

API de Usuarios

	Llamada	Uso
	GET /usuarios/listar <hr/> Devuelve todos los usuarios de la base de datos en un vector.	Parametros -> XXX Devuelve -> Vector <hr/> OK -> Devuelve un vector con el listado de los usuarios ERROR 1 -> Error al acceder a la base de datos
	GET /usuarios/recuperar_usuario_por_id <hr/> Devuelve el usuario con la id pasada por parametro.	Parametros -> _id Devuelve -> json con todos los datos del usuario <hr/> OK -> Devuelve un vector con el listado de los usuarios ERROR 1 -> Error al acceder a la base de datos
	GET /usuarios/recuperar_usuario_por_usuario <hr/> Devuelve todos los datos de usuario con el nombre de usuario pasado por parametro	Parametros -> usuario Devuelve -> json con todos los datos del usuario <hr/> OK -> Devuelve un vector con el listado de los usuarios ERROR 1 -> Error al acceder a la base de datos ERROR 2 -> Usuario no encontrado
	GET /usuarios/recuperar_nombre_por_usuario <hr/> Devuelve el nombre del usuario pasado por parametro	Parametros -> usuario Devuelve -> nombre (String) <hr/> OK -> Devuelve el nombre del usuario pasado por parametro ERROR 1 -> Error al acceder a la base de datos ERROR 2 -> Usuario no encontrado
	GET /usuarios/comprobar_usuario <hr/> Busca el usuario pasado por parametro en la base de datos	Parametros -> usuario Devuelve -> 0 (Usuario encontrado) ///// 1 (Usuario no encontrado) <hr/> OK 0 -> Usuario encontrado OK 1 -> Usuario no encontrado ERROR 2 -> Error al acceder a la base de datos
	POST /usuarios/guardar <hr/> Guarda el usuario con los datos pasados por parametro	Parametros -> usuario,pass,nombre,apellidos,email,telefono,ciudad,esguia,_id(opc) Devuelve -> 0 (Todo OK) <hr/> OK 0 -> Usuario guardado correctamente ERROR 1 -> Error al guardar nuevo usuario ERROR 2 -> Error al buscar usuario por la _id ERROR 3 -> Error al modificar el usuario
	POST /usuarios/eliminar	Parametros -> _id

	<p>Elimina el usuario de la base de datos con el <code>_id</code> pasado por parametro</p>	<p>Devuelve -> 0 (Todo OK)</p> <hr/> <p>OK 0 -> Usuario guardado correctamente</p> <p>ERROR 1 -> Error al intentar eliminar el usuario</p>
	<p>GET /usuarios/login</p> <hr/> <p>Comprueba el login, si devuelve un objeto con los datos del usuario es correcto</p>	<p>Parametros -> usuario, pass</p> <p>Devuelve -> json con los datos del usuario logeado</p> <hr/> <p>OK -> Devuelve un json con los datos del usuario</p> <p>ERROR 1 -> Error al conectar</p> <p>ERROR 2 -> Usuario no encontrado</p> <p>ERROR 3 -> Contraseña incorrecta</p>
	<p>POST /usuarios/registroUsuario</p> <hr/> <p>Guarda el usuario con los datos pasados por parametro</p>	<p>Parametros -> usuario,pass,nombre,apellidos,email</p> <p>Devuelve -> 0 (Todo OK)</p> <hr/> <p>OK 0 -> Usuario guardado correctamente</p> <p>ERROR 1 -> Error al guardar nuevo usuario</p>
	<p>POST /usuarios/registro_guia</p> <hr/> <p>Realiza el registro de un usuario a guia</p>	<p>Parametros -> usuario</p> <p>Devuelve -> 0 (Todo OK)</p> <hr/> <p>OK 0 -> Usuario pasado a guia correctamente</p> <p>ERROR 1 -> Error al conectar</p> <p>ERROR 2 -> Usuario no encontrado</p> <p>ERROR 3 -> Error al guardar el usuario</p>
	<p>POST /usuarios/modificar_pass</p> <hr/> <p>Cambia la contraseña pasado por parametro del usuario, ademas comprueba si son correctas</p>	<p>Parametros -> usuario, pass1, pass2</p> <p>Devuelve -> 0 (Todo OK)</p> <hr/> <p>OK 0 -> Contraseña modificada correctamente</p> <p>ERROR 1 -> No se han recibido todos los parametros</p> <p>ERROR 2 -> Las contraseñas no coinciden</p> <p>ERROR 3 -> Error al buscar el usuario en la base de datos</p> <p>ERROR 4 -> Error al guardar el usuario</p>
	<p>POST /usuarios/modificarPerfil</p> <hr/> <p>Guarda el usuario con los datos pasados por parametro</p>	<p>Parametros -> usuario,pass,nombre,apellidos,email,telefono,ciudad,_id</p> <p>Devuelve -> 0 (Todo OK)</p> <hr/> <p>OK 0 -> Usuario guardado correctamente</p> <p>ERROR 1 -> Error al guardar nuevo usuario</p> <p>ERROR 2 -> Error al buscar usuario por la <code>_id</code></p> <p>ERROR 3 -> Error al modificar el usuario</p>
	<p>POST /usuarios/subirFotoPerfil</p> <hr/> <p>Se realiza un foto de perfil de la foto subida desde el formnulario</p>	<p>Parametros -> file (Enviado mediante form/data)</p> <p>Devuelve -> Redirección a <code>modificar_perfil.html</code></p> <hr/> <p>OK -> Redirige a la web modificar perfil</p>

		ERROR 1 -> Error al conectar con la base de datos ERROR 2 -> Error al subir la imagen
	POST /usuarios/actualizarVotos <hr/> <p>Actualiza la puntuacion del guia con el nuevo voto introducido en la ruta</p>	Parametros -> usuario, num Devuelve -> 0 (Todo OK) <hr/> OK 0 -> Actualizacion realizada con exito ERROR 1 -> Parametros no recibidos ERROR 2 -> Error al realizar la peticion a la base de datos ERROR 3 -> Datos no encontrados ERROR 4 -> Error al guardar los nuevos datos
	POST /usuarios/modificarSaldo <hr/> <p>Añade o elimina saldo del guia</p>	Parametros -> usuario, num, modo (1-sumar, 2-restar) Devuelve -> 0 (Todo OK) <hr/> OK 0 -> Actualizacion realizada con exito ERROR 1 -> Error al realizar la peticion a la base de datos ERROR 2 -> Datos no encontrados ERROR 3 -> Error al guardar los nuevos datos

API de Rutas

Llamada	Uso
GET /rutas/listar <hr/> <p>Devuelve todas las rutas de la base de datos en un vector</p>	Parametros -> X Devuelve -> Vector <hr/> OK -> Devuelve un vector con el listado de las rutas ERROR 1 -> Error al acceder a la base de datos
GET /rutas/listarRutasPorGuia <hr/> <p>Devuelve todas las rutas de la base de datos en un vector del guia pasado por parametro. Ademàs segun la opcion si es: 1- Devuelve solo las rutas siguiente a la fecha actual. 2- Devuelve solo las rutas anteriores a la fecha actual.</p>	Parametros -> guia, opcion (1 o 2) Devuelve -> Vector <hr/> OK -> Devuelve un vector con el listado de las rutas ERROR 1 -> Parametros no recibidos ERROR 2 -> Error al acceder a la base de datos (opcion==1) ERROR 3 -> Error al acceder a la base de datos (opcion==2)
GET /rutas/busquedaGeo <hr/> <p>Devuelve las rutas mas cercanas al punto indicado a una distancia pasada por parametro (por defecto 5km)</p>	Parametros -> coord, distancia(int) Devuelve -> Vector <hr/> OK -> Devuelve un vector con el listado de las rutas ERROR 1 -> Error al acceder a la base de datos
GET /rutas/recuperar_ruta_por_id <hr/>	Parametros -> _id, opcion Devuelve -> json con todos los datos de la ruta <hr/>

<p>Devuelve todas las rutas de la base de datos en un vector. Si recibe una opcion:</p> <p>1- Devuelve solo las rutas siguiente a la fecha actual</p> <p>2- Devuelve solo las rutas anteriores a la fecha actual</p>	<p>OK -> Devuelve un json con los datos de la ruta pasada por parametro</p> <p>ERROR 1 -> Error al acceder a la base de datos (sin opcion)</p> <p>ERROR 2 -> Error al acceder a la base de datos (opcion 2)</p> <p>ERROR 3 -> Error al acceder a la base de datos (opcion 1)</p>
<p>POST /rutas/guardar</p> <hr/> <p>Guarda la ruta con los datos pasados por parametro</p>	<p>Parametros -> titulo,descripcion,fecha,hora,duracion,inicio,llegada,capacidad,apuntados,guia,precio,_id(opc)</p> <p>Devuelve -> 0 (Todo OK)</p> <hr/> <p>OK 0 -> Ruta guardada correctamente</p> <p>ERROR 1 -> Error al guardar la nueva ruta</p> <p>ERROR 2 -> Error al buscar la ruta por la _id</p> <p>ERROR 3 -> Error al modificar la ruta</p>
<p>POST /rutas/eliminar</p> <hr/> <p>Elimina la ruta de la base de datos con el _id pasado por parametro</p>	<p>Parametros -> _id</p> <p>Devuelve -> 0 (Todo OK)</p> <hr/> <p>OK 0 -> Ruta eliminada correctamente</p> <p>ERROR 1 -> Error al intentar eliminar la ruta</p>
<p>POST /rutas/crearRuta</p> <hr/> <p>Guarda la ruta con los datos pasados por parametro</p>	<p>Parametros -> titulo, descripcion,fecha,hora,duracion,inicio,llegada,capacidad,apuntados,guia,precio</p> <p>Devuelve -> 0 (Todo OK)</p> <hr/> <p>OK 0 -> Ruta guardada correctamente</p> <p>ERROR 1 -> Error al guardar la nueva ruta</p>
<p>POST /rutas/modificarRuta</p> <hr/> <p>Modifica la ruta con los parametros y el id pasados</p>	<p>Parametros -> titulo, descripcion,fecha,hora,duracion,inicio,llegada,capacidad,apuntados,guia,precio,_id</p> <p>Devuelve -> 0 (Todo OK)</p> <hr/> <p>OK 0 -> Ruta guardada correctamente</p> <p>ERROR 1 -> Error al guardar la ruta modificada</p> <p>ERROR 2 -> Error al realizar la petición a la base de datos</p>
<p>POST /rutas/actualizarApuntados</p> <hr/> <p>Modifica el numero de apuntados sumando o restando dependiendo de los parametros</p>	<p>Parametros -> opcion (1- Sumar, 2- Restar), num, _id</p> <p>Devuelve -> 0 (Todo OK)</p> <hr/> <p>OK 0 -> Apuntados actualizado correctamente</p> <p>ERROR 1 -> Parametros no recibidos</p> <p>ERROR 2 -> Error al realizar la petición a la base de datos</p> <p>ERROR 3 -> Ruta llena</p> <p>ERROR 4 -> Error al insertar los nuevos datos en la bse de datos</p>
<p>GET /rutas/estaLlena</p> <hr/> <p>Comprueba si la ruta esta llena o no</p>	<p>Parametros -> _id</p> <p>Devuelve -> 0 (No esta llena) o 1 (Esta llena)</p> <hr/> <p>OK 0 -> Ruta no llena</p>

OK 1 -> Ruta llena

ERROR 2 -> Error al realizar la petición a la base de datos

API de Comentarios

	Llamada	Uso
	GET /comentarios/listar Devuelve todos los comentarios de la base de datos en un vector	Parametros -> X Devuelve -> Vector OK -> Devuelve un vector con el listado de los comentarios ERROR 1 -> Error al acceder a la base de datos
	GET /comentarios/recuperar_comentario_por_id Devuelve el comentario con la id pasada por parametro	Parametros -> _id Devuelve -> json con todos los datos del comentario OK -> Devuelve un json con los datos del comentario pasado por parametro ERROR 1 -> Error al acceder a la base de datos
	POST /comentarios/guardar Guarda el comentario con los datos pasados por parametro	Parametros -> usuarioCom, nombreUsuarioCom, rutaCom, comentario, fecha, _id(opc) Devuelve -> 0 (Todo OK) OK 0 -> Comentario guardado correctamente ERROR 1 -> Error al guardar el nuevo comentario ERROR 2 -> Error al buscar el comentario por la _id ERROR 3 -> Error al modificar el comentario
	POST /comentarios/eliminar Elimina el comentario de la base de datos con el _id pasado por parametro	Parametros -> _id Devuelve -> 0 (Todo OK) OK 0 -> Comentario eliminado correctamente ERROR 1 -> Error al intentar eliminar el comentario
	GET /comentarios/listarComentariosPorRuta Devuelve todos los comentarios que coincida con la ruta pasada por parametro	Parametros -> rutaCom Devuelve -> Vector OK -> Devuelve un vector con el listado de los comentarios ERROR 1 -> Parametro rutaCom no recibido ERROR 2 -> Error al acceder a la base de datos

API de Reservas

	Llamada	Uso
	GET /reservas/listar 	Parametros -> X Devuelve -> Vector

	Devuelve todas las reservas de la base de datos en un vector	OK -> Devuelve un vector con el listado de las reservas ERROR 1 -> Error al acceder a la base de datos
	GET /reservas/listarReservasPorUsuario <hr/> Devuelve todas las reservas de la base de datos en un vector del usuario pasado por parametro	Parametros -> usuario Devuelve -> Vector <hr/> OK -> Devuelve un vector con el listado de las reservas ERROR 1 -> Error al acceder a la base de datos ERROR 2 -> Parametro no recibido
	GET /reservas/listarReservasPorRuta <hr/> Devuelve todas las reservas de la base de datos en un vector de la ruta pasado por parametro	Parametros -> idRuta Devuelve -> Vector <hr/> OK -> Devuelve un vector con el listado de las reservas ERROR 1 -> Error al acceder a la base de datos ERROR 2 -> Parametro no recibido
	GET /reservas/listarReservasPorGuia <hr/> Devuelve todas las reservas de la base de datos en un vector del guia pasado por parametro	Parametros -> guia Devuelve -> Vector <hr/> OK -> Devuelve un vector con el listado de las reservas ERROR 1 -> Error al acceder a la base de datos ERROR 2 -> Parametro no recibido
	GET /reservas/recuperar_reserva_por_id <hr/> Devuelve la reserva con la id pasada por parametro	Parametros -> _id Devuelve -> json con todos los datos de la reserva <hr/> OK -> Devuelve un json con los datos de la reserva pasado por parametro ERROR 1 -> Error al acceder a la base de datos
	POST /reservas/guardar <hr/> Guarda la reserva con los datos pasados por parametro	Parametros -> rutaRes, usuarioRes, fecha, puntuacion, _id(opc) Devuelve -> 0 (Todo OK) <hr/> OK 0 -> Reserva guardada correctamente ERROR 1 -> Error al guardar la nueva reserva ERROR 2 -> Error al buscar la reserva por la _id ERROR 3 -> Error al modificar la reserva
	POST /reservas/eliminar <hr/> Elimina la reserva de la base de datos con el _id pasado por parametro	Parametros -> _id Devuelve -> 0 (Todo OK) <hr/> OK 0 -> Reserva eliminada correctamente ERROR 1 -> Error al intentar eliminar la reserva
	POST /reservas/crearReservaUsuario <hr/> Guarda la reserva con los datos pasados por parametro en la pagina de ruta, cuando un usuario solicita una ruta	Parametros -> rutaRes, usuarioRes Devuelve -> 0 (Todo OK) <hr/> OK 0 -> Reserva guardada correctamente

		ERROR 1 -> Error al guardar la nueva reserva ERROR 2 -> Parametros no recibidos
GET /reservas/puedeReservar <hr/> Comprueba si el usuario puede realizar reservas de la ruta pasada por parametro	Parametros -> rutaRes, usuarioRes Devuelve -> 0 -> Si puede reservar, obj (_id) -> No puede reservar y devuelve un objeto con la _id <hr/> OK 0 -> Si puede reservar OK object(_id) -> No puede reservar y devuelve un object con la _id de la reserva ERROR 1 -> Parametros no recibidos ERROR 2 -> Error al realizar la busqueda en la base de datos	
GET /reservas/puedeVotar <hr/> Comprueba si el usuario puede realizar votos en la ruta	Parametros -> _id Devuelve -> 0 -> Si puede votar, 1 No puede votar, 2 -> Ya ha votado <hr/> OK 0 -> Si puede votar OK 1 -> No puede votar OK 2 -> Ya ha votado ERROR 3 -> Parametro _id no recibido ERROR 4 -> Error al realizar la busqueda en la base de datos	
GET /reservas/actualizarVoto <hr/> Introduce en la base de datos el voto realizado por el usuario	Parametros -> _id, num Devuelve -> 0 OK <hr/> OK 0 -> Actualizacion del voto en la base de datos realizado correctamente ERROR 1 -> Parametros no recibidos ERROR 2 -> Error al realizar la peticion a la base de datos ERROR 3 -> Reserva no encontrada ERROR 4 -> La reserva no se ha guardado correctamente al insertar los datos en la db	

API de FotosRuta

	Llamada	Uso
	GET /fotosRuta/listar <hr/> Devuelve todos las fotos de las rutas de la base de datos en un vector	Parametros -> X Devuelve -> Vector <hr/> OK -> Devuelve un vector con el listado de las fotos ERROR 1 -> Error al acceder a la base de datos
	GET /fotosRuta/recuperar_fotoRuta_por_id <hr/> Devuelve la foto con la id pasada por parametro	Parametros -> _id Devuelve -> json con todos los datos de la foto de ruta <hr/> OK -> Devuelve un json con los datos da la foto pasado por parametro ERROR 1 -> Error al acceder a la base de datos
		Parametros -> idRuta

	GET /fotosRuta/recuperarFotoRutaPorIdRuta	Devuelve -> un vector con datos json con los datos de las fotos de ruta
	Devuelve las fotos con la id de ruta pasada por parametro	OK -> Devuelve un vector con los datos de las fotos perteneciente a la idRuta pasada por parametro ERROR 1 -> Parametros no recibidos ERROR 2 -> Error al acceder a la base de datos
	POST /fotosRuta/guardar	Parametros -> url, idRuta, _id(opc) Devuelve -> 0 (Todo OK)
	Guarda el comentario con los datos pasados por parametro	OK 0 -> Foto de Ruta guardada correctamente ERROR 1 -> Error al guardar la nueva foto de Ruta ERROR 2 -> Error al buscar el la foto por _id ERROR 3 -> Error al modificar los datos de la foto
	POST /fotosRuta/eliminar	Parametros -> _id Devuelve -> 0 (Todo OK)
	Elimina la foto de ruta de la base de datos con el _id pasado por parametro	OK 0 -> Foto de ruta eliminado correctamente ERROR 1 -> Error al intentar eliminar la foto de la ruta
	POST /fotosRuta/subirFotosRuta	Parametros -> file Devuelve -> redirección a gestionarRutas.html
	Sube al servidor la foto pasada por el parametro file del formulario	OK 0 -> Redirige a gestionarRutas.html ERROR 1 -> Foto no guardada correctamente

API de Twitter

	Llamada	Uso
	GET /twitter/recogerDatos	Parametros -> user Devuelve -> json con datos de la sesion
	Si el usuario esta logeado con twitter devuelve los datos de la sesion	OK -> Devuelve un json con los datos de la sesión
	GET /login/twitter	Parametros -> XXX Devuelve -> una redireccion dependiendo del estado del login
	Se realiza el login con twitter. Se redirige a una web de twitter donde de realizara la autentificación.	OK -> Devuelve una redirección
	GET /twitter/twittearComentario	Parametros -> usuario, texto Devuelve -> 0 -> OK
	Se escribe un tweet en twitter con el contenido del comentario.	OK 0 -> Comentario enviado correctamente ERROR 1 -> Parametros no recibidos ERROR 2 -> Error al realizar la petición al servidor

ERROR 3 -> Usuario no encontrado
ERROR 4 -> Usuario no logeado con twitter
ERROR 5 -> Error al intentar escribir el tweet

API de Paypal

	Llamada	Uso
	GET /paypal/realizarPago Envia la informacion para que paypal realice el pago	Parametros -> usuario Devuelve -> json con los datos de la peticion paypal realizada OK -> Devuelve un json con los datos de paypal ERROR 1 -> Error al acceder a la base de datos ERROR 2 -> Error al intentar realizar el pago
	GET /paypal/retirarSaldo json con los datos de la peticion paypal realizada	Parametros -> usuario Devuelve -> json con los datos de la peticion paypal realizada OK -> Devuelve un json con los datos de paypal ERROR 1 -> Error al acceder a la base de datos ERROR 2 -> Error al intentar retirar el pago