condición if, then else.

Dentro del if está la condición is empty, que verifica si alguno de los campos de TextBox se encuentra vacío, si la condición se cumple, hace un llamado a al componente de sonido falto y lo reproduce, suelta una notificación que con el texto (“No pueden quedar campos vacíos”), y por ultimo esconde el teclado con el método Hide.

Se hace uso de variables, en específico Initialize global to, que junto con un join y campos de texto permiten guardar la dirección del servidor web, para su posterior uso.

Si la condición no se cumple (“else”), cambia el valor de la url de Web1, por la variable global declara anteriormente y con un text se concatena con registrar.php, donde se encuentra el archivo .php que contiene las instrucciones necesarias para insertar registros en una base de datos.

Hace un llamado a Web1 y con el método PostTexttext, y join, se envían los parámetros de los TextBox.

Con el método GotText, se recupera lo que contiene la página Web1, se usan las condicionales if, then else.

If: Se hace uso del método containst text, piece, para verificar que el contenido recuperado concuerde. Si la condición se cumple, se llama a Notifier1, que muestra una alerta con el texto (“Pokemón registrado!”), se esconde el teclado, se reproduce Sound2 y se cierra la ventana.

else: se muestra una alerta con el texto (“Ocurrió un error”), se esconde el teclado, y se reproduce Sound1.

**btn\_Cancelar**: Contiene el texto (“Regresar”).

Con el evento click, hace un llamado al audio cerrar, que reproduce un sonido. Y a su vez hace un llamado al método close screen que cierra la pantalla.

**Button1**: Botón que permite acceder a un componente de audio. El botón cuenta con una imagen de fondo.

Con el evento click, hace un llamado al audio gliglar, que reproduce un sonido.

**Notifier1**: Componente que permite emitir una notificación cuando sucede algún evento en específico.

**Web1**: Componente que permite hacer uso de archivos web, como archivos php.

**Sound1**: Componente de audio que permite reproducir un sonido.

**Sound2**: Componente de audio que permite reproducir un sonido.

**gliglar**: Componente de audio que permite reproducir un sonido.

**cerrar**: Componente de audio que permite reproducir un sonido.

**falto**: Componente de audio que permite reproducir un sonido.

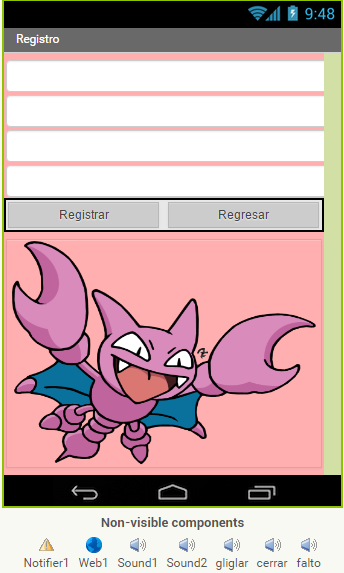


Imagen 3(Interfaz Registro)

**Nota**: Los Eventos están relacionados en la sección blocks para su visualización se anexaron imágenes de los bloques en la carpeta de Imágenes(Blocks)/ Blocks Registro.png.

El código php se encuentra en la carpeta Codigophp/registrar.php.

## **Actualizar**

## **Componentes**

**Actualizar**: Registro: La pantalla Actualizar, permite actualizar registros a una base de datos, por medio de componente web. Con el evento Initialize la aplicación oculta al iniciar, la barra de estado y el título de la ventana.

**Txt\_nombre**: Campo de texto que permite hacer uso del teclado y escribir texto dentro. Dentro del tiene un Hint con el texto (“Nombre”).

**txt\_tipo**: Campo de texto que permite hacer uso del teclado y escribir texto dentro. Dentro contiene un Hint con el texto (“Tipo”).

**txt\_region**: Campo de texto que permite hacer uso del teclado y escribir texto dentro. Dentro contiene un Hint con el texto (“Región”).

**txt\_generacion**: Campo de texto que permite hacer uso del teclado y escribir texto dentro. Dentro contiene un Hint con el texto (“Generación”), Este campo debe ser numérico.

**HorizontalArrangement1**: Layout que permite dar formato a los componentes que estén dentro.

**btn\_actualizar**: Contiene el texto (“Registrar”).

Cuando se acciona el evento click, se pasa a las instructivas de condición if, then else.

Dentro del if está la condición is empty, que verifica que el campo Txt\_nombre se encuentra vacío, si la condición se cumple, hace un llamado a al componente de sonido falto y lo reproduce, suelta una notificación que con el texto (“Necesita ingresar un nombre”), y por ultimo esconde el teclado con el método Hide.

Se hace uso de variables, en específico Initialize global to, que junto con un join y campos de texto permiten guardar la dirección del servidor web, para su posterior uso.

Si la condición no se cumple (“else”), cambia el valor de la url de Web1, por la variable global declara anteriormente y con un text se concatena con actualizar.php, donde se encuentra el archivo .php que contiene las instrucciones necesarias para actualizar registros en una base de datos.

Hace un llamado a Web1 y con el método PostTexttext, y join, se envían los parámetros de los TextBox.

Con el método GotText, se recupera lo que contiene la página Web1, se usan las condicionales if, then else.

If: Se hace uso del método containst text, piece, para verificar que el contenido recuperado concuerde. Si la condición se cumple, se llama a Notifier1, que muestra una alerta con el texto (“Pokemón Actualizado!”), se esconde el teclado, se reproduce Sound2 y se cierra la ventana.

else: se muestra una alerta con el texto (“Ocurrió un error”), se esconde el teclado, y se reproduce Sound1.

**btn\_cancelar**: Contiene el texto (“Regresar”).

Con el evento click, hace un llamado al audio cerrar, que reproduce un sonido. Y a su vez hace un llamado al método close screen que cierra la pantalla.

**Button1**: Botón que permite acceder a un componente de audio. El botón cuenta con una imagen de fondo.

Con el evento click, hace un llamado al audio Hitmo, que reproduce un sonido.

**Notifier1**: Componente que permite emitir una notificación cuando sucede algún evento en específico.

**Web1**: Componente que permite hacer uso de archivos web, como archivos php.

**Sound1**: Componente de audio que permite reproducir un sonido.

**Sound2**: Componente de audio que permite reproducir un sonido.

**Hitmo**: Componente de audio que permite reproducir un sonido.

**cerrar**: Componente de audio que permite reproducir un sonido.

**falto**: Componente de audio que permite reproducir un sonido.



Imagen 4(Interfaz Actualizar)

**Nota**: Los Eventos están relacionados en la sección blocks para su visualización se anexaron imágenes de los bloques en la carpeta de Imágenes(Blocks)/ Blocks Actualizar.png.

El código php se encuentra en la carpeta Codigophp/actualizar.php.

## **Mostrar**

## **Componentes**

**Mostrar**: La pantalla Mostrar, permite visualizar registros de una base de datos, por medio de componente web. Con el evento Initialize la aplicación oculta al iniciar, la barra de estado y el título de la ventana. Cuando se acciona el evento initialize de la pantalla mostrar, cambia el valor de la url de Web1, por la variable global declara anteriormente y con un text se concatena con mostrar.php donde se encuentra el archivo .php que contiene las instrucciones necesarias para mostrar registros en una base de datos y hace un llamado con el método call a Web1.Get,esto con el propósito de llamar al método web1 previamente.

**HorizontalArrangement1**: Layout que permite dar formato a los componentes que estén dentro.

**btn\_cancelar**: Contiene el texto (“Regresar”).

Con el evento click, hace un llamado al audio cerrar, que reproduce un sonido. Y a su vez hace un llamado al método close screen que cierra la pantalla.

**Lista**: Se hace uso de variables, en específico Initialize global to, que junto con un join y campos de texto permiten guardar la dirección del servidor web, para su posterior uso.

Con el evento Before Picking del componente lista, cambia el valor de la url de Web1, por la variable global declara anteriormente y con un text se concatena con mostrar.php donde se encuentra el archivo .php que contiene las instrucciones necesarias para mostrar registros en una base de datos y hace un llamado con el método call a Web1.Get.

Con el método GotText, se recupera lo que contiene la página Web1, hace un llamado a Web1 y con el método PostTexttext, y join, se envían los parámetros del txt\_nombre, si es diferente de null , dentro del código php se realiza una consulta para mostrar ese único registro.

Se hace uso de variables, en específico Initialize global to, y se concatena con un text vacío.

Con la variable global separador to, se reemplaza el texto recuperado, para darle un formato adecuado.

Con el método set to, se llena la lista con los elementos string recuperados con el método get de la variable global separador.

**txt\_Nombre**: Campo de texto que permite hacer uso del teclado y escribir texto dentro. Dentro del tiene un Hint con el texto (“Nombre”).

**Button1**: Botón que permite acceder a un componente de audio. El botón cuenta con una imagen de fondo.

Con el evento click, hace un llamado al audio Sound1, que reproduce un sonido.

**Web1**: Componente que permite hacer uso de archivos web, como archivos php.

**Sound1**: Componente de audio que permite reproducir un sonido.

**cerrar**: Componente de audio que permite reproducir un sonido.

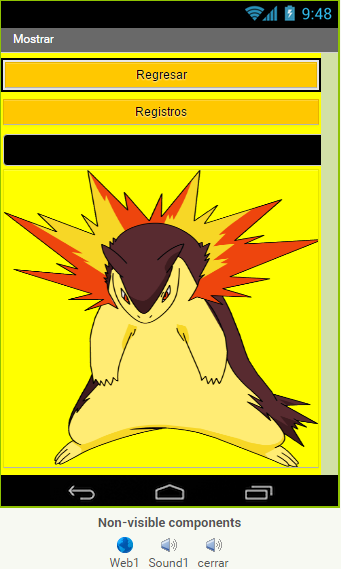


Imagen 5(Interfaz Mostrar)

**Nota**: Los Eventos están relacionados en la sección blocks para su visualización se anexaron imágenes de los bloques en la carpeta de Imágenes(Blocks)/ Blocks Mostrar.png.

El código php se encuentra en la carpeta Codigophp/mostrar.php.

## **Eliminar**

### **Componentes**

**Eliminar:** La pantalla Eliminar, permite eliminar registros de una base de datos, por medio de componente web. Con el evento Initialize la aplicación oculta al iniciar, la barra de estado y el título de la ventana.

**Etiqueta**: Componente que permite mostrar un texto en pantalla contiene (“Pokemon a eliminar”).

**btn\_cancelar**: Contiene el texto (“Regresar”).

Con el evento click, hace un llamado al audio cerrar, que reproduce un sonido. Y a su vez hace un llamado al método close screen que cierra la pantalla.

**txt\_Name**: Campo de texto que permite hacer uso del teclado y escribir texto dentro. Dentro del tiene un Hint con el texto (“Nombre”).

**btn\_Eliminar**: Contiene una imagen de fondo.

Cuando se acciona el evento click, se pasa a las instructivas de condición if, then else.

Dentro del if está la condición is empty, que verifica que el campo Txt\_nombre se encuentra vacío, si la condición se cumple, hace un llamado a al componente de sonido falto y lo reproduce, suelta una notificación que con el texto (“No pueden quedar campos vacíos”), y por ultimo esconde el teclado con el método Hide.

Se hace uso de variables, en específico Initialize global to, que junto con un join y campos de texto permiten guardar la dirección del servidor web, para su posterior uso.

Si la condición no se cumple (“else”), cambia el valor de la url de Web1, por la variable global declara anteriormente y con un text se concatena con eliminar.php, donde se encuentra el archivo .php que contiene las instrucciones necesarias para eliminar registros en una base de datos.

Hace un llamado a Web1 y con el método PostTexttext, y join, se envían el parámetro txt\_Name.

Con el método GotText, se recupera lo que contiene la página Web1, se usan las condicionales if, then else.

If: Se hace uso del método containst text, piece, para verificar que el contenido recuperado concuerde. Si la condición se cumple, se llama a Notifier1, que muestra una alerta con el texto (“Pokemón Eliminado!”), se esconde el teclado, se reproduce Sound2 y se cierra la ventana.

else: se muestra una alerta con el texto (“Ocurrió un error”), se esconde el teclado, y se reproduce Sound1.

**Web1**: Componente que permite hacer uso de archivos web, como archivos php.

**Notifier1**: Componente que permite emitir una notificación cuando sucede algún evento en específico.

**Sound1**: Componente de audio que permite reproducir un sonido.

**Sound2**: Componente de audio que permite reproducir un sonido.

**cerrar**: Componente de audio que permite reproducir un sonido.

**falto**: Componente de audio que permite reproducir un sonido.

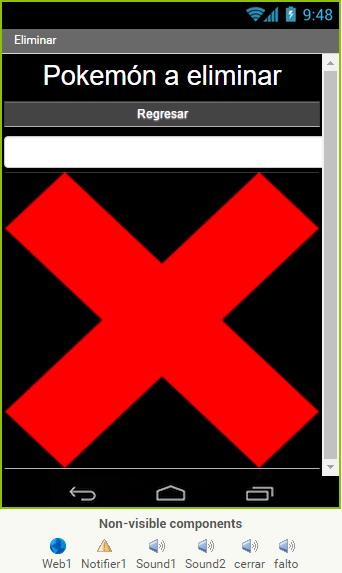


Imagen 6(Interfaz Eliminar)

**Nota**: Los Eventos están relacionados en la sección blocks para su visualización se anexaron imágenes de los bloques en la carpeta de Imágenes(Blocks)/ Blocks Eliminar.png.

El código php se encuentra en la carpeta Codigophp/eliminar.php.

En algunas partes de la descripción de los bloques no se nombran algunos métodos debido a que ya se habían mencionado con anterioridad.

## **Base de datos**

La base de datos pokemon usada para la aplicación, se encuentra en un servidor web, fue diseñada con la herramienta PhpMyadmin

La base de datos contiene una única tabla llamada pokemon con los siguientes campos:

**Nombre**: Varchar de longitud 30 donde se guarda el nombre del pokemon registrado, este campo es la llave primaria , elegido así con fines del propósito general de una pokedex, donde solo se puede tener registrado un pokemon , y dos pokemon no pueden llevar el mismo nombre, porque no hay variantes de sus demás atributos, si se quisiera tener un registro de pokemon con características propias , se tendría que asignar un ID auto incrementable a cada registro, ya que dos pokemon del mismo nombre pueden tener características propias como peso y altura.

**Tipo**: Varchar de longitud 40 donde se guarda el tipo del pokemon , ejemplo, tipo fantasma, sin embargo, un pokemon puede tener más de un tipo “Fuego/fantasma” por lo tanto se necesita un tamaño considerable para guardar el dato.

**Region**: Varchar de longitud 40 donde se almacena la región de donde proviene el pokemon.

**Generacion**: Este campo es de valor entero, donde se almacena la generación en la que fue implementado el pokemon.

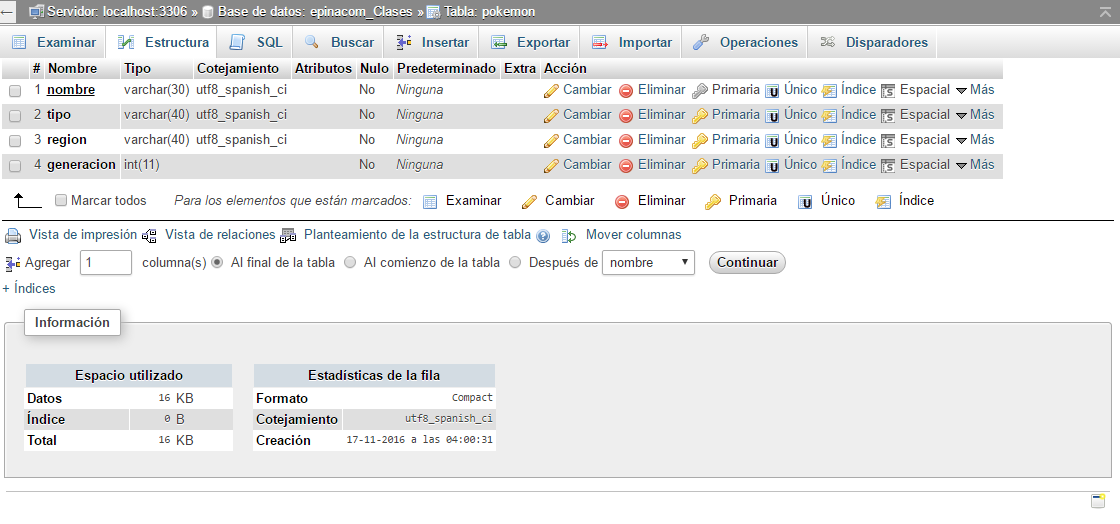


Imagen 7(base de datos en el servidor)