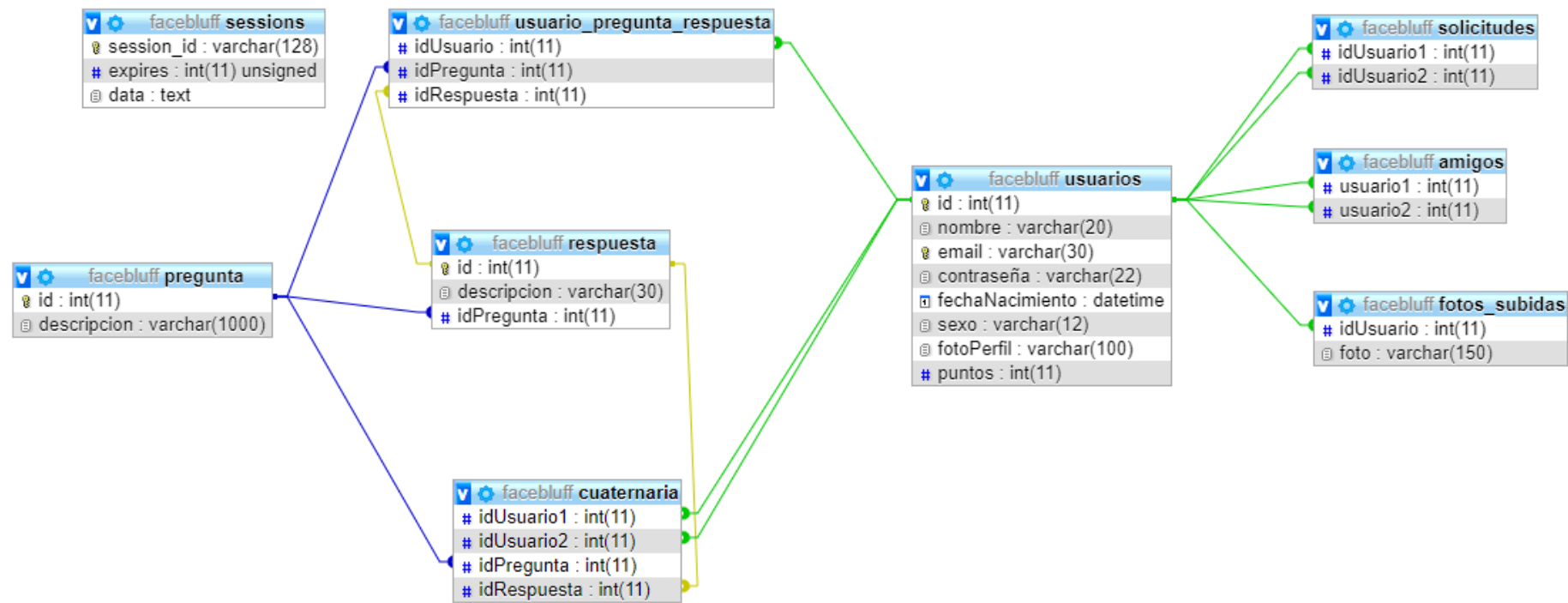


# FACEBLUFF

FRANCISCO RAFAEL GARCÍA ROFES



LUIS ANTONIO ROJAS RAMIREZ

# DISEÑO DE LA BASE DE DATOS: Modelo Entidad-Relación



# DISEÑO DE LA BASE DE DATOS

## Modelo Relacional

- ▶ -Usuario(IdUsuario,Email,contraseña,fotoPerfil,fechaNacimiento,puntos)
- ▶ -Relacion amistad entre usuarios (IdUsuario1,IdUsuario2)
- ▶ -Relacion solicitud entre usuarios(IdUsuario1,IdUsuario2)
- ▶ -Relación entre preguntas , usuario y respuesta(IdUsuario,idpregunta,idRespuesta)
- ▶ -Relación entre preguntas , usuario1,usuario2 y respuesta(IdUsuario1,idUsuario2,idpregunta,idRespuesta)
- ▶ -Respuestas(idRespuesta,idPregunta,descripcion)
- ▶ -Tabla Preguntas (idPregunta,descripcion)
- ▶ -Tabla Fotos subidas(idUsuario,foto)
- ▶  nombre relación  Foreign Key

# ESTRUCTURA DE LA APLICACIÓN

## Explicación

- ▶ Hemos dividido la aplicación en 3 elementos:
  - ▶ Routers
  - ▶ Controlador
  - ▶ Modelos

# ESTRUCTURA DE LA APLICACIÓN

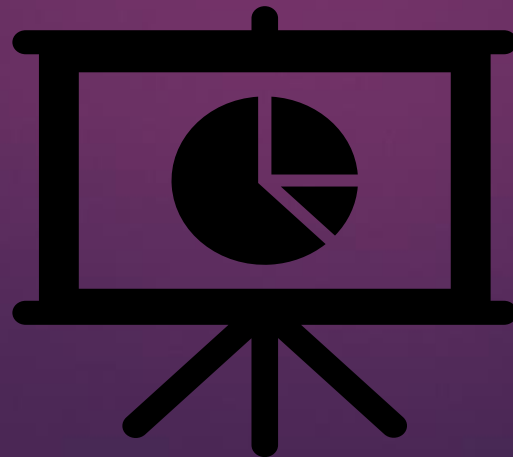
## Routers

- ▶ Hemos creado 3 routers en función del origen de la petición:
  - ▶ RouterUsuarios : es el encargado de gestionar todas las URL provenientes de lo relacionado con usuario.
  - ▶ RouterRespuestas : es el encargado de gestionar todas las URL provenientes de la parte de respuestas
  - ▶ RouterPreguntas : es el encargado de gestionar las URL relacionadas con preguntas.

# ESTRUCTURA DE LA APLICACIÓN

## Controlador

- ▶ Se encarga de almacenar todas la funciones que reciben los routers, con esto se consigue que la aplicación este ordenada y se facilite el uso a cualquier tipo de arreglo o mantenimiento futuro.



# ESTRUCTURA DE LA APLICACIÓN

## Modelo

- ▶ En la parte de modelo al igual que en los routers lo hemos dividido según su origen en 5 modelos:
  - ▶ Amigos
  - ▶ Preguntas
  - ▶ Respuestas
  - ▶ Solicitudes
  - ▶ Usuarios

# RUTAS GESTIONADAS

## Router usuarios

- ▶ Las rutas que gestionamos en este módulo son las siguientes:
  - ▶ Rutas relacionadas con el login , el log out...
  - ▶ Formularios tanto de creación como de modificación y sus respectivos procesamientos de datos
  - ▶ Aceptar , rechazar y solicitar amistad.
  - ▶ Subida de fotos asociadas a los puntos de un usuarios.



# RUTAS GESTIONADAS

## Router preguntas

- ▶ En este otro modulo respondemos a :
  - ▶ El visionado de las 5 preguntas aleatorias
  - ▶ Creación de las nuevas preguntas
  - ▶ Toda la capa de procesamiento de las citadas anteriormente

# RUTAS GESTIONADAS

## Router respuestas

- ▶ Este router es el más escueto de los que tenemos ya que solo se encarga de añadir respuestas , adivinar la respuesta de otro usuario y de su procesamiento



# Implementación de la sesión de un usuario

- ▶ En esta parte hay que prestar especial atención a la actualización del atributo sessions dentro de request ya que cualquier fallo desencadenaría que el usuario actual perdiera los datos que lleva mientras use la página, así como acordarse de borrarlo cuando hagamos la desconexión.



# Implementación de la sesión de un usuario

- ▶ Actualizaremos nuestro id de usuario cuando:
  - ▶ Entremos en la página web a través del login.
  - ▶ Nos desconectemos de la página.
  - ▶ Nos creamos un nuevo usuario.



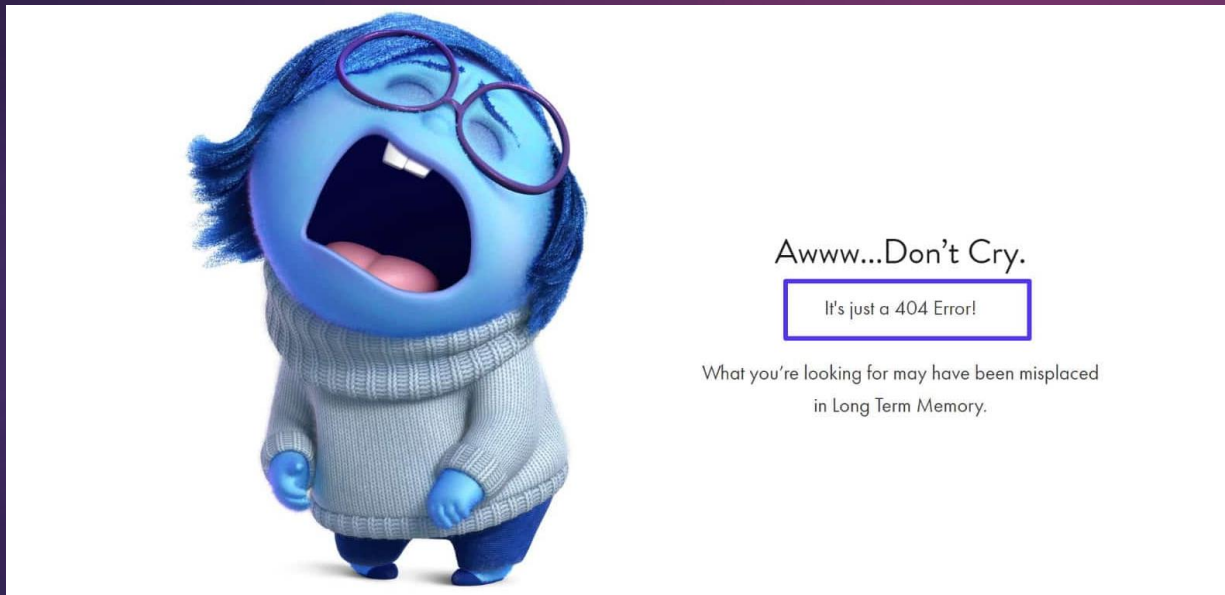
# RESTRICCIÓN PARA USUARIOS NO LOGEADOS

- ▶ Este tema lo gestionamos a través de un middleware que comprueba si el id del usuario está definido o en caso de que lo este ,cuál es su valor.
- ▶ El middleware debe ser colocado después de rutas como login o el formulario pero antes de las rutas que dan acceso a la aplicación web para que si se intenta acceder a esas rutas se te redirija al login obligándote a iniciar sesión o a crearte un usuario.



# Errores habituales

- ▶ Dos errores con los que nos vamos a enfrentar en cualquier página web van a ser el error 404 y el error 500.



# Error 404

- ▶ Es un error que se da cuenta se intenta acceder a una ruta que no estamos gestionando en ese momento.
- ▶ Lo que hacemos en este caso es usar un middleware que coge cualquier ruta que no haya sido capturada por la app y la redirige a una pagina de error 404, mostrando al usuario que ha ocurrido un error con su petición



# ERROR 500

- ▶ Este error se da cuando ha ocurrido algún error interno en el servidor.
- ▶ Si un usuario al hacer una petición hay un fallo, o cuando se hace alguna consulta en la base datos ocurre un error le redirigimos mediante un middleware a la página de error 500 donde se le notifica que ha habido un interno y que la página no va a estar disponible.